甲南大学　博士学位論文

自動定理証明における補助命題の実装について
-幾何学基礎論を題材として-

甲南大学大学院
自然科学研究科　知能情報学専攻

2023 年 3 月

岩間詞也

目 次

概 要

自動定理証明 (ATP) は使用すれば証明が実行されるわけではない. ATP を用いて証明を実行するにはソフトウェアが認識可能な形で理論に沿った条件を構築しなければならない. これまで人間が ATP を用いずに行ってきた証明と, ATP を用いた証明には差異が存在し, それを解消するための技法は研究の途上にある. 定理証明とは, 数学的対象と関係の構築を含む活動として提示される. そこに ATP を活用するには, それらを表現する手順と関連する概念についての洞察が必要である.

本論文では, 第 1 章において, ATP およびその成り立ちについて示し, ATP が扱える各種問題について実装例を交えて示す. 第 2 章において, ATP 機能を有するソフトウェア (定理証明支援系) の現状と補助命題の必要性を明らかにする. 第 3 章では, 題材として論理学の中でも特に古い歴史と正確性を有するユークリッド原論およびヒルベルトの幾何学基礎論について, 定理証明支援系の一つである Isabelle/HOL を用いて, 補助命題の実装を行い, その過程において生じる問題およびその解決について示す. 最後に第 4 章において, 本論文のまとめと今後の課題を示す.

1 はじめに

初等幾何学という分野の始まりとして, ユークリッド原論で著されたユークリッド幾何学という数学体系が存在する. ユークリッド原論では幾何学における基礎的概念に対する定義, および一連の公理, 公準が与えられ, 公理系が確立された. また, 点, 線, 面などの定義が与えられ, それらを基に理論体系が構築されている. それに対しヒルベルトはユークリッド原論におけるこれらの定義が数学的に厳密ではないとして, 公理の中でそれらの関係性のみを構築することで厳密化を図った. 一方で ATP は, こうした数学的定理の妥当性, 整合性を機械的に確認する目的で開発されたものである. しかしながらヒルベルトによる幾何学基礎論 (以下, 幾何学基礎論と記す) の証明を ATP を用いて試みた結果, そのままでは証明できない箇所が散見されることが判明した. これらの問題を解決するために補助命題を作成することが, 本研究の目的である.

1.1 自動定理証明

自動定理証明 (Automated Theorem Proving 以下, ATP と記す) とは, 論理学の法則に従って, 正式な推論の機械化を試みる分野である. この分野のルーツは, フレーゲが数学の基礎として適した形式言語を開発する最初の包括的な取り組みである Begriffsschrift[1] を開発した十九世紀の終わりにさかのぼる. しかし, ラッセルによりフレーゲのシステムが一貫していないことを示すパラドックス (自身を含まない集合全体の集合の矛盾) が発見された. ラッセル達は, タイプ理論に基づいて独自のシステムを考案し,「Principia Mathematica」にて, それが数学の基礎としてどのように役立つかを示した. その後, ヒルベルトは自然数論や実数論などの数学の理論を記号化した「公理系」と「推論規則」からなる 「形式的体系」として考察することを提案した. 自然演繹においては, 各論理接続詞の意味が推論規則によって説明される. 定理証明において我々が持つ基本的な概念は, 証拠から導出される事実を認めることである [1].

現実の事象に例えるならば, 雨空を見上げて「雨」という視覚的な証拠を入手し, それに基づいて「雨が降っている」という判断を下すようなものである.

判断と命題の概念の分離, およびそれに対応する証拠と証明の概念の分離は, 論理システムを定義するために使用されてきた様々な形式に影響を及ぼしている. 特に, ヒルベルトの提唱した体系は, 仮想的な認識を使用せずに「事象 A が真である」という判断を定義できるものである. このような定義付けは, 判断を使用する上で役立つ. ただし, 公理における含意の自由な使用によって, 論理全体に破綻が生じていないことを常に補償しなければならない. 判断が直ちに自明になるものでなく,「A は B を伴う」といった含意を持つ場合, その導出には多くの仮説が必要になることが多い. つまり, 各公理の結合にこそ重点を置く必要がある [1].

多くの ATP で採用されている手順は, 提案された定理から公理に向かって逆方向に作業するか, 公理から定理に向かって順方向に作業するかのどちらかに分類できる. これらの組み合わせにより, 定理証明の効率が向上することを期待するのは自然なことである.

[1] concept notation

1.2 問題例

本節では, どのような問題が ATP で扱えるのかを示す. 以下は ATP の一つである Isabelle による証明の例である.

```
theorem
  fixes "n"::"nat"
  shows "odd (n^3+n^2+1)"
proof
  assume "even (n^3+n^2+1)"
  then have "odd (n^3+n^2)"by auto
  then have "odd (n^2*(n+1))"by auto
  then have "odd (n^2) ∧ odd (n+1)" by auto
  then have "odd n ∧ even n"by auto
  then show False by auto
qed
```

これは「自然数 n (Isabelle では 0 も自然数に含まれる) について n^3+n^2+1 は奇数である」という問題の証明を行っている. 4 行目の「proof」以下が実際の証明過程となり, 各行の意味は次のようになる.

- 5 行目 : 「n^3+n^2+1 が偶数である」とする (背理法の指定)
- 6 行目 : 「n^3+n^2」は (偶数である「n^3+n^2+1」から 1 を引いたものなので) 奇数である
- 7 行目 : (各項を共通因数 n^2 でくくり出した)「$n^2*(n+1)$」もまた奇数である
- 8 行目 : (積が奇数となるのは奇数*奇数の場合のみなので)「n^2」,「$n+1$」が共に奇数となる
- 9 行目 : (「n^2」が奇数になるのは n が奇数の場合であり,「$n+1$」が奇数になるのは n が偶数の場合なので) n は奇数でありかつ偶数である
- 10 行目 : (9 行目のような n は存在しないため) 矛盾が導かれた
- 11 行目 : 証明完了

次のような言葉 (記号) だけを使って作られる文を考える.

- 有理係数の多変数多項式
- 等号, 不等号, $\wedge$ (かつ), $\rightarrow$ (ならば), $\forall$ (すべて), $\exists$ (存在) などの論理記号

  (ただし変数は実数上を動くものとする)

例として,「$\forall x.\, x^2 \geq 0$」と書けば,「全ての実数 x について $x^2 \geq 0$」を意味する. このような言語で書ける文を, ここでは仮に R 文と呼ぶこととする. ここで, どんな R 文 ϕ についても, 次のことが成り立つ.

- ϕ は限量子 $\forall, \exists$ を含まない文に変形できる (限量子除去)
- ϕ は真である $\Longleftrightarrow$ ϕ は実閉体の公理系 RCF から証明できる (完全性)
- ϕ が真かどうかは機械的に判定できる (決定可能性)

ゆえに, このような R 文については自動証明がある程度機能することが分かる (ただし変数の個数が増えるに連れ, 計算時間が二重指数関数的に増加する恐れがある) [2].

1.3 限量子除去を用いた証明

ATP による限量子除去の一例を示すため, 次のような問題を考える.

$$\frac{\forall x.\, P \to Q}{P \to \forall x.\, Q}$$

これは,「全ての実数 x について P が成り立つとき Q も成り立つならば, P が成り立つとき全ての実数 x について Q が成り立つ」を意味する. この問題を Isabelle 上で表現すると以下のようになる.

```
lemma "(∀x. P ⟶ Q x) ⟹ P ⟶ (∀x. Q x)"
```

Isabelle には多くの推論規則が関数として組み込まれている. その一部を以下に示す.

- impI : (∃P ⟹ ∃Q) ⟹ ∃P ⟶ ∃Q
- allI : (∧x. ∃P x) ⟹ ∀x. ∃P x
- spec : ∀x. ∃P x ⟹ ∃P ?x
- mp : ⟦∃P ⟶ ∃Q; ∃P⟧⟹ ∃Q

Isabelle ではこれらを指定することで限量子除去を行う過程を表示させることができる. その実装と, 各関数実行後の問題 (Isabelle では subgoal (未解決の副題) と表示される) を以下に示す.

```
lemma "(∀x. P ⟶ Q x) ⟹ P ⟶ (∀x. Q x)"
  apply (rule impI)
  apply (rule allI)
  apply (drule spec)
  apply (drule mp)
  apply auto
  done
```

- 初期状態 : 1. ∀x. P ⟶ Q x ⟹ P ⟶ (∀ x. Q x)
  「全ての実数 x について P が成り立つとき $Q\,x$ も成り立つならば, P が成り立つとき全ての実数 x について $Q\,x$ が成り立つ」
- impI 実行後 : 1. ∀x. P ⟶ Q x ⟹ P ⟹ ∀ x. Q x
  「全ての実数 x について P が成り立つとき $Q\,x$ も成り立つならば, P が成り立つならば全ての実数 x について $Q\,x$ が成り立つ」
- allI 実行後 : 1. ∧x. ∀x. P ⟶ Q x ⟹ P ⟹ Q x
  「任意の実数 x について, 全ての x について P が成り立つならば $Q\,x$ も成り立つとき, P が成り立つならば $Q\,x$ が成り立つ」
- spec 実行後 : 1. ∧x. P ⟹ P ⟶ Q (?x4 x) ⟹ Q x
  「任意の実数 x について P が成り立つならば, P が成り立つとき Q の項 $x4$ に x を代入したものが成り立つならば $Q\,x$ が成り立つ」
- mp 実行後 :
  1. ∧x. P ⟹ P
  2. ∧x. P ⟹ Q (?x4 x) ⟹ Q x

  「任意の実数 x について P が成り立つならば, P が成り立つ」,
  「任意の実数 x について P が成り立つならば, Q の項 $x4$ に x を代入したものが成り立つならば $Q\,x$ が成り立つ」

最初の命題が限量子を含まない二つの副題へと分解されたことが確認できる (それぞれ「P ⟹ P」,「Q x ⟹ Q x」と言い換えられるため, 直ちに証明が完了する).

当然ながら, 実数の範囲のみで扱える問題は多くない. 自然数 (整数) や関数, 集合等が自由に扱えて初めて数理論理学に対応するといえる. 一方で, R 文の言語には個々の自然数 (0, 1, 2, . . .) は

含まれているものの, それらから「全ての自然数について」を主張することはできない (数学基礎論やコンピュータ科学では 0 も自然数に含める). 「関数」や「集合」一般についても同様である.

1.4 数学的帰納法を用いた証明

対称を表す変数 x, y, z に対して, 集合を表す変数 X, Y, Z を使い,「x が X に属する」という意味を, 記号を用いて「$x \in X$」のように表すことで, 最低限の集合概念を用いることも可能である. たとえば「n が自然数である」ことは「$\forall X.\,(0 \in X \wedge \forall x.\,(x \in X \to x+1 \in X) \to n \in X)$」と書き表せこれが意味するのは,「0 を含み +1 について閉じているどんな集合 X にも n は属する」. より簡潔に表現するなら,「数学的帰納法を使えるのが自然数だ」となる [2].

例として, 問題「n が 1 以上の自然数であるとき, 2^$n > n$」を Isabelle 上で数学的帰納法を用いて証明する様子を以下に示す.

```
theorem
  fixes n :: "nat"
  shows "2^(Suc n) > Suc n"
proof (induct n)
  case 0
  show "2^(Suc 0) > Suc 0" by simp
next
  fix k :: "nat"
  assume "2^(Suc k) > Suc k"
  then have "2*2^(Suc k) > 2*(Suc k)" by simp
  then have "2^((Suc k)+1) > 2*(Suc k)" by simp
  then have "2*(Suc k) ≥ ((Suc k)+1) ⟹ 2^(Suc (Suc k)) > (Suc (Suc k))" by simp
  then have P1 : "(Suc k) - 1 ≥ 0 ⟹ 2^(Suc (Suc k)) > (Suc (Suc k))" by simp
  have P2 : "(Suc k) - 1 ≥ 0" by simp
  from P1 P2 show "2^(Suc (Suc k)) > (Suc (Suc k))" by simp
qed
```

Isabelle には 0 を含む自然数 (nat) が標準概念として組み込まれている (Suc n は Isabelle の自然数 nat における $n+1$ の意). これは, 証明したい問題に対して $n = 1$ の場合を証明し, 続いて $n = k$ について, この命題が成り立つという仮定の元, $n =$ (Suc k) でも成り立つことを証明するという最も基本的な数学的帰納法の形式である.

証明部分 (4 行目 proof (induct n) 以下) の各行の意味は次のようになる.

- 4 行目 : n に関する帰納法の使用を宣言
- 5 行目 : $n = 0$ のとき (命題中の n が全て (Suc n) であるため, $n = 1$ のときを意味する)
- 6 行目 : 2^1 > 1 は自明.
- 7, 8, 9 行目 : 次に $n = k$ について, この命題が成り立つと仮定する
  (以下, 簡便のため Suc k を k と書く)
- 10, 11 行目 : 両辺に 2 を掛けて左辺を整理
- 12 行目 : $2k \geq k+1$ ならば $n = k+1$ について命題が成り立つ
- 13 行目 : (左辺を整理して) $k - 1 \geq 0$ ならば $n = k+1$ について命題が成り立つ
- 14 行目 : $k - 1 \geq 0$ は自明
- 15 行目 : 13, 14 行目より, $n = k+1$ について命題が成り立つ
- 16 行目 : 証明完了

集合を用いて関数を表す, もしくは自然数全体を集合により定義する手法は既知のものである. ひいては, 上記で例示した R 文では集合を十全に扱えないことが問題といえる. ただし, 集合に対応できるように言語の拡張を試みると, それまで保証されていた都合の良い理論的性質 (限量子除去・完全性・決定可能性) はたちまち失われることとなる [2]. これは ATP の実装において回避不可能な制約である.

1.5 新規の定義を用いた証明

不完全あるいは決定不能であっても自動証明が全く適用できないというわけではない. 例として, Isabelle 上に特定の二項関係を新たに作成する様子を示す.

```
locale Rule_1 =
  fixes Relation :: "term ⇒ term ⇒ bool" (infixr "[#]" 50)
  assumes rev : "⟦P [#] Q⟧⟹ Q [#] P"
    and trans : "⟦P [#] Q; Q [#] R⟧⟹ P [#] R"
```

これは, 新たに与えた二項間の何らかの関係について, 対称律 (rev), 推移律 (trans) を与える様子である. 論理式の形式で表せば以下のようになる.

$$\frac{P\ [\#]\ Q}{Q\ [\#]\ P} \qquad \frac{P\ [\#]\ Q \quad Q\ [\#]\ R}{P\ [\#]\ R}$$

ここで対象律を与えずに推移律のみを許可すれば, 値の大小のような一方通行の関係を定義することも可能である. これらを用いた証明の例を以下に示す.

```
theorem (in Rule_1)
  assumes "P [#] Q" "R [#] Q"
  shows "P [#] R"
proof -
  have "Q [#] R" using assms by (simp add:rev)
  thus "P [#] R" using assms by (blast intro:trans)
qed
```

各行の意味は次のようになる.

- 1 行目 : 定理の開始と作成した二項関係 (Rule_1) 使用の宣言
- 2, 3 行目 : 仮定と証明したい結論 (⟦P [#] Q; R [#] Q⟧⟹ P [#] R)
- 4 行目 : 証明開始
- 5 行目 : 対称律 (rev) より, 仮定 (R [#] Q) から Q [#] R が導ける.
- 6 行目 : 推移律 (trans) より, 5 行目および仮定 (P [#] Q) から P [#] R が導ける.
- 7 行目 : 証明完了

論理学においては, 先の R 文で表現できるような論理式を一階述語論理, 自然数全体といった集合の概念を追加し, これを拡張したものを二階述語論理と呼ぶ. これらは性質が全く異なる. 中でももっとも顕著なのが完全性である [2].

- 全ての妥当な一階の文を導出できる完全な証明系が存在する (完全性)
- 全ての妥当な二階の文を導出できる完全な証明系は存在しない (不完全性)

二階およびそれ以上の述語論理 (高階述語論理) の ATP 実装を挑戦することについては未だに試行段階である. そのため本研究では Isabelle/HOL を用いて一階述語論理で証明できる初等幾何学の証明支援を示す.

自動証明においては「目標定理の否定を仮定して公理と矛盾することを示す」ことが有効である場合が多い. たとえば「前提 P から Q を結論してよい」という定理の証明を目標としたとき, その否定として「前提 P から $\neg Q$ を結論してよい」を仮定して, これが公理と矛盾することを示せばよいことになる. 一方で目標定理が「$\forall x.\phi(x)$ から $\phi(t)$ を結論してよい」のようになると, この否定に対してどのような t を選べば矛盾に到達できるかは全く定かではない [2].

2　定理証明支援系

前節で触れた Isabelle を含む, ATP 機能を持ったソフトウェアを一般に定理証明支援器もしくは定理証明支援系と呼ぶ [3]. これらが開発された背景には, 人間が書いた数学の証明とコンピュータを用いて行った数学の証明の両方の「正しさ」を機械的に保証してくれる存在が求められたことが挙げられる. 定理証明支援系の中核となる部分 (カーネルという) は小さいため, その部分にバグがある可能性は低く, またそれ自体が形式論理の長い研究を経て選定された公理であるので, 理論的に誤った結論を導かないことが十分確認されている. しかしながら, その高い汎用性と信頼性の一方で, 証明の作成の際, すべての証明手続きを明確に形式化しなければならないという課題を抱えている [3].

定理証明支援系では, 対話的に証明を構成する. 定理証明支援系に含まれるソフトウェアは数多く存在するため, それら個々のソフトウェアについて詳しい言及は行わない. ここではその中でも特に現在ユーザが多く, 盛んに研究の対象となっている定理証明支援系 Coq を例に, 証明の作成過程を示す.

また他にも本研究の過程では, 多くの幾何学的証明に対応するといわれる動的数学ソフトウェア GeoGebra を用いた証明を検討した [4]. しかし GeoGebra はユーザが入力した局所的状況に対する証明には有効に機能する一方で, 与えた条件に対して常に真偽が判定可能であるかという問題には不向きであり, 扱える証明の種類に関して深い調査が求められることが判明した.

2.1　Coq

Coq は GNU LGPL ライセンスで配布されているフリーソフトウェアである. Calculus of constructions(英語版) という高階型システム (Thierry Coquand と Gérard Huet が 1984 年に創始したもので, 英語では CoC と略せてシステム名 Coq に至る) に基づく. 正しい証明は正しく型がつくラムダ式であるというカリー＝ハワード同型対応を利用しているので, Coq の証明言語は型付きラムダ計算の一種である. 1991 年以降 Coq が用いている Calculus of Constructions の変種は,Christine Paulin-Mohring によるもので, 帰納的構成を直接含み, Calculus of Inductive Constructions (CIC) という名前がついている. Coq により達成された有名な証明には, 四色定理 (2004 年) や, フェイト・トンプソンの定理 (奇数位数定理) (2012 年 9 月) などが存在する.

Coq は数学的な定義, 実行可能なアルゴリズムおよび定理について, マシンでチェックされた証明を半対話的に開発, 記述することを目的としている [5].

Coq では, まず言明の入力によって証明のゴールが定まる. 次にユーザが証明のスクリプトの記述を始め, その証明の最初のステップを入力すると, 元々のゴールに辿り着くための新たなゴールが返される. ユーザはそれらのゴールに対してスクリプトの記述を続ける. このプロセスを繰り返すことで段階的に証明の作成が続き, 最終的にゴールが返らなくなる. その時点で証明が完了し, ソフトウェアによる保証が付く.

2.1.1　Coq 使用例

始めに Coq による簡単な証明例を示す.

$$\dfrac{\dfrac{P \quad Q}{R} \qquad \dfrac{P}{Q}}{R} \qquad P$$

これは,「P および Q が成り立つときに R が成り立ち, P が成り立つとき Q が成り立つならば, P が成り立つとき R が成り立つ」を意味する. この問題を Coq の上で表現すると以下のようになる.

```
Lemma test1 (P Q R : Prop) : (P -> Q -> R) -> (P -> Q) -> P -> R.
intro H1.
```

```
intro H2.
intro H3.
specialize (H1 H3).
specialize (H2 H3).
specialize (H1 H2).
trivial.
Qed.
```

Coq には証明中の指定した行での進行状況を表示させる機能が存在する. 例として, 上記の入力 4 行目「intro H3.」まで実行させた場合の証明進行状況は以下のように示される.

```
1 subgoal
P, Q, R : Prop
H1 : P -> Q -> R
H2 : P -> Q
H3 : P
_____________________________(1/1)
R
```

証明すべき副題 (subgoal) が 1 つであること, 変数 P, Q, R の型について (Prop は命題の意), 前提条件 (H1〜H3), そして最終的に証明すべき命題 R (= True) がそれぞれ表示されている. 入力 5 行目「specialize (H1 H3).」は,「H3 を H1 で簡約化する」というコマンドであり, それによって次の進行状況は以下のようになる.

```
1 subgoal
P, Q, R : Prop
H1 : Q -> R
H2 : P -> Q
H3 : P
_____________________________(1/1)
R
```

「P および Q が成り立つときに R が成り立つ」という前提 H1 が,「P が成り立つ」という前提 H3 により,「Q が成り立つときに R が成り立つ」に書き換わった. 入力 6, 7 行目でも同様に前提の書き換えが行われる.

```
1 subgoal
P, Q, R : Prop
H1 : R
H2 : Q
H3 : P
_____________________________(1/1)
R
```

入力 8 行目「trivial.」により, 前提と subgoal の比較が行われる. 今回の場合, 前提 H1 がこれと一致し, またそれにより全ての副題が解決されたことで, 以下のような表示に切り替わる.

```
No more subgoals.
```

入力 9 行目「Qed.」により, 証明完了. このように, 証明一行毎の進行状況を精査できることが, Coq の大きな特徴である.

また Coq にも Isabelle と同様に, 特定の項関係を新たに定義する機能が存在する. ここでは比較の為, 先述したものと全く同じ二項関係および問題を Coq の上に実装する様子を示す.

```
Inductive Relation : Set -> Set -> Prop :=
```

```
| rev : forall (X Y : Set), Relation X Y -> Relation Y X
| trans : forall (X Y Z : Set), Relation X Y -> Relation Y Z -> Relation X Z.

Notation "( X [#] Y )" := (Relation X Y).

Lemma test2 (P Q R : Set): (P [#] Q) -> (R [#] Q) -> (P [#] R).
intro H1.
intro H2.
specialize(rev R Q).
intro H3.
specialize(H3 H2).
specialize(trans P Q R).
intro H4.
specialize(H4 H1).
specialize(H4 H3).
trivial.
Qed.
```

Inductive から始まる三行がユーザ定義の二項関係を作る関数であり, 直後の Notation 文は以降の入力および表示を易くするための書き換えを指示する関数である. Lemma から始まる証明を前提条件の読み込み (3 行目, intro H2.) まで実行した状態が以下のようになる.

```
1 subgoal
P, Q, R : Prop
H1 : (P [#] Q)
H2 : (R [#] Q)
_____________________________________(1/1)
(P [#] R)
```

4 行目で作成した定義の使用を宣言している. ここで指定されている定義 rev は二個の引数 Set を必要とする (重複可) 関数であるため, ここではこの証明内で既に定義されている Set 変数 R, Q を指定している. この時点での証明状況が以下のようになる.

```
1 subgoal
P, Q, R : Prop
H1 : (P [#] Q)
H2 : (R [#] Q)
_____________________________________(1/1)
((R [#] Q) -> (Q [#] R)) -> (P [#] R)
```

このように Coq は指定された定義と変数によって導き出される推論を証明すべき命題に加えるような振る舞いを見せる. そのため続く 5 行目で, 新しく加えられたこの一文を前提条件として改めて読み込ませている.

```
1 subgoal
P, Q, R : Prop
H1 : (P [#] Q)
H2 : (R [#] Q)
H3 : (R [#] Q) -> (Q [#] R)
_____________________________________(1/1)
(P [#] R)
```

6 行目は先の証明例でも示したように, 前提 H3 を H2 で簡約化している. このようにして対称

律 (rev) を性質として持つ何らかの二項関係を Coq 上で扱うことができる.

7〜10 行目はこれと同様に推移律 (trans) を使用する一連の流れである. 8 行目まで, および 10 行目まで実行させた証明状況をそれぞれ以下に示す.

```
1 subgoal
P, Q, R : Prop
H1 : (P [#] Q)
H2 : (R [#] Q)
H3 : (Q [#] R)
H4 : (P [#] Q) -> (Q [#] R) -> (P [#] R)
_____________________________________(1/1)
(P [#] R)
```

```
1 subgoal
P, Q, R : Prop
H1 : (P [#] Q)
H2 : (R [#] Q)
H3 : (Q [#] R)
H4 : (P [#] R)
_____________________________________(1/1)
(P [#] R)
```

二度の簡約化を経た前提 H4 と証明すべき命題が一致したため, 以降の trivial 文により証明が完了される.

2.1.2 Coq 実装例

これまでの研究として, Coq と Isabelle を用いて, ユークリッド原論 [6] の定理を証明する活動を提案した. その目的は, ATP におけるプログラミングの再興である.

ここでは, [6] の Book1 の命題 1_2 : 「与えられた点において, 与えられた線分に等しい線分をつくること」を対象とする (命題 1_2 の証明には, 本来命題 1_1 の証明と, 円と円周上の点に関する定義が必要だが, ここでは省略する).

始めに, 点および線分を表す要素の定義を行う.

```
Definition Point := nat.
Definition Line := Point * Point : Set.
```

Point は自然数一つの型,Line は 2 つの Point からなる集合という形で定義される. Line A(線分 A) を定義したとき,“fst A”および “snd A”とすると, 線分 A を構成するそれぞれの点を指定できる.

続けて, 新たに定義した型に関する関数を宣言する.

```
Definition LPL := Line * Line : Set.

Inductive L_eq : Line -> Line -> Prop :=
| L_ref : forall (l : Line), L_eq l l
| L_rev : forall (l1 l2 : Line), L_eq l1 l2 -> L_eq l2 l1
| L_suii_1 : forall (l1 l2 l3 : Line), L_eq l1 l2 -> L_eq l2 l3 -> L_eq l1 l3
| L_P_rev_1 : forall (p1 p2 : Point), forall(l : Line), L_eq l (p1, p2) -> L_eq l (p2, p1).
```

4 つの関数を簡便に表記すると, 以下のようになる. 以下では, 点 A と点 B をつなぐ線分 AB を AB と表記する.

- L_ref : AB = AB (反射律)

- L_rev : AB = CD ならば CD = AB (対称律)
- L_suii_1 : AB = CD かつ CD = EF ならば AB = EF (推移律)
- L_p_rev_1 : AB = CD ならば AB = DC

同様にして, 必要な公理を関数として定義していく.

```
Definition LPL := Line * Line : Set.

Inductive LPL_eq : LPL -> LPL -> Prop :=
| LPL_ref : forall (lp : LPL), LPL_eq lp lp
| LPL_rev : forall (lp1 lp2 : LPL), LPL_eq lp1 lp2 -> LPL_eq lp2 lp1
| LPL_suii_1 : forall (lp1 lp2 lp3 : LPL), LPL_eq lp1 lp2 -> LPL_eq lp2 lp3 -> LPL_eq lp1 lp3
| LPL_trans_1 : forall (l1 l2 l3 l4: Line), LPL_eq lp1 lp2 -> LPL_eq lp2 lp3 -> LPL_eq lp1 lp3.

Definition LPL_trans_2 : forall(l1 l2 l3 l4 : Line), LPL_eq (l1, l2) (l3, l4) -> L_eq l1 l3 -> L_eq l2
l4.
```

LPL は線分+線分の型である. 各関数を簡便に表記すると, 以下のようになる (以下, 線分 AB+線分 CD を AB+CD と表記する).

- LPL_ref : AB+CD = AB+CD (反射律).
- LPL_rev : AB+CD = EF+GH ならば EF+GH = AB+CD (対称律).
- LPL_suii_1 : AB+CD = EF+GH かつ EF+GH = IJ+KL ならば AB+CD = IJ+KL (推移律).
- LPL_trans_1 : AB = CD かつ EF = GH ならば AB+EF = CD+GH.
  (原論の Book1 公理 2「等しいものに等しいものが加えられれば, 全体は等しい」より)
- LPL_trans_2 : AB+CD = EF+GH かつ AB = EF ならば CD = GH.
  (原論の Book1 公理 3「等しいものから等しいものがひかれれば, 残りは等しい」より)

以降, LPL_eq を略記できるようにしておく.

```
Notation "( x [@@] y )" := (LPL_eq x y).
```

証明に必要となる, 命題 1_1 :「与えられた線分の上に正三角形をつくること」(与えられた線分に, その 2 点それぞれと繋ぐことで正三角形となるような点をつくる) を, ここでは以下のような定義として実装する.

```
Definition def_Prop1_1 (l : Line) (p : Point) : Prop := L_eq l (fst l, p) /\ L_eq l (snd l, p).
```

“線分 AB と点 P が命題 1_1 の関係にある とは, AB = AP かつ AB = BP である” という内容になっている (今回の証明に必要な箇所のみを抜き出した形である).

ここから証明に入る. まずは改めて命題 1_2 の内容を示す.

命題 1_2. 与えられた点において与えられた線分に等しい線分をつくること.

- 仮定

a1. 与えられた点を A, 線分を BΓ とする.

a2. 線分 AB の上に等辺三角形 ΔAB をつくる (命題 1).

a3. ΔA を延長して線分 AE, ΔB を延長して線分 BZ をそれぞれ描く.

a4. 中心 B, 半径 BΓ をもって円を描き, BZ との交点を H とする.

a5. 中心 Δ, 半径 ΔH をもって円を描き, AE との交点を Λ とする.

- 証明

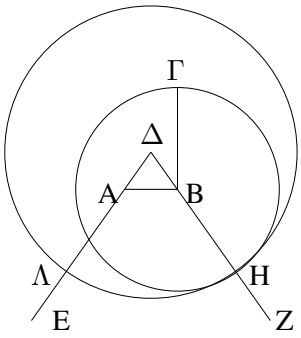

図 1: ユークリッド原論 命題 1_2

P1. a5 より, ΔΛ = ΔH (ΔA+AΛ = ΔB+BH).

P2. a4 より, BH = BΓ. よって ΔB+BH = ΔB+BΓ.

P3. P1 P2 より, ΔA+AΛ = ΔB+BΓ.

P4. a2 より, ΔA = ΔB.

S1. P3 P4 より, AΛ = BΓ.

・ S1 より, 点 A において線分 BΓ に等しい線分 AΛ がつくられている.

これらの仮定および結論を Coq 上で表現したものが次のようになる (図 2).

```
Variable A B Δ Γ H Λ : Point.
Proposition Prop1_2:
def_Prop1_1 (A,B) Δ ->
L_eq (B,Γ) (B,H) ->
L_eq (Δ,H) (Δ,Λ) ->
LPL_eq ((A,Δ),(A,Λ)) ((B,Δ),(B,H)) ->
L_eq (B,Γ) (A,Λ).
```

```
1 subgoal
s1, s2, s3 : Line
p1, A, B, Δ, Γ, H, Λ : Point
H0 : def_Prop1_1 (A, B) Δ
H1 : ((B, Γ) [@] (B, H))
H2 : ((Δ, H) [@] (Δ, Λ))
H3 : ((A, Δ, (A, Λ))
     [@@] (B, Δ, (B, H)))
______________________________________
((B, Γ) [@] (A, Λ))
```

図 2: 仮定および結論を Coq で表現したもの

前提 1. AB の上に等辺三角形 ΔAB となる Δ がつくられている.

前提 2. BΓ = BH.

前提 3. ΔH = ΔΛ.

前提 4. AΔ+AΛ = BΔ+BH.

結論. BΓ = AΛ.

(ゴールウィンドウは intros. を実行した状態)

H0 は論理積を含む前提である. このような前提は intros. では分離されないため, 以下のコマンドを使用する.

destruct H0.

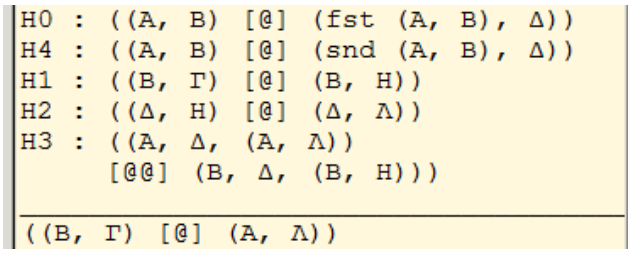

図 3: 前提の両辺を入れ替えたものの生成

図 3 のように, 前提 H0 が 2 つの新たな前提 H0, H4 に分解された. 次に, 後の書き換えの為に前提 H0 の両辺を入れ替えた前提を生成する.

specialize (L_rev (A, B) (fst (A, B), Δ)).
intro.
specialize (H5 H0).

対称律にあたる定義関数 L_rev に前提 H0 の引数 “(A, B)”と “(fst (A, B), Δ))”を指定し, 追加された前提 H5 を H0 で書き換える.

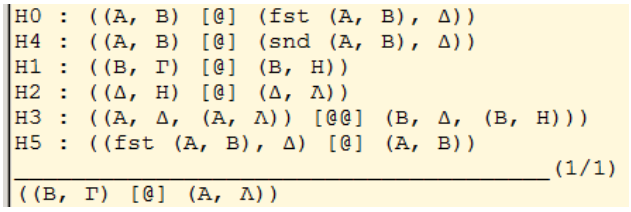

図 4: 追加された前提を書き換える

定義 “l1 = l2 → l2 = l1”, 前提 “AB = AΔ”より, 前提 H5 “AΔ = AB”が生成された.
次に, 前提 H5 と H4 を Line の推移律に当てはめることを考える.

specialize (L_suii_1 (A, Δ) (A, B) (B, Δ)).
intro.
specialize (H6 H5).
specialize (H6 H4).

定義関数 L_suii_1 に H5 と H4 の引数を指定, 追加された前提 H6 をそれぞれを指定して書き換える.
定義 “l1 = l2 → l2 = l3 → l1 = l3”, 前提 “AΔ = AB”, “AB = BΔ”より, 新たな前提 H6 “AΔ = BΔ”が生成された.
次に, 前提 H3 と H6 を LPL の定義関数に当てはめる.

specialize (LPL_trans_2 (A, Δ) (A, Λ) (B, Δ) (B, H)).
intro.
specialize (H7 H3).
specialize (H7 H6).

定義関数 LPL_trans_2 に H3 の引数を指定, 追加された前提 H7 を H3 と H6 で書き換える.

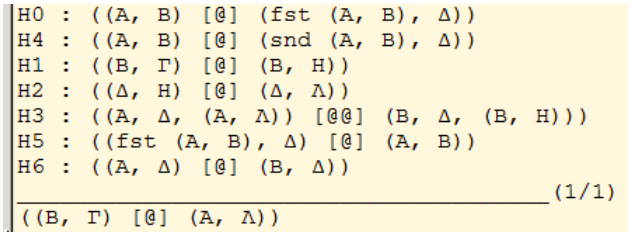

図 5: 新たな前提の生成

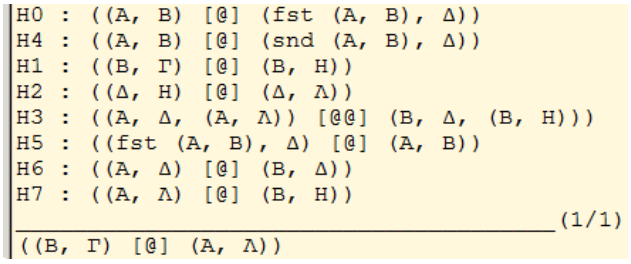

図 6: 新たな前提と両辺の書き換え

定義 “l1+l2 = l3+l4 → l1 = l3 → l2 = l4”, 前提 “AΔ+AΛ = BΔ+BH”,“AΔ = BΔ” より, 新たな前提 H7 “AΛ = BH” が生成された. 次に, 後の書き換えの為, 前提 H1 の両辺を入れ替える.

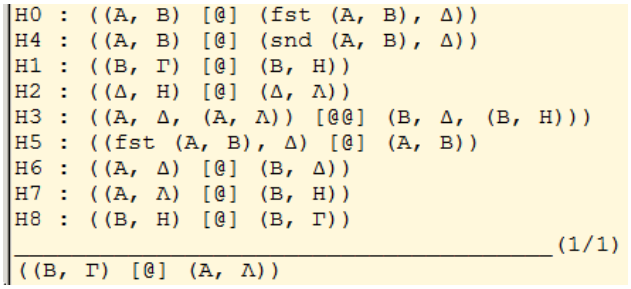

図 7: H1 の両辺入れ替え

次に, 前提 H7 と H8 を Line の推移律 (L_suii_1) に当てはめる.
最後に, 結論を前提 H9 に合わせた形に書き換える.

apply (L_rev (A, Λ) (B, Γ)).

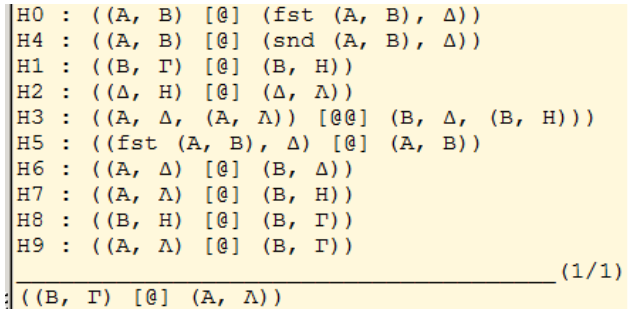

図 8: 推移律に当てはめる

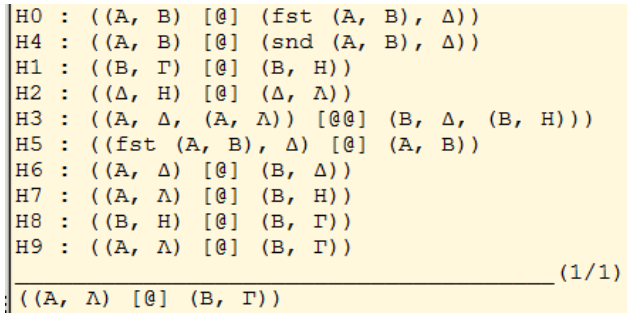

図 9: 証明の終了

前提の中に結論と合致するものが現れたので trivial で副題証明を終える. 全ての副題が証明されたので, Qed で証明を終了する.

最後に, 証明の全文を示す.

```
Variable A B Δ Γ H Λ : Point.
Proposition Prop1_2:
def_Prop1_1 (A, B) Δ ->
L_eq (B, Γ) (B, H) ->
L_eq (Δ, H) (Δ, Λ) ->
LPL_eq ((A, Δ), (A, Λ)) ((B, Δ), (B, H)) ->
L_eq (B, Γ) (A, Λ).
Proof. intros.
destruct H0.
specialize (L_rev (A, B) (fst (A,  B), Δ)). intro.
specialize (H5 H0).
specialize (L_suii_1 (A, Δ) (A, B) (B, Δ)). intro.
specialize (H6 H5).
specialize (H6 H4).
specialize (LPL_trans_2 (A, Δ) (A, Λ) (B, Δ) (B, H)). intro.
specialize (H7 H3).
specialize (H7 H6).
specialize (L_rev (B, Γ) (B, H)). intro.
specialize (H8 H1).
specialize (L_suii_1 (A, Λ) (B, H) (B, Γ)). intro.
specialize (H9 H7).
specialize (H9 H8).
apply (L_rev (A, Λ) (B, Γ)).
trivial.
Qed.
```

2.2 Isabelle

Isabelle は, もともとケンブリッジ大学とミュンヘン工科大学で開発されていた. Isabelle は入力に一般的な数学記号を使用することが可能であり, 作成した定義と証明を含むプログラムファイル

(.thy ファイル) を, タイプセットドキュメント (紙, 本, 論文) を自動的に生成できる LaTeX ソースに変換することを可能とする開発環境が提供されている. Isabelle には, 初等整数論 (ガウスの平方剰余の法則など), 分析 (極限の性質, 導関数, 積分), 代数 (Sylow の定理まで), 集合論 (Sylow の定理まで) など, 正式に検証された数学の大規模な理論ライブラリが付属している [7]. また, 数学とソフトウェアエンジニアリングの両方に由来する, 膨大な数の寄稿された検証研究を公式サイト (Archive of FormalProofs [8]) から閲覧可能である.

Isabelle の実装例については前章にて示してきた為, ここでは省略する.

Isabelle と Coq それぞれの証明機能における最大の差異は, 結論を導き出すのがユーザ側かソフトウェア側かという点である.

H_0〜H_n の条件から A という結論を導き出すとき, Coq の場合は注目する条件 H_i および H_j を指定し, 必要な処理を指示することで, ユーザはその処理の結果として結論 A を取得することになる.

一方で Isabelle の場合, 条件の指定までは共通 (簡易な証明であれば指定の省略も可能) であるが, その後ユーザは求める結論 A を指定し, これまでに構築した規則等からそれが導き出せるかという問いをソフトウェアへ尋ねる形となっている.

ATP という分野において, これらは決してどちらが優れているかと論議されるものではないが, より我々の目的に沿った機能を持つのはどちらかという点では確かな差となる. すなわち, 定理証明という見地により近くあるのはどちらであるかを比較し, 求める結論を先に指定する後者の証明形式であると考えた.

また, これまでの例で示した通り, Coq は細かな証明状況の変遷を追う機能に優れている反面, 行数が嵩むため証明が煩雑化し易いという一面を持っている.

さらに, Isabelle の持つ特徴として, 入力にエラーを発見した場合にそれを含む行の処理を飛ばして以降の文の実行を試みるという点がある. これを利用して Isabelle では, 疑似的に結論から逆向きの証明を完成させることが可能である. これは常に一行ずつ処理を行い, エラー原因が解決しない限り以降の文を認識しない Coq では起こり得ない.

この機能は, 人間の手で書かれた証明との間に散見される諸問題に対して, 何が解決すれば求める結論を導き出せるかを明確にするという点において有効であると予想した.

このような観点から, 本研究では使用する定理証明支援系として Isabelle を選択するに至った.

2.3 補助命題

本研究によって,『幾何学基礎論』[9] の証明の中には人間の認識に頼った証明が用いられ, そのままでは定理証明支援系の上に構築できない箇所が散見されることが判明した. 以下の図を基に, その一例を示す.

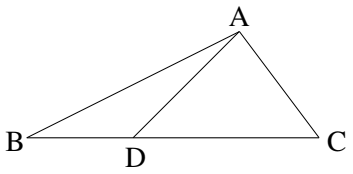

図 10: 例図

条件として, 三点 ABC が三角形であること, 点 D が線分 BC 上の点であると与えられているとする. このとき紙に書いて行う証明において, 三点 ABD や三点 ADC が三角形であるかを問われた

ならば, 多くの人間が即座に真であると判断するだろう. 実際に [9] の中でこれに当たる証明については明記されてはいない.

しかしながら, 定理証明支援系においてこのような箇所を含む証明を実装するには, それらが公理・定義・定理の上に三角形であると導けることを余さず証明する必要がある. これが, 人間の認識による証明と, 定理証明支援系への実装の間に生じる齟齬の一例である.

また, 既存公理からなる各定理および補題の証明を実装するにあたり,「これは公理や定義から明らかである」として, その証明が明文化されていない幾つかの事例に遭遇した. その殆どは考察するまでもなく導けるものであるが, Isabelle への実装を行う上では, その全てを詳らかに構築することが求められる.

一例として, ここでは二つの角の大小関係の唯一性について挙げる.

・二つの角 α および β に対して次の三つの場合

$$\alpha < \beta \wedge \beta > \alpha.$$

$$\alpha \equiv \beta.$$

$$\alpha > \beta \wedge \beta < \alpha.$$

のうちの唯一つが常に生起する.

[9] においては, 角の大小に関する定義を述べた後, 以上が直ちに主張できるとされている. こちらも人間の認識においては議論されるものではないが, 定理証明支援系の中で実装するには, それまでの公理や定義などから導出しなければならない.

そこで本研究ではこれらについて, 既存の公理・定義・定理から導出できる新たな命題を作成し, 幾何学基礎論の補完として明文化することを試みた. それら新たに作成した定理を, ここでは補助命題と呼ぶ.

なお本研究で実装された証明は, いずれも一階述語論理の範囲に留まっている. Isabelle/HOL で用いている論理は高階述語論理の体系である. その体系の表現としては、高階述語論理 > 一階述語論理 > 命題論理 となり, 命題論理は決定可能, 一階述語論理は半決定可能, 高階述語論理は決定不能である. そのため本研究における目的は, 幾何学基礎論において証明した人間の証明を Isabelle/HOL を用いて一階述語論理の証明として支援することを対象とする (一階述語論理は半決定可能であり, 機械的な証明を行うことはできず, 人間の手で証明することが必要である). さらに高階述語論理は決定不能なので Isabelle/HOL は使いものにならず, 人間が全て証明するしかないということではない. 人間の証明を支援することで証明全体の精度を高めることができる. そこに Isabelle/HOL で用いた自動定理証明の価値がある.

3　補助命題の実装 -幾何学基礎論を題材として-

本研究では, 現存する最古の大規模な数学の体系と呼ばれるユークリッド原論, ついでその再構築により生まれたとされる幾何学基礎論を題材として, それぞれを ATP 上で実装することを行った.

3.1　ユークリッド原論

ユークリッド原論は, 紀元前 3 世紀の数学者ユークリッドにより編纂された 13 冊からなる古代の数学書であり, そこでは定義, 公準 (要請), 公理 (共通理念), 命題 (定理と構造) および命題の数学的証明がまとめられている [6].

本研究では, ユークリッド原論の第 2 巻までの内容を Isabelle を用いて凡そ表現するに至った. ここでは, そのうちの第 1 巻 (Book1) までの内容を主に示す.

第 1 巻の前提 (定義・公準・公理) および各命題の内容が以下のようになる.

・定義

1. 点とは部分を持たないものである.
2. 線とは幅のない長さである.
3. 線の端は点である.
4. 直線とはその上にある点について一様に横たわる線である.
5. 面とはその長さと幅のみをもつものである.
6. 面の端は線である.
7. 平面とはその上にある直線について一様に横たわる面である.
8. 平面角とは平面上にあって互いに交わりかつ一直線をなすことのない二つの線相互のかたむきである.
9. 角をはさむ線が直線であるとき, その角は直線角とよばれる.
10. 直線が直線の上に立てられて接角を互いに等しくするとき, 等しい角の双方は直角であり, 上に立つ直線はその下の直線に対して垂線とよばれる.
11. 鈍角とは直角より大きい角である.
12. 鋭角とは直角より小さい角である.
13. 境界とはあるものの端である.
14. 図形とは一つまたは二つ以上の境界によってかこまれたものである.
15. 円とは一つの線にかこまれた平面図形で, その図形の内部にある 1 点からそれへひかれたすべての線分が互いに等しいものである.
16. この点は円の中心とよばれる.
17. 円の直径とは円の中心を通り両方向で円周によって限られた任意の線分であり, それはまた円を 2 等分する.
18. 半円とは直径とそれによって切り取られた弧とによってかこまれた図形である. 半円の中心は円のそれと同じである.
19. 直線図形とは線分にかこまれた図形であり, 三辺形とは三つの, 四辺形とは四つの, 多辺形とは四つより多くの線分にかこまれた図形である.
20. 三辺形のうち, 等辺三角形とは三つの等しい辺をもつもの, 二等辺三角形とは二つだけ等しい辺をもつもの, 不等辺三角形とは三つの不等な辺をもつものである.
21. さらに三角形のうち, 直角三角形とは直角をもつもの, 鈍角三角形とは鈍角をもつもの, 鋭角三角形とは三つの鋭角をもつものである.

22. 四辺形のうち, 正方形とは等辺でかつ角が直角のもの, 矩形とは角が直角で, 等辺でないもの, 菱形とは等辺で, 角が直角でないもの, 長斜方形とは対辺と対角が等しいが, 等辺でなく角が直角でないものである. これら以外の四辺形はトラペジオンと呼ばれるとせよ.
23. 平行線とは, 同一の平面上にあって, 両方向に限りなく延長しても, いずれの方向においても互いに交わらない直線である.

・公準 (要請)

1. 任意の点から任意の点へ直線をひくこと.
2. および有限直線を連続して一直線に延長すること.
3. および任意の点と距離 (半径) とをもって円を描くこと.
4. およびすべての直角は互いに等しいこと.
5. およびすべての 1 直線が 2 直線に交わり同じ側の内角の和を 2 直角より小さくするならば, この 2 直線は限りなく延長されると 2 直角より小さい角にある側において交わること.

・公理 (共通理念)

1. 同じものに等しいものはまた互いに等しい.
2. また等しいものに等しいものが加えられれば全体は等しい.
3. また等しいものから等しいものがひかれれば, 残りは等しい.
4. また不等なものに等しいものが加えられれば全体は不等である.
5. また同じものの 2 倍は互いに等しい.
6. また同じものの半分は互いに等しい.
7. また互いに重なり合うものは互いに等しい.
8. また全体は部分より大きい.
9. また 2 線分は面積をかこまない.

・命題

1. 与えられた有限な直線 (線分) の上に等辺三角形をつくること.
2. 与えられた点において与えられた線分に等しい線分をつくること.
3. 二つの不等な線分が与えられたとき, 大きいものから小さいものに等しい線分を切り取ること.
4. もし二つの三角形が 2 辺が 2 辺にそれぞれ等しく, その等しい 2 辺にはさまれる角が等しいならば, 底辺は底辺に等しく, 三角形は三角形に等しく, 残りの 2 角は残りの 2 角に, すなわち等しい辺が対する角はそれぞれ等しいであろう.
5. 二等辺三角形の底辺の上にある角は互いに等しく, 等しい辺が延長されるとき, 底辺の下の角は互いに等しいであろう.
6. もし三角形の 2 角が互いに等しければ, 等しい角に対する辺も互いに等しいであろう.
7. 一つの線分を底辺として, 三角形をなす 2 線分にそれぞれ等しく, 同じ側に異なった点で交わり, 最初の 2 線分と同じ端をもつ他の 2 線分をつくることはできない.
8. もし二つの三角形において 2 辺が 2 辺にそれぞれ等しく, 底辺も底辺に等しければ, 等しい辺にはさまれた角もまた等しいであろう.
9. 与えられた直線角を 2 等分すること.
10. 与えられた線分を 2 等分すること.
11. 与えられた直線にその上の与えられた点から直角に直線をひくこと.
12. 与えられた無限直線にその上にない与えられた点から垂線を下ろすこと.
13. もし直線が直線の上に立てられて二つの角をつくるならば, 二つの直角かまたはその和が 2 直角に等しい角をつくるであろう.

14. もし任意の直線に対してその上の点において同じ側にない 2 直線が接角の和を 2 直線に等しくするならば, この 2 直線は互いに一直線をなすであろう.
15. もし 2 直線が互いに交わるならば, 対頂角を互いに等しくする.
16. すべての三角形において辺の一つが延長されるとき, 外角は内対角のいずれよりも大きい.
17. すべての三角形においてどの 2 角をとってもその和は 2 直角より小さい.
18. すべての三角形において大きい辺は大きい角に対する.
19. すべての三角形において大きい角には大きい辺が対する.
20. すべての三角形においてどの 2 辺をとってもその和は残りの 1 辺より大きい.
21. もし三角形の辺の一つにその両端から三角形の内部で交わる 2 線分がつくられるならば, つくられた 2 線分はその和が三角形の残りの 2 辺の和より小さいが, より大きい角をはさむ.
22. 与えられた 3 線分に等しい 3 線分から三角形をつくること. ただしどの 2 線分をとってもその和は残りの線分より大きくなければならない.
23. 与えられた直線上にその上の点において与えられた直線角に等しい直線角をつくる.
24. もし二つの三角形において 2 辺が 2 辺にそれぞれ等しく, 等しい線分によってはさまれる角の一方が他方より大きいならば, 底辺も底辺より大きい.
25. もし二つの三角形において 2 辺が 2 辺にそれぞれ等しく, 底辺が底辺より大きいならば, 等しい線分にはさまれる角も一方が他方より大きいであろう.
26. もし二つの三角形において 2 角が 2 角にそれぞれ等しく,1 辺が 1 辺に, すなわち等しい 2 角にはさまれる辺かまたは等しい角の一つに対する辺が等しければ, 残りの 2 辺も残りの 2 辺に等しく, 残りの角も残りの角に等しいであろう.
27. もし 1 直線が 2 直線に交わってなす錯角が互いに等しければ, この 2 直線は互いに平行であろう.
28. もし 1 直線が 2 直線に交わってなす一つの外角が同じ側の内対角に等しいかまたは同側内角の和が 2 直角に等しければ, この 2 直線は互いに平行であろう.
29. 一つの直線が二つの平行線に交わってなす錯角は互いに等しく, 外角は内対角に等しく, 同側内角の和は 2 直角に等しい.
30. 同一の直線に平行な 2 直線はまた互いに平行である.
31. 与えられた点を通り, 与えられた直線に平行線をひく.
32. すべての三角形において 1 辺が延長されるとき, 外角は二つの内対角の和に等しく, 三角形の三つの内角の和は 2 直角に等しい.

3.1.1 ユークリッド原論の実装

ユークリッド原論の実装を Isabelle を用いて行った様子を示す. 当時の研究では, 各命題を証明したのち, それを新たな関数として宣言し, 以後の証明に使用していくという方式を採用している. 以下は, 原論の命題 1_2 までの証明である.

Isabelle 上に前提に即した関数を作成する (ここでは命題 1_2 までに関わるもののみ提示する). 以下, 実装において各要素が, どの前提のもとに成り立つものであるかを付記する (定義なら [Th.n] 公準なら [Po.n] 公理なら [Ax.n] など).

・型の宣言

```
datatype point = ”char”
datatype segment = Se ”point” ”point”
datatype circle = Ci ”point” ”point”
```

上から順に, 点, 線分, 円の型宣言である.
次に, これらが線や円などの性質を持った型であることを関数の形で実装していく.

・線分

線分のデータ型は二つの点で定義する. 線分 AB を “Se A B”とする.

```
locale dist =
  fixes ldist :: ”segment ⇒ segment ⇒ bool” (infixl ”[@]” 50)
  assumes dist_ref [simp,intro] : ”s1 [@] s1”
    and dist_rev1 : ”⟦s1 [@] s2⟧ ⟹ s2 [@] s1”
    and dist_rev3 : ”⟦(Se x1 y1) [@] (Se x2 y2)⟧⟹ (Se x1 y1) [@] (Se y2 x2)”
```

各 assumes 文の意味は以下の通りである.

(以下, 線分 AB を AB, 線分 AB+線分 CD を AB+CD とする)

- AB = AB
- AB = CD ならば CD = AB
- AB = CD ならば AB = DC

・円周上の点

円のデータ型は二つの点により定義する. 中心 A, 半径 AB をもって描く円 “Ci A B”とする. ここでは指定した円と, その円周上の点との関係を関数化する.

```
locale circledef = dist +
  fixes lcircle :: ”point ⇒ circle ⇒ bool” (infixl ”[on]” 50)
  assumes circle_dist1 : ”⟦p [on] (Ci c r)⟧ ⟹ (Se c r) [@] (Se c p)”
    and circle_dist3 : ”⟦p [on] (Ci c r); p [on] (Ci r c)⟧⟹ (Se p c) [@] (Se p r)”
```

fixes 文は,「点が円周上にあるとき」を意味する. 各 assumes 文の意味は以下の通りである.

(以下, 円 AB とは中心 A, 半径 AB をもって描く円とする)

1. 点 P が円 AB の円周上の点であるとき AB = PA [Th.15]
2. 点 P が円 AB の円周上の点でありかつ円 BA 上の点 (二円の交点) であるとき
   PA = PB [Th.15][Ax.1]

命題 1-1. 与えられた有限な直線 (線分) の上に等辺三角形をつくること.

- 仮定

a1. 与えられた線分を AB とする.

a2. 中心 A, 半径 AB をもって円を描く.

a3. 中心 B, 半径 BA をもって円を描く.

a4. 2 円の交点を Γ とする.

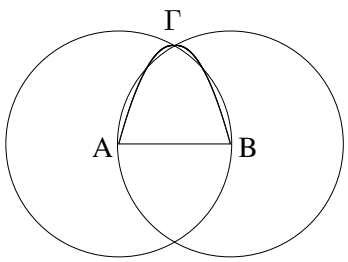

図 11: 命題 1-1

- 証明

P1. a2 より, AB = ΓA

P2. a3 より, AB = ΓB

P3. a4 より, ΓA = ΓB

・ P1 P2 P3 より, 線分 AB 上に等辺三角形 ΓAB がつくられている.

Isabelle 上での実装が以下のようになる.

```
theorem (in areadef) Proposition1_1:
  fixes A B Γ :: point
    and AB ΓA ΓB :: segment
    and CircAB CircBA :: circle
  assumes "AB = Se A B" "ΓA = Se Γ A" "ΓB = Se Γ B"
    "CircAB = Ci A B" "CircBA = Ci B A"
    "Γ [on] CircAB" "Γ [on] CircBA"
  shows "AB [@] ΓA" and "AB [@] ΓB" and "ΓB [@] ΓA"
  proof -
  from assms show "AB [@] ΓA" by (simp add:circle_dist1 dist_rev3)
  from assms show "AB [@] ΓB" by (simp add:circle_dist1 dist_rev1 dist_rev3)
  from assms show "ΓB [@] ΓA" by (simp add:circle_dist3)
qed
```

以後の証明の為, この命題を関数化する.

```
locale L_Proposition1_1 = areadef +
  fixes L_Prop1_1 :: "point ⇒ segment ⇒ bool" ("[p1-1] _,_")
  assumes Prop1_1 : "⟦[p1-1] pn,(Se p1 p2)⟧
    ⟹ Se p1 p2 [@] Se pn p1 ∧ Se p1 p2 [@] Se pn p2 ∧ Se pn p1 [@] Se pn p2"
```

命題 1-2. 与えられた点において与えられた線分に等しい線分をつくること.

- 仮定

a1. 与えられた点を A, 線分を BΓ とする.

a2. 線分 AB の上に等辺三角形 ΔAB をつくる (命題 1_1).

a3. ΔA を延長して線分 AE, ΔB を延長して線分 BZ をそれぞれ描く.

a4. 中心 B, 半径 BΓ をもって円を描き, BZ との交点を H とする.

a5. 中心 Δ, 半径 ΔH をもって円を描き, AE との交点を Λ とする.

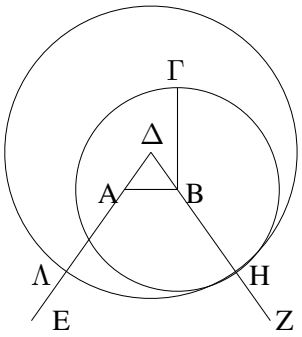

図 12: 命題 1-2

- 証明

P1. a5 より, ΔΛ = ΔH (ΔA+AΛ = ΔB+BH)

P2. a4 より, BH = BΓ. よって ΔB+BH = ΔB+BΓ

P3. P1 P2 より, ΔA+AΛ = ΔB+BΓ

P4. a2 より, ΔA = ΔB

S1. P3 P4 より, AΛ = BΓ

・ S1 より, 点 A において線分 BΓ に等しい線分 AΛ がつくられている.

Isabelle 上での実装が以下のようになる.

```
theorem (in L_Proposition1_1) Proposition1_2:
  fixes A B Γ Δ Λ H :: point
    and ΔA AΛ ΔB BΓ AB BH ΔΛ ΔH :: segment
    and ΔA_AΛ ΔB_BΓ ΔB_BH :: seg_list
    and CircBΓ CircΔH :: circle
  assumes
    "ΔA = Se Δ A" "AΛ = Se A Λ" "ΔB = Se Δ B"
    "BΓ = Se B Γ" "AB = Se A B" "BH = Se B H"
    "ΔΛ = Se Δ Λ" "ΔH = Se Δ H"
    "ΔA_AΛ = Sel ΔA AΛ" "ΔB_BΓ = Sel ΔB BΓ" "ΔB_BH = Sel ΔB BH"
    "ΔΛ [@-@] ΔA_AΛ" "ΔH [@-@] ΔB_BH"
    "CircBΓ = Ci B Γ" "CircΔH = Ci Δ H"
    "Λ [on] CircΔH" "H [on] CircBΓ" "H [on] CircΔH" "Γ [on] CircBΓ"
    "[p1-1]Δ,AB"
  shows "AΛ [@] BΓ"
proof -
  from assms have P1:"ΔA_AΛ [@@] ΔB_BH" by (simp add:circle_dist2 dist_rev2 dist_list_list)
  from assms have P2:"ΔB_BH [@@] ΔB_BΓ" by (simp add:dist_list1 list_rev2 circle_dist2 dist_rev2)
  from assms P1 P2 have P3:"ΔA_AΛ [@@] ΔB_BΓ" by (blast intro:list_trans)
  from assms have P4:"ΔA [@] ΔB" by (simp add:Prop1_1)
  from assms P3 P4 show "AΛ [@] BΓ" by (simp add:list_dist3)
qed
```

ユークリッド幾何学を ATP に実装するに当たって問題となったのは, 面や角といった各要素の定義解釈である. 特に「直線上に存在する点」や「重なった点および線」など, 明文化された定義が存在しないものも多く, 既存の前提から導き出せないそれらは, 何を以て “それ”と見做すかという点に証明者の主観が含まれることを回避できない.

そこで本研究では, 実装の対象を次節より述べる幾何学基礎論へと移行させた.

3.2 ヒルベルトの公理 - 幾何学基礎論

『幾何学基礎論』[9] は 1899 年にヒルベルトによって著作された数学書である. この著書は公理主義, 形式主義思想によってユークリッド幾何学の公理系を最も厳密に吟味した著作としても有名である.

ヒルベルトは, [9] の第 1 章において, ユークリッド幾何学の厳密な再構成を行った. そこでは, 5 種類の公理群が列挙され, 32 の定理が証明されている. 公理群は, 結合公理 (I1〜I3), 順序公理 (Ⅱ1〜Ⅱ4), 合同公理 (Ⅲ1〜Ⅲ5), 平行線公理 (IV), 連続性公理 (V1 〜V2) の 15 の公理からなる (空間幾何に関する結合公理 I4〜I8 は除く). これらの公理と定理を, ここではそれぞれ, HaI-1〜HaV-2, Ht-1〜Ht-32 と表記する. ヒルベルトの公理系では, 点や直線といった基本的な概念は, 無定義用語として扱われ, 公理でその関係性だけが規定される.

3.2.1 幾何学基礎論の実装

幾何学基礎論の実装を Isabelle を用いて行った様子を示す.

こちらは, 証明した各命題がそのまま以降の証明に使用可能な定理となるような構造に改良している. 以下は, 結合公理 (I1〜I3) まで, およびそれらに付随する補題の証明である. Isabelle プログラム全文については, 付録として章末に転載している他, [8] にも寄稿している.

・型の宣言

```
datatype Point = ”char”
datatype Segment = Se ”Point” ”Point”
datatype Line = Li ”Point” ”Point”
datatype Geo_object =
   Poi ”Point”
   | Seg ”Segment”
   | Lin ”Line”
datatype sign = add | sub
datatype Geo_objects = Emp | Geos ”Geo_object” ”sign” ”Geo_objects”

locale Eq_relation =
   fixes Eq :: ”Geo_objects ⇒ Geo_objects ⇒ bool”
      and Inv :: ”bool ⇒ bool”
   assumes Eq_refl [simp,intro] : ”Eq obs obs”
      and Eq_rev : ”⟦Eq obs1 obs2⟧ ⟹ Eq obs2 obs1”
      and Eq_trans : ”⟦Eq obs1 obs2; Eq obs2 obs3⟧ ⟹ Eq obs1 obs3”
      and Inv_def : ”Inv b1 ⟷ ¬ b1”
```

これらは点, 線分, 直線と, それらを包括する型 (Geo_object), さらに複数の Geo_object 型を扱う Geo_objects 型の宣言と, その等価, 反射律, 対称律, 推移律および逆の設定である. なお, ここでは省略しているが, 幾何学基礎論における角の合同には公理に推移律が含まれていないため, 別の関数を使用している.

・定義, 公理

```
locale Definition_1 = Eq_relation +
   fixes Line_on :: ”Line ⇒ Point ⇒ bool”

locale Axiom_1 = Definition_1 +
   assumes Line_exist : ”⟦¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)⟧ ⟹
         ∃l. Line_on l p1 ∧ Line_on l p2”
      and Line_unique : ”⟦Line_on l1 p1; Line_on l1 p2; Line_on l2 p1; Line_on l2 p2;
         ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)⟧ ⟹
         Eq (Geos (Lin l1) add Emp) (Geos (Lin l2) add Emp)”
      and Line_on_exist :
         ”∃p q. Line_on l1 p ∧ Line_on l1 q ∧ ¬ Eq (Geos (Poi p) add Emp) (Geos (Poi q) add Emp)”
      and Line_not_on_exist : ”∃p q r. ¬ Line_on (Li p1 p2) p ∧ ¬ Line_on (Li p1 p2) q
         ∧ ¬ Line_on (Li p1 p2) r ∧ ¬ Eq (Geos (Poi p) add Emp) (Geos (Poi q) add Emp)
         ∧ ¬ Eq (Geos (Poi q) add Emp) (Geos (Poi r) add Emp)”
         ∧ ¬ Eq (Geos (Poi r) add Emp) (Geos (Poi p) add Emp)”
```

各 assumes 文の意味は以下の通りである.

- 相異なる二点に対し, これら二点の各々と結合する少なくとも一つの直線が常に存在する. [HaI-1]
- 相異なる二点に対し, これら二点の各々と結合する直線は一つより多くは存在しない. [HaI-2]
- 一直線上には常に少なくとも二点が存在する. [HaI-3]
- 一直線上にない少なくとも三点が存在する. [HaI-3]

[9] において,『結合する』という言葉の代わりに「直線 a が A および B を通る」,「A が a の上にある」,「A は a の点である」,「a の上に点 A が存在する」などの言葉を用いること, さらに A が直線 a の上にありかつ他の直線 b の上にあるとき,「直線 a, b は A において交わる」,「直線 a, b が点 A を共有する」などのように言い表すともされている.

・Isabelle へ実装する上で必要となった規則

```
locale Incidence_Rule = Axiom_1 +
   assumes Point_Eq : ”⟦P1(p1); Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)⟧ ⟹ P1(p2)”
      and Line_on_trans : ”⟦Eq (Geos (Lin l1) add Emp) (Geos (Lin l2) add Emp); Line_on l1 p1⟧ ⟹
```

Line_on l2 p1”
and Line_on_rule : ”Line_on (Li p1 p2) p1 ∧ Line_on (Li p1 p2) p2”

各 assumes 文の意味は以下の通りである.

- A, A’ が同一の点であれば, P(A) = P(A’).
- a, a’ が同一の直線であれば, a の上にある点は a’ の上にもある.
- 直線 AB とは, 二点 A, B を通る直線である.

・補題 (一部抜粋)

- 二点 A, B が異なる点であるとき, 直線 AB と直線 BA は同一の直線である.

```
lemma(in Incidence_Rule) Line_rev :
  assumes ”¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)”
  shows ”Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p2 p1)) add Emp)”
proof -
  from assms have P1 : ”Line_on (Li p1 p2) p1 ∧ Line_on (Li p1 p2) p2” by (simp add:Line_on_rule)
  have P2 : ”Line_on (Li p2 p1) p1 ∧ Line_on (Li p2 p1) p2” by (simp add:Line_on_rule)
  from assms P1 P2 show ”Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p2 p1)) add Emp)”
    by (blast intro:Line_unique)
qed
```

- 点 C が直線 AB 上にない点ならば, 二点 A, C は相異なる点である.

```
lemma(in Incidence_Rule) Line_not_on_Point :
  assumes N :
    ”¬ Line_on (Li p1 p2) p3”
  shows ”¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)”
proof
  assume W : ”Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)”
  have P1 : ”Line_on (Li p1 p2) p1” by (simp add:Line_on_rule)
  from W P1 have P2 : ”Line_on (Li p1 p2) p3” by (simp add:Point_Eq)
  from N P2 show False by simp
qed
```

- 二点 A, B および二点 A, C がそれぞれ相異なる点であり, 直線 AB 上に点 C が存在するならば, 直線 AC 上に点 B が存在する.

```
lemma(in Incidence_Rule) Line_on_rev :
  assumes
    ”¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)”
    ”¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)”
    ”Line_on (Li p1 p2) p3”
  shows ”Line_on (Li p1 p3) p2”
proof -
  have P1 : ”Line_on (Li p1 p2) p1” by (simp add:Line_on_rule)
  have P2 : ”Line_on (Li p1 p3) p1” by (simp add:Line_on_rule)
  have P3 : ”Line_on (Li p1 p3) p3” by (simp add:Line_on_rule)
  from assms P1 P2 P3 have P4 : ”Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p1 p3)) add
Emp)”
    by (simp add:Line_unique)
  have P5 : ”Line_on (Li p1 p2) p2” by (simp add:Line_on_rule)
  from P4 P5 show ”Line_on (Li p1 p3) p2” by (simp add:Line_on_trans)
qed
```

3.2.2 先行研究　角の取り扱い

幾何学基礎論における角の定義を簡易に表せば,「一つの点から出る相異なる二つの半直線の組」となる. また, 中心の点 (頂点) および各半直線上にある任意の点を以て, 角を成す三点とする

ことを公理の中で認めている.

そのため, 定理証明支援系における角の実装には, 以下の二つの方式が考えられる [10].

- 角を成す二直線 l, m のペア lm で直線 l から直線 m に反時計回りで到達する角度を表現する方法 (以下, 線角と呼ぶ).
- 頂点 A と二半直線上の点 B, C から角 BAC を表現する方法 (以下, 点角と呼ぶ).

[10] では線角について, 点の選び方に由来する自由度がなく一意的で簡明である, 錯角や同位角など平行線に関する規則を適用できる条件が点配置の影響を受けず一定であるといった, 点角に対する優位性を示している.

これを示す例として次のような状況が挙げられている.

直線 k = AB と直線 l = DC が平行であり, 直線 m は直線 k および l と点 A, D で交わっているものとする.

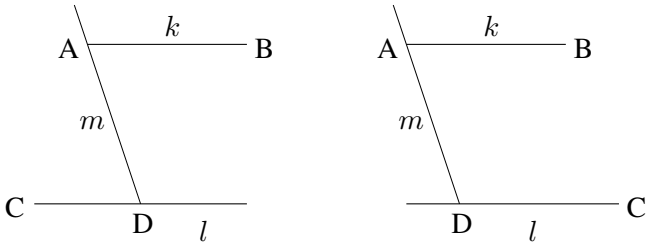

図 13: 線角・点角における錯角について

直線 l 上の点 C の位置によって左右どちらかの図に場合分けされる.

このとき線角および点角で平行線の錯角や同位角をそれぞれ表すと, 線角であればどちらの図の場合でも, 角 mk と角 ml が錯角, 角 km と角 lm が同位角であると容易に表せるが, 点角の場合は点 C の位置がどちらの図の状態であるかを先に明確にしなければならない.

一方で線角は, 人間的な推論を考える場合, 三角形の合同条件を表すのが困難であると述べている [10]. その原因は線角が角度の向きの情報を含むことが原因であり, ここでは解決のために推論が点配置によらないような枠組のなかで合同を取り扱えるようにする方法を構成, 実装することを主眼としている.

これを受けて本研究では, Isabelle に実装するにあたり点角を採用した.

その理由の一端には [10] でも述べられている通り, 通常の初等幾何で用いられるのが点角であることも含まれるが, 幾何学基礎論においては, 間の定義や平面における直線の側の定義など, 点配置に関する公理が角の公理より前に数多く存在していることが大きい.

先の例であれば, 直線 m について二点 B, C が同じ側にあるか異なる側にあるかという条件設定が角の実装方式に関わらず必要となる可能性が高い. このように線角の優位性を利用できる機会は少ないと判断したためである.

3.3 幾何学基礎論の補完

[9] の証明の中には,「これは公理や定義から明らかである」として, その証明が明文化されていない幾つかの事例が存在する. まずはそれらに至るまでに必要な各公理, 定理および補題について示す.

HaI-1 二点 A, B に対し, これらの二点の各々と結合する少なくとも一つの直線が常に存在する.
HaI-2 二点 A, B に対し, これらの二点の各々と結合する直線は一つより多くは存在しない.

この二つの公理は直線の一意性を示している. そこで以下のような補題を考える.

補題 1　二直線 a, b の上にそれぞれ相異なる二点 A, B が存在するとき, a, b は同一の直線である. また, 二直線が同一の直線であるとき, 一方の上に点 C が存在するならば, 他方の上にも C が存在する.

相異なる二点 A, B を共有する直線は HaI-2 より一意であることが明らかであり, したがってこの補題の求めるところが容易に導かれる.

HaI-3　一直線上には常に少なくとも二点が存在する. 一直線上にない少なくとも三点が存在する.

この公理により, 与えられた直線に対しその上にない点の存在が保証される.

定義　一直線上の点は互いにある関係を有する. これを記述するのに『間』という言葉を用いる.
HaII-1　点 B が点 A と点 C との間にあれば, A, B, C は一直線上の相異なる三点であって かつ B はまた C と A の間にある.
HaII-2　二点 A, C に対し直線 AC 上に常に少なくとも一点 B が存在して, C が A と B との間にある.

ここで『間』という概念が定義される. 本論文では,「点 B が点 A と点 C との間にある」を以後「Bet (A, C) B」のように表記する. この公理は三点それぞれの非一致性, 同一直線上への存在, さらには Bet (A, C) B ならば Bet (C, A) B でもあることを同時に保証している.

HaII-3　一直線上にある任意の三点のうちで, 他の二点の間にあり得るものは一点より多くはない.

この公理から以下の補題が導かれる.

補題 2　一直線上に相異なる三点 A, B, C が与えられたとき, Bet (A, C) B であれば ¬ Bet (A, B) C かつ ¬ Bet (B, C) A である.
証明　もし Bet (A, B) C であれば, B および C が他の二点の間にあることとなり, そのような点が一点より多くなる. そのため Bet (A, B) C では有り得ない. Bet (B, C) A も同様である. したがって, 補題の求めるところが導かれる.

定義　一直線 a の上にある二点 A, B を考え, この二点 A, B の組みを『線分』と名付け, AB または BA と表す. A と B の間にある点を線分 AB の点, あるいは AB の内点とし, 点 A, B を AB の端点, また a 上にあり, AB の内点および端点でない全ての点を AB の外点とする.
HaII-4　A, B, C が一直線上にない三点であり, a を A, B, C の何れをも通らない直線とする. a が線分 AB の点を通ればこれはまた線分 AC もしくは線分 BC の点を通る.

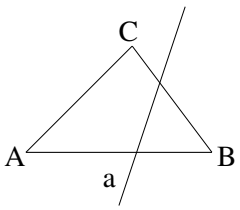

図 14: HaII-4

この公理を直感的に表せば,「直線が三角形の内部を通るなら再び外部を通る. このとき二本より多くの辺は通らない」ということになる (ただしこの時点ではまだ多角形が定義されていないため, これはあくまで直感的な理解に留まる).

ここまでの公理から幾つかの定理が導かれる. そのうち本稿に関わるものを以下に示す.

Ht-4　一直線上にある任意の三点 A, B, C のうちで, 他の二点の間にある一点が常に存在する.
Ht-5　一直線上に任意の四点が与えられたとき, これらの点を A, B, C, D を以て表し, B を A と C の間にあり かつ A と D の間にあるように, また C を A と D の間にあり かつ B と D の間にあるようにすることが常に可能である.

Ht-5 は一直線上に相異なる四点が存在するとき, その順序整理が可能であることを保証する定理である. その証明において, このような四点の位置について, HaⅡ-3 および Ht-4 より, 以下のように区別可能であると述べている.

一直線上に相異なる四点が存在するとき, まずそのうち三点に注目する. Ht-4 より他の二点の間にある一点が常に存在するため, そのような点を Q とおき, 残りの二点をそれぞれ P, R とおく. 与えられた最後の点を S とおくと, 各点の位置は以下の五つの場合に区別される.

(1) Bet (P, S) R.

(2) Bet (R, S) P.

(3) Bet (P, R) S ∧ Bet (P, S) Q.

(4) Bet (P, Q) S.

(5) Bet (Q, S) P.

以後, Bet (P, R) Q であるような一直線上の相異なる四点 P, Q, R, S に対して, Ht-5(1〜5) と表記して, その位置の場合を表す. さらに以下の二つの補題を示し, それぞれの場合に各補題が対応していることを示す.

補題 3 A, B, C, D が一直線上の相異なる四点とする. このとき, Bet (A, C) B ∧ Bet (B, D) C ⟹ Bet (A, D) B ∧ Bet (A, D) C.
補題 4 A, B, C, D が一直線上の相異なる四点とする. このとき, Bet (A, C) B ∧ Bet (A, D) C ⟹ Bet (A, D) B ∧ Bet (B, D) C.

これらの補題の証明は [9] で行われている. Ht-5(1〜5) の各場合において, 四点 P, Q, R, S を必要に応じて A, B, C, D を以て表せば以下のようになる.

(P, Q, R, S = A, B, C, D は P = A, Q = B, R = C, S = D を表す)

Ht-5(1) P, Q, R, S = A, B, C, D ⟹ Bet (A, C) B ∧ Bet (A, D) C.

Ht-5(2) P, Q, R, S = C, B, A, D ⟹ Bet (C, A) B ∧ Bet (D, A) C (HaⅡ-2 より Bet (A, C) B ∧ Bet (A, D) C).

Ht-5(3) P, Q, R, S = A, B, D, C ⟹ Bet (A, D) B ∧ Bet (A, D) C ∧ Bet (A, C) B.

Ht-5(4) P, Q, R, S = A, C, B, D ⟹ Bet (A, D) C ∧ Bet (A, C) B.

Ht-5(5) P, Q, R, S = B, C, D, A ⟹ Bet (B, D) C ∧ Bet (C, A) B (HaⅡ-2 より Bet (B, D) C ∧ Bet (A, C) B).

Ht-5(1〜4) に補題 4 が, Ht-5(5) に補題 3 が対応することが示され, Ht-5 が証明される.

3.3.1 平面における直線の側

平面における直線の側という概念は, 以下の定理と共に定義されている.

Ht-8 任意の直線 a は, a 上にない点を次の性質を有する二つの領域に分ける : 相異なる領域から各々任意の二点 A, B をとるとき, それらの定める線分 AB の上には a の点が存在し, 同一の領域からとった任意の二点 A, A' の定める線分 AA' は a の点を含まない.
定義 点 A, A' を『平面において直線 a の同じ側にある』といい, 点 A, B を『平面において直線 a の相異なる側にある』という.

[9] において, この定義はその後, 直線の側および多角形の定義へと使用されており, これ以上の言及は為されていない. しかしながら, この概念に関する以下の問題について, この定義から明らかである, あるいは特別な困難なしに辿り着くと主張するのは難しいと考える.

問題 1 ある平面における相異なる三点 A, B, C について, 直線 a の相異なる側に二点 A, B が存在し, また二点 A, C が a の相異なる側に存在するとき, B, C は a の同じ側に存在する.
問題 2 ある平面における相異なる三点 A, B, C について, 直線 a の同じ側に二点 A, B が存在し, また二点 A, C が a の相異なる側に存在するとき, B, C は a の相異なる側に存在する.

この問題が関係する箇所の一例として, [9] では, この後に示す角の定義において, 一点の頂点から出でて角を構成する二本の半直線が, 平面上の点を内外二つの領域に分けるとしている. そして

この頂点から出る半直線は全く角の内部に含まれるか, あるいは全く角の外部に出ることが容易に証明できると述べている. これは言い換えれば,『ある領域に存在する点に対して, 相異なる側にあるすべての点は同じ領域にある』あるいは『同じ領域に存在する幾つかの点のうちのいずれかに対して相異なる領域にある点は, 他の点とも異なる領域にある』となり, これらの問題の示すところを主張していると考えられる.

本論文では,「ある平面において二点 A, B が直線 a の同じ側にある」および「ある平面において二点 A, B が直線 a の異なる側にある」を, 以後「Sameside : a (A, B)」,「Diffside : a (A, B)」のようにそれぞれ表記する.

問題の解決のために, まずは以下の補題および命題を証明する.

補題 5　Sameside : a (A, B) であるとき, Sameside : a (B, A) である. また Diffside : a (A, B) であるとき, Diffside : a (B, A) である.

これは, 線分の定義に含まれる端点の可逆性から容易に導かれる.

補助命題 1　Sameside : a (A, B) ならば, ¬ Diffside : a (A, B) である.
証明　定義より, A, B は直線 a 上にない相異なる二点である. もし Diffside : a (A, B) であれば, 線分 AB の上に a の点が存在する. しかし Sameside : a (A, B) から定義より, 線分 AB が a の点を含まないことに矛盾する. よって Sameside : a (A, B) ならば, ¬ Diffside : a (A, B) である.

補助命題 2　Diffside : a (A, B) ならば, ¬ Sameside : a (A, B) である.
証明　もし Sameside : a (A, B) であれば, 補助命題 1 より, ¬ Diffside : a (A, B) である. すなわち, Diffside : a (A, B) ∧ Sameside : a (A, B) は有り得ない. よって, Diffside : a (A, B) ならば, ¬ Sameside : a (A, B) である.

補助命題 3　直線 a および相異なる二点 A, B について, ¬ Sameside : a (A, B) かつ, それぞれが a 上の点でないならば, Diffside : a (A, B) である.
証明　もし線分 AB が a の点を含むならば, 仮定および定義より, Sameside : a (A, B) となり, 仮定に矛盾する. よって, 線分 AB が a の点を含まない. 仮定および線分 AB が a の点を含まないため, 定義より, Diffside : a (A, B) である.

補助命題 4　直線 a および相異なる二点 A, B について, ¬ Diffside : a (A, B) かつ, それぞれが a 上の点でないならば, Sameside : a (A, B) である.
証明　もし ¬ Sameside : a (A, B) ならば, 仮定および補助命題 3 より, Diffside : a (A, B) となり, 仮定に矛盾する. よってこのとき, Sameside : a (A, B) である.

平面における直線の側の定義は, 注目する二点が互いに相異なりかつ対象の直線上に無い場合においてのみ機能する. ゆえに, ¬ Sameside : a (A, B) ならば直ちに Diffside : a (A, B) となるわけではなく, 逆もまた然りである.

次に, 以下の命題の証明を行う.

補助命題 5　Bet (A, C) B であるとき, 点 A を通り かつ三点 A, C, B を含む直線とは相異なる直線を b とおけば, Sameside : b (B, C) である.

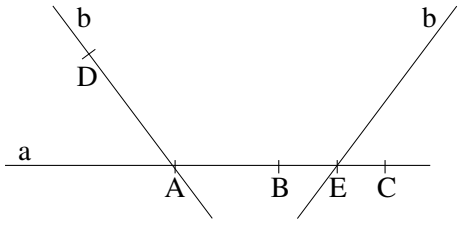

図 15: 補助命題 5 (Diffside : b (B, C) を仮定したとき)

証明　HaI-3 より, 三点 A, C, B を含む直線 a の上にない点 D をとり, 二点 D, A を通る直線を b とする. このとき点 A を通り かつ a と相異なる直線 b に対して, Sameside : b (B, C) であると主張する. まず二直線 a, b が相異なる直線であることを示す. もし二直線が同一の直線であれば, b の上に D が存在するため, 補題 1 より, a の上にも D が存在することになり, a 上にない点 D をとったことに矛盾する. よって, a, b は相異なる直線である. もし Diffside : b (B, C) であれば, 定義より, Bet (B, C) E となる点 E が存在し, b が E を通る. ここで, 二点 A, E が同一の点であれば, Bet (B, C) A となり, 仮定および補題 2 より, ¬ Bet (B, C) A であるため, これに矛盾する. よって, A, E は相異なる点である. このとき, 二直線 a, b はともに相異なる二点 A, E と結合するため, 補題 1 より同一の直線となり, 二直線が相異なる直線であることに矛盾する. よって, ¬ Diffside : b (B, C) である. 仮定および定義より, A, C は相異なる二点である. a が A, C を含む直線であり, b が A を含む直線であるため, もし b が C を含むならば, 補題 1 より, a, b が同一の直線となって, これらが相異なる直線であることに矛盾する. よって b は C を含まない. また, b が B を含まないことも同様にして導ける. そして, 仮定および定義より, B, C は相異なる二点である. 相異なる二点 B, C について ¬ Diffside : b (B, C) かつ, それぞれが b 上の点でないため, 補助命題 4 より, Sameside : b (B, C) である. したがって, 命題の求めるところが証明された.

補助命題 6　直線 a および相異なる三点 A, B, C について, Diffside : a (A, B) かつ Diffside : a (A, C) のとき, Sameside : a (B, C) である.

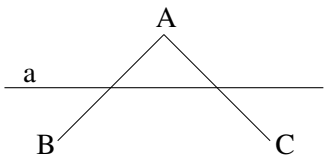

図 16: 補助命題 6

証明　まず三点のうちの一点が残る二点の各々と結合する直線上に存在しない場合を考える. 点 C が直線 AB 上にないとすれば, 三点 A, B, C は一直線上にない点であり, 定義より, a は A, B, C のいずれも通らない直線であるため, HaⅡ-4 より, a が三つの線分の何れかを通るならば, 必ず残りの二線分のどちらか一方を通る. 定義より, a は線分 AB および線分 AC を通るため, 残る線分 BC は通らない. よってこの場合, Sameside : a (B, C) である. 残りの点においても同様に導ける.

次に三点が一直線 b 上にある場合を考える. まず a と AB の共有する点を D とおく. Bet (A, B) D から, HaⅡ-1 より, D は A および B と相異なる点である. ここで, 二点 D, C が同一の点であれば, D は a 上の点であるため, C が a 上にあることとなり, a が A, B, C のいずれも通らないことに矛盾する. よって, D, C は相異なる点である. すなわち, 四点 A, B, C, D は相異なる点である. ここで, Bet (A, B) D から, HaⅡ-1 より, 三点 A, B, D は一直線上の点である. b およびこの直線はともに相異なる二点 A, B を共有するため, 補題 1 より b の上にも D が存在する. よって, b は四点 A, B, C, D を通る. すなわち A, B, C, D は一直線上の相異なる四点である.

A, B, C, D が一直線上の相異なる四点であり, Bet (A, B) D であるため, 残る点 C の位置は, P, Q, R, S = A, D, B, C と変換することで Ht-5(1〜5) で示した通りに場合分けできる.

(1) Bet (A, C) B.

(2) Bet (B, C) A.

(3) Bet (A, B) C ∧ Bet (A, C) D.

(4) Bet (A, D) C.

(5) Bet (D, C) A.

ここで簡便化の為, 補題 3 および 4 を以下のように書き換える.

補題 3'　P, Q, R, S が一直線上の相異なる四点とする. このとき, Bet (P, R) Q ∧ Bet (Q, S) R ⟹ Bet (P, S) Q ∧ Bet (P, S) R.
補題 4'　P, Q, R, S が一直線上の相異なる四点とする. このとき, Bet (P, R) Q ∧ Bet (P, S) R ⟹ Bet (P, S) Q ∧ Bet (Q, S) R.

(1) の場合, Bet (A, B) D ∧ Bet (A, C) B である. ここで, A, B, C, D = P, R, S, Q と書き換えれば, Bet (P, R) Q ∧ Bet (P, S) R となる. よって補題 4' より Bet (P, S) Q ∧ Bet (Q, S) R. 書き換えを直せば, Bet (A, C) D ∧ Bet (D, C) B となる. 直線 a が点 D を通り かつ三点 D, C, B を含む直線とは異なることを示す. もし a がこの直線と同一であれば, 補題 1 より, B および C が a 上にあることとなり, a が A, B, C のいずれも通らないことに矛盾する. よって a は点 D を通り かつ三点 D, C, B を含む直線とは相異なる直線である. したがって, 補助命題 5 および補題 5 より Sameside : a (B, C) である.

(2) の場合, Bet (A, B) D ∧ Bet (B, C) A である. よって, HaⅡ-1 より, Bet (B, A) D ∧ Bet (B, C) A となる. ここで, A, B, C, D = R, P, S, Q と書き換えれば, Bet (P, R) Q ∧ Bet (P, S) R となる. よって補題 4' より Bet (P, S) Q ∧ Bet (Q, S) R. 書き換えを直せば, Bet (B, C) D ∧ Bet (D, C) A となる. 直線 a が点 D を通り かつ三点 D, C, A を含む直線とは異なることを示す. もし a がこの直線と同一であれば, 補題 1 より, C および A が a 上にあることとなり, a が A, B, C のいずれも通らないことに矛盾する. よって a は点 D を通り かつ三点 D, C, A を含む直線とは相異なる直線である. そのため補助命題 5, 補題 5 および補助命題 1 より, ¬ Diffside : a (A, C) となって, 仮定に矛盾する. よって点 C の位置は (2) の場合ではない.

(3) の場合, Bet (A, B) C ∧ Bet (A, C) D である. よって, HaⅡ-1 より, Bet (B, A) D ∧ Bet (C, A) D となる. ここで, A, B, C, D = R, P, Q, S と書き換えれば, Bet (P, R) Q ∧ Bet (Q, S) R となる. よって補題 3' より, Bet (P, S) Q ∧ Bet (P, S) R. 書き換えを直せば, Bet (B, D) C ∧ Bet (B, D) A となり, さらに HaⅡ-1 より, Bet (D, B) C となる. 直線 a が点 D を通り かつ三点 D, B, C を含む直線とは異なることを示す. もし a がこの直線と同一であれば, 補題 1 より, B および C が a 上にあることとなり, a が A, B, C のいずれも通らないことに矛盾する. よって a は点 D を通り かつ二点三点 D, B, C を含む直線とは相異なる直線である. したがって, 補助命題 5 より, Sameside : a (B, C) である.

(4) の場合, Bet (A, B) D ∧ Bet (A, D) C である. よって, HaⅡ-1 より, Bet (D, A) C となる. 直線 a が点 D を通り かつ三点 D, A, C を含む直線とは異なることを示す. もし a がこの直線と同一であれば, 補題 1 より, A および C が a 上にあることとなり, a が A, B, C のいずれも通らないことに矛盾する. よって a は点 D を通り かつ三点 D, A, C を含む直線とは相異なる直線である. そのため補助命題 5 および補助命題 1 より, ¬ Diffside : a (A, C) となって, 仮定に矛盾する. よって点 C の位置は (4) の場合ではない.

(5) の場合, Bet (A, B) D ∧ Bet (D, C) A である. このとき仮定に矛盾することを (2) の場合において既に証明済みである. よって点 C の位置は (5) の場合ではない.

点 C の位置としてあり得るすべての場合 (1), (3) において Sameside : a (B, C) であることが導かれる. したがって, 命題の求めるところが証明された.

補助命題 7　Bet (A, C) B であるとき, 点 B を通り かつ三点 A, C, B を含む直線とは相異なる直線を b とおけば, Diffside : b (A, C) である.
証明　HaI-3 より, 三点 A, C, B を含む直線 a の上にない点 D をとり, 二点 D, B を通る直線を b とする. このとき, Diffside : b (A, C) であると主張する. まず二直線 a, b が相異なる直線であることを示す. もし二直線が同一の直線であれば, 補題 1 より, a が D を通ることになり, a の上にない点 D をとったことに矛盾する. よって a, b は相異なる直線である. このとき, 線分 AC と直線 b は点 B を共有する. よって定義より, Diffside : b (A, C) である. したがって, 命題の求めるところが証明された.

補助命題 8　直線 a および相異なる三点 A, B, C について, Sameside : a (A, B) かつ Diffside : a (A, C) ならば, Diffside : a (B, C) である.

まず三点のうちの一点が残る二点の各々と結合する直線上に存在しない場合を考える. 点 C が直線 AB 上にないとすれば, 三点 A, B, C は一直線上にない点であり, 定義より, a は A, B, C のいずれも通らない直線であるため, HaⅡ-4 より, a が三つの線分の何れかを通るならば, 必ず残りの二線分のどちらか一方を通る. 定義より, a は線分 AC を通り, 線分 AB を通らないので, 残る線分 BC を

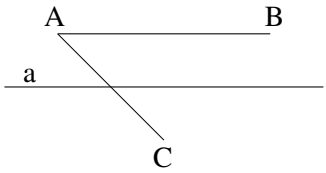

図 17: 補助命題 8

通る. よって Diffside : a (B, C) である. 残りの点においても同様に導ける.

次に三点が一直線 b 上にある場合を考える. まず a と線分 AC の共有する点を D とおく. Bet (A, C) D から, HaⅡ-1 より, D は A および C と相異なる点である. ここで, 二点 D, B が同一の点であれば, D は a 上の点であるため, B が a 上にあることとなり, a が A, B, C のいずれも通らないことに矛盾する. よって, 二点 D, B は相異なる点である. すなわち, 四点 A, B, C, D は相異なる点である. Bet (A, C) D から, HaⅡ-1 より, 三点 A, C, D は同一直線上の点である. b およびこの直線は, ともに相異なる二点 A, C を共有するため, 補題 1 より b の上にも D が存在する. よって, b は四点 A, B, C, D を通る. すなわち A, B, C, D は一直線上の相異なる四点である.

A, B, C, D が一直線上の相異なる四点であり, Bet (A, C) D であるため, 残る点 B の位置は, P, Q, R, S = A, D, C, B と変換することで Ht-5(1〜5) で示した通りに場合分けできる.

(1) Bet (A, B) C.

(2) Bet (C, B) A.

(3) Bet (A, C) B ∧ Bet (A, B) D.

(4) Bet (A, D) B.

(5) Bet (D, B) A.

(1) の場合, Bet (A, C) D ∧ Bet (A, B) C である. ここで, A, B, C, D = P, S, R, Q と書き換えれば, Bet (P, R) Q ∧ Bet (P, S) R となる. よって補題 4' より, Bet (P, S) Q ∧ Bet (Q, S) R. 書き換えを直せば, Bet (A, B) D ∧ Bet (D, B) C となる. 直線 a が点 D を通り かつ三点 A, B, D を含む直線とは異なることを示す. もし a がこの直線と同一であれば, 補題 1 より, A および B が a 上にあることとなり, a が A, B, C のいずれも通らないことに矛盾する. よって a は点 D を通り かつ三点 A, B, D を含む直線とは相異なる直線である. したがって, 補助命題 7 および補助命題 2 より, ¬ Sameside : a (A, B) となって, 仮定に矛盾する. よって点 B の位置は (1) の場合ではない.

(2) の場合, Bet (A, C) D ∧ Bet (C, B) A である. HaⅡ-1 より, Bet (C, A) D ∧ Bet (C, B) A となる. ここで, A, B, C, D = R, S, P, Q と書き換えれば, Bet (P, R) Q ∧ Bet (P, S) R となる. よって補題 4' より, Bet (P, S) Q ∧ Bet (Q, S) R. 書き換えを直せば, Bet (C, B) D ∧ Bet (D, B) A となる. 直線 a が点 D を通り かつ三点 C, B, D を含む直線とは異なることを示す. もし a がこの直線と同一であれば, 補題 1 より, C および B が a 上にあることとなり, a が A, B, C のいずれも通らないことに矛盾する. よって a は点 D を通り かつ三点 C, B, D を含む直線とは相異なる直線である. したがって, 補助命題 7 および補題 5 より, Diffside : a (B, C) である.

(3) の場合, Bet (A, C) B ∧ Bet (A, B) D である. このとき仮定に矛盾することを (1) の場合において既に証明済みである. よって点 B の位置は (3) の場合ではない.

(4) の場合, Bet (A, C) D ∧ Bet (A, D) B である. ここで, A, B, C, D = P, Q, S, R と書き換えれば, Bet (P, S) R ∧ Bet (P, R) Q となる. よって補題 4' より, Bet (P, S) Q ∧ Bet (Q, S) R. 書き換えを直せば, Bet (A, C) B ∧ Bet (B, C) D となり, さらに HaⅡ-1 より, Bet (C, B) D となる. このとき, Diffside : a (B, C) であることは, (2) の場合において既に証明済みである.

(5) の場合, Bet (A, C) D ∧ Bet (D, B) A である. HaⅡ-1 より, Bet (A, C) D ∧ Bet (B, D) A となる. ここで, A, B, C, D = Q, P, S, R と書き換えれば, Bet (Q, S) R ∧ Bet (P, R) Q となる. よって補題

3' より, Bet (P, S) Q ∧ Bet (P, S) R. 書き換えを直せば, Bet (B, C) A ∧ Bet (B, C) D となる. このとき, Diffside : a (B, C) であることは, (4) の場合において既に証明済みである.

点 B の位置としてあり得るすべての場合 (2), (4), (5) において, Diffside : a (B, C) であることが導かれる. したがって, 命題の求めるところが証明された.

補助命題 6, 8 により, 問題 1, 2 がそれぞれ解決される.

3.3.2 角の内外

角および角の内部の点は以下のように定義されている.

定義 A, A', O を一直線 a 上の三点とし, O は A と A' の間にはないとする. このとき点 A, A' は直線 a において点 O の同じ側にあるといい, 点 O の同じ側にある直線 a の全ての点を, 点 O から出る半直線という.

定義 h, k を一点 O より出でて相異なる直線に属する半直線とする. この二つの半直線の組を角と名付け, ∠(h, k) または ∠(k, h) で表す. 半直線 h, k を角の辺, 点 O を角の頂点という. また, 頂点が B, 各辺の上に各々点 A, C がある角を, ∠ABC と表す.

HaIII-4 与えられた平面上の与えられた半直線を一辺とし, この直線に対して与えられた側に任意の角を唯一通りに合同に移すことができる. 任意の角はそれ自身に合同である, 即ち常に ∠(h, k) ≡ ∠(h, k). また角の定義においてはその回転の向きを問題としない. そのため, 以下の記号は全て同意義である.

$$\angle(h,k) \equiv \angle(h',k'), \angle(h,k) \equiv \angle(k',h'), \angle(k,h) \equiv \angle(h',k'), \angle(k,h) \equiv \angle(k',h').$$

定義 半直線 h は直線 $\bar{h}$ に属し, 半直線 k は直線 $\bar{k}$ に属するとする. h および k は, これに点 O を加えれば, 平面上の残りの点を二つの領域に分ける : $\bar{k}$ に対して h と同じ側にあり かつ $\bar{h}$ に対して k と同じ側にあるすべての点をこの角の内部にあるといい, その他すべての点をこの角の外部にあるという.

これらの定義から, 以下の補題が導かれる.

補題 6 ∠ABC の内部に点 D があるならばまた, Sameside : BA (D, C) ∧ Sameside : BC (D, A) である.

補題 7 ∠ABC の外部に点 D があるならばまた, ¬ (Sameside : BA (D, C) ∧ Sameside : BC (D, A)) である.

ここで, この概念に関する以下の問題について, これらの定義から明らかである, あるいは特別な困難なしに辿り着くと主張するのは難しいと考える.

問題 3 h, k, l をそれぞれ点 O から出る相異なる直線に属する半直線とし, 半直線 l は直線 $\bar{l}$ に属するとする. h, k が $\bar{l}$ に対して同じ側にあるとき, h が ∠(k, l) の内部にあるか, あるいは k が ∠(h, l) の内部にある.

この問題は Ht-15 の証明の中で, 定義より明らかであると主張されている.

問題の解決のために, まずは以下の命題を証明する.

補助命題 9 ∠ABC の内部に点 D があるとき, Diffside : BD (A, C) である.

証明 定義より, 二直線 BA, BC は相異なる直線であり, また三点 A, B, C は相異なる点である. HaII-2 より, 直線 CB 上に存在して, Bet (C, E) B であるような点 E をとる. 二直線 EC, BC はともに相異なる二点 B, C を通るため, 補題 1 より, 同一の直線である. ここで, もし EC が点 A を通るならば, 二直線 BA, EC はともに相異なる二点 A, B を通るため, 補題 1 より, 同一の直線となり, したがって BA, BC が同一の直線となって, これは仮定に矛盾する. よって EC は A を通らない. E が A と同一の点である場合も同様の矛盾が起こるため, 二点 E, A は相異なる点である. また Bet (C, E) B から, HaII-1 より, 三点 C, E, B は相異なる三点である. よって三点 A, E, C は一直線上にない相異なる三点である.

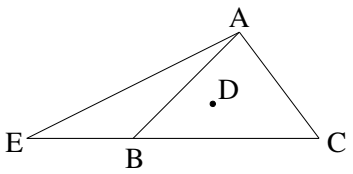

図 18: 補助命題 9

仮定および補題 6 より, Sameside : BA (D, C) ∧ Sameside : BC (D, A) である. Sameside : BA (D, C) から定義より, 点 D は直線 BA の上になく, また二点 D, C は相異なる点である. もし直線 BD の上に A があれば, 二直線 BD, BA はともに相異なる二点 A, B を含み, BD 上に D が存在するため, 補題 1 より, BA 上にも D が存在することとなって, これに矛盾する. よって, BD 上に A はない. また, 二点 D, A が相異なる点であり, BD 上に点 C がないことも, 定義から同様に導ける. ここでもし BD が E を通るならば, EC, BD がともに相異なる二点 B, E を通る直線となり, 補題 1 より BD の上に C が存在することとなって, これに矛盾する. よって BD は E を通らない. A, E, C が一直線上にない三点であり, BD がそのいずれも通らない直線であって, 線分 EC の点 B を通るため, HaⅡ-4 より, 直線 BD は線分 AE もしくは線分 AC を通る.

BD が線分 AE を通る場合を考える. このとき共通点を F とおけば, Bet (A, E) F である. 直線 BA が三点 A, E, F を含む直線と相異なる直線であることを示す. もし BA がこの直線と同一であれば, 補題 1 より, これらおよび直線 BC はともに相異なる二点 B, E を含むため, 補題 1 より, BA と BC が同一の直線となって, 仮定に矛盾する. よって BA は三点 A, E, F を含む直線と相異なる直線である. Bet (A, E) F および BA が点 A を通り かつ三点 A, E, F を含む直線と相異なる直線であることから, 補助命題 5 より, Sameside : BA (E, F) である. 直線 BA が三点 C, E, B を含む直線 a とも相異なる直線であることを示す. 二直線 a, EC はともに相異なる二点 E, C を通るため, 補題 1 より, a と EC は同一の直線である. ここで, もし BA が a と同一の直線であれば, EC は BC と同一の直線であるため, BA と BC が同一の直線となり, 仮定に矛盾する. よって, BA は a と相異なる直線である. Bet (C, E) B および BA が点 B を通り かつ三点 C, E, B を含む直線 a と相異なる直線であることから, 補助命題 7 より, Diffside : BA (E, C) である. Bet (A, E) F から, HaⅡ-1 より, 二点 E, F は相異なる点である. もし二点 C, F が同一の点であれば, 二直線 AE, EC がともに相異なる二点 E, F を含み, また AE の上に点 A が存在するため, 補題 1 より, EC の上に A が存在することとなり, EC が A を通らないことに矛盾する. よって二点 C, F は相異なる点であり, したがって三点 E, F, C が相異なる点となる. 相異なる三点 E, F, C について, Sameside : BA (E, F) かつ Diffside : BA (E, C) であることから, 補助命題 8 より, Diffside : BA (C, F) である. もし二点 D, F が同一の点であれば, Diffside : BA (C, F) より Diffside : BA (C, D) となり, 補題 5 および補助命題 2 より, ¬ Sameside : BA (D, C) となって, Sameside : BA (D, C) であることに矛盾する. よって, 二点 D, F は相異なる点であり, したがって三点 C, D, F が相異なる点となる. 相異なる三点 C, D, F について, Sameside : BA (C, D) かつ Diffside : BA (C, F) であることから, 補助命題 8 および補題 5 より, Diffside : BA (F, D) である. 仮定より二点 B, D は相異なる点である. もし二点 B, F が同一の点であれば, 二直線 AE, EC がともに相異なる二点 B, E を含み, また AE の上に点 A が存在するため, 補題 1 より, EC の上に A が存在することとなり, EC が A を通らないことに矛盾する. よって二点 B, F は相異なる点であり, したがって三点 B, D, F は一直線上にある相異なる点である. このとき Ht-4 および補題 2 より, この三点の位置は以下の三つの場合に区別される.

(1) Bet (B, D) F ∧ ¬ Bet (D, F) B ∧ ¬ Bet (F, B) D.

(2) ¬ Bet (B, D) F ∧ Bet (D, F) B ∧ ¬ Bet (F, B) D.

(3) ¬ Bet (B, D) F ∧ ¬ Bet (D, F) B ∧ Bet (F, B) D.

もし直線 BA が三点 B, D, F を含む直線 b と同一の直線であれば, 補題 1 より, b の上に A が存

在することとなる. 二直線 b, BD はともに相異なる二点 B, D を含み, また b の上に A が存在するため, 補題 1 より, BD の上に A が存在することとなり, A が BD の上にないことに矛盾する. よって BA は b と相異なる直線である. したがって (1) の場合, Bet (B, D) F および BA が点 B を通り かつ三点 B, D, F を含む直線 a と相異なる直線であることから, 補助命題 5, 補題 5 および補助命題 1 より, ¬ Diffside : BA (F, D) となり, Diffside : BA (F, D) であることに矛盾する. よって三点の位置は (1) の場合ではない. 同様に, 直線 BA は三点 B, F, D を含む直線とも相異なる直線である. したがって (3) の場合, 定義より, Bet (B, F) D および BA が点 B を通り かつ三点 B, F, D を含む直線とは相異なる直線であることから, 補助命題 5 および補助命題 1 より, ¬ Diffside : BA (F, D) となり, Diffside : BA (F, D) であることに矛盾する. よって三点の位置は (3) の場合ではない. したがって, 三点の位置は (2) の場合であり, すなわち, Bet (D, F) B である.

Sameside : BC (D, A) から, 定義より, BC は D を通らない直線である. もし直線 BC が三点 D, F, B を含む直線と同一であれば, 補題 1 より, BC の上に D が存在することとなり, これに矛盾する. よって BC は三点 D, F, B を含む直線とは相異なる直線である. したがって, Bet (D, F) B および BC が点 B を通り かつ三点 D, F, B を含む直線とは相異なる直線であることから, 補助命題 7 および補題 5 より, Diffside : BC (F, D) である. もし直線 EC が三点 E, A, F を含む直線と同一の直線であれば, 三点 A, E, C が同一直線上にあることになり, これらが一直線上にない三点であることに矛盾する. よって EC は三点 E, A, F を含む直線とは相異なる直線である. Bet (A, E) F より, 定義から Bet (E, A) F であって, EC が点 E を通り, 三点 E, A, F を含む直線とは相異なる直線であることから, 補助命題 5 および補題 5 より, Sameside : EC (F, A) である. ここで, EC と BC は同一の直線であるため, Sameside : BC (F, A) となる. Bet (A, E) F より, 二点 A, F は相異なる点である. したがって, 三点 F, A, D は相異なる点である. 相異なる三点 F, A, D について, Sameside : BC (F, A) かつ Diffside : BC (F, D) であることから, 補助命題 8 より, Diffside : BC (A, D) となり, 補題 5 および補助命題 2 より, ¬ Sameside : BC (D, A) となって, Sameside : BC (D, A) であることに矛盾する. よって, 直線 BD は線分 AE を通らない, すなわち線分 AC を通る. このとき, 定義より, Diffside : BD (A, C) である. したがって, 命題の求めるところが証明された.

補助命題 10　∠ABC の内部に点 D があり, 直線 BD 上の二点 B, D と相異なる点 E があって, ¬ Bet (E, D) B であるとき, E は ∠ABC の内部の点である.

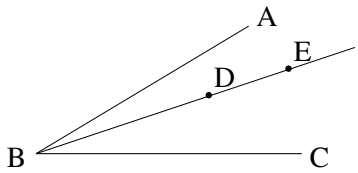

図 19: 補助命題 10

この補題は, D が ∠ABC の内部の点であれば, 半直線 BD は ∠ABC の内部にあることを表している. 以下で証明を示す.

証明　∠ABC の内部に D があるので, 定義より, Sameside : BA (C, D) かつ Sameside : BC (A, D) である. Diffside : BA (C, E) であるとする. このとき定義より, 二点 C, E は相異なる点であり, また仮定より, 二点 C, D および二点 D, E もまたそれぞれ相異なる点である. すなわち, 三点 C, D, E は相異なる点である. 相異なる三点 C, D, E について, Sameside : BA (C, D) かつ Diffside : BA (C, E) であることから, 補助命題 8 より, Diffside : BA (D, E) となる. このとき定義より, BA 上に点 F があって Bet (E, D) F である. もし二点 F, B が同一の点であれば, Bet (E, D) B となり, 仮定に矛盾する. よって, 二点 F, B は相異なる点である. ここで仮定より, 二直線 BD, ED はともに相異なる二点 E, D を含み, また Bet (E, D) F から定義より, ED が F を含むため, 補題 1 より, BD が F を通る. このとき, 二直線 BA, BD はともに相異なる二点 B, F を含み, また BD が D を含むため, 補題 1 より, BA が D を通ることとなり, これは仮定に矛盾する. よって, ¬ Diffside : BA (C, E) である. Sameside : BA

(C, D) から定義より, BA は C を通らない. ここで, 二直線 ED, BD はともに相異なる二点 E, D を含み, また BD が B を含むため, 補題 1 より, ED が B を通る. もし BA が E を通れば, 二直線 ED, BA がともに相異なる二点 C, E を含み, また ED が D を含むため, 補題 1 より, BA が D を通る. しかし, Sameside : BA (C, D) から定義より, BA は D を通らないことに矛盾する. よって, BA は E を通らない. 相異なる二点 C, E について, ¬ Diffside : BA (C, E) かつ, それぞれが BA 上の点でないため, 補助命題 4 より, Sameside : BA (C, E) である. Sameside : BC (A, E) であることも同様にして証明できる. したがって, ∠ABC の内部に D があり, 直線 BD 上の二点 B, D と異なる点 E があって, ¬ Bet (E, D) B であるとき, E は ∠ABC の内部の点である. これで, 命題の求めるところが証明された.

補助命題 11　∠ABC に対して, Bet (A, C) D ならば, D は ∠ABC の内部の点である.
証明　Sameside : BA (D, C) かつ Sameside : BC (D, A) であることを示す. 定義より, 二直線 BA, BC は相異なる直線であり, また三点 A, B, C は相異なる点であり, さらに三点 A, C, D もまた相異なる点である. もし, 直線 BA と三点 A, C, D を含む直線 a が同一の直線であれば, BA の上に点 B が存在するため, 補題 1 より, a が B を含む. すると, 直線 BC と a がともに相異なる二点 B, C を含むため, 補題 1 よりこれらが同一の直線, すなわち BA, BC が同一の直線となって, 二直線が相異なる直線であることに矛盾する. よって, BA は a と相異なる直線である. BC と a についても同様に相異なる直線であることを導ける. Bet (A, C) D であり, BA が三点 A, C, D を含む直線と相異なる直線であるため, 補助命題 5 より, Sameside : BA (D, C) である. さらに, 定義より Bet (C, A) D であり, BC が三点 A, C, D を含む直線と相異なる直線であるため, 補助命題 5 より, Sameside : BC (D, A) である. Sameside : BA (D, C) かつ Sameside : BC (D, A) であるため, 定義より, D は ∠ABC の内部の点である. これで, 命題の求めるところが証明された.

補助命題 12　Sameside : BC (A, D) であり, 二直線 BA, BD が相異なる直線ならば, ∠ABC の外部に D があり かつ ∠DBC の外部に A があることはない.
証明　もし, ∠ABC の外部に D があり かつ ∠DBC の外部に A があるならば, 補題 7 より, 以下の九つの場合に区別される.

(1) ¬ Sameside : BA (D, C) ∧ Sameside : BC (D, A) ∧ ¬ Sameside : BD (A, C) ∧ Sameside : BC (A, D).

(2) ¬ Sameside : BA (D, C) ∧ Sameside : BC (D, A) ∧ Sameside : BD (A, C) ∧ ¬ Sameside : BC (A, D).

(3) ¬ Sameside : BA (D, C) ∧ Sameside : BC (D, A) ∧
¬ Sameside : BD (A, C) ∧ ¬ Sameside : BC (A, D).

(4) Sameside : BA (D, C) ∧ ¬ Sameside : BC (D, A) ∧ ¬ Sameside : BD (A, C) ∧ Sameside : BC (A, D).

(5) Sameside : BA (D, C) ∧ ¬ Sameside : BC (D, A) ∧ Sameside : BD (A, C) ∧ ¬ Sameside : BC (A, D).

(6) Sameside : BA (D, C) ∧ ¬ Sameside : BC (D, A) ∧
¬ Sameside : BD (A, C) ∧ ¬ Sameside : BC (A, D).

(7) ¬ Sameside : BA (D, C) ∧ ¬ Sameside : BC (D, A) ∧
¬ Sameside : BD (A, C) ∧ Sameside : BC (A, D).

(8) ¬ Sameside : BA (D, C) ∧ ¬ Sameside : BC (D, A) ∧
Sameside : BD (A, C) ∧ ¬ Sameside : BC (A, D).

(9) ¬ Sameside : BA (D, C) ∧ ¬ Sameside : BC (D, A) ∧
¬ Sameside : BD (A, C) ∧ ¬ Sameside : BC (A, D).

仮定より, Sameside : BC (A, D) であるため, このうち (4), (5), (6), (7), (8), (9) の場合は有り得ない. また補題 5 より, Sameside : BC (D, A) となるため, 残りの内の (2), (3) の場合も否定される. そのため, 残る (1) の場合について, 矛盾を導けばよい.

∠ABC について注目する. 定義より, 二直線 BA, BC は相異なる直線であり, また三点 A, B, C は相異なる点である. HaⅡ-2 より, 直線 CB 上に存在して, Bet (C, E) B であるような点 E をとる. 二直

線 EC, BC はともに相異なる二点 B, C を通るため, 補題 1 より, 同一の直線である. ここで, もし EC が点 A を通るならば, 二直線 BA, EC はともに相異なる二点 A, B を通るため, 補題 1 より同一の直線となる. したがって, BA, BC が同一の直線となり, これは仮定に矛盾する. よって, EC は A を通らない. ここで, E が A と同一の点である場合, これに矛盾するため, 二点 E, A は相異なる点である. さらに, Bet (C, E) B から, HaⅡ-1 より, 三点 C, E, B は相異なる三点である. よって三点 A, E, C は一直線上にない相異なる三点である. (1) の場合において, ¬ Sameside : BD (A, C) である. もし BD が A を通るならば, 二直線 BA, BD がともに相異なる二点 A, B を含むため, 補題 1 より, BA, BD が同一の直線となって, 仮定に矛盾する. よって, BD は A を通らない. もし BD が C を通るならば, 二直線 BD, BC がともに相異なる二点 B, C を含み, BD が D を含むため, 補題 1 より BC が D を含む. しかしこのとき, Sameside : BC (D, A) から定義より, BC は D を含まないことに矛盾する. よって, BD は C を通らない. 相異なる二点 A, C について, ¬ Sameside : BD (A, C) かつ, それぞれが BD 上の点でないため, 補助命題 3 より, Diffside : BD (A, C) である. ここで, もし BD が E を通るならば, EC, BD がともに相異なる二点 B, E を通る直線となり, 補題 1 より, BD の上に C が存在することとなって, BD が C を通らないことに矛盾する. よって BD は E を通らない. したがって, A, E, C が一直線上にない三点であり, BD がそのいずれも通らない直線であって, 線分 EC の点 B を通るため, HaⅡ-4 より, 直線 BD は線分 AE もしくは線分 AC を通る.

BD が線分 AE を通る場合を考える. 定義および補題 5 より, Diffside : BD (E, A) である. もし, BD が三点 C, E, B を含む直線と同一ならば, BD が E を通らないことに矛盾する. よって, この二直線は相異なる直線である. したがって, Bet (C, E) B および BD が点 B を通り かつ三点 C, E, B を含む直線とは相異なる直線であることから, 補助命題 7 および補題 5 より, Diffside : BD (E, C) である. すなわち, 三点 A, E, C が相異なる三点であって, Diffside : BD (E, A) かつ Diffside : BD (E, C) であるため, 補助命題 6 および補助命題 1 より, ¬ Diffside : BD (A, C) となって, Diffside : BD (A, C) であることに矛盾する.

BD が線分 AC を通る場合を考える. このとき, 共有点を F とおけば, F は BD 上の点であって, Bet (A, C) F である. またこのとき, 補助命題 11 より, F は ∠ABC の内部の点である. もし, 二点 D, F が同一の点であれば, D が ∠ABC の内部の点となる. D, F が相異なる点である場合を考える. もし, 二点 B F が同一の点であれば, Bet (A, C) F より, Bet (A, C) B となり, 定義より, 三点 A, C, B を含む直線 a が存在する. このとき, 補題 1 より, 直線 BA および BC がそれぞれ a と同一の直線となり, したがって二直線が同一の直線となって, 仮定に矛盾する. よって, 二点 B, F は相異なる点である. すなわち, 三点 B, D, F は一直線上の相異なる点である. このとき Ht-4 および補題 2 より, この三点の位置は以下の三つの場合に区別される.

(i) Bet (B, D) F ∧ ¬ Bet (D, F) B ∧ ¬ Bet (F, B) D.

(ii) ¬ Bet (B, D) F ∧ Bet (D, F) B ∧ ¬ Bet (F, B) D.

(iii) ¬ Bet (B, D) F ∧ ¬ Bet (D, F) B ∧ Bet (F, B) D.

(ii) の場合について考える. 補題 1 より, 直線 BD と, 三点 D, F, B を含む直線 b は同一の直線である. もし, 三点 D, F, B を含む直線 b と直線 EC が同一の直線であれば, 補題 1 より, 直線 BD の上に E が存在することとなり, BD が E を通らないことに矛盾する. よって, EC と b は相異なる直線である. また, Bet (C, E) B なので, 補題 1 より, 直線 EC と三点 C, E, B を含む直線は同一の直線であり, すなわち EC が B を通る. したがって, Bet (D, F) B および EC が B を通り かつ三点 D, F, B を含む直線と相異なる直線であることから, 補助命題 7 および補題 5 より, Diffside : EC (F, D) である. もし, 三点 A, C, F を含む直線 c と直線 EC が同一の直線であれば, 補題 1 より EC 上に A が存在することとなり, EC が A を通らないことに矛盾する. よって, EC と c は相異なる直線である. Bet (A, C) F から, 定義より, Bet (C, A) F である. したがって, Bet (C, A) F および EC が C を通り かつ三点 A, C, F を含む直線と相異なる直線であることから, 補助命題 5 より, Sameside : EC (F, A) である. Bet (A, C) F から, 定義より, 二点 A, F は相異なる点である. また, Bet (D, F) B から, 定義より, 二点 D, F は相異なる点である. 二点 D, A が相異なる点であるため, 三点 F, A, D は相異なる点である. 三点 F, A, D について, Sameside : EC (F, A) かつ Diffside : EC (F, D) であることから, 補助命題 8 より, Diffside : EC (A, D) である. また, 二直線 BC, EC が同一の直線であるため, Diffside : BC (A, D), すなわち補助命題 2 より, ¬ Sameside : BC (A, D) となって, Sameside : BC (A, D) であることに矛盾する. よって, (ii) の場合ではない.

(i) または (iii) の場合, ¬ Bet (D, F) B である. ∠ABC の内部に F があり, 直線 BF 上に二点 B, F と異なる点 D があって, ¬ Bet (D, F) B なので, 補助命題 10 より, D は ∠ABC の内部の点である. よってこの場合も, ∠ABC の外部に D があり かつ ∠DBC の外部に A があることはない. 全ての場合において矛盾が導かれる. したがって, Sameside : BC (A, D) であり, 二直線 BA, BD が相異なる直線ならば, ∠ABC の外部に D があり かつ ∠DBC の外部に A があることはない. これで, 命題の求めるところが証明された.

補助命題 13　h, k, l をそれぞれ点 O から出る相異なる直線に属する半直線とし, 半直線 l は直線 $\bar{l}$ に属するとする. h, k が $\bar{l}$ に対して同じ側にあるとき, h が ∠(k, l) の内部にあるか, あるいは k が ∠(h, l) の内部にある.

証明　半直線 h, k, l 上の点をそれぞれ H, K, L とし, 半直線 h, k がそれぞれ直線 $\bar{h}, \bar{k}$ に属するとする. もし H が K に等しければ, 二直線 $\bar{h}, \bar{k}$ がそれぞれ相異なる二点 O, H を通るため, 補題 1 より, $\bar{h}, \bar{k}$ が同一の直線となって, 仮定に矛盾する. よって二点 H, K は相異なる点である. H, L および K, L についても同様である. したがって, 三点 H, K, L は相異なる点である.

h が ∠(k, l) の内部にあるか, あるいは k が ∠(h, l) の内部にあることを示すには, 以下の二つの条件について, それぞれ矛盾を導けばよい.

(1) h が ∠(k, l) の内部にあり かつ k が ∠(h, l) の内部にある.

(2) h が ∠(k, l) の外部にあり かつ k が ∠(h, l) の外部にある.

(1) の場合について考える. 仮定より, h が ∠(k, l) の内部にあるため, h 上の点 H は ∠(k, l) の内部にある. よって, 補助命題 2 9 より, ¬ Sameside : $\bar{h}$ (K, L) である. 一方, k が ∠(h, l) の内部にあることから, 補題 6 より Sameside : $\bar{h}$ (K, L) となり, これに矛盾する.

次に (2) の場合を考える. 仮定より, h が ∠(k, l) の外部にあるため, h 上の点 H は ∠(k, l) の外部にある. 同様に, k が ∠(h, l) の外部にあるため, k 上の点 K は ∠(h, l) の外部にある. 仮定より, Sameside : $\bar{l}$ (H, K) である. このとき, 補助命題 12 より, H が ∠(k, l) の外部にあり かつ K が ∠(h, l) の外部にあることはないため, これに矛盾する.

全ての条件において矛盾が導かれる. したがって, 命題の求めるところが証明された.

補助命題 13 により問題 3 は解決される.

3.3.3　三角形の実在

三角形は以下のように定義されている.

定義　線分 AB, BC, CD, . . . , KL の組を点 A を点 L に連結する折線といい, 点 A, B, C, D, . . . , K, L を折線の点という. 特に点 A, B, C, D, . . . , K, L が全て同一平面上にあり かつ点 L が点 A に一致するとき, この折線を多角形と名付け, 点 A, B, C, D, . . . , K を多角形の頂点という. 3 個, 4 個, . . . , n 個の頂点を持つ多角形をそれぞれ三角形, 四角形, . . . , n 角形という.

定義　多角形の頂点が全て互いに異なり, 多角形の頂点がその一辺の上に在ること無く かつ多角形の如何なる二辺も共通点がないとき, これを単一多角形という.

[9] において, この二つの定義からなる単一三角形と呼ぶべき図形を, 単に三角形と表し, 様々な公理および定理に使用している. しかしながら, この概念に関する以下の問題について, これらの定義から明らかである, あるいは特別な困難なしに辿り着くと主張するのは難しいと考える.

問題 4　ある三角形において, その一辺を含む直線上に, その直線上にある頂点と相異なる点をとったとき, その点およびその辺の端点の一方と, 三角形の残りの頂点は三角形をつくる.

この問題が関係する箇所の一例として, Ht-12 (三角形の第一合同定理) の証明の中で, 三角形 ABC, 三角形 A'B'C' を与え, BC ≠ B'C' を仮定し, B'C' 上に BC = B'D' となるような点 D' をとって, A'B'D' に対しそれが三角形であることを前提とする HaⅢ-5 を適用している. 線分を合同に移すことの一意可能性はこの以前に証明されており, したがってこのとき点 D' が二点 B', C' のいず

れとも相異なる点であることは容易に導ける. つまりこのときの三点 A', B', D' が三角形をつくることの保証は, この問題の示すところを主張していると考えられる.

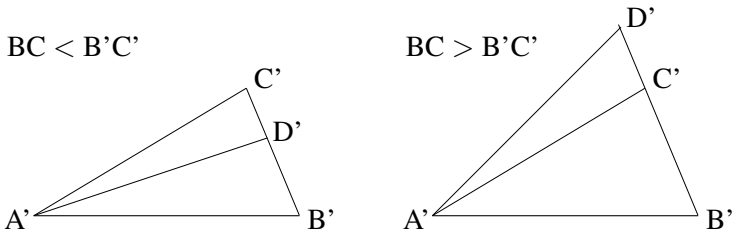

図 20: HaIII-5

ここでの三角形の定義を, 以下のように書き直す.

定義　三点 A, B, C が相異なる点であり, そのうちどの一点をとっても他の二点の間に在ることなくかつどの二辺をとっても共通点はないとき, これを三角形 ABC という. またこのとき, 三点 A, B, C が三角形 ABC をつくるとする.

本論文では,「三点 A, B, C が三角形 ABC をつくる」を以後「△ABC」のように表記する.
この定義から, 以下の命題を証明する.

補助命題 14　△ABC のとき, どの二頂点をとっても, それらを通る直線上に残りの頂点はない.
証明　定義より, 三点 A, B, C は相異なる点であり, そのうちどの一点をとっても他の二点の間に在ることがない. もし直線 AB 上に C があるならば, A, B, C は一直線上にある相異なる三点であるため, Ht-4 より, 他の二点の間にある一点が常に存在することとなり, これは定義に矛盾する. よって直線 AB 上に C はない. 直線 BC, 直線 CA についても同様である. したがって, 命題の求めるところが導かれる.

補助命題 15　三点が相異なる点であり, 二点を通る直線上に残りの一点がないならば, この三点は三角形をつくる.
証明　相異なる三点について, どの一点をとっても他の二点の間に在ることなく, かつどの二線分をとっても共通点はないことを示すには, 相異なる三点をそれぞれ A, B, C とおき, 以下の六つの条件について, それぞれ矛盾を導けばよい.

(1) Bet (A, B) C.
(2) Bet (B, C) A.
(3) Bet (C, A) B.
(4) 線分 AB と線分 BC に共通点がある.
(5) 線分 BC と線分 CA に共通点がある.
(6) 線分 CA と線分 AB に共通点がある.

(1), (2), (3) において, 定義より三点 A, B, C が同一直線 a 上に存在する. a と直線 AB はそれぞれ相異なる二点 A, B を通り, また a 上に C が存在するため, 補題 1 より, AB 上に C が存在することとなり, 仮定に矛盾する.

(4) において, 線分 AB と線分 BC の共通点を点 D とする. すなわち Bet (A, B) D ∧ Bet (B, C) D である. 定義より, 三点 A, B, D が同一直線 b 上に存在する. 二直線 b, AB はそれぞれ相異なる二点 A, B を通り, また b 上に D が存在するため, 補題 1 より, AB 上に D が存在する. また同様に直線 BC 上にも D が存在する. Bet (A, B) D から, HaII-1 より, 二点 B, D は相異なる点である. このとき, 二直線 BC, AB はそれぞれ相異なる二点 B, D を通り, また BC 上に C が存在するため, 補題 1 より AB 上に C が存在することとなり, 仮定に矛盾する.

(5) において, 線分 BC と線分 CA の共通点を点 E とする. すなわち Bet (B, C) E ∧ Bet (C, A) E である. 定義より三点 B, C, E が同一直線 c 上に存在する. 二直線 c, BC はそれぞれ相異なる二点 B, C を通り, また c 上に E が存在するため, 補題 1 より BC 上に E が存在する. また同様に直線 CA 上にも E が存在する. Bet (B, C) E から, HaⅡ-1 より, 二点 C, E は相異なる点である. このとき, 二直線 CA, BC はそれぞれ相異なる二点 C, E を通り, また CA 上に A が存在するため, 補題 1 より BC 上に A が存在する. したがって, 相異なる三点 A, B, C が同一直線 BC 上に存在する. このとき, 二直線 BC, AB はそれぞれ相異なる二点 A, B を通り, また BC 上に C が存在するため, 補題 1 より AB 上に C が存在することとなり, 仮定に矛盾する.

(6) において, 線分 CA と線分 AB の共通点を点 F とする. すなわち Bet (C, A) F ∧ Bet (A, B) F である. 定義より三点 C, A, F が同一直線 d 上に存在する. 二直線 d, CA はそれぞれ相異なる二点 C, A を通り, また d 上に F が存在するため, 補題 1 より直線 CA 上に F が存在する. また同様に直線 AB 上にも F が存在する. Bet (C, A) F から, HaⅡ-1 より, 二点 A, F は相異なる点である. このとき, 二直線 CA, AB はそれぞれ相異なる二点 A, F を通り, また CA 上に C が存在するため, 補題 1 より AB 上に C が存在することとなり, 仮定に矛盾する.

全ての条件において矛盾が導かれる. したがって, 命題の求めるところが証明された.

補助命題 16　△ABC のとき, △BCA ∧ △CAB ∧ △CBA ∧ △BAC ∧ △ACB.
証明　定義より三点 A, B, C は相異なる点であり, 補助命題 14 より直線 BC 上に A はない. したがって, 補助命題 15 より △ABC のとき, △BCA である. 同様にして, 補助命題 15 より △BCA のとき, △CAB. すなわち, △ABC のとき, △CAB である. 次に, 定義より三点 A, B, C は相異なる点であり, 補助命題 14 より直線 BC 上に A はない. ここで, 二直線 CB, BC はそれぞれ相異なる二点 B, C を通るため, もし CB 上に A が存在すれば, 補題 1 より BC 上に A が存在することとなり, これに矛盾する. したがって, 直線 CB 上に A はない. よって, 補助命題 15 より △ABC のとき, △CBA である. 同様にして, 補助命題 15 より △CAB のとき, △BAC. すなわち, △ABC のとき, △BAC である. さらに同様に, 補助命題 15 より △BCA のとき, △ACB. すなわち, △ABC のとき, △ACB である. したがって, △ABC のとき, △BCA ∧ △CAB ∧ △CBA ∧ △BAC ∧ △ACB. これで命題の求めるところが証明された.

補助命題 17　ある三角形において, その一辺を含む直線上に, その直線上にある頂点と相異なる点をとったとき, その点およびその辺の端点の一方と, 三角形の残りの頂点は三角形をつくる.
証明　与えられた三角形をつくる三点を A, B, C とおき, 辺 BC を含む直線上に B, C と相異なる点 D をとったとき, △ABD ∧ △ACD を主張する. 定義より三点 A, B, C は相異なる点であり, 補助命題 14 より直線 BC 上に A はない. また仮定より三点 B, C, D は一直線 a 上にある相異なる三点である. 二直線 a, BD はそれぞれ相異なる二点 B, D を通り, また a 上に C が存在するため, 補題 1 より BD 上に C が存在する. 二直線 BD, BC はそれぞれ相異なる二点 B, C を通るため, もし BD 上に A が存在すれば, 補題 1 より BC 上に A が存在することとなり, BC 上に A がないことに矛盾する. よって, BD 上に A はない. ここで, BD 上には D が存在するため, もし二点 D, A が同一の点であれば, BD 上に A が存在することとなり, これに矛盾する. よって二点 D, A は相異なる点である. したがって, 三点 A, B, D は相異なる点であり, 直線 BD 上に A がないので, 定理 8 より △ABD. また二直線 a, CD はそれぞれ相異なる二点 C, D を通り, また a 上に B が存在するため, 補題 1 より CD 上に B が存在する. 二直線 CD, BD はそれぞれ相異なる二点 B, D を通るため, もし CD 上に A が存在すれば, 補題 1 より BD 上に A が存在することとなり, BD 上に A がないことに矛盾する. よって, CD 上に A はない. ここで, CD 上には C が存在するため, もし二点 C, A が同一の点であれば, CD 上に A が存在することとなり, これに矛盾する. よって二点 C, A は相異なる点である. したがって, 三点 A, C, D は相異なる点であり, 直線 CD 上に A がないので, 定理 8 より △ACD. これで命題の求めるところが証明された.

補助命題 17 により問題 4 は解決される.

3.3.4　合同な二線分と間

線分を合同に移し得ることについては, 以下の公理が与えられている.

HaIII-1　A, B を一直線 a 上の二点とし, 更に A’ を同じ直線または他の直線 a’ 上の点とするとき, 直線 a’ 上の A’ に対して与えられた側に常に少なくとも一点 B’ を見出し, 線分 AB が線分 A’B’ に合同または相等しくなるようにすることが出来る.

本論文では,「線分 AB が線分 A’B’ に合同」を以後「AB ≡ A’B’」のように表記する.
線分の定義および HaIII-2 より, 以下もそれぞれ成り立つとされる.

AB ≡ A’B’ ⟹ AB ≡ B’A’, BA ≡ A’B’, BA ≡ B’A’.

AB ≡ AB.

AB ≡ A’B’ ⟹ A’B’ ≡ AB.

AB ≡ A’B’ ∧ A’B’ ≡ A”B” ⟹ AB ≡ A”B”.

先述の通り, 線分を合同に移すことの一意可能性は [9] において証明されている. ここで以下のような命題を考える.

補題 8　一直線 a 上の三点 A, B, C において, A, B が a において C の同じ側にあるならば, ¬ Bet (A, B) C であり, A, B が a において C の異なる側にあるならば, Bet (A, B) C である. また, 三点 A, B, C が一直線 a 上の点であり, ¬ Bet (A, B) C ならば, A, B が a において C の同じ側にある.

補助命題 18　一直線上の三点 A, B, C において, A, B および A, C がそれぞれ相異なる点であり, ¬ Bet (B, C) A かつ AB ≡ AC であるならば, B, C は同一の点である.

補題 8 は半直線の定義から即座に導かれる. 以下に補助命題 18 の証明を示す.

証明　仮定および補題 8 より, B, C が三点を含む直線において A と同じ側にある. もし, B, C が相異なる点であれば, 点 A から出る半直線上に線分 AB と合同な二通りの線分を移せることとなり, 線分を合同に移すことの一意可能性に矛盾する. したがって, 命題の求めるところが導かれる.

ここで, 合同な二線分による, ある関係を表す公理を示す.

HaIII-3　AB および BC を直線 a 上の共通点のない二線分, さらに A’B’ および B’C’ を同じ直線または他の直線 a’ 上にあって同様に共通点のない二線分とする. このとき, AB ≡ A’B’ かつ BC ≡ B’C’ ならば, 常にまた AC ≡ A’C’ である.

この公理の利用のため, 以下の命題を証明する.

補助命題 19　Bet (A, C) B ならば, 二線分 AB, BC は共通点をもたない.
証明　定義より三点 A, B, C は一直線上に存在する相異なる三点である. AB, BC が共有点 D をもつとする. このとき, Bet (A, B) D ∧ Bet (B, C) D である. ここで定義より, 三点 A, B, D および三点 B, C, D もまた一直線上に存在する相異なる三点であり, それぞれが存在する三直線がいずれも相異なる二点を共有するため, HaI-2 より同一の直線である. すなわち A, B, C, D は一直線 a 上の相異なる四点である. ここで, A, B, C, D ⟹ S, R, P, Q と書き換えれば, Bet (B, C) D ∧ Bet (A, C) B ⟹ Bet (R, P) Q ∧ Bet (S, P) R から, 定義より Bet (P, R) Q ∧ Bet (P, S) R となる. よって補題 4’ より Bet (P, S) Q. 書き換えを直せば, Bet (C, A) D となる. HaI-3 より, 直線 a 上にない点 E をとり, 二点 E, D を通る直線を b とする. 二直線 a, b が同一の直線であれば, 補題 1 より a 上に E が存在することとなって矛盾する. よって a, b は相異なる直線である. a が三点 C, A, D を含むため, 補助命題 7 より, Diffside : b (C, A) となる. Bet (A, B) D ∧ Bet (B, C) D から, 同様にして, Diffside : b (A, B), Diffside : b (B, C) が導ける. また補題 5 より, Diffside : b (B, A) である. 三点 B, C, A が相異なる三点であり, Diffside : b (B, C) かつ Diffside : b (B, A) となるため, 補助命題 1, 6 より, ¬ Diffside : b (C, A) となって, Diffside : b (C, A) に矛盾する. よって, 二線分 AB, BC は共通点をもたない. これで命題の求めるところが証明された.

ここで, Ht-15 の証明過程の一部を示す.

・ h, k, l および h’, k’, l’ は, それぞれ点 O, O’ から出る半直線.

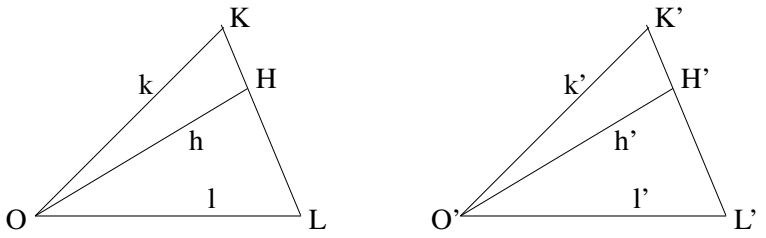

図 21: Ht-15

・ h, k および h’, k’ はそれぞれ l および l’ に対して同時に同じ側にある.

・ h が ∠(k, l) の内部にある.

・ K, L はそれぞれ k, l 上にとられた点.

・ K’, L’ はそれぞれ, OK ≡ O’K’, OL ≡ O’L’ となるように k’, l’ 上にとられた点.

・ H は線分 KL と h の交点.

・ H’ は OH ≡ O’H’ となるように h’ 上にとられた点.

・ LK ≡ L’K’ および LH ≡ L’H’ は証明済み.

[9] では, ここで HaⅢ-3 を用いて HK ≡ H’K’ を導けると主張している.

条件より Bet (L, K) H が導かれるため, Bet (L’, K’) H’ が判明すれば, 補助命題 19 より HaⅢ-3 の要件を満たすことが分かる. また同様に ¬ Bet (K’, H’) L’ も条件から導かれるため, HaⅡ-3 および Ht-4 より, Bet (L’, K’) H’ ∧ ¬ Bet (L’, H’) K’ または ¬ Bet (L’, K’) H’ ∧ Bet (L’, H’) K’ であることには容易に辿り着く.

そこで, 上記の条件下における Bet (L’, H’) K’ の否定を目的として, 以下の問題を考える.

問題 5　二直線 a, a’ 上にそれぞれ三点 A, B, C および A’, B’, C’ が存在し, 以下の条件が与えられたとする. このとき, 少なくとも一つの条件が誤りである.

(1) Bet (A, C) B.

(2) Bet (A’, B’) C’.

(3) AB ≡ A’B’.

(4) AC ≡ A’C’.

問題の解決のため, 以下の命題を証明する.

補助命題 20　A, B, C を Bet (A, C) B であるような一直線 a 上の三点とする. また直線 a’ 上に三点 A’, B’, C’ が存在し, A’, B’ および A’, C’ がそれぞれ相異なる点であるとする. このとき, AB ≡ A’B’, AC ≡ A’C’ かつ B’, C’ が a’ において A’ の同じ側にあるならば, Bet (A’, C’) B’ である.

証明　B, A が a 上の二点であることから, HaⅢ-1 より, a’ 上の B’ に対して A’ と同じ側に点 A” を見出し, BA ≡ B’A” とする. また, B, C が a 上の二点であることから, HaⅢ-1 より, a’ 上の B’ に対して A” と異なる側に点 C” を見出し, BC ≡ B’C” とする. 仮定, 定義および HaⅢ-2 より, B’A’ ≡ BA ∧ BA ≡ B’A” となって, B’A’ ≡ B’A”. 一直線 a’ 上の三点 B’, A’, A” において, 仮定および定義から, B’, A’ および B’, A” がそれぞれ相異なる点であり, 補題 8 より, ¬ Bet (A’, A”) B’ となるため, 補助命題 18 より, A’, A” は同一の点である. 補題 8 より, Bet (A”, C”) B’ であり, A’, A” は同一の点であるため, Bet (A’, C”) B’ となる. すなわち補助命題 19 より, 二線分 A’B’, B’C” は共通点をもたない. 仮定より Bet (A, C) B でもあるため, 同様に補助命題 19 より, 二線分 AB, BC も共通点をもたない. さらに仮定, 定義および HaⅢ-2 より, AB ≡ A’B’ かつ BC ≡ B’C” となるため, HaⅢ-3 より, AC ≡ A’C” となる. ここで仮定より B’, C’ が a’ において A’ の同じ側にあるため, 補題 8 より ¬ Bet (B’, C’) A’ であり, また Bet (A’, C”) B’ から定義より, Bet (C”, A’) B’ である. もし, Bet (C”, C’) A’ ならば, A’, B’, C’, C”

$\Longrightarrow$ R, Q, S, P と書き換えると, Bet (P, R) Q $\wedge$ Bet (P, S) R となる. よって補題 4' より, Bet (Q, S) R. 書き換えを直せば, Bet (B', C') A' となって, $\neg$ Bet (B', C') A' に矛盾する. したがって, $\neg$ Bet (C", C') A' である. 一直線 a' 上の三点 A', C", C' において, 仮定および定義から, A', C" および A', C' がそれぞれ相異なる点であり, $\neg$ Bet (C", C') A' なので, 補助命題 18 より, C", C' は同一の点である. したがって, Bet (A', C") B' から, Bet (A', C') B' となる. これで命題の求めるところが証明された.

補助命題 21　A, B, C を Bet (A, C) B であるような一直線 a 上の三点とし, さらに A' を同じ直線または他の直線 a' 上の点とするとき, 直線 a' 上の A' に対して与えられた側に, それぞれ AB $\equiv$ A'B', AC $\equiv$ A'C' となるような二点 B', C' をとれる. またこのとき, B', C' が a' において A' の同じ側にあるならば, Bet (A', C') B' となる.

証明　HaI-3 より, a' 上に A' と相異なる点 D をとる. 二点 A, B が a 上の二点であることから, HaⅢ-1 より, a' 上の A' に対して D と同じ側に点 B' を見出し, AB $\equiv$ A'B' とし, 同様に二点 A, C が a 上の二点であることから, HaⅢ-1 より, a' 上の A' に対して B と同じ側に点 C' を見出し, AC $\equiv$ A'C' とする. B' を D と異なる側に見出し, 同様に C' を見出せば, このような二点 B', C' を a' 上の A' に対して与えられた側にとれる. このとき, Bet (A', C') B' であると主張する.

A, B, C が Bet (A, C) B であるような一直線 a 上の三点であり, また直線 a' 上に三点 A', B', C' が存在して, 定義より A', B' および A', C' がそれぞれ相異なる点であり, AB $\equiv$ A'B', AC $\equiv$ A'C' かつ B', C' が a' において A' の同じ側にあるため, 補助命題 20 より Bet (A', C') B' である. これで命題の求めるところが証明された.

これは二線分を同じ側へ合同に移すことで, 三点の間の関係が維持されることの保証である. 問題となる命題の証明に移る.

補助命題 22　二直線 a, a' 上にそれぞれ三点 A, B, C および A', B', C' が存在し, 以下の条件が与えられたとする. このとき, 少なくとも一つの条件が誤りである.

(1) Bet (A, C) B.

(2) Bet (A', B') C'.

(3) AB $\equiv$ A'B'.

(4) AC $\equiv$ A'C'.

証明　各条件および定義より, A, B, C が Bet (A, C) B であるような一直線上の三点であり, さらに A', B' が同じ直線または他の直線 a' 上の点であることから, 補助命題 21 より, a' 上の A' に対して B' と同じ側に, それぞれ AB $\equiv$ A'B", AC $\equiv$ A'C" となるような二点 B", C" をとる. B", C" が a' において A' の同じ側にあるため, Bet (A', C") B" である. 条件, 定義および HaⅢ-2 より, A'B' $\equiv$ AB $\wedge$ AB $\equiv$ A'B" となって, A'B' $\equiv$ A'B". 一直線 a' 上の三点 A', B', B" において, 定義より A', B' および A', B" がそれぞれ相異なる点であり, 補題 8 より, $\neg$ Bet (B', B") A' であって, A'B' $\equiv$ A'B" とあるため, 補助命題 18 より, B', B" は同一の点である. したがって, Bet (A', C") B" から, Bet (A', C") B' となる. 条件, 定義および HaⅢ-2 より, A'C' $\equiv$ AC $\wedge$ AC $\equiv$ A'C" となって, A'C' $\equiv$ A'C". ここで補題 8 より, $\neg$ Bet (B', C") A' であり, 条件および定義より, Bet (B', A') C' である. もし, Bet (C', C") A' ならば, A', B', C', C" $\Longrightarrow$ R, P, Q, S と書き換えると, Bet (P, R) Q $\wedge$ Bet (Q, S) R となる. よって補題 3' より, Bet (P, S) R. 書き換えを直せば, Bet (B', C") A' となって, $\neg$ Bet (B', C") A' に矛盾する. したがって, $\neg$ Bet (C', C") A' である. 一直線上の三点 A', C', C" において, 定義より A', C' および A', C" がそれぞれ相異なる点であり, $\neg$ Bet (C', C") A' かつ A'C' $\equiv$ A'C" であるため, 補助命題 18 より, C', C" は同一の点である. したがって, Bet (A', C") B" から, Bet (A', C') B' となる. ところが, 条件および HaⅡ-3 より, $\neg$ Bet (B', C') A' $\wedge$ $\neg$ Bet (A', C') B' であるため, これに矛盾する. よってこれらの条件の少なくとも一つは誤りである. これで命題の求めるところが証明された.

補助命題 22 により問題 5 は解決される. また, 先述した HT-15 の証明過程においては, 補助命題 22 の条件 (1), (3), (4) が証明済みであるため, 残る条件 (2) にあたる Bet (L', H') K' の否定が達成される.

3.3.5 角の大小関係

角の大小については, 以下の定理の元で定義されている.

Ht-20 任意の二角 ∠(h, k) および ∠(h', l') が与えられたとき, ∠(h, k) を h' を一辺として l' の在る側に作るとき k' がその内部にある半直線となれば, ∠(h', l') を h を一辺として k の在る側に作れば, l はその外部にある半直線となる. 逆もまた然りである.

図 22: Ht-20

定義 この定理の示す如く ∠(h, k) を作り ∠(h', l') の内部にある半直線 k' を得るとき, ∠(h, k) は ∠(h', l') よりも小であるといい, 記号で ∠(h, k) < ∠(h', l') と表す. また外部にある k' を得るとき, ∠(h, k) は ∠(h', l') よりも大であるといい, 記号で ∠(h, k) > ∠(h', l') と表す.

これらおよび補助命題 10 より, 以下のような補題が直ちに導かれる.

補題 9 任意の二角 ∠AOB および ∠C に対し, ∠DOB ≡ ∠C かつ Sameside : OB (D, A) となるような点 D をとる. このとき, D が ∠AOB の内部の点であるならばまた, ∠C < ∠AOB であり, D が ∠AOB の外部の点であるならば, ∠C > ∠AOB である.

また, 角の合同の推移性については, 以下の定理で証明されている.

Ht-19 二つの角 ∠(h', k') および ∠(h'', k'') が第三の角 ∠(h, k) に合同ならば, ∠(h', k') はまた ∠(h'', k'') に合同である.

ここで [9] では, 以下の二つの問題が分かると述べられている.

問題 6 二つの角 α および β に対して次の三つの場合

$\alpha < \beta \wedge \beta > \alpha$.

$\alpha \equiv \beta$.

$\alpha > \beta \wedge \beta < \alpha$.

のうちの唯一つが常に生起する.

問題 7 角の大小関係は推移的である, 即ち

1. $\alpha > \beta \wedge \beta > \gamma$.
2. $\alpha > \beta \wedge \beta \equiv \gamma$.
3. $\alpha \equiv \beta \wedge \beta > \gamma$.

なる三つの仮定のそれぞれから $\alpha > \gamma$ が導かれる.

これらの問題について, 定義から明らかである, あるいは特別な困難なしに辿り着くと主張するのは難しいと考える.

問題の解決のために, まずは以下の命題を証明する.

補題 10　角 ∠AOB および点 C が与えられ, Sameside : OB (C, A) であるとき, C が ∠AOB の外部の点でないならば, C が ∠AOB の内部の点であるか, もしくは二直線 OA, OC が同一の直線である.
補題 11　二角 ∠AOB, ∠COB が与えられ, Sameside : OB (C, A) であるとき, 二直線 OA, OC が同一の直線であるならばまた, ∠AOB ≡ ∠COB.

補題 10 は角の定義, 角の内部に関する定義および平面と直線の側に関する定義より直ちに導かれる. 以下で補題 11 の証明を示す.

証明　仮定および定義より, 二点 C, A はそれぞれ直線 OB の上にない. もし Bet (A, C) O ならば, OB の上にない二点 C, A について, それらの定める線分 CA 上に OB の点 O が存在するため, 定義より, Diffside : OB (C, A) となる. すると補助命題 2 より, ¬ Sameside : OB (C, A) となって, 仮定に矛盾する. よって, ¬ Bet (A, C) O である. すなわち, 二直線 OA, OC が同一の直線であるならば, 定義および補題 1 より, C は半直線 OA 上にある. よって定義より ∠AOB ≡ ∠COB. 一方で, ∠AOB ≡ ∠COB であって, OA, OC が相異なる直線であるとすれば, これは直線に対して与えられた側に任意の角を唯一通りに合同に移せることに矛盾する. よって, 二角 ∠AOB, ∠COB が与えられ, Sameside : OB (C, A) であるとき, ∠AOB ≡ ∠COB ならば, OA, OC は同一の直線である. したがって, 補題の求めるところが証明された.

補助命題 23　二角 ∠AOB, ∠C が与えられたとき, ¬ ∠C > ∠AOB ならば, ∠AOB ≡ ∠C ∨ ∠C < ∠AOB.
証明　HaⅢ-1, HaⅢ-4 より, ∠DOB ≡ ∠C かつ Sameside : OB (D, A) となるような点 D をとる. もし D が ∠AOB の外部の点であれば, 補題 9 より, ∠C > ∠AOB となって仮定に矛盾する. したがって補題 10 より, D が ∠AOB の内部の点であるか, もしくは二直線 OA, OD が同一の直線である. D が ∠AOB の内部の点であれば定義より, ∠C < ∠AOB であり, また OA, OD が同一の直線であれば補題 11, Ht-19 および HaⅢ-4 より, ∠AOB ≡ ∠C. したがって, 命題の求めるところが証明された.

補助命題 24　二角 ∠AOB, ∠C が与えられたとき, ¬ ∠C ≡ ∠AOB ならば, ∠AOB < ∠C ∨ ∠C < ∠AOB.
証明　HaⅢ-1, HaⅢ-4 より, ∠DOB ≡ ∠C かつ Sameside : OB (D, A) となるような点 D をとる. もし二直線 OA, OD が同一の直線ならば, 仮定および補題 11 より, ∠AOB ≡ ∠DOB. このとき仮定および Ht-19 より, ∠C ≡ ∠AOB となって仮定に矛盾する. したがって, OA, OD は相異なる直線である. また仮定および定義より, OA と OB, OD と OB はそれぞれ相異なる直線である. すなわち, 三半直線 OA, OB, OD は点 O から出る相異なる直線に属する半直線であり, 半直線 OB は直線 OB に属し, 二点 D, A が OB に対して同じ側にあるため, 補助命題 13 より, D が ∠AOB の内部にあるか, あるいは A が ∠DOB の内部にある. A が ∠DOB の内部にあるならば, 仮定および補題 9 より, ∠AOB < ∠C であり, また D が ∠AOB の内部にあるならば, 仮定および補題 9 より, ∠C < ∠AOB である. したがって, 命題の求めるところが証明された.

補助命題 25　二角 ∠AOB, ∠C が与えられたとき, ∠C ≡ ∠AOB ならば, ¬ ∠C < ∠AOB ∧ ¬ ∠AOB < ∠C.
証明　HaⅢ-1, HaⅢ-4 より, ∠DOB ≡ ∠C かつ Sameside : OB (D, A) となるような点 D をとる. このとき Ht-19 より ∠DOB ≡ ∠AOB となって, 仮定および補題 11 より, 二直線 OA, OD は同一の直線である. すると, OD の上に D が存在するため, 補題 1 より OA の上にも D が存在する. もし ∠C < ∠AOB ならば, 仮定および補題 9 より, D が ∠AOB の内部の点である. すなわち補題 6 より, Sameside : OA (D, B) ∧ Sameside : OB (D, A) となって, 定義より, D が OA の上にない点となり, これに矛盾する. したがって, ¬ ∠C < ∠AOB となる. また仮定および HaⅢ-4 より, ∠C ≡ ∠AOB でもあるため同様にして, ¬ ∠AOB < ∠C を導ける. これで, 命題の求めるところが証明された.

補助命題 26　二角 ∠AOB, ∠A'O'B' が与えられたとき, ∠AOB < ∠A'O'B' ならば, ¬ ∠AOB ≡ ∠A'O'B' ∧ ¬ ∠A'O'B' < ∠AOB.
証明　HaⅢ-1, HaⅢ-4 より, ∠COB ≡ ∠A'O'B' かつ Sameside : OB (C, A) となるような点 C をとり, 同様に, ∠C'O'B' ≡ ∠AOB かつ Sameside : O'B' (C', A') となるような点 C' をとる. 仮定および補題 9 より, C' は ∠A'O'B' の内部の点である. すなわち補助命題 10 より, 半直線 O'C' が ∠A'O'B' の内部にある. 与えられた二角 ∠AOB, ∠A'O'B' に対して, ∠AOB を O'B' を一辺として A' の在る

側に作ったとき, O'C' がその内部にある半直線となっており, ∠A'O'B' を OB を一辺として A の在る側に作っているため, Ht-20 より, OC はその外部にある半直線である. すなわち定義より, C は ∠AOB の内部の点ではなく, また半直線 OA 上にない点である. もし ∠A'O'B' < ∠AOB ならば, 仮定および補題 9 より, C が ∠AOB の内部の点となって矛盾する. よって, ¬ ∠A'O'B' < ∠AOB. また, もし ∠AOB ≡ ∠A'O'B' であれば, Ht-19 より, ∠AOB ≡ ∠COB となって, 仮定および補題 11 より, 二直線 OA, OC が同一の直線となる. すると直線 OC の上に点 C が存在するため, 補題 1 より直線 OA の上にも C が存在する. このとき, C は半直線 OA 上にない点であるため, 残る場合は Bet (A, C) O である. Sameside : OB (C, A) から, 定義より, 二点 C, A は直線 OB 上にない. もし Bet (A, C) O ならば, OB 上にない二点 C, A について, それらの定める線分 CA 上に OB の点 O が存在するため, 定義より, Diffside : OB (C, A) となる. すると補助命題 2 より, ¬ Sameside : OB (C, A) となって, Sameside : OB (C, A) に矛盾する. よって, ¬ ∠AOB ≡ ∠A'O'B'. したがって, 命題の求めるところが証明された.

補題 12　二角 α, β が与えられたとき, $\alpha < \beta$ ならば, $\beta > \alpha$.
証明　もし, $\neg\, \beta > \alpha$ ならば, 補助命題 23 より, $\alpha \equiv \beta \vee \beta < \alpha$. このとき, $\alpha \equiv \beta$ ならば, 補助命題 25 より, $\neg\, \alpha < \beta \wedge \neg\, \beta < \alpha$ となって, 仮定に矛盾する. また, $\beta < \alpha$ ならば, 補助命題 26 より, $\neg\, \beta \equiv \alpha \wedge \neg\, \alpha < \beta$ となって, 仮定に矛盾する. すなわち, $\beta > \alpha$ となり, 命題の求めるところが証明される.

補助命題 27　二角 α, β が与えられたとき, 次の三つの場合

$$\alpha < \beta \wedge \beta > \alpha.$$
$$\alpha \equiv \beta.$$
$$\alpha > \beta \wedge \beta < \alpha.$$

のうちの唯一つが常に生起する.
証明　$\alpha < \beta \wedge \beta > \alpha$ である場合, 補助命題 26 より, $\neg\, \alpha \equiv \beta \wedge \neg\, \beta < \alpha$. すなわち, $\alpha < \beta \wedge \beta > \alpha \Longrightarrow \neg\, \alpha \equiv \beta \wedge \neg\, (\alpha > \beta \wedge \beta < \alpha)$ となる. $\alpha > \beta \wedge \beta < \alpha$ である場合も同様に, 補助命題 26 より, $\neg\, \beta \equiv \alpha \wedge \neg\, \alpha < \beta$. よって, HaⅢ-4 より, $\alpha > \beta \wedge \beta < \alpha \Longrightarrow \neg\, \alpha \equiv \beta \wedge \neg\, (\alpha < \beta \wedge \beta > \alpha)$. 次に, $\alpha \equiv \beta$ である場合, 補助命題 25 より, $\neg\, \alpha < \beta \wedge \neg\, \beta < \alpha$. よって, $\alpha \equiv \beta \Longrightarrow \neg\, (\alpha < \beta \wedge \beta > \alpha) \wedge \neg\, (\alpha > \beta \wedge \beta < \alpha)$. したがって, 三つの場合のうち二つ以上が同時に生起することはない.

最後に, 全ての場合が生起しない可能性を考える. すると, $\neg\, \alpha \equiv \beta$ から補助命題 24 より, $\beta < \alpha \vee \alpha < \beta$. このとき, $\beta < \alpha$ ならば, 補題 12 より, $\alpha > \beta$ であるため, $\alpha > \beta \wedge \beta < \alpha$ が生起される. また, $\alpha < \beta$ であるときも同様にして, $\alpha < \beta \wedge \beta > \alpha$ が生起される. よって, 全ての場合が生起しないことはない. したがって, 三つの場合のうちの唯一つが常に生起する. これで命題の求めるところが証明された.

補助命題 28　三つの角 ∠A, ∠BOC および ∠B'O'C' が与えられたとき, ∠A ≡ ∠BOC ∧ ∠BOC > ∠B'O'C' ならば, ∠A > ∠B'O'C'.
証明　HaⅢ-1, HaⅢ-4 より, ∠DOC ≡ ∠B'O'C' かつ Sameside : OC (D, B) となるような点 D をとる. 仮定および補助命題 27 より, ∠B'O'C' < ∠BOC. よって, 補題 9 より, D は ∠BOC の内部の点である. すなわち, 補助命題 10 より, 半直線 OD が ∠BOC の内部にある. 次に, HaⅢ-1, HaⅢ-4 より, ∠D'O'C' ≡ ∠BOC かつ Sameside : O'C' (D', B') となるような点 D' をとる. 与えられた二角 ∠BOC, ∠B'O'C' に対して, ∠BOC を O'C' を一辺として B' の在る側に作ったとき, OD がその内部にある半直線となっており, ∠B'O'C' を OC を一辺として B の在る側に作っているため, Ht-20 より, O'D' はその外部にある半直線である. すなわち, 定義より, D' は ∠B'O'C' の外部の点である. ∠D'O'C' ≡ ∠BOC から仮定, Ht-19 および HaⅢ-4 より, ∠D'O'C' ≡ ∠A である. 二角 ∠A, ∠B'O'C' に対し, ∠D'O'C' ≡ ∠A かつ Sameside : O'C' (D', B') となるような点 D' をとり, D' が ∠B'O'C' の外部の点であるため, 補題 9 より, ∠A > ∠B'O'C' である. したがって, 命題の求めるところが証明された.

補助命題 29　三つの角 ∠AOB, ∠C および ∠D が与えられたとき, ∠AOB > ∠C ∧ ∠C ≡ ∠D ならば, ∠AOB > ∠D.

証明　HaⅢ-1, HaⅢ-4 より, ∠EOB ≡ ∠C かつ Sameside : OB (E, A) となるような点 E をとる. 仮定および補助命題 27 より, ∠C < ∠AOB. よって, 補題 9 より, E は ∠AOB の内部の点である. ∠EOB ≡ ∠C から仮定, Ht-19 および HaⅢ-4 より, ∠EOB ≡ ∠D である. 二角 ∠D, ∠AOB に対し, ∠EOB ≡ ∠D かつ Sameside : OB (E, A) となるような点 E をとり, E が ∠AOB の内部の点であるため, 補題 9 および補題 12 より, ∠AOB > ∠D である. したがって, 命題の求めるところが証明された.

補題 13　相異なる二直線 a, b が与えられたとき, それぞれの上に二点 A, B が存在するならば, この二点は同一の点である.
証明　もし A, B が相異なる点であれば, 補題 1 より a, b が同一の直線となって仮定に矛盾する. よってこれらは同一の点である. したがって, 補題の求めるところが証明された.

補助命題 30　五つの点 A, B, C, D, E について, 以下の条件が与えられたとする. このとき, 少なくとも一つの条件が誤りである.

1. Diffside : AB (C, D).
2. Diffside : AC (B, D).
3. Sameside : AE (C, B).
4. Sameside : AE (D, B).

証明　条件 1 および定義より Bet(C, D) F となる点 F が存在し, AB が F を通る. また条件 2 からも同様に Bet(B, D) G となる点 G が存在し, AC が G を通る.

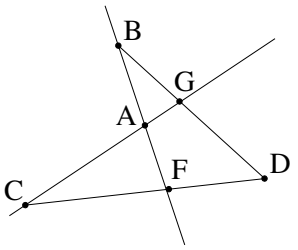

図 23: 補助命題 30 (条件 1, 2 を満たす例)

Bet(B, D) G から HaⅡ-1 より, 三点 B, D, G は相異なる点であり, 直線 BD は G を通る. もし直線 CD が G を通れば, 二直線 BD, CD の上にそれぞれ相異なる二点 D, G が存在するため, 補題 1 よりこれらが同一の直線となる. またこのとき BD の上に B が存在するため, 補題 1 より CD の上にも B が存在する. Diffside : AB (C, D) から定義より, C, D は直線 AB 上にない. ここでもし二直線 CD, AB が同一の直線であれば, CD の上に D が存在するため, 補題 1 より AB の上にも D が存在することとなり, これに矛盾する. よって, CD, AB は相異なる直線である. Bet(C, D) F から HaⅡ-1 より, 三点 C, D, F は相異なる点であり, 直線 CD は F を通る. 相異なる二直線 CD, AB の上にそれぞれ二点 B, F が存在するため, 補題 13 よりこれらは同一の点である. すなわち Bet(C, D) F より, Bet(C, D) B となり, HaⅡ-1 より, 直線 CD が三点 C, D, B を通る. Diffside : AC (B, D) から定義より, B, D は直線 AC 上にない. ここでもし二直線 CD, AC が同一の直線であれば, CD の上に D が存在するため, 補題 1 より AC の上にも D が存在することとなり, これに矛盾する. よって, CD, AC は相異なる直線である. Bet(C, D) B であり, C を含む直線 AC と三点 C, D, B を含む直線 CD が相異なる直線であるため, 補助命題 5 より, Sameside : AC (B, D) となる. このとき補助命題 1 より, ¬ Diffside : AC (B, D) となって, 条件 2 に矛盾する. よって, CD は G を通らない. すなわち C, D, G は一直線上にない三点である. もし AB が G を通れば, 二直線 BD, AB の上にそれぞれ相異なる二点 B, G が存在するため, 補題 1 よりこれらが同一の直線となる. またこのとき BD の上に D が存在するため, 補題 1 より AB の上にも D が存在することとなり, D が直線 AB 上にないことに矛盾する. よって, AB は G を通らない. C, D, G が一直線上にない三点であり, AB がそのいずれも通らない直線であって, 線分 CD の点 F を通るため, HaⅡ-4 より, 直線 AB は線分 CG もしくは線分 DG の点を

通る. AB が線分 DG の点を通る場合を考える. すなわち, Bet(D, G) H となる点 H が存在し, AB が H を通る. Bet(B, D) G から HaⅡ-1 および補題 2 より, ¬ Bet(D, G) B である. もし二点 B, H が同一の点であれば, Bet(D, G) H より Bet(D, G) B となって, これに矛盾する. よって, B, H は相異なる点である. ここで, Bet(D, G) H から HaⅡ-1 より, 直線 DG は H を通る. また, Bet(B, D) G から HaⅡ-1 より, 直線 DG は B を通る. すると, 二直線 DG, AB の上にそれぞれ相異なる二点 B, H が存在するため, 補題 1 よりこれらが同一の直線となる. またこのとき DG の上に D が存在するため, 補題 1 より AB の上にも D が存在することとなり, D が直線 AB 上にないことに矛盾する. よって, AB は線分 DG の点を通らない. したがって, AB は線分 CG の点を通る. すなわち, Bet(C, G) I となる点 I が存在し, AB が I を通る. Bet(C, G) I から HaⅡ-1 より, 三点 C, G, I は相異なる点であり, 直線 CG は I を通る. 二直線 AC, CG の上にそれぞれ相異なる二点 C, G が存在するため, 補題 1 よりこれらは同一の直線である. よって AC の上に A が存在するため, 補題 1 より, CG の上にも A が存在する. もし二直線 CG, AB が同一の直線であれば, CG の上に C が存在するため, 補題 1 より AB の上にも C が存在することとなり, C が AB 上にないことに矛盾する. よって, CG, AB は相異なる直線である. 相異なる二直線 CG, AB の上にそれぞれ二点 A, I が存在するため, 補題 13 よりこれらは同一の点である. すなわち Bet(C, G) I より, Bet(C, G) A となる. もし二点 A, B が同一の点であれば, AC の上に B が存在することとなり, B が AC 上にないことに矛盾する. よって, A, B は相異なる点である. もし CG の上に B があれば, 二直線 CG, AB の上に相異なる二点 A, B が存在するため, 補題 1 よりこれらが同一の直線となり, CG, AB は相異なる直線であることに矛盾する. よって, CG の上に B はない. すなわち, C, G, B は一直線上にない三点である. Bet(C, G) A から HaⅡ-1 より, 二点 G, A は相異なる点である. 条件 3 および定義より, 直線 AE は線分 CB の点を含まず, 二点 B, C は直線 AE の上にない. ここでもし AE の上に G があれば, 二直線 CG, AE の上にそれぞれ相異なる二点 G, A が存在するため, 補題 1 よりこれらが同一の直線となる. すると, CG の上に C が存在するため, 補題 1 より AE の上にも C が存在することとなり, これに矛盾する. よって, AE の上に G はない. C, G, B が一直線上にない三点であり, AE がそのいずれも通らない直線であって, 線分 CG の点 A を通るため, HaⅡ-4 より, 直線 AE は線分 CB もしくは線分 GB の点を通る. ここで, AE は線分 CB の点を含まないため, 線分 GB の点を通る. すなわち, Bet(G, B) J となる点 J が存在し, AE が J を通る. Bet(G, B) J から HaⅡ-1 より, Bet(B, G) J である. すなわち, Bet(B, G) J ∧ Bet(B, D) G である. ここで, B, J, G, D = P, Q, R, S と書き換えれば, Bet(P, R) Q ∧ Bet(P, S) R となる. よって補題 4' より, Bet(P, S) Q ∧ Bet(Q, S) R. 書き換えを直せば, Bet(B, D) J ∧ Bet(J, D) G となる. Bet(B, D) J から HaⅡ-1 より, Bet(D, B) J であり, 直線 AE が J を通るため, AE が線分 DB の点を通ることとなる. ところが条件 4 および定義より, 直線 AE は線分 DB の点を含まない. よってこれらの条件の少なくとも一つは誤りである. これで命題の求めるところが証明された.

補助命題 31　直線 a および三点 A, B, C について, 二点 A, C が相異なる点であり, Sameside : a (A, B) ∧ Sameside : a (B, C) ならば, Sameside : a (A, C) である.
証明　もし, Diffside : a (A, C) ならば, 仮定, 補助命題 8 および補助命題 2 より, ¬ Sameside : a (B, C) となって, 仮定に矛盾する. よって, ¬ Diffside : a (A, C) である. Sameside : a (A, B) から定義より, A は a 上の点ではなく, また Sameside : a (B, C) から同様に, C も a 上の点ではない. 相異なる二点 A, C について, ¬ Diffside : a (A, C) かつ, それぞれが a 上の点でないため, 補助命題 4 より, Sameside : a (A, C) である. これで命題の求めるところが証明された.

補助命題 32　三つの角 ∠AOB, ∠C および ∠D が与えられたとき, ∠AOB > ∠C ∧ ∠C > ∠D ならば, ∠AOB > ∠D.
証明　HaⅢ-1, HaⅢ-4 より, ∠EOB ≡ ∠C かつ Sameside : OB (E, A) となるような点 E をとる. また同様にして, ∠FOB ≡ ∠D かつ Sameside : OB (F, A) となるような点 F をとる. 仮定および補助命題 27 より, ∠C < ∠AOB. よって, 補題 9 より, E は ∠AOB の内部の点である. ∠EOB ≡ ∠C ∧ ∠C > ∠D となるため, 補助命題 27, 28 より, ∠D < ∠EOB である. よって, 補助命題 26 より, ¬ ∠D ≡ ∠EOB ∧ ¬ ∠EOB < ∠D. ここでもし, 二点 E, F が同一の点であれば, 定義および HaⅢ-4 より, ∠EOB ≡ ∠FOB となる. このとき, ∠FOB ≡ ∠D なので, HaⅢ-4 および Ht-19 より, ∠D ≡ ∠EOB となってこれに矛盾する. よって, E, F は相異なる点である. Sameside : OB (E, A) から定義より, E, A は相異なる点であり, また Sameside : OB (F, A) から同様に, F, A も相異なる点である. すなわち, 三点 F, A, E は相異なる点である. Sameside : OB (E, A) から補題 5 より, Sameside : OB (A, E) である. 直線 OB および相異なる三点 F, A, E について, Sameside : OB (F, A) ∧ Sameside : OB (A, E) である

ため, 補助命題 31 より, Sameside : OB (F, E) である. 任意の二角 ∠EOB および ∠D に対し, ∠FOB ≡ ∠D かつ Sameside : OB (F, E) となるような点 F をとっているため, 補題 9 より F が ∠EOB の内部の点であるならばまた, ∠D < ∠EOB である. ここで, ∠D < ∠EOB であるため, F は ∠EOB の内部の点である. このとき補題 6 より, Sameside : OE (F, B) ∧ Sameside : OB (F, E) である. また, E が ∠AOB の内部の点であるため, 補題 6 および補助命題 9 より, Sameside : OA (E, B) ∧ Sameside : OB (E, A) ∧ Diffside : OE (A, B) である. Sameside : OE (F, B) ∧ Diffside : OE (A, B) から補題 5 より, Sameside : OE (B, F) ∧ Diffside : OE (B, A) であり, 定義より二点 B, F および二点 B, A はそれぞれ相異なる点である. F, A も相異なる点であるため, 三点 B, F, A は相異なる点である. 直線 OE および相異なる三点 B, F, A について, Sameside : OE (B, F) かつ Diffside : OE (B, A) であるため, 補助命題 8 および補題 5 より, Diffside : OE (A, F) である. Sameside : OA (E, B) から補題 5 より, Sameside : OA (B, E) であり, 定義より, B, E は相異なる点である. B, F および E, F が相異なる点であるため, 三点 B, E, F は相異なる点である. もし, Diffside : OA (B, F) ならば, 直線 OA および相異なる三点 B, E, F について, Sameside : OA (B, E) かつ Diffside : OA (B, F) であるため, 補助命題 8 より, Diffside : OA (E, F) となる. このとき, 五つの点 O, E, A, F, B について四つの条件, Diffside : OE (A, F), Diffside : OA (E, F), Sameside : OB (A, E), Sameside : OB (F, A) が与えられるため, 補助命題 30 より少なくとも一つの条件が誤りである. よって, ¬ Diffside : OA (B, F) である. Sameside : OA (E, B) から定義より, OA の上に B はない. Sameside : OB (F, A) から定義より, OB の上に F, A はない. ∠C < ∠AOB から補助命題 26 より, ¬ ∠C ≡ ∠AOB ∧ ¬ ∠AOB < ∠C である. もし, Bet(A, F) O であれば, 直線 OB およびその上にない二点 A, F に対し, それらの定める線分 AF の上に OB の点 O が存在するため, 定義より Diffside : OB (F, A) となり, 補助命題 2 より, ¬ Sameside : OB (F, A) となって, Sameside : OB (F, A) であることに矛盾する. よって, ¬ Bet(A, F) O. すなわちもし, OA の上に F があるならば, A, F はいずれも O より出でる同一の半直線上にある. したがって, 定義および HaⅢ-4 より, ∠FOB ≡ ∠AOB となる. すると, ∠FOB ≡ ∠AOB ∧ ∠FOB ≡ ∠D となるため, HaⅢ-4 および Ht-19 より, ∠D ≡ ∠AOB. よって仮定より, ∠C > ∠D ∧ ∠D ≡ ∠AOB となるため, 補助命題 27, 29 より, ∠AOB < ∠C となって, ¬ ∠AOB < ∠C に矛盾する. よって, OA の上に F はない. 直線 OA および相異なる二点 B, F について, ¬ Diffside : OA (B, F) かつ, それぞれが OA 上の点でないため, 補助命題 4 および補題 5 より, Sameside : OA (F, B) である. すなわち, Sameside : OA (B, F) ∧ Sameside : OB (F, A) であるため, 補題 6 より, F が ∠AOB の内部にある. 任意の二角 ∠AOB および ∠D に対し, ∠FOB ≡ ∠D かつ Sameside : OB (F, A) となるような点 F をとっているため, 補題 9 より F が ∠AOB の内部の点であるならばまた, ∠D < ∠AOB である. ここで, F が ∠AOB の内部にあるため, ∠D < ∠AOB であり, 補助命題 27 より, ∠AOB > ∠D である. したがって, 命題の求めるところが証明された.

問題 6 は補助命題 27 に内包されており, 問題 7 は補助命題 28, 29 および 33 によって解決される.

3.3.6 線分の大小定義の回避

[9] に於いて, 線分の大小関係については, 角の大小関係に対応する関係が公理および定義より直ちに導かれると主張されている. 一方で, 線分の大小それ自体を定義として明文化している箇所は存在しない. しかしながら, 次のような定理が存在している.

Ht-23 任意の三角形に於いて大なる辺に対する内角は小なる辺に対する内角よりも大である.

さらにはこの定理を用いて次の定理の証明が行なわれている.

Ht-24 二つの内角が等しい三角形は二等辺である.

本稿ではこの問題の解決のために, Ht-23 を用いない Ht-24 の証明を目指す.
まずは問題の解決に必要となる定義および定理を示す.

Ht-11 合同な二辺を有する一つの三角形においてその辺に対する角は合同である. 換言すれば二等辺三角形において両底角は相等しい.

定義　二つの角が頂点と一辺とを共有し, 他の共通ならざる辺が一直線をなすとき, 補角という.

定義　三角形 ABC に属する角 ∠ABC, ∠BCA および ∠CAB をその三角形の内角, それの補角を三角形の外角という.

Ht-22　三角形の外角はこれに接せざる内角の何れよりも大である.

次に以下の命題を証明する.

補題 14　△ABC ならば, 三点 A, B, C は ∠ABC をつくる.
証明　補助命題 14 より, 点 C は直線 AB 上にない点である. もし二直線 AC, AB が同一の直線ならば, 補題 1 より AC の点 C が AB 上にも存在することとなり, これに矛盾する. よって AC, AB は相異なる直線である. また定義より, A, B および A, C はそれぞれ相異なる点である. すなわち半直線 AB, AC はそれぞれ一点 A より出でて相異なる直線に属する半直線である. したがって, 補題の求めるところが証明された.

補助命題 33　C が ∠AOB の内部の点であるとき, ∠AOB > ∠COB.
証明　二角 ∠AOB および ∠COB に対し, ∠DOB ≡ ∠COB かつ Sameside : OB (D, A) となるような点 D をとる. 定義および補題 5 より, Sameside : OB (A, C) である. 二点 D, C が異なる点であれば, Sameside : OB (D, A) ∧ Sameside : OB (A, C) から補助命題 31 より, Sameside : OB (D, C) となる. ∠COB ≡ ∠DOB であるため, もし二直線 OD, OC が相異なる直線であれば, 直線 OB に対して与えられた側に任意の角を唯一通りに合同に移せることに矛盾する. よって, 二直線 OD, OC は同一の直線である. このとき, もし Bet (D, C) B ならば, 線分 DC の上に直線 OB の点 B が存在するため, 定義より, Diffside : OB (D, C) となる. すると補助命題 2 より, ¬ Sameside : OB (D, C) となって, Sameside : OB (D, C) であることに矛盾する. よって, ¬ Bet (D, C) B である. またこのとき, 三点 O, C, D が相異なる点となる. ∠AOB の内部に点 C があり, 直線 OC 上の二点 O, C と相異なる点 D があって, ¬ Bet (D, C) B であるため, 補助命題 10 より, D は ∠AOB の内部の点である. 二点 D C が同一の点である場合も, D は ∠AOB の内部の点である. すなわち補題 9 より, ∠COB < ∠AOB となり, 補題 12 より, ∠AOB > ∠COB である. したがって, 命題の求めるところが証明された.

補助命題 34　△ABC について, Bet (B, C) D ∧ AC ≡ CD ならば, ∠BAC > ∠ABC.

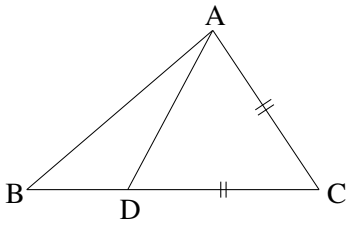

図 24: 補助命題 34

証明　補助命題 16 より △BAC であり, また補題 14 より, 三点 B, A, C は ∠BAC をつくる. ∠BAC に対して, Bet (B, C) D であるため, 補助命題 11 より, D は ∠BAC の内部の点である. よって補助命題 33 より, ∠BAC > ∠DAC である. ここで Bet (B, C) D から定義より, D は直線 BC 上にある B, C と相異なる点である. △ABC において, その一辺 BC を含む直線上に, その直線上にある頂点 B, C と相異なる点 D がとられているため, 補助命題 17 より, D と B または C および三角形の残りの頂点 A は三角形をつくる. よって, △CAD および △DBA である. また仮定より, AC ≡ CD であるため, △CAD は二等辺三角形である. よって Ht-11 より, ∠DAC ≡ ∠CDA である. 頂点 D および辺 DA を共有し, 共通ならざる辺 DB, DC が一直線 BC をなすため, 定義より, ∠ADB と ∠ADC は補角である. また定義より, ∠CDA は △DBA の外角である. よって Ht-22 より, ∠CDA > ∠ABD である. また定義より, ∠ABD ≡ ∠ABC である. ∠DAC ≡ ∠CDA ∧ ∠CDA > ∠ABD であるため, 補助命題 28

より, ∠DAC > ∠ABD である. また, ∠CAD > ∠ABD ∧ ∠ABD ≡ ∠ABC であるため, 補助命題 29 より, ∠DAC > ∠ABC である. さらに, ∠BAC > ∠DAC ∧ ∠DAC > ∠ABC であるため, 補助命題 32 より, ∠BAC > ∠ABC である. したがって, 命題の求めるところが証明された.

補助命題 35　△ABC について, Bet (A, C) D ∧ AB ≡ AC ならば, ∠ADB > ∠ABD.

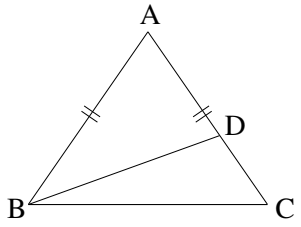

図 25: 補助命題 35

証明　仮定および Ht-11 より, ∠ABC ≡ ∠ACB である. Bet (A, C) D から定義より, D は直線 AC 上にある A, C と相異なる点である. △ABC において, その一辺 AC を含む直線上に, その直線上にある頂点 A, C と相異なる点 D がとられているため, 補助命題 17 より, D と A または C および三角形の残りの頂点 B は三角形をつくる. よって, △DCB および △DAB である. 頂点 D および辺 DB を共有し, 共通ならざる辺 DA, DC が一直線 AC をなすため, 定義より, ∠ADB と ∠BDC は補角である. また定義より, ∠ADB は △DCB の外角である. よって Ht-22 より, ∠ADB > ∠DCB である. ここで, ∠ABC ≡ ∠ACB ∧ ∠DCB ≡ ∠ACB であるため, HaⅢ-1 および Ht-19 より, ∠ABC ≡ ∠DCB である. すなわち定義および HaⅢ-2 より, ∠ADB > ∠DCB ∧ ∠DCB ≡ ∠ABC となるため, 補助命題 29 より, ∠ADB > ∠ABC である. 仮定および補助命題 16 より, △CBA であり, また補題 14 よりこの三点 C, B, A は ∠CBA をつくる. 仮定および HaⅡ-1 より, Bet (C, A) D である. ∠CBA に対して, Bet (C, A) D であるため, 補助命題 11 より D は ∠CBA の内部の点である. よって補助命題 33 より, ∠CBA > ∠DBA. また定義および HaⅢ-4 より, ∠DBA ≡ ∠ABD である. すると, ∠CBA > ∠DBA ∧ ∠DBA ≡ ∠ABD であるため, 補助命題 29 より, ∠CBA > ∠ABD である. さらに定義および HaⅢ-4 より, ∠ABC ≡ ∠CBA であり, ∠CBA > ∠ABD であるため, 補助命題 28 より, ∠ABC > ∠ABD である. すなわち, ∠ADB > ∠ABC ∧ ∠ABC > ∠ABD であるため, 補助命題 32 より, ∠ADB > ∠ABD である. したがって, 命題の求めるところが証明された.

問題となる定理の証明に移る.

Ht-24　二つの内角が等しい三角形は二等辺である.
証明　三角形を △ABC, 等しい二角を ∠ABC および ∠ACB とする. このとき補助命題 27 より, ¬ (∠ABC < ∠ACB ∧ ∠ACB > ∠ABC) および ¬ (∠ABC > ∠ACB ∧ ∠ACB < ∠ABC) である. HaⅢ-1 より, 直線 AC 上の点 A について C と同じ側に AB ≡ AD となるような点 D をつくる. ここで, ¬ AC ≡ AD である場合を考える. A, C, D は一直線 AC 上の三点であり, また定義より A, C および A, D はそれぞれ相異なる点である. また, C, D は直線 AC において A の同じ側にある二点であり, 補題 8 より, ¬ Bet (C, D) A である. もし C, D が同一の点であれば, 点 A から出る半直線上に AB と合同な二通りの線分を移せることとなり, 線分を合同に移すことの一意可能性に矛盾する. よって, C, D もまた相異なる点である. したがって, 三点 A, C, D は一直線 AC 上にある相異なる点である. このとき Ht-4 および補題 2 より, この三点の位置は以下の三つの場合に区別される.

(1) Bet (A, C) D ∧ ¬ Bet (C, D) A ∧ ¬ Bet (D, A) C.

(2) ¬ Bet (A, C) D ∧ Bet (C, D) A ∧ ¬ Bet (D, A) C.

(3) ¬ Bet (A, C) D ∧ ¬ Bet (C, D) A ∧ Bet (D, A) C.

¬ Bet (C, D) A が既に導かれているため, (2) の場合は有り得ない.

次に, (1) の場合について考える. △ABC および補助命題 16 より, △BCA である. また, HaⅡ-1 より, Bet (C, A) D である. さらに AB ≡ AD なので, 定義および HaⅢ-2 より, BA ≡ AD である. △BCA について, Bet (C, A) D ∧ BA ≡ AD であるため, 補助命題 34 より, ∠CBA > ∠BCA である. 定義および HaⅢ-4 より, ∠ABC ≡ ∠CBA である. すると, ∠ABC ≡ ∠CBA であり, ∠CBA > ∠BCA であるため, 補助命題 28 より, ∠ABC > ∠BCA である. また定義および HaⅢ-4 より, ∠BCA ≡ ∠ACB である. すると, ∠ABC > ∠BCA ∧ ∠BCA ≡ ∠ACB であるため, 補助命題 29 より, ∠ABC > ∠ACB である. これは, ¬ (∠ABC > ∠ACB ∧ ∠ACB < ∠ABC) に矛盾する.

最後に, (3) の場合について考える. HaⅡ-1 より, Bet (A, D) C である. △ABC および補助命題 14 より, 直線 AC 上に B はない. 一方で D は AC 上の点であるため, もし B, D が同一の点であれば, AC 上にも B が存在することとなり, これに矛盾する. よって B, D は相異なる点である. また, 二直線 AC, AD 上にそれぞれ相異なる二点 A, D が存在するため, 補題 1 より, AC, AD は同一の直線である. もし AD 上に B が存在すれば, 補題 1 より AC 上にも B が存在することとなり, 同じく矛盾する. よって, B は AD 上にない. さらに △ABC および定義より, 二点 A, B は相異なる点である. ここで, 三点 A, B, D が相異なる点であり, 二点 A, D を通る直線上に残りの一点 B がないため, 定理 8 よりこの三点は三角形をつくる. すなわち, △ABD を導ける. △ABD について, Bet (A, D) C ∧ AB ≡ AD であるため, 補助命題 35 より, ∠ACB > ∠ABC である. これは, ¬ (∠ABC < ∠ACB ∧ ∠ACB > ∠ABC) に矛盾する.

¬ AC ≡ AD について, 三点の位置としてあり得るすべての場合について矛盾が導かれる. すなわち AC ≡ AD である. したがって, 定義および HaⅢ-2 より, AB ≡ AC となり, △ABC が二等辺三角形であることが導かれる. これで, 定理の求めるところが証明された.

こうして, Ht-23 を用いることなく Ht-24 が証明される.

4 考察

本研究における主要な目的は, 自動定理証明において補助命題を作成する必要性を明らかにし、それらによって発見した諸問題の解決を行なうことであった. その目的を達成するために題材として, 平面幾何の範囲かつ点列と平行の概念を除いた幾何学基礎論に対して Isabelle を用いて証明出来ることを示した.

今回作成したプログラムは, その冗長性について最適化を試みていない. Isabelle の仕様として, 証明すべき複数の副題を同時に解決させるようなプログラムの実現は多くの場合において可能である. しかし一度に多くの副題を扱おうとすると処理時間が増加することが問題である. すなわち行数と処理時間のどちらの増加を問題視するか, という問いに対し後者を選択した形である. これにも理由は存在し, Isabelle には定理ファイルの冒頭にて指定した別の定理ファイルを読み込み, 新たなプログラムを追加できる. そのため今回作成した補助命題を流用した新たな証明研究の可能性が見込める. 処理時間が膨大となった定理ファイルは, この機能を使い読み込む際に支障をきたす可能性が高い. そのため本研究では, プログラムとしての見易さよりファイルとしての実用性を重視した.

角の取り扱いとして, 線角より点角を採用したことの是非については, 概ね想定通りの効果を確認した. [10] でも示されている通り, 三角形の合同条件について実装に特別な苦労を必要としない点が大きく, また前章補助命題 30 の証明のように, 各点の位置関係が角に関わらず指定済みである状況が多いため, 線角の優位性を利用できる機会が特に見当たらないという結果となった.

4.1 本研究の成果

今回作成した 35 の補助命題が, 既存定理の実装にどのように使用されたかをまとめた表が以下のようになる. なお, 使用された補助命題の証明に使用された補助命題も各項目に含めている.

既存定理	補助命題
Ht-11	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16
Ht-12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17
Ht-13	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17
Ht-14	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,19
Ht-15	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18 19,20,21,22,31
Ht-16	2, 14,31
Ht-17	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18 19,20,21,22,31
Ht-18	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18 19,20,21,22,31
Ht-19	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18 19,20,21,22,31
Ht-20	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18 19,20,21,22,31
Ht-21	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18 19,20,21,22,23,24,25,26,27,28,29,30,31,32
Ht-22	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18 19,20,21,22,23,24,25,26,27,28,31
Ht-24	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18 19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35
Ht-25	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18 19,20,21,22,23,24,25,26,27,28,29,31
Ht-26	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18 19,20,21,22,23,24,25,26,27,28,29,30,31,32

表 1: 補助命題と既存定理の対応表

各補助命題がそれぞれどのような目的で導出されたかを, 以下に改めて示す.

補助命題 1〜8 は, 間や平面における直線の側の定義から導かれる定理であり, 点や直線の位置関係を示す目的で多くの定理に繰り返し使用されている.

補助命題 9〜12 は, 角の内部にある点と頂点を含む直線と, その角を構成する二直線の関係を示す定理であり, 一方の半直線を共有する二角の関係等の証明に用いられる.

補助命題 13 は, 一方の半直線を共有する二角について, それが合同な角でない限り, 一方が他方の内部にあるか外部にあることを証明する定理である. 後に任意の二角の大小関係に関する補助命題の導出に必要となる.

補助命題 14〜17 は, 三角形を構成する条件に関する定理である. 「三角形から一部を切り取ったものもまた三角形である」という認識は, 紙上に描いた図を人間の目で見る限りは自明であるが, ソフトウェアの上にそれを表現するにはこのような補助命題を用意しなければならない.

補助命題 18 は, 一直線上の三点について, ある一点について残りの二点が同じ側にありかつ, それぞれと構築される線分が合同であるならば, 二点が同一の点であるとする定理である. この補助命題および線分を合同に移すことの一意可能性に関する既存定理から, 一直線上に移した合同な二線分から同一の点を見出すことに使われている.

補助命題 19〜22 は, 線分の合同関係と間の関係に関する制約を見出す定理である. この定理は, Ht-15 の実装に際して発見された問題の解決を行っている.

補助命題 23〜32 は, 角の大小関係の唯一性および推移性に関する定理である. [9] の中ではこれらの関係性を示すにとどまり, その保証については言及されていない.

補助命題 33〜35 は, 線分の大小を前提に持つ既存定理 Ht-23 を, 間の定義を用いて再構築した

ものである. これにより未定義である線分の大小を利用しない Ht-24 の証明を実現している.

今回作成した補助命題は, いずれも既存の公理, 定義, 定理から導かれるものであり, 新たな定義を必要とするものではない. しかしながらこの表は, 多くの既存定理の証明に補完の余地が残されていたことを示しており, 幾何学という分野において人間の認識に拠る「暗黙の了解」が, いかに多くの場面で用いられてきたかが窺えるものとなっている.

4.2 今後の課題

今回作成した定理ファイルには, 合同公理の一部および平行線公理以降の定義, 定理は実装されていない. これらが第 1 章において示した高階述語論理に属するために, 有効な目標定理の否定を導き出す実装が論理的に可能であるかどうかについては, 本研究では明らかに出来なかった.

今回の研究では一階述語論理の範囲でしか使用していないが, Isabelle のライブラリには高階述語論理を扱う開発環境 (HOL) が含まれている.

Isabelle/HOL はそもそも証明支援システムであり対話式である. そのため機械的な証明は行うが, 全自動ではなく, 人間の手で証明する部分が必要である. つまり機械と人間が協力し合って証明を完成させる形態である.

示したい主張が簡単な場合 (命題論理のような決定可能な論理で記述できる場合) は, 自動で処理されるが, それ以外の多くの場合は, まず人間が紙で証明を書き, その証明の流れに従って機械が証明を補いながら形式的な証明を共同で作成する作業をすることになる. 高階述語論理は決定不能であり, 機械的な証明 (自動証明) はできず, 人間が証明を作っておく必要がある。

そのため合同公理の一部および平行線公理以降については, 如何にして「妥当な」プログラムを実現するかが今後の課題となる.

また本研究の発展可能性の一つとして, Ht-23 の実装に際して見つかった問題が挙げられる. 前章で示したように, [9] において線分の大小は, その関係性が角の大小と対応するという主張に留まり, それ自体の定義が与えられていない. 角の大小の関係性については今回の補助命題により導き出せるため, これを利用した独自の定義を作成することが可能であると予想される.

以上のような, 幾何学基礎論の題材を基に補助命題の必要性を明らかに出来たことが本研究の成果であり, 今後の課題もこの題材が有望であることを示すことが出来た.

付録 - Isabelle プログラム

A.1 幾何学基礎論

```
theory Incidence imports Main begin

datatype Point = ”char”
datatype Segment = Se ”Point” ”Point”
datatype Line = Li ”Point” ”Point”
datatype Angle = An ”Point” ”Point” ”Point”
datatype Triangle = Tr ”Point” ”Point” ”Point”
datatype Geo_object =
   Poi ”Point”
   | Seg ”Segment”
   | Lin ”Line”
   | Ang ”Angle”
   | Tri ”Triangle”
datatype sign = add | sub
datatype Geo_objects = Emp | Geos ”Geo_object” ”sign” ”Geo_objects”

locale Eq_relation =
   fixes Eq :: ”Geo_objects ⇒ Geo_objects ⇒ bool”
      and Inv :: ”bool ⇒ bool”
   assumes Eq_refl [simp,intro] : ”Eq obs obs”
      and Eq_rev : ”⟦Eq obs1 obs2⟧⟹ Eq obs2 obs1”
      and Eq_trans : ”⟦Eq obs1 obs3; Eq obs2 obs3⟧⟹ Eq obs1 obs2”
      and Inv_def : ”Inv b1 ⟷ ¬ b1”

locale Definition_1 = Eq_relation +
   fixes Line_on :: ”Line ⇒ Point ⇒ bool”

locale Axiom_1 = Definition_1 +
   assumes Line_exist : ”⟦¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)⟧
          ⟹ ∃l. Line_on l p1 ∧ Line_on l p2”
      and Line_unique : ”⟦Line_on l1 p1; Line_on l1 p2; Line_on l2 p1; Line_on l2 p2;
          ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)⟧⟹ Eq (Geos (Lin l1) add Emp)
(Geos (Lin l2) add Emp)”
      and Line_on_exist : ”∃p q. Line_on l1 p ∧ Line_on l1 q
          ∧ ¬ Eq (Geos (Poi p) add Emp) (Geos (Poi q) add Emp)”
      and Line_not_on_exist : ”∃p q r. ¬ Line_on l1 p ∧ ¬ Line_on l1 q ∧ ¬ Line_on l1 r
          ∧ ¬ Eq (Geos (Poi p) add Emp) (Geos (Poi q) add Emp)
          ∧ ¬ Eq (Geos (Poi q) add Emp) (Geos (Poi r) add Emp)
          ∧ ¬ Eq (Geos (Poi r) add Emp) (Geos (Poi p) add Emp)”

locale Incidence_Rule = Axiom_1 +
   assumes Point_Eq : ”⟦P1(p1); Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)⟧⟹ P1(p2)”
      and Line_on_trans : ”⟦Eq (Geos (Lin l1) add Emp) (Geos (Lin l2) add Emp); Line_on l1 p1⟧
          ⟹ Line_on l2 p1”
      and Line_on_rule : ”Line_on (Li p1 p2) p1 ∧ Line_on (Li p1 p2) p2”

lemma(in Incidence_Rule) Eq_not_trans :
   assumes N :
      ”¬ Eq obs1 obs2”
      ”Eq obs2 obs3”
   shows ”¬ Eq obs1 obs3”
proof
   assume W : ”Eq obs1 obs3”
   from assms have P1 : ”Eq obs3 obs2” by (simp add:Eq_rev)
   from W P1 have P2 : ”Eq obs1 obs2” by (blast intro:Eq_trans)
   from N P2 show False by simp
qed

lemma(in Incidence_Rule) Line_rev :
   assumes ”¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)”
```

```
  shows "Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p2 p1)) add Emp)"
proof -
  from assms have P1 : "Line_on (Li p1 p2) p1 ∧ Line_on (Li p1 p2) p2" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li p2 p1) p1 ∧ Line_on (Li p2 p1) p2" by (simp add:Line_on_rule)
  from assms P1 P2 show "Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p2 p1)) add Emp)" by
(blast intro:Line_unique)
qed

lemma(in Incidence_Rule) Line_not_on_Point :
  assumes N :
    "¬ Line_on (Li p1 p2) p3"
  shows "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)"
proof
  assume W : "Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)"
  have P1 : "Line_on (Li p1 p2) p1" by (simp add:Line_on_rule)
  from W P1 have P2 : "Line_on (Li p1 p2) p3" by (simp add:Point_Eq)
  from N P2 show False by simp
qed

lemma(in Incidence_Rule) Line_not_on_trans :
  assumes
    "Eq (Geos (Lin l1) add Emp) (Geos (Lin l2) add Emp)"
    "¬ Line_on l1 p1"
  shows "¬ Line_on l2 p1"
proof -
  from assms have P1 : "Eq (Geos (Lin l2) add Emp) (Geos (Lin l1) add Emp)" by (simp add:Eq_rev)
  from P1 have P2 : "Line_on l2 p1 ⟹ Line_on l1 p1" by (simp add:Line_on_trans)
  from assms P2 show "¬ Line_on l2 p1" by blast
qed

lemma(in Incidence_Rule) Line_on_rev :
  assumes
    "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)"
    "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)"
    "Line_on (Li p1 p2) p3"
  shows "Line_on (Li p1 p3) p2"
proof -
  have P1 : "Line_on (Li p1 p2) p1" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li p1 p3) p1" by (simp add:Line_on_rule)
  have P3 : "Line_on (Li p1 p3) p3" by (simp add:Line_on_rule)
  from assms P1 P2 P3 have P4 : "Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p1 p3)) add
Emp)" by (simp add:Line_unique)
  have P5 : "Line_on (Li p1 p2) p2" by (simp add:Line_on_rule)
  from P4 P5 show "Line_on (Li p1 p3) p2" by (simp add:Line_on_trans)
qed

lemma(in Incidence_Rule) Line_not_Eq :
  assumes
    "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)"
    "¬ Line_on (Li p1 p2) p3"
  shows "¬ Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p1 p3)) add Emp)"
proof -
  have P1 : "Line_on (Li p1 p3) p3" by (simp add:Line_on_rule)
  have P2 : "Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p1 p3)) add Emp) ⟹
    Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin (Li p1 p2)) add Emp)" by (simp add:Eq_rev)
  from P1 P2 have P3 : "Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p1 p3)) add Emp) ⟹
    Line_on (Li p1 p2) p3" by (simp add:Line_on_trans)
  from assms P3 show "¬ Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p1 p3)) add Emp)" by
blast
qed

lemma(in Incidence_Rule) Line_not_Eq_on :
  assumes N :
    "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)"
    "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)"
    "¬ Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p1 p3)) add Emp)"
```

```
  shows "¬ Line_on (Li p1 p2) p3"
proof
  assume W : "Line_on (Li p1 p2) p3"
  have P1 : "Line_on (Li p1 p2) p1" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li p1 p3) p1" by (simp add:Line_on_rule)
  have P3 : "Line_on (Li p1 p3) p3" by (simp add:Line_on_rule)
  from N W P1 P2 P3 have P4 : "Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p1 p3)) add Emp)"
by (simp add:Line_unique)
  from N P4 show False by simp
qed

lemma(in Incidence_Rule) Line_unique_Point :
  assumes
    "¬ Eq (Geos (Lin l1) add Emp) (Geos (Lin l2) add Emp)"
    "Line_on l1 p1" "Line_on l1 p2"
    "Line_on l2 p1" "Line_on l2 p2"
  shows "Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)"
proof -
  from assms have P1 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp) ⟹
    Eq (Geos (Lin l1) add Emp) (Geos (Lin l2) add Emp)" by (simp add:Line_unique)
  from assms P1 show "Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)" by blast
qed

lemma(in Incidence_Rule) Line_not_on_Eq :
  assumes N :
    "¬ Line_on l1 p1"
    "Line_on l2 p1"
  shows "¬ Eq (Geos (Lin l1) add Emp) (Geos (Lin l2) add Emp)"
proof
  assume W : "Eq (Geos (Lin l1) add Emp) (Geos (Lin l2) add Emp)"
  from N W have P1 : "Line_on l1 p1" by (blast intro:Line_on_trans Eq_rev)
  from N P1 show False by simp
qed

lemma(in Incidence_Rule) Line_cross_not_on :
  assumes
    "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)"
    "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p4) add Emp)"
    "¬ Line_on (Li p1 p2) p3"
    "Line_on (Li p2 p3) p4"
  shows "¬ Line_on (Li p1 p2) p4"
proof -
  have P1 : "Line_on (Li p1 p2) p2" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li p2 p3) p2" by (simp add:Line_on_rule)
  from assms P1 P2 have P3 : "Line_on (Li p1 p2) p4 ⟹ Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos
(Lin (Li p2 p3)) add Emp)" by (simp add:Line_unique)
  have P4 : "Line_on (Li p2 p3) p3" by (simp add:Line_on_rule)
  from P3 P4 have P5 : "Line_on (Li p1 p2) p4 ⟹ Line_on (Li p1 p2) p3" by (blast intro:Line_on_trans
Eq_rev)
  from assms P5 show "¬ Line_on (Li p1 p2) p4" by blast
qed

end


theory Order imports Incidence begin

locale Definition_2 = Incidence_Rule +
  fixes Line_on_Seg :: "Line ⇒ Segment ⇒ bool"
    and Bet_Point :: "Segment ⇒ Point ⇒ bool"
    and Seg_on_Seg :: "Segment ⇒ Segment ⇒ bool"
    and Line_on_Line :: "Line ⇒ Line ⇒ bool"
    and Plane_sameside :: "Line ⇒ Point ⇒ Point ⇒ bool"
    and Plane_diffside :: "Line ⇒ Point ⇒ Point ⇒ bool"
  assumes Bet_Point_def : "⟦Bet_Point (Se p1 p2) p3⟧⟹ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi
p2) add Emp)
```

```
        ∧ ¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp) ∧ ¬ Eq (Geos (Poi p3) add Emp)
(Geos (Poi p1) add Emp)"
    and Bet_rev : "⟦Bet_Point (Se p1 p2) p3⟧⟹ Bet_Point (Se p2 p1) p3"
    and Line_Bet_exist : "⟦Bet_Point (Se p1 p2) p3⟧⟹ ∃l. Line_on l p1 ∧ Line_on l p2 ∧ Line_on l
p3"
    and Seg_rev : "Eq (Geos (Seg (Se p1 p2)) add Emp) (Geos (Seg (Se p2 p1)) add Emp)"
    and Plane_sameside_def : "Plane_sameside l1 p1 p2 ⟷ ¬ Line_on_Seg l1 (Se p1 p2) ∧ ¬ Line_on
l1 p1 ∧ ¬ Line_on l1 p2 ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)"
    and Plane_diffside_def : "Plane_diffside l1 p1 p2 ⟷ (∃p. Bet_Point (Se p1 p2) p ∧ Line_on l1 p
∧ ¬ Line_on l1 p1 ∧ ¬ Line_on l1 p2)"

locale Axiom_2 = Definition_2 +
  assumes Bet_extension : "⟦Line_on l1 p1; Line_on l1 p2; ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi
p2) add Emp)⟧⟹ ∃p. Bet_Point (Se p1 p) p2 ∧ Line_on l1 p"
    and Bet_iff : "⟦Bet_Point (Se p1 p2) p3⟧⟹ Inv (Bet_Point (Se p2 p3) p1) ∧ Inv (Bet_Point (Se p3
p1) p2)"
    and Pachets_axiom : "⟦¬ Line_on (Li p1 p2) p3; Bet_Point (Se p1 p2) p4; Line_on l1 p4;
        ¬ Line_on l1 p1; ¬ Line_on l1 p2; ¬ Line_on l1 p3⟧⟹
        Line_on_Seg l1 (Se p1 p3) ∧ ¬ Line_on_Seg l1 (Se p2 p3)
        ∨ Line_on_Seg l1 (Se p2 p3) ∧ ¬ Line_on_Seg l1 (Se p1 p3)"
    and Seg_move_sameside : "⟦Line_on l1 p1; Line_on l1 p2; ¬ Eq (Geos (Poi p1) add Emp) (Geos
(Poi p2) add Emp);
        ¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p4) add Emp)⟧⟹
        ∃p. Eq (Geos (Seg (Se p3 p4)) add Emp) (Geos (Seg (Se p1 p)) add Emp) ∧ ¬ Bet_Point (Se
p p2) p1 ∧ Line_on l1 p ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p) add Emp)"
    and Seg_move_diffside : "⟦Line_on l1 p1; Line_on l1 p2; ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi
p2) add Emp);
        ¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p4) add Emp)⟧⟹
        ∃p. Eq (Geos (Seg (Se p3 p4)) add Emp) (Geos (Seg (Se p1 p)) add Emp) ∧ Bet_Point (Se p
p2) p1 ∧ Line_on l1 p ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p) add Emp)"

locale Order_Rule = Axiom_2 +
  assumes Bet_Point_Eq : "⟦Bet_Point (Se p1 p2) p3; Eq (Geos (Poi p1) add Emp) (Geos (Poi p4) add
Emp)⟧⟹ Bet_Point (Se p4 p2) p3"
    and Line_on_Seg_rule : "Line_on_Seg l1 (Se p1 p2) ⟷ (∃p. Line_on l1 p ∧ Bet_Point (Se p1 p2)
p)"
    and Seg_on_Seg_rule : "Seg_on_Seg (Se p1 p2) (Se p3 p4) ⟷ (∃p. Bet_Point (Se p1 p2) p ∧
Bet_Point (Se p3 p4) p)"
    and Line_on_Line_rule : "Line_on_Line l1 l2 ⟷ (∃p. Line_on l1 p ∧ Line_on l2 p)"
    and Seg_Point_Eq : "⟦Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)⟧⟹ Eq (Geos (Seg
(Se p3 p1)) add Emp) (Geos (Seg (Se p3 p2)) add Emp)"

lemma(in Order_Rule) Line_Bet_on :
  assumes
    "Bet_Point (Se p1 p2) p3"
  shows "Line_on (Li p1 p2) p3" and "Line_on (Li p2 p1) p3"
    and "Line_on (Li p2 p3) p1" and "Line_on (Li p3 p2) p1"
    and "Line_on (Li p1 p3) p2" and "Line_on (Li p3 p1) p2"
proof -
  from assms have "∃l. Line_on l p1 ∧ Line_on l p2 ∧ Line_on l p3" by (blast intro:Line_Bet_exist)
  then obtain l1 :: Line where P1 : "Line_on l1 p1 ∧ Line_on l1 p2 ∧ Line_on l1 p3" by blast
  from assms have P2 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)"
    by (simp add:Bet_Point_def)
  have P3 : "Line_on (Li p1 p2) p1 ∧ Line_on (Li p1 p2) p2" by (simp add:Line_on_rule)
  from P1 have P4 : "Line_on l1 p1" by simp
  from P1 have P5 : "Line_on l1 p2" by simp
  from P2 P3 P4 P5 have P6 : "Eq (Geos (Lin l1) add Emp) (Geos (Lin (Li p1 p2)) add Emp)" by (simp
add:Line_unique)
  from P1 P6 show P7 : "Line_on (Li p1 p2) p3" by (simp add:Line_on_trans)
  from assms have P8 : "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p1) add Emp)"
    by (simp add:Bet_Point_def)
  from P2 P7 P8 show "Line_on (Li p1 p3) p2" by (blast intro:Line_on_rev Eq_rev)
  from P2 P7 P8 show "Line_on (Li p3 p1) p2" by (blast intro:Line_on_trans Line_on_rev Eq_rev Line_rev)
  from P2 P7 show "Line_on (Li p2 p1) p3" by (blast intro:Line_on_trans Line_rev)
```

```
  from assms have P9 : "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp)"
    by (simp add:Bet_Point_def)
  from P2 P7 P9 show "Line_on (Li p2 p3) p1" by (blast intro:Line_on_rev Line_on_trans Line_rev
Eq_rev)
  from P9 have P10 : "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p2) add Emp)" by (blast intro:Eq_rev)
  from assms P2 P7 P8 P10 show "Line_on (Li p3 p2) p1" by (blast intro:Line_on_rev Bet_Point_def
Line_on_trans Eq_rev Line_rev)
qed

lemma(in Order_Rule) Line_Bet_not_Eq :
  assumes N :
    "Bet_Point (Se p1 p2) p3"
    "¬ Line_on (Li p1 p2) p4"
  shows "¬ Eq (Geos (Lin (Li p4 p3)) add Emp) (Geos (Lin (Li p4 p2)) add Emp)"
proof
  assume W : "Eq (Geos (Lin (Li p4 p3)) add Emp) (Geos (Lin (Li p4 p2)) add Emp)"
  have P1 : "Line_on (Li p4 p3) p3" by (simp add:Line_on_rule)
  from W P1 have P2 : "Line_on (Li p4 p2) p3" by (simp add:Line_on_trans)
  have P3 : "Line_on (Li p4 p2) p2" by (simp add:Line_on_rule)
  from N have P4 : "Line_on (Li p1 p2) p3" by (simp add:Line_Bet_on)
  have P5 : "Line_on (Li p1 p2) p2" by (simp add:Line_on_rule)
  from assms have P6 : "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp)"
    by (simp add:Bet_Point_def)
  from P2 P3 P4 P5 P6 have P7 : "Eq (Geos (Lin (Li p4 p2)) add Emp) (Geos (Lin (Li p1 p2)) add
Emp)" by (simp add:Line_unique)
  have P8 : "Line_on (Li p4 p2) p4" by (simp add:Line_on_rule)
  from P7 P8 have P9 : "Line_on (Li p1 p2) p4" by (simp add:Line_on_trans)
  from N P9 show False by simp
qed

theorem(in Order_Rule) Seg_density :
  assumes "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)"
  shows "∃p. Bet_Point (Se A C) p"
proof -
  have "∃p q r. ¬ Line_on (Li A C) p ∧ ¬ Line_on (Li A C) q ∧ ¬ Line_on (Li A C) r
          ∧ ¬ Eq (Geos (Poi p) add Emp) (Geos (Poi q) add Emp) ∧ ¬ Eq (Geos (Poi q) add Emp) (Geos
(Poi r) add Emp)
          ∧ ¬ Eq (Geos (Poi r) add Emp) (Geos (Poi p) add Emp)" by (blast intro:Line_not_on_exist)
  then obtain E :: Point where P1 : "¬ Line_on (Li A C) E" by blast
  then have P2 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi E) add Emp)" by (simp add:Line_not_on_Point)
  have P3 : "Line_on (Li A E) A ∧ Line_on (Li A E) E" by (simp add:Line_on_rule)
  from P2 P3 have "∃p. Bet_Point (Se A p) E ∧ Line_on (Li A E) p" by (simp add:Bet_extension)
  then obtain F :: Point where P4 : "Bet_Point (Se A F) E ∧ Line_on (Li A E) F" by blast
  then have P5 : "Line_on (Li A F) E" by (simp add:Line_Bet_on)
  from P4 have P6 : "Bet_Point (Se A F) E" by simp
  from P6 have P7 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi F) add Emp)" by (simp add:Bet_Point_def)
  from P2 P4 P6 P7 have P8 : "Line_on (Li A E) F" by (simp add:Line_on_rev)
  from assms P1 have P9 : "¬ Eq (Geos (Lin (Li A C)) add Emp) (Geos (Lin (Li A E)) add Emp)" by
(simp add:Line_not_Eq)
  have P10 : "Line_on (Li A F) A" by (simp add:Line_on_rule)
  from P2 P3 P5 P10 have P11 : "Eq (Geos (Lin (Li A E)) add Emp) (Geos (Lin (Li A F)) add Emp)"
by (blast intro:Line_unique)
  from P9 P11 have P12 : "¬ Eq (Geos (Lin (Li A C)) add Emp) (Geos (Lin (Li A F)) add Emp)" by
(simp add:Eq_not_trans)
  from assms P7 P12 have P13 : "¬ Line_on (Li A C) F" by (simp add:Line_not_Eq_on)
  from assms P7 P13 have P14 : "¬ Line_on (Li A F) C" by (blast intro:Line_on_rev)
  have "Line_on (Li A F) F" by (simp add:Line_on_rule)
  then have P15 : "Eq (Geos (Poi F) add Emp) (Geos (Poi C) add Emp) ⟹ Line_on (Li A F) C" by
(simp add:Point_Eq)
  from P14 P15 have P16 : "¬ Eq (Geos (Poi F) add Emp) (Geos (Poi C) add Emp)" by blast
  have P17 : "Line_on (Li F C) F ∧ Line_on (Li F C) C" by (simp add:Line_on_rule)
  from P16 P17 have "∃p. Bet_Point (Se F p) C ∧ Line_on (Li F C) p" by (simp add:Bet_extension)
  then obtain G :: Point where P18 : "Bet_Point (Se F G) C ∧ Line_on (Li F C) G" by blast
  from P18 have P19 : "Line_on (Li F G) C" by (simp add:Line_Bet_on)
  from P18 have P20 : "Bet_Point (Se F G) C" by simp
  then have P21 : "¬ Eq (Geos (Poi F) add Emp) (Geos (Poi G) add Emp)" by (simp add:Bet_Point_def)
```

from P20 have P22 : ”Line_on (Li F C) G” by (simp add:Line_Bet_on)
from P7 P14 P21 P22 have P23 : ”¬ Line_on (Li A F) G” by (simp add:Line_cross_not_on)
from P6 P23 have P24 : ”¬ Eq (Geos (Lin (Li G E)) add Emp) (Geos (Lin (Li G F)) add Emp)” by (simp add:Line_Bet_not_Eq)
from P5 have P25 : ”Eq (Geos (Poi E) add Emp) (Geos (Poi G) add Emp) ⟹ Line_on (Li A F) G” by (simp add:Point_Eq)
from P23 P25 have P26 : ”¬ Eq (Geos (Poi G) add Emp) (Geos (Poi E) add Emp)” by (blast intro:Eq_rev)
from P21 have P27 : ”¬ Eq (Geos (Poi G) add Emp) (Geos (Poi F) add Emp)” by (blast intro:Eq_rev)
from P24 P26 P27 have P28 : ”¬ Line_on (Li G E) F” by (simp add:Line_not_Eq_on)
from P26 P28 have P29 : ”¬ Line_on (Li E G) F” by (blast intro:Line_rev Line_on_trans Eq_rev)
have P30 : ”Line_on (Li E G) E” by (simp add:Line_on_rule)
have P31 : ”Line_on (Li A E) E” by (simp add:Line_on_rule)
have P32 : ”Line_on (Li A E) A” by (simp add:Line_on_rule)
from P2 P30 P31 P32 have P33 : ”Line_on (Li E G) A ⟹ Eq (Geos (Lin (Li A E)) add Emp) (Geos (Lin (Li E G)) add Emp)” by (simp add:Line_unique)
from P8 P33 have P34 : ”Line_on (Li E G) A ⟹ Line_on (Li E G) F” by (simp add:Line_on_trans)
from P29 P34 have P35 : ”¬ Line_on (Li E G) A” by blast
have P36 : ”Line_on (Li E G) G” by (simp add:Line_on_rule)
have P37 : ”Line_on (Li F G) G” by (simp add:Line_on_rule)
from P20 have P38 : ”¬ Eq (Geos (Poi G) add Emp) (Geos (Poi C) add Emp)” by (simp add:Bet_Point_def)
from P19 P36 P37 P38 have P39 : ”Line_on (Li E G) C ⟹ Eq (Geos (Lin (Li F G)) add Emp) (Geos (Lin (Li E G)) add Emp)” by (simp add:Line_unique)
have P40 : ”Line_on (Li F G) F” by (simp add:Line_on_rule)
from P39 P40 have P41 : ”Line_on (Li E G) C ⟹ Line_on (Li E G) F” by (simp add:Line_on_trans)
from P29 P41 have P42 : ”¬ Line_on (Li E G) C” by blast
from P6 P14 P29 P30 P35 P42 have P43 : ”Line_on_Seg (Li E G) (Se A C) ∧ ¬ Line_on_Seg (Li E G) (Se F C) ∨ Line_on_Seg (Li E G) (Se F C) ∧ ¬ Line_on_Seg (Li E G) (Se A C)” by (simp add:Pachets_axiom)
then have ”Line_on_Seg (Li E G) (Se F C) ⟹ ∃p. Line_on (Li E G) p ∧ Bet_Point (Se F C) p” by (simp add:Line_on_Seg_rule)
then obtain D :: Point where P44 : ”Line_on_Seg (Li E G) (Se F C) ⟹ Line_on (Li E G) D ∧ Bet_Point (Se F C) D” by blast
from P44 have P46 : ”Line_on_Seg (Li E G) (Se F C) ⟹ Bet_Point (Se F C) D” by simp
from P46 have ”Line_on_Seg (Li E G) (Se F C) ⟹ ¬ Eq (Geos (Poi D) add Emp) (Geos (Poi F) add Emp)” by (simp add:Bet_Point_def)
from P46 have P47 : ”Line_on_Seg (Li E G) (Se F C) ⟹ Line_on (Li F D) C” by (simp add:Line_Bet_on)
have P48 : ”Line_on (Li F D) F” by (simp add:Line_on_rule)
have P49 : ”Line_on (Li F G) F” by (simp add:Line_on_rule)
from P16 P19 P47 P48 P49 have P50 : ”Line_on_Seg (Li E G) (Se F C) ⟹
Eq (Geos (Lin (Li F D)) add Emp) (Geos (Lin (Li F G)) add Emp)” by (simp add:Line_unique)
have P51 : ”Line_on (Li F D) D” by (simp add:Line_on_rule)
from P50 P51 have P52 : ”Line_on_Seg (Li E G) (Se F C) ⟹ Line_on (Li F G) D” by (simp add:Line_on_trans)
have P53 : ”Line_on (Li F G) G” by (simp add:Line_on_rule)
have P54 : ”Line_on (Li E G) G” by (simp add:Line_on_rule)
from P46 have P55 : ”Line_on_Seg (Li E G) (Se F C) ⟹ Eq (Geos (Poi D) add Emp) (Geos (Poi G) add Emp)
⟹ Bet_Point (Se F C) G” by (simp add:Point_Eq)
from P20 have ”Inv (Bet_Point (Se G C) F) ∧ Inv (Bet_Point (Se C F) G)” by (simp add:Bet_iff)
then have ”¬ Bet_Point (Se C F) G” by (simp add:Inv_def)
then have P56 : ”¬ Bet_Point (Se F C) G” by (blast intro:Bet_rev)
from P55 P56 have P57 : ”Line_on_Seg (Li E G) (Se F C) ⟹ ¬ Eq (Geos (Poi D) add Emp) (Geos (Poi G) add Emp)” by blast
from P44 P52 P53 P54 P57 have P58 : ”Line_on_Seg (Li E G) (Se F C) ⟹
Eq (Geos (Lin (Li E G)) add Emp) (Geos (Lin (Li F G)) add Emp)” by (blast intro:Line_unique)
from P26 have P59 : ”Eq (Geos (Lin (Li E G)) add Emp) (Geos (Lin (Li G E)) add Emp)” by (simp add:Line_rev Eq_rev)
from P27 have P60 : ”Eq (Geos (Lin (Li F G)) add Emp) (Geos (Lin (Li G F)) add Emp)” by (simp add:Line_rev Eq_rev)
from P58 P59 P60 have P61 : ”Line_on_Seg (Li E G) (Se F C) ⟹
Eq (Geos (Lin (Li G E)) add Emp) (Geos (Lin (Li G F)) add Emp)” by (blast intro:Eq_trans Eq_rev)
from P24 P61 have P62 : ”¬ Line_on_Seg (Li E G) (Se F C)” by blast
from P43 P62 have ”Line_on_Seg (Li E G) (Se A C) ∧ ¬ Line_on_Seg (Li E G) (Se F C)” by blast
then have ”∃p. Line_on (Li E G) p ∧ Bet_Point (Se A C) p” by (simp add:Line_on_Seg_rule)
thus ”∃p. Bet_Point (Se A C) p” by blast

```
qed

lemma(in Order_Rule) Line_Bet_not_on :
  assumes
    "Line_on (Li p1 p2) p3"
    "¬ Line_on (Li p1 p2) p4"
    "Bet_Point (Se p3 p4) p5"
  shows "Inv (Line_on (Li p1 p2) p5)"
proof -
  from assms have "¬ Eq (Geos (Poi p5) add Emp) (Geos (Poi p3) add Emp)" by (simp add:Bet_Point_def)
  then have P1 : "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p5) add Emp)" by (blast intro:Eq_rev)
  from assms have P2 : "Line_on (Li p3 p5) p4" by (simp add:Line_Bet_on)
  have P3 : "Line_on (Li p3 p5) p3" by (simp add:Line_on_rule)
  have P4 : "Line_on (Li p3 p5) p5" by (simp add:Line_on_rule)
  from assms P1 P3 P4 have P5 : "Line_on (Li p1 p2) p5 ⟹ Eq (Geos (Lin (Li p3 p5)) add Emp) (Geos
(Lin (Li p1 p2)) add Emp)" by (simp add:Line_unique)
  from P2 P5 have P6 : "Line_on (Li p1 p2) p5 ⟹ Line_on (Li p1 p2) p4" by (simp add:Line_on_trans)
  from assms P6 have "¬ Line_on (Li p1 p2) p5" by blast
  thus "Inv (Line_on (Li p1 p2) p5)" by (simp add:Inv_def)
qed

lemma(in Order_Rule) Line_not_on_ex :
  assumes N :
    "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)"
    "¬ Line_on (Li p1 p2) p3"
    "Line_on (Li p1 p4) p2"
  shows "¬ Line_on (Li p1 p4) p3"
proof
  assume W : "Line_on (Li p1 p4) p3"
  have P1 : "Line_on (Li p1 p2) p2" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li p1 p2) p1" by (simp add:Line_on_rule)
  have P3 : "Line_on (Li p1 p4) p1" by (simp add:Line_on_rule)
  from N P1 P2 P3 have P4 : "Eq (Geos (Lin (Li p1 p4)) add Emp) (Geos (Lin (Li p1 p2)) add Emp)"
by (simp add:Line_unique)
  from W P4 have P5 : "Line_on (Li p1 p2) p3" by (simp add:Line_on_trans)
  from N P5 show False by simp
qed

lemma(in Order_Rule) Line_on_dens :
  assumes
    "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)"
    "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p4) add Emp)"
    "Line_on (Li p1 p2) p3"
    "Line_on (Li p1 p4) p3"
  shows "Line_on (Li p2 p4) p3"
proof -
  have P1 : "Line_on (Li p1 p2) p1" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li p1 p4) p1" by (simp add:Line_on_rule)
  from assms P1 P2 have P3 : "Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p1 p4)) add Emp)"
by (simp add:Line_unique)
  have P4 : "Line_on (Li p1 p2) p2" by (simp add:Line_on_rule)
  from P3 P4 have P5 : "Line_on (Li p1 p4) p2" by (simp add:Line_on_trans)
  have P6 : "Line_on (Li p1 p4) p4" by (simp add:Line_on_rule)
  have P7 : "Line_on (Li p2 p4) p2" by (simp add:Line_on_rule)
  have P8 : "Line_on (Li p2 p4) p4" by (simp add:Line_on_rule)
  from assms P5 P6 P7 P8 have P9 : "Eq (Geos (Lin (Li p1 p4)) add Emp) (Geos (Lin (Li p2 p4)) add
Emp)" by (simp add:Line_unique)
  from assms P9 show "Line_on (Li p2 p4) p3" by (simp add:Line_on_trans)
qed

lemma(in Order_Rule) Bet_case_lemma1 :
  assumes
    "Line_on l1 A"
    "Line_on l1 B"
    "Line_on l1 C"
    "¬ Bet_Point (Se B A) C"
```

```
    "¬ Bet_Point (Se C B) A"
    "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)"
    "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)"
    "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)"
    "¬ Line_on (Li A C) D"
    "Bet_Point (Se B G) D"
  shows "∃p. Line_on (Li A D) p ∧ Bet_Point (Se G C) p"
proof -
  have P1 : "Line_on (Li A C) A" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li A C) C" by (simp add:Line_on_rule)
  from assms P1 P2 have P3 : "Eq (Geos (Lin l1) add Emp) (Geos (Lin (Li A C)) add Emp)" by (simp
add:Line_unique)
  from assms P3 have P4 : "Line_on (Li A C) B" by (simp add:Line_on_trans)
  have P11 : "Line_on (Li B G) B" by (simp add:Line_on_rule)
  from assms P2 P4 P11 have P12 : "Line_on (Li B G) C ⟹ Eq (Geos (Lin (Li B G)) add Emp) (Geos
(Lin (Li A C)) add Emp)" by (simp add:Line_unique)
  from assms have P13 : "Line_on (Li B G) D" by (simp add:Line_Bet_on)
  from P12 P13 have P14 : "Line_on (Li B G) C ⟹ Line_on (Li A C) D" by (simp add:Line_on_trans)
  from assms P14 have P15 : "¬ Line_on (Li B G) C" by blast
  have P16 : "Line_on (Li A D) A" by (simp add:Line_on_rule)
  from assms P1 P4 P16 have P17 : "Line_on (Li A D) B ⟹ Eq (Geos (Lin (Li A D)) add Emp) (Geos
(Lin (Li A C)) add Emp)" by (simp add:Line_unique)
  have P18 : "Line_on (Li A D) D" by (simp add:Line_on_rule)
  from P17 P18 have P19 : "Line_on (Li A D) B ⟹ Line_on (Li A C) D" by (simp add:Line_on_trans)
  from assms P19 have P20 : "¬ Line_on (Li A D) B" by blast
  from assms P1 P2 P16 have P21 : "Line_on (Li A D) C ⟹ Eq (Geos (Lin (Li A D)) add Emp) (Geos
(Lin (Li A C)) add Emp)" by (simp add:Line_unique)
  from P18 P21 have P22 : "Line_on (Li A D) C ⟹ Line_on (Li A C) D" by (simp add:Line_on_trans)
  from assms P22 have P23 : "¬ Line_on (Li A D) C" by blast
  from assms P1 P4 P11 have P24 : "Line_on (Li B G) A ⟹ Eq (Geos (Lin (Li B G)) add Emp) (Geos
(Lin (Li A C)) add Emp)" by (simp add:Line_unique)
  from P13 P24 have P25 : "Line_on (Li B G) A ⟹ Line_on (Li A C) D" by (simp add:Line_on_trans)
  from assms P25 have P26 : "¬ Line_on (Li B G) A" by blast
  have P27 : "Line_on (Li B G) G" by (simp add:Line_on_rule)
  from assms have P28 : "¬ Eq (Geos (Poi G) add Emp) (Geos (Poi D) add Emp)"
    by (simp add:Bet_Point_def)
  from P13 P18 P27 P28 have P29 : "Line_on (Li A D) G ⟹ Eq (Geos (Lin (Li A D)) add Emp) (Geos
(Lin (Li B G)) add Emp)" by (simp add:Line_unique)
  from P16 P29 have P30 : "Line_on (Li A D) G ⟹ Line_on (Li B G) A" by (simp add:Line_on_trans)
  from P26 P30 have P31 : "¬ Line_on (Li A D) G" by blast
  from assms P15 P18 P20 P23 P31 have P32 : "Line_on_Seg (Li A D) (Se B C) ∧ ¬ Line_on_Seg (Li
A D) (Se G C)
    ∨ Line_on_Seg (Li A D) (Se G C) ∧ ¬ Line_on_Seg (Li A D) (Se B C)" by (simp add:Pachets_axiom)
  have "Line_on_Seg (Li A D) (Se B C) ⟹ ∃p. Line_on (Li A D) p ∧ Bet_Point (Se B C) p" by (simp
add:Line_on_Seg_rule)
  then obtain A2 :: Point where P33 : "Line_on_Seg (Li A D) (Se B C) ⟹ Line_on (Li A D) A2 ∧
Bet_Point (Se B C) A2" by blast
  from assms have P34 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)" by (blast intro:Eq_rev)
  from assms P34 have P35 : "¬ Eq (Geos (Lin (Li A C)) add Emp) (Geos (Lin (Li A D)) add Emp)"
by (simp add:Line_not_Eq)
  have P36 : "Line_on (Li B C) B" by (simp add:Line_on_rule)
  have P37 : "Line_on (Li B C) C" by (simp add:Line_on_rule)
  from assms P2 P4 P36 P37 have P38 : "Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li A C)) add
Emp)" by (simp add:Line_unique)
  from P33 have P39 : "Line_on_Seg (Li A D) (Se B C) ⟹ Bet_Point (Se B C) A2" by simp
  then have P40 : "Line_on_Seg (Li A D) (Se B C) ⟹ Line_on (Li B C) A2" by (simp add:Line_Bet_on)
  from P38 P40 have P41 : "Line_on_Seg (Li A D) (Se B C) ⟹ Line_on (Li A C) A2" by (simp
add:Line_on_trans)
  from P1 P16 P33 P35 P41 have P42 : "Line_on_Seg (Li A D) (Se B C) ⟹ Eq (Geos (Poi A2) add
Emp) (Geos (Poi A) add Emp)" by (simp add:Line_unique_Point)
  from P39 P42 have P43 : "Line_on_Seg (Li A D) (Se B C) ⟹ Bet_Point (Se B C) A" by (simp
add:Point_Eq)
  from assms have P44 : "¬ Bet_Point (Se B C) A" by (blast intro:Bet_rev)
  from P43 P44 have P45 : "¬ Line_on_Seg (Li A D) (Se B C)" by blast
  from P32 P45 have "Line_on_Seg (Li A D) (Se G C) ∧ ¬ Line_on_Seg (Li A D) (Se B C)" by blast
  thus "∃p. Line_on (Li A D) p ∧ Bet_Point (Se G C) p" by (simp add:Line_on_Seg_rule)
```

```
qed

lemma(in Order_Rule) Bet_case_lemma2 :
  assumes
    "Line_on l1 A"
    "Line_on l1 B"
    "Line_on l1 C"
    "¬ Bet_Point (Se B A) C"
    "¬ Bet_Point (Se C B) A"
    "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)"
    "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)"
    "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)"
  shows "Bet_Point (Se A C) B"
proof -
  have P1 : "Line_on (Li A C) A" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li A C) C" by (simp add:Line_on_rule)
  from assms P1 P2 have P3 : "Eq (Geos (Lin l1) add Emp) (Geos (Lin (Li A C)) add Emp)" by (simp
add:Line_unique)
  from assms P3 have P4 : "Line_on (Li A C) B" by (simp add:Line_on_trans)
  have "∃p q r. ¬ Line_on (Li A C) p ∧ ¬ Line_on (Li A C) q ∧ ¬ Line_on (Li A C) r
    ∧ ¬ Eq (Geos (Poi p) add Emp) (Geos (Poi q) add Emp) ∧ ¬ Eq (Geos (Poi q) add Emp) (Geos
(Poi r) add Emp)
    ∧ ¬ Eq (Geos (Poi r) add Emp) (Geos (Poi p) add Emp)" by (blast intro:Line_not_on_exist)
  then obtain D :: Point where P5 : "¬ Line_on (Li A C) D" by blast
  from P4 have P6 : "Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp) ⟹ Line_on (Li A C) D" by
(simp add:Point_Eq)
  from P5 P6 have P7 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)" by blast
  have P8 : "Line_on (Li B D) B" by (simp add:Line_on_rule)
  have P9 : "Line_on (Li B D) D" by (simp add:Line_on_rule)
  from P7 P8 P9 have "∃p. Bet_Point (Se B p) D ∧ Line_on (Li B D) p" by (simp add:Bet_extension)
  then obtain G :: Point where P10 : "Bet_Point (Se B G) D" by blast
  from assms P5 P10 have "∃p. Line_on (Li A D) p ∧ Bet_Point (Se G C) p"
    by (simp add:Bet_case_lemma1)
  then obtain E :: Point where P11 : "Line_on (Li A D) E ∧ Bet_Point (Se G C) E" by blast
  from assms have P12 : "¬ Bet_Point (Se B C) A" by (blast intro:Bet_rev)
  from assms have P13 : "¬ Bet_Point (Se A B) C" by (blast intro:Bet_rev)
  from assms have P14 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)" by (blast intro:Eq_rev)
  from assms have P15 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp)" by (blast intro:Eq_rev)
  from assms have P16 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi B) add Emp)" by (blast intro:Eq_rev)
  from P14 have P17 : "Eq (Geos (Lin (Li A C)) add Emp) (Geos (Lin (Li C A)) add Emp)" by (simp
add:Line_rev)
  from P5 P17 have P18 : "¬ Line_on (Li C A) D" by (simp add:Line_not_on_trans)
  from assms P10 P12 P13 P14 P15 P16 P18 have "∃p. Line_on (Li C D) p ∧ Bet_Point (Se G A) p" by
(simp add:Bet_case_lemma1)
  then obtain F :: Point where P19 : "Line_on (Li C D) F ∧ Bet_Point (Se G A) F" by blast
  have P20 : "Line_on (Li B G) B" by (simp add:Line_on_rule)
  have P21 : "Line_on (Li B G) G" by (simp add:Line_on_rule)
  from P10 have P22 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi G) add Emp)"
    by (simp add:Bet_Point_def)
  from P4 P20 P21 P22 have P23 : "Line_on (Li A C) G ⟹ Eq (Geos (Lin (Li B G)) add Emp) (Geos
(Lin (Li A C)) add Emp)" by (simp add:Line_unique)
  from P10 have P24 : "Line_on (Li B G) D" by (simp add:Line_Bet_on)
  from P23 P24 have P25 : "Line_on (Li A C) G ⟹ Line_on (Li A C) D" by (simp add:Line_on_trans)
  from P5 P25 have P26 : "¬ Line_on (Li A C) G" by blast
  from P11 have P27 : "Bet_Point (Se C G) E" by (blast intro:Bet_rev)
  have P28 : "Line_on (Li C G) C" by (simp add:Line_on_rule)
  from assms P1 P2 P28 have P29 : "Line_on (Li C G) A ⟹ Eq (Geos (Lin (Li C G)) add Emp) (Geos
(Lin (Li A C)) add Emp)" by (simp add:Line_unique)
  have P30 : "Line_on (Li C G) G" by (simp add:Line_on_rule)
  from P29 P30 have P31 : "Line_on (Li C G) A ⟹ Line_on (Li A C) G" by (simp add:Line_on_trans)
  from P26 P31 have P32 : "¬ Line_on (Li C G) A" by blast
  from P27 P32 have "¬ Eq (Geos (Lin (Li A E)) add Emp) (Geos (Lin (Li A G)) add Emp)" by (simp
add:Line_Bet_not_Eq)
  then have P33 : "¬ Eq (Geos (Lin (Li A G)) add Emp) (Geos (Lin (Li A E)) add Emp)" by (blast
intro:Eq_rev)
  from P19 have P34 : "Bet_Point (Se A G) F" by (blast intro:Bet_rev)
```

then have P35 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi G) add Emp)" by (simp add:Bet_Point_def)
from P27 have P36 : "Line_on (Li C G) E" by (simp add:Line_Bet_on)
then have P37 : "Eq (Geos (Poi E) add Emp) (Geos (Poi A) add Emp) ⟹ Line_on (Li C G) A" by (simp add:Point_Eq)
from P32 P37 have P38 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi E) add Emp)" by (blast intro:Eq_rev)
from P33 P35 P38 have P39 : "¬ Line_on (Li A G) E" by (simp add:Line_not_Eq_on)
from P14 P26 P35 have P40 : "¬ Line_on (Li A G) C" by (blast intro:Line_on_rev)
from P34 P40 have P41 : "¬ Eq (Geos (Lin (Li C F)) add Emp) (Geos (Lin (Li C G)) add Emp)" by (simp add:Line_Bet_not_Eq)
from P34 have P42 : "Line_on (Li A G) F" by (simp add:Line_Bet_on)
then have P43 : "Eq (Geos (Poi F) add Emp) (Geos (Poi C) add Emp) ⟹ Line_on (Li A G) C" by (simp add:Point_Eq)
from P40 P43 have P44 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi F) add Emp)" by (blast intro:Eq_rev)
from P27 have P45 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi G) add Emp)"
by (simp add:Bet_Point_def)
from P41 P44 P45 have P46 : "¬ Line_on (Li C F) G" by (simp add:Line_not_Eq_on)
from P35 have P47 : "Eq (Geos (Lin (Li A G)) add Emp) (Geos (Lin (Li G A)) add Emp)" by (simp add:Line_rev)
from P40 P47 have P48 : "¬ Line_on (Li G A) C" by (simp add:Line_not_on_trans)
from P19 have P49 : "Bet_Point (Se G A) F" by simp
from P48 P49 have P50 : "¬ Eq (Geos (Lin (Li C F)) add Emp) (Geos (Lin (Li C A)) add Emp)" by (simp add:Line_Bet_not_Eq)
from assms P44 P50 have P51 : "¬ Line_on (Li C F) A" by (simp add:Line_not_Eq_on)
have P52 : "Line_on (Li C F) C" by (simp add:Line_on_rule)
from P27 have P53 : "¬ Eq (Geos (Poi E) add Emp) (Geos (Poi C) add Emp)"
by (simp add:Bet_Point_def)
from P28 P36 P52 P53 have P54 : "Line_on (Li C F) E ⟹ Eq (Geos (Lin (Li C G)) add Emp) (Geos (Lin (Li C F)) add Emp)" by (simp add:Line_unique)
from P30 P54 have P55 : "Line_on (Li C F) E ⟹ Line_on (Li C F) G" by (simp add:Line_on_trans)
from P46 P55 have P56 : "¬ Line_on (Li C F) E" by blast
have P57 : "Line_on (Li C F) F" by (simp add:Line_on_rule)
from P34 P39 P46 P51 P56 P57 have P58 : "Line_on_Seg (Li C F) (Se A E) ∧ ¬ Line_on_Seg (Li C F) (Se G E)
∨ Line_on_Seg (Li C F) (Se G E) ∧ ¬ Line_on_Seg (Li C F) (Se A E)" by (simp add:Pachets_axiom)
have "Line_on_Seg (Li C F) (Se G E) ⟹ ∃p. Line_on (Li C F) p ∧ Bet_Point (Se G E) p" by (simp add:Line_on_Seg_rule)
then obtain D2 :: Point where P59 : "Line_on_Seg (Li C F) (Se G E) ⟹ Line_on (Li C F) D2 ∧ Bet_Point (Se G E) D2" by blast
then have P60 : "Line_on_Seg (Li C F) (Se G E) ⟹ Bet_Point (Se G E) D2" by simp
then have P61 : "Line_on_Seg (Li C F) (Se G E) ⟹ Line_on (Li G E) D2" by (simp add:Line_Bet_on)
have P62 : "Line_on (Li G E) G" by (simp add:Line_on_rule)
have P63 : "Line_on (Li G E) E" by (simp add:Line_on_rule)
from P27 have P64 : "¬ Eq (Geos (Poi G) add Emp) (Geos (Poi E) add Emp)"
by (simp add:Bet_Point_def)
from P27 have P66 : "Line_on (Li G E) C" by (simp add:Line_Bet_on)
from P59 have P67 : "Line_on_Seg (Li C F) (Se G E) ⟹ Line_on (Li C F) D2" by simp
from P60 have P68 : "Line_on_Seg (Li C F) (Se G E) ⟹ Eq (Geos (Poi D2) add Emp) (Geos (Poi C) add Emp) ⟹
Bet_Point (Se G E) C" by (simp add:Point_Eq)
from P27 have "Inv (Bet_Point (Se G E) C) ∧ Inv (Bet_Point (Se E C) G)" by (simp add:Bet_iff)
then have P69 : "¬ Bet_Point (Se G E) C ∧ ¬ Bet_Point (Se E C) G" by (simp add:Inv_def)
from P68 P69 have P70 : "Line_on_Seg (Li C F) (Se G E) ⟹ ¬ Eq (Geos (Poi D2) add Emp) (Geos (Poi C) add Emp)" by blast
from P52 P61 P66 P67 P70 have P71 : "Line_on_Seg (Li C F) (Se G E) ⟹ Eq (Geos (Lin (Li G E)) add Emp) (Geos (Lin (Li C F)) add Emp)" by (simp add:Line_unique)
from P63 P71 have P72 : "Line_on_Seg (Li C F) (Se G E) ⟹ Line_on (Li C F) E" by (simp add:Line_on_trans)
from P56 P72 have P73 : "¬ Line_on_Seg (Li C F) (Se G E)" by blast
from P58 P73 have "Line_on_Seg (Li C F) (Se A E) ∧ ¬ Line_on_Seg (Li C F) (Se G E)" by blast
then have "∃p. Line_on (Li C F) p ∧ Bet_Point (Se A E) p" by (simp add:Line_on_Seg_rule)
then obtain D3 :: Point where P74 : "Line_on (Li C F) D3 ∧ Bet_Point (Se A E) D3" by blast
then have P75 : "Line_on (Li C F) D3" by simp
from P74 have P76 : "Bet_Point (Se A E) D3" by simp
then have P77 : "Line_on (Li A E) D3" by (simp add:Line_Bet_on)

from P19 have P78 : ”Line_on (Li C D) F” by simp
from P2 have P79 : ”Eq (Geos (Poi C) add Emp) (Geos (Poi D) add Emp) ⟹ Line_on (Li A C) D” by (simp add:Point_Eq)
from P5 P79 have P80 : ”¬ Eq (Geos (Poi C) add Emp) (Geos (Poi D) add Emp)” by blast
from P44 P78 P80 have P81 : ”Line_on (Li C F) D” by (simp add:Line_on_rev)
from P11 have P82 : ”Line_on (Li A D) E” by simp
from P1 have P83 : ”Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp) ⟹ Line_on (Li A C) D” by (simp add:Point_Eq)
from P5 P83 have P84 : ”¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)” by blast
from P38 P82 P84 have P85 : ”Line_on (Li A E) D” by (simp add:Line_on_rev)
have P86 : ”Line_on (Li A E) E” by (simp add:Line_on_rule)
then have P87 : ”Eq (Geos (Lin (Li A E)) add Emp) (Geos (Lin (Li C F)) add Emp) ⟹ Line_on (Li C F) E” by (simp add:Line_on_trans)
from P56 P87 have P88 : ”¬ Eq (Geos (Lin (Li A E)) add Emp) (Geos (Lin (Li C F)) add Emp)” by blast
from P75 P77 P81 P85 P88 have P89 : ”Eq (Geos (Poi D3) add Emp) (Geos (Poi D) add Emp)” by (simp add:Line_unique_Point)
from P76 P89 have P90 : ”Bet_Point (Se A E) D” by (simp add:Point_Eq)
have P91 : ”Line_on (Li A E) A” by (simp add:Line_on_rule)
from assms P1 P2 P91 have P92 : ”Line_on (Li A E) C ⟹ Eq (Geos (Lin (Li A E)) add Emp) (Geos (Lin (Li A C)) add Emp)” by (simp add:Line_unique)
from P85 P92 have P93 : ”Line_on (Li A E) C ⟹ Line_on (Li A C) D” by (simp add:Line_on_trans)
from P5 P93 have P94 : ”¬ Line_on (Li A E) C” by blast
from assms P1 P4 P20 have P95 : ”Line_on (Li B G) A ⟹ Eq (Geos (Lin (Li B G)) add Emp) (Geos (Lin (Li A C)) add Emp)” by (simp add:Line_unique)
from P24 P95 have P96 : ”Line_on (Li B G) A ⟹ Line_on (Li A C) D” by (simp add:Line_on_trans)
from P5 P96 have P97 : ”¬ Line_on (Li B G) A” by blast
from assms P2 P4 P20 have P98 : ”Line_on (Li B G) C ⟹ Eq (Geos (Lin (Li B G)) add Emp) (Geos (Lin (Li A C)) add Emp)” by (simp add:Line_unique)
from P24 P98 have P99 : ”Line_on (Li B G) C ⟹ Line_on (Li A C) D” by (simp add:Line_on_trans)
from P5 P99 have P100 : ”¬ Line_on (Li B G) C” by blast
from P21 P62 P63 P64 have P101 : ”Line_on (Li B G) E ⟹ Eq (Geos (Lin (Li G E)) add Emp) (Geos (Lin (Li B G)) add Emp)” by (simp add:Line_unique)
from P66 P101 have P102 : ”Line_on (Li B G) E ⟹ Line_on (Li B G) C” by (simp add:Line_on_trans)
from P100 P102 have P103 : ”¬ Line_on (Li B G) E” by blast
from P24 P90 P94 P97 P100 P103 have P104 : ”Line_on_Seg (Li B G) (Se A C) ∧ ¬ Line_on_Seg (Li B G) (Se E C)
∨ Line_on_Seg (Li B G) (Se E C) ∧ ¬ Line_on_Seg (Li B G) (Se A C)” by (simp add:Pachets_axiom)
have ”Line_on_Seg (Li B G) (Se E C) ⟹ ∃p. Line_on (Li B G) p ∧ Bet_Point (Se E C) p” by (simp add:Line_on_Seg_rule)
then obtain B2 :: Point where P105 : ”Line_on_Seg (Li B G) (Se E C) ⟹ Line_on (Li B G) B2 ∧ Bet_Point (Se E C) B2” by blast
then have P106 : ”Line_on_Seg (Li B G) (Se E C) ⟹ Bet_Point (Se E C) B2” by simp
then have P107 : ”Line_on_Seg (Li B G) (Se E C) ⟹ Line_on (Li E C) B2” by (simp add:Line_Bet_on)
from P105 have P108 : ”Line_on_Seg (Li B G) (Se E C) ⟹ Line_on (Li B G) B2” by simp
have P109 : ”Line_on (Li E C) E” by (simp add:Line_on_rule)
have P110 : ”Line_on (Li E C) C” by (simp add:Line_on_rule)
from P28 P36 P53 P109 P110 have P111 : ”Eq (Geos (Lin (Li C G)) add Emp) (Geos (Lin (Li E C)) add Emp)” by (simp add:Line_unique)
from P30 P111 have P112 : ”Line_on (Li E C) G” by (simp add:Line_on_trans)
from P106 have P113 : ”Line_on_Seg (Li B G) (Se E C) ⟹ Eq (Geos (Poi B2) add Emp) (Geos (Poi G) add Emp) ⟹
Bet_Point (Se E C) G” by (simp add:Point_Eq)
from P69 P113 have P114 : ”Line_on_Seg (Li B G) (Se E C) ⟹ ¬ Eq (Geos (Poi B2) add Emp) (Geos (Poi G) add Emp)” by blast
from P21 P107 P108 P112 P114 have P115 : ”Line_on_Seg (Li B G) (Se E C) ⟹ Eq (Geos (Lin (Li E C)) add Emp) (Geos (Lin (Li B G)) add Emp)” by (simp add:Line_unique)
from P109 P115 have P116 : ”Line_on_Seg (Li B G) (Se E C) ⟹ Line_on (Li B G) E” by (simp add:Line_on_trans)
from P103 P116 have P117 : ”¬ Line_on_Seg (Li B G) (Se E C)” by blast
from P104 P117 have ”Line_on_Seg (Li B G) (Se A C)” by blast
then have ”∃p. Line_on (Li B G) p ∧ Bet_Point (Se A C) p” by (simp add:Line_on_Seg_rule)
then obtain B3 :: Point where P118 : ”Line_on (Li B G) B3 ∧ Bet_Point (Se A C) B3” by blast
from P24 have P119 : ”Eq (Geos (Lin (Li B G)) add Emp) (Geos (Lin (Li A C)) add Emp) ⟹ Line_on (Li A C) D” by (simp add:Line_on_trans)
from P5 P119 have P120 : ”¬ Eq (Geos (Lin (Li B G)) add Emp) (Geos (Lin (Li A C)) add Emp)” by

```
blast
  from P118 have P121 : "Line_on (Li B G) B3" by simp
  from P118 have P122 : "Bet_Point (Se A C) B3" by simp
  then have P123 : "Line_on (Li A C) B3" by (simp add:Line_Bet_on)
  from P4 P20 P120 P121 P123 have P124 : "Eq (Geos (Poi B3) add Emp) (Geos (Poi B) add Emp)"
by (simp add:Line_unique_Point)
  from P122 P124 show "Bet_Point (Se A C) B" by (simp add:Point_Eq)
qed

lemma(in Order_Rule) Bet_case :
  assumes
    "Line_on l1 A"
    "Line_on l1 B"
    "Line_on l1 C"
    "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)"
    "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)"
    "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)"
  shows "Bet_Point (Se A C) B ∨ Bet_Point (Se C B) A ∨ Bet_Point (Se B A) C"
proof -
  from assms have P1 : "¬ Bet_Point (Se B A) C ∧ ¬ Bet_Point (Se C B) A ⟹ Bet_Point (Se A C) B"
by (simp add:Bet_case_lemma2)
  from assms have P2 : "¬ Bet_Point (Se C B) A ∧ ¬ Bet_Point (Se A C) B ⟹ Bet_Point (Se B A) C"
by (simp add:Bet_case_lemma2)
  from assms have P3 : "¬ Bet_Point (Se A C) B ∧ ¬ Bet_Point (Se B A) C ⟹ Bet_Point (Se C B) A"
by (simp add:Bet_case_lemma2)
  from P1 P2 P3 show "Bet_Point (Se A C) B ∨ Bet_Point (Se C B) A ∨ Bet_Point (Se B A) C" by blast
qed

lemma(in Order_Rule) Bet_case_fact :
  assumes
    "Bet_Point (Se A C) B ∨ Bet_Point (Se C B) A ∨ Bet_Point (Se B A) C"
  shows
    "Bet_Point (Se A C) B ∧ ¬ Bet_Point (Se C B) A ∧ ¬ Bet_Point (Se B A) C
    ∨ ¬ Bet_Point (Se A C) B ∧ Bet_Point (Se C B) A ∧ ¬ Bet_Point (Se B A) C
    ∨ ¬ Bet_Point (Se A C) B ∧ ¬ Bet_Point (Se C B) A ∧ Bet_Point (Se B A) C"
proof -
  have "Bet_Point (Se A C) B ⟹ Inv (Bet_Point (Se C B) A) ∧ Inv (Bet_Point (Se B A) C)" by (simp
add:Bet_iff)
  then have P1 : "Bet_Point (Se A C) B ⟹ ¬ Bet_Point (Se C B) A ∧ ¬ Bet_Point (Se B A) C" by
(simp add:Inv_def)
  have "Bet_Point (Se C B) A ⟹ Inv (Bet_Point (Se A C) B) ∧ Inv (Bet_Point (Se B A) C)" by (simp
add:Bet_iff)
  then have P2 : "Bet_Point (Se C B) A ⟹ ¬ Bet_Point (Se A C) B ∧ ¬ Bet_Point (Se B A) C" by
(simp add:Inv_def)
  have "Bet_Point (Se B A) C ⟹ Inv (Bet_Point (Se A C) B) ∧ Inv (Bet_Point (Se C B) A)" by (simp
add:Bet_iff)
  then have P3 : "Bet_Point (Se B A) C ⟹ ¬ Bet_Point (Se A C) B ∧ ¬ Bet_Point (Se C B) A" by
(simp add:Inv_def)
  from assms P1 P2 P3 show "Bet_Point (Se A C) B ∧ ¬ Bet_Point (Se C B) A ∧ ¬ Bet_Point (Se B A)
C
    ∨ ¬ Bet_Point (Se A C) B ∧ Bet_Point (Se C B) A ∧ ¬ Bet_Point (Se B A) C
    ∨ ¬ Bet_Point (Se A C) B ∧ ¬ Bet_Point (Se C B) A ∧ Bet_Point (Se B A) C" by blast
qed

lemma(in Order_Rule) Bet_swap_lemma_1 :
  assumes
    "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)"
    "Bet_Point (Se A C) B"
    "Bet_Point (Se B D) C"
  shows "Line_on (Li A D) B ∧ Line_on (Li A D) C"
proof -
  from assms have P1 : "Line_on (Li A B) C" by (simp add:Line_Bet_on)
  have P2 : "Line_on (Li A B) B" by (simp add:Line_on_rule)
  have P3 : "Line_on (Li B C) C" by (simp add:Line_on_rule)
  have P4 : "Line_on (Li B C) B" by (simp add:Line_on_rule)
  from assms have P5 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi B) add Emp)"
```

```
    by (simp add:Bet_Point_def)
  from P1 P2 P3 P4 P5 have P6 : "Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li A B)) add Emp)"
by (simp add:Line_unique)
  from assms have P7 : "Line_on (Li B C) D" by (simp add:Line_Bet_on)
  from P6 P7 have P8 : "Line_on (Li A B) D" by (simp add:Line_on_trans)
  from assms have "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp)" by (simp add:Bet_Point_def)
  then have P9 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)" by (blast intro:Eq_rev)
  from assms P8 P9 have P10 : "Line_on (Li A D) B" by (simp add:Line_on_rev)
  have P11 : "Line_on (Li A D) D" by (simp add:Line_on_rule)
  from assms have P12 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)"
    by (simp add:Bet_Point_def)
  from P4 P7 P10 P11 P12 have P13 : "Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li A D)) add
Emp)" by (simp add:Line_unique)
  from P3 P13 have P14 : "Line_on (Li A D) C" by (simp add:Line_on_trans)
  from P10 P14 show "Line_on (Li A D) B ∧ Line_on (Li A D) C" by simp
qed

lemma(in Order_Rule) Bet_swap_lemma_2 :
  assumes
    "Bet_Point (Se p1 p3) p2"
    "¬ Line_on (Li p1 p3) p4"
    "¬ Line_on (Li p2 p5) p3"
    "¬ Line_on (Li p2 p5) p1"
    "¬ Line_on (Li p2 p5) p4"
    "Bet_Point (Se p3 p5) p4"
  shows "∃p. Line_on (Li p2 p5) p ∧ Bet_Point (Se p1 p4) p"
proof -
  have P1 : "Line_on (Li p2 p5) p2" by (simp add:Line_on_rule)
  from assms P1 have P2 : "Line_on_Seg (Li p2 p5) (Se p1 p4) ∧ ¬ Line_on_Seg (Li p2 p5) (Se p3 p4) ∨
Line_on_Seg (Li p2 p5) (Se p3 p4) ∧ ¬ Line_on_Seg (Li p2 p5) (Se p1 p4)" by (simp add:Pachets_axiom)
  then have "Line_on_Seg (Li p2 p5) (Se p3 p4) ⟹ ∃p. Line_on (Li p2 p5) p ∧ Bet_Point (Se p3 p4)
p" by (simp add:Line_on_Seg_rule)
  then obtain p6 :: Point where P3 : "Line_on_Seg (Li p2 p5) (Se p3 p4) ⟹ Line_on (Li p2 p5) p6 ∧
Bet_Point (Se p3 p4) p6" by blast
  from assms have "¬ Eq (Geos (Poi p4) add Emp) (Geos (Poi p3) add Emp)" by (simp add:Bet_Point_def)
  then have P4 : "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p4) add Emp)" by (blast intro:Eq_rev)
  from P3 have P5 : "Line_on_Seg (Li p2 p5) (Se p3 p4) ⟹ Bet_Point (Se p3 p4) p6" by simp
  from P3 have P6 : "Line_on_Seg (Li p2 p5) (Se p3 p4) ⟹ Line_on (Li p3 p6) p4" by (simp
add:Line_Bet_on)
  from assms have P7 : "Line_on (Li p3 p5) p4" by (simp add:Line_Bet_on)
  have P8 : "Line_on (Li p3 p6) p3" by (simp add:Line_on_rule)
  have P9 : "Line_on (Li p3 p5) p3" by (simp add:Line_on_rule)
  from P4 P6 P7 P8 P9 have P10 : "Line_on_Seg (Li p2 p5) (Se p3 p4) ⟹
    Eq (Geos (Lin (Li p3 p5)) add Emp) (Geos (Lin (Li p3 p6)) add Emp)" by (simp add:Line_unique)
  have P11 : "Line_on (Li p3 p5) p5" by (simp add:Line_on_rule)
  from P10 P11 have P12 : "Line_on_Seg (Li p2 p5) (Se p3 p4) ⟹ Line_on (Li p3 p6) p5" by (simp
add:Line_on_trans)
  have P13 : "Line_on (Li p2 p5) p5" by (simp add:Line_on_rule)
  have P14 : "Line_on (Li p3 p6) p6" by (simp add:Line_on_rule)
  from P5 have P15 : "Line_on_Seg (Li p2 p5) (Se p3 p4) ⟹ Eq (Geos (Poi p6) add Emp) (Geos (Poi
p5) add Emp) ⟹
    Bet_Point (Se p3 p4) p5" by (simp add:Point_Eq)
  from assms have "Inv (Bet_Point (Se p5 p4) p3) ∧ Inv (Bet_Point (Se p4 p3) p5)" by (simp add:Bet_iff)
  then have "¬ Bet_Point (Se p4 p3) p5" by (simp add:Inv_def)
  then have P16 : "¬ Bet_Point (Se p3 p4) p5" by (blast intro:Bet_rev)
  from P15 P16 have P17 : "Line_on_Seg (Li p2 p5) (Se p3 p4) ⟹ ¬ Eq (Geos (Poi p6) add Emp)
(Geos (Poi p5) add Emp)" by blast
  from P3 P12 P13 P14 P17 have P18 : "Line_on_Seg (Li p2 p5) (Se p3 p4) ⟹
    Eq (Geos (Lin (Li p3 p6)) add Emp) (Geos (Lin (Li p2 p5)) add Emp)" by (simp add:Line_unique)
  from P8 P18 have P19 : "Line_on_Seg (Li p2 p5) (Se p3 p4) ⟹ Line_on (Li p2 p5) p3" by (simp
add:Line_on_trans)
  from assms P19 have P20 : "¬ Line_on_Seg (Li p2 p5) (Se p3 p4)" by blast
  from P2 P3 P20 have "Line_on_Seg (Li p2 p5) (Se p1 p4)" by blast
  thus "∃p. Line_on (Li p2 p5) p ∧ Bet_Point (Se p1 p4) p" by (simp add:Line_on_Seg_rule)
qed
```

```
lemma(in Order_Rule) Bet_swap_lemma_3 :
  assumes
    "Bet_Point (Se p1 p3) p2"
    "Bet_Point (Se p3 p5) p4"
    "¬ Line_on (Li p1 p3) p5"
  shows "∃p. Bet_Point (Se p1 p4) p ∧ Bet_Point (Se p5 p2) p"
proof -
  from assms have P1 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)"
    by (simp add:Bet_Point_def)
  then have P2 : "Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin (Li p3 p1)) add Emp)" by (simp
add:Line_rev)
  from assms P2 have P3 : "¬ Line_on (Li p3 p1) p5" by (simp add:Line_not_on_trans)
  from assms have P4 : "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p5) add Emp)"
    by (simp add:Bet_Point_def)
  from P1 have P5 : "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p1) add Emp)" by (blast intro:Eq_rev)
  from P3 P4 P5 have P6 : "¬ Line_on (Li p3 p5) p1" by (blast intro:Line_on_rev)
  from assms have P7 : "Bet_Point (Se p5 p3) p4" by (simp add:Bet_rev)
  from P4 have P8 : "Eq (Geos (Lin (Li p3 p5)) add Emp) (Geos (Lin (Li p5 p3)) add Emp)" by (simp
add:Line_rev)
  from P8 P6 have P9 : "¬ Line_on (Li p5 p3) p1" by (simp add:Line_not_on_trans)
  from P7 P9 have P10 : "¬ Eq (Geos (Lin (Li p1 p4)) add Emp) (Geos (Lin (Li p1 p3)) add Emp)" by
(simp add:Line_Bet_not_Eq)
  from assms have "Line_on (Li p3 p5) p4" by (simp add:Line_Bet_on)
  then have P11 : "Eq (Geos (Poi p4) add Emp) (Geos (Poi p1) add Emp) ⟹ Line_on (Li p3 p5) p1"
by (simp add:Point_Eq)
  from P6 P11 have "¬ Eq (Geos (Poi p4) add Emp) (Geos (Poi p1) add Emp)" by blast
  then have P12 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p4) add Emp)" by (blast intro:Eq_rev)
  from P1 P10 P12 have P13 : "¬ Line_on (Li p1 p4) p3" by (simp add:Line_not_Eq_on)
  from P1 P12 P13 have P14 : "¬ Line_on (Li p1 p3) p4" by (blast intro:Line_on_rev)
  from assms have P15 : "¬ Eq (Geos (Lin (Li p5 p2)) add Emp) (Geos (Lin (Li p5 p3)) add Emp)" by
(simp add:Line_Bet_not_Eq)
  from assms have P16 : "Line_on (Li p1 p3) p2" by (simp add:Line_Bet_on)
  then have P17 : "Eq (Geos (Poi p2) add Emp) (Geos (Poi p5) add Emp) ⟹ Line_on (Li p1 p3) p5"
by (simp add:Point_Eq)
  from assms P17 have "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p5) add Emp)" by blast
  then have P18 : "¬ Eq (Geos (Poi p5) add Emp) (Geos (Poi p2) add Emp)" by (blast intro:Eq_rev)
  from P4 have P19 : "¬ Eq (Geos (Poi p5) add Emp) (Geos (Poi p3) add Emp)" by (blast intro:Eq_rev)
  from P15 P18 P19 have P20 : "¬ Line_on (Li p5 p2) p3" by (simp add:Line_not_Eq_on)
  from P18 have P21 : "Eq (Geos (Lin (Li p5 p2)) add Emp) (Geos (Lin (Li p2 p5)) add Emp)" by (simp
add:Line_rev)
  from P20 P21 have P22 : "¬ Line_on (Li p2 p5) p3" by (simp add:Line_not_on_trans)
  from assms have P23 :"Bet_Point (Se p3 p1) p2" by (blast intro:Bet_rev)
  from P3 P23 have P24 : "¬ Eq (Geos (Lin (Li p5 p2)) add Emp) (Geos (Lin (Li p5 p1)) add Emp)" by
(simp add:Line_Bet_not_Eq)
  have "Line_on (Li p3 p1) p1" by (simp add:Line_on_rule)
  then have P25 : "Eq (Geos (Poi p1) add Emp) (Geos (Poi p5) add Emp) ⟹ Line_on (Li p3 p1) p5"
by (simp add:Point_Eq)
  from P3 P25 have P26 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p5) add Emp)" by blast
  then have P27 : "¬ Eq (Geos (Poi p5) add Emp) (Geos (Poi p1) add Emp)" by (blast intro:Eq_rev)
  from P18 P24 P27 have P28 : "¬ Line_on (Li p5 p2) p1" by (simp add:Line_not_Eq_on)
  from P21 P28 have P29 : "¬ Line_on (Li p2 p5) p1" by (simp add:Line_not_on_trans)
  from assms have P31 : "Line_on (Li p3 p4) p5" by (simp add:Line_Bet_on)
  have P32 : "Line_on (Li p3 p4) p4" by (simp add:Line_on_rule)
  have P33 : "Line_on (Li p2 p5) p5" by (simp add:Line_on_rule)
  from assms have P34 : "¬ Eq (Geos (Poi p5) add Emp) (Geos (Poi p4) add Emp)"
    by (simp add:Bet_Point_def)
  from P31 P32 P33 P34 have P35 : "Line_on (Li p2 p5) p4 ⟹ Eq (Geos (Lin (Li p3 p4)) add Emp)
(Geos (Lin (Li p2 p5)) add Emp)" by (simp add:Line_unique)
  have P36 : "Line_on (Li p3 p4) p3" by (simp add:Line_on_rule)
  from P35 P36 have P37 : "Line_on (Li p2 p5) p4 ⟹ Line_on (Li p2 p5) p3" by (simp add:Line_on_trans)
  from P22 P37 have P38 : "¬ Line_on (Li p2 p5) p4" by blast
  from assms P14 P22 P29 P38 have "∃p. Line_on (Li p2 p5) p ∧ Bet_Point (Se p1 p4) p" by (simp
add:Bet_swap_lemma_2)
  then obtain p6 :: Point where P39 : "Line_on (Li p2 p5) p6 ∧ Bet_Point (Se p1 p4) p6" by blast
  from P12 have P40 : "Eq (Geos (Lin (Li p1 p4)) add Emp) (Geos (Lin (Li p4 p1)) add Emp)" by (simp
add:Line_rev)
```

```
  from P13 P40 have P41 : "¬ Line_on (Li p4 p1) p3" by (simp add:Line_not_on_trans)
  from assms P6 have P42 : "¬ Eq (Geos (Lin (Li p1 p4)) add Emp) (Geos (Lin (Li p1 p5)) add Emp)"
by (simp add:Line_Bet_not_Eq)
  from P12 P26 P42 have P43 : "¬ Line_on (Li p1 p4) p5" by (simp add:Line_not_Eq_on)
  from P40 P43 have P44 : "¬ Line_on (Li p4 p1) p5" by (simp add:Line_not_on_trans)
  from assms have "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p1) add Emp)" by (simp add:Bet_Point_def)
  then have P45 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)" by (blast intro:Eq_rev)
  from P1 P16 P45 have P47 : "Line_on (Li p1 p2) p3" by (simp add:Line_on_rev)
  from P47 have P48 : "Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p1 p4)) add Emp) ⟹
Line_on (Li p1 p4) p3" by (simp add:Line_on_trans)
  from P13 P48 have P49 : "¬ Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p1 p4)) add Emp)"
by blast
  from P12 P45 P49 have P50 : "¬ Line_on (Li p1 p2) p4" by (simp add:Line_not_Eq_on)
  from P12 P45 P50 have P51 : "¬ Line_on (Li p1 p4) p2" by (blast intro:Line_on_rev)
  from P40 P51 have P52 : "¬ Line_on (Li p4 p1) p2" by (simp add:Line_not_on_trans)
  from P18 P19 P20 have P53 : "¬ Line_on (Li p5 p3) p2" by (blast intro:Line_on_rev)
  from P7 P23 P41 P44 P52 P53 have "∃p. Line_on (Li p4 p1) p ∧ Bet_Point (Se p5 p2) p" by (simp
add:Bet_swap_lemma_2)
  then obtain p7 :: Point where P54 : "Line_on (Li p4 p1) p7 ∧ Bet_Point (Se p5 p2) p7" by blast
  from P33 P44 have P55 : "¬ Eq (Geos (Lin (Li p4 p1)) add Emp) (Geos (Lin (Li p2 p5)) add Emp)"
by (simp add:Line_not_on_Eq)
  from P39 have P56 : "Line_on (Li p4 p1) p6" by (simp add:Line_Bet_on)
  from P54 have P57 : "Line_on (Li p2 p5) p7" by (simp add:Line_Bet_on)
  from P39 P54 P55 P56 P57 have P58 : "Eq (Geos (Poi p7) add Emp) (Geos (Poi p6) add Emp)" by
(blast intro:Line_unique_Point)
  from P54 have P59 : "Bet_Point (Se p5 p2) p7" by simp
  from P58 P59 have P60 : "Bet_Point (Se p5 p2) p6" by (simp add:Point_Eq)
  from P39 P60 show "∃p. Bet_Point (Se p1 p4) p ∧ Bet_Point (Se p5 p2) p" by blast
qed

lemma(in Order_Rule) Bet_swap_lemma_4 :
  assumes
    "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)"
    "Bet_Point (Se A E) G"
    "Bet_Point (Se D G) H"
    "¬ Line_on (Li A D) E"
  shows "∃p. Line_on (Li H E) p ∧ Bet_Point (Se D A) p"
proof -
  from assms have P1 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi E) add Emp)"
    by (simp add:Bet_Point_def)
  from assms P1 have P2 : "¬ Line_on (Li A E) D" by (blast intro:Line_on_rev)
  from P1 have P3 : "Eq (Geos (Lin (Li A E)) add Emp) (Geos (Lin (Li E A)) add Emp)" by (simp
add:Line_rev)
  from P2 P3 have P4 : "¬ Line_on (Li E A) D" by (simp add:Line_not_on_trans)
  from assms have P5 : "Bet_Point (Se E A) G" by (simp add:Bet_rev)
  from P4 P5 have P6 : "¬ Eq (Geos (Lin (Li D G)) add Emp) (Geos (Lin (Li D A)) add Emp)" by
(simp add:Line_Bet_not_Eq)
  from assms have P7 : "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi G) add Emp)"
    by (simp add:Bet_Point_def)
  from assms have P8 : "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi A) add Emp)" by (blast intro:Eq_rev)
  from P6 P7 P8 have P9 : "¬ Line_on (Li D G) A" by (simp add:Line_not_Eq_on)
  from assms P2 have P10 : "¬ Eq (Geos (Lin (Li D G)) add Emp) (Geos (Lin (Li D E)) add Emp)" by
(simp add:Line_Bet_not_Eq)
  have "Line_on (Li A D) D" by (simp add:Line_on_rule)
  then have P11 : "Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li A D) E" by
(simp add:Point_Eq)
  from assms P11 have P12 : "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp)" by blast
  from P7 P10 P12 have P13 : "¬ Line_on (Li D G) E" by (simp add:Line_not_Eq_on)
  from assms P13 have P14 : "¬ Eq (Geos (Lin (Li E H)) add Emp) (Geos (Lin (Li E G)) add Emp)" by
(simp add:Line_Bet_not_Eq)
  from assms have "Line_on (Li D G) H" by (simp add:Line_Bet_on)
  then have P15 : "Eq (Geos (Poi H) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li D G) E" by
(simp add:Point_Eq)
  from P13 P15 have P16 : "¬ Eq (Geos (Poi E) add Emp) (Geos (Poi H) add Emp)" by (blast in-
tro:Eq_rev)
  from assms have "Line_on (Li D G) G" by (simp add:Line_on_rule)
```

then have P17 : ”Eq (Geos (Poi G) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li D G) E” by (simp add:Point_Eq)
from P13 P17 have P18 : ”¬ Eq (Geos (Poi E) add Emp) (Geos (Poi G) add Emp)” by (blast intro:Eq_rev)
from P14 P16 P18 have P19 : ”¬ Line_on (Li E H) G” by (simp add:Line_not_Eq_on)
from P7 have P20 : ”Eq (Geos (Lin (Li D G)) add Emp) (Geos (Lin (Li G D)) add Emp)” by (simp add:Line_rev)
from P13 P20 have P21 : ”¬ Line_on (Li G D) E” by (simp add:Line_not_on_trans)
from assms have P22 : ”Bet_Point (Se G D) H” by (simp add:Bet_rev)
from P21 P22 have P23 : ”¬ Eq (Geos (Lin (Li E H)) add Emp) (Geos (Lin (Li E D)) add Emp)” by (simp add:Line_Bet_not_Eq)
from P12 have P24 : ”¬ Eq (Geos (Poi E) add Emp) (Geos (Poi D) add Emp)” by (blast intro:Eq_rev)
from P16 P23 P24 have P25 : ”¬ Line_on (Li E H) D” by (simp add:Line_not_Eq_on)
from P16 have P26 : ”Eq (Geos (Lin (Li E H)) add Emp) (Geos (Lin (Li H E)) add Emp)” by (simp add:Line_rev)
from P25 P26 have P27 : ”¬ Line_on (Li H E) D” by (simp add:Line_not_on_trans)
have P28 : ”Line_on (Li A E) A” by (simp add:Line_on_rule)
have P29 : ”Line_on (Li A E) E” by (simp add:Line_on_rule)
have P30 : ”Line_on (Li E H) E” by (simp add:Line_on_rule)
from P1 P28 P29 P30 have P31 : ”Line_on (Li E H) A ⟹ Eq (Geos (Lin (Li A E)) add Emp) (Geos (Lin (Li E H)) add Emp)” by (simp add:Line_unique)
from assms have P32 : ”Line_on (Li A E) G” by (simp add:Line_Bet_on)
from assms have ”¬ Eq (Geos (Poi G) add Emp) (Geos (Poi A) add Emp)” by (simp add:Bet_Point_def)
then have P33 : ”¬ Eq (Geos (Poi A) add Emp) (Geos (Poi G) add Emp)” by (blast intro:Eq_rev)
from P31 P32 have P34 : ”Line_on (Li E H) A ⟹ Line_on (Li E H) G” by (simp add:Line_on_trans)
from P19 P34 have P35 : ”¬ Line_on (Li E H) A” by blast
from P26 P35 have P36 : ”¬ Line_on (Li H E) A” by (simp add:Line_not_on_trans)
from P26 P19 have P37 : ”¬ Line_on (Li H E) G” by (simp add:Line_not_on_trans)
have P38 : ”Line_on (Li H E) H” by (simp add:Line_on_rule)
from assms P9 P27 P36 P37 P38 have P39 : ”Line_on_Seg (Li H E) (Se D A) ∧ ¬ Line_on_Seg (Li H E) (Se G A) ∨ Line_on_Seg (Li H E) (Se G A) ∧ ¬ Line_on_Seg (Li H E) (Se D A)” by (simp add:Pachets_axiom)
then have ”Line_on_Seg (Li H E) (Se G A) ⟹ ∃p. Line_on (Li H E) p ∧ Bet_Point (Se G A) p” by (simp add:Line_on_Seg_rule)
then obtain C2 :: Point where P40 : ”Line_on_Seg (Li H E) (Se G A) ⟹ Line_on (Li H E) C2 ∧ Bet_Point (Se G A) C2” by blast
from assms have P41 : ”Line_on (Li G A) E” by (simp add:Line_Bet_on)
from P40 have P42 : ”Line_on_Seg (Li H E) (Se G A) ⟹ Line_on (Li G A) C2”
by (simp add:Line_Bet_on)
have P43 : ”Line_on (Li H E) E” by (simp add:Line_on_rule)
from P40 have ”Line_on_Seg (Li H E) (Se G A) ⟹ Bet_Point (Se G A) C2” by simp
then have P44 : ”Line_on_Seg (Li H E) (Se G A) ⟹ Eq (Geos (Poi C2) add Emp) (Geos (Poi E) add Emp) ⟹
Bet_Point (Se G A) E” by (simp add:Point_Eq)
from assms have ”Inv (Bet_Point (Se E G) A) ∧ Inv (Bet_Point (Se G A) E)” by (simp add:Bet_iff)
then have P45 : ”¬ Bet_Point (Se G A) E” by (simp add:Inv_def)
from P44 P45 have P46 : ”Line_on_Seg (Li H E) (Se G A) ⟹ ¬ Eq (Geos (Poi C2) add Emp) (Geos (Poi E) add Emp)” by blast
from P40 P41 P42 P43 P46 have P47 : ”Line_on_Seg (Li H E) (Se G A) ⟹
Eq (Geos (Lin (Li G A)) add Emp) (Geos (Lin (Li H E)) add Emp)” by (simp add:Line_unique)
have P48 : ”Line_on (Li G A) G” by (simp add:Line_on_rule)
from P47 P48 have P49 : ”Line_on_Seg (Li H E) (Se G A) ⟹ Line_on (Li H E) G” by (simp add:Line_on_trans)
from P37 P49 have P50 : ”¬ Line_on_Seg (Li H E) (Se G A)” by blast
from P39 P40 P50 have ”Line_on_Seg (Li H E) (Se D A)” by blast
thus ”∃p. Line_on (Li H E) p ∧ Bet_Point (Se D A) p” by (simp add:Line_on_Seg_rule)
qed

lemma(in Order_Rule) Bet_swap_lemma_5 :
assumes
”Bet_Point (Se A C) B”
”Bet_Point (Se B D) C”
”Bet_Point (Se C F) E”
”¬ Line_on (Li A D) F”
”¬ Line_on (Li A C) F”
shows ”Bet_Point (Se A D) C”

```
proof -
  from assms have P1 : "Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp) ⟹ Bet_Point (Se D C) B" by (simp add:Bet_Point_Eq)
  from assms have "Inv (Bet_Point (Se D C) B) ∧ Inv (Bet_Point (Se C B) D)" by (simp add:Bet_iff)
  then have P2 : "¬ Bet_Point (Se D C) B" by (simp add:Inv_def)
  from P1 P2 have P3 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)" by blast
  from assms P3 have P4 : "Line_on (Li A D) B ∧ Line_on (Li A D) C" by (simp add:Bet_swap_lemma_1)
  then have P5 : "Line_on (Li A D) C" by simp
  from assms have "∃p. Bet_Point (Se A E) p ∧ Bet_Point (Se F B) p" by (simp add:Bet_swap_lemma_3)
  then obtain G :: Point where P6 : "Bet_Point (Se A E) G ∧ Bet_Point (Se F B) G" by blast
  from P3 have P7 : "Eq (Geos (Lin (Li A D)) add Emp) (Geos (Lin (Li D A)) add Emp)" by (simp add:Line_rev)
  from P4 P7 have P8 : "Line_on (Li D A) B" by (blast intro:Line_on_trans)
  from assms P7 have P9 : "¬ Line_on (Li D A) F" by (simp add:Line_not_on_trans)
  have P10 : "Line_on (Li D A) D" by (simp add:Line_on_rule)
  have P11 : "Line_on (Li D B) D" by (simp add:Line_on_rule)
  have P12 : "Line_on (Li D B) B" by (simp add:Line_on_rule)
  from assms have P13 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)"
    by (simp add:Bet_Point_def)
  from P8 P10 P11 P12 P13 have P14 : "Eq (Geos (Lin (Li D A)) add Emp) (Geos (Lin (Li D B)) add Emp)" by (simp add:Line_unique)
  from P9 P14 have P15 : "¬ Line_on (Li D B) F" by (simp add:Line_not_on_trans)
  from assms have P16 : "Bet_Point (Se D B) C" by (simp add:Bet_rev)
  from P6 have P17 : "Bet_Point (Se B F) G" by (simp add:Bet_rev)
  from P15 P16 P17 have "∃p. Bet_Point (Se D G) p ∧ Bet_Point (Se F C) p"
    by (simp add:Bet_swap_lemma_3)
  then obtain H :: Point where P18 : "Bet_Point (Se D G) H ∧ Bet_Point (Se F C) H" by blast
  from assms have P19 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)"
    by (simp add:Bet_Point_def)
  then have P20 : "Eq (Geos (Lin (Li A C)) add Emp) (Geos (Lin (Li C A)) add Emp)" by (simp add:Line_rev)
  from assms P20 have P21 : "¬ Line_on (Li C A) F" by (simp add:Line_not_on_trans)
  from P19 have P22 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)" by (blast intro:Eq_rev)
  from assms have P23 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi F) add Emp)"
    by (simp add:Bet_Point_def)
  from P21 P22 P23 have P24 : "¬ Line_on (Li C F) A" by (blast intro:Line_on_rev)
  from assms have P25 : "Bet_Point (Se F C) E" by (simp add:Bet_rev)
  from P23 have P26 : "Eq (Geos (Lin (Li C F)) add Emp) (Geos (Lin (Li F C)) add Emp)" by (simp add:Line_rev)
  from P24 P26 have P27 : "¬ Line_on (Li F C) A" by (simp add:Line_not_on_trans)
  from P25 P27 have P28 : "¬ Eq (Geos (Lin (Li A E)) add Emp) (Geos (Lin (Li A C)) add Emp)" by (simp add:Line_Bet_not_Eq)
  from P25 have "Line_on (Li F C) E" by (simp add:Line_Bet_on)
  then have P29 : "Eq (Geos (Poi E) add Emp) (Geos (Poi A) add Emp) ⟹ Line_on (Li F C) A" by (simp add:Point_Eq)
  from P27 P29 have "¬ Eq (Geos (Poi E) add Emp) (Geos (Poi A) add Emp)" by blast
  then have P30 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi E) add Emp)" by (blast intro:Eq_rev)
  from P19 P28 P30 have P31 : "¬ Line_on (Li A E) C" by (simp add:Line_not_Eq_on)
  from P19 P30 P31 have P32 : "¬ Line_on (Li A C) E" by (blast intro:Line_on_rev)
  from P5 P19 P32 have P33 : "¬ Line_on (Li A D) E" by (simp add:Line_not_on_ex)
  from P3 P30 P33 have P34 : "¬ Line_on (Li A E) D" by (blast intro:Line_on_rev)
  from P30 have P35 : "Eq (Geos (Lin (Li A E)) add Emp) (Geos (Lin (Li E A)) add Emp)" by (simp add:Line_rev)
  from P18 have P36 : "Bet_Point (Se D G) H" by simp
  from P6 have P37 : "Bet_Point (Se A E) G" by simp
  from P3 P18 P33 P37 have "∃p. Line_on (Li H E) p ∧ Bet_Point (Se D A) p"
    by (simp add:Bet_swap_lemma_4)
  then obtain C2 :: Point where P38 : "Line_on (Li H E) C2 ∧ Bet_Point (Se D A) C2" by blast
  have "Line_on (Li H E) E" by (simp add:Line_on_rule)
  then have P39 : "Eq (Geos (Lin (Li H E)) add Emp) (Geos (Lin (Li A D)) add Emp) ⟹ Line_on (Li A D) E" by (simp add:Line_on_trans)
  from P33 P39 have P40 : "¬ Eq (Geos (Lin (Li H E)) add Emp) (Geos (Lin (Li A D)) add Emp)" by blast
  from P23 have P41 : "¬ Eq (Geos (Poi F) add Emp) (Geos (Poi C) add Emp)" by (blast intro:Eq_rev)
  from P25 have P42 : "Line_on (Li F E) C" by (simp add:Line_Bet_on)
  from P18 have P43 : "Line_on (Li F H) C" by (simp add:Line_Bet_on)
```

```
  from P36 have P44 : "Eq (Geos (Poi H) add Emp) (Geos (Poi E) add Emp) ⟹ Bet_Point (Se D G)
E" by (simp add:Point_Eq)
  then have P45 : "Eq (Geos (Poi H) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li D G) E" by
(simp add:Line_Bet_on)
  have P46 : "Line_on (Li D G) G" by (simp add:Line_on_rule)
  have P47 : "Line_on (Li A E) E" by (simp add:Line_on_rule)
  from P37 have P48 : "Line_on (Li A E) G" by (simp add:Line_Bet_on)
  from P44 have P49 : "Eq (Geos (Poi H) add Emp) (Geos (Poi E) add Emp) ⟹ ¬ Eq (Geos (Poi G)
add Emp) (Geos (Poi E) add Emp)" by (simp add:Bet_Point_def)
  from P45 P46 P47 P48 P49 have P50 : "Eq (Geos (Poi H) add Emp) (Geos (Poi E) add Emp) ⟹
    Eq (Geos (Lin (Li D G)) add Emp) (Geos (Lin (Li A E)) add Emp)" by (simp add:Line_unique)
  have P51 : "Line_on (Li D G) D" by (simp add:Line_on_rule)
  from P50 P51 have P52 : "Eq (Geos (Poi H) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li A E)
D" by (simp add:Line_on_trans)
  from P34 P52 have P53 : "¬ Eq (Geos (Poi E) add Emp) (Geos (Poi H) add Emp)" by (blast in-
tro:Eq_rev)
  from P41 P42 P43 P53 have P54 : "Line_on (Li E H) C" by (blast intro:Line_on_dens)
  from P53 have P55 : "Eq (Geos (Lin (Li E H)) add Emp) (Geos (Lin (Li H E)) add Emp)" by (simp
add:Line_rev)
  from P54 P55 have P56 : "Line_on (Li H E) C" by (blast intro:Line_on_trans)
  from P38 have P57 : "Line_on (Li A D) C2" by (simp add:Line_Bet_on)
  from P5 P38 P40 P56 P57 have P58 : "Eq (Geos (Poi C2) add Emp) (Geos (Poi C) add Emp)" by
(blast intro:Line_unique_Point)
  from P38 have P59 : "Bet_Point (Se D A) C2" by simp
  from P58 P59 have "Bet_Point (Se D A) C" by (simp add:Point_Eq)
  thus "Bet_Point (Se A D) C" by (simp add:Bet_rev)
qed

theorem(in Order_Rule) Bet_swap_234_134 :
  assumes
    "Bet_Point (Se A C) B"
    "Bet_Point (Se B D) C"
  shows "Bet_Point (Se A D) C"
proof -
  from assms have P1 : "Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp) ⟹ Bet_Point (Se D C)
B" by (simp add:Bet_Point_Eq)
  from assms have "Inv (Bet_Point (Se D C) B) ∧ Inv (Bet_Point (Se C B) D)" by (simp add:Bet_iff)
  then have P2 : "¬ Bet_Point (Se D C) B" by (simp add:Inv_def)
  from P1 P2 have P3 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)" by blast
  from assms P3 have "Line_on (Li A D) B ∧ Line_on (Li A D) C" by (simp add:Bet_swap_lemma_1)
  then have P4 : "Line_on (Li A D) C" by simp
  have "∃p q r. ¬ Line_on (Li A D) p ∧ ¬ Line_on (Li A D) q ∧ ¬ Line_on (Li A D) r
          ∧ ¬ Eq (Geos (Poi p) add Emp) (Geos (Poi q) add Emp) ∧ ¬ Eq (Geos (Poi q) add Emp) (Geos
(Poi r) add Emp)
          ∧ ¬ Eq (Geos (Poi r) add Emp) (Geos (Poi p) add Emp)" by (blast intro:Line_not_on_exist)
  then obtain F :: Point where P5 : "¬ Line_on (Li A D) F" by blast
  from P4 have P6 : "Eq (Geos (Poi C) add Emp) (Geos (Poi F) add Emp) ⟹ Line_on (Li A D) F" by
(simp add:Point_Eq)
  from P5 P6 have "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi F) add Emp)" by blast
  then have "∃p. Bet_Point (Se C F) p" by (simp add:Seg_density)
  then obtain E :: Point where P7 : "Bet_Point (Se C F) E" by blast
  have P8 : "Line_on (Li A D) A" by (simp add:Line_on_rule)
  have P9 : "Line_on (Li A C) C" by (simp add:Line_on_rule)
  have P10 : "Line_on (Li A C) A" by (simp add:Line_on_rule)
  from assms have P11 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)"
    by (simp add:Bet_Point_def)
  from P4 P8 P9 P10 P11 have "Eq (Geos (Lin (Li A C)) add Emp) (Geos (Lin (Li A D)) add Emp)" by
(simp add:Line_unique)
  then have P12 : "Line_on (Li A C) F ⟹ Line_on (Li A D) F" by (simp add:Line_on_trans)
  from P5 P12 have P13 : "¬ Line_on (Li A C) F" by blast
  from assms P5 P7 P13 show "Bet_Point (Se A D) C" by (blast intro:Bet_swap_lemma_5)
qed

theorem(in Order_Rule) Bet_swap_234_124 :
  assumes
    "Bet_Point (Se A C) B"
```

```
    ”Bet_Point (Se B D) C”
  shows ”Bet_Point (Se A D) B”
proof -
  from assms have P1 : ”Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp) ⟹ Bet_Point (Se D C)
B” by (simp add:Bet_Point_Eq)
  from assms have ”Inv (Bet_Point (Se D C) B) ∧ Inv (Bet_Point (Se C B) D)” by (simp add:Bet_iff)
  then have P2 : ”¬ Bet_Point (Se D C) B” by (simp add:Inv_def)
  from P1 P2 have P3 : ”¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)” by blast
  from assms P3 have ”Line_on (Li A D) B ∧ Line_on (Li A D) C” by (simp add:Bet_swap_lemma_1)
  then have P4 : ”Line_on (Li A D) B” by simp
  have ”∃p q r. ¬ Line_on (Li A D) p ∧ ¬ Line_on (Li A D) q ∧ ¬ Line_on (Li A D) r
          ∧ ¬ Eq (Geos (Poi p) add Emp) (Geos (Poi q) add Emp) ∧ ¬ Eq (Geos (Poi q) add Emp) (Geos
(Poi r) add Emp)
          ∧ ¬ Eq (Geos (Poi r) add Emp) (Geos (Poi p) add Emp)” by (blast intro:Line_not_on_exist)
  then obtain F :: Point where P5 : ”¬ Line_on (Li A D) F” by blast
  from P4 have P6 : ”Eq (Geos (Poi B) add Emp) (Geos (Poi F) add Emp) ⟹ Line_on (Li A D) F” by
(simp add:Point_Eq)
  from P5 P6 have ”¬ Eq (Geos (Poi B) add Emp) (Geos (Poi F) add Emp)” by blast
  then have ”∃p. Bet_Point (Se B F) p” by (simp add:Seg_density)
  then obtain E :: Point where P7 : ”Bet_Point (Se B F) E” by blast
  from assms have P8 : ”Bet_Point (Se D B) C” by (simp add:Bet_rev)
  from assms have P9 : ”Bet_Point (Se C A) B” by (simp add:Bet_rev)
  from P3 have P10 : ”Eq (Geos (Lin (Li A D)) add Emp) (Geos (Lin (Li D A)) add Emp)” by (simp
add:Line_rev)
  from P5 P10 have P11 : ”¬ Line_on (Li D A) F” by (simp add:Line_not_on_trans)
  from P4 P10 have P12 : ”Line_on (Li D A) B” by (simp add:Line_on_trans)
  have P13 : ”Line_on (Li D A) D” by (simp add:Line_on_rule)
  have P14 : ”Line_on (Li D B) D” by (simp add:Line_on_rule)
  have P15 : ”Line_on (Li D B) B” by (simp add:Line_on_rule)
  from assms have P16 : ”¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)”
    by (simp add:Bet_Point_def)
  from P12 P13 P14 P15 P16 have ”Eq (Geos (Lin (Li D B)) add Emp) (Geos (Lin (Li D A)) add Emp)”
by (simp add:Line_unique)
  then have P17 : ”Line_on (Li D B) F ⟹ Line_on (Li D A) F” by (simp add:Line_on_trans)
  from P11 P17 have P18 : ”¬ Line_on (Li D B) F” by blast
  from P7 P8 P9 P11 P18 have ”Bet_Point (Se D A) B” by (blast intro:Bet_swap_lemma_5)
  thus ”Bet_Point (Se A D) B” by (blast intro:Bet_rev)
qed

theorem(in Order_Rule) Bet_swap_134_234 :
  assumes
    ”Bet_Point (Se A C) B”
    ”Bet_Point (Se A D) C”
  shows ”Bet_Point (Se B D) C”
proof -
  from assms have P2 : ”¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)”
    by (simp add:Bet_Point_def)
  from assms have P3 : ”¬ Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp)”
    by (simp add:Bet_Point_def)
  then have P4 : ”¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)” by (blast intro:Eq_rev)
  from assms have P5 : ”Line_on (Li A B) C” by (simp add:Line_Bet_on)
  have P6 : ”Line_on (Li A B) A” by (simp add:Line_on_rule)
  from assms have P7 : ”Line_on (Li A D) C” by (simp add:Line_Bet_on)
  have P8 : ”Line_on (Li A D) A” by (simp add:Line_on_rule)
  from P2 P5 P6 P7 P8 have P9 : ”Eq (Geos (Lin (Li A B)) add Emp) (Geos (Lin (Li A D)) add Emp)”
by (simp add:Line_unique)
  have P10 : ”Line_on (Li A B) B” by (simp add:Line_on_rule)
  from P9 P10 have P11 : ”Line_on (Li A D) B” by (simp add:Line_on_trans)
  from assms have P12 : ”¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)”
    by (simp add:Bet_Point_def)
  then have P13 : ”Eq (Geos (Lin (Li A D)) add Emp) (Geos (Lin (Li D A)) add Emp)” by (simp
add:Line_rev)
  from P11 P13 have P14 : ”Line_on (Li D A) B” by (simp add:Line_on_trans)
  from P12 have P15 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi A) add Emp)” by (blast intro:Eq_rev)
  from assms have P16 : ”Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp) ⟹ Bet_Point (Se A C)
D” by (simp add:Point_Eq)
```

from assms have ”Inv (Bet_Point (Se D C) A) ∧ Inv (Bet_Point (Se C A) D)” by (simp add:Bet_iff)
then have ”¬ Bet_Point (Se C A) D” by (simp add:Inv_def)
then have P17 : ”¬ Bet_Point (Se A C) D” by (blast intro:Bet_rev)
from P16 P17 have P18 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi B) add Emp)” by (blast intro:Eq_rev)
from P14 P15 P18 have P19 : ”Line_on (Li D B) A” by (simp add:Line_on_rev)
from P18 have P20 : ”Eq (Geos (Lin (Li D B)) add Emp) (Geos (Lin (Li B D)) add Emp)” by (simp add:Line_rev)
from P19 P20 have P21 : ”Line_on (Li B D) A” by (simp add:Line_on_trans)
have P22 : ”Line_on (Li B D) B” by (simp add:Line_on_rule)
from P4 P8 P11 P21 P22 have P23 : ”Eq (Geos (Lin (Li A D)) add Emp) (Geos (Lin (Li B D)) add Emp)” by (simp add:Line_unique)
from P7 P23 have P24 : ”Line_on (Li B D) C” by (simp add:Line_on_trans)
have ”∃p q r. ¬ Line_on (Li A D) p ∧ ¬ Line_on (Li A D) q ∧ ¬ Line_on (Li A D) r
∧ ¬ Eq (Geos (Poi p) add Emp) (Geos (Poi q) add Emp) ∧ ¬ Eq (Geos (Poi q) add Emp) (Geos (Poi r) add Emp)
∧ ¬ Eq (Geos (Poi r) add Emp) (Geos (Poi p) add Emp)” by (blast intro:Line_not_on_exist)
then obtain F :: Point where P25 : ”¬ Line_on (Li A D) F” by blast
from P11 have P26 : ”Eq (Geos (Poi B) add Emp) (Geos (Poi F) add Emp) ⟹ Line_on (Li A D) F” by (simp add:Point_Eq)
from P25 P26 have ”¬ Eq (Geos (Poi B) add Emp) (Geos (Poi F) add Emp)” by blast
then have ”∃p. Bet_Point (Se B F) p” by (simp add:Seg_density)
then obtain G :: Point where P27 : ”Bet_Point (Se B F) G” by blast
from P11 P25 P27 have ”Inv (Line_on (Li A D) G)” by (simp add:Line_Bet_not_on)
then have P28 : ”¬ Line_on (Li A D) G” by (simp add:Inv_def)
from assms P25 have P29 : ”¬ Eq (Geos (Lin (Li F C)) add Emp) (Geos (Lin (Li F D)) add Emp)” by (simp add:Line_Bet_not_Eq)
from P7 have P30 : ”Eq (Geos (Poi C) add Emp) (Geos (Poi F) add Emp) ⟹ Line_on (Li A D) F” by (simp add:Point_Eq)
from P25 P30 have P31 : ”¬ Eq (Geos (Poi F) add Emp) (Geos (Poi C) add Emp)” by (blast intro:Eq_rev)
have P32 : ”Line_on (Li A D) D” by (simp add:Line_on_rule)
then have P33 : ”Eq (Geos (Poi D) add Emp) (Geos (Poi F) add Emp) ⟹ Line_on (Li A D) F” by (simp add:Point_Eq)
from P25 P33 have P34 : ”¬ Eq (Geos (Poi F) add Emp) (Geos (Poi D) add Emp)” by (blast intro:Eq_rev)
from P29 P31 P34 have P35 : ”¬ Line_on (Li F C) D” by (simp add:Line_not_Eq_on)
from P31 have P36 : ”Eq (Geos (Lin (Li F C)) add Emp) (Geos (Lin (Li C F)) add Emp)” by (simp add:Line_rev)
from P35 P36 have P37 : ”¬ Line_on (Li C F) D” by (simp add:Line_not_on_trans)
from assms have P38 : ”Bet_Point (Se D A) C” by (simp add:Bet_rev)
from P13 P25 have P39 : ”¬ Line_on (Li D A) F” by (simp add:Line_not_on_trans)
from P38 P39 have P40 : ”¬ Eq (Geos (Lin (Li F C)) add Emp) (Geos (Lin (Li F A)) add Emp)” by (simp add:Line_Bet_not_Eq)
from P8 have P41 : ”Eq (Geos (Poi A) add Emp) (Geos (Poi F) add Emp) ⟹ Line_on (Li A D) F” by (simp add:Point_Eq)
from P25 P41 have P42 : ”¬ Eq (Geos (Poi F) add Emp) (Geos (Poi A) add Emp)” by (blast intro:Eq_rev)
from P31 P40 P42 have P43 : ”¬ Line_on (Li F C) A” by (simp add:Line_not_Eq_on)
from P36 P43 have P44 : ”¬ Line_on (Li C F) A” by (simp add:Line_not_on_trans)
from P2 have P45 : ”¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)” by (blast intro:Eq_rev)
from P31 have P46 : ”¬ Eq (Geos (Poi C) add Emp) (Geos (Poi F) add Emp)” by (blast intro:Eq_rev)
from P44 P45 P46 have P47 : ”¬ Line_on (Li C A) F” by (blast intro:Line_on_rev)
from P45 have P48 : ”Eq (Geos (Lin (Li C A)) add Emp) (Geos (Lin (Li A C)) add Emp)” by (simp add:Line_rev)
from P47 P48 have P49 : ”¬ Line_on (Li A C) F” by (simp add:Line_not_on_trans)
from assms P49 have P50 : ”¬ Eq (Geos (Lin (Li F B)) add Emp) (Geos (Lin (Li F C)) add Emp)” by (simp add:Line_Bet_not_Eq)
from P11 have P51 : ”Eq (Geos (Poi B) add Emp) (Geos (Poi F) add Emp) ⟹ Line_on (Li A D) F” by (simp add:Point_Eq)
from P25 P51 have P52 : ”¬ Eq (Geos (Poi F) add Emp) (Geos (Poi B) add Emp)” by (blast intro:Eq_rev)
from P31 P50 P52 have P53 : ”¬ Line_on (Li F B) C” by (simp add:Line_not_Eq_on)
from P27 have P54 : ”Line_on (Li B F) G” by (simp add:Line_Bet_on)
from P27 have P56 : ”Line_on (Li F B) G” by (simp add:Line_Bet_on)
have P57 : ”Line_on (Li F B) F” by (simp add:Line_on_rule)

have P58 : ”Line_on (Li C F) F” by (simp add:Line_on_rule)
from P27 have P59 : ”¬ Eq (Geos (Poi F) add Emp) (Geos (Poi G) add Emp)”
by (simp add:Bet_Point_def)
from P56 P57 P58 P59 have P60 : ”Line_on (Li C F) G ⟹ Eq (Geos (Lin (Li C F)) add Emp) (Geos (Lin (Li F B)) add Emp)” by (simp add:Line_unique)
have P61 : ”Line_on (Li C F) C” by (simp add:Line_on_rule)
from P60 P61 have P62 : ”Line_on (Li C F) G ⟹ Line_on (Li F B) C” by (simp add:Line_on_trans)
from P53 P62 have P63 : ”¬ Line_on (Li C F) G” by blast
have P64 : ”Line_on (Li C F) C” by (simp add:Line_on_rule)
from assms P28 P37 P44 P63 P64 have P65 : ”Line_on_Seg (Li C F) (Se A G) ∧ ¬ Line_on_Seg (Li C F) (Se D G) ∨ Line_on_Seg (Li C F) (Se D G) ∧ ¬ Line_on_Seg (Li C F) (Se A G)” by (simp add:Pachets_axiom)
then have ”Line_on_Seg (Li C F) (Se A G) ⟹ ∃p. Line_on (Li C F) p ∧ Bet_Point (Se A G) p” by (simp add:Line_on_Seg_rule)
then obtain H :: Point where P66 : ”Line_on_Seg (Li C F) (Se A G) ⟹ Line_on (Li C F) H ∧ Bet_Point (Se A G) H” by blast
from P9 have P67 : ”Line_on (Li A B) F ⟹ Line_on (Li A D) F” by (simp add:Line_on_trans)
from P25 P67 have P68 : ”¬ Line_on (Li A B) F” by blast
from P4 have P69 : ”Eq (Geos (Lin (Li A B)) add Emp) (Geos (Lin (Li B A)) add Emp)” by (simp add:Line_rev)
from P68 P69 have P70 : ”¬ Line_on (Li B A) F” by (simp add:Line_not_on_trans)
from P3 P27 P66 P70 have ”Line_on_Seg (Li C F) (Se A G) ⟹ ∃p. Line_on (Li H F) p ∧ Bet_Point (Se A B) p” by (simp add:Bet_swap_lemma_4)
then obtain E :: Point where P71 : ”Line_on_Seg (Li C F) (Se A G) ⟹ Line_on (Li H F) E ∧ Bet_Point (Se A B) E” by blast
then have P72 : ”Line_on_Seg (Li C F) (Se A G) ⟹ Line_on (Li A B) E” by (simp add:Line_Bet_on)
from P36 have P73 : ”Eq (Geos (Lin (Li C F)) add Emp) (Geos (Lin (Li F C)) add Emp)” by (simp add:Eq_rev)
from P66 P73 have P74 : ”Line_on_Seg (Li C F) (Se A G) ⟹ Line_on (Li F C) H” by (simp add:Line_on_trans)
from P66 have ”Line_on_Seg (Li C F) (Se A G) ⟹ Bet_Point (Se A G) H” by simp
then have ”Line_on_Seg (Li C F) (Se A G) ⟹ Eq (Geos (Poi H) add Emp) (Geos (Poi F) add Emp) ⟹
Bet_Point (Se A G) F” by (simp add:Point_Eq)
then have P75 : ”Line_on_Seg (Li C F) (Se A G) ⟹ Eq (Geos (Poi H) add Emp) (Geos (Poi F) add Emp) ⟹
Line_on (Li A G) F” by (simp add:Line_Bet_on)
have P76 : ”Line_on (Li A G) G” by (simp add:Line_on_rule)
have P77 : ”Line_on (Li B F) F” by (simp add:Line_on_rule)
from P54 P59 P75 P76 P77 have P78 : ”Line_on_Seg (Li C F) (Se A G) ⟹ Eq (Geos (Poi H) add Emp) (Geos (Poi F) add Emp) ⟹
Eq (Geos (Lin (Li A G)) add Emp) (Geos (Lin (Li B F)) add Emp)” by (simp add:Line_unique)
have P79 : ”Line_on (Li A G) A” by (simp add:Line_on_rule)
from P78 P79 have P80 : ”Line_on_Seg (Li C F) (Se A G) ⟹ Eq (Geos (Poi H) add Emp) (Geos (Poi F) add Emp) ⟹
Line_on (Li B F) A” by (simp add:Line_on_trans)
have P81 : ”Line_on (Li B F) B” by (simp add:Line_on_rule)
from P4 P6 P10 P80 P81 have P82 : ”Line_on_Seg (Li C F) (Se A G) ⟹ Eq (Geos (Poi H) add Emp) (Geos (Poi F) add Emp) ⟹
Eq (Geos (Lin (Li B F)) add Emp) (Geos (Lin (Li A B)) add Emp)” by (simp add:Line_unique)
from P77 P82 have P83 : ”Line_on_Seg (Li C F) (Se A G) ⟹ Eq (Geos (Poi H) add Emp) (Geos (Poi F) add Emp) ⟹
Line_on (Li A B) F” by (simp add:Line_on_trans)
from P68 P83 have P84 : ”Line_on_Seg (Li C F) (Se A G) ⟹ ¬ Eq (Geos (Poi F) add Emp) (Geos (Poi H) add Emp)” by (blast intro:Eq_rev)
from P46 have P85 : ”¬ Eq (Geos (Poi F) add Emp) (Geos (Poi C) add Emp)” by (blast intro:Eq_rev)
from P74 P84 P85 have P86 : ”Line_on_Seg (Li C F) (Se A G) ⟹ Line_on (Li F H) C” by (blast intro:Line_on_rev)
from P84 have P87 : ”Line_on_Seg (Li C F) (Se A G) ⟹ Eq (Geos (Lin (Li F H)) add Emp) (Geos (Lin (Li H F)) add Emp)” by (simp add:Line_rev)
from P86 P87 have P88 : ”Line_on_Seg (Li C F) (Se A G) ⟹ Line_on (Li H F) C” by (simp add:Line_on_trans)
from P71 have ”Line_on_Seg (Li C F) (Se A G) ⟹ Bet_Point (Se A B) E” by simp
then have P89 : ”Line_on_Seg (Li C F) (Se A G) ⟹ Eq (Geos (Poi E) add Emp) (Geos (Poi C) add Emp) ⟹ Bet_Point (Se A B) C” by (simp add:Point_Eq)
from assms have ”Inv (Bet_Point (Se C B) A) ∧ Inv (Bet_Point (Se B A) C)” by (simp add:Bet_iff)

```
  then have "¬ Bet_Point (Se B A) C" by (simp add:Inv_def)
  then have P90 : "¬ Bet_Point (Se A B) C" by (blast intro:Bet_rev)
  from P89 P90 have P91 : "Line_on_Seg (Li C F) (Se A G) ⟹ ¬ Eq (Geos (Poi E) add Emp) (Geos
(Poi C) add Emp)" by blast
  from P5 P71 P72 P88 P91 have P92 : "Line_on_Seg (Li C F) (Se A G) ⟹
    Eq (Geos (Lin (Li A B)) add Emp) (Geos (Lin (Li H F)) add Emp)" by (simp add:Line_unique)
  from P4 P11 P12 have P93 : "Line_on (Li A B) D" by (simp add:Line_on_rev)
  from P92 P93 have P94 : "Line_on_Seg (Li C F) (Se A G) ⟹ Line_on (Li H F) D" by (simp
add:Line_on_trans)
  have P95 : "Line_on (Li C F) F" by (simp add:Line_on_rule)
  have P96 : "Line_on (Li H F) H" by (simp add:Line_on_rule)
  have P97 : "Line_on (Li H F) F" by (simp add:Line_on_rule)
  from P66 P84 P95 P96 P97 have P98 : "Line_on_Seg (Li C F) (Se A G) ⟹
    Eq (Geos (Lin (Li H F)) add Emp) (Geos (Lin (Li C F)) add Emp)" by (simp add:Line_unique)
  from P94 P98 have P99 : "Line_on_Seg (Li C F) (Se A G) ⟹ Line_on (Li C F) D" by (simp
add:Line_on_trans)
  from P37 P99 have P100 : "¬ Line_on_Seg (Li C F) (Se A G)" by blast
  from P65 P100 have "Line_on_Seg (Li C F) (Se D G)" by blast
  then have "∃p. Line_on (Li C F) p ∧ Bet_Point (Se D G) p" by (simp add:Line_on_Seg_rule)
  then obtain H2 :: Point where P101 : "Line_on (Li C F) H2 ∧ Bet_Point (Se D G) H2" by blast
  from P23 have "Eq (Geos (Lin (Li B D)) add Emp) (Geos (Lin (Li A D)) add Emp)" by (simp
add:Eq_rev)
  then have P102 : "Line_on (Li B D) F ⟹ Line_on (Li A D) F" by (simp add:Line_on_trans)
  from P25 P102 have P103 : "¬ Line_on (Li B D) F" by blast
  from P18 have P104 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)" by (blast intro:Eq_rev)
  from P27 P101 P103 P104 have "∃p. Line_on (Li H2 F) p ∧ Bet_Point (Se D B) p" by (simp
add:Bet_swap_lemma_4)
  then obtain C2 :: Point where P105 : "Line_on (Li H2 F) C2 ∧ Bet_Point (Se D B) C2" by blast
  have "Line_on (Li H2 F) F" by (simp add:Line_on_rule)
  then have P106 : "Eq (Geos (Lin (Li H2 F)) add Emp) (Geos (Lin (Li B D)) add Emp) ⟹ Line_on
(Li B D) F" by (simp add:Line_on_trans)
  from P103 P106 have P107 : "¬ Eq (Geos (Lin (Li H2 F)) add Emp) (Geos (Lin (Li B D)) add Emp)"
by blast
  from P73 P101 have P108 : "Line_on (Li F C) H2" by (simp add:Line_on_trans)
  from P101 have "Bet_Point (Se D G) H2" by simp
  then have "Eq (Geos (Poi H2) add Emp) (Geos (Poi F) add Emp) ⟹ Bet_Point (Se D G) F" by (simp
add:Point_Eq)
  then have P109 : "Eq (Geos (Poi H2) add Emp) (Geos (Poi F) add Emp) ⟹ Line_on (Li D G) F" by
(simp add:Line_Bet_on)
  have P110 : "Line_on (Li D G) G" by (simp add:Line_on_rule)
  from P54 P59 P77 P109 P110 have P111 : "Eq (Geos (Poi H2) add Emp) (Geos (Poi F) add Emp) ⟹
    Eq (Geos (Lin (Li B F)) add Emp) (Geos (Lin (Li D G)) add Emp)" by (simp add:Line_unique)
  from P81 P111 have P112 : "Eq (Geos (Poi H2) add Emp) (Geos (Poi F) add Emp) ⟹ Line_on (Li
D G) B" by (simp add:Line_on_trans)
  have P113 : "Line_on (Li D G) D" by (simp add:Line_on_rule)
  from P11 P18 P32 P112 P113 have P114 : "Eq (Geos (Poi H2) add Emp) (Geos (Poi F) add Emp) ⟹
    Eq (Geos (Lin (Li D G)) add Emp) (Geos (Lin (Li A D)) add Emp)" by (simp add:Line_unique)
  from P110 P114 have P115 : "Eq (Geos (Poi H2) add Emp) (Geos (Poi F) add Emp) ⟹ Line_on (Li
A D) G" by (simp add:Line_on_trans)
  from P28 P115 have P116 : "¬ Eq (Geos (Poi F) add Emp) (Geos (Poi H2) add Emp)" by (blast
intro:Eq_rev)
  from P31 P108 P116 have P117 : "Line_on (Li F H2) C" by (simp add:Line_on_rev)
  from P116 have P118 : "Eq (Geos (Lin (Li F H2)) add Emp) (Geos (Lin (Li H2 F)) add Emp)" by
(simp add:Line_rev)
  from P117 P118 have P119 : "Line_on (Li H2 F) C" by (simp add:Line_on_trans)
  from P105 have P121 : "Line_on (Li B D) C2" by (simp add:Line_Bet_on)
  from P24 P105 P107 P119 P121 have P122 : "Eq (Geos (Poi C2) add Emp) (Geos (Poi C) add Emp)"
by (blast intro:Line_unique_Point)
  from P105 have P123 : "Bet_Point (Se D B) C2" by simp
  from P122 P123 have "Bet_Point (Se D B) C" by (simp add:Point_Eq)
  thus "Bet_Point (Se B D) C" by (blast intro:Bet_rev)
qed

lemma(in Order_Rule) Bet_swap_134_124 :
  assumes
    "Bet_Point (Se A C) B"
```

```
      ”Bet_Point (Se A D) C”
   shows ”Bet_Point (Se A D) B”
proof -
   from assms have P1 : ”Bet_Point (Se B D) C” by (blast intro:Bet_swap_134_234)
   from assms P1 show ”Bet_Point (Se A D) B” by (blast intro:Bet_swap_234_124)
qed

theorem(in Order_Rule) Bet_swap_243_124 :
   assumes
      ”Bet_Point (Se A D) B”
      ”Bet_Point (Se B D) C”
   shows ”Bet_Point (Se A C) B”
proof -
   from assms have P1 : ”Bet_Point (Se D B) C” by (simp add:Bet_rev)
   from assms have P2 : ”Bet_Point (Se D A) B” by (simp add:Bet_rev)
   from P1 P2 have ”Bet_Point (Se C A) B” by (blast intro:Bet_swap_134_234)
   thus ”Bet_Point (Se A C) B” by (simp add:Bet_rev)
qed

theorem(in Order_Rule) Bet_swap_243_143 :
   assumes
      ”Bet_Point (Se A D) B”
      ”Bet_Point (Se B D) C”
   shows ”Bet_Point (Se A D) C”
proof -
   from assms have P1 : ”Bet_Point (Se D B) C” by (simp add:Bet_rev)
   from assms have P2 : ”Bet_Point (Se D A) B” by (simp add:Bet_rev)
   from P1 P2 have ”Bet_Point (Se D A) C” by (blast intro:Bet_swap_134_124)
   thus ”Bet_Point (Se A D) C” by (simp add:Bet_rev)
qed

lemma(in Order_Rule) Bet_four_Point_case :
   assumes
      ”Line_on l1 P”
      ”Line_on l1 Q”
      ”Line_on l1 R”
      ”Line_on l1 S”
      ”Bet_Point (Se P R) Q”
      ”¬ Eq (Geos (Poi P) add Emp) (Geos (Poi S) add Emp)”
      ”¬ Eq (Geos (Poi Q) add Emp) (Geos (Poi S) add Emp)”
      ”¬ Eq (Geos (Poi R) add Emp) (Geos (Poi S) add Emp)”
   shows ”Bet_Point (Se P S) R ∨ Bet_Point (Se R S) P
      ∨ Bet_Point (Se P R) S ∧ Bet_Point (Se P S) Q
      ∨ Bet_Point (Se P Q) S ∨ Bet_Point (Se Q S) P”
proof -
   from assms have P1 : ”¬ Eq (Geos (Poi P) add Emp) (Geos (Poi R) add Emp)”
      by (simp add:Bet_Point_def)
   from assms have P2 : ”¬ Eq (Geos (Poi S) add Emp) (Geos (Poi P) add Emp)” by (blast intro:Eq_rev)
   from assms P1 P2 have ”Bet_Point (Se P S) R ∨ Bet_Point (Se S R) P ∨ Bet_Point (Se R P) S”
      by (simp add:Bet_case)
   then have P3 : ”Bet_Point (Se P S) R ∨ Bet_Point (Se R S) P ∨ Bet_Point (Se P R) S” by (blast in-
tro:Bet_rev)
   from assms have P4 : ”¬ Eq (Geos (Poi S) add Emp) (Geos (Poi Q) add Emp)” by (blast intro:Eq_rev)
   from assms have P5 : ”¬ Eq (Geos (Poi Q) add Emp) (Geos (Poi P) add Emp)”
      by (simp add:Bet_Point_def)
   from assms P4 P5 have ”Bet_Point (Se P Q) S ∨ Bet_Point (Se Q S) P ∨ Bet_Point (Se S P) Q” by
(simp add:Bet_case)
   then have P6 : ”Bet_Point (Se P Q) S ∨ Bet_Point (Se Q S) P ∨ Bet_Point (Se P S) Q” by (blast
intro:Bet_rev)
   from P3 P6 show ”Bet_Point (Se P S) R ∨ Bet_Point (Se R S) P
      ∨ Bet_Point (Se P R) S ∧ Bet_Point (Se P S) Q
      ∨ Bet_Point (Se P Q) S ∨ Bet_Point (Se Q S) P” by blast
qed

lemma(in Order_Rule) Plane_diffside_rev :
   assumes
```

```
    "Plane_diffside l1 p1 p2"
  shows "Plane_diffside l1 p2 p1"
proof -
  from assms have "∃p. Bet_Point (Se p1 p2) p ∧ Line_on l1 p ∧ ¬ Line_on l1 p1 ∧ ¬ Line_on l1 p2"
by (simp add:Plane_diffside_def)
  then obtain p3 :: Point where P1 : "Bet_Point (Se p1 p2) p3 ∧ Line_on l1 p3 ∧ ¬ Line_on l1 p1 ∧ ¬
Line_on l1 p2" by blast
  then have P2 : "Bet_Point (Se p2 p1) p3" by (simp add:Bet_rev)
  from P1 P2 have "∃p. Bet_Point (Se p2 p1) p ∧ Line_on l1 p ∧ ¬ Line_on l1 p2 ∧ ¬ Line_on l1 p1"
by blast
  thus "Plane_diffside l1 p2 p1" by (simp add:Plane_diffside_def)
qed

lemma(in Order_Rule) Plane_sameside_rev :
  assumes
    "Plane_sameside l1 p1 p2"
  shows "Plane_sameside l1 p2 p1"
proof -
  have "Line_on_Seg l1 (Se p2 p1) ⟹ ∃p. Line_on l1 p ∧ Bet_Point (Se p2 p1) p"
    by (simp add:Line_on_Seg_rule)
  then obtain p3 :: Point where P1 : "Line_on_Seg l1 (Se p2 p1) ⟹
    Line_on l1 p3 ∧ Bet_Point (Se p2 p1) p3" by blast
  then have P2 : "Line_on_Seg l1 (Se p2 p1) ⟹ Bet_Point (Se p1 p2) p3" by (simp add:Bet_rev)
  from P1 P2 have "Line_on_Seg l1 (Se p2 p1) ⟹ ∃p. Line_on l1 p ∧ Bet_Point (Se p1 p2) p" by blast
  then have "Line_on_Seg l1 (Se p2 p1) ⟹ Line_on_Seg l1 (Se p1 p2)" by (simp add:Line_on_Seg_rule)
  then have P3 : "¬ Line_on_Seg l1 (Se p1 p2) ⟹ ¬ Line_on_Seg l1 (Se p2 p1)" by blast
  from assms have P4 : "¬ Line_on_Seg l1 (Se p1 p2) ∧ ¬ Line_on l1 p1
    ∧ ¬ Line_on l1 p2 ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)"
    by (simp add:Plane_sameside_def)
  from P3 P4 have P5 : "¬ Line_on_Seg l1 (Se p2 p1)" by blast
  from P4 have P6 : "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p1) add Emp)" by (blast intro:Eq_rev)
  from P4 P5 P6 show "Plane_sameside l1 p2 p1" by (simp add:Plane_sameside_def)
qed

lemma(in Order_Rule) Plane_sameside_not_diffside :
  assumes N :
    "Plane_sameside l1 p1 p2"
  shows "¬ Plane_diffside l1 p1 p2"
proof
  assume W : "Plane_diffside l1 p1 p2"
  then have "∃p. Bet_Point (Se p1 p2) p ∧ Line_on l1 p ∧ ¬ Line_on l1 p1 ∧ ¬ Line_on l1 p2" by (simp
add:Plane_diffside_def)
  then have "∃p. Line_on l1 p ∧ Bet_Point (Se p1 p2) p" by blast
  then have P1 : "Line_on_Seg l1 (Se p1 p2)" by (simp add:Line_on_Seg_rule)
  from N have P2 : "¬ Line_on_Seg l1 (Se p1 p2)" by (simp add:Plane_sameside_def)
  from P1 P2 show False by blast
qed

lemma(in Order_Rule) Plane_diffside_not_sameside :
  assumes N :
    "Plane_diffside l1 p1 p2"
  shows "¬ Plane_sameside l1 p1 p2"
proof
  assume W : "Plane_sameside l1 p1 p2"
  then have P1 : "¬ Plane_diffside l1 p1 p2" by (simp add:Plane_sameside_not_diffside)
  from N P1 show False by blast
qed

lemma(in Order_Rule) Plane_not_sameside_diffside :
  assumes "¬ Plane_sameside l1 p1 p2"
    "¬ Line_on l1 p1" "¬ Line_on l1 p2"
    "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)"
  shows "Plane_diffside l1 p1 p2"
proof -
  from assms have P1 : "¬ Line_on_Seg l1 (Se p1 p2) ⟹ Plane_sameside l1 p1 p2"
    by (simp add:Plane_sameside_def)
```

```
  from assms P1 have P2 : "Line_on_Seg l1 (Se p1 p2)" by blast
  from P2 have P3 : "∃p. Line_on l1 p ∧ Bet_Point (Se p1 p2) p" by (simp add:Line_on_Seg_rule)
  from assms P3 have "∃p. Bet_Point (Se p1 p2) p
    ∧ Line_on l1 p ∧ ¬ Line_on l1 p1 ∧ ¬ Line_on l1 p2" by blast
  thus "Plane_diffside l1 p1 p2" by (simp add:Plane_diffside_def)
qed

lemma(in Order_Rule) Plane_not_diffside_sameside :
  assumes "¬ Plane_diffside l1 p1 p2"
    "¬ Line_on l1 p1" "¬ Line_on l1 p2"
    "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)"
  shows "Plane_sameside l1 p1 p2"
proof -
  from assms have P1 : "¬ Plane_sameside l1 p1 p2 ⟹ Plane_diffside l1 p1 p2"
    by (simp add:Plane_not_sameside_diffside)
  from assms P1 show "Plane_sameside l1 p1 p2" by blast
qed

lemma(in Order_Rule) Plane_Line_diff_trans :
  assumes
    "Plane_diffside l1 p1 p2"
    "Eq (Geos (Lin l1) add Emp) (Geos (Lin l2) add Emp)"
  shows "Plane_diffside l2 p1 p2"
proof -
  from assms have "∃p. Bet_Point (Se p1 p2) p ∧ Line_on l1 p ∧ ¬ Line_on l1 p1 ∧ ¬ Line_on l1 p2"
by (simp add:Plane_diffside_def)
  then obtain p3 :: Point where P1 : "Bet_Point (Se p1 p2) p3 ∧ Line_on l1 p3 ∧ ¬ Line_on l1 p1 ∧ ¬
Line_on l1 p2" by blast
  from assms P1 have P2 : "Line_on l2 p3" by (simp add:Line_on_trans)
  from assms P1 have P3 : "¬ Line_on l2 p1" by (simp add:Line_not_on_trans)
  from assms P1 have P4 : "¬ Line_on l2 p2" by (simp add:Line_not_on_trans)
  from P1 P2 P3 P4 have "∃p. Bet_Point (Se p1 p2) p ∧ Line_on l2 p ∧ ¬ Line_on l2 p1 ∧ ¬ Line_on l2
p2" by blast
  thus "Plane_diffside l2 p1 p2" by (simp add:Plane_diffside_def)
qed

lemma(in Order_Rule) Plane_Line_trans :
  assumes
    "Plane_sameside l1 p1 p2"
    "Eq (Geos (Lin l1) add Emp) (Geos (Lin l2) add Emp)"
  shows "Plane_sameside l2 p1 p2"
proof -
  have "Line_on_Seg l2 (Se p1 p2) ⟹ ∃p. Line_on l2 p ∧ Bet_Point (Se p1 p2) p"
    by (simp add:Line_on_Seg_rule)
  then obtain p3 :: Point where P1 : "Line_on_Seg l2 (Se p1 p2) ⟹ Line_on l2 p3 ∧ Bet_Point (Se p1
p2) p3" by blast
  from assms P1 have P2 : "Line_on_Seg l2 (Se p1 p2) ⟹ Line_on l1 p3" by (blast intro:Line_on_trans
Eq_rev)
  from P1 P2 have "Line_on_Seg l2 (Se p1 p2) ⟹ ∃p. Line_on l1 p ∧ Bet_Point (Se p1 p2) p" by blast
  then have P3 : "Line_on_Seg l2 (Se p1 p2) ⟹ Line_on_Seg l1 (Se p1 p2)"
    by (simp add:Line_on_Seg_rule)
  from assms have P4 : "¬ Line_on_Seg l1 (Se p1 p2) ∧ ¬ Line_on l1 p1
    ∧ ¬ Line_on l1 p2 ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)"
    by (simp add:Plane_sameside_def)
  from P3 P4 have P5 : "¬ Line_on_Seg l2 (Se p1 p2)" by blast
  from assms P4 have P6 : "Line_on l2 p1 ⟹ Line_on l1 p1" by (blast intro:Line_on_trans Eq_rev)
  from P4 P6 have P7 : "¬ Line_on l2 p1" by blast
  from assms P4 have P8 : "Line_on l2 p2 ⟹ Line_on l1 p2" by (blast intro:Line_on_trans Eq_rev)
  from P4 P8 have P9 : "¬ Line_on l2 p2" by blast
  from P4 P5 P7 P9 show "Plane_sameside l2 p1 p2" by (simp add:Plane_sameside_def)
qed

lemma(in Order_Rule) Line_other_Point :
  assumes "Line_on l1 p1"
  shows "∃p. Line_on l1 p ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p) add Emp)"
proof -
```

have ”∃p q. Line_on l1 p ∧ Line_on l1 q ∧ ¬ Eq (Geos (Poi p) add Emp) (Geos (Poi q) add Emp)” by (blast intro:Line_on_exist)
then obtain p2 p3 :: Point where P1 : ”Line_on l1 p2 ∧ Line_on l1 p3 ∧ ¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp)” by blast
then have P2 : ”Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp) ∧ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp) ⟹
Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp)” by (blast intro:Eq_trans Eq_rev)
from P1 P2 have ”¬ (Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp) ∧ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp))” by blast
then have P3 : ”Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp) ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)
∨ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp) ∧ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)
∨ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp) ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)” by blast
from P1 have P4 : ”Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp) ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp) ⟹
∃p. Line_on l1 p ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p) add Emp)” by blast
from P1 have P5 : ”¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp) ∧ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp) ⟹
∃p. Line_on l1 p ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p) add Emp)” by blast
from P1 have P6 : ”¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp) ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp) ⟹
∃p. Line_on l1 p ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p) add Emp)” by blast
from P3 P4 P5 P6 show ”∃p. Line_on l1 p ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p) add Emp)” by blast
qed

lemma(in Order_Rule) Plane_Bet_sameside :
assumes
”Bet_Point (Se p1 p3) p2”
”Line_on l1 p1”
”¬ Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin l1) add Emp)”
shows ”Plane_sameside l1 p2 p3”
proof -
from assms have ”∃p. Line_on l1 p ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p) add Emp)” by (simp add:Line_other_Point)
then obtain p4 :: Point where P1 : ”Line_on l1 p4 ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p4) add Emp)” by blast
have P2 : ”Line_on (Li p4 p1) p4” by (simp add:Line_on_rule)
have P3 : ”Line_on (Li p4 p1) p1” by (simp add:Line_on_rule)
have ”Plane_diffside (Li p4 p1) p2 p3 ⟹
(∃p. Bet_Point (Se p2 p3) p ∧ Line_on (Li p4 p1) p ∧ ¬ Line_on (Li p4 p1) p2 ∧ ¬ Line_on (Li p4 p1) p3)” by (simp add:Plane_diffside_def)
then obtain p5 :: Point where P4 : ”Plane_diffside (Li p4 p1) p2 p3 ⟹
Bet_Point (Se p2 p3) p5 ∧ Line_on (Li p4 p1) p5 ∧ ¬ Line_on (Li p4 p1) p2 ∧ ¬ Line_on (Li p4 p1) p3” by blast
then have P5 : ”Plane_diffside (Li p4 p1) p2 p3 ⟹ Bet_Point (Se p3 p2) p5” by (simp add:Bet_rev)
from assms have P6 : ”Bet_Point (Se p3 p1) p2” by (simp add:Bet_rev)
from P5 P6 have ”Plane_diffside (Li p4 p1) p2 p3 ⟹ Bet_Point (Se p3 p1) p5”
by (blast intro:Bet_swap_134_124)
then have P7 : ”Plane_diffside (Li p4 p1) p2 p3 ⟹ Line_on (Li p3 p1) p5” by (simp add:Line_Bet_on)
have P8 : ”Line_on (Li p3 p1) p1” by (simp add:Line_on_rule)
from P4 have ”Plane_diffside (Li p4 p1) p2 p3 ⟹ Bet_Point (Se p2 p3) p5” by simp
then have P9 : ”Plane_diffside (Li p4 p1) p2 p3 ⟹ Eq (Geos (Poi p5) add Emp) (Geos (Poi p1) add Emp) ⟹
Bet_Point (Se p2 p3) p1” by (simp add:Point_Eq)
from assms have ”Inv (Bet_Point (Se p3 p2) p1) ∧ Inv (Bet_Point (Se p2 p1) p3)” by (simp add:Bet_iff)
then have ”¬ Bet_Point (Se p3 p2) p1” by (simp add:Inv_def)
then have P10 : ”¬ Bet_Point (Se p2 p3) p1” by (blast intro:Bet_rev)
from P9 P10 have P11 : ”Plane_diffside (Li p4 p1) p2 p3 ⟹ ¬ Eq (Geos (Poi p5) add Emp) (Geos (Poi p1) add Emp)” by blast
from P3 P4 P7 P8 P11 have P12 : ”Plane_diffside (Li p4 p1) p2 p3 ⟹
Eq (Geos (Lin (Li p3 p1)) add Emp) (Geos (Lin (Li p4 p1)) add Emp)” by (simp add:Line_unique)
have P13 : ”Line_on (Li p3 p1) p3” by (simp add:Line_on_rule)
from P12 P13 have P14 : ”Plane_diffside (Li p4 p1) p2 p3 ⟹ Line_on (Li p4 p1) p3” by (simp

```
add:Line_on_trans)
  from P4 P14 have P15 : "¬ Plane_diffside (Li p4 p1) p2 p3" by blast
  from assms P1 P2 P3 have "Eq (Geos (Lin (Li p4 p1)) add Emp) (Geos (Lin l1) add Emp)" by (simp
add:Line_unique)
  then have P16 : "Plane_diffside l1 p2 p3 ⟹ Plane_diffside (Li p4 p1) p2 p3"
    by (blast intro:Plane_Line_diff_trans Eq_rev)
  from P15 P16 have P17 : "¬ Plane_diffside l1 p2 p3" by blast
  from assms have P18 : "Line_on (Li p1 p3) p2" by (simp add:Line_Bet_on)
  have P19 : "Line_on (Li p1 p3) p1" by (simp add:Line_on_rule)
  have P20 : "Line_on (Li p1 p3) p3" by (simp add:Line_on_rule)
  from assms have P21 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)"
    by (simp add:Bet_Point_def)
  from assms P19 P20 P21 have P22 : "Line_on l1 p3 ⟹ Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos
(Lin l1) add Emp)" by (simp add:Line_unique)
  from assms P22 have P23 : "¬ Line_on l1 p3" by blast
  from assms have P24 : "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p1) add Emp)"
    by (simp add:Bet_Point_def)
  from assms P18 P19 P24 have P25 : "Line_on l1 p2 ⟹ Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos
(Lin l1) add Emp)" by (simp add:Line_unique)
  from assms P25 have P26 : "¬ Line_on l1 p2" by blast
  from assms have "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p2) add Emp)"
    by (simp add:Bet_Point_def)
  then have P27 : "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp)" by (blast intro:Eq_rev)
  from P17 P23 P26 P27 show "Plane_sameside l1 p2 p3" by (simp add:Plane_not_diffside_sameside)
qed

lemma(in Order_Rule) Plane_Bet_diffside :
  assumes
    "Bet_Point (Se p1 p3) p2"
    "Line_on l1 p2"
    "¬ Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin l1) add Emp)"
  shows "Plane_diffside l1 p1 p3"
proof -
  from assms have "∃p. Line_on l1 p ∧ ¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p) add Emp)" by
(simp add:Line_other_Point)
  then obtain p4 :: Point where P1 : "Line_on l1 p4 ∧ ¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p4)
add Emp)" by blast
  from assms have P2 : "Line_on (Li p1 p3) p2" by (simp add:Line_Bet_on)
  from assms P1 P2 have P3 : "Line_on (Li p1 p3) p4 ⟹ Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos
(Lin l1) add Emp)" by (simp add:Line_unique)
  from assms P3 have P4 : "¬ Line_on (Li p1 p3) p4" by blast
  have P5 : "Line_on (Li p4 p2) p4" by (simp add:Line_on_rule)
  have P6 : "Line_on (Li p4 p2) p2" by (simp add:Line_on_rule)
  from assms P4 have P7 : "¬ Eq (Geos (Lin (Li p4 p2)) add Emp) (Geos (Lin (Li p4 p3)) add Emp)"
by (simp add:Line_Bet_not_Eq)
  from assms have "Eq (Geos (Poi p2) add Emp) (Geos (Poi p4) add Emp) ⟹ Bet_Point (Se p1 p3)
p4" by (simp add:Point_Eq)
  then have P8 : "Eq (Geos (Poi p2) add Emp) (Geos (Poi p4) add Emp) ⟹ Line_on (Li p1 p3) p4" by
(simp add:Line_Bet_on)
  from assms P4 P8 have P9 : "¬ Eq (Geos (Poi p4) add Emp) (Geos (Poi p2) add Emp)" by (blast
intro:Eq_rev)
  have "Line_on (Li p1 p3) p3" by (simp add:Line_on_rule)
  then have P10 : "Eq (Geos (Poi p3) add Emp) (Geos (Poi p4) add Emp) ⟹ Line_on (Li p1 p3) p4"
by (simp add:Point_Eq)
  from assms P4 P10 have P11 : "¬ Eq (Geos (Poi p4) add Emp) (Geos (Poi p3) add Emp)" by (blast
intro:Eq_rev)
  from P7 P9 P11 have P12 : "¬ Line_on (Li p4 p2) p3" by (simp add:Line_not_Eq_on)
  from assms have P13 : "Bet_Point (Se p3 p1) p2" by (simp add:Bet_rev)
  from assms have "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)" by (simp add:Bet_Point_def)
  then have P14 : "Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin (Li p3 p1)) add Emp)" by (simp
add:Line_rev)
  from assms P4 P14 have P15 : "¬ Line_on (Li p3 p1) p4" by (simp add:Line_not_on_trans)
  from P13 P15 have P16 : "¬ Eq (Geos (Lin (Li p4 p2)) add Emp) (Geos (Lin (Li p4 p1)) add Emp)"
by (simp add:Line_Bet_not_Eq)
  have "Line_on (Li p1 p3) p1" by (simp add:Line_on_rule)
  then have P17 : "Eq (Geos (Poi p1) add Emp) (Geos (Poi p4) add Emp) ⟹ Line_on (Li p1 p3) p4"
```

```
by (simp add:Point_Eq)
  from assms P4 P17 have P18 : "¬ Eq (Geos (Poi p4) add Emp) (Geos (Poi p1) add Emp)" by (blast
intro:Eq_rev)
  from P9 P16 P18 have P19 : "¬ Line_on (Li p4 p2) p1" by (simp add:Line_not_Eq_on)
  from assms P6 P12 P19 have "∃p. Bet_Point (Se p1 p3) p ∧ Line_on (Li p4 p2) p ∧ ¬ Line_on (Li p4
p2) p1 ∧ ¬ Line_on (Li p4 p2) p3" by blast
  then have P20 : "Plane_diffside (Li p4 p2) p1 p3" by (simp add:Plane_diffside_def)
  from assms P1 P5 P6 have P21 : "Eq (Geos (Lin (Li p4 p2)) add Emp) (Geos (Lin l1) add Emp)" by
(simp add:Line_unique)
  from P20 P21 show "Plane_diffside l1 p1 p3" by (simp add:Plane_Line_diff_trans)
qed

lemma(in Order_Rule) Plane_trans_inv :
  assumes
    "Plane_diffside l1 A B"
    "Plane_diffside l1 A C"
    "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)"
  shows "Plane_sameside l1 B C"
proof -
  from assms have "∃p. Bet_Point (Se A B) p ∧ Line_on l1 p ∧ ¬ Line_on l1 A ∧ ¬ Line_on l1 B" by
(simp add:Plane_diffside_def)
  then obtain D :: Point where P1 : "Bet_Point (Se A B) D ∧ Line_on l1 D ∧ ¬ Line_on l1 A ∧ ¬
Line_on l1 B" by blast
  then have P2 : "Bet_Point (Se A B) D" by simp
  from assms have "∃p. Bet_Point (Se A C) p ∧ Line_on l1 p ∧ ¬ Line_on l1 A ∧ ¬ Line_on l1 C" by
(simp add:Plane_diffside_def)
  then obtain p2 :: Point where P3 : "Bet_Point (Se A C) p2 ∧ Line_on l1 p2 ∧ ¬ Line_on l1 A ∧ ¬
Line_on l1 C" by blast
  then have "Bet_Point (Se A C) p2" by simp
  then have P4 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)" by (simp add:Bet_Point_def)
  from P3 have P5 : "¬ Line_on l1 C" by simp
  from P1 have P6 : "Line_on l1 D" by simp
  from P1 have P7 : "¬ Line_on l1 A" by simp
  from P1 have P8 : "¬ Line_on l1 B" by simp
  from P2 P5 P6 P7 P8 have P9 : "¬ Line_on (Li A B) C ⟹ Line_on_Seg l1 (Se A C) ∧ ¬ Line_on_Seg
l1 (Se B C)
          ∨ Line_on_Seg l1 (Se B C) ∧ ¬ Line_on_Seg l1 (Se A C)" by (simp add:Pachets_axiom)
  from P3 have "Bet_Point (Se A C) p2 ∧ Line_on l1 p2" by simp
  then have "∃p. Line_on l1 p ∧ Bet_Point (Se A C) p" by blast
  then have P10 : "Line_on_Seg l1 (Se A C)" by (simp add:Line_on_Seg_rule)
  from P9 P10 have P11 : "¬ Line_on (Li A B) C ⟹ ¬ Line_on_Seg l1 (Se B C)" by blast
  from assms P5 P8 P11 have P12 : "¬ Line_on (Li A B) C ⟹ Plane_sameside l1 B C" by (simp
add:Plane_sameside_def)
  from P6 have P13 : "Eq (Geos (Poi D) add Emp) (Geos (Poi C) add Emp) ⟹ Line_on l1 C" by (simp
add:Point_Eq)
  from P5 P13 have P14 : "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi C) add Emp)" by blast
  from P2 have P15 : "Line_on (Li A B) D" by (simp add:Line_Bet_on)
  from P2 have P16 : "Line_on (Li A B) A" by (simp add:Line_on_rule)
  from P2 have P17 : "Line_on (Li A B) B" by (simp add:Line_on_rule)
  from assms P2 P4 P14 P15 P16 P17 have P18 : "Line_on (Li A B) C ⟹ Bet_Point (Se A C) B ∨
Bet_Point (Se B C) A
     ∨ Bet_Point (Se A B) C ∧ Bet_Point (Se A C) D ∨ Bet_Point (Se A D) C ∨ Bet_Point (Se D C) A"
by (simp add:Bet_four_Point_case)
  from P2 have P19 : "Line_on (Li A B) C ⟹ Bet_Point (Se A C) B ⟹ Bet_Point (Se D C) B" by
(blast intro:Bet_swap_134_234)
  have "Line_on (Li D C) C" by (simp add:Line_on_rule)
  then have P20 : "Eq (Geos (Lin (Li D C)) add Emp) (Geos (Lin l1) add Emp) ⟹ Line_on l1 C" by
(simp add:Line_on_trans)
  from P5 P20 have P21 : "¬ Eq (Geos (Lin (Li D C)) add Emp) (Geos (Lin l1) add Emp)" by blast
  from P6 P19 P21 have P22 : "Line_on (Li A B) C ⟹ Bet_Point (Se A C) B ⟹ Plane_sameside l1
B C" by (simp add:Plane_Bet_sameside)
  from P2 have "Bet_Point (Se B A) D" by (simp add:Bet_rev)
  then have P23 : "Bet_Point (Se B C) A ⟹ Bet_Point (Se D C) A" by (blast intro:Bet_swap_134_234)
  from P6 P21 P23 have P24 : "Bet_Point (Se B C) A ⟹ Plane_sameside l1 A C"
    by (simp add:Plane_Bet_sameside)
  from assms have P25 : "¬ Plane_sameside l1 A C" by (simp add:Plane_diffside_not_sameside)
```

```
  from P24 P25 have P26 : "¬ Bet_Point (Se B C) A" by blast
  have "Bet_Point (Se A B) C ∧ Bet_Point (Se A C) D ⟹ Bet_Point (Se B A) C ∧ Bet_Point (Se C A)
D" by (simp add:Bet_rev)
  then have P27 : "Bet_Point (Se A B) C ∧ Bet_Point (Se A C) D ⟹ Bet_Point (Se D B) C" by (blast
intro:Bet_swap_243_124 Bet_rev)
  have "Line_on (Li D B) B" by (simp add:Line_on_rule)
  then have P28 : "Eq (Geos (Lin (Li D B)) add Emp) (Geos (Lin l1) add Emp) ⟹ Line_on l1 B" by
(simp add:Line_on_trans)
  from P8 P28 have P29 : "¬ Eq (Geos (Lin (Li D B)) add Emp) (Geos (Lin l1) add Emp)" by blast
  from P6 P27 P29 have P30 : "Bet_Point (Se A B) C ∧ Bet_Point (Se A C) D ⟹ Plane_sameside l1 B
C" by (simp add:Plane_Bet_sameside Plane_sameside_rev)
  have P31 : "Bet_Point (Se A D) C ⟹ Bet_Point (Se D A) C" by (simp add:Bet_rev)
  have "Line_on (Li D A) A" by (simp add:Line_on_rule)
  then have P32 : "Eq (Geos (Lin (Li D A)) add Emp) (Geos (Lin l1) add Emp) ⟹ Line_on l1 A" by
(simp add:Line_on_trans)
  from P7 P32 have P33 : "¬ Eq (Geos (Lin (Li D A)) add Emp) (Geos (Lin l1) add Emp)" by blast
  from P6 P31 P33 have P34 : "Bet_Point (Se A D) C ⟹ Plane_sameside l1 A C"
    by (simp add:Plane_Bet_sameside Plane_sameside_rev)
  from P25 P34 have P35 : "¬ Bet_Point (Se A D) C" by blast
  from P6 P21 have P36 : "Bet_Point (Se D C) A ⟹ Plane_sameside l1 A C"
    by (simp add:Plane_Bet_sameside)
  from P25 P36 have P37 : "¬ Bet_Point (Se D C) A" by blast
  from P18 P22 P26 P30 P35 P37 have P38 : "Line_on (Li A B) C ⟹ Plane_sameside l1 B C" by blast
  from P12 P38 show "Plane_sameside l1 B C" by blast
qed

lemma(in Order_Rule) Plane_trans :
  assumes
    "Plane_sameside l1 A B"
    "Plane_diffside l1 A C"
  shows "Plane_diffside l1 B C"
proof -
  from assms have "∃p. Bet_Point (Se A C) p ∧ Line_on l1 p ∧ ¬ Line_on l1 A ∧ ¬ Line_on l1 C" by
(simp add:Plane_diffside_def)
  then obtain D :: Point where P1 : "Bet_Point (Se A C) D ∧ Line_on l1 D ∧ ¬ Line_on l1 A ∧ ¬
Line_on l1 C" by blast
  from assms have P2 : "¬ Line_on l1 B" by (simp add:Plane_sameside_def)
  from P1 have P3 : "Bet_Point (Se A C) D" by simp
  from P1 have P4 : "¬ Line_on l1 A" by simp
  from P1 have P5 : "¬ Line_on l1 C" by simp
  from P1 have P6 : "Line_on l1 D" by simp
  from P2 P3 P4 P5 P6 have P7 : "¬ Line_on (Li A C) B ⟹ Line_on_Seg l1 (Se A B) ∧ ¬ Line_on_Seg
l1 (Se C B)
    ∨ Line_on_Seg l1 (Se C B) ∧ ¬ Line_on_Seg l1 (Se A B)" by (simp add:Pachets_axiom)
  have P8 : "Line_on_Seg l1 (Se A B) ⟹ ∃p. Line_on l1 p ∧ Bet_Point (Se A B) p" by (simp
add:Line_on_Seg_rule)
  from P2 P4 P8 have "Line_on_Seg l1 (Se A B) ⟹ ∃p. Bet_Point (Se A B) p ∧ Line_on l1 p ∧ ¬
Line_on l1 A ∧ ¬ Line_on l1 B" by blast
  then have "Line_on_Seg l1 (Se A B) ⟹ Plane_diffside l1 A B" by (simp add:Plane_diffside_def)
  then have P9 : "Line_on_Seg l1 (Se A B) ⟹ ¬ Plane_sameside l1 A B"
    by (simp add:Plane_diffside_not_sameside)
  from assms P9 have P10 : "¬ Line_on_Seg l1 (Se A B)" by blast
  from P7 P10 have "¬ Line_on (Li A C) B ⟹ Line_on_Seg l1 (Se C B)"by blast
  then have P11 : "¬ Line_on (Li A C) B ⟹ ∃p. Line_on l1 p ∧ Bet_Point (Se C B) p" by (simp
add:Line_on_Seg_rule)
  from P2 P5 P11 have "¬ Line_on (Li A C) B ⟹ ∃p. Bet_Point (Se C B) p ∧ Line_on l1 p ∧ ¬ Line_on
l1 C ∧ ¬ Line_on l1 B" by blast
  then have "¬ Line_on (Li A C) B ⟹ Plane_diffside l1 C B" by (simp add:Plane_diffside_def)
  then have P12 : "¬ Line_on (Li A C) B ⟹ Plane_diffside l1 B C" by (simp add:Plane_diffside_rev)
  have P13 : "Line_on (Li A C) A" by (simp add:Line_on_rule)
  have P14 : "Line_on (Li A C) C" by (simp add:Line_on_rule)
  from P3 have P15 : "Line_on (Li A C) D" by (simp add:Line_Bet_on)
  from assms have "Eq (Geos (Poi C) add Emp) (Geos (Poi B) add Emp) ⟹ Plane_sameside l1 A C"
by (blast intro:Point_Eq Eq_rev)
  then have P16 : "Eq (Geos (Poi C) add Emp) (Geos (Poi B) add Emp) ⟹ ¬ Plane_diffside l1 A C"
by (simp add:Plane_sameside_not_diffside)
```

```
  from assms P16 have P17 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi B) add Emp)" by blast
  from P6 have P18 : "Eq (Geos (Poi D) add Emp) (Geos (Poi B) add Emp) ⟹ Line_on l1 B" by (simp
add:Point_Eq)
  from P2 P18 have P19 : "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi B) add Emp)" by blast
  from assms have P20 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)"
    by (simp add:Plane_sameside_def)
  from assms P3 P13 P14 P15 P17 P19 P20 have P21 : "Line_on (Li A C) B ⟹ Bet_Point (Se A B) C
∨ Bet_Point (Se C B) A
    ∨ Bet_Point (Se A C) B ∧ Bet_Point (Se A B) D ∨ Bet_Point (Se A D) B ∨ Bet_Point (Se D B) A"
by (simp add:Bet_four_Point_case)
  from P3 have P22 : "Bet_Point (Se A B) C ⟹ Bet_Point (Se A B) D" by (blast intro:Bet_swap_134_124)
  have "Line_on (Li A B) A" by (simp add:Line_on_rule)
  then have P23 : "Eq (Geos (Lin (Li A B)) add Emp) (Geos (Lin l1) add Emp) ⟹ Line_on l1 A" by
(simp add:Line_on_trans)
  from P4 P23 have P24 : "¬ Eq (Geos (Lin (Li A B)) add Emp) (Geos (Lin l1) add Emp)" by blast
  from P6 P22 P24 have "Bet_Point (Se A B) C ⟹ Plane_diffside l1 A B" by (simp add:Plane_Bet_diffside)
  then have P25 : "Bet_Point (Se A B) C ⟹ ¬ Plane_sameside l1 A B"
    by (simp add:Plane_diffside_not_sameside)
  from assms P25 have P26 : "¬ Bet_Point (Se A B) C" by blast
  from P3 have P27 : "Bet_Point (Se C A) D" by (simp add:Bet_rev)
  from P27 have P28 : "Bet_Point (Se C B) A ⟹ Bet_Point (Se C B) D" by (blast intro:Bet_swap_134_124)
  have "Line_on (Li C B) B" by (simp add:Line_on_rule)
  then have P29 : "Eq (Geos (Lin (Li C B)) add Emp) (Geos (Lin l1) add Emp) ⟹ Line_on l1 B" by
(simp add:Line_on_trans)
  from P2 P29 have P30 : "¬ Eq (Geos (Lin (Li C B)) add Emp) (Geos (Lin l1) add Emp)" by blast
  from P6 P28 P30 have "Bet_Point (Se C B) A ⟹ Plane_diffside l1 C B" by (simp add:Plane_Bet_diffside)
  then have P31 : "Bet_Point (Se C B) A ⟹ Plane_diffside l1 B C" by (blast intro:Plane_diffside_rev)
  from P6 P24 have "Bet_Point (Se A B) D ⟹ Plane_diffside l1 A B" by (simp add:Plane_Bet_diffside)
  then have P32 : "Bet_Point (Se A B) D ⟹ ¬ Plane_sameside l1 A B"
    by (simp add:Plane_diffside_not_sameside)
  from assms P32 have "¬ Bet_Point (Se A B) D" by blast
  then have P33 : "¬ (Bet_Point (Se A C) B ∧ Bet_Point (Se A B) D)" by blast
  from P3 have P34 : "Bet_Point (Se A D) B ⟹ Bet_Point (Se C B) D" by (blast intro:Bet_swap_134_234
Bet_rev)
  from P6 P30 P34 have "Bet_Point (Se A D) B ⟹ Plane_diffside l1 C B" by (simp add:Plane_Bet_diffside)
  then have P35 : "Bet_Point (Se A D) B ⟹ Plane_diffside l1 B C" by (simp add:Plane_diffside_rev)
  from P27 have P36 : "Bet_Point (Se D B) A ⟹ Bet_Point (Se C B) D" by (blast intro:Bet_swap_234_124
Bet_rev)
  from P6 P30 P36 have "Bet_Point (Se D B) A ⟹ Plane_diffside l1 C B" by (simp add:Plane_Bet_diffside)
  then have P37 : "Bet_Point (Se D B) A ⟹ Plane_diffside l1 B C" by (simp add:Plane_diffside_rev)
  from P21 P26 P31 P33 P35 P37 have P38 : "Line_on (Li A C) B ⟹ Plane_diffside l1 B C" by blast
  from P12 P38 show "Plane_diffside l1 B C" by blast
qed

lemma(in Order_Rule) Plane_sameside_trans :
  assumes
    "Plane_sameside l1 A B"
    "Plane_sameside l1 B C"
    "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)"
  shows "Plane_sameside l1 A C"
proof -
  from assms have P1 : "Plane_diffside l1 A C ⟹ Plane_diffside l1 B C" by (blast intro:Plane_trans)
  from assms have P2 : "¬ Plane_diffside l1 B C" by (simp add:Plane_sameside_not_diffside)
  from P1 P2 have P3 : "¬ Plane_diffside l1 A C" by blast
  from assms have P4 : "¬ Line_on l1 A" by (simp add:Plane_sameside_def)
  from assms have P5 : "¬ Line_on l1 C" by (simp add:Plane_sameside_def)
  from assms have P6 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)" by (blast intro:Eq_rev)
  from P3 P4 P5 P6 show "Plane_sameside l1 A C" by (simp add:Plane_not_diffside_sameside)
qed

lemma (in Order_Rule) Seg_Bet_not_on :
  assumes
    "Bet_Point (Se p1 p3) p2"
  shows "¬ Seg_on_Seg (Se p1 p2) (Se p2 p3)"
proof -
  from assms have "∃l. Line_on l p1 ∧ Line_on l p3 ∧ Line_on l p2" by (simp add:Line_Bet_exist)
```

then obtain l1 :: Line where P1 : "Line_on l1 p1 ∧ Line_on l1 p3 ∧ Line_on l1 p2" by blast
have "Seg_on_Seg (Se p1 p2) (Se p2 p3) ⟹ ∃p. Bet_Point (Se p1 p2) p ∧ Bet_Point (Se p2 p3) p" by (simp add:Seg_on_Seg_rule)
then obtain p4 :: Point where P2 : "Seg_on_Seg (Se p1 p2) (Se p2 p3) ⟹ Bet_Point (Se p1 p2) p4 ∧ Bet_Point (Se p2 p3) p4" by blast
then have P3 : "Seg_on_Seg (Se p1 p2) (Se p2 p3) ⟹ Bet_Point (Se p2 p1) p4" by (blast intro:Bet_rev)
from assms have P4 : "Bet_Point (Se p3 p1) p2" by (simp add:Bet_rev)
from P3 P4 have P5 : "Seg_on_Seg (Se p1 p2) (Se p2 p3) ⟹ Bet_Point (Se p3 p1) p4" by (blast intro:Bet_swap_243_143)
have "∃p q r. ¬ Line_on l1 p ∧ ¬ Line_on l1 q ∧ ¬ Line_on l1 r
∧ ¬ Eq (Geos (Poi p) add Emp) (Geos (Poi q) add Emp) ∧ ¬ Eq (Geos (Poi q) add Emp) (Geos (Poi r) add Emp)
∧ ¬ Eq (Geos (Poi r) add Emp) (Geos (Poi p) add Emp)" by (blast intro:Line_not_on_exist)
then obtain p5 :: Point where P6 : "¬ Line_on l1 p5" by blast
have P7 : "Line_on (Li p5 p4) p5" by (simp add:Line_on_rule)
have P8 : "Line_on (Li p3 p1) p3" by (simp add:Line_on_rule)
have P9 : "Line_on (Li p3 p1) p1" by (simp add:Line_on_rule)
from assms have P10 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)"
by (simp add:Bet_Point_def)
from P1 P8 P9 P10 have "Eq (Geos (Lin (Li p3 p1)) add Emp) (Geos (Lin l1) add Emp)" by (simp add:Line_unique)
then have P11 : "Line_on (Li p3 p1) p5 ⟹ Line_on l1 p5" by (simp add:Line_on_trans)
from P6 P11 have P12 : "¬ Line_on (Li p3 p1) p5" by blast
from P7 have P13 : "Eq (Geos (Lin (Li p5 p4)) add Emp) (Geos (Lin (Li p3 p1)) add Emp) ⟹ Line_on (Li p3 p1) p5" by (simp add:Line_on_trans)
from P12 P13 have P14 : "¬ Eq (Geos (Lin (Li p3 p1)) add Emp) (Geos (Lin (Li p5 p4)) add Emp)" by (blast intro:Eq_rev)
have P15 : "Line_on (Li p5 p4) p4" by (simp add:Line_on_rule)
from P5 P14 P15 have P16 : "Seg_on_Seg (Se p1 p2) (Se p2 p3) ⟹ Plane_diffside (Li p5 p4) p3 p1" by (simp add:Plane_Bet_diffside)
have P17 : "Line_on (Li p1 p2) p1" by (simp add:Line_on_rule)
have P18 : "Line_on (Li p1 p2) p2" by (simp add:Line_on_rule)
from assms have P19 : "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p1) add Emp)"
by (simp add:Bet_Point_def)
from P1 P17 P18 P19 have "Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin l1) add Emp)" by (simp add:Line_unique)
then have P20 : "Line_on (Li p1 p2) p5 ⟹ Line_on l1 p5" by (simp add:Line_on_trans)
from P6 P20 have P21 : "¬ Line_on (Li p1 p2) p5" by blast
from P7 have P22 : "Eq (Geos (Lin (Li p5 p4)) add Emp) (Geos (Lin (Li p1 p2)) add Emp) ⟹ Line_on (Li p1 p2) p5" by (simp add:Line_on_trans)
from P21 P22 have P23 : "¬ Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p5 p4)) add Emp)" by (blast intro:Eq_rev)
from P2 have P24 : "Seg_on_Seg (Se p1 p2) (Se p2 p3) ⟹ Bet_Point (Se p1 p2) p4" by simp
from P15 P23 P24 have "Seg_on_Seg (Se p1 p2) (Se p2 p3) ⟹ Plane_diffside (Li p5 p4) p1 p2" by (simp add:Plane_Bet_diffside)
then have P25 : "Seg_on_Seg (Se p1 p2) (Se p2 p3) ⟹ Plane_diffside (Li p5 p4) p2 p1" by (simp add:Plane_diffside_rev)
have P26 : "Line_on (Li p2 p3) p2" by (simp add:Line_on_rule)
have P27 : "Line_on (Li p2 p3) p3" by (simp add:Line_on_rule)
from assms have P28 : "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p2) add Emp)"
by (simp add:Bet_Point_def)
from P1 P26 P27 P28 have "Eq (Geos (Lin (Li p2 p3)) add Emp) (Geos (Lin l1) add Emp)" by (simp add:Line_unique)
then have P29 : "Line_on (Li p2 p3) p5 ⟹ Line_on l1 p5" by (simp add:Line_on_trans)
from P6 P29 have P30 : "¬ Line_on (Li p2 p3) p5" by blast
from P7 have P31 : "Eq (Geos (Lin (Li p5 p4)) add Emp) (Geos (Lin (Li p2 p3)) add Emp) ⟹ Line_on (Li p2 p3) p5" by (simp add:Line_on_trans)
from P30 P31 have P32 : "¬ Eq (Geos (Lin (Li p2 p3)) add Emp) (Geos (Lin (Li p5 p4)) add Emp)" by (blast intro:Eq_rev)
from P2 have P33 : "Seg_on_Seg (Se p1 p2) (Se p2 p3) ⟹ Bet_Point (Se p2 p3) p4" by simp
from P15 P32 P33 have P34 : "Seg_on_Seg (Se p1 p2) (Se p2 p3) ⟹ Plane_diffside (Li p5 p4) p2 p3" by (simp add:Plane_Bet_diffside)
from P10 P25 P28 P34 have "Seg_on_Seg (Se p1 p2) (Se p2 p3) ⟹ Plane_sameside (Li p5 p4) p1 p3" by (blast intro:Plane_trans_inv)
then have "Seg_on_Seg (Se p1 p2) (Se p2 p3) ⟹ Plane_sameside (Li p5 p4) p3 p1" by (simp add:Plane_sameside_rev)

```
  then have P35 : "Seg_on_Seg (Se p1 p2) (Se p2 p3) ⟹ ¬ Plane_diffside (Li p5 p4) p3 p1" by (simp
add:Plane_sameside_not_diffside)
  from P16 P35 show "¬ Seg_on_Seg (Se p1 p2) (Se p2 p3)" by blast
qed

end


theory Congruence imports Order begin

locale Definition_3 = Order_Rule +
  fixes Def :: "Geo_object ⇒ bool"
    and Cong :: "Geo_objects ⇒ Geo_objects ⇒ bool"
    and Gr :: "Geo_objects ⇒ Geo_objects ⇒ bool"
    and Ang_inside :: "Angle ⇒ Point ⇒ bool"
    and Right_angle :: "Angle ⇒ bool"
  assumes Tri_def : "Def (Tri (Tr p1 p2 p3)) ⟷ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add
Emp)
          ∧ ¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp) ∧ ¬ Eq (Geos (Poi p3) add Emp)
(Geos (Poi p1) add Emp)
          ∧ ¬ Bet_Point (Se p1 p2) p3 ∧ ¬ Bet_Point (Se p2 p3) p1 ∧ ¬ Bet_Point (Se p3 p1) p2
          ∧ ¬ Seg_on_Seg (Se p1 p2) (Se p2 p3) ∧ ¬ Seg_on_Seg (Se p2 p3) (Se p3 p1) ∧ ¬ Seg_on_Seg
(Se p3 p1) (Se p1 p2)"
    and Cong_refl [simp,intro] : "Cong obs obs"
    and Ang_def : "Def (Ang (An p1 p2 p3)) ⟷ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add
Emp)
          ∧ ¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp) ∧ ¬ Eq (Geos (Poi p3) add Emp)
(Geos (Poi p1) add Emp)
          ∧ ¬ Eq (Geos (Lin (Li p2 p1)) add Emp) (Geos (Lin (Li p2 p3)) add Emp)"
    and Ang_rev : "⟦Cong (Geos (ang1) add Emp) (Geos (ang2) add Emp)⟧⟹
          Cong (Geos (ang2) add Emp) (Geos (ang1) add Emp)"
    and Ang_roll : "Cong (Geos (Ang (An p1 p2 p3)) add Emp) (Geos (Ang (An p3 p2 p1)) add Emp)
          ∧ Eq (Geos (Ang (An p1 p2 p3)) add Emp) (Geos (Ang (An p3 p2 p1)) add Emp)"
    and Ang_inside_def : "Ang_inside (An p1 p2 p3) p ⟷ Def (Ang (An p1 p2 p3)) ∧ Plane_sameside
(Li p2 p1) p3 p ∧ Plane_sameside (Li p2 p3) p1 p"
    and Ang_Point_swap : "⟦Def (Ang (An p1 p2 p3)); Line_on (Li p2 p1) p4; ¬ Bet_Point (Se p1 p4)
p2;
          Line_on (Li p2 p3) p5; ¬ Bet_Point (Se p3 p5) p2; ¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi
p4) add Emp);
          ¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p5) add Emp)⟧⟹
          Eq (Geos (Ang (An p1 p2 p3)) add Emp) (Geos (Ang (An p4 p2 p5)) add Emp) ∧ Def (Ang
(An p4 p2 p5))"
    and Ang_Right_angle_def : "Right_angle (An p1 p2 p3) ⟷
          (∃p. Cong (Geos (Ang (An p1 p2 p3)) add Emp) (Geos (Ang (An p1 p2 p)) add Emp)
          ∧ Bet_Point (Se p3 p) p2 ∧ Def (Ang (An p1 p2 p3)) ∧ Def (Ang (An p1 p2 p)))"
    and Tri_Cong_def : "Cong (Geos (Tri (Tr p11 p12 p13)) add Emp) (Geos (Tri (Tr p21 p22 p23))
add Emp)
          ⟷ Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p22)) add Emp)
          ∧ Eq (Geos (Seg (Se p12 p13)) add Emp) (Geos (Seg (Se p22 p23)) add Emp)
          ∧ Eq (Geos (Seg (Se p13 p11)) add Emp) (Geos (Seg (Se p23 p21)) add Emp)
          ∧ Cong (Geos (Ang (An p12 p11 p13)) add Emp) (Geos (Ang (An p22 p21 p23)) add Emp)
          ∧ Cong (Geos (Ang (An p13 p12 p11)) add Emp) (Geos (Ang (An p23 p22 p21)) add Emp)
          ∧ Cong (Geos (Ang (An p11 p13 p12)) add Emp) (Geos (Ang (An p21 p23 p22)) add Emp)"
    and Ang_greater_def : "⟦Cong (Geos (Ang an1) add Emp) (Geos (Ang (An p4 p2 p3)) add Emp);
          Plane_sameside (Li p2 p3) p4 p1⟧⟹
          Ang_inside (An p1 p2 p3) p4 ⟷ Gr (Geos (Ang (An p1 p2 p3)) add Emp) (Geos (Ang an1)
add Emp)"
    and Ang_less_def : "⟦Cong (Geos (Ang an1) add Emp) (Geos (Ang (An p4 p2 p3)) add Emp);
          Plane_sameside (Li p2 p3) p4 p1; ¬ Ang_inside (An p1 p2 p3) p4;
          ¬ Eq (Geos (Lin (Li p2 p1)) add Emp) (Geos (Lin (Li p2 p4)) add Emp)⟧⟹
          Gr (Geos (Ang an1) add Emp) (Geos (Ang (An p1 p2 p3)) add Emp)"

locale Axiom_3 = Definition_3 +
  assumes Seg_add : "⟦Line_on l1 p11; Line_on l1 p12; Line_on l1 p13; ¬ Seg_on_Seg (Se p11 p12) (Se
p12 p13);
```

```
        Line_on l2 p21; Line_on l2 p22; Line_on l2 p23; ¬ Seg_on_Seg (Se p21 p22) (Se p22 p23);
        Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p22)) add Emp);
        Eq (Geos (Seg (Se p12 p13)) add Emp) (Geos (Seg (Se p22 p23)) add Emp)⟧⟹
        Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 p23)) add Emp)"
  and Seg_sub : "⟦Line_on l1 p11; Line_on l1 p12; Line_on l1 p13; ¬ Seg_on_Seg (Se p11 p12) (Se
p12 p13);
        Line_on l2 p21; Line_on l2 p22; Line_on l2 p23; ¬ Seg_on_Seg (Se p21 p22) (Se p22 p23);
        Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 p23)) add Emp)⟧⟹
        Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p22)) add Emp)
        ∧ Eq (Geos (Seg (Se p12 p13)) add Emp) (Geos (Seg (Se p22 p23)) add Emp)"
  and Ang_move_sameside : "⟦¬ Line_on (Li p1 p2) p3; Def (Ang a1)⟧⟹ ∃p. Cong (Geos (Ang a1)
add Emp) (Geos (Ang (An p p1 p2)) add Emp) ∧ Plane_sameside (Li p1 p2) p p3"
  and Ang_move_diffside : "⟦¬ Line_on (Li p1 p2) p3; Def (Ang a1)⟧⟹ ∃p. Cong (Geos (Ang a1)
add Emp) (Geos (Ang (An p p1 p2)) add Emp) ∧ Plane_diffside (Li p1 p2) p p3"
  and Ang_move_unique : "⟦Cong (Geos (Ang an1) add Emp) (Geos (Ang (An p1 p2 p3)) add Emp);
        Cong (Geos (Ang an1) add Emp) (Geos (Ang (An p4 p2 p3)) add Emp);
        Plane_sameside (Li p2 p3) p1 p4⟧⟹
        Eq (Geos (Lin (Li p1 p2)) add Emp) (Geos (Lin (Li p4 p2)) add Emp) ∧ ¬ Bet_Point (Se p1
p4) p2"
  and Tri_week_SAS : "⟦Def (Tri (Tr p11 p12 p13)); Def (Tri (Tr p21 p22 p23));
        Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p22)) add Emp);
        Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 p23)) add Emp);
        Cong (Geos (Ang (An p12 p11 p13)) add Emp) (Geos (Ang (An p22 p21 p23)) add Emp)⟧
        ⟹ Cong (Geos (Ang (An p13 p12 p11)) add Emp) (Geos (Ang (An p23 p22 p21)) add Emp)"

locale Congruence_Rule = Axiom_3 +
  assumes Ang_weektrans : "⟦Eq (Geos (Ang an1) add Emp) (Geos (Ang an2) add Emp);
        Cong (Geos (Ang an2) add Emp) (Geos (Ang an3) add Emp)⟧⟹ Cong (Geos (Ang an1) add
Emp) (Geos (Ang an3) add Emp)"

lemma (in Congruence_Rule) Seg_Bet_add :
  assumes
    "Bet_Point (Se p11 p13) p12"
    "Bet_Point (Se p21 p23) p22"
    "Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p22)) add Emp)"
    "Eq (Geos (Seg (Se p12 p13)) add Emp) (Geos (Seg (Se p22 p23)) add Emp)"
  shows "Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 p23)) add Emp)"
proof -
  from assms have "∃l. Line_on l p11 ∧ Line_on l p13 ∧ Line_on l p12" by (simp add:Line_Bet_exist)
  then obtain l1 :: Line where P1 : "Line_on l1 p11 ∧ Line_on l1 p13 ∧ Line_on l1 p12" by blast
  from assms have "∃l. Line_on l p21 ∧ Line_on l p23 ∧ Line_on l p22" by (simp add:Line_Bet_exist)
  then obtain l2 :: Line where P2 : "Line_on l2 p21 ∧ Line_on l2 p23 ∧ Line_on l2 p22" by blast
  from assms have P3 : "¬ Seg_on_Seg (Se p11 p12) (Se p12 p13)" by (simp add:Seg_Bet_not_on)
  from assms have P4 : "¬ Seg_on_Seg (Se p21 p22) (Se p22 p23)" by (simp add:Seg_Bet_not_on)
  from assms P1 P2 P3 P4 show "Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 p23)) add
Emp)" by (blast intro:Seg_add)
qed

lemma (in Congruence_Rule) Tri_sinple_def :
  assumes
    "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)"
    "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)"
    "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)"
    "¬ Line_on (Li A B) C"
  shows "Def (Tri (Tr A B C))"
proof -
  have P1 : "Bet_Point (Se A B) C ⟹ Line_on (Li A B) C" by (simp add:Line_Bet_on)
  from assms P1 have P2 : "¬ Bet_Point (Se A B) C" by blast
  from assms have P3 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp)" by (blast intro:Eq_rev)
  from P3 have P4 : "Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li A B)) add Emp)" by (simp
add:Line_rev)
  have P5 : "Bet_Point (Se B C) A ⟹ Line_on (Li A B) C" by (simp add:Line_Bet_on)
  from assms P5 have P6 : "¬ Bet_Point (Se B C) A" by blast
  have P7 : "Bet_Point (Se C A) B ⟹ Line_on (Li A B) C" by (simp add:Line_Bet_on)
```

from assms P7 have P8 : "¬ Bet_Point (Se C A) B" by blast
have "Seg_on_Seg (Se A B) (Se B C) ⟹ ∃p. Bet_Point (Se A B) p ∧ Bet_Point (Se B C) p" by (simp add:Seg_on_Seg_rule)
then obtain D :: Point where P9 : "Seg_on_Seg (Se A B) (Se B C) ⟹ Bet_Point (Se A B) D ∧ Bet_Point (Se B C) D" by blast
have P10 : "Line_on (Li A B) B" by (simp add:Line_on_rule)
from P9 have P11 : "Seg_on_Seg (Se A B) (Se B C) ⟹ Line_on (Li A B) D" by (simp add:Line_Bet_on)
have P12 : "Line_on (Li B C) B" by (simp add:Line_on_rule)
from P9 have P13 : "Seg_on_Seg (Se A B) (Se B C) ⟹ Line_on (Li B C) D" by (simp add:Line_Bet_on)
from P9 have "Seg_on_Seg (Se A B) (Se B C) ⟹ Bet_Point (Se A B) D" by simp
then have P14 : "Seg_on_Seg (Se A B) (Se B C) ⟹ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)" by (simp add:Bet_Point_def)
from P10 P11 P12 P13 P14 have P15 : "Seg_on_Seg (Se A B) (Se B C) ⟹ Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li A B)) add Emp)" by (simp add:Line_unique)
have P16 : "Line_on (Li B C) C" by (simp add:Line_on_rule)
from P15 P16 have P17 : "Seg_on_Seg (Se A B) (Se B C) ⟹ Line_on (Li A B) C" by (simp add:Line_on_trans)
from assms P17 have P18 : "¬ Seg_on_Seg (Se A B) (Se B C)" by blast
have "Seg_on_Seg (Se B C) (Se C A) ⟹ ∃p. Bet_Point (Se B C) p ∧ Bet_Point (Se C A) p" by (simp add:Seg_on_Seg_rule)
then obtain E :: Point where P19 : "Seg_on_Seg (Se B C) (Se C A) ⟹ Bet_Point (Se B C) E ∧ Bet_Point (Se C A) E" by blast
then have P20 : "Seg_on_Seg (Se B C) (Se C A) ⟹ Line_on (Li B C) E" by (simp add:Line_Bet_on)
have P21 : "Line_on (Li C A) C" by (simp add:Line_on_rule)
from P19 have P22 : "Seg_on_Seg (Se B C) (Se C A) ⟹ Line_on (Li C A) E" by (simp add:Line_Bet_on)
from P19 have "Seg_on_Seg (Se B C) (Se C A) ⟹ Bet_Point (Se B C) E" by simp
then have P23 : "Seg_on_Seg (Se B C) (Se C A) ⟹ ¬ Eq (Geos (Poi C) add Emp) (Geos (Poi E) add Emp)" by (simp add:Bet_Point_def)
from P16 P20 P21 P22 P23 have P24 : "Seg_on_Seg (Se B C) (Se C A) ⟹ Eq (Geos (Lin (Li C A)) add Emp) (Geos (Lin (Li B C)) add Emp)" by (simp add:Line_unique)
have P25 : "Line_on (Li C A) A" by (simp add:Line_on_rule)
from P24 P25 have P26 : "Seg_on_Seg (Se B C) (Se C A) ⟹ Line_on (Li B C) A" by (simp add:Line_on_trans)
from assms P3 P26 have P27 : "Seg_on_Seg (Se B C) (Se C A) ⟹ Line_on (Li B A) C" by (blast intro:Line_on_rev)
from P4 P27 have P28 : "Seg_on_Seg (Se B C) (Se C A) ⟹ Line_on (Li A B) C" by (simp add:Line_on_trans)
from assms P28 have P29 : "¬ Seg_on_Seg (Se B C) (Se C A)" by blast
have "Seg_on_Seg (Se C A) (Se A B) ⟹ ∃p. Bet_Point (Se C A) p ∧ Bet_Point (Se A B) p" by (simp add:Seg_on_Seg_rule)
then obtain F :: Point where P30 : "Seg_on_Seg (Se C A) (Se A B) ⟹ Bet_Point (Se C A) F ∧ Bet_Point (Se A B) F" by blast
then have P31 : "Seg_on_Seg (Se C A) (Se A B) ⟹ Line_on (Li C A) F" by (simp add:Line_Bet_on)
have P32 : "Line_on (Li A B) A" by (simp add:Line_on_rule)
from P30 have P33 : "Seg_on_Seg (Se C A) (Se A B) ⟹ Line_on (Li A B) F" by (simp add:Line_Bet_on)
from P30 have "Seg_on_Seg (Se C A) (Se A B) ⟹ Bet_Point (Se C A) F" by simp
then have P34 : "Seg_on_Seg (Se C A) (Se A B) ⟹ ¬ Eq (Geos (Poi A) add Emp) (Geos (Poi F) add Emp)" by (simp add:Bet_Point_def)
from P25 P31 P32 P33 P34 have P35 : "Seg_on_Seg (Se C A) (Se A B) ⟹ Eq (Geos (Lin (Li C A)) add Emp) (Geos (Lin (Li A B)) add Emp)" by (simp add:Line_unique)
from P21 P35 have P36 : "Seg_on_Seg (Se C A) (Se A B) ⟹ Line_on (Li A B) C" by (simp add:Line_on_trans)
from assms P36 have P37 : "¬ Seg_on_Seg (Se C A) (Se A B)" by blast
from assms P2 P6 P8 P18 P29 P37 show "Def (Tri (Tr A B C))" by (simp add:Tri_def)
qed

lemma (in Congruence_Rule) Tri_def_Line :
assumes
"Def (Tri (Tr A B C))"
shows "¬ Line_on (Li A B) C ∧ ¬ Line_on (Li B C) A ∧ ¬ Line_on (Li C A) B"
proof -
from assms have P1 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)
∧ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp) ∧ ¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)
∧ ¬ Bet_Point (Se A B) C ∧ ¬ Bet_Point (Se B C) A ∧ ¬ Bet_Point (Se C A) B" by (simp add:Tri_def)

```
  have P2 : "Line_on (Li A B) B" by (simp add:Line_on_rule)
  have P3 : "Line_on (Li A B) A" by (simp add:Line_on_rule)
  from P1 P2 P3 have P4 : "Line_on (Li A B) C ⟹ Bet_Point (Se A C) B ∨ Bet_Point (Se C B) A ∨
Bet_Point (Se B A) C" by (simp add:Bet_case)
  from P1 have P5 : "¬ Bet_Point (Se A C) B" by (blast intro:Bet_rev)
  from P1 have P6 : "¬ Bet_Point (Se C B) A" by (blast intro:Bet_rev)
  from P1 have P7 : "¬ Bet_Point (Se B A) C" by (blast intro:Bet_rev)
  from P4 P5 P6 P7 have P8 : "¬ Line_on (Li A B) C" by blast
  have P9 : "Line_on (Li B C) B" by (simp add:Line_on_rule)
  have P10 : "Line_on (Li B C) C" by (simp add:Line_on_rule)
  from P1 P9 P10 have P11 : "Line_on (Li B C) A ⟹ Bet_Point (Se A C) B ∨ Bet_Point (Se C B) A ∨
Bet_Point (Se B A) C" by (simp add:Bet_case)
  from P5 P6 P7 P11 have P12 : "¬ Line_on (Li B C) A" by blast
  have P13 : "Line_on (Li C A) C" by (simp add:Line_on_rule)
  have P14 : "Line_on (Li C A) A" by (simp add:Line_on_rule)
  from P1 P13 P14 have P15 : "Line_on (Li C A) B ⟹ Bet_Point (Se A C) B ∨ Bet_Point (Se C B) A
∨ Bet_Point (Se B A) C" by (simp add:Bet_case)
  from P5 P6 P7 P15 have P16 : "¬ Line_on (Li C A) B" by blast
  from P8 P12 P16 show "¬ Line_on (Li A B) C ∧ ¬ Line_on (Li B C) A ∧ ¬ Line_on (Li C A) B" by
simp
qed

lemma (in Congruence_Rule) Tri_def_trans :
  assumes
    "Def (Tri (Tr A B C))"
  shows "Def (Tri (Tr B C A))"
proof -
  from assms have P1 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)
        ∧ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp) ∧ ¬ Eq (Geos (Poi C) add Emp)
(Geos (Poi A) add Emp)
        ∧ ¬ Bet_Point (Se A B) C ∧ ¬ Bet_Point (Se B C) A ∧ ¬ Bet_Point (Se C A) B" by (simp
add:Tri_def)
  from assms have P2 : "¬ Line_on (Li B C) A" by (simp add:Tri_def_Line)
  from P1 P2 show "Def (Tri (Tr B C A))" by (simp add:Tri_sinple_def)
qed

lemma (in Congruence_Rule) Tri_def_rev :
  assumes
    "Def (Tri (Tr A B C))"
  shows "Def (Tri (Tr C B A))"
proof -
  from assms have P1 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)
        ∧ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp) ∧ ¬ Eq (Geos (Poi C) add Emp)
(Geos (Poi A) add Emp)
        ∧ ¬ Bet_Point (Se A B) C ∧ ¬ Bet_Point (Se B C) A ∧ ¬ Bet_Point (Se C A) B" by (simp
add:Tri_def)
  from assms have P2 : "¬ Line_on (Li B C) A" by (simp add:Tri_def_Line)
  from P1 have P3 : "Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li C B)) add Emp)" by (simp
add:Line_rev)
  from P2 P3 have P4 : "¬ Line_on (Li C B) A" by (simp add:Line_not_on_trans)
  from P1 have P5 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi B) add Emp)" by (blast intro:Eq_rev)
  from P1 have P6 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp)" by (blast intro:Eq_rev)
  from P1 have P7 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)" by (blast intro:Eq_rev)
  from P4 P5 P6 P7 show "Def (Tri (Tr C B A))" by (simp add:Tri_sinple_def)
qed

lemma (in Congruence_Rule) Tri_def_extension :
  assumes
    "Def (Tri (Tr A B C))"
    "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)"
    "Line_on (Li B C) D"
  shows "Def (Tri (Tr A B D))"
proof -
  from assms have P1 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)" by (simp add:Tri_def)
  from assms have P2 : "¬ Line_on (Li B C) A" by (simp add:Tri_def_Line)
  from assms have P3 : "Eq (Geos (Poi D) add Emp) (Geos (Poi A) add Emp) ⟹ Line_on (Li B C) A"
```

```
by (simp add:Point_Eq)
  from P2 P3 have P4 : "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi A) add Emp)" by blast
  from assms have P5 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)" by (simp add:Tri_def)
  from assms P5 have P6 : "Line_on (Li B D) C" by (simp add:Line_on_rev)
  have P7 : "Line_on (Li B D) B" by (simp add:Line_on_rule)
  have P8 : "Line_on (Li B C) B" by (simp add:Line_on_rule)
  have P9 : "Line_on (Li B C) C" by (simp add:Line_on_rule)
  from P5 P6 P7 P8 P9 have "Eq (Geos (Lin (Li B D)) add Emp) (Geos (Lin (Li B C)) add Emp)" by
(simp add:Line_unique)
  then have P10 : "Line_on (Li B D) A ⟹ Line_on (Li B C) A" by (simp add:Line_on_trans)
  from P2 P10 have P11 : "¬ Line_on (Li B D) A" by blast
  from assms P1 P4 P11 have "Def (Tri (Tr B D A))" by (simp add:Tri_sinple_def)
  thus "Def (Tri (Tr A B D))" by (simp add:Tri_def_trans)
qed

lemma (in Congruence_Rule) Ang_to_Tri :
  assumes
    "Def (Ang (An A B C))"
  shows "Def (Tri (Tr A B C))"
proof -
  from assms have P1 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)
    ∧ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp) ∧ ¬ Eq (Geos (Poi C) add Emp) (Geos
(Poi A) add Emp)
    ∧ ¬ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp)" by (simp add:Ang_def)
  have P2 : "Line_on (Li B A) B" by (simp add:Line_on_rule)
  have P3 : "Line_on (Li B C) B" by (simp add:Line_on_rule)
  have P4 : "Line_on (Li B C) C" by (simp add:Line_on_rule)
  from P1 have P5 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)" by simp
  from P1 have "Eq (Geos (Lin (Li A B)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (simp
add:Line_rev)
  then have P6 : "Line_on (Li A B) C ⟹ Line_on (Li B A) C" by (simp add:Line_on_trans)
  from P2 P3 P4 P5 P6 have P7 : "Line_on (Li A B) C ⟹ Eq (Geos (Lin (Li B A)) add Emp) (Geos
(Lin (Li B C)) add Emp)" by (simp add:Line_unique)
  from P1 P7 have P8 : "¬ Line_on (Li A B) C" by blast
  from P1 P8 show "Def (Tri (Tr A B C))" by (simp add:Tri_sinple_def)
qed

lemma (in Congruence_Rule) Ang_sinple_def :
  assumes
    "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)"
    "¬ Line_on (Li A B) C"
  shows "Def (Ang (An A B C))"
proof -
  from assms have P1 : "Eq (Geos (Lin (Li A B)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (simp
add:Line_rev)
  from assms P1 have P2 : "¬ Line_on (Li B A) C" by (simp add:Line_not_on_trans)
  have "Line_on (Li B A) B" by (simp add:Line_on_rule)
  then have P3 : "Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp) ⟹ Line_on (Li B A) C" by
(simp add:Point_Eq)
  from P2 P3 have P4 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)" by blast
  have "Line_on (Li B A) A" by (simp add:Line_on_rule)
  then have P5 : "Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp) ⟹ Line_on (Li B A) C" by
(simp add:Point_Eq)
  from P2 P5 have P6 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)" by (blast intro:Eq_rev)
  have "Line_on (Li B C) C" by (simp add:Line_on_rule)
  then have P7 : "Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li B A)) add Emp) ⟹ Line_on (Li B
A) C" by (simp add:Line_on_trans)
  from P2 P7 have P8 : "¬ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp)" by (blast
intro:Eq_rev)
  from assms P4 P6 P8 show "Def (Ang (An A B C))" by (simp add:Ang_def)
qed

lemma (in Congruence_Rule) Tri_to_Ang :
  assumes
    "Def (Tri (Tr A B C))"
  shows "Def (Ang (An A B C))"
```

```
proof -
  from assms have P1 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)
    ∧ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp) ∧ ¬ Eq (Geos (Poi C) add Emp) (Geos
(Poi A) add Emp)" by (simp add:Tri_def)
  from assms have P2 : "¬ Line_on (Li A B) C" by (simp add:Tri_def_Line)
  from P1 P2 show "Def (Ang (An A B C))" by (simp add:Ang_sinple_def)
qed

lemma (in Congruence_Rule) Ang_def_rev :
  assumes
    "Def (Ang (An A B C))"
  shows "Def (Ang (An C B A))"
proof -
  from assms have P1 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)
    ∧ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp) ∧ ¬ Eq (Geos (Poi C) add Emp) (Geos
(Poi A) add Emp)
    ∧ ¬ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp)" by (simp add:Ang_def)
  have P2 : "Line_on (Li B A) A" by (simp add:Line_on_rule)
  have P3 : "Line_on (Li B A) B" by (simp add:Line_on_rule)
  have P4 : "Line_on (Li B C) B" by (simp add:Line_on_rule)
  from P1 have P5 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)" by simp
  from P2 P3 P4 P5 have P6 : "Line_on (Li B C) A ⟹ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin
(Li B C)) add Emp)" by (simp add:Line_unique)
  from P1 P6 have P7 : "¬ Line_on (Li B C) A" by blast
  from P1 have P8 : "Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li C B)) add Emp)" by (simp
add:Line_rev)
  from P7 P8 have P9 : "¬ Line_on (Li C B) A" by (simp add:Line_not_on_trans)
  from P1 have P10 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi B) add Emp)" by (blast intro:Eq_rev)
  from P9 P10 show "Def (Ang (An C B A))" by (simp add:Ang_sinple_def)
qed

lemma (in Congruence_Rule) Ang_def_inv :
  assumes
    "Def (Ang (An A B C))"
  shows "Def (Ang (An A C B))"
proof -
  from assms have P1 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)
    ∧ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp) ∧ ¬ Eq (Geos (Poi C) add Emp) (Geos
(Poi A) add Emp)
    ∧ ¬ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp)" by (simp add:Ang_def)
  have P2 : "Line_on (Li B A) A" by (simp add:Line_on_rule)
  have P3 : "Line_on (Li B A) B" by (simp add:Line_on_rule)
  have P4 : "Line_on (Li B C) B" by (simp add:Line_on_rule)
  from P1 have P5 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)" by simp
  from P2 P3 P4 P5 have P6 : "Line_on (Li B C) A ⟹ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin
(Li B C)) add Emp)" by (simp add:Line_unique)
  from P1 P6 have P7 : "¬ Line_on (Li B C) A" by blast
  have P8 : "Line_on (Li A C) C" by (simp add:Line_on_rule)
  have P9 : "Line_on (Li B C) C" by (simp add:Line_on_rule)
  from P1 P4 P8 P9 have P10 : "Line_on (Li A C) B ⟹ Eq (Geos (Lin (Li A C)) add Emp) (Geos (Lin
(Li B C)) add Emp)" by (simp add:Line_unique)
  have P11 : "Line_on (Li A C) A" by (simp add:Line_on_rule)
  from P10 P11 have P12 : "Line_on (Li A C) B ⟹ Line_on (Li B C) A" by (simp add:Line_on_trans)
  from P7 P12 have P13 : "¬ Line_on (Li A C) B" by blast
  from P1 have P14 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)" by (blast intro:Eq_rev)
  from P13 P14 show "Def (Ang (An A C B))" by (simp add:Ang_sinple_def)
qed

lemma (in Congruence_Rule) Ang_def_extension :
  assumes
    "Def (Ang (An A B C))"
    "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)"
    "Line_on (Li B C) D"
  shows "Def (Ang (An A B D))"
proof -
  from assms have P1 : "Def (Tri (Tr A B C))" by (simp add:Ang_to_Tri)
```

```
  from assms P1 have "Def (Tri (Tr A B D))" by (simp add:Tri_def_extension)
  thus "Def (Ang (An A B D))" by (simp add:Tri_to_Ang)
qed

lemma (in Congruence_Rule) Bet_end_Point :
  shows "¬ Bet_Point (Se p1 p1) p2"
proof
  assume W : "Bet_Point (Se p1 p1) p2"
  then have P1 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p1) add Emp)" by (simp add:Bet_Point_def
del:Eq_refl)
  have P2 : "Eq (Geos (Poi p1) add Emp) (Geos (Poi p1) add Emp)" by simp
  from P1 P2 show False by blast
qed

lemma (in Congruence_Rule) Seg_Plane_sameside :
  assumes
    "Line_on l1 A"
    "Line_on l1 B"
    "Line_on l1 C"
    "¬ Line_on l1 D"
    "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)"
    "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)"
    "¬ Bet_Point (Se B C) A"
  shows "Plane_sameside (Li D A) B C ∨ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)"
proof -
  have "Line_on (Li D A) D" by (simp add:Line_on_rule)
  then have P1 : "Eq (Geos (Lin (Li D A)) add Emp) (Geos (Lin l1) add Emp) ⟹ Line_on l1 D" by
(simp add:Line_on_trans)
  from assms P1 have P2 : "¬ Eq (Geos (Lin (Li D A)) add Emp) (Geos (Lin l1) add Emp)" by blast
  have "Plane_diffside (Li D A) B C ⟹ ∃p. Bet_Point (Se B C) p ∧ Line_on (Li D A) p ∧ ¬ Line_on
(Li D A) B ∧ ¬ Line_on (Li D A) C" by (simp add:Plane_diffside_def)
  then obtain E :: Point where P3 : "Plane_diffside (Li D A) B C ⟹ Bet_Point (Se B C) E ∧ Line_on
(Li D A) E" by blast
  then have P4 : "Plane_diffside (Li D A) B C ⟹ Line_on (Li B C) E" by (simp add:Line_Bet_on)
  have P5 : "Line_on (Li B C) B" by (simp add:Line_on_rule)
  have P6 : "Line_on (Li B C) C" by (simp add:Line_on_rule)
  from assms P5 P6 have P7 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp) ⟹ Eq (Geos
(Lin (Li B C)) add Emp) (Geos (Lin l1) add Emp)" by (simp add:Line_unique)
  from P4 P7 have P8 : "Plane_diffside (Li D A) B C ⟹ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C)
add Emp) ⟹
    Line_on l1 E" by (simp add:Line_on_trans)
  have P9 : "Line_on (Li D A) A" by (simp add:Line_on_rule)
  from assms P2 P3 P8 P9 have P10 : "Plane_diffside (Li D A) B C ⟹ ¬ Eq (Geos (Poi B) add Emp)
(Geos (Poi C) add Emp) ⟹
    Eq (Geos (Poi E) add Emp) (Geos (Poi A) add Emp)" by (simp add:Line_unique_Point)
  from P3 have P11 : "Plane_diffside (Li D A) B C ⟹ Bet_Point (Se B C) E" by simp
  from P10 P11 have P12 : "Plane_diffside (Li D A) B C ⟹ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi
C) add Emp) ⟹
    Bet_Point (Se B C) A" by (simp add:Point_Eq)
  from assms P12 have "¬ Plane_diffside (Li D A) B C ∨ Eq (Geos (Poi B) add Emp) (Geos (Poi C)
add Emp)" by blast
  then have P13 : "¬ Plane_diffside (Li D A) B C ∧ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add
Emp)
    ∨ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)" by blast
  from assms P9 have P14 : "Line_on (Li D A) B ⟹ Eq (Geos (Lin (Li D A)) add Emp) (Geos (Lin
l1) add Emp)" by (simp add:Line_unique)
  from P2 P14 have P15 : "¬ Line_on (Li D A) B" by blast
  from assms P9 have P16 : "Line_on (Li D A) C ⟹ Eq (Geos (Lin (Li D A)) add Emp) (Geos (Lin
l1) add Emp)" by (simp add:Line_unique)
  from P2 P16 have P17 : "¬ Line_on (Li D A) C" by blast
  from P15 P17 have P18 : "¬ Plane_diffside (Li D A) B C ∧ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi
C) add Emp) ⟹
    Plane_sameside (Li D A) B C" by (simp add:Plane_not_diffside_sameside)
  from P13 P18 show "Plane_sameside (Li D A) B C ∨ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add
Emp)" by blast
qed
```

```
lemma (in Congruence_Rule) Seg_move_unique :
  assumes
    "Line_on l1 A"
    "Line_on l1 B"
    "Line_on l1 C"
    "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)"
    "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)"
    "Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se A C)) add Emp)"
    "¬ Bet_Point (Se B C) A"
  shows "Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)"
proof -
  have "∃p q r. ¬ Line_on l1 p ∧ ¬ Line_on l1 q ∧ ¬ Line_on l1 r
          ∧ ¬ Eq (Geos (Poi p) add Emp) (Geos (Poi q) add Emp) ∧ ¬ Eq (Geos (Poi q) add Emp) (Geos
(Poi r) add Emp)
          ∧ ¬ Eq (Geos (Poi r) add Emp) (Geos (Poi p) add Emp)" by (blast intro:Line_not_on_exist)
  then obtain D :: Point where P1 : "¬ Line_on l1 D" by blast
  have P2 : "Line_on (Li A D) D" by (simp add:Line_on_rule)
  have P3 : "Line_on (Li A B) A" by (simp add:Line_on_rule)
  have P4 : "Line_on (Li A B) B" by (simp add:Line_on_rule)
  from assms P3 P4 have P5 : "Eq (Geos (Lin l1) add Emp) (Geos (Lin (Li A B)) add Emp)" by (simp
add:Line_unique)
  from assms P5 have P6 : "Line_on (Li A B) C" by (simp add:Line_on_trans)
  from P1 P5 have P7 : "¬ Line_on (Li A B) D" by (simp add:Line_not_on_trans)
  from assms P7 have "Def (Ang (An A B D))" by (simp add:Ang_sinple_def)
  then have P8 : "Def (Ang (An D A B))" by (blast intro:Ang_def_rev Ang_def_inv)
  then have "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi A) add Emp)" by (simp add:Ang_def)
  then have P9 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)" by (blast intro:Eq_rev)
  have P10 : "¬ Bet_Point (Se D D) A" by (simp add:Bet_end_Point)
  from assms P2 P6 P8 P9 P10 have "Eq (Geos (Ang (An D A B)) add Emp) (Geos (Ang (An D A C))
add Emp)" by (simp add:Ang_Point_swap)
  then have P11 : "Cong (Geos (Ang (An D A B)) add Emp) (Geos (Ang (An D A C)) add Emp)" by
(simp add:Ang_weektrans)
  from assms P7 have "Def (Tri (Tr A B D))" by (blast intro:Ang_sinple_def Ang_to_Tri)
  then have "Def (Tri (Tr D B A))" by (simp add:Tri_def_rev)
  then have P12 : "Def (Tri (Tr A D B))" by (simp add:Tri_def_trans)
  have P13 : "Line_on (Li A C) A" by (simp add:Line_on_rule)
  have P14 : "Line_on (Li A C) C" by (simp add:Line_on_rule)
  from assms P13 P14 have P15 : "Eq (Geos (Lin l1) add Emp) (Geos (Lin (Li A C)) add Emp)" by
(simp add:Line_unique)
  from P1 P15 have P16 : "¬ Line_on (Li A C) D" by (simp add:Line_not_on_trans)
  from assms P16 have "Def (Tri (Tr A C D))" by (blast intro:Ang_sinple_def Ang_to_Tri)
  then have "Def (Tri (Tr D C A))" by (simp add:Tri_def_rev)
  then have P17 : "Def (Tri (Tr A D C))" by (simp add:Tri_def_trans)
  from assms P11 P12 P17 have P18 : "Cong (Geos (Ang (An B D A)) add Emp) (Geos (Ang (An C D
A)) add Emp)" by (simp add:Tri_week_SAS)
  have P19 : "Cong (Geos (Ang (An A D B)) add Emp) (Geos (Ang (An B D A)) add Emp)" by (simp
add:Ang_roll)
  have P20 : "Eq (Geos (Ang (An B D A)) add Emp) (Geos (Ang (An A D B)) add Emp)" by (simp
add:Ang_roll)
  from P18 P20 have P21 : "Cong (Geos (Ang (An A D B)) add Emp) (Geos (Ang (An C D A)) add
Emp)" by (blast intro:Ang_weektrans Eq_rev)
  from assms P1 have P22 : "Plane_sameside (Li D A) B C ∨ Eq (Geos (Poi B) add Emp) (Geos (Poi
C) add Emp)" by (simp add:Seg_Plane_sameside)
  from P19 P21 have P23 : "Plane_sameside (Li D A) B C ⟹ Eq (Geos (Lin (Li B D)) add Emp)
(Geos (Lin (Li C D)) add Emp)" by (simp add:Ang_move_unique)
  have P24 : "Line_on (Li B D) B" by (simp add:Line_on_rule)
  from P23 P24 have P25 : "Plane_sameside (Li D A) B C ⟹ Line_on (Li C D) B" by (simp
add:Line_on_trans)
  have P26 : "Line_on (Li B C) B" by (simp add:Line_on_rule)
  have P27 : "Line_on (Li B C) C" by (simp add:Line_on_rule)
  have P28 : "Line_on (Li C D) C" by (simp add:Line_on_rule)
  from P25 P26 P27 P28 have P29 : "Plane_sameside (Li D A) B C ⟹ ¬ Eq (Geos (Poi B) add Emp)
(Geos (Poi C) add Emp) ⟹
     Eq (Geos (Lin (Li C D)) add Emp) (Geos (Lin (Li B C)) add Emp)" by (simp add:Line_unique)
  have P30 : "Line_on (Li C D) D" by (simp add:Line_on_rule)
```

```
  from P29 P30 have P31 : "Plane_sameside (Li D A) B C ⟹ ¬ Eq (Geos (Poi B) add Emp) (Geos
(Poi C) add Emp) ⟹
    Line_on (Li B C) D" by (simp add:Line_on_trans)
  from assms P26 P27 have P32 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp) ⟹ Eq (Geos
(Lin (Li B C)) add Emp) (Geos (Lin l1) add Emp)" by (simp add:Line_unique)
  from P31 P32 have P33 : "Plane_sameside (Li D A) B C ⟹ ¬ Eq (Geos (Poi B) add Emp) (Geos
(Poi C) add Emp) ⟹
    Line_on l1 D" by (simp add:Line_on_trans)
  from P1 P33 have P34 : "Plane_sameside (Li D A) B C ⟹ Eq (Geos (Poi B) add Emp) (Geos (Poi
C) add Emp)" by blast
  from P22 P34 show "Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)" by blast
qed

lemma (in Congruence_Rule) Seg_not_Eq_Point :
  assumes
    "¬ Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se A C)) add Emp)"
  shows "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)"
proof -
  have P1 : "Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp) ⟹
    Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se A C)) add Emp)" by (simp add:Seg_Point_Eq)
  from assms P1 show "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)" by blast
qed

lemma (in Congruence_Rule) Ang_replace :
  assumes
    "Def (Ang (An A B C))"
    "Def (Ang (An A1 B1 C1))"
    "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A1 B1 C1)) add Emp)"
  shows "∃p. Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An p B1 C1)) add Emp)
    ∧ Eq (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An p B1 C1)) add Emp)
    ∧ Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se B1 p)) add Emp) ∧ Line_on (Li B1 A1) p ∧
¬ Bet_Point (Se p A1) B1 ∧ Def (Ang (An p B1 C1))"
    and "∃p. Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A1 B1 p)) add Emp)
    ∧ Eq (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An A1 B1 p)) add Emp)
    ∧ Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 p)) add Emp) ∧ Line_on (Li B1 C1) p ∧
¬ Bet_Point (Se p C1) B1 ∧ Def (Ang (An A1 B1 p))"
    and "∃p q. Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An p B1 q)) add Emp)
    ∧ Eq (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An p B1 q)) add Emp)
    ∧ Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se B1 p)) add Emp) ∧ Line_on (Li B1 A1) p ∧
¬ Bet_Point (Se p A1) B1
    ∧ Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 q)) add Emp) ∧ Line_on (Li B1 C1) q ∧
¬ Bet_Point (Se q C1) B1 ∧ Def (Ang (An p B1 q))"
proof -
  from assms have P1 : "¬ Eq (Geos (Poi A1) add Emp) (Geos (Poi B1) add Emp)" by (simp add:Ang_def)
  then have P2 : "¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi A1) add Emp)" by (blast intro:Eq_rev)
  have P3 : "Line_on (Li B1 A1) A1" by (simp add:Line_on_rule)
  have P4 : "Line_on (Li B1 A1) B1" by (simp add:Line_on_rule)
  from assms have "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)" by (simp add:Ang_def)
  then have P5 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp)" by (blast intro:Eq_rev)
  from P2 P3 P4 P5 have "∃p. Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se B1 p)) add Emp) ∧
¬ Bet_Point (Se p A1) B1 ∧ Line_on (Li B1 A1) p ∧ ¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi p) add
Emp)" by (simp add:Seg_move_sameside)
  then obtain A2 :: Point where P6 : "Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se B1 A2)) add
Emp) ∧ ¬ Bet_Point (Se A2 A1) B1 ∧ Line_on (Li B1 A1) A2 ∧ ¬ Eq (Geos (Poi B1) add Emp) (Geos
(Poi A2) add Emp)" by blast
  from assms have P7 : "¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi C1) add Emp)" by (simp add:Ang_def)
  have P8 : "Line_on (Li B1 C1) B1" by (simp add:Line_on_rule)
  have P9 : "Line_on (Li B1 C1) C1" by (simp add:Line_on_rule)
  from assms have P10 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)" by (simp add:Ang_def)
  from P7 P8 P9 P10 have "∃p. Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 p)) add Emp) ∧
¬ Bet_Point (Se p C1) B1 ∧ Line_on (Li B1 C1) p ∧ ¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi p) add
Emp)" by (simp add:Seg_move_sameside)
  then obtain C2 :: Point where P11 : "Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 C2)) add
Emp) ∧ ¬ Bet_Point (Se C2 C1) B1 ∧ Line_on (Li B1 C1) C2 ∧ ¬ Eq (Geos (Poi B1) add Emp) (Geos
(Poi C2) add Emp)" by blast
  have P12 : "¬ Bet_Point (Se C1 C1) B1" by (simp add:Bet_end_Point)
```

```
  from P6 have P13 : "¬ Bet_Point (Se A1 A2) B1" by (blast intro:Bet_rev)
  from assms P3 P6 P7 P9 P12 P13 have P14 : "Eq (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang
(An A2 B1 C1)) add Emp)" by (simp add:Ang_Point_swap)
  from assms P14 have P15 : "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A2 B1 C1))
add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from assms have P16 : "¬ Eq (Geos (Lin (Li B1 A1)) add Emp) (Geos (Lin (Li B1 C1)) add Emp)"
by (simp add:Ang_def)
  from P6 have P17 : "Line_on (Li B1 A1) A2" by simp
  from P6 have P18 : "¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi A2) add Emp)" by simp
  from P4 P8 P17 P18 have P19 : "Line_on (Li B1 C1) A2 ⟹ Eq (Geos (Lin (Li B1 A1)) add Emp)
(Geos (Lin (Li B1 C1)) add Emp)" by (simp add:Line_unique)
  from P16 P19 have P20 : "¬ Line_on (Li B1 C1) A2" by blast
  from P7 P20 have "Def (Ang (An B1 C1 A2))" by (simp add:Ang_sinple_def)
  then have P21 : "Def (Ang (An A2 B1 C1))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P6 P14 P15 P21 show "∃p. Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An p B1 C1))
add Emp)
    ∧ Eq (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An p B1 C1)) add Emp)
    ∧ Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se B1 p)) add Emp) ∧ Line_on (Li B1 A1) p ∧
¬ Bet_Point (Se p A1) B1 ∧ Def (Ang (An p B1 C1))" by blast
  have P22 : "¬ Bet_Point (Se A1 A1) B1" by (simp add:Bet_end_Point)
  from P11 have P23 : "¬ Bet_Point (Se C1 C2) B1" by (blast intro:Bet_rev)
  from assms P2 P3 P7 P11 P22 P23 have P24 : "Eq (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang
(An A1 B1 C2)) add Emp)" by (simp add:Ang_Point_swap)
  from assms P24 have P25 : "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A1 B1 C2))
add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from P11 have P26 : "Line_on (Li B1 C1) C2" by simp
  from P11 have P27 : "¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi C2) add Emp)" by simp
  from P4 P8 P26 P27 have P28 : "Line_on (Li B1 A1) C2 ⟹ Eq (Geos (Lin (Li B1 A1)) add Emp)
(Geos (Lin (Li B1 C1)) add Emp)" by (simp add:Line_unique)
  from P16 P28 have P29: "¬ Line_on (Li B1 A1) C2" by blast
  from P2 P29 have "Def (Ang (An B1 A1 C2))" by (simp add:Ang_sinple_def)
  then have P30 : "Def (Ang (An A1 B1 C2))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P11 P24 P25 P30 show "∃p. Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A1 B1
p)) add Emp)
    ∧ Eq (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An A1 B1 p)) add Emp)
    ∧ Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 p)) add Emp) ∧ Line_on (Li B1 C1) p ∧
¬ Bet_Point (Se p C1) B1 ∧ Def (Ang (An A1 B1 p))" by blast
  from assms P6 P11 P13 P17 P23 P26 have P31 : "Eq (Geos (Ang (An A1 B1 C1)) add Emp) (Geos
(Ang (An A2 B1 C2)) add Emp)" by (simp add:Ang_Point_swap)
  from assms P31 have P32 : "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A2 B1 C2))
add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  have P33 : "Line_on (Li B1 C2) B1" by (simp add:Line_on_rule)
  have P34 : "Line_on (Li B1 C2) C2" by (simp add:Line_on_rule)
  from P8 P26 P27 P33 P34 have "Eq (Geos (Lin (Li B1 C2)) add Emp) (Geos (Lin (Li B1 C1)) add
Emp)" by (simp add:Line_unique)
  then have P35 : "Line_on (Li B1 C2) A2 ⟹ Line_on (Li B1 C1) A2" by (simp add:Line_on_trans)
  from P20 P35 have P36 : "¬ Line_on (Li B1 C2) A2" by blast
  from P11 P36 have "Def (Ang (An B1 C2 A2))" by (simp add:Ang_sinple_def)
  then have P37 : "Def (Ang (An A2 B1 C2))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P6 P11 P21 P30 P31 P32 P37 show "∃p q. Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang
(An p B1 q)) add Emp)
    ∧ Eq (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An p B1 q)) add Emp)
    ∧ Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se B1 p)) add Emp) ∧ Line_on (Li B1 A1) p ∧
¬ Bet_Point (Se p A1) B1
    ∧ Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 q)) add Emp) ∧ Line_on (Li B1 C1) q ∧
¬ Bet_Point (Se q C1) B1 ∧ Def (Ang (An p B1 q))"
  by blast
qed

theorem (in Congruence_Rule) Tri_isosceles:
  assumes
    "Def (Tri (Tr A B C))"
    "Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se A C)) add Emp)"
  shows "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A C B)) add Emp)"
proof -
  from assms have P1 : "Eq (Geos (Seg (Se A C)) add Emp) (Geos (Seg (Se A B)) add Emp)" by (simp
```

```
add:Eq_rev)
  have P2 : "Cong (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An C A B)) add Emp)" by (simp
add:Ang_roll)
  from assms have "Def (Tri (Tr C B A))" by (simp add:Tri_def_rev)
  then have P3 : "Def (Tri (Tr A C B))" by (simp add:Tri_def_trans)
  from assms P1 P2 P3 have P4 : "Cong (Geos (Ang (An C B A)) add Emp) (Geos (Ang (An B C A))
add Emp)" by (simp add:Tri_week_SAS)
  have P5 : "Eq (Geos (Ang (An C B A)) add Emp) (Geos (Ang (An A B C)) add Emp)" by (simp
add:Ang_roll)
  from P4 P5 have P6 : "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An B C A)) add Emp)"
by (blast intro:Ang_weektrans Eq_rev)
  have P7 : "Eq (Geos (Ang (An B C A)) add Emp) (Geos (Ang (An A C B)) add Emp)" by (simp
add:Ang_roll)
  from P6 P7 show "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A C B)) add Emp)" by
(blast intro:Ang_weektrans Eq_rev Ang_rev)
qed

lemma (in Congruence_Rule) Tri_week_ASA :
  assumes N :
    "Def (Tri (Tr A B C))"
    "Def (Tri (Tr A1 B1 C1))"
    "Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se A1 B1)) add Emp)"
    "Cong (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An C1 A1 B1)) add Emp)"
    "Cong (Geos (Ang (An C B A)) add Emp) (Geos (Ang (An C1 B1 A1)) add Emp)"
  shows "¬¬ Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 C1)) add Emp)"
proof
  assume W : "¬ Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 C1)) add Emp)"
  have P1 : "Line_on (Li B1 C1) B1" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li B1 C1) C1" by (simp add:Line_on_rule)
  from assms have P3 : "¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi C1) add Emp)" by (simp add:Tri_def)
  from assms have P4 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)" by (simp add:Tri_def)
  from P1 P2 P3 P4 have "∃p. Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 p)) add Emp)
    ∧ ¬ Bet_Point (Se p C1) B1 ∧ Line_on (Li B1 C1) p ∧ ¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi
p) add Emp)" by (simp add:Seg_move_sameside)
  then obtain D1 :: Point where P5 : "Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 D1)) add
Emp)
    ∧ ¬ Bet_Point (Se D1 C1) B1 ∧ Line_on (Li B1 C1) D1 ∧ ¬ Eq (Geos (Poi B1) add Emp) (Geos
(Poi D1) add Emp)" by blast
  from W have P6 : "¬ Eq (Geos (Seg (Se B1 C1)) add Emp) (Geos (Seg (Se B C)) add Emp)" by (blast
intro:Eq_rev)
  from P5 P6 have "¬ Eq (Geos (Seg (Se B1 C1)) add Emp) (Geos (Seg (Se B1 D1)) add Emp)" by
(simp add:Eq_not_trans)
  then have P7 : "¬ Eq (Geos (Poi C1) add Emp) (Geos (Poi D1) add Emp)"
    by (simp add:Seg_not_Eq_Point)
  from assms have P8 : "¬ Line_on (Li B1 C1) A1" by (simp add:Tri_def_Line)
  from P5 have P9 : "Line_on (Li B1 C1) D1" by simp
  then have P10 : "Eq (Geos (Poi D1) add Emp) (Geos (Poi A1) add Emp) ⟹ Line_on (Li B1 C1) A1"
by (simp add:Point_Eq)
  from P8 P10 have P11 : "¬ Eq (Geos (Poi D1) add Emp) (Geos (Poi A1) add Emp)" by blast
  from assms have P12 : "¬ Eq (Geos (Poi A1) add Emp) (Geos (Poi B1) add Emp)" by (simp
add:Tri_def)
  have P13 : "Line_on (Li A1 B1) B1" by (simp add:Line_on_rule)
  from P5 have P14 : "¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi D1) add Emp)" by simp
  from assms P7 P9 P14 have "Def (Tri (Tr A1 B1 D1))" by (blast intro:Tri_def_extension)
  then have "Def (Tri (Tr D1 B1 A1))" by (simp add:Tri_def_rev)
  then have P15 : "Def (Tri (Tr B1 A1 D1))" by (simp add:Tri_def_trans)
  from assms have P16 : "Def (Tri (Tr B A C))" by (blast intro:Tri_def_rev Tri_def_trans)
  from assms have P17 : "Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se B1 A1)) add Emp)" by
(blast intro:Seg_rev Eq_trans)
  from P5 have P18 : "Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 D1)) add Emp)" by simp
  have P19 : "Line_on (Li B1 A1) A1" by (simp add:Line_on_rule)
  from assms have "Def (Tri (Tr C1 B1 A1))" by (simp add:Tri_def_rev)
  then have P20 : "Def (Ang (An C1 B1 A1))" by (simp add:Tri_to_Ang)
  have P21 : "Line_on (Li B1 A1) A1" by (simp add:Line_on_rule)
  have P22 : "¬ Bet_Point (Se A1 A1) B1" by (simp add:Bet_end_Point)
  from P5 have P23 : "¬ Bet_Point (Se C1 D1) B1" by (blast intro:Bet_rev)
```

from P12 have P24 : "¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi A1) add Emp)" by (blast intro:Eq_rev)
from P9 P14 P19 P20 P21 P22 P23 P24 have P25 : "Eq (Geos (Ang (An C1 B1 A1)) add Emp) (Geos (Ang (An D1 B1 A1)) add Emp)" by (simp add:Ang_Point_swap)
from assms P25 have P26 : "Cong (Geos (Ang (An C B A)) add Emp) (Geos (Ang (An D1 B1 A1)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
have P27 : "Eq (Geos (Ang (An C B A)) add Emp) (Geos (Ang (An A B C)) add Emp)" by (simp add:Ang_roll)
from P26 P27 have P28 : "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An D1 B1 A1)) add Emp)" by (blast intro:Ang_weektrans Eq_rev)
have P29 : "Eq (Geos (Ang (An D1 B1 A1)) add Emp) (Geos (Ang (An A1 B1 D1)) add Emp)" by (simp add:Ang_roll)
from P28 P29 have P30 : "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A1 B1 D1)) add Emp)" by (blast intro:Ang_weektrans Eq_rev Ang_rev)
from P15 P16 P17 P18 P30 have P31 : "Cong (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An D1 A1 B1)) add Emp)" by (simp add:Tri_week_SAS)
from P1 P2 P3 P5 P8 P23 have P32 : "Plane_sameside (Li A1 B1) C1 D1 ∨ Eq (Geos (Poi C1) add Emp) (Geos (Poi D1) add Emp)" by (simp add:Seg_Plane_sameside)
from assms P31 have P33 : "Plane_sameside (Li A1 B1) C1 D1 ⟹ Eq (Geos (Lin (Li C1 A1)) add Emp) (Geos (Lin (Li D1 A1)) add Emp)" by (simp add:Ang_move_unique)
from assms have "¬ Eq (Geos (Poi C1) add Emp) (Geos (Poi A1) add Emp)" by (simp add:Tri_def)
then have P34 : "Eq (Geos (Lin (Li C1 A1)) add Emp) (Geos (Lin (Li A1 C1)) add Emp)" by (simp add:Line_rev)
from P11 have P35 : "Eq (Geos (Lin (Li D1 A1)) add Emp) (Geos (Lin (Li A1 D1)) add Emp)" by (simp add:Line_rev)
from P33 P34 P35 have P36 : "Plane_sameside (Li A1 B1) C1 D1 ⟹ Eq (Geos (Lin (Li A1 C1)) add Emp) (Geos (Lin (Li A1 D1)) add Emp)" by (blast intro:Eq_trans Eq_rev)
have P37 : "Line_on (Li A1 C1) C1" by (simp add:Line_on_rule)
have P38 : "Line_on (Li A1 C1) A1" by (simp add:Line_on_rule)
have P39 : "Line_on (Li A1 D1) D1" by (simp add:Line_on_rule)
have P40 : "Line_on (Li A1 D1) A1" by (simp add:Line_on_rule)
from P37 have P41 : "Eq (Geos (Poi C1) add Emp) (Geos (Poi D1) add Emp) ⟹ Line_on (Li A1 C1) D1" by (simp add:Point_Eq)
from P11 P38 P39 P40 P41 have P42 : "Eq (Geos (Poi C1) add Emp) (Geos (Poi D1) add Emp) ⟹
Eq (Geos (Lin (Li A1 C1)) add Emp) (Geos (Lin (Li A1 D1)) add Emp)" by (simp add:Line_unique)
from P32 P36 P42 have P43 : "Eq (Geos (Lin (Li A1 C1)) add Emp) (Geos (Lin (Li A1 D1)) add Emp)" by blast
from P2 P7 P9 P37 have P44 : "Line_on (Li A1 C1) D1 ⟹ Eq (Geos (Lin (Li B1 C1)) add Emp) (Geos (Lin (Li A1 C1)) add Emp)" by (simp add:Line_unique)
from P1 P44 have P45 : "Line_on (Li A1 C1) D1 ⟹ Line_on (Li A1 C1) B1"
by (simp add:Line_on_trans)
from assms have "Def (Tri (Tr C1 B1 A1))" by (simp add:Tri_def_rev)
then have P46 : "¬ Line_on (Li A1 C1) B1" by (simp add:Tri_def_Line)
from P45 P46 have P47 : "¬ Line_on (Li A1 C1) D1" by blast
have P48 : "Line_on (Li A1 D1) D1" by (simp add:Line_on_rule)
from P47 P48 have P49 : "¬ Eq (Geos (Lin (Li A1 C1)) add Emp) (Geos (Lin (Li A1 D1)) add Emp)" by (simp add:Line_not_on_Eq)
from P43 P49 show False by blast
qed

theorem (in Congruence_Rule) Tri_SAS:
assumes
"Def (Tri (Tr A B C))"
"Def (Tri (Tr A1 B1 C1))"
"Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se A1 B1)) add Emp)"
"Eq (Geos (Seg (Se A C)) add Emp) (Geos (Seg (Se A1 C1)) add Emp)"
"Cong (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An B1 A1 C1)) add Emp)"
shows "Cong (Geos (Tri (Tr A B C)) add Emp) (Geos (Tri (Tr A1 B1 C1)) add Emp)"
proof -
from assms have P1 : "Cong (Geos (Ang (An C B A)) add Emp) (Geos (Ang (An C1 B1 A1)) add Emp)" by (simp add:Tri_week_SAS)
have P2 : "Eq (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An C A B)) add Emp)" by (simp add:Ang_roll)
have P3 : "Eq (Geos (Ang (An C1 A1 B1)) add Emp) (Geos (Ang (An B1 A1 C1)) add Emp)" by (simp add:Ang_roll)
from assms P2 have P4 : "Cong (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An B1 A1 C1)) add

```
Emp)" by (blast intro:Ang_weektrans Eq_rev)
  from P3 P4 have P5 : "Cong (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An C1 A1 B1)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from assms P1 P5 have P6 : "¬¬ Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 C1)) add
Emp)" by (simp add:Tri_week_ASA)
  from assms have P7 : "Eq (Geos (Seg (Se C A)) add Emp) (Geos (Seg (Se C1 A1)) add Emp)" by
(blast intro:Seg_rev Eq_rev Eq_trans)
  from assms have P8 : "Def (Tri (Tr A C B))" by (blast intro:Tri_def_rev Tri_def_trans)
  from assms have P9 : "Def (Tri (Tr A1 C1 B1))" by (blast intro:Tri_def_rev Tri_def_trans)
  from assms P5 P8 P9 have P10 : "Cong (Geos (Ang (An B C A)) add Emp) (Geos (Ang (An B1 C1
A1)) add Emp)" by (simp add:Tri_week_SAS)
  have P11 : "Eq (Geos (Ang (An B C A)) add Emp) (Geos (Ang (An A C B)) add Emp)" by (simp
add:Ang_roll)
  have P12 : "Eq (Geos (Ang (An B1 C1 A1)) add Emp) (Geos (Ang (An A1 C1 B1)) add Emp)" by
(simp add:Ang_roll)
  from P10 P11 have P13 : "Cong (Geos (Ang (An A C B)) add Emp) (Geos (Ang (An B1 C1 A1)) add
Emp)" by (blast intro:Ang_weektrans Eq_rev)
  from P10 P12 P13 have P14 : "Cong (Geos (Ang (An A C B)) add Emp) (Geos (Ang (An A1 C1 B1))
add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from assms P1 P6 P7 P14 show "Cong (Geos (Tri (Tr A B C)) add Emp) (Geos (Tri (Tr A1 B1 C1))
add Emp)" by (simp add:Tri_Cong_def)
qed

theorem (in Congruence_Rule) Tri_ASA:
  assumes
    "Def (Tri (Tr A B C))"
    "Def (Tri (Tr A1 B1 C1))"
    "Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se A1 B1)) add Emp)"
    "Cong (Geos (Ang (An C B A)) add Emp) (Geos (Ang (An C1 B1 A1)) add Emp)"
    "Cong (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An C1 A1 B1)) add Emp)"
  shows "Cong (Geos (Tri (Tr A B C)) add Emp) (Geos (Tri (Tr A1 B1 C1)) add Emp)"
proof -
  from assms have P1 : "Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se B1 A1)) add Emp)" by
(blast intro:Seg_rev Eq_rev Eq_trans)
  from assms have P2 : "Def (Tri (Tr B A C))" by (blast intro:Tri_def_rev Tri_def_trans)
  from assms have P3 : "Def (Tri (Tr B1 A1 C1))" by (blast intro:Tri_def_rev Tri_def_trans)
  from assms P1 P2 P3 have P4 : "¬¬ Eq (Geos (Seg (Se A C)) add Emp) (Geos (Seg (Se A1 C1)) add
Emp)" by (simp add:Tri_week_ASA)
  have P5 : "Eq (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An B A C)) add Emp)" by (simp
add:Ang_roll)
  have P6 : "Eq (Geos (Ang (An C1 A1 B1)) add Emp) (Geos (Ang (An B1 A1 C1)) add Emp)" by
(simp add:Ang_roll)
  from assms P5 have P7 : "Cong (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An C1 A1 B1)) add
Emp)" by (blast intro:Ang_weektrans Eq_rev)
  from P6 P7 have P8 : "Cong (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An B1 A1 C1)) add
Emp)" by (blast intro:Ang_weektrans Eq_rev Ang_rev)
  from assms P1 P4 P8 show "Cong (Geos (Tri (Tr A B C)) add Emp) (Geos (Tri (Tr A1 B1 C1)) add
Emp)" by (simp add:Tri_SAS)
qed

theorem (in Congruence_Rule) Ang_complementary :
  assumes
    "Def (Ang (An A B C))"
    "Def (Ang (An A1 B1 C1))"
    "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A1 B1 C1)) add Emp)"
    "Bet_Point (Se A D) B"
    "Bet_Point (Se A1 D1) B1"
  shows
    "Cong (Geos (Ang (An C B D)) add Emp) (Geos (Ang (An C1 B1 D1)) add Emp)"
proof -
  from assms have "∃p. Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An p B1 C1)) add Emp)
    ∧ Eq (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An p B1 C1)) add Emp)
    ∧ Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se B1 p)) add Emp)
    ∧ Line_on (Li B1 A1) p ∧ ¬ Bet_Point (Se p A1) B1 ∧ Def (Ang (An p B1 C1))" by (simp
add:Ang_replace)
  then obtain A2 :: Point where P1 : "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A2 B1
```

```
C1)) add Emp)
      ∧ Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se B1 A2)) add Emp) ∧ Line_on (Li B1 A1) A2
      ∧ ¬ Bet_Point (Se A2 A1) B1 ∧ Def (Ang (An A2 B1 C1))" by blast
  from assms P1 have "∃p. Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A2 B1 p)) add
Emp)
      ∧ Eq (Geos (Ang (An A2 B1 C1)) add Emp) (Geos (Ang (An A2 B1 p)) add Emp)
      ∧ Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 p)) add Emp)
      ∧ Line_on (Li B1 C1) p ∧ ¬ Bet_Point (Se p C1) B1 ∧ Def (Ang (An A2 B1 p))" by (simp
add:Ang_replace)
  then obtain C2 :: Point where P2 : "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A2 B1
C2)) add Emp)
      ∧ Eq (Geos (Seg (Se B C)) add Emp) (Geos (Seg (Se B1 C2)) add Emp) ∧ Line_on (Li B1 C1) C2
      ∧ ¬ Bet_Point (Se C2 C1) B1 ∧ Def (Ang (An A2 B1 C2))" by blast
  from assms have "Def (Tri (Tr A B C))" by (simp add:Ang_to_Tri)
  then have P3 : "Def (Tri (Tr B A C))" by (blast intro:Tri_def_rev Tri_def_trans)
  from P2 have "Def (Tri (Tr A2 B1 C2))" by (simp add:Ang_to_Tri)
  then have P4 : "Def (Tri (Tr B1 A2 C2))" by (blast intro:Tri_def_rev Tri_def_trans)
  from P1 P2 P3 P4 have P5 : "Cong (Geos (Tri (Tr B A C)) add Emp) (Geos (Tri (Tr B1 A2 C2)) add
Emp)" by (simp add:Tri_SAS)
  then have P6 : "Eq (Geos (Seg (Se A C)) add Emp) (Geos (Seg (Se A2 C2)) add Emp)" by (simp
add:Tri_Cong_def)
  from P5 have P7 : "Cong (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An C2 A2 B1)) add Emp)"
by (simp add:Tri_Cong_def)
  have P8 : "Line_on (Li B1 D1) B1" by (simp add:Line_on_rule)
  from assms have P9 : "Line_on (Li A1 D1) B1" by (simp add:Line_Bet_on)
  from assms have p10 : "¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi A1) add Emp)" by (simp
add:Bet_Point_def)
  from assms have P11 : "Line_on (Li B1 A1) D1" by (simp add:Line_Bet_on)
  have P12 : "Line_on (Li B1 D1) D1" by (simp add:Line_on_rule)
  have P13 : "Line_on (Li B1 A1) B1" by (simp add:Line_on_rule)
  from assms have P14 : "¬ Eq (Geos (Poi D1) add Emp) (Geos (Poi B1) add Emp)" by (simp
add:Bet_Point_def)
  from P8 P11 P12 P13 P14 have P15 : "Eq (Geos (Lin (Li B1 A1)) add Emp) (Geos (Lin (Li B1 D1))
add Emp)" by (simp add:Line_unique)
  from P1 P15 have P16 : "Line_on (Li B1 D1) A2" by (simp add:Line_on_trans)
  from P4 have P17 : "¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi A2) add Emp)" by (simp add:Tri_def)
  from assms have P18 : "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi B) add Emp)"
    by (simp add:Bet_Point_def)
  then have P19 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)" by (blast intro:Eq_rev)
  from P8 P16 P17 P19 have "∃p. Eq (Geos (Seg (Se B D)) add Emp) (Geos (Seg (Se B1 p)) add Emp)
    ∧ Bet_Point (Se p A2) B1 ∧ Line_on (Li B1 D1) p ∧ ¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi p)
add Emp)" by (simp add:Seg_move_diffside)
  then obtain D2 :: Point where P20 : "Eq (Geos (Seg (Se B D)) add Emp) (Geos (Seg (Se B1 D2)) add
Emp)
    ∧ Bet_Point (Se D2 A2) B1 ∧ Line_on (Li B1 D1) D2 ∧ ¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi
D2) add Emp)" by blast
  from P1 have P21 : "Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se A2 B1)) add Emp)" by (blast
intro:Seg_rev Eq_trans)
  from P20 have P22 : "Bet_Point (Se A2 D2) B1" by (blast intro:Bet_rev)
  from assms P20 P21 P22 have P23 : "Eq (Geos (Seg (Se A D)) add Emp) (Geos (Seg (Se A2 D2)) add
Emp)" by (blast intro:Seg_Bet_add)
  from P3 have P24 : "Def (Tri (Tr C A B))" by (blast intro:Tri_def_rev)
  from assms have P25 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)"
    by (simp add:Bet_Point_def)
  from assms have P26 : "Line_on (Li A B) D" by (simp add:Line_Bet_on)
  from P24 P25 P26 have P27 : "Def (Tri (Tr C A D))" by (simp add:Tri_def_extension)
  from P4 have P28 : "Def (Tri (Tr C2 A2 B1))" by (blast intro:Tri_def_rev)
  from P22 have P29 : "¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi D2) add Emp)"
    by (simp add:Bet_Point_def)
  from P22 have P30 : "Line_on (Li A2 B1) D2" by (simp add:Line_Bet_on)
  from P28 P29 P30 have P31 : "Def (Tri (Tr C2 A2 D2))" by (simp add:Tri_def_extension)
  from P24 have P32 : "Def (Ang (An C A B))" by (simp add:Tri_to_Ang)
  from P27 have P33 : "Def (Ang (An C A D))" by (simp add:Tri_to_Ang)
  have P34 : "Line_on (Li A C) C" by (simp add:Line_on_rule)
  have P35 : "¬ Bet_Point (Se C C) A" by (simp add:Bet_end_Point)
  from assms have "Inv (Bet_Point (Se D B) A)" by (simp add:Bet_iff)
```

then have "¬ Bet_Point (Se D B) A" by (simp add:Inv_def)
then have P36 : "¬ Bet_Point (Se B D) A" by (blast intro:Bet_rev)
from P33 have "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)" by (simp add:Ang_def)
then have P37 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)" by (blast intro:Eq_rev)
from P25 P26 P32 P33 P34 P35 P36 P37 have P38 : "Eq (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An C A D)) add Emp)" by (simp add:Ang_Point_swap)
from P28 have P39 : "Def (Ang (An C2 A2 B1))" by (simp add:Tri_to_Ang)
from P31 have P40 : "Def (Ang (An C2 A2 D2))" by (simp add:Tri_to_Ang)
have P41 : "Line_on (Li A2 C2) C2" by (simp add:Line_on_rule)
have P42 : "¬ Bet_Point (Se C2 C2) A2" by (simp add:Bet_end_Point)
from P20 have "Inv (Bet_Point (Se B1 D2) A2)" by (simp add:Bet_iff)
then have P43 : "¬ Bet_Point (Se B1 D2) A2" by (simp add:Inv_def)
from P40 have "¬ Eq (Geos (Poi C2) add Emp) (Geos (Poi A2) add Emp)" by (simp add:Ang_def)
then have P44 : "¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi C2) add Emp)" by (blast intro:Eq_rev)
from P29 P30 P39 P40 P41 P42 P43 P44 have P45 : "Eq (Geos (Ang (An C2 A2 B1)) add Emp) (Geos (Ang (An C2 A2 D2)) add Emp)" by (simp add:Ang_Point_swap)
from P7 P38 have P46 : "Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An C2 A2 B1)) add Emp)" by (blast intro:Ang_weektrans Eq_rev)
from P45 P46 have P47 : "Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An C2 A2 D2)) add Emp)" by (blast intro:Ang_weektrans Eq_rev Ang_rev)
from P27 have P48 : "Def (Tri (Tr A C D))" by (blast intro:Tri_def_trans Tri_def_rev)
from P31 have P49 : "Def (Tri (Tr A2 C2 D2))" by (blast intro:Tri_def_trans Tri_def_rev)
from P6 P23 P47 P48 P49 have P50 : "Cong (Geos (Tri (Tr A C D)) add Emp) (Geos (Tri (Tr A2 C2 D2)) add Emp)" by (simp add:Tri_SAS)
then have P51 : "Cong (Geos (Ang (An A D C)) add Emp) (Geos (Ang (An A2 D2 C2)) add Emp)" by (simp add:Tri_Cong_def)
from P50 have "Eq (Geos (Seg (Se C D)) add Emp) (Geos (Seg (Se C2 D2)) add Emp)" by (simp add:Tri_Cong_def)
then have P52 : "Eq (Geos (Seg (Se D C)) add Emp) (Geos (Seg (Se D2 C2)) add Emp)" by (blast intro:Seg_rev Eq_trans)
from assms have P53 : "Line_on (Li D A) B" by (simp add:Bet_rev Line_Bet_on)
from assms have "Inv (Bet_Point (Se B A) D)" by (simp add:Bet_iff)
then have "¬ Bet_Point (Se B A) D" by (simp add:Inv_def)
then have P54 : "¬ Bet_Point (Se A B) D" by (blast intro:Bet_rev)
have P55 : "Line_on (Li D C) C" by (simp add:Line_on_rule)
have P56 : "¬ Bet_Point (Se C C) D" by (simp add:Bet_end_Point)
from P48 have P57 : "Def (Ang (An A D C))" by (simp add:Tri_to_Ang Ang_def_inv)
from P57 have P58 : "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi C) add Emp)" by (simp add:Ang_def)
from P18 P53 P54 P55 P56 P57 P58 have P59 : "Eq (Geos (Ang (An A D C)) add Emp) (Geos (Ang (An B D C)) add Emp) ∧ Def (Ang (An B D C))" by (simp add:Ang_Point_swap)
from P22 have P60 : "Line_on (Li D2 A2) B1" by (simp add:Line_Bet_on)
from P20 have "Inv (Bet_Point (Se A2 B1) D2)" by (simp add:Bet_iff)
then have P61 : "¬ Bet_Point (Se A2 B1) D2" by (simp add:Inv_def)
have P62 : "Line_on (Li D2 C2) C2" by (simp add:Line_on_rule)
have P63 : "¬ Bet_Point (Se C2 C2) D2" by (simp add:Bet_end_Point)
from P49 have P64 : "Def (Ang (An A2 D2 C2))" by (simp add:Tri_to_Ang Ang_def_inv)
from P20 have P65 : "¬ Eq (Geos (Poi D2) add Emp) (Geos (Poi B1) add Emp)" by (blast intro:Eq_rev)
from P64 have P66 : "¬ Eq (Geos (Poi D2) add Emp) (Geos (Poi C2) add Emp)" by (simp add:Ang_def)
from P60 P61 P62 P63 P64 P65 P66 have P67 : "Eq (Geos (Ang (An A2 D2 C2)) add Emp) (Geos (Ang (An B1 D2 C2)) add Emp) ∧ Def (Ang (An B1 D2 C2))" by (simp add:Ang_Point_swap)
from P51 P59 have P68 : "Cong (Geos (Ang (An B D C)) add Emp) (Geos (Ang (An A2 D2 C2)) add Emp)" by (blast intro:Ang_weektrans Eq_rev)
from P67 P68 have P69 : "Cong (Geos (Ang (An B D C)) add Emp) (Geos (Ang (An B1 D2 C2)) add Emp)" by (blast intro:Ang_weektrans Eq_rev Ang_rev)
from P59 have "Def (Tri (Tr B D C))" by (simp add:Ang_to_Tri)
then have P70 : "Def (Tri (Tr D B C))" by (blast intro:Tri_def_trans Tri_def_rev)
from P67 have "Def (Tri (Tr B1 D2 C2))" by (simp add:Ang_to_Tri)
then have P71 : "Def (Tri (Tr D2 B1 C2))" by (blast intro:Tri_def_trans Tri_def_rev)
from P20 have "Eq (Geos (Seg (Se B D)) add Emp) (Geos (Seg (Se B1 D2)) add Emp)" by simp
then have P72 : "Eq (Geos (Seg (Se D B)) add Emp) (Geos (Seg (Se D2 B1)) add Emp)" by (blast intro:Seg_rev Eq_trans)
from P52 P69 P70 P71 P72 have "Cong (Geos (Tri (Tr D B C)) add Emp) (Geos (Tri (Tr D2 B1 C2)) add Emp)" by (simp add:Tri_SAS)
then have P73 : "Cong (Geos (Ang (An C B D)) add Emp) (Geos (Ang (An C2 B1 D2)) add Emp)" by (simp add:Tri_Cong_def)

from P71 have ”Def (Tri (Tr C2 B1 D2))” by (blast intro:Tri_def_rev)
then have P74 : ”Def (Ang (An C2 B1 D2))” by (simp add:Tri_to_Ang)
from assms have P75 : ”¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi C1) add Emp)” by (simp add:Ang_def)
from P71 have P76 : ”¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi C2) add Emp)” by (simp add:Tri_def)
from P2 P75 P76 have P77 : ”Line_on (Li B1 C2) C1” by (simp add:Line_on_rev)
from P14 have P78 : ”¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi D1) add Emp)” by (blast intro:Eq_rev)
from P74 have P79 : ”¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi D2) add Emp)” by (simp add:Ang_def)
from P20 P78 P79 have P80 : ”Line_on (Li B1 D2) D1” by (simp add:Line_on_rev)
from assms have ”¬ Eq (Geos (Lin (Li B1 A1)) add Emp) (Geos (Lin (Li B1 C1)) add Emp)” by (simp add:Ang_def)
then have P81 : ”¬ Eq (Geos (Lin (Li B1 C1)) add Emp) (Geos (Lin (Li B1 A1)) add Emp)” by (blast intro:Eq_rev)
have P82 : ”Line_on (Li A1 D1) D1” by (simp add:Line_on_rule)
from P9 P11 P13 P14 P82 have P83 : ”Eq (Geos (Lin (Li B1 A1)) add Emp) (Geos (Lin (Li A1 D1)) add Emp)” by (simp add:Line_unique)
from P81 P83 have P84 : ”¬ Eq (Geos (Lin (Li B1 C1)) add Emp) (Geos (Lin (Li A1 D1)) add Emp)” by (simp add:Eq_not_trans)
then have P85 : ”¬ Eq (Geos (Lin (Li A1 D1)) add Emp) (Geos (Lin (Li B1 C1)) add Emp)” by (blast intro:Eq_rev)
have P86 : ”Line_on (Li B1 C1) B1” by (simp add:Line_on_rule)
from assms P85 P86 have P87 : ”Plane_diffside (Li B1 C1) A1 D1” by (simp add:Plane_Bet_diffside)
then have P88 : ”Plane_diffside (Li B1 C1) D1 A1” by (simp add:Plane_diffside_rev)
have P89 : ”Bet_Point (Se D2 D1) B1 ⟹ Line_on (Li D2 D1) B1” by (simp add:Line_Bet_on)
have P90 : ”Line_on (Li D2 D1) D1” by (simp add:Line_on_rule)
from P9 P14 P82 P89 P90 have P91 : ”Bet_Point (Se D2 D1) B1 ⟹ Eq (Geos (Lin (Li A1 D1)) add Emp) (Geos (Lin (Li D2 D1)) add Emp)” by (simp add:Line_unique)
from P84 P91 have ”Bet_Point (Se D2 D1) B1 ⟹ ¬ Eq (Geos (Lin (Li B1 C1)) add Emp) (Geos (Lin (Li D2 D1)) add Emp)” by (simp add:Eq_not_trans)
then have P92 : ”Bet_Point (Se D2 D1) B1 ⟹ ¬ Eq (Geos (Lin (Li D2 D1)) add Emp) (Geos (Lin (Li B1 C1)) add Emp)” by (blast intro:Eq_rev)
from P86 P92 have ”Bet_Point (Se D2 D1) B1 ⟹ Plane_diffside (Li B1 C1) D2 D1” by (simp add:Plane_Bet_diffside)
then have P93 : ”Bet_Point (Se D2 D1) B1 ⟹ Plane_diffside (Li B1 C1) D1 D2”
by (simp add:Plane_diffside_rev)
from P20 have ”Eq (Geos (Poi D2) add Emp) (Geos (Poi A1) add Emp) ⟹ Bet_Point (Se A1 A2) B1” by (blast intro:Bet_Point_Eq)
then have P94 : ”Eq (Geos (Poi D2) add Emp) (Geos (Poi A1) add Emp) ⟹ Bet_Point (Se A2 A1) B1” by (simp add:Bet_rev)
from P1 P94 have P95 : ”¬ Eq (Geos (Poi A1) add Emp) (Geos (Poi D2) add Emp)” by (blast intro:Eq_rev)
from P88 P93 P95 have ”Bet_Point (Se D2 D1) B1 ⟹ Plane_sameside (Li B1 C1) A1 D2” by (blast intro:Plane_trans_inv)
then have P96 : ”Bet_Point (Se D2 D1) B1 ⟹ Plane_sameside (Li B1 C1) D2 A1” by (simp add:Plane_sameside_rev)
from P1 have ”Def (Tri (Tr A2 B1 C1))” by (simp add:Ang_to_Tri)
then have P97 : ”¬ Line_on (Li B1 C1) A2” by (simp add:Tri_def_Line)
have ”Line_on (Li D2 A2) A2” by (simp add:Line_on_rule)
then have P98 : ”Eq (Geos (Lin (Li D2 A2)) add Emp) (Geos (Lin (Li B1 C1)) add Emp) ⟹ Line_on (Li B1 C1) A2” by (simp add:Line_on_trans)
from P97 P98 have P99 : ”¬ Eq (Geos (Lin (Li D2 A2)) add Emp) (Geos (Lin (Li B1 C1)) add Emp)” by blast
from P20 have P100 : ”Bet_Point (Se D2 A2) B1” by simp
from P86 P99 P100 have P101 : ”Plane_diffside (Li B1 C1) D2 A2” by (simp add:Plane_Bet_diffside)
from P96 P101 have ”¬ Eq (Geos (Poi A1) add Emp) (Geos (Poi A2) add Emp) ⟹ Bet_Point (Se D2 D1) B1 ⟹
Plane_diffside (Li B1 C1) A1 A2” by (simp add:Plane_trans)
then have P102 : ”¬ Eq (Geos (Poi A1) add Emp) (Geos (Poi A2) add Emp) ⟹ Bet_Point (Se D2 D1) B1 ⟹
¬ Plane_sameside (Li B1 C1) A1 A2” by (simp add:Plane_diffside_not_sameside)
have P103 : ”Line_on (Li B1 A1) A1” by (simp add:Line_on_rule)
from assms have ”Def (Tri (Tr C1 B1 A1))” by (simp add:Ang_to_Tri Tri_def_rev)
then have P104 : ”¬ Line_on (Li B1 A1) C1” by (simp add:Tri_def_Line)
from P4 have P105 : ”¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi A2) add Emp)” by (simp add:Tri_def)
from P1 p10 P13 P103 P104 P105 have P106 : ”Plane_sameside (Li C1 B1) A2 A1 ∨ Eq (Geos (Poi

```
A2) add Emp) (Geos (Poi A1) add Emp)" by (simp add:Seg_Plane_sameside)
  from assms have "¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi C1) add Emp)" by (simp add:Ang_def)
  then have P107 : "Eq (Geos (Lin (Li C1 B1)) add Emp) (Geos (Lin (Li B1 C1)) add Emp)" by (simp
add:Line_rev Eq_rev)
  from P106 P107 have P108 : "Plane_sameside (Li B1 C1) A1 A2 ∨ Eq (Geos (Poi A1) add Emp)
(Geos (Poi A2) add Emp)" by (blast intro:Plane_sameside_rev Plane_Line_trans Eq_rev)
  from P102 P108 have P109 : "¬ Eq (Geos (Poi A1) add Emp) (Geos (Poi A2) add Emp) ⟹ ¬
Bet_Point (Se D2 D1) B1" by blast
  from P22 have P110 : "Eq (Geos (Poi A1) add Emp) (Geos (Poi A2) add Emp) ⟹ Bet_Point (Se A1
D2) B1" by (blast intro:Bet_Point_Eq Eq_rev)
  then have P111 : "Eq (Geos (Poi A1) add Emp) (Geos (Poi A2) add Emp) ⟹ Line_on (Li A1 D2)
B1" by (simp add:Line_Bet_on)
  have P112 : "Line_on (Li A1 D2) A1" by (simp add:Line_on_rule)
  have P113 : "Line_on (Li A1 D1) A1" by (simp add:Line_on_rule)
  from p10 have P114 : "¬ Eq (Geos (Poi A1) add Emp) (Geos (Poi B1) add Emp)" by (blast in-
tro:Eq_rev)
  from P9 P111 P112 P113 P114 have P115 : "Eq (Geos (Poi A1) add Emp) (Geos (Poi A2) add Emp)
⟹ Eq (Geos (Lin (Li A1 D1)) add Emp) (Geos (Lin (Li A1 D2)) add Emp)" by (simp add:Line_unique)
  from P84 P115 have "Eq (Geos (Poi A1) add Emp) (Geos (Poi A2) add Emp) ⟹ ¬ Eq (Geos (Lin
(Li B1 C1)) add Emp) (Geos (Lin (Li A1 D2)) add Emp)" by (simp add:Eq_not_trans)
  then have P116 : "Eq (Geos (Poi A1) add Emp) (Geos (Poi A2) add Emp) ⟹ ¬ Eq (Geos (Lin (Li
A1 D2)) add Emp) (Geos (Lin (Li B1 C1)) add Emp)" by (blast intro:Eq_rev)
  from P86 P110 P116 have P117 : "Eq (Geos (Poi A1) add Emp) (Geos (Poi A2) add Emp) ⟹
Plane_diffside (Li B1 C1) A1 D2" by (simp add:Plane_Bet_diffside)
  have "Eq (Geos (Poi A1) add Emp) (Geos (Poi A2) add Emp) ⟹ Bet_Point (Se D2 D1) B1 ⟹ ¬
Eq (Geos (Poi D2) add Emp) (Geos (Poi D1) add Emp)" by (simp add:Bet_Point_def)
  then have P118 : "Eq (Geos (Poi A1) add Emp) (Geos (Poi A2) add Emp) ⟹ Bet_Point (Se D2 D1)
B1 ⟹
    ¬ Eq (Geos (Poi D1) add Emp) (Geos (Poi D2) add Emp)" by (blast intro:Eq_rev)
  from P87 P117 P118 have P119 : "Eq (Geos (Poi A1) add Emp) (Geos (Poi A2) add Emp) ⟹
Bet_Point (Se D2 D1) B1 ⟹
    Plane_sameside (Li B1 C1) D1 D2" by (blast intro:Plane_trans_inv)
  from P93 have P120 : "Bet_Point (Se D2 D1) B1 ⟹ ¬ Plane_sameside (Li B1 C1) D1 D2" by (simp
add:Plane_diffside_not_sameside)
  from P119 P120 have P121 : "Eq (Geos (Poi A1) add Emp) (Geos (Poi A2) add Emp) ⟹ ¬ Bet_Point
(Se D2 D1) B1" by blast
  from P109 P121 have P122 : "¬ Bet_Point (Se D2 D1) B1" by blast
  from P2 P74 P75 P77 P78 P80 P122 have P123 : "Eq (Geos (Ang (An C2 B1 D2)) add Emp) (Geos
(Ang (An C1 B1 D1)) add Emp)" by (simp add:Ang_Point_swap)
  from P73 P123 show "Cong (Geos (Ang (An C B D)) add Emp) (Geos (Ang (An C1 B1 D1)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
qed

theorem (in Congruence_Rule) Ang_vertical :
  assumes
    "Def (Ang (An A B C))"
    "Bet_Point (Se A D) B"
    "Bet_Point (Se C E) B"
  shows "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An D B E)) add Emp)"
    and "Cong (Geos (Ang (An C B D)) add Emp) (Geos (Ang (An A B E)) add Emp)"
proof -
  have P1 : "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An C B A)) add Emp)" by (simp
add:Ang_roll)
  from assms have P2 : "Def (Ang (An C B A))" by (simp add:Ang_def_rev)
  from assms P1 P2 show "Cong (Geos (Ang (An C B D)) add Emp) (Geos (Ang (An A B E)) add
Emp)" by (simp add:Ang_complementary)
  from assms have P3 : "Line_on (Li B A) D" by (simp add:Line_Bet_on)
  from assms have "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi B) add Emp)" by (simp add:Bet_Point_def)
  then have P4 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)" by (blast intro:Eq_rev)
  from P2 P3 P4 have P5 : "Def (Ang (An C B D))" by (simp add:Ang_def_extension)
  then have P6 : "Def (Ang (An D B C))" by (simp add:Ang_def_rev)
  have P7 : "Cong (Geos (Ang (An C B D)) add Emp) (Geos (Ang (An D B C)) add Emp)" by (simp
add:Ang_roll)
  from assms have P8 : "Bet_Point (Se D A) B" by (simp add:Bet_rev)
  from assms P5 P6 P7 P8 have P9 : "Cong (Geos (Ang (An D B E)) add Emp) (Geos (Ang (An C B
A)) add Emp)" by (simp add:Ang_complementary)
```

from P1 have P10 : ”Eq (Geos (Ang (An C B A)) add Emp) (Geos (Ang (An A B C)) add Emp)” by (simp add:Ang_roll)
from P9 P10 show ”Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An D B E)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
qed

lemma (in Congruence_Rule) Ang_inside_Planeside :
assumes ”Ang_inside (An A B C) D”
shows ”Plane_diffside (Li B D) A C”
proof -
from assms have P1 : ”Plane_sameside (Li B A) C D ∧ Plane_sameside (Li B C) A D” by (simp add:Ang_inside_def)
then have P2 : ”¬ Line_on (Li B A) C” by (simp add:Plane_sameside_def)
have P3 : ”Line_on (Li B A) B” by (simp add:Line_on_rule)
then have P4 : ”Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp) ⟹ Line_on (Li B A) C” by (simp add:Point_Eq)
from P2 P4 have P5 : ”¬ Eq (Geos (Poi C) add Emp) (Geos (Poi B) add Emp)” by (blast intro:Eq_rev)
have P6 : ”Line_on (Li B C) B” by (simp add:Line_on_rule)
have P7 : ”Line_on (Li B C) C” by (simp add:Line_on_rule)
from P5 P6 P7 have ”∃p. Bet_Point (Se C p) B ∧ Line_on (Li B C) p” by (simp add:Bet_extension)
then obtain E :: Point where P8 : ”Bet_Point (Se C E) B ∧ Line_on (Li B C) E” by blast
then have P9 : ”Line_on (Li C E) B” by (simp add:Line_Bet_on)
have P10 : ”Line_on (Li C E) C” by (simp add:Line_on_rule)
from P5 P6 P7 P9 P10 have P11 : ”Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li C E)) add Emp)” by (simp add:Line_unique)
from P1 have P12 : ”¬ Line_on (Li B C) A” by (simp add:Plane_sameside_def)
from P11 P12 have P13 : ”¬ Line_on (Li C E) A” by (simp add:Line_not_on_trans)
have P14 : ”Line_on (Li B D) B” by (simp add:Line_on_rule)
from P1 have P15 : ”¬ Line_on (Li B A) D” by (simp add:Plane_sameside_def)
from P3 have P16 : ”Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp) ⟹ Line_on (Li B A) D” by (simp add:Point_Eq)
from P15 P16 have P17 : ”¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)” by blast
from P6 have P18 : ”Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp) ⟹ Line_on (Li B C) A” by (simp add:Point_Eq)
from P12 P18 have P19 : ”¬ Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp)” by blast
from P17 P19 have P20 : ”Line_on (Li B D) A ⟹ Line_on (Li B A) D” by (simp add:Line_on_rev)
from P15 P20 have P21 : ”¬ Line_on (Li B D) A” by blast
from P1 have P22 : ”¬ Line_on (Li B C) D” by (simp add:Plane_sameside_def)
from P5 have P23 : ”¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)” by (blast intro:Eq_rev)
from P17 P23 have P24 : ”Line_on (Li B D) C ⟹ Line_on (Li B C) D” by (simp add:Line_on_rev)
from P22 P24 have P25 : ”¬ Line_on (Li B D) C” by blast
from P8 have P26 : ”Bet_Point (Se C E) B” by simp
then have P27 : ”¬ Eq (Geos (Poi E) add Emp) (Geos (Poi B) add Emp)” by (simp add:Bet_Point_def)
from P8 have P28 : ”Line_on (Li B C) E” by simp
from P6 P14 P27 P28 have P29 : ”Line_on (Li B D) E ⟹ Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li B D)) add Emp)” by (simp add:Line_unique)
from P7 P29 have P30 : ”Line_on (Li B D) E ⟹ Line_on (Li B D) C” by (simp add:Line_on_trans)
from P25 P30 have P31 : ”¬ Line_on (Li B D) E” by blast
from P13 P14 P21 P25 P26 P31 have P32 : ”Line_on_Seg (Li B D) (Se C A) ∧ ¬ Line_on_Seg (Li B D) (Se E A)
∨ Line_on_Seg (Li B D) (Se E A) ∧ ¬ Line_on_Seg (Li B D) (Se C A)” by (simp add:Pachets_axiom)
have ”Line_on_Seg (Li B D) (Se E A) ⟹ ∃p. Line_on (Li B D) p ∧ Bet_Point (Se E A) p” by (simp add:Line_on_Seg_rule)
then obtain F :: Point where P33 : ”Line_on_Seg (Li B D) (Se E A) ⟹ Line_on (Li B D) F ∧ Bet_Point (Se E A) F” by blast
then have P34 : ”Line_on_Seg (Li B D) (Se E A) ⟹ Bet_Point (Se A E) F” by (simp add:Bet_rev)
from P3 have P35 : ”Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li A E)) add Emp) ⟹ Line_on (Li A E) B” by (simp add:Line_on_trans)
have P36 : ”Line_on (Li A E) E” by (simp add:Line_on_rule)
from P6 P27 P28 P35 P36 have ”Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li A E)) add Emp) ⟹
Eq (Geos (Lin (Li A E)) add Emp) (Geos (Lin (Li B C)) add Emp)” by (simp add:Line_unique)
then have P37 : ”Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li A E)) add Emp) ⟹
Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp)” by (blast intro:Eq_trans)
have P38 : ”Line_on (Li B A) A” by (simp add:Line_on_rule)
from P37 P38 have P39 : ”Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp) ⟹

Line_on (Li B C) A" by (simp add:Line_on_trans)
from P12 P39 have P40 : "¬ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp)" by blast
from P37 P40 have P41 : "¬ Eq (Geos (Lin (Li A E)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (blast intro:Eq_rev)
from P34 P38 P41 have P42 : "Line_on_Seg (Li B D) (Se E A) ⟹ Plane_sameside (Li B A) E F" by (simp add:Plane_Bet_sameside Plane_sameside_rev)
from P11 have P43 : "Eq (Geos (Lin (Li C E)) add Emp) (Geos (Lin (Li B A)) add Emp) ⟹ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp)" by (blast intro:Eq_trans)
from P40 P43 have P44 : "¬ Eq (Geos (Lin (Li C E)) add Emp) (Geos (Lin (Li B A)) add Emp)" by blast
from P3 P26 P44 have P45 : "Plane_diffside (Li B A) E C"
by (simp add:Plane_Bet_diffside Plane_diffside_rev)
from P34 have P46 : "Line_on_Seg (Li B D) (Se E A) ⟹ Line_on (Li A E) F" by (simp add:Line_Bet_on)
have P47 : "Line_on (Li C E) E" by (simp add:Line_on_rule)
have P48 : "Line_on (Li A E) A" by (simp add:Line_on_rule)
from P42 P45 have P49 : "Line_on_Seg (Li B D) (Se E A) ⟹ Plane_diffside (Li B A) F C" by (simp add:Plane_trans)
from P1 have P50 : "Plane_sameside (Li B A) C D" by (simp add:Point_Eq)
then have "Eq (Geos (Poi D) add Emp) (Geos (Poi F) add Emp) ⟹ Plane_sameside (Li B A) C F" by (simp add:Point_Eq)
then have "Eq (Geos (Poi D) add Emp) (Geos (Poi F) add Emp) ⟹ ¬ Plane_diffside (Li B A) C F" by (simp add:Plane_sameside_not_diffside)
then have P51 : "Eq (Geos (Poi D) add Emp) (Geos (Poi F) add Emp) ⟹ ¬ Plane_diffside (Li B A) F C" by (blast intro:Plane_diffside_rev)
from P49 P51 have P52 : "Line_on_Seg (Li B D) (Se E A) ⟹ ¬ Eq (Geos (Poi D) add Emp) (Geos (Poi F) add Emp)" by blast
from P49 P50 have P53 : "Line_on_Seg (Li B D) (Se E A) ⟹ Plane_diffside (Li B A) D F" by (simp add:Plane_trans Plane_diffside_rev)
from P46 have P54 : "Line_on_Seg (Li B D) (Se E A) ⟹ Eq (Geos (Poi F) add Emp) (Geos (Poi B) add Emp) ⟹
Line_on (Li A E) B" by (simp add:Point_Eq)
from P26 have P55 : "Line_on (Li C E) B" by (simp add:Line_Bet_on)
from P27 P36 P47 P54 P55 have P56 : "Line_on_Seg (Li B D) (Se E A) ⟹ Eq (Geos (Poi F) add Emp) (Geos (Poi B) add Emp) ⟹
Eq (Geos (Lin (Li A E)) add Emp) (Geos (Lin (Li C E)) add Emp)" by (simp add:Line_unique)
from P48 P56 have P57 : "Line_on_Seg (Li B D) (Se E A) ⟹ Eq (Geos (Poi F) add Emp) (Geos (Poi B) add Emp) ⟹
Line_on (Li C E) A" by (simp add:Line_on_trans)
from P13 P57 have P58 : "Line_on_Seg (Li B D) (Se E A) ⟹ ¬ Eq (Geos (Poi F) add Emp) (Geos (Poi B) add Emp)" by blast
have P59 : "Line_on (Li B D) D" by (simp add:Line_on_rule)
from P14 P17 P33 P52 P58 P59 have P60 : "Line_on_Seg (Li B D) (Se E A) ⟹ Bet_Point (Se B F) D ∨ Bet_Point (Se F D) B ∨ Bet_Point (Se D B) F" by (simp add:Bet_case)
have P61 : "Line_on (Li B F) B" by (simp add:Line_on_rule)
have P62 : "Line_on (Li B F) F" by (simp add:Line_on_rule)
from P33 have P63 : "Line_on_Seg (Li B D) (Se E A) ⟹ Line_on (Li B D) F" by simp
from P14 P58 P61 P62 P63 have "Line_on_Seg (Li B D) (Se E A) ⟹ Eq (Geos (Lin (Li B F)) add Emp) (Geos (Lin (Li B D)) add Emp)" by (simp add:Line_unique)
then have P64 : "Line_on_Seg (Li B D) (Se E A) ⟹ Eq (Geos (Lin (Li B F)) add Emp) (Geos (Lin (Li B A)) add Emp) ⟹
Eq (Geos (Lin (Li B D)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (blast intro:Eq_trans)
from P38 have P65 : "Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B D)) add Emp) ⟹ Line_on (Li B D) A" by (simp add:Line_on_trans)
from P21 P65 have P66 : "¬ Eq (Geos (Lin (Li B D)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (blast intro:Eq_rev)
from P64 P66 have P67 : "Line_on_Seg (Li B D) (Se E A) ⟹ ¬ Eq (Geos (Lin (Li B F)) add Emp) (Geos (Lin (Li B A)) add Emp)" by blast
from P3 P67 have "Line_on_Seg (Li B D) (Se E A) ⟹ Bet_Point (Se B F) D ⟹ Plane_sameside (Li B A) D F" by (simp add:Plane_Bet_sameside Plane_sameside_rev)
then have P68 : "Line_on_Seg (Li B D) (Se E A) ⟹ Bet_Point (Se B F) D ⟹ ¬ Plane_diffside (Li B A) D F" by (simp add:Plane_sameside_not_diffside)
from P53 P68 have P69 : "Line_on_Seg (Li B D) (Se E A) ⟹ ¬ Bet_Point (Se B F) D" by blast
have P70 : "Bet_Point (Se D B) F ⟹ Bet_Point (Se B D) F" by (simp add:Bet_rev)
from P3 P66 P70 have "Line_on_Seg (Li B D) (Se E A) ⟹ Bet_Point (Se D B) F ⟹ Plane_sameside (Li B A) D F" by (simp add:Plane_Bet_sameside Plane_sameside_rev)

```
  then have P71 : "Line_on_Seg (Li B D) (Se E A) ⟹ Bet_Point (Se D B) F ⟹ ¬ Plane_diffside (Li
B A) D F" by (simp add:Plane_sameside_not_diffside)
  from P53 P71 have P72 : "Line_on_Seg (Li B D) (Se E A) ⟹ ¬ Bet_Point (Se D B) F" by blast
  from P60 P69 P72 have P73 : "Line_on_Seg (Li B D) (Se E A) ⟹ Bet_Point (Se F D) B" by blast
  have "Line_on (Li F D) D" by (simp add:Line_on_rule)
  then have P74 : "Eq (Geos (Lin (Li F D)) add Emp) (Geos (Lin (Li B C)) add Emp) ⟹ Line_on (Li
B C) D" by (simp add:Line_on_trans)
  from P22 P74 have P75 : "¬ Eq (Geos (Lin (Li F D)) add Emp) (Geos (Lin (Li B C)) add Emp)" by
blast
  from P6 P73 P75 have P76 : "Line_on_Seg (Li B D) (Se E A) ⟹ Plane_diffside (Li B C) F D" by
(simp add:Plane_Bet_diffside)
  from P33 have P77 : "Line_on_Seg (Li B D) (Se E A) ⟹ Bet_Point (Se E A) F" by simp
  have "Line_on (Li E A) A" by (simp add:Line_on_rule)
  then have P78 : "Eq (Geos (Lin (Li E A)) add Emp) (Geos (Lin (Li C E)) add Emp) ⟹ Line_on (Li
C E) A" by (simp add:Line_on_trans)
  from P13 P78 have P79 : "¬ Eq (Geos (Lin (Li E A)) add Emp) (Geos (Lin (Li C E)) add Emp)" by
blast
  from P47 P77 P79 have P80 : "Line_on_Seg (Li B D) (Se E A) ⟹ Plane_sameside (Li C E) F A" by
(simp add:Plane_Bet_sameside)
  from P11 P80 have P81 : "Line_on_Seg (Li B D) (Se E A) ⟹ Plane_sameside (Li B C) F A" by (blast
intro:Eq_rev Plane_Line_trans)
  from P76 P81 have "Line_on_Seg (Li B D) (Se E A) ⟹ Plane_diffside (Li B C) A D" by (simp
add:Plane_trans)
  then have P82 : "Line_on_Seg (Li B D) (Se E A) ⟹ ¬ Plane_sameside (Li B C) A D" by (simp
add:Plane_diffside_not_sameside)
  from P1 P82 have P83 : "¬ Line_on_Seg (Li B D) (Se E A)" by blast
  from P32 P83 have "Line_on_Seg (Li B D) (Se C A)" by blast
  then have P84 : "∃p. Line_on (Li B D) p ∧ Bet_Point (Se C A) p" by (simp add:Line_on_Seg_rule)
  from P21 P25 P84 have "∃p. Bet_Point (Se C A) p ∧ Line_on (Li B D) p ∧ ¬ Line_on (Li B D) C ∧ ¬
Line_on (Li B D) A" by blast
  then have "Plane_diffside (Li B D) C A" by (simp add:Plane_diffside_def)
  thus "Plane_diffside (Li B D) A C" by (simp add:Plane_diffside_rev)
qed

lemma (in Congruence_Rule) Ang_inside_Bet_Point :
  assumes
    "Bet_Point (Se p1 p3) p2"
    "¬ Eq (Geos (Lin (Li p4 p1)) add Emp) (Geos (Lin (Li p4 p3)) add Emp)"
    "¬ Eq (Geos (Poi p4) add Emp) (Geos (Poi p1) add Emp)"
    "¬ Eq (Geos (Poi p4) add Emp) (Geos (Poi p3) add Emp)"
  shows "Ang_inside (An p1 p4 p3) p2"
proof -
  have P1 : "Line_on (Li p1 p3) p1" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li p1 p3) p3" by (simp add:Line_on_rule)
  have P3 : "Line_on (Li p4 p1) p4" by (simp add:Line_on_rule)
  have P4 : "Line_on (Li p4 p1) p1" by (simp add:Line_on_rule)
  have P5 : "Line_on (Li p4 p3) p4" by (simp add:Line_on_rule)
  have P6 : "Line_on (Li p4 p3) p3" by (simp add:Line_on_rule)
  from assms have P7 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)"
    by (simp add:Bet_Point_def)
  from P2 have P8 : "Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin (Li p4 p1)) add Emp) ⟹ Line_on
(Li p4 p1) p3" by (simp add:Line_on_trans)
  from assms P3 P5 P6 P8 have P9 : "Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin (Li p4 p1)) add
Emp) ⟹
    Eq (Geos (Lin (Li p4 p1)) add Emp) (Geos (Lin (Li p4 p3)) add Emp)" by (simp add:Line_unique)
  from assms P9 have P10 : "¬ Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin (Li p4 p1)) add Emp)"
by blast
  from P1 have P11 : "Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin (Li p4 p3)) add Emp) ⟹
Line_on (Li p4 p3) p1" by (simp add:Line_on_trans)
  from assms P3 P4 P5 P11 have P12 : "Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin (Li p4 p3))
add Emp) ⟹
    Eq (Geos (Lin (Li p4 p1)) add Emp) (Geos (Lin (Li p4 p3)) add Emp)" by (simp add:Line_unique)
  from assms P12 have P13 : "¬ Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin (Li p4 p3)) add Emp)"
by blast
  from assms P4 P10 have "Plane_sameside (Li p4 p1) p2 p3" by (simp add:Plane_Bet_sameside)
  then have P14 : "Plane_sameside (Li p4 p1) p3 p2" by (simp add:Plane_sameside_rev)
```

```
  from assms have P15 : "Bet_Point (Se p3 p1) p2" by (simp add:Bet_rev)
  from P7 have P16 : "Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin (Li p3 p1)) add Emp)" by (simp
add:Line_rev)
  from P13 P16 have P17 : "¬ Eq (Geos (Lin (Li p3 p1)) add Emp) (Geos (Lin (Li p4 p3)) add Emp)"
by (blast intro:Eq_rev Eq_trans)
  from assms P6 P15 P17 have "Plane_sameside (Li p4 p3) p2 p1" by (simp add:Plane_Bet_sameside)
  then have P18 : "Plane_sameside (Li p4 p3) p1 p2" by (simp add:Plane_sameside_rev)
  from assms have P19 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p4) add Emp)" by (blast in-
tro:Eq_rev)
  from P7 have P20 : "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p1) add Emp)" by (blast intro:Eq_rev)
  from assms P19 P20 have P21 : "Def (Ang (An p1 p4 p3))" by (simp add:Ang_def)
  from P14 P18 P21 show "Ang_inside (An p1 p4 p3) p2" by (simp add:Ang_inside_def)
qed

lemma (in Congruence_Rule) Ang_inside_HalfLine :
  assumes
    "Ang_inside (An A B C) D"
    "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi E) add Emp)"
    "Line_on (Li B D) E"
    "¬ Bet_Point (Se E D) B"
  shows
    "Ang_inside (An A B C) E"
proof -
  from assms have P1 : "Def (Ang (An A B C)) ∧ Plane_sameside (Li B A) C D ∧ Plane_sameside (Li
B C) A D" by (simp add:Ang_inside_def)
  have "Plane_diffside (Li B A) C E ⟹ ∃p. Bet_Point (Se C E) p ∧ Line_on (Li B A) p
    ∧ ¬ Line_on (Li B A) C ∧ ¬ Line_on (Li B A) E" by (simp add:Plane_diffside_def)
  then obtain p1 :: Point where "Plane_diffside (Li B A) C E ⟹ Bet_Point (Se C E) p1" by blast
  from assms P1 have "Plane_diffside (Li B A) C E ⟹ Plane_diffside (Li B A) D E" by (blast in-
tro:Plane_trans)
  then have "Plane_diffside (Li B A) C E ⟹ ∃p. Bet_Point (Se D E) p ∧ Line_on (Li B A) p
    ∧ ¬ Line_on (Li B A) D ∧ ¬ Line_on (Li B A) E" by (simp add:Plane_diffside_def)
  then obtain F :: Point where P2 : "Plane_diffside (Li B A) C E ⟹ Bet_Point (Se D E) F ∧ Line_on
(Li B A) F" by blast
  then have "Plane_diffside (Li B A) C E ⟹ Bet_Point (Se E D) F" by (simp add:Bet_rev)
  then have P3 : "Plane_diffside (Li B A) C E ⟹ Eq (Geos (Poi F) add Emp) (Geos (Poi B) add Emp)
    ⟹ Bet_Point (Se E D) B" by (simp add:Point_Eq)
  from assms P3 have P4 : "Plane_diffside (Li B A) C E ⟹ ¬ Eq (Geos (Poi F) add Emp) (Geos (Poi
B) add Emp)" by blast
  have P5 : "Line_on (Li B D) D" by (simp add:Line_on_rule)
  have P6 : "Line_on (Li E D) E" by (simp add:Line_on_rule)
  have P7 : "Line_on (Li E D) D" by (simp add:Line_on_rule)
  from assms P5 P6 P7 have P8 : "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹
    Eq (Geos (Lin (Li E D)) add Emp) (Geos (Lin (Li B D)) add Emp)" by (simp add:Line_unique)
  from P2 have P9 : "Plane_diffside (Li B A) C E ⟹ Line_on (Li E D) F" by (simp add:Line_Bet_on)
  from P8 P9 have P10 : "Plane_diffside (Li B A) C E ⟹
    ¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li B D) F" by (simp
add:Line_on_trans)
  have P11 : "Line_on (Li B A) B" by (simp add:Line_on_rule)
  have P12 : "Line_on (Li B D) B" by (simp add:Line_on_rule)
  from P2 P4 P10 P11 P12 have P13 : "Plane_diffside (Li B A) C E ⟹
    ¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ Eq (Geos (Lin (Li B D)) add Emp)
(Geos (Lin (Li B A)) add Emp)" by (blast intro:Line_unique)
  from P5 P13 have P14 : "Plane_diffside (Li B A) C E ⟹
    ¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li B A) D" by (simp
add:Line_on_trans)
  from P1 have P15 : "¬ Line_on (Li B A) D" by (simp add:Plane_sameside_def)
  from P14 P15 have P16 : "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ ¬ Plane_diffside
(Li B A) C E" by blast
  from assms P1 have "Plane_diffside (Li B C) A E ⟹ Plane_diffside (Li B C) D E" by (blast in-
tro:Plane_trans)
  then have "Plane_diffside (Li B C) A E ⟹ ∃p. Bet_Point (Se D E) p ∧ Line_on (Li B C) p
    ∧ ¬ Line_on (Li B C) D ∧ ¬ Line_on (Li B C) E" by (simp add:Plane_diffside_def)
  then obtain G :: Point where P17 : "Plane_diffside (Li B C) A E ⟹ Bet_Point (Se D E) G ∧ Line_on
(Li B C) G" by blast
  then have "Plane_diffside (Li B C) A E ⟹ Bet_Point (Se E D) G" by (simp add:Bet_rev)
```

then have P18 : ”Plane_diffside (Li B C) A E ⟹ Eq (Geos (Poi G) add Emp) (Geos (Poi B) add Emp)
⟹ Bet_Point (Se E D) B” by (simp add:Point_Eq)
from assms P18 have P19 : ”Plane_diffside (Li B C) A E ⟹ ¬ Eq (Geos (Poi G) add Emp) (Geos (Poi B) add Emp)” by blast
from P17 have P20 : ”Plane_diffside (Li B C) A E ⟹ Line_on (Li E D) G” by (simp add:Line_Bet_on)
from P8 P20 have P21 : ”Plane_diffside (Li B C) A E ⟹
¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li B D) G” by (simp add:Line_on_trans)
have P22 : ”Line_on (Li B C) B” by (simp add:Line_on_rule)
from P12 P17 P19 P21 P22 have P23 : ”Plane_diffside (Li B C) A E ⟹
¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp)
⟹ Eq (Geos (Lin (Li B D)) add Emp) (Geos (Lin (Li B C)) add Emp)” by (blast intro:Line_unique)
from P5 P23 have P24 : ”Plane_diffside (Li B C) A E ⟹
¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li B C) D” by (simp add:Line_on_trans)
from P1 have P25 : ”¬ Line_on (Li B C) D” by (simp add:Plane_sameside_def)
from P24 P25 have P26 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ ¬ Plane_diffside (Li B C) A E” by blast
from P1 have P27 : ”¬ Line_on (Li B A) C” by (simp add:Plane_sameside_def)
from P8 P12 have P28 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li E D) B” by (blast intro:Line_on_trans Eq_rev)
from assms P6 P11 P28 have P29 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li B A) E ⟹
Eq (Geos (Lin (Li E D)) add Emp) (Geos (Lin (Li B A)) add Emp)” by (simp add:Line_unique)
from P7 P29 have P30 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹
Line_on (Li B A) E ⟹ Line_on (Li B A) D” by (simp add:Line_on_trans)
from P15 P30 have P31 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹
¬ Line_on (Li B A) E” by blast
from P1 have P32 : ”¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)” by (simp add:Ang_def)
have P33 : ”Line_on (Li B C) C” by (simp add:Line_on_rule)
from P12 P22 P32 P33 have P34 : ”Line_on (Li B D) C ⟹
Eq (Geos (Lin (Li B D)) add Emp) (Geos (Lin (Li B C)) add Emp)” by (simp add:Line_unique)
from P5 P34 have P35 : ”Line_on (Li B D) C ⟹ Line_on (Li B C) D” by (simp add:Line_on_trans)
from P25 P35 have P36 : ”¬ Line_on (Li B D) C” by blast
from assms have P37 : ”Eq (Geos (Poi E) add Emp) (Geos (Poi C) add Emp) ⟹ Line_on (Li B D) C” by (simp add:Point_Eq)
from P36 P37 have P38 : ”¬ Eq (Geos (Poi C) add Emp) (Geos (Poi E) add Emp)” by (blast intro:Eq_rev)
from P16 P27 P31 P38 have P39 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹
Plane_sameside (Li B A) C E” by (simp add:Plane_not_diffside_sameside)
from P1 have P40 : ”¬ Line_on (Li B C) A” by (simp add:Plane_sameside_def)
from assms P6 P22 P28 have P41 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li B C) E ⟹
Eq (Geos (Lin (Li E D)) add Emp) (Geos (Lin (Li B C)) add Emp)” by (simp add:Line_unique)
from P7 P41 have P42 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹
Line_on (Li B C) E ⟹ Line_on (Li B C) D” by (simp add:Line_on_trans)
from P25 P42 have P43 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ ¬ Line_on (Li B C) E” by blast
from P39 have P44 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ ¬ Line_on (Li B A) E” by (simp add:Plane_sameside_def)
have ”Line_on (Li B A) A” by (simp add:Line_on_rule)
then have P45 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹
Eq (Geos (Poi A) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li B A) E” by (simp add:Point_Eq)
from P44 P45 have P46 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹
¬ Eq (Geos (Poi A) add Emp) (Geos (Poi E) add Emp)” by blast
from P26 P40 P43 P46 have P47 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹
Plane_sameside (Li B C) A E” by (simp add:Plane_not_diffside_sameside)
from P1 P39 P47 have P48 : ”¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ Ang_inside (An A B C) E” by (simp add:Ang_inside_def)
from assms have P49 : ”Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp) ⟹ Ang_inside (An A B C) E” by (simp add:Point_Eq)
from P48 P49 show ”Ang_inside (An A B C) E” by blast
qed

lemma (in Congruence_Rule) Ang_outside_Planeside :

```
  assumes
    "Def (Ang (An A B C))"
    "¬ Ang_inside (An A B C) D"
  shows "¬ (Plane_sameside (Li B A) C D ∧ Plane_sameside (Li B C) A D)"
  and "¬ Plane_sameside (Li B A) C D ∧ Plane_sameside (Li B C) A D
       ∨ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D
       ∨ ¬ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D"
proof -
  from assms have P1 : "Plane_sameside (Li B A) C D ∧ Plane_sameside (Li B C) A D ⟹ Ang_inside
(An A B C) D" by (simp add:Ang_inside_def)
  from assms P1 show "¬ (Plane_sameside (Li B A) C D ∧ Plane_sameside (Li B C) A D)" by blast
  thus "¬ Plane_sameside (Li B A) C D ∧ Plane_sameside (Li B C) A D
       ∨ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D
       ∨ ¬ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D" by blast
qed

lemma (in Congruence_Rule) Ang_outside_exclusive :
  assumes
    "Plane_sameside (Li B C) A D"
    "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)"
    "¬ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B D)) add Emp)"
  shows
    "¬ (¬ Ang_inside (An A B C) D ∧ ¬ Ang_inside (An D B C) A)"
proof -
  from assms have P1 : "¬ Line_on (Li B C) A" by (simp add:Plane_sameside_def)
  from assms P1 have "Def (Ang (An B C A))" by (simp add:Ang_sinple_def)
  then have P2 : "Def (Ang (An A B C))" by (blast intro:Ang_def_rev Ang_def_inv)
  then have P3 : "¬ Ang_inside (An A B C) D ⟹ ¬ Plane_sameside (Li B A) C D ∧ Plane_sameside
(Li B C) A D
       ∨ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D
       ∨ ¬ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D"
    by (simp add:Ang_outside_Planeside)
  from assms have P4 : "¬ Line_on (Li B C) D" by (simp add:Plane_sameside_def)
  from assms P4 have "Def (Ang (An B C D))" by (simp add:Ang_sinple_def)
  then have P5 : "Def (Ang (An D B C))" by (blast intro:Ang_def_rev Ang_def_inv)
  then have P6 : "¬ Ang_inside (An D B C) A ⟹ ¬ Plane_sameside (Li B D) C A ∧ Plane_sameside
(Li B C) D A
       ∨ Plane_sameside (Li B D) C A ∧ ¬ Plane_sameside (Li B C) D A
       ∨ ¬ Plane_sameside (Li B D) C A ∧ ¬ Plane_sameside (Li B C) D A"
    by (simp add:Ang_outside_Planeside)
  from P3 P6 have P7 : "¬ Ang_inside (An A B C) D ∧ ¬ Ang_inside (An D B C) A ⟹
    ¬ Plane_sameside (Li B A) C D ∧ Plane_sameside (Li B C) A D ∧ ¬ Plane_sameside (Li B D) C A
∧ Plane_sameside (Li B C) D A
    ∨ ¬ Plane_sameside (Li B A) C D ∧ Plane_sameside (Li B C) A D ∧ Plane_sameside (Li B D) C A
∧ ¬ Plane_sameside (Li B C) D A
    ∨ ¬ Plane_sameside (Li B A) C D ∧ Plane_sameside (Li B C) A D ∧ ¬ Plane_sameside (Li B D) C
A ∧ ¬ Plane_sameside (Li B C) D A
    ∨ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D ∧ ¬ Plane_sameside (Li B D) C
A ∧ Plane_sameside (Li B C) D A
    ∨ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D ∧ Plane_sameside (Li B D) C A
∧ ¬ Plane_sameside (Li B C) D A
    ∨ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D ∧ ¬ Plane_sameside (Li B D) C
A ∧ ¬ Plane_sameside (Li B C) D A
    ∨ ¬ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D ∧ ¬ Plane_sameside (Li B D)
C A ∧ Plane_sameside (Li B C) D A
    ∨ ¬ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D ∧ Plane_sameside (Li B D) C
A ∧ ¬ Plane_sameside (Li B C) D A
    ∨ ¬ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D ∧ ¬ Plane_sameside (Li B D)
C A ∧ ¬ Plane_sameside (Li B C) D A" by blast
  from assms have P8 : "Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D ∧ ¬
Plane_sameside (Li B D) C A ∧ Plane_sameside (Li B C) D A
    ∨ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D ∧ Plane_sameside (Li B D) C A
∧ ¬ Plane_sameside (Li B C) D A
    ∨ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D ∧ ¬ Plane_sameside (Li B D) C
A ∧ ¬ Plane_sameside (Li B C) D A
    ∨ ¬ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D ∧ ¬ Plane_sameside (Li B D)
```

C A ∧ Plane_sameside (Li B C) D A
∨ ¬ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D ∧ Plane_sameside (Li B D) C A ∧ ¬ Plane_sameside (Li B C) D A
∨ ¬ Plane_sameside (Li B A) C D ∧ ¬ Plane_sameside (Li B C) A D ∧ ¬ Plane_sameside (Li B D) C A ∧ ¬ Plane_sameside (Li B C) D A ⟹ False" by blast
from assms have "Plane_sameside (Li B C) D A" by (simp add:Plane_sameside_rev)
then have P9 : "¬ Plane_sameside (Li B A) C D ∧ Plane_sameside (Li B C) A D ∧ Plane_sameside (Li B D) C A ∧ ¬ Plane_sameside (Li B C) D A
∨ ¬ Plane_sameside (Li B A) C D ∧ Plane_sameside (Li B C) A D ∧ ¬ Plane_sameside (Li B D) C A ∧ ¬ Plane_sameside (Li B C) D A ⟹ False" by blast
have P10 : "Line_on (Li C B) C" by (simp add:Line_on_rule)
have P11 : "Line_on (Li C B) B" by (simp add:Line_on_rule)
from assms P10 P11 have "∃p. Bet_Point (Se C p) B ∧ Line_on (Li C B) p" by (blast intro:Bet_extension Eq_rev)
then obtain E :: Point where P12 : "Bet_Point (Se C E) B ∧ Line_on (Li C B) E" by blast
then have P13 : "Line_on (Li E C) B" by (simp add:Line_Bet_on)
have P14 : "Line_on (Li B A) B" by (simp add:Line_on_rule)
have P15 : "Line_on (Li B A) A" by (simp add:Line_on_rule)
have P16 : "Line_on (Li B C) B" by (simp add:Line_on_rule)
have P17 : "Line_on (Li B C) C" by (simp add:Line_on_rule)
from P16 have P18 : "Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp) ⟹ Line_on (Li B C) A" by (simp add:Point_Eq)
from P1 P18 have P19 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp)" by blast
from P13 P14 P15 P19 have P20 : "Line_on (Li E C) A ⟹ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li E C)) add Emp)" by (simp add:Line_unique)
have P21 : "Line_on (Li E C) C" by (simp add:Line_on_rule)
from assms P13 P16 P17 P21 have P22 : "Eq (Geos (Lin (Li E C)) add Emp) (Geos (Lin (Li B C)) add Emp)" by (simp add:Line_unique)
from P20 P22 have P23 : "Line_on (Li E C) A ⟹ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp)" by (blast intro:Eq_trans)
from P15 P23 have P24 : "Line_on (Li E C) A ⟹ Line_on (Li B C) A" by (simp add:Line_on_trans)
from P1 P24 have P25 : "¬ Line_on (Li E C) A" by blast
have P26 : "Line_on (Li E C) E" by (simp add:Line_on_rule)
from P21 have P27 : "Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp) ⟹ Line_on (Li E C) A" by (simp add:Point_Eq)
from P25 P27 have P28 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)" by blast
from P12 have P29 : "Bet_Point (Se C E) B" by simp
then have P30 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi E) add Emp)" by (simp add:Bet_Point_def Eq_rev)
have "Plane_diffside (Li B D) C A ⟹ ∃p. Bet_Point (Se C A) p ∧ Line_on (Li B D) p ∧ ¬ Line_on (Li B D) C ∧ ¬ Line_on (Li B D) A" by (simp add:Plane_diffside_def)
then have P31 : "Plane_diffside (Li B D) C A ⟹ ¬ Line_on (Li B D) C ∧ ¬ Line_on (Li B D) A" by blast
have P32 : "Line_on (Li B D) B" by (simp add:Line_on_rule)
from P29 have P33 : "¬ Eq (Geos (Poi E) add Emp) (Geos (Poi B) add Emp)"
by (simp add:Bet_Point_def)
from P13 P26 P32 P33 have P34 : "Line_on (Li B D) E ⟹ Eq (Geos (Lin (Li E C)) add Emp) (Geos (Lin (Li B D)) add Emp)" by (simp add:Line_unique)
from P21 P34 have P35 : "Line_on (Li B D) E ⟹ Line_on (Li B D) C" by (simp add:Line_on_trans)
from P31 P35 have P36 : "Plane_diffside (Li B D) C A ⟹ ¬ Line_on (Li B D) E" by blast
from P29 have P37 : "Bet_Point (Se E C) B" by (simp add:Bet_rev)
from P25 P31 P32 P36 P37 have P38 : "Plane_diffside (Li B D) C A ⟹
Line_on_Seg (Li B D) (Se E A) ∧ ¬ Line_on_Seg (Li B D) (Se C A)
∨ Line_on_Seg (Li B D) (Se C A) ∧ ¬ Line_on_Seg (Li B D) (Se E A)" by (simp add:Pachets_axiom)
have P39 : "Line_on_Seg (Li B D) (Se E A) ⟹ ∃p. Line_on (Li B D) p ∧ Bet_Point (Se E A) p" by (simp add:Line_on_Seg_rule)
from P31 P36 P39 have "Plane_diffside (Li B D) C A ∧ Line_on_Seg (Li B D) (Se E A) ⟹
∃p. Bet_Point (Se E A) p ∧ Line_on (Li B D) p ∧ ¬ Line_on (Li B D) E ∧ ¬ Line_on (Li B D) A" by blast
then have P40 : "Plane_diffside (Li B D) C A ∧ Line_on_Seg (Li B D) (Se E A) ⟹ Plane_diffside (Li B D) E A" by (simp add:Plane_diffside_def)
from P26 have P41 : "Eq (Geos (Lin (Li E C)) add Emp) (Geos (Lin (Li B D)) add Emp) ⟹ Line_on (Li B D) E" by (simp add:Line_on_trans)
from P36 P41 have P42 : "Plane_diffside (Li B D) C A ⟹ ¬ Eq (Geos (Lin (Li E C)) add Emp) (Geos (Lin (Li B D)) add Emp)" by blast
from P32 P37 P42 have P43 : "Plane_diffside (Li B D) C A ⟹ Plane_diffside (Li B D) E C" by (simp

add:Plane_Bet_diffside)
from P28 have P44 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)" by (blast intro:Eq_rev)
from P30 P40 P43 P44 have "Plane_diffside (Li B D) C A ∧ Line_on_Seg (Li B D) (Se E A) ⟹ Plane_sameside (Li B D) A C" by (blast intro:Plane_trans_inv)
then have P45 : "Plane_diffside (Li B D) C A ∧ Line_on_Seg (Li B D) (Se E A) ⟹ Plane_sameside (Li B D) C A" by (blast intro:Plane_sameside_rev)
have P46 : "Plane_diffside (Li B D) C A ⟹ ¬ Plane_sameside (Li B D) C A"
by (simp add:Plane_diffside_not_sameside)
from P45 P46 have P47 : "Plane_diffside (Li B D) C A ⟹ ¬ Line_on_Seg (Li B D) (Se E A)" by blast
from P38 P47 have "Plane_diffside (Li B D) C A ⟹ Line_on_Seg (Li B D) (Se C A)" by blast
then have "Plane_diffside (Li B D) C A ⟹ ∃p. Line_on (Li B D) p ∧ Bet_Point (Se C A) p" by (simp add:Line_on_Seg_rule)
then obtain F :: Point where P48 : "Plane_diffside (Li B D) C A ⟹ Line_on (Li B D) F ∧ Bet_Point (Se C A) F" by blast
from P15 have P49 : "Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp) ⟹ Line_on (Li B C) A" by (simp add:Line_on_trans)
from P1 P49 have P50 : "¬ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp)" by blast
from P48 have P51 : "Plane_diffside (Li B D) C A ⟹ Bet_Point (Se A C) F" by (simp add:Bet_rev)
from assms P19 P50 P51 have P52 : "Plane_diffside (Li B D) C A ⟹ Ang_inside (An A B C) F" by (simp add:Ang_inside_Bet_Point)
then have "Plane_diffside (Li B D) C A ⟹ Eq (Geos (Poi F) add Emp) (Geos (Poi D) add Emp) ⟹ Ang_inside (An A B C) D" by (simp add:Point_Eq)
then have P53 : "¬ Ang_inside (An A B C) D ⟹ Plane_diffside (Li B D) C A ⟹ ¬ Eq (Geos (Poi D) add Emp) (Geos (Poi F) add Emp)" by (blast intro:Eq_rev)
from P48 have "Plane_diffside (Li B D) C A ⟹ Bet_Point (Se C A) F" by simp
then have "Plane_diffside (Li B D) C A ⟹ Eq (Geos (Poi F) add Emp) (Geos (Poi B) add Emp) ⟹ Bet_Point (Se C A) B" by (simp add:Point_Eq)
then have P54 : "Plane_diffside (Li B D) C A ⟹ Eq (Geos (Poi F) add Emp) (Geos (Poi B) add Emp) ⟹ Line_on (Li B C) A" by (simp add:Line_Bet_on)
from P1 P54 have P55 : "Plane_diffside (Li B D) C A ⟹ ¬ Eq (Geos (Poi F) add Emp) (Geos (Poi B) add Emp)" by blast
have P56 : "Line_on (Li B D) D" by (simp add:Line_on_rule)
from P16 have P57 : "Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp) ⟹ Line_on (Li B C) D" by (simp add:Point_Eq)
from P4 P57 have P58 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)" by blast
from P32 P48 P53 P55 P56 P58 have "¬ Ang_inside (An A B C) D ⟹ Plane_diffside (Li B D) C A ⟹
Bet_Point (Se B F) D ∨ Bet_Point (Se F D) B ∨ Bet_Point (Se D B) F" by (simp add:Bet_case)
then have P59 : "¬ Ang_inside (An A B C) D ⟹ Plane_diffside (Li B D) C A ⟹
Bet_Point (Se B F) D ∧ ¬ Bet_Point (Se F D) B ∧ ¬ Bet_Point (Se D B) F
∨ ¬ Bet_Point (Se B F) D ∧ Bet_Point (Se F D) B ∧ ¬ Bet_Point (Se D B) F
∨ ¬ Bet_Point (Se B F) D ∧ ¬ Bet_Point (Se F D) B ∧ Bet_Point (Se D B) F"
by (simp add:Bet_case_fact)
from P26 have P60 : "Eq (Geos (Lin (Li E C)) add Emp) (Geos (Lin (Li B D)) add Emp) ⟹ Line_on (Li B D) E" by (simp add:Line_on_trans)
from P36 P60 have P61 : "Plane_diffside (Li B D) C A ⟹ ¬ Eq (Geos (Lin (Li B D)) add Emp) (Geos (Lin (Li E C)) add Emp)" by (blast intro:Eq_rev)
have P62 : "Bet_Point (Se F D) B ⟹ Line_on (Li F D) B" by (simp add:Line_Bet_on)
have P63 : "Line_on (Li F D) D" by (simp add:Line_on_rule)
from P32 P56 P58 P62 P63 have P64 : "Bet_Point (Se F D) B ⟹ Eq (Geos (Lin (Li F D)) add Emp) (Geos (Lin (Li B D)) add Emp)" by (simp add:Line_unique)
from P61 P64 have P65 : "Plane_diffside (Li B D) C A ⟹ Bet_Point (Se F D) B ⟹
¬ Eq (Geos (Lin (Li F D)) add Emp) (Geos (Lin (Li E C)) add Emp)" by (blast intro:Eq_trans)
from P13 P65 have P66 : "Plane_diffside (Li B D) C A ⟹ Bet_Point (Se F D) B ⟹ Plane_diffside (Li E C) F D" by (simp add:Plane_Bet_diffside)
have "Line_on (Li C A) A" by (simp add:Line_on_rule)
then have P67 : "Eq (Geos (Lin (Li C A)) add Emp) (Geos (Lin (Li E C)) add Emp) ⟹ Line_on (Li E C) A" by (simp add:Line_on_trans)
from P25 P67 have P68 : "¬ Eq (Geos (Lin (Li C A)) add Emp) (Geos (Lin (Li E C)) add Emp)" by blast
from P48 have P69 : "Plane_diffside (Li B D) C A ⟹ Bet_Point (Se C A) F" by simp
from P21 P68 P69 have P71 : "Plane_diffside (Li B D) C A ⟹ Plane_sameside (Li E C) F A" by (simp add:Plane_Bet_sameside)
from P66 P71 have P72 : "Plane_diffside (Li B D) C A ⟹ Bet_Point (Se F D) B ⟹

```
    Plane_diffside (Li E C) A D" by (simp add:Plane_trans)
  from P22 P72 have "Plane_diffside (Li B D) C A ⟹ Bet_Point (Se F D) B ⟹
    Plane_diffside (Li B C) A D" by (simp add:Plane_Line_diff_trans)
  then have "Plane_diffside (Li B D) C A ⟹ Bet_Point (Se F D) B ⟹
    ¬ Plane_sameside (Li B C) A D" by (simp add:Plane_diffside_not_sameside)
  then have P73 : "Plane_sameside (Li B C) A D ∧ Plane_diffside (Li B D) C A ⟹ ¬ Bet_Point (Se F
D) B" by blast
  have P74 : "Bet_Point (Se B F) D ⟹ Line_on (Li B F) D" by (simp add:Line_Bet_on)
  from P59 have P75 : "¬ Ang_inside (An A B C) D ⟹ Plane_diffside (Li B D) C A ⟹
    Bet_Point (Se B F) D ⟹ ¬ Bet_Point (Se D F) B" by (blast intro:Bet_rev)
  from P52 P58 P74 P75 have "¬ Ang_inside (An A B C) D ⟹ Plane_diffside (Li B D) C A ⟹
    Bet_Point (Se B F) D ⟹ Ang_inside (An A B C) D" by (simp add:Ang_inside_HalfLine)
  then have P76 : "¬ Ang_inside (An A B C) D ⟹ Plane_diffside (Li B D) C A ⟹ ¬ Bet_Point (Se
B F) D" by blast
  have P77 : "Bet_Point (Se D B) F ⟹ Line_on (Li B F) D" by (simp add:Line_Bet_on)
  from P59 have P78 : "¬ Ang_inside (An A B C) D ⟹ Plane_diffside (Li B D) C A ⟹
    Bet_Point (Se D B) F ⟹ ¬ Bet_Point (Se D F) B" by (blast intro:Bet_rev)
  from P52 P58 P77 P78 have "¬ Ang_inside (An A B C) D ⟹ Plane_diffside (Li B D) C A ⟹
    Bet_Point (Se D B) F ⟹ Ang_inside (An A B C) D" by (simp add:Ang_inside_HalfLine)
  then have P79 : "¬ Ang_inside (An A B C) D ⟹ Plane_diffside (Li B D) C A ⟹ ¬ Bet_Point (Se
D B) F" by blast
  from P59 P73 P76 P79 have "¬ Ang_inside (An A B C) D ∧ Plane_sameside (Li B C) A D ∧
Plane_diffside (Li B D) C A ⟹ False" by blast
  then have P80 : "¬ Ang_inside (An A B C) D ∧ ¬ Plane_sameside (Li B A) C D
    ∧ Plane_sameside (Li B C) A D ∧ Plane_diffside (Li B D) C A ∧ Plane_sameside (Li B C) D A ⟹
False" by simp
  from P5 have "Def (Tri (Tr B D C))" by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
  then have P81 : "¬ Line_on (Li B D) C" by (simp add:Tri_def_Line)
  from P14 P15 P19 P32 have P82 : "Line_on (Li B D) A ⟹
    Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B D)) add Emp)" by (simp add:Line_unique)
  from assms P82 have P83 : "¬ Line_on (Li B D) A" by blast
  from P28 P81 P83 have P84 : "¬ Plane_sameside (Li B D) C A ⟹ Plane_diffside (Li B D) C A" by
(simp add:Plane_not_sameside_diffside)
  from P80 P84 have P85 : "¬ Ang_inside (An A B C) D ∧ ¬ Plane_sameside (Li B A) C D
    ∧ Plane_sameside (Li B C) A D ∧ ¬ Plane_sameside (Li B D) C A ∧ Plane_sameside (Li B C) D A
⟹ False" by simp
  from P7 P8 P9 P85 show "¬ (¬ Ang_inside (An A B C) D ∧ ¬ Ang_inside (An D B C) A)" by blast
qed

lemma (in Congruence_Rule) Ang_inside_case :
  assumes
    "Def (Ang (An A B C))"
    "Def (Ang (An D B C))"
    "Plane_sameside (Li B C) A D"
    "¬ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B D)) add Emp)"
  shows
    "Ang_inside (An A B C) D ∧ ¬ Ang_inside (An D B C) A
    ∨ ¬ Ang_inside (An A B C) D ∧ Ang_inside (An D B C) A"
proof -
  have P1 : "Ang_inside (An A B C) D ⟹ Plane_sameside (Li B A) C D ∧ Plane_sameside (Li B C)
A D" by (simp add:Ang_inside_def)
  have P2 : "Ang_inside (An D B C) A ⟹ Plane_sameside (Li B D) C A ∧ Plane_sameside (Li B C)
D A" by (simp add:Ang_inside_def)
  have P3 : "Ang_inside (An A B C) D ⟹ Plane_diffside (Li B D) A C"
    by (simp add:Ang_inside_Planeside)
  have "Plane_diffside (Li B D) A C ⟹ Plane_diffside (Li B D) C A" by (simp add:Plane_diffside_rev)
  then have P4 : "Plane_diffside (Li B D) A C ⟹ ¬ Plane_sameside (Li B D) C A" by (simp
add:Plane_diffside_not_sameside)
  from P3 P4 have P5 : "Ang_inside (An A B C) D ⟹ ¬ Plane_sameside (Li B D) C A" by (simp
add:Plane_diffside_rev)
  from P2 P5 have P6 : "Ang_inside (An A B C) D ∧ Ang_inside (An D B C) A ⟹ False" by simp
  from assms have P7 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)" by (simp add:Ang_def)
  from assms P7 have P8 : "¬ Ang_inside (An A B C) D ∧ ¬ Ang_inside (An D B C) A ⟹ False" by
(simp add:Ang_outside_exclusive)
  from P6 P8 show "Ang_inside (An A B C) D ∧ ¬ Ang_inside (An D B C) A
    ∨ ¬ Ang_inside (An A B C) D ∧ Ang_inside (An D B C) A" by blast
```

```
qed

lemma (in Congruence_Rule) Plane_sameside_HalfLine :
  assumes
    "Plane_sameside l1 p1 p2"
    "Line_on l1 p3"
    "Line_on (Li p3 p1) p4"
    "¬ Bet_Point (Se p4 p1) p3"
    "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p4) add Emp)"
    "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p4) add Emp)"
  shows "Plane_sameside l1 p1 p4"
proof -
  from assms have P1 : "¬ Line_on l1 p1 ∧ ¬ Line_on l1 p2
    ∧ ¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)" by (simp add:Plane_sameside_def)
  have "Plane_diffside l1 p1 p4 ⟹
    ∃p. Bet_Point (Se p1 p4) p ∧ Line_on l1 p ∧ ¬ Line_on l1 p1 ∧ ¬ Line_on l1 p4" by (simp
add:Plane_diffside_def)
  then obtain p5 :: Point where P2 : "Plane_diffside l1 p1 p4 ⟹
    Bet_Point (Se p1 p4) p5 ∧ Line_on l1 p5 ∧ ¬ Line_on l1 p4" by blast
  from assms have P3 : "Eq (Geos (Poi p3) add Emp) (Geos (Poi p1) add Emp) ⟹ Line_on l1 p1" by
(simp add:Point_Eq)
  from P1 P3 have P4 : "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p1) add Emp)" by blast
  then have P5 : "Eq (Geos (Lin (Li p3 p1)) add Emp) (Geos (Lin (Li p1 p3)) add Emp)" by (simp
add:Line_rev)
  from assms P5 have P6 : "Line_on (Li p1 p3) p4" by (simp add:Line_rev Line_on_trans)
  from P4 have P7 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)" by (blast intro:Eq_rev)
  from assms P6 P7 have P8 : "Line_on (Li p1 p4) p3" by (simp add:Line_on_rev)
  from P2 have P9 : "Plane_diffside l1 p1 p4 ⟹ Line_on (Li p1 p4) p5" by (simp add:Line_Bet_on)
  from assms P2 P8 P9 have P10 : "Plane_diffside l1 p1 p4 ⟹ ¬ Eq (Geos (Poi p3) add Emp) (Geos
(Poi p5) add Emp) ⟹
    Eq (Geos (Lin (Li p1 p4)) add Emp) (Geos (Lin l1) add Emp)" by (simp add:Line_unique)
  have P11 : "Line_on (Li p1 p4) p1" by (simp add:Line_on_rule)
  from P10 P11 have P12 : "Plane_diffside l1 p1 p4 ⟹ ¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p5)
add Emp) ⟹
    Line_on l1 p1" by (simp add:Line_on_trans)
  from P1 P12 have P13 : "Plane_diffside l1 p1 p4 ⟹ Eq (Geos (Poi p5) add Emp) (Geos (Poi p3) add
Emp)" by (blast intro:Eq_rev)
  from P2 have P14 : "Plane_diffside l1 p1 p4 ⟹ Bet_Point (Se p1 p4) p5" by simp
  from P13 P14 have "Plane_diffside l1 p1 p4 ⟹ Bet_Point (Se p1 p4) p3" by (simp add:Point_Eq)
  then have P15 : "Plane_diffside l1 p1 p4 ⟹ Bet_Point (Se p4 p1) p3" by (simp add:Bet_rev)
  from assms P15 have P16 : "¬ Plane_diffside l1 p1 p4" by blast
  have P17 : "Line_on (Li p3 p1) p3" by (simp add:Line_on_rule)
  from assms P17 have P18 : "Line_on l1 p4 ⟹
    Eq (Geos (Lin (Li p3 p1)) add Emp) (Geos (Lin l1) add Emp)" by (simp add:Line_unique)
  have P19 : "Line_on (Li p3 p1) p1" by (simp add:Line_on_rule)
  from P18 P19 have P20 : "Line_on l1 p4 ⟹ Line_on l1 p1" by (simp add:Line_on_trans)
  from P1 P20 have P21 : "¬ Line_on l1 p4" by blast
  from assms P1 P16 P21 show "Plane_sameside l1 p1 p4" by (simp add:Plane_not_diffside_sameside)
qed

lemma (in Congruence_Rule) Plane_Bet_sameside_rev :
  assumes
    "Plane_sameside l1 p1 p3"
    "Line_on l1 p2"
    "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)"
    "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp)"
    "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p1) add Emp)"
    "Line_on l2 p1" "Line_on l2 p2" "Line_on l2 p3"
    "¬ Eq (Geos (Lin l1) add Emp) (Geos (Lin l2) add Emp)"
  shows "Bet_Point (Se p3 p2) p1 ∨ Bet_Point (Se p2 p1) p3"
proof -
  from assms have P1 : "Bet_Point (Se p1 p3) p2 ∨ Bet_Point (Se p3 p2) p1 ∨ Bet_Point (Se p2 p1) p3"
by (simp add:Bet_case)
  have P2 : "Line_on (Li p1 p3) p1" by (simp add:Line_on_rule)
  have P3 : "Line_on (Li p1 p3) p3" by (simp add:Line_on_rule)
  from assms P2 P3 have P4 : " Eq (Geos (Lin l2) add Emp) (Geos (Lin (Li p1 p3)) add Emp)" by (simp
```

```
add:Line_unique)
  from assms P4 have "¬ Eq (Geos (Lin l1) add Emp) (Geos (Lin (Li p1 p3)) add Emp)" by (blast
intro:Eq_trans)
  then have P6 : "¬ Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin l1) add Emp)" by (blast in-
tro:Eq_rev)
  from assms P6 have "Bet_Point (Se p1 p3) p2 ⟹ Plane_diffside l1 p1 p3"
    by (simp add:Plane_Bet_diffside)
  then have P7 : "Bet_Point (Se p1 p3) p2 ⟹ ¬ Plane_sameside l1 p1 p3"
    by (simp add:Plane_diffside_not_sameside)
  from assms P7 have P8 : "¬ Bet_Point (Se p1 p3) p2" by blast
  from P1 P8 show "Bet_Point (Se p3 p2) p1 ∨ Bet_Point (Se p2 p1) p3" by blast
qed

lemma (in Congruence_Rule) Seg_Bet_relation :
  assumes N :
    "Bet_Point (Se p1 p2) p3"
  shows "¬ Eq (Geos (Seg (Se p1 p2)) add Emp) (Geos (Seg (Se p1 p3)) add Emp)"
proof
  assume W : "Eq (Geos (Seg (Se p1 p2)) add Emp) (Geos (Seg (Se p1 p3)) add Emp)"
  from N have "Inv (Bet_Point (Se p2 p3) p1) ∧ Inv (Bet_Point (Se p3 p1) p2)" by (simp add:Bet_iff)
  then have P1 : "¬ Bet_Point (Se p2 p3) p1" by (simp add:Inv_def)
  have P2 : "Line_on (Li p1 p2) p1" by (simp add:Line_on_rule)
  have P3 : "Line_on (Li p1 p2) p2" by (simp add:Line_on_rule)
  from N have P4 : "Line_on (Li p1 p2) p3" by (simp add:Line_Bet_on)
  from N have P5 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)" by (simp add:Bet_Point_def)
  from N have "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p1) add Emp)" by (simp add:Bet_Point_def)
  then have P6 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p3) add Emp)" by (blast intro:Eq_rev)
  from W P1 P2 P3 P4 P5 P6 have P7 : "Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp)" by
(blast intro:Seg_move_unique)
  from N have P8 : "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp)" by (simp add:Bet_Point_def)
  from P7 P8 show False by blast
qed

lemma (in Congruence_Rule) Seg_Bet_move_lemma1 :
  assumes
    "Bet_Point (Se p11 p13) p12"
    "Line_on l1 p21" "Line_on l1 p22" "Line_on l1 p23"
    "¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi p22) add Emp)"
    "¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi p23) add Emp)"
    "Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p22)) add Emp)"
    "Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 p23)) add Emp)"
    "¬ Bet_Point (Se p22 p23) p21"
  shows "Bet_Point (Se p21 p23) p22"
proof -
  from assms have P1 : "¬ Eq (Geos (Poi p22) add Emp) (Geos (Poi p21) add Emp)" by (blast in-
tro:Eq_rev)
  from assms have "¬ Eq (Geos (Poi p12) add Emp) (Geos (Poi p11) add Emp)"
    by (simp add:Bet_Point_def)
  then have P2 : "¬ Eq (Geos (Poi p11) add Emp) (Geos (Poi p12) add Emp)" by (blast intro:Eq_rev)
  from assms P1 P2 have "∃p. Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p22 p)) add Emp)
    ∧ ¬ Bet_Point (Se p p21) p22 ∧ Line_on l1 p ∧ ¬ Eq (Geos (Poi p22) add Emp) (Geos (Poi p) add
Emp)" by (simp add:Seg_move_sameside)
  then obtain p211 :: Point where P3 : "Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p22
p211)) add Emp)
    ∧ ¬ Bet_Point (Se p211 p21) p22 ∧ Line_on l1 p211 ∧ ¬ Eq (Geos (Poi p22) add Emp) (Geos (Poi
p211) add Emp)" by blast
  from assms have "¬ Eq (Geos (Poi p13) add Emp) (Geos (Poi p12) add Emp)"
    by (simp add:Bet_Point_def)
  then have P4 : "¬ Eq (Geos (Poi p12) add Emp) (Geos (Poi p13) add Emp)" by (blast intro:Eq_rev)
  from assms P3 P4 have "∃p. Eq (Geos (Seg (Se p12 p13)) add Emp) (Geos (Seg (Se p22 p)) add Emp)
    ∧ Bet_Point (Se p p211) p22 ∧ Line_on l1 p ∧ ¬ Eq (Geos (Poi p22) add Emp) (Geos (Poi p) add
Emp)" by (simp add:Seg_move_diffside)
  then obtain p231 :: Point where P5 : "Eq (Geos (Seg (Se p12 p13)) add Emp) (Geos (Seg (Se p22
p231)) add Emp)
    ∧ Bet_Point (Se p231 p211) p22 ∧ Line_on l1 p231 ∧ ¬ Eq (Geos (Poi p22) add Emp) (Geos (Poi
p231) add Emp)" by blast
```

```
  have P6 : "Eq (Geos (Seg (Se p21 p22)) add Emp) (Geos (Seg (Se p22 p21)) add Emp)" by (simp
add:Seg_rev)
  from assms have P7 : "Eq (Geos (Seg (Se p21 p22)) add Emp) (Geos (Seg (Se p11 p12)) add Emp)"
by (simp add:Eq_rev)
  from P3 P7 have P8 : "Eq (Geos (Seg (Se p22 p211)) add Emp) (Geos (Seg (Se p21 p22)) add Emp)"
by (blast intro:Eq_trans Eq_rev)
  from P6 P8 have P9 : "Eq (Geos (Seg (Se p22 p211)) add Emp) (Geos (Seg (Se p22 p21)) add Emp)"
by (blast intro:Eq_trans)
  from assms P1 P3 P9 have P10 : "Eq (Geos (Poi p211) add Emp) (Geos (Poi p21) add Emp)" by (blast
intro:Seg_move_unique)
  from P5 have P11 : "Bet_Point (Se p211 p231) p22" by (simp add:Bet_rev)
  from P10 P11 have P12 : "Bet_Point (Se p21 p231) p22" by (simp add:Bet_Point_Eq)
  have P13 : "Line_on (Li p11 p12) p11" by (simp add:Line_on_rule)
  have P14 : "Line_on (Li p11 p12) p12" by (simp add:Line_on_rule)
  from assms have P15 : "Line_on (Li p11 p12) p13" by (simp add:Line_Bet_on)
  from assms have P16 : "¬ Seg_on_Seg (Se p11 p12) (Se p12 p13)" by (simp add:Seg_Bet_not_on)
  from P12 have P17 : "¬ Seg_on_Seg (Se p21 p22) (Se p22 p231)" by (simp add:Seg_Bet_not_on)
  from assms P5 P13 P14 P15 P16 P17 have P18 : "Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg
(Se p21 p231)) add Emp)" by (simp add:Seg_add)
  from assms P18 have P19 : "Eq (Geos (Seg (Se p21 p231)) add Emp) (Geos (Seg (Se p21 p23)) add
Emp)" by (blast intro:Eq_trans Eq_rev)
  from P12 have P20 : "¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi p231) add Emp)" by (simp
add:Bet_Point_def)
  from P12 have P21 : "Bet_Point (Se p231 p21) p22" by (simp add:Bet_rev)
  from P21 have P22 : "Bet_Point (Se p231 p23) p21 ⟹ Bet_Point (Se p22 p23) p21" by (blast in-
tro:Bet_swap_134_234)
  from assms P22 have P23 : "¬ Bet_Point (Se p231 p23) p21" by blast
  from assms P5 P19 P20 P23 have P24 : "Eq (Geos (Poi p231) add Emp) (Geos (Poi p23) add Emp)"
by (blast intro:Seg_move_unique)
  from P21 P24 have "Bet_Point (Se p23 p21) p22" by (simp add:Bet_Point_Eq)
  thus "Bet_Point (Se p21 p23) p22" by (simp add:Bet_rev)
qed

lemma (in Congruence_Rule) Seg_Bet_move_sameside :
  assumes
    "Bet_Point (Se p11 p13) p12"
    "Line_on l1 p21" "Line_on l1 p4"
    "¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi p4) add Emp)"
  shows "∃p q. Bet_Point (Se p21 q) p ∧ Line_on l1 p ∧ Line_on l1 q
    ∧ Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p)) add Emp)
    ∧ Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 q)) add Emp)
    ∧ ¬ Bet_Point (Se p p4) p21 ∧ ¬ Bet_Point (Se q p4) p21"
proof -
  from assms have "¬ Eq (Geos (Poi p12) add Emp) (Geos (Poi p11) add Emp)"
    by (simp add:Bet_Point_def)
  then have P1 : "¬ Eq (Geos (Poi p11) add Emp) (Geos (Poi p12) add Emp)" by (blast intro:Eq_rev)
  from assms P1 have "∃p. Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p)) add Emp)
    ∧ ¬ Bet_Point (Se p p4) p21 ∧ Line_on l1 p ∧ ¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi p) add
Emp)" by (simp add:Seg_move_sameside)
  then obtain p22 :: Point where P2 : "Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p22))
add Emp)
    ∧ ¬ Bet_Point (Se p22 p4) p21 ∧ Line_on l1 p22 ∧ ¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi
p22) add Emp)" by blast
  from assms have P3 : "¬ Eq (Geos (Poi p11) add Emp) (Geos (Poi p13) add Emp)" by (simp
add:Bet_Point_def)
  from assms P2 P3 have "∃p. Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 p)) add Emp)
    ∧ ¬ Bet_Point (Se p p22) p21 ∧ Line_on l1 p ∧ ¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi p) add
Emp)" by (simp add:Seg_move_sameside)
  then obtain p23 :: Point where P4 : "Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 p23))
add Emp)
    ∧ ¬ Bet_Point (Se p23 p22) p21 ∧ Line_on l1 p23 ∧ ¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi
p23) add Emp)" by blast
  from P4 have "¬ Bet_Point (Se p23 p22) p21" by simp
  then have P5 : "¬ Bet_Point (Se p22 p23) p21" by (blast intro:Bet_rev)
  from assms P2 P4 P5 have P6 : "Bet_Point (Se p21 p23) p22" by (blast intro:Seg_Bet_move_lemma1)
  then have P7 : "Bet_Point (Se p23 p21) p22" by (simp add:Bet_rev)
```

```
  from P7 have P8 : "Bet_Point (Se p23 p4) p21 ⟹ Bet_Point (Se p22 p4) p21"
    by (blast intro:Bet_swap_134_234)
  from P2 P8 have P9 : "¬ Bet_Point (Se p23 p4) p21" by blast
  from P2 P4 P6 P9 show "∃p q. Bet_Point (Se p21 q) p ∧ Line_on l1 p ∧ Line_on l1 q
    ∧ Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p)) add Emp)
    ∧ Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 q)) add Emp)
    ∧ ¬ Bet_Point (Se p p4) p21 ∧ ¬ Bet_Point (Se q p4) p21" by blast
qed

lemma (in Congruence_Rule) Seg_Bet_move_diffside :
  assumes
    "Bet_Point (Se p11 p13) p12"
    "Line_on l1 p21" "Line_on l1 p4"
    "¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi p4) add Emp)"
  shows "∃p q. Bet_Point (Se p21 q) p ∧ Line_on l1 p ∧ Line_on l1 q
    ∧ Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p)) add Emp)
    ∧ Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 q)) add Emp)
    ∧ Bet_Point (Se p p4) p21 ∧ Bet_Point (Se q p4) p21"
proof -
  from assms have "¬ Eq (Geos (Poi p12) add Emp) (Geos (Poi p11) add Emp)"
    by (simp add:Bet_Point_def)
  then have P1 : "¬ Eq (Geos (Poi p11) add Emp) (Geos (Poi p12) add Emp)" by (blast intro:Eq_rev)
  from assms P1 have "∃p. Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p)) add Emp)
    ∧ Bet_Point (Se p p4) p21 ∧ Line_on l1 p ∧ ¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi p) add
Emp)" by (simp add:Seg_move_diffside)
  then obtain p22 :: Point where P2 : "Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p22))
add Emp)
    ∧ Bet_Point (Se p22 p4) p21 ∧ Line_on l1 p22 ∧ ¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi p22)
add Emp)" by blast
  from assms have P3 : "¬ Eq (Geos (Poi p11) add Emp) (Geos (Poi p13) add Emp)" by (simp
add:Bet_Point_def)
  from assms P2 P3 have "∃p. Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 p)) add Emp)
    ∧ ¬ Bet_Point (Se p p22) p21 ∧ Line_on l1 p ∧ ¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi p) add
Emp)" by (simp add:Seg_move_sameside)
  then obtain p23 :: Point where P4 : "Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 p23))
add Emp)
    ∧ ¬ Bet_Point (Se p23 p22) p21 ∧ Line_on l1 p23 ∧ ¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi
p23) add Emp)" by blast
  from P4 have "¬ Bet_Point (Se p23 p22) p21" by simp
  then have P5 : "¬ Bet_Point (Se p22 p23) p21" by (blast intro:Bet_rev)
  from assms P2 P4 P5 have P6 : "Bet_Point (Se p21 p23) p22" by (blast intro:Seg_Bet_move_lemma1)
  then have P7 : "Bet_Point (Se p23 p21) p22" by (simp add:Bet_rev)
  from P2 P7 have P8 : "Bet_Point (Se p23 p4) p21" by (blast intro:Bet_swap_234_134)
  from P2 P4 P6 P8 show "∃p q. Bet_Point (Se p21 q) p ∧ Line_on l1 p ∧ Line_on l1 q
    ∧ Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p)) add Emp)
    ∧ Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 q)) add Emp)
    ∧ Bet_Point (Se p p4) p21 ∧ Bet_Point (Se q p4) p21" by blast
qed

lemma (in Congruence_Rule) Seg_Bet_wrong_relation :
  assumes
    "Bet_Point (Se p11 p13) p12"
    "Bet_Point (Se p21 p22) p23"
    "Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p22)) add Emp)"
    "Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 p23)) add Emp)"
  shows False
proof -
  have P1 : "Line_on (Li p21 p22) p21" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li p21 p22) p22" by (simp add:Line_on_rule)
  from assms have P3 : "¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi p22) add Emp)" by (simp
add:Bet_Point_def)
  from assms P1 P2 P3 have "∃p q. Bet_Point (Se p21 q) p ∧ Line_on (Li p21 p22) p ∧ Line_on (Li p21
p22) q
    ∧ Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 p)) add Emp)
    ∧ Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 q)) add Emp)
    ∧ ¬ Bet_Point (Se p p22) p21 ∧ ¬ Bet_Point (Se q p22) p21" by (simp add:Seg_Bet_move_sameside)
```

```
  then obtain pn2 pn3 :: Point where P4 : "Bet_Point (Se p21 pn3) pn2 ∧ Line_on (Li p21 p22) pn2 ∧
Line_on (Li p21 p22) pn3
    ∧ Eq (Geos (Seg (Se p11 p12)) add Emp) (Geos (Seg (Se p21 pn2)) add Emp)
    ∧ Eq (Geos (Seg (Se p11 p13)) add Emp) (Geos (Seg (Se p21 pn3)) add Emp)
    ∧ ¬ Bet_Point (Se pn2 p22) p21 ∧ ¬ Bet_Point (Se pn3 p22) p21" by blast
  then have P5 : "Bet_Point (Se p21 pn3) pn2" by simp
  then have "¬ Eq (Geos (Poi pn2) add Emp) (Geos (Poi p21) add Emp)" by (simp add:Bet_Point_def)
  then have P6 : "¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi pn2) add Emp)" by (blast intro:Eq_rev)
  from assms P4 have P7 : "Eq (Geos (Seg (Se p21 pn2)) add Emp) (Geos (Seg (Se p21 p22)) add
Emp)" by (blast intro:Eq_trans)
  from P1 P2 P3 P4 P6 P7 have P8 : "Eq (Geos (Poi pn2) add Emp) (Geos (Poi p22) add Emp)" by
(blast intro:Seg_move_unique)
  from P5 P8 have "Bet_Point (Se p21 pn3) p22" by (simp add:Point_Eq)
  then have P9 : "Bet_Point (Se pn3 p21) p22" by (simp add:Bet_rev)
  from assms have P10 : "Line_on (Li p21 p22) p23" by (simp add:Line_Bet_on)
  from assms have "¬ Eq (Geos (Poi p23) add Emp) (Geos (Poi p21) add Emp)"
    by (simp add:Bet_Point_def)
  then have P11 : "¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi p23) add Emp)" by (blast intro:Eq_rev)
  from P5 have P12 : "¬ Eq (Geos (Poi p21) add Emp) (Geos (Poi pn3) add Emp)"
    by (simp add:Bet_Point_def)
  from assms P4 have P13 : "Eq (Geos (Seg (Se p21 pn3)) add Emp) (Geos (Seg (Se p21 p23)) add
Emp)" by (blast intro:Eq_trans)
  from assms have P14 : "Bet_Point (Se p22 p21) p23" by (simp add:Bet_rev)
  have P15 : "Bet_Point (Se pn3 p23) p21 ⟹ Bet_Point (Se p23 pn3) p21" by (simp add:Bet_rev)
  from P14 P15 have "Bet_Point (Se pn3 p23) p21 ⟹ Bet_Point (Se p22 pn3) p21" by (blast in-
tro:Bet_swap_234_134)
  then have P16 : "Bet_Point (Se pn3 p23) p21 ⟹ Bet_Point (Se pn3 p22) p21" by (simp add:Bet_rev)
  from P4 P16 have P17 : "¬ Bet_Point (Se pn3 p23) p21" by blast
  from P1 P4 P10 P11 P12 P13 P17 have P18 : "Eq (Geos (Poi pn3) add Emp) (Geos (Poi p23) add
Emp)" by (blast intro:Seg_move_unique)
  from P9 P18 have P19 : "Bet_Point (Se p23 p21) p22" by (simp add:Bet_Point_Eq)
  from assms have "Inv (Bet_Point (Se p22 p23) p21) ∧ Inv (Bet_Point (Se p23 p21) p22)" by (simp
add:Bet_iff)
  then have P20 : "¬ Bet_Point (Se p23 p21) p22" by (simp add:Inv_def)
  from P19 P20 show False by blast
qed

lemma (in Congruence_Rule) Ang_inside_trans :
  assumes
    "Ang_inside (An A B C) D" "Def (Ang (An A B C))"
    "Line_on (Li B A1) A" "¬ Bet_Point (Se A A1) B"
    "Line_on (Li B C1) C" "¬ Bet_Point (Se C C1) B"
    "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi A1) add Emp)"
    "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C1) add Emp)"
  shows "Ang_inside (An A1 B C1) D"
proof -
  from assms have P1 : "Plane_sameside (Li B A) C D ∧ Plane_sameside (Li B C) A D" by (simp
add:Ang_inside_def)
  have P2 : "Line_on (Li B A) B" by (simp add:Line_on_rule)
  have P3 : "Line_on (Li B A1) B" by (simp add:Line_on_rule)
  have P4 : "Line_on (Li B A) A" by (simp add:Line_on_rule)
  from assms have P5 : "Def (Tri (Tr A B C))" by (simp add:Ang_to_Tri)
  from P5 have P6 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)" by (simp add:Tri_def)
  from assms P2 P3 P4 P6 have P7 : "Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B A1)) add
Emp)" by (simp add:Line_unique)
  have P8 : "Line_on (Li B C) B" by (simp add:Line_on_rule)
  have P9 : "Line_on (Li B C1) B" by (simp add:Line_on_rule)
  have P10 : "Line_on (Li B C) C" by (simp add:Line_on_rule)
  from P5 have P11 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)" by (simp add:Tri_def)
  from assms P8 P9 P10 P11 have P12 : "Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li B C1)) add
Emp)" by (simp add:Line_unique)
  have P13 : "Plane_diffside (Li B A) C1 D ⟹ Plane_diffside (Li B A) D C1"
    by (simp add:Plane_diffside_rev)
  from P1 have P14 : "Plane_sameside (Li B A) D C" by (simp add:Plane_sameside_rev)
  from P13 P14 have P15 : "Plane_diffside (Li B A) C1 D ⟹ Plane_diffside (Li B A) C C1" by (simp
add:Plane_trans)
```

then have "Plane_diffside (Li B A) C1 D ⟹ ∃p. Bet_Point (Se C C1) p
∧ Line_on (Li B A) p ∧ ¬ Line_on (Li B A) C ∧ ¬ Line_on (Li B A) C1"
by (simp add:Plane_diffside_def)
then obtain B1 :: Point where P16 : "Plane_diffside (Li B A) C1 D ⟹ Bet_Point (Se C C1) B1
∧ Line_on (Li B A) B1 ∧ ¬ Line_on (Li B A) C ∧ ¬ Line_on (Li B A) C1" by blast
from P16 have P17 : "Plane_diffside (Li B A) C1 D ⟹ Bet_Point (Se C C1) B1" by simp
then have P18 : "Plane_diffside (Li B A) C1 D ⟹ Line_on (Li C C1) B1" by (simp add:Line_Bet_on)
have P19 : "Line_on (Li B C1) C1" by (simp add:Line_on_rule)
have P20 : "Line_on (Li C C1) C" by (simp add:Line_on_rule)
have P21 : "Line_on (Li C C1) C1" by (simp add:Line_on_rule)
from assms P19 P20 P21 have P22 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi C1) add Emp) ⟹
Plane_diffside (Li B A) C1 D ⟹ Eq (Geos (Lin (Li B C1)) add Emp) (Geos (Lin (Li C C1)) add
Emp)" by (simp add:Line_unique)
from P9 P22 have P23 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi C1) add Emp) ⟹
Plane_diffside (Li B A) C1 D ⟹ Line_on (Li C C1) B" by (simp add:Line_on_trans)
from P21 have P24 : "Eq (Geos (Lin (Li C C1)) add Emp) (Geos (Lin (Li B A)) add Emp) ⟹ Line_on
(Li B A) C1" by (simp add:Line_on_trans)
from P16 P24 have P25 : "Plane_diffside (Li B A) C1 D ⟹
¬ Eq (Geos (Lin (Li C C1)) add Emp) (Geos (Lin (Li B A)) add Emp)" by blast
from P2 P16 P18 P23 P25 have P26 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi C1) add Emp) ⟹
Plane_diffside (Li B A) C1 D ⟹ Eq (Geos (Poi B1) add Emp) (Geos (Poi B) add Emp)" by (simp
add:Line_unique_Point)
from P17 P26 have P27 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi C1) add Emp) ⟹
Plane_diffside (Li B A) C1 D ⟹ Bet_Point (Se C C1) B" by (simp add:Point_Eq)
from assms P27 have P28 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi C1) add Emp) ⟹
¬ Plane_diffside (Li B A) C1 D" by blast
from assms P2 P9 P19 have P29 : "Line_on (Li B A) C1 ⟹
Eq (Geos (Lin (Li B C1)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (simp add:Line_unique)
from P12 P29 have P30 : "Line_on (Li B A) C1 ⟹
Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (blast intro:Eq_trans)
from P10 P30 have P31 : "Line_on (Li B A) C1 ⟹ Line_on (Li B A) C" by (simp add:Line_on_trans)
from P1 have P32 : "¬ Line_on (Li B A) C" by (simp add:Plane_sameside_def)
from P31 P32 have P33 : "¬ Line_on (Li B A) C1" by blast
from P1 have P34 : "¬ Line_on (Li B A) D" by (simp add:Plane_sameside_def)
from P12 P19 have "Line_on (Li B C) C1" by (blast intro:Line_on_trans Eq_rev)
then have P35 : "Eq (Geos (Poi C1) add Emp) (Geos (Poi D) add Emp) ⟹ Line_on (Li B C) D" by
(simp add:Point_Eq)
from P1 have P36 : "¬ Line_on (Li B C) D" by (simp add:Plane_sameside_def)
from P35 P36 have P37 : "¬ Eq (Geos (Poi C1) add Emp) (Geos (Poi D) add Emp)" by blast
from P28 P33 P34 P37 have P38 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi C1) add Emp) ⟹
Plane_sameside (Li B A) C1 D" by (simp add:Plane_not_diffside_sameside)
from P14 have "Eq (Geos (Poi C) add Emp) (Geos (Poi C1) add Emp) ⟹ Plane_sameside (Li B A)
D C1" by (simp add:Point_Eq)
then have P39 : "Eq (Geos (Poi C) add Emp) (Geos (Poi C1) add Emp) ⟹ Plane_sameside (Li B A)
C1 D" by (simp add:Plane_sameside_rev)
from P38 P39 have P40 : "Plane_sameside (Li B A) C1 D" by blast
from P7 P40 have P41 : "Plane_sameside (Li B A1) C1 D" by (simp add:Plane_Line_trans)
have P42 : "Plane_diffside (Li B C) A1 D ⟹ Plane_diffside (Li B C) D A1"
by (simp add:Plane_diffside_rev)
from P1 have P43 : "Plane_sameside (Li B C) D A" by (simp add:Plane_sameside_rev)
from P42 P43 have P44 : "Plane_diffside (Li B C) A1 D ⟹ Plane_diffside (Li B C) A A1" by (simp
add:Plane_trans)
then have "Plane_diffside (Li B C) A1 D ⟹ ∃p. Bet_Point (Se A A1) p
∧ Line_on (Li B C) p ∧ ¬ Line_on (Li B C) A ∧ ¬ Line_on (Li B C) A1"
by (simp add:Plane_diffside_def)
then obtain B2 :: Point where P45 : "Plane_diffside (Li B C) A1 D ⟹ Bet_Point (Se A A1) B2
∧ Line_on (Li B C) B2 ∧ ¬ Line_on (Li B C) A ∧ ¬ Line_on (Li B C) A1" by blast
from P45 have P46 : "Plane_diffside (Li B C) A1 D ⟹ Bet_Point (Se A A1) B2" by simp
then have P47 : "Plane_diffside (Li B C) A1 D ⟹ Line_on (Li A A1) B2" by (simp add:Line_Bet_on)
have P48 : "Line_on (Li B A1) A1" by (simp add:Line_on_rule)
have P49 : "Line_on (Li A A1) A" by (simp add:Line_on_rule)
have P50 : "Line_on (Li A A1) A1" by (simp add:Line_on_rule)
from assms P48 P49 P50 have P51 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi A1) add Emp) ⟹
Plane_diffside (Li B C) A1 D ⟹ Eq (Geos (Lin (Li B A1)) add Emp) (Geos (Lin (Li A A1)) add
Emp)" by (simp add:Line_unique)
from P3 P51 have P52 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi A1) add Emp) ⟹

```
    Plane_diffside (Li B C) A1 D ⟹ Line_on (Li A A1) B" by (simp add:Line_on_trans)
  from P50 have P53 : "Eq (Geos (Lin (Li A A1)) add Emp) (Geos (Lin (Li B C)) add Emp) ⟹
Line_on (Li B C) A1" by (simp add:Line_on_trans)
  from P45 P53 have P54 : "Plane_diffside (Li B C) A1 D ⟹
    ¬ Eq (Geos (Lin (Li A A1)) add Emp) (Geos (Lin (Li B C)) add Emp)" by blast
  from P8 P45 P47 P52 P54 have P55 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi A1) add Emp) ⟹
    Plane_diffside (Li B C) A1 D ⟹ Eq (Geos (Poi B2) add Emp) (Geos (Poi B) add Emp)" by (simp
add:Line_unique_Point)
  from P46 P55 have P56 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi A1) add Emp) ⟹
    Plane_diffside (Li B C) A1 D ⟹ Bet_Point (Se A A1) B" by (simp add:Point_Eq)
  from assms P56 have P57 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi A1) add Emp) ⟹
    ¬ Plane_diffside (Li B C) A1 D" by blast
  from assms P3 P8 P48 have P58 : "Line_on (Li B C) A1 ⟹
    Eq (Geos (Lin (Li B A1)) add Emp) (Geos (Lin (Li B C)) add Emp)" by (simp add:Line_unique)
  from P7 P58 have P59 : "Line_on (Li B C) A1 ⟹
    Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (blast intro:Eq_trans)
  from P10 P59 have P60 : "Line_on (Li B C) A1 ⟹ Line_on (Li B A) C" by (simp add:Line_on_trans)
  from P32 P60 have P61 : "¬ Line_on (Li B C) A1" by blast
  from P7 P48 have "Line_on (Li B A) A1" by (blast intro:Line_on_trans Eq_rev)
  then have P62 : "Eq (Geos (Poi A1) add Emp) (Geos (Poi D) add Emp) ⟹ Line_on (Li B A) D" by
(simp add:Point_Eq)
  from P34 P62 have P63 : "¬ Eq (Geos (Poi A1) add Emp) (Geos (Poi D) add Emp)" by blast
  from P36 P57 P61 P63 have P64 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi A1) add Emp) ⟹
    Plane_sameside (Li B C) A1 D" by (simp add:Plane_not_diffside_sameside)
  from P43 have "Eq (Geos (Poi A) add Emp) (Geos (Poi A1) add Emp) ⟹ Plane_sameside (Li B C)
D A1" by (simp add:Point_Eq)
  then have P65 : "Eq (Geos (Poi A) add Emp) (Geos (Poi A1) add Emp) ⟹ Plane_sameside (Li B C)
A1 D" by (simp add:Plane_sameside_rev)
  from P64 P65 have P66 : "Plane_sameside (Li B C) A1 D" by blast
  from P12 P66 have P67 : "Plane_sameside (Li B C1) A1 D" by (simp add:Plane_Line_trans)
  from P12 have P68 : "Line_on (Li B C1) A1 ⟹ Line_on (Li B C) A1" by (blast intro:Line_on_trans
Eq_rev)
  from P61 P68 have P69 : "¬ Line_on (Li B C1) A1" by blast
  from assms P69 have "Def (Ang (An B C1 A1))" by (simp add:Ang_sinple_def)
  then have P70 : "Def (Ang (An A1 B C1))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P41 P67 P70 show "Ang_inside (An A1 B C1) D" by (simp add:Ang_inside_def)
qed

lemma (in Congruence_Rule) Ang_sub_lemma1 :
  assumes
    "Plane_sameside (Li o1 l1) h1 k1"
    "¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi l1) add Emp)"
    "Plane_sameside (Li o2 l2) h2 k2"
    "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi l2) add Emp)"
    "Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An h2 o2 l2)) add Emp)"
    "Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An k2 o2 l2)) add Emp)"
    "¬ Eq (Geos (Lin (Li o1 h1)) add Emp) (Geos (Lin (Li o1 k1)) add Emp)"
    "¬ Eq (Geos (Lin (Li o2 h2)) add Emp) (Geos (Lin (Li o2 k2)) add Emp)"
    "Ang_inside (An k1 o1 l1) h1"
  shows
    "Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 k2)) add Emp)"
    "Ang_inside (An k2 o2 l2) h2"
proof -
  from assms have P1 : "¬ Line_on (Li o1 l1) h1" by (simp add:Plane_sameside_def)
  from assms P1 have "Def (Ang (An o1 l1 h1))" by (simp add:Ang_sinple_def)
  then have P2 : "Def (Ang (An h1 o1 l1))" by (blast intro:Ang_def_rev Ang_def_inv)
  from assms have P3 : "¬ Line_on (Li o1 l1) k1" by (simp add:Plane_sameside_def)
  from assms P3 have "Def (Ang (An o1 l1 k1))" by (simp add:Ang_sinple_def)
  then have P4 : "Def (Ang (An k1 o1 l1))" by (blast intro:Ang_def_rev Ang_def_inv)
  from assms have P5 : "¬ Line_on (Li o2 l2) h2" by (simp add:Plane_sameside_def)
  from assms P5 have "Def (Ang (An o2 l2 h2))" by (simp add:Ang_sinple_def)
  then have P6 : "Def (Ang (An h2 o2 l2))" by (blast intro:Ang_def_rev Ang_def_inv)
  from assms have P7 : "¬ Line_on (Li o2 l2) k2" by (simp add:Plane_sameside_def)
  from assms P7 have "Def (Ang (An o2 l2 k2))" by (simp add:Ang_sinple_def)
  then have P8 : "Def (Ang (An k2 o2 l2))" by (blast intro:Ang_def_rev Ang_def_inv)
  from assms P4 P8 have "∃p q. Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An p o2 q))
```

add Emp)
∧ Eq (Geos (Ang (An k2 o2 l2)) add Emp) (Geos (Ang (An p o2 q)) add Emp)
∧ Eq (Geos (Seg (Se o1 k1)) add Emp) (Geos (Seg (Se o2 p)) add Emp) ∧ Line_on (Li o2 k2) p ∧ ¬ Bet_Point (Se p k2) o2
∧ Eq (Geos (Seg (Se o1 l1)) add Emp) (Geos (Seg (Se o2 q)) add Emp) ∧ Line_on (Li o2 l2) q ∧ ¬ Bet_Point (Se q l2) o2 ∧ Def (Ang (An p o2 q))" by (simp add:Ang_replace)
then obtain k21 l21 :: Point where P9 : "Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An k21 o2 l21)) add Emp)
∧ Eq (Geos (Ang (An k2 o2 l2)) add Emp) (Geos (Ang (An k21 o2 l21)) add Emp)
∧ Eq (Geos (Seg (Se o1 k1)) add Emp) (Geos (Seg (Se o2 k21)) add Emp) ∧ Line_on (Li o2 k2) k21 ∧ ¬ Bet_Point (Se k21 k2) o2
∧ Eq (Geos (Seg (Se o1 l1)) add Emp) (Geos (Seg (Se o2 l21)) add Emp) ∧ Line_on (Li o2 l2) l21 ∧ ¬ Bet_Point (Se l21 l2) o2 ∧ Def (Ang (An k21 o2 l21))" by blast
from assms have "Plane_diffside (Li o1 h1) k1 l1" by (simp add:Ang_inside_Planeside)
then have "∃p. Bet_Point (Se k1 l1) p ∧ Line_on (Li o1 h1) p ∧ ¬ Line_on (Li o1 h1) k1 ∧ ¬ Line_on (Li o1 h1) l1" by (simp add:Plane_diffside_def)
then obtain h11 :: Point where P10 : "Bet_Point (Se k1 l1) h11 ∧ Line_on (Li o1 h1) h11 ∧ ¬ Line_on (Li o1 h1) k1 ∧ ¬ Line_on (Li o1 h1) l1" by blast
then have "Eq (Geos (Poi h11) add Emp) (Geos (Poi o1) add Emp) ⟹
Bet_Point (Se k1 l1) o1" by (blast intro:Point_Eq)
then have P11 : "Eq (Geos (Poi h11) add Emp) (Geos (Poi o1) add Emp) ⟹
Line_on (Li o1 l1) k1" by (simp add:Line_Bet_on)
from P3 P11 have P12 : "¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi h11) add Emp)" by (blast intro:Eq_rev)
have P13 : "Line_on (Li o2 h2) o2" by (simp add:Line_on_rule)
have P14 : "Line_on (Li o2 h2) h2" by (simp add:Line_on_rule)
from P6 have "¬ Eq (Geos (Poi h2) add Emp) (Geos (Poi o2) add Emp)" by (simp add:Ang_def)
then have P15 : "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi h2) add Emp)" by (blast intro:Eq_rev)
from P12 P13 P14 P15 have "∃p. Eq (Geos (Seg (Se o1 h11)) add Emp) (Geos (Seg (Se o2 p)) add Emp)
∧ ¬ Bet_Point (Se p h2) o2 ∧ Line_on (Li o2 h2) p ∧ ¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi p) add Emp)" by (simp add:Seg_move_sameside)
then obtain h21 :: Point where P16 : "Eq (Geos (Seg (Se o1 h11)) add Emp) (Geos (Seg (Se o2 h21)) add Emp)
∧ ¬ Bet_Point (Se h21 h2) o2 ∧ Line_on (Li o2 h2) h21 ∧ ¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi h21) add Emp)" by blast
have P17 : "Line_on (Li o1 l1) o1" by (simp add:Line_on_rule)
have "Line_on (Li h1 h11) h1" by (simp add:Line_on_rule)
then have P18 : "Eq (Geos (Lin (Li h1 h11)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹ Line_on (Li o1 l1) h1" by (simp add:Line_on_trans)
from P1 P18 have P19 : "¬ Eq (Geos (Lin (Li h1 h11)) add Emp) (Geos (Lin (Li o1 l1)) add Emp)" by blast
from P17 P19 have "Bet_Point (Se h1 h11) o1 ⟹ Plane_diffside (Li o1 l1) h1 h11" by (simp add:Plane_Bet_diffside)
then have P20 : "Bet_Point (Se h1 h11) o1 ⟹ Plane_diffside (Li o1 l1) h11 h1"
by (simp add:Plane_diffside_rev)
from P10 have P21 : "Bet_Point (Se l1 k1) h11" by (simp add:Bet_rev)
have P22 : "Line_on (Li o1 l1) l1" by (simp add:Line_on_rule)
have "Line_on (Li l1 k1) k1" by (simp add:Line_on_rule)
then have P23 : "Eq (Geos (Lin (Li l1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹ Line_on (Li o1 l1) k1" by (simp add:Line_on_trans)
from P3 P23 have P24 : "¬ Eq (Geos (Lin (Li l1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp)" by blast
from P21 P22 P24 have P25 : "Plane_sameside (Li o1 l1) h11 k1" by (simp add:Plane_Bet_sameside)
from P20 P25 have "Bet_Point (Se h1 h11) o1 ⟹ Plane_diffside (Li o1 l1) h1 k1"
by (simp add:Plane_trans Plane_diffside_rev)
then have P26 : "Bet_Point (Se h1 h11) o1 ⟹ ¬ Plane_sameside (Li o1 l1) h1 k1" by (simp add:Plane_diffside_not_sameside)
from assms P26 have P27 : "¬ Bet_Point (Se h1 h11) o1" by blast
have P28 : "¬ Bet_Point (Se l1 l1) o1" by (simp add:Bet_end_Point)
from assms P2 P10 P22 P27 P28 P12 have P29 : "Eq (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An h11 o1 l1)) add Emp) ∧ Def (Ang (An h11 o1 l1))" by (simp add:Ang_Point_swap)
from P9 have P30 : "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi l21) add Emp)" by (simp add:Ang_def)
from P16 have P31 : "¬ Bet_Point (Se h2 h21) o2" by (blast intro:Bet_rev)
from P9 have P32 : "¬ Bet_Point (Se l2 l21) o2" by (blast intro:Bet_rev)
from P6 P9 P16 P30 P31 P32 have P33 : "Eq (Geos (Ang (An h2 o2 l2)) add Emp) (Geos (Ang (An

h21 o2 l21)) add Emp) ∧ Def (Ang (An h21 o2 l21))" by (simp add:Ang_Point_swap)
from assms P29 have P34 : "Cong (Geos (Ang (An h11 o1 l1)) add Emp) (Geos (Ang (An h2 o2 l2)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P33 P34 have P35 : "Cong (Geos (Ang (An h11 o1 l1)) add Emp) (Geos (Ang (An h21 o2 l21)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P29 have "Def (Tri (Tr h11 o1 l1))" by (simp add:Ang_to_Tri)
then have P36 : "Def (Tri (Tr o1 h11 l1))" by (blast intro:Tri_def_rev Tri_def_trans)
from P33 have "Def (Tri (Tr h21 o2 l21))" by (simp add:Ang_to_Tri)
then have P37 : "Def (Tri (Tr o2 h21 l21))" by (blast intro:Tri_def_rev Tri_def_trans)
from P9 P16 P35 P36 P37 have P38 : "Cong (Geos (Tri (Tr o1 h11 l1)) add Emp) (Geos (Tri (Tr o2 h21 l21)) add Emp)" by (simp add:Tri_SAS)
then have P39 : "Cong (Geos (Ang (An o1 l1 h11)) add Emp) (Geos (Ang (An o2 l21 h21)) add Emp)" by (simp add:Tri_Cong_def)
from P4 have P40 : "Def (Tri (Tr o1 k1 l1))" by (simp add:Ang_to_Tri Tri_def_rev Tri_def_trans)
from P9 have P41 : "Def (Tri (Tr o2 k21 l21))" by (simp add:Ang_to_Tri Tri_def_rev Tri_def_trans)
from P9 P40 P41 have P42 : "Cong (Geos (Tri (Tr o1 k1 l1)) add Emp) (Geos (Tri (Tr o2 k21 l21)) add Emp)" by (simp add:Tri_SAS)
then have P43 : "Cong (Geos (Ang (An o1 l1 k1)) add Emp) (Geos (Ang (An o2 l21 k21)) add Emp)" by (simp add:Tri_Cong_def)
from P4 have P44 : "Def (Ang (An o1 l1 k1))" by (blast intro:Ang_def_rev Ang_def_inv)
have P45 : "Line_on (Li l1 o1) o1" by (simp add:Line_on_rule)
have P46 : "¬ Bet_Point (Se o1 o1) l1" by (simp add:Bet_end_Point)
from P10 have P47 : "Line_on (Li l1 k1) h11" by (simp add:Line_Bet_on)
from P10 have "Inv(Bet_Point (Se h11 k1) l1)" by (simp add:Bet_iff)
then have "¬ Bet_Point (Se h11 k1) l1" by (simp add:Inv_def)
then have P48 : "¬ Bet_Point (Se k1 h11) l1" by (blast intro:Bet_rev)
from assms have P49 : "¬ Eq (Geos (Poi l1) add Emp) (Geos (Poi o1) add Emp)" by (blast intro:Eq_rev)
from P10 have "Bet_Point (Se k1 l1) h11" by simp
then have P50 : "¬ Eq (Geos (Poi l1) add Emp) (Geos (Poi h11) add Emp)" by (simp add:Bet_Point_def)
from P44 P45 P46 P47 P48 P49 P50 have P51 : "Eq (Geos (Ang (An o1 l1 k1)) add Emp) (Geos (Ang (An o1 l1 h11)) add Emp) ∧ Def (Ang (An o1 l1 h11))" by (simp add:Ang_Point_swap)
have P52 : "Eq (Geos (Ang (An o2 l21 k21)) add Emp) (Geos (Ang (An k21 l21 o2)) add Emp)" by (simp add:Ang_roll)
from P43 P52 have P53 : "Cong (Geos (Ang (An o1 l1 k1)) add Emp) (Geos (Ang (An k21 l21 o2)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P51 P53 have P54 : "Cong (Geos (Ang (An o1 l1 h11)) add Emp) (Geos (Ang (An k21 l21 o2)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
have P55 : "Eq (Geos (Ang (An o2 l21 h21)) add Emp) (Geos (Ang (An h21 l21 o2)) add Emp)" by (simp add:Ang_roll)
from P39 P55 have P56 : "Cong (Geos (Ang (An o1 l1 h11)) add Emp) (Geos (Ang (An h21 l21 o2)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
have P57 : "Line_on (Li o2 l2) o2" by (simp add:Line_on_rule)
have P58 : "Line_on (Li l21 o2) o2" by (simp add:Line_on_rule)
have P59 : "Line_on (Li l21 o2) l21" by (simp add:Line_on_rule)
from P9 have P60 : "Line_on (Li o2 l2) l21" by simp
from P30 P57 P58 P59 P60 have P61 : "Eq (Geos (Lin (Li o2 l2)) add Emp) (Geos (Lin (Li l21 o2)) add Emp)" by (simp add:Line_unique)
from assms P16 P57 have P62 : "¬ Eq (Geos (Poi h2) add Emp) (Geos (Poi h21) add Emp) ⟹
Plane_sameside (Li o2 l2) h21 h2" by (blast intro:Plane_sameside_HalfLine Plane_sameside_rev)
from assms have P63 : "Plane_sameside (Li o2 l2) k2 h2" by (simp add:Plane_sameside_rev)
from P41 have P64 : "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi k21) add Emp)" by (simp add:Tri_def)
from P9 P57 P63 P64 have P65 : "¬ Eq (Geos (Poi k2) add Emp) (Geos (Poi k21) add Emp) ⟹
Plane_sameside (Li o2 l2) k2 k21" by (simp add:Plane_sameside_HalfLine)
then have P66 : "¬ Eq (Geos (Poi k2) add Emp) (Geos (Poi k21) add Emp) ⟹
Plane_sameside (Li o2 l2) k21 k2" by (simp add:Plane_sameside_rev)
from assms P62 have P67 : "¬ Eq (Geos (Poi h2) add Emp) (Geos (Poi h21) add Emp) ⟹
¬ Eq (Geos (Poi k2) add Emp) (Geos (Poi h21) add Emp) ⟹
Plane_sameside (Li o2 l2) h21 k2" by (blast intro:Plane_sameside_trans)
have P68 : "Line_on (Li o2 k2) o2" by (simp add:Line_on_rule)
from P9 have P69 : "Eq (Geos (Poi k21) add Emp) (Geos (Poi h21) add Emp) ⟹ Line_on (Li o2 k2) h21" by (blast intro:Point_Eq)
from P9 P13 P16 P68 P69 have P70 : "Eq (Geos (Poi k21) add Emp) (Geos (Poi h21) add Emp) ⟹
Eq (Geos (Lin (Li o2 h2)) add Emp) (Geos (Lin (Li o2 k2)) add Emp)" by (blast intro:Line_unique)
from assms P70 have P71 : "¬ Eq (Geos (Poi k21) add Emp) (Geos (Poi h21) add Emp)" by blast
from P65 P67 P71 have P72 : "¬ Eq (Geos (Poi h2) add Emp) (Geos (Poi h21) add Emp) ⟹

¬ Eq (Geos (Poi k2) add Emp) (Geos (Poi k21) add Emp) ⟹
¬ Eq (Geos (Poi k2) add Emp) (Geos (Poi h21) add Emp) ⟹
Plane_sameside (Li o2 l2) k21 h21" by (blast intro:Plane_sameside_trans Plane_sameside_rev)
from P66 have P73 : "¬ Eq (Geos (Poi k2) add Emp) (Geos (Poi k21) add Emp) ⟹
Eq (Geos (Poi k2) add Emp) (Geos (Poi h21) add Emp) ⟹
Plane_sameside (Li o2 l2) k21 h21" by (simp add:Point_Eq)
from P71 P72 P73 have P74 : "¬ Eq (Geos (Poi h2) add Emp) (Geos (Poi h21) add Emp) ⟹
¬ Eq (Geos (Poi k2) add Emp) (Geos (Poi k21) add Emp) ⟹
Plane_sameside (Li o2 l2) k21 h21" by blast
from P63 have P75 : "Eq (Geos (Poi h2) add Emp) (Geos (Poi h21) add Emp) ⟹
¬ Eq (Geos (Poi k2) add Emp) (Geos (Poi k21) add Emp) ⟹
Plane_sameside (Li o2 l2) k2 h21" by (simp add:Point_Eq)
from P66 P71 P75 have P76 : "Eq (Geos (Poi h2) add Emp) (Geos (Poi h21) add Emp) ⟹
¬ Eq (Geos (Poi k2) add Emp) (Geos (Poi k21) add Emp) ⟹
Plane_sameside (Li o2 l2) k21 h21" by (blast intro:Plane_sameside_trans Eq_rev)
from assms have P77 : "Eq (Geos (Poi k2) add Emp) (Geos (Poi k21) add Emp) ⟹
Plane_sameside (Li o2 l2) h2 k21" by (simp add:Point_Eq)
from P62 P71 P77 have P78 : "¬ Eq (Geos (Poi h2) add Emp) (Geos (Poi h21) add Emp) ⟹
Eq (Geos (Poi k2) add Emp) (Geos (Poi k21) add Emp) ⟹
Plane_sameside (Li o2 l2) k21 h21" by (blast intro:Plane_sameside_trans Plane_sameside_rev)
from P77 have P79 : "Eq (Geos (Poi h2) add Emp) (Geos (Poi h21) add Emp) ⟹
Eq (Geos (Poi k2) add Emp) (Geos (Poi k21) add Emp) ⟹
Plane_sameside (Li o2 l2) k21 h21" by (blast intro:Point_Eq Plane_sameside_rev)
from P71 P74 P76 P78 P79 have P80 : "Plane_sameside (Li o2 l2) k21 h21" by blast
from P61 P74 P80 have P81 : "Plane_sameside (Li l21 o2) k21 h21" by (simp add:Plane_Line_trans)
from P54 P56 P74 P81 have P82 : "Eq (Geos (Lin (Li k21 l21)) add Emp) (Geos (Lin (Li h21 l21)) add Emp) ∧ ¬ Bet_Point (Se k21 h21) l21" by (simp add:Ang_move_unique)
have P83 : "Line_on (Li h21 l21) h21" by (simp add:Line_on_rule)
from P74 P82 P83 have P84 : "Line_on (Li k21 l21) h21" by (blast intro:Line_on_trans Eq_rev)
have P85 : "Line_on (Li k21 l21) k21" by (simp add:Line_on_rule)
have P86 : "Line_on (Li k21 l21) l21" by (simp add:Line_on_rule)
from P9 have "¬ Eq (Geos (Poi l21) add Emp) (Geos (Poi k21) add Emp)" by (simp add:Ang_def)
then have P87 : "¬ Eq (Geos (Poi k21) add Emp) (Geos (Poi l21) add Emp)" by (blast intro:Eq_rev)
from P33 have P88 : "¬ Eq (Geos (Poi l21) add Emp) (Geos (Poi h21) add Emp)" by (simp add:Ang_def)
from P71 have P89 : "¬ Eq (Geos (Poi h21) add Emp) (Geos (Poi k21) add Emp)" by (blast intro:Eq_rev)
from P81 have P90 : "¬ Line_on (Li l21 o2) k21" by (simp add:Plane_sameside_def)
from P85 have P91 : "Eq (Geos (Lin (Li k21 l21)) add Emp) (Geos (Lin (Li l21 o2)) add Emp) ⟹
Line_on (Li l21 o2) k21" by (simp add:Line_on_trans)
from P90 P91 have P92 : "¬ Eq (Geos (Lin (Li l21 o2)) add Emp) (Geos (Lin (Li k21 l21)) add Emp)" by (blast intro:Eq_rev)
from assms P59 P81 P84 P85 P86 P87 P88 P89 P92 have P93 : "Bet_Point (Se h21 l21) k21 ∨ Bet_Point (Se l21 k21) h21" by (simp add:Plane_Bet_sameside_rev)
from P10 have P94 : "Bet_Point (Se k1 l1) h11" by simp
then have P95 : "Bet_Point (Se l1 k1) h11" by (simp add:Bet_rev)
from P42 have P96 : "Eq (Geos (Seg (Se k1 l1)) add Emp) (Geos (Seg (Se k21 l21)) add Emp)" by (simp add:Tri_Cong_def)
then have P97 : "Eq (Geos (Seg (Se l1 k1)) add Emp) (Geos (Seg (Se l21 k21)) add Emp)" by (blast intro:Eq_rev Eq_trans Seg_rev)
from P38 have "Eq (Geos (Seg (Se h11 l1)) add Emp) (Geos (Seg (Se h21 l21)) add Emp)" by (simp add:Tri_Cong_def)
then have P98 : "Eq (Geos (Seg (Se l1 h11)) add Emp) (Geos (Seg (Se l21 h21)) add Emp)" by (blast intro:Eq_rev Eq_trans Seg_rev)
from P95 P97 P98 have P99 : "¬ Bet_Point (Se l21 h21) k21" by (blast intro:Seg_Bet_wrong_relation)
then have P100 : "¬ Bet_Point (Se h21 l21) k21" by (blast intro:Bet_rev)
from P93 P100 have "Bet_Point (Se l21 k21) h21" by blast
then have P101 : "¬ Seg_on_Seg (Se k21 h21) (Se h21 l21)" by (simp add:Bet_rev Seg_Bet_not_on)
have P102 : "Line_on (Li k1 l1) k1" by (simp add:Line_on_rule)
have P103 : "Line_on (Li k1 l1) l1" by (simp add:Line_on_rule)
from P94 have P104 : "Line_on (Li k1 l1) h11" by (simp add:Line_Bet_on)
from P94 have P105 : "¬ Seg_on_Seg (Se k1 h11) (Se h11 l1)" by (simp add:Seg_Bet_not_on)
from assms P84 P85 P86 P96 P101 P102 P103 P104 P105 have P106 : "Eq (Geos (Seg (Se k1 h11)) add Emp) (Geos (Seg (Se k21 h21)) add Emp)" by (simp add:Seg_sub)
from P42 have P107 : "Cong (Geos (Ang (An l1 k1 o1)) add Emp) (Geos (Ang (An l21 k21 o2)) add Emp)" by (simp add:Tri_Cong_def)
from P4 have P108 : "Def (Ang (An l1 k1 o1))" by (blast intro:Ang_def_rev Ang_def_inv)

from P94 have ”Inv (Bet_Point (Se l1 h11) k1)” by (simp add:Bet_iff)
then have P109 : ”¬ Bet_Point (Se l1 h11) k1” by (simp add:Inv_def)
have P110 : ”Line_on (Li k1 o1) o1” by (simp add:Line_on_rule)
have P111 : ”¬ Bet_Point (Se o1 o1) k1” by (simp add:Bet_end_Point)
from P94 have ”¬ Eq (Geos (Poi h11) add Emp) (Geos (Poi k1) add Emp)” by (simp add:Bet_Point_def)
then have P112 : ”¬ Eq (Geos (Poi k1) add Emp) (Geos (Poi h11) add Emp)” by (blast intro:Eq_rev)
from P108 have P113 : ”¬ Eq (Geos (Poi k1) add Emp) (Geos (Poi o1) add Emp)” by (simp add:Ang_def)
from P104 P108 P109 P110 P111 P112 P113 have P114 : ”Eq (Geos (Ang (An l1 k1 o1)) add Emp) (Geos (Ang (An h11 k1 o1)) add Emp) ∧ Def (Ang (An h11 k1 o1))” by (simp add:Ang_Point_swap)
from P9 have P115 : ”Def (Ang (An l21 k21 o2))” by (blast intro:Ang_def_rev Ang_def_inv)
have P116 : ”Line_on (Li k21 o2) o2” by (simp add:Line_on_rule)
have P117 : ”¬ Bet_Point (Se o2 o2) k21” by (simp add:Bet_end_Point)
from P64 have P119 : ”¬ Eq (Geos (Poi k21) add Emp) (Geos (Poi o2) add Emp)” by (blast intro:Eq_rev)
from assms P71 P84 P99 P115 P116 P117 P119 have P120 : ”Eq (Geos (Ang (An l21 k21 o2)) add Emp) (Geos (Ang (An h21 k21 o2)) add Emp) ∧ Def (Ang (An h21 k21 o2))” by (simp add:Ang_Point_swap)
from P107 P114 have P121 : ”Cong (Geos (Ang (An h11 k1 o1)) add Emp) (Geos (Ang (An l21 k21 o2)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P120 P121 have P122 : ” Cong (Geos (Ang (An h11 k1 o1)) add Emp) (Geos (Ang (An h21 k21 o2)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P114 have P123 : ”Def (Tri (Tr k1 h11 o1))” by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
from P120 have P124 : ”Def (Tri (Tr k21 h21 o2))” by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
from P9 have P125 : ”Eq (Geos (Seg (Se k1 o1)) add Emp) (Geos (Seg (Se k21 o2)) add Emp)” by (blast intro:Seg_rev Eq_trans Eq_rev)
from P106 P122 P123 P124 P125 have ”Cong (Geos (Tri (Tr k1 h11 o1)) add Emp) (Geos (Tri (Tr k21 h21 o2)) add Emp)” by (simp add:Tri_SAS)
then have P126 : ”Cong (Geos (Ang (An k1 o1 h11)) add Emp) (Geos (Ang (An k21 o2 h21)) add Emp)” by (simp add:Tri_Cong_def)
have P127 : ”Eq (Geos (Ang (An k1 o1 h11)) add Emp) (Geos (Ang (An h11 o1 k1)) add Emp)” by (simp add:Ang_roll)
have P128 : ”Eq (Geos (Ang (An k21 o2 h21)) add Emp) (Geos (Ang (An h21 o2 k21)) add Emp)” by (simp add:Ang_roll)
from P126 P127 have P129 : ”Cong (Geos (Ang (An h11 o1 k1)) add Emp) (Geos (Ang (An k21 o2 h21)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P128 P129 have P130 : ”Cong (Geos (Ang (An h11 o1 k1)) add Emp) (Geos (Ang (An h21 o2 k21)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P114 have P131 : ”Def (Ang (An h11 o1 k1))” by (simp add:Ang_def_inv)
from P2 have ”¬ Eq (Geos (Poi h1) add Emp) (Geos (Poi o1) add Emp)” by (simp add:Ang_def)
then have P132 : ”¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi h1) add Emp)” by (blast intro:Eq_rev)
from P10 P12 P132 have P133 : ”Line_on (Li o1 h11) h1” by (blast intro:Line_on_rev)
from P27 have P134 : ”¬ Bet_Point (Se h11 h1) o1” by (blast intro:Bet_rev)
have P135 : ”Line_on (Li o1 k1) k1” by (simp add:Line_on_rule)
have P136 : ”¬ Bet_Point (Se k1 k1) o1” by (simp add:Bet_end_Point)
from P4 have ”¬ Eq (Geos (Poi k1) add Emp) (Geos (Poi o1) add Emp)” by (simp add:Ang_def)
then have P137 : ”¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi k1) add Emp)” by (blast intro:Eq_rev)
from P131 P132 P133 P134 P135 P136 P137 have P138 : ”Eq (Geos (Ang (An h11 o1 k1)) add Emp) (Geos (Ang (An h1 o1 k1)) add Emp) ∧ Def (Ang (An h1 o1 k1))” by (simp add:Ang_Point_swap)
from P120 have P139 : ”Def (Ang (An h21 o2 k21))” by (simp add:Ang_def_inv)
from P15 P16 have P140 : ”Line_on (Li o2 h21) h2” by (simp add:Line_on_rev)
from P16 have P141 : ”¬ Bet_Point (Se h21 h2) o2” by (blast intro:Bet_rev)
from P8 have ”¬ Eq (Geos (Poi k2) add Emp) (Geos (Poi o2) add Emp)” by (simp add:Ang_def)
then have P142 : ”¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi k2) add Emp)” by (blast intro:Eq_rev)
from P9 P64 P142 have P143 : ”Line_on (Li o2 k21) k2” by (simp add:Line_on_rev)
from P9 P15 P139 P140 P141 P142 P143 have P143 : ”Eq (Geos (Ang (An h21 o2 k21)) add Emp) (Geos (Ang (An h2 o2 k2)) add Emp) ∧ Def (Ang (An h2 o2 k2))” by (simp add:Ang_Point_swap)
from P130 P138 have P145 : ”Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h21 o2 k21)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P143 P145 show ”Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 k2)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P93 P100 have P146 : ”Bet_Point (Se k21 l21) h21” by (blast intro:Bet_rev)
from P9 have P147 : ”¬ Eq (Geos (Lin (Li o2 k21)) add Emp) (Geos (Lin (Li o2 l21)) add Emp)”
by (simp add:Ang_def)
from P30 P64 P146 P147 have P148 : ”Ang_inside (An k21 o2 l21) h21”
by (simp add:Ang_inside_Bet_Point)
from P8 have P149 : ”¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi l2) add Emp)” by (simp add:Ang_def)
from P9 P142 P148 P149 have P150 : ”Ang_inside (An k2 o2 l2) h21” by (simp add:Ang_inside_trans)

```
  from P15 P16 have P151 : "Line_on (Li o2 h21) h2" by (simp add:Line_on_rev)
  from P16 have P152 : "¬ Bet_Point (Se h2 h21) o2" by (blast intro:Bet_rev)
  from P15 P150 P151 P152 show "Ang_inside (An k2 o2 l2) h2" by (simp add:Ang_inside_HalfLine)
qed

theorem (in Congruence_Rule) Ang_sub :
  assumes
    "Plane_sameside (Li o1 l1) h1 k1"
    "¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi l1) add Emp)"
    "Plane_sameside (Li o2 l2) h2 k2"
    "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi l2) add Emp)"
    "Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An h2 o2 l2)) add Emp)"
    "Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An k2 o2 l2)) add Emp)"
    "¬ Eq (Geos (Lin (Li o1 h1)) add Emp) (Geos (Lin (Li o1 k1)) add Emp)"
    "¬ Eq (Geos (Lin (Li o2 h2)) add Emp) (Geos (Lin (Li o2 k2)) add Emp)"
  shows
    "Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 k2)) add Emp)"
proof -
  from assms have P1 : "¬ Line_on (Li o1 l1) h1" by (simp add:Plane_sameside_def)
  from assms P1 have "Def (Ang (An o1 l1 h1))" by (simp add:Ang_sinple_def)
  then have P2 : "Def (Ang (An h1 o1 l1))" by (blast intro:Ang_def_rev Ang_def_inv)
  from assms have P3 : "¬ Line_on (Li o1 l1) k1" by (simp add:Plane_sameside_def)
  from assms P3 have "Def (Ang (An o1 l1 k1))" by (simp add:Ang_sinple_def)
  then have P4 : "Def (Ang (An k1 o1 l1))" by (blast intro:Ang_def_rev Ang_def_inv)
  from assms P2 P4 have P5 : "Ang_inside (An h1 o1 l1) k1 ∧ ¬ Ang_inside (An k1 o1 l1) h1
    ∨ ¬ Ang_inside (An h1 o1 l1) k1 ∧ Ang_inside (An k1 o1 l1) h1" by (simp add:Ang_inside_case)
  from assms have P6 : "Ang_inside (An k1 o1 l1) h1 ⟹
    Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 k2)) add Emp)" by (simp
add:Ang_sub_lemma1)
  from assms have P7 : "Plane_sameside (Li o1 l1) k1 h1" by (simp add:Plane_sameside_rev)
  from assms have P8 : "Plane_sameside (Li o2 l2) k2 h2" by (simp add:Plane_sameside_rev)
  from assms have P9 : "¬ Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 h1)) add Emp)" by
(blast intro:Eq_rev)
  from assms have P10 : "¬ Eq (Geos (Lin (Li o2 k2)) add Emp) (Geos (Lin (Li o2 h2)) add Emp)" by
(blast intro:Eq_rev)
  from assms P7 P8 P9 P10 have P11 : "Ang_inside (An h1 o1 l1) k1 ⟹
    Cong (Geos (Ang (An k1 o1 h1)) add Emp) (Geos (Ang (An k2 o2 h2)) add Emp)" by (simp
add:Ang_sub_lemma1)
  have P12 : "Eq (Geos (Ang (An k1 o1 h1)) add Emp) (Geos (Ang (An h1 o1 k1)) add Emp)" by (simp
add:Ang_roll)
  have P13 : "Eq (Geos (Ang (An k2 o2 h2)) add Emp) (Geos (Ang (An h2 o2 k2)) add Emp)" by (simp
add:Ang_roll)
  from P11 P12 have P14 : "Ang_inside (An h1 o1 l1) k1 ⟹
    Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An k2 o2 h2)) add Emp)" by (blast in-
tro:Ang_weektrans Ang_rev Eq_rev)
  from P13 P14 have P15 : "Ang_inside (An h1 o1 l1) k1 ⟹
    Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 k2)) add Emp)" by (blast in-
tro:Ang_weektrans Ang_rev Eq_rev)
  from P5 P6 P15 show "Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 k2)) add
Emp)" by blast
qed

theorem (in Congruence_Rule) Ang_add :
  assumes
    "Plane_diffside (Li o1 l1) h1 k1"
    "¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi l1) add Emp)"
    "Plane_diffside (Li o2 l2) h2 k2"
    "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi l2) add Emp)"
    "Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An h2 o2 l2)) add Emp)"
    "Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An k2 o2 l2)) add Emp)"
    "¬ Eq (Geos (Lin (Li o1 h1)) add Emp) (Geos (Lin (Li o1 k1)) add Emp)"
    "¬ Eq (Geos (Lin (Li o2 h2)) add Emp) (Geos (Lin (Li o2 k2)) add Emp)"
  shows
    "Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 k2)) add Emp)"
proof -
  from assms have "∃p. Bet_Point (Se h1 k1) p ∧ Line_on (Li o1 l1) p
```

∧ ¬ Line_on (Li o1 l1) h1 ∧ ¬ Line_on (Li o1 l1) k1" by (simp add:Plane_diffside_def)
then obtain l11 :: Point where P1 : "Bet_Point (Se h1 k1) l11 ∧ Line_on (Li o1 l1) l11
∧ ¬ Line_on (Li o1 l1) h1 ∧ ¬ Line_on (Li o1 l1) k1" by blast
have P2 : "Line_on (Li o1 l1) o1" by (simp add:Line_on_rule)
then have P3 : "Eq (Geos (Poi o1) add Emp) (Geos (Poi k1) add Emp) ⟹ Line_on (Li o1 l1) k1" by (simp add:Point_Eq)
from P1 P3 have P4 : "¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi k1) add Emp)" by blast
have P5 : "Line_on (Li o1 k1) o1" by (simp add:Line_on_rule)
have P6 : "Line_on (Li o1 k1) k1" by (simp add:Line_on_rule)
from P4 P5 P6 have "∃p. Eq (Geos (Seg (Se o1 k1)) add Emp) (Geos (Seg (Se o1 p)) add Emp)
∧ Bet_Point (Se p k1) o1 ∧ Line_on (Li o1 k1) p ∧ ¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi p) add Emp)" by (simp add:Seg_move_diffside)
then obtain k11 :: Point where P7 : "Eq (Geos (Seg (Se o1 k1)) add Emp) (Geos (Seg (Se o1 k11)) add Emp)
∧ Bet_Point (Se k11 k1) o1 ∧ Line_on (Li o1 k1) k11 ∧ ¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi k11) add Emp)" by blast
from P7 have P8 : "Bet_Point (Se k11 k1) o1" by blast
have "Line_on (Li k11 k1) k1" by (simp add:Line_on_rule)
then have P9 : "Eq (Geos (Lin (Li k11 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹ Line_on (Li o1 l1) k1" by (simp add:Line_on_trans)
from P1 P9 have P10 : "¬ Eq (Geos (Lin (Li k11 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp)" by blast
from P2 P8 P10 have "Plane_diffside (Li o1 l1) k11 k1" by (simp add:Plane_Bet_diffside)
then have P11 : "Plane_diffside (Li o1 l1) k1 k11" by (simp add:Plane_diffside_rev)
from assms have P12 : "Plane_diffside (Li o1 l1) k1 h1" by (simp add:Plane_diffside_rev)
from P2 have P13 : "Eq (Geos (Poi o1) add Emp) (Geos (Poi h1) add Emp) ⟹ Line_on (Li o1 l1) h1" by (simp add:Point_Eq)
from P1 P13 have P14 : "¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi h1) add Emp)" by blast
have P15 : "Line_on (Li o1 h1) o1" by (simp add:Line_on_rule)
have P16 : "Line_on (Li o1 h1) h1" by (simp add:Line_on_rule)
from P5 P14 P15 P16 have P17 : "Line_on (Li o1 k1) h1
⟹ Eq (Geos (Lin (Li o1 h1)) add Emp) (Geos (Lin (Li o1 k1)) add Emp)" by (simp add:Line_unique)
from assms P17 have P18 : "¬ Line_on (Li o1 k1) h1" by blast
from P7 have P19 : "Eq (Geos (Poi k11) add Emp) (Geos (Poi h1) add Emp) ⟹ Line_on (Li o1 k1) h1" by (blast intro:Point_Eq)
from P18 P19 have P20 : "¬ Eq (Geos (Poi k11) add Emp) (Geos (Poi h1) add Emp)" by blast
from P11 P12 P20 have "Plane_sameside (Li o1 l1) k11 h1" by (blast intro:Plane_trans_inv)
then have P21 : "Plane_sameside (Li o1 l1) h1 k11" by (simp add:Plane_sameside_rev)
from assms have "∃p. Bet_Point (Se h2 k2) p ∧ Line_on (Li o2 l2) p
∧ ¬ Line_on (Li o2 l2) h2 ∧ ¬ Line_on (Li o2 l2) k2" by (simp add:Plane_diffside_def)
then obtain l21 :: Point where P22 : "Bet_Point (Se h2 k2) l21 ∧ Line_on (Li o2 l2) l21
∧ ¬ Line_on (Li o2 l2) h2 ∧ ¬ Line_on (Li o2 l2) k2" by blast
have P23 : "Line_on (Li o2 l2) o2" by (simp add:Line_on_rule)
then have P24 : "Eq (Geos (Poi o2) add Emp) (Geos (Poi k2) add Emp) ⟹ Line_on (Li o2 l2) k2" by (simp add:Point_Eq)
from P22 P24 have P25 : "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi k2) add Emp)" by blast
have P26 : "Line_on (Li o2 k2) o2" by (simp add:Line_on_rule)
have P27 : "Line_on (Li o2 k2) k2" by (simp add:Line_on_rule)
from P4 P25 P26 P27 have "∃p. Eq (Geos (Seg (Se o1 k1)) add Emp) (Geos (Seg (Se o2 p)) add Emp)
∧ Bet_Point (Se p k2) o2 ∧ Line_on (Li o2 k2) p ∧ ¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi p) add Emp)" by (simp add:Seg_move_diffside)
then obtain k21 :: Point where P28 : "Eq (Geos (Seg (Se o1 k1)) add Emp) (Geos (Seg (Se o2 k21)) add Emp)
∧ Bet_Point (Se k21 k2) o2 ∧ Line_on (Li o2 k2) k21 ∧ ¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi k21) add Emp)" by blast
from P28 have P29 : "Bet_Point (Se k21 k2) o2" by blast
have "Line_on (Li k21 k2) k2" by (simp add:Line_on_rule)
then have P30 : "Eq (Geos (Lin (Li k21 k2)) add Emp) (Geos (Lin (Li o2 l2)) add Emp) ⟹ Line_on (Li o2 l2) k2" by (simp add:Line_on_trans)
from assms have "∃p. Bet_Point (Se h2 k2) p ∧ Line_on (Li o2 l2) p
∧ ¬ Line_on (Li o2 l2) h2 ∧ ¬ Line_on (Li o2 l2) k2" by (simp add:Plane_diffside_def)
from P22 P30 have P31 : "¬ Eq (Geos (Lin (Li k21 k2)) add Emp) (Geos (Lin (Li o2 l2)) add Emp)" by blast
from P23 P29 P31 have "Plane_diffside (Li o2 l2) k21 k2" by (simp add:Plane_Bet_diffside)
then have P32 : "Plane_diffside (Li o2 l2) k2 k21" by (simp add:Plane_diffside_rev)
from assms have P33 : "Plane_diffside (Li o2 l2) k2 h2" by (simp add:Plane_diffside_rev)

from P23 have P34 : ”Eq (Geos (Poi o2) add Emp) (Geos (Poi h2) add Emp) ⟹ Line_on (Li o2 l2) h2” by (simp add:Point_Eq)
from P22 P34 have P35 : ”¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi h2) add Emp)” by blast
have P36 : ”Line_on (Li o2 h2) o2” by (simp add:Line_on_rule)
have P37 : ”Line_on (Li o2 h2) h2” by (simp add:Line_on_rule)
from P26 P35 P36 P37 have P38 : ”Line_on (Li o2 k2) h2
⟹ Eq (Geos (Lin (Li o2 h2)) add Emp) (Geos (Lin (Li o2 k2)) add Emp)” by (simp add:Line_unique)
from assms P38 have P39 : ”¬ Line_on (Li o2 k2) h2” by blast
from P28 have P40 : ”Eq (Geos (Poi k21) add Emp) (Geos (Poi h2) add Emp) ⟹ Line_on (Li o2 k2) h2” by (blast intro:Point_Eq)
from P39 P40 have P41 : ”¬ Eq (Geos (Poi k21) add Emp) (Geos (Poi h2) add Emp)” by blast
from P32 P33 P41 have ”Plane_sameside (Li o2 l2) k21 h2” by (blast intro:Plane_trans_inv)
then have P42 : ”Plane_sameside (Li o2 l2) h2 k21” by (simp add:Plane_sameside_rev)
from assms P1 have ”Def (Ang (An o1 l1 k1))” by (simp add:Ang_sinple_def)
then have P43 : ”Def (Ang (An k1 o1 l1))” by (blast intro:Ang_def_rev Ang_def_inv)
from assms P22 have ”Def (Ang (An o2 l2 k2))” by (simp add:Ang_sinple_def)
then have P44 : ”Def (Ang (An k2 o2 l2))” by (blast intro:Ang_def_rev Ang_def_inv)
from P8 have P45 : ”Bet_Point (Se k1 k11) o1” by (simp add:Bet_rev)
from P28 have P46 : ”Bet_Point (Se k2 k21) o2” by (simp add:Bet_rev)
from assms P43 P44 P45 P46 have P47 : ”Cong (Geos (Ang (An l1 o1 k11)) add Emp) (Geos (Ang (An l2 o2 k21)) add Emp)” by (simp add:Ang_complementary)
have P48 : ”Eq (Geos (Ang (An l1 o1 k11)) add Emp) (Geos (Ang (An k11 o1 l1)) add Emp)” by (simp add:Ang_roll)
have P49 : ”Eq (Geos (Ang (An l2 o2 k21)) add Emp) (Geos (Ang (An k21 o2 l2)) add Emp)” by (simp add:Ang_roll)
from P47 P48 have P50 : ”Cong (Geos (Ang (An k11 o1 l1)) add Emp) (Geos (Ang (An l2 o2 k21)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P49 P50 have P51 : ”Cong (Geos (Ang (An k11 o1 l1)) add Emp) (Geos (Ang (An k21 o2 l2)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
have P52 : ”Line_on (Li o1 k11) k11” by (simp add:Line_on_rule)
have P53 : ”Line_on (Li o1 k11) o1” by (simp add:Line_on_rule)
from P5 P7 P52 P53 have P54 : ”Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 k11)) add Emp)” by (blast intro:Line_unique)
from P54 have P55 : ”Eq (Geos (Lin (Li o1 h1)) add Emp) (Geos (Lin (Li o1 k11)) add Emp)
⟹ Eq (Geos (Lin (Li o1 h1)) add Emp) (Geos (Lin (Li o1 k1)) add Emp)” by (blast intro:Eq_trans Eq_rev)
from assms P55 have P56 : ”¬ Eq (Geos (Lin (Li o1 h1)) add Emp) (Geos (Lin (Li o1 k11)) add Emp)” by blast
have P57 : ”Line_on (Li o2 k21) k21” by (simp add:Line_on_rule)
have P58 : ”Line_on (Li o2 k21) o2” by (simp add:Line_on_rule)
from P26 P28 P57 P58 have P59 : ”Eq (Geos (Lin (Li o2 k2)) add Emp) (Geos (Lin (Li o2 k21)) add Emp)” by (blast intro:Line_unique)
from P59 have P60 : ”Eq (Geos (Lin (Li o2 h2)) add Emp) (Geos (Lin (Li o2 k21)) add Emp)
⟹ Eq (Geos (Lin (Li o2 h2)) add Emp) (Geos (Lin (Li o2 k2)) add Emp)” by (blast intro:Eq_trans Eq_rev)
from assms P60 have P61 : ”¬ Eq (Geos (Lin (Li o2 h2)) add Emp) (Geos (Lin (Li o2 k21)) add Emp)” by blast
from assms P21 P42 P51 P56 P61 have P62 : ”Cong (Geos (Ang (An h1 o1 k11)) add Emp) (Geos (Ang (An h2 o2 k21)) add Emp)” by (simp add:Ang_sub)
have P63 : ”Eq (Geos (Ang (An h1 o1 k11)) add Emp) (Geos (Ang (An k11 o1 h1)) add Emp)” by (simp add:Ang_roll)
have P64 : ”Eq (Geos (Ang (An h2 o2 k21)) add Emp) (Geos (Ang (An k21 o2 h2)) add Emp)” by (simp add:Ang_roll)
from P62 P63 have P65 : ”Cong (Geos (Ang (An k11 o1 h1)) add Emp) (Geos (Ang (An h2 o2 k21)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P64 P65 have P66 : ”Cong (Geos (Ang (An k11 o1 h1)) add Emp) (Geos (Ang (An k21 o2 h2)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P54 have P67 : ”Line_on (Li o1 k11) h1 ⟹ Line_on (Li o1 k1) h1” by (blast intro:Line_on_trans Eq_rev)
from P18 P67 have P68 : ”¬ Line_on (Li o1 k11) h1” by blast
from P7 P68 have ”Def (Ang (An o1 k11 h1))” by (simp add:Ang_sinple_def)
then have P69 : ”Def (Ang (An k11 o1 h1))” by (blast intro:Ang_def_rev Ang_def_inv)
from P59 have P70 : ”Line_on (Li o2 k21) h2 ⟹ Line_on (Li o2 k2) h2” by (blast intro:Line_on_trans Eq_rev)
from P39 P70 have P71 : ”¬ Line_on (Li o2 k21) h2” by blast
from P28 P71 have ”Def (Ang (An o2 k21 h2))” by (simp add:Ang_sinple_def)

```
  then have P72 : "Def (Ang (An k21 o2 h2))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P8 P29 P66 P69 P72 show "Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2
k2)) add Emp)" by (simp add:Ang_complementary)
qed

lemma (in Congruence_Rule) Ang_split_lemma1 :
  assumes N :
    "Def (Ang (An h1 o1 k1))" "Def (Ang (An h2 o2 k2))"
    "Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 k2)) add Emp)"
    "Cong (Geos (Ang (An l1 o1 k1)) add Emp) (Geos (Ang (An l2 o2 k2)) add Emp)"
    "Plane_sameside (Li o1 k1) h1 l1"
    "Plane_sameside (Li o2 k2) h2 l2"
    "¬ Eq (Geos (Lin (Li o1 l1)) add Emp) (Geos (Lin (Li o1 h1)) add Emp)"
  shows
    "¬ Eq (Geos (Lin (Li o2 l2)) add Emp) (Geos (Lin (Li o2 h2)) add Emp)"
proof
  assume W : "Eq (Geos (Lin (Li o2 l2)) add Emp) (Geos (Lin (Li o2 h2)) add Emp)"
  have P1 : "Line_on (Li o2 k2) o2" by (simp add:Line_on_rule)
  from N have P2 : "¬ Line_on (Li o2 k2) h2 ∧ ¬ Line_on (Li o2 k2) l2
    ∧ ¬ Eq (Geos (Poi h2) add Emp) (Geos (Poi l2) add Emp)" by (simp add:Plane_sameside_def)
  from P1 P2 have "Bet_Point (Se h2 l2) o2 ⟹ (∃p. Bet_Point (Se h2 l2) p
    ∧ Line_on (Li o2 k2) p ∧ ¬ Line_on (Li o2 k2) h2 ∧ ¬ Line_on (Li o2 k2) l2)" by blast
  then have "Bet_Point (Se h2 l2) o2 ⟹ Plane_diffside (Li o2 k2) h2 l2"
    by (simp add:Plane_diffside_def)
  then have P3 : "Bet_Point (Se h2 l2) o2 ⟹ ¬ Plane_sameside (Li o2 k2) h2 l2"
    by (simp add:Plane_diffside_not_sameside)
  from N P3 have P4 : "¬ Bet_Point (Se h2 l2) o2" by blast
  have P5 : "Line_on (Li o2 l2) l2" by (simp add:Line_on_rule)
  from W P5 have P6 : "Line_on (Li o2 h2) l2" by (simp add:Line_on_trans)
  have P7 : "Line_on (Li o2 k2) k2" by (simp add:Line_on_rule)
  have P8 : "¬ Bet_Point (Se k2 k2) o2" by (simp add:Bet_end_Point)
  from P1 have P9 : "Eq (Geos (Poi o2) add Emp) (Geos (Poi l2) add Emp) ⟹ Line_on (Li o2 k2) l2"
by (simp add:Point_Eq)
  from P2 P9 have P10 : "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi l2) add Emp)" by blast
  from N have "Def (Tri (Tr h2 o2 k2))" by (simp add:Ang_to_Tri)
  then have P11 : "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi k2) add Emp)" by (simp add:Tri_def)
  from N P4 P6 P7 P8 P10 P11 have P12 :
    "Eq (Geos (Ang (An h2 o2 k2)) add Emp) (Geos (Ang (An l2 o2 k2)) add Emp) ∧ Def (Ang (An
l2 o2 k2))" by (simp add:Ang_Point_swap)
  from N P12 have P13 : "Cong (Geos (Ang (An l2 o2 k2)) add Emp) (Geos (Ang (An h1 o1 k1)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from N have P14 : "Cong (Geos (Ang (An l2 o2 k2)) add Emp) (Geos (Ang (An l1 o1 k1)) add Emp)"
by (blast intro:Ang_rev)
  from N P13 P14 have P15 : "Eq (Geos (Lin (Li h1 o1)) add Emp) (Geos (Lin (Li l1 o1)) add Emp) ∧
¬ Bet_Point (Se h1 l1) o1" by (simp add:Ang_move_unique)
  from N have "Def (Tri (Tr h1 o1 k1))" by (simp add:Ang_to_Tri)
  then have "¬ Eq (Geos (Poi h1) add Emp) (Geos (Poi o1) add Emp)" by (simp add:Tri_def)
  then have P16 : "Eq (Geos (Lin (Li h1 o1)) add Emp) (Geos (Lin (Li o1 h1)) add Emp)" by (simp
add:Line_rev)
  from N have P17 : "¬ Line_on (Li o1 k1) h1 ∧ ¬ Line_on (Li o1 k1) l1
    ∧ ¬ Eq (Geos (Poi h1) add Emp) (Geos (Poi l1) add Emp)" by (simp add:Plane_sameside_def)
  have P18 : "Line_on (Li o1 k1) o1" by (simp add:Line_on_rule)
  then have P19 : "Eq (Geos (Poi o1) add Emp) (Geos (Poi l1) add Emp) ⟹ Line_on (Li o1 k1) l1" by
(simp add:Point_Eq)
  from P17 P19 have "¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi l1) add Emp)" by blast
  then have P20 : "Eq (Geos (Lin (Li o1 l1)) add Emp) (Geos (Lin (Li l1 o1)) add Emp)" by (simp
add:Line_rev)
  from P15 P16 have P21 : "Eq (Geos (Lin (Li o1 h1)) add Emp) (Geos (Lin (Li l1 o1)) add Emp)" by
(blast intro:Eq_rev Eq_trans)
  from P20 P21 have P22 : "Eq (Geos (Lin (Li o1 l1)) add Emp) (Geos (Lin (Li o1 h1)) add Emp)" by
(blast intro:Eq_rev Eq_trans)
  from N P22 show False by blast
qed

theorem (in Congruence_Rule) Ang_split :
  assumes
```

```
    "Def (Ang (An h1 o1 k1))" "Def (Ang (An h2 o2 k2))"
    "Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 k2)) add Emp)"
    "Ang_inside (An h1 o1 k1) l1"
  shows
    "∃p. Ang_inside (An h2 o2 k2) p
    ∧ Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An h2 o2 p)) add Emp)
    ∧ Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An k2 o2 p)) add Emp)"
proof -
  from assms have P1 : "Plane_sameside (Li o1 h1) k1 l1 ∧ Plane_sameside (Li o1 k1) h1 l1" by (simp
add:Ang_inside_def)
  from assms have P2 : "¬ Line_on (Li o2 k2) h2" by (simp add:Ang_to_Tri Tri_def_Line)
  from P1 have P3 : "¬ Line_on (Li o1 k1) l1" by (simp add:Plane_sameside_def)
  from assms have P4 : "Def (Tri (Tr h1 o1 k1))" by (simp add:Ang_to_Tri)
  then have P5 : "¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi k1) add Emp)" by (simp add:Tri_def)
  from P3 P5 have "Def (Ang (An o1 k1 l1))" by (simp add:Ang_sinple_def)
  then have P6 : "Def (Ang (An k1 o1 l1))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P2 P6 have "∃p. Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An p o2 k2)) add Emp)
    ∧ Plane_sameside (Li o2 k2) p h2" by (simp add:Ang_move_sameside)
  then obtain l2 :: Point where P7 : "Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An l2 o2
k2)) add Emp)
    ∧ Plane_sameside (Li o2 k2) l2 h2" by blast
  from assms have "Def (Tri (Tr h2 o2 k2))" by (simp add:Ang_to_Tri)
  then have P8 : "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi k2) add Emp)" by (simp add:Tri_def)
  have P9 : "Eq (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An l1 o1 k1)) add Emp)" by (simp
add:Ang_roll)
  from P7 P9 have P10 : "Cong (Geos (Ang (An l1 o1 k1)) add Emp) (Geos (Ang (An l2 o2 k2)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from P1 have P11 : "¬ Line_on_Seg (Li o1 h1) (Se k1 l1) ∧ ¬ Line_on (Li o1 h1) k1
    ∧ ¬ Line_on (Li o1 h1) l1 ∧ ¬ Eq (Geos (Poi k1) add Emp) (Geos (Poi l1) add Emp)" by (simp
add:Plane_sameside_def)
  have "Line_on (Li o1 l1) l1" by (simp add:Line_on_rule)
  then have P12 : "Eq (Geos (Lin (Li o1 l1)) add Emp) (Geos (Lin (Li o1 h1)) add Emp) ⟹ Line_on
(Li o1 h1) l1" by (simp add:Line_on_trans)
  from P11 P12 have P13 : "¬ Eq (Geos (Lin (Li o1 l1)) add Emp) (Geos (Lin (Li o1 h1)) add Emp)"
by blast
  from P7 have P14 : "Plane_sameside (Li o2 k2) h2 l2" by (simp add:Plane_sameside_rev)
  from assms P1 P10 P13 P14 have P15 : "¬ Eq (Geos (Lin (Li o2 l2)) add Emp) (Geos (Lin (Li o2 h2))
add Emp)" by (simp add:Ang_split_lemma1)
  from P1 have P16 :   "Plane_sameside (Li o1 k1) l1 h1" by (simp add:Plane_sameside_rev)
  from P7 have P17 : "Plane_sameside (Li o2 k2) l2 h2" by simp
  from assms P5 P8 P10 P13 P15 P16 P17 have P18 :
    "Cong (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An l2 o2 h2)) add Emp)
    ∧ Ang_inside (An h2 o2 k2) l2" by (simp add:Ang_sub_lemma1)
  have P19 : "Eq (Geos (Ang (An l2 o2 k2)) add Emp) (Geos (Ang (An k2 o2 l2)) add Emp)" by (simp
add:Ang_roll)
  from P7 P19 have P20 : "Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An k2 o2 l2)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  have P21 : "Eq (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An h1 o1 l1)) add Emp)" by (simp
add:Ang_roll)
  have P22 : "Eq (Geos (Ang (An l2 o2 h2)) add Emp) (Geos (Ang (An h2 o2 l2)) add Emp)" by (simp
add:Ang_roll)
  from P18 P21 have P23 : "Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An l2 o2 h2)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from P22 P23 have P24 : "Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An h2 o2 l2)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from P18 P20 P24 show "∃p. Ang_inside (An h2 o2 k2) p
    ∧ Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An h2 o2 p)) add Emp)
    ∧ Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An k2 o2 p)) add Emp)" by blast
qed

theorem (in Congruence_Rule) Ang_split_unique :
  assumes
    "Def (Ang (An h1 o1 k1))" "Def (Ang (An h2 o2 k2))"
    "Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 k2)) add Emp)"
    "Ang_inside (An h1 o1 k1) l1"
    "Ang_inside (An h2 o2 k2) l21"
```

```
    "Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An h2 o2 l21)) add Emp)"
    "Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An k2 o2 l21)) add Emp)"
    "Ang_inside (An h2 o2 k2) l22"
    "Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An h2 o2 l22)) add Emp)"
    "Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An k2 o2 l22)) add Emp)"
  shows
    "Eq (Geos (Lin (Li o2 l21)) add Emp) (Geos (Lin (Li o2 l22)) add Emp)"
proof -
  from assms have "Plane_sameside (Li o2 h2) k2 l21 ∧ Plane_sameside (Li o2 k2) h2 l21" by (simp
add:Ang_inside_def)
  then have P1 : "Plane_sameside (Li o2 k2) l21 h2" by (simp add:Plane_sameside_rev)
  from assms have P2 : "Plane_sameside (Li o2 h2) k2 l22 ∧ Plane_sameside (Li o2 k2) h2 l22" by
(simp add:Ang_inside_def)
  from P1 P2 have P3 : "¬ Eq (Geos (Poi l22) add Emp) (Geos (Poi l21) add Emp) ⟹
    Plane_sameside (Li o2 k2) l21 l22" by (simp add:Plane_sameside_trans)
  have P4 : "Eq (Geos (Ang (An k2 o2 l21)) add Emp) (Geos (Ang (An l21 o2 k2)) add Emp)" by (simp
add:Ang_roll)
  from assms P4 have P5 : "Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An l21 o2 k2))
add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  have P6 : "Eq (Geos (Ang (An k2 o2 l22)) add Emp) (Geos (Ang (An l22 o2 k2)) add Emp)" by (simp
add:Ang_roll)
  from assms P6 have P7 : "Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An l22 o2 k2))
add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from P3 P5 P7 have P8 : "¬ Eq (Geos (Poi l22) add Emp) (Geos (Poi l21) add Emp) ⟹
    Eq (Geos (Lin (Li l21 o2)) add Emp) (Geos (Lin (Li l22 o2)) add Emp)"
    by (simp add:Ang_move_unique)
  have P9 : "Line_on (Li o2 k2) o2" by (simp add:Line_on_rule)
  from P1 have P10 : "¬ Line_on (Li o2 k2) l21" by (simp add:Plane_sameside_def)
  from P9 have P11 : "Eq (Geos (Poi o2) add Emp) (Geos (Poi l21) add Emp) ⟹ Line_on (Li o2 k2)
l21" by (simp add:Point_Eq)
  from P10 P11 have P12 : "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi l21) add Emp)" by blast
  then have P13 : "Eq (Geos (Lin (Li o2 l21)) add Emp) (Geos (Lin (Li l21 o2)) add Emp)" by (simp
add:Line_rev)
  from P2 have P14 : "¬ Line_on (Li o2 k2) l22" by (simp add:Plane_sameside_def)
  from P9 have P15 : "Eq (Geos (Poi o2) add Emp) (Geos (Poi l22) add Emp) ⟹ Line_on (Li o2 k2)
l22" by (simp add:Point_Eq)
  from P14 P15 have "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi l22) add Emp)" by blast
  then have P16 : "Eq (Geos (Lin (Li o2 l22)) add Emp) (Geos (Lin (Li l22 o2)) add Emp)" by (simp
add:Line_rev)
  from P8 P13 have P17 : "¬ Eq (Geos (Poi l22) add Emp) (Geos (Poi l21) add Emp) ⟹
    Eq (Geos (Lin (Li o2 l21)) add Emp) (Geos (Lin (Li l22 o2)) add Emp)" by (blast intro:Eq_rev
Eq_trans)
  from P16 P17 have P18 : "¬ Eq (Geos (Poi l22) add Emp) (Geos (Poi l21) add Emp) ⟹
    Eq (Geos (Lin (Li o2 l21)) add Emp) (Geos (Lin (Li o2 l22)) add Emp)" by (blast intro:Eq_rev
Eq_trans)
  have P19 : "Line_on (Li o2 l21) o2" by (simp add:Line_on_rule)
  have P20 : "Line_on (Li o2 l21) l21" by (simp add:Line_on_rule)
  have P21 : "Line_on (Li o2 l22) o2" by (simp add:Line_on_rule)
  have "Line_on (Li o2 l22) l22" by (simp add:Line_on_rule)
  then have P22 : "Eq (Geos (Poi l22) add Emp) (Geos (Poi l21) add Emp) ⟹ Line_on (Li o2 l22)
l21" by (simp add:Point_Eq)
  from P12 P19 P20 P21 P22 have P23 : "Eq (Geos (Poi l22) add Emp) (Geos (Poi l21) add Emp) ⟹
    Eq (Geos (Lin (Li o2 l21)) add Emp) (Geos (Lin (Li o2 l22)) add Emp)" by (simp add:Line_unique)
  from P18 P23 show "Eq (Geos (Lin (Li o2 l21)) add Emp) (Geos (Lin (Li o2 l22)) add Emp)" by blast
qed

lemma (in Congruence_Rule) Tri_week_SSS_lemma1 :
  assumes
    "Plane_diffside (Li x y) z1 z2"
    "¬ Eq (Geos (Poi x) add Emp) (Geos (Poi y) add Emp)"
    "Eq (Geos (Seg (Se x z1)) add Emp) (Geos (Seg (Se x z2)) add Emp)"
    "Eq (Geos (Seg (Se y z1)) add Emp) (Geos (Seg (Se y z2)) add Emp)"
    "∃p. Bet_Point (Se z1 z2) p ∧ Line_on (Li x y) p ∧ Eq (Geos (Poi x) add Emp) (Geos (Poi p) add
Emp)"
  shows "Cong (Geos (Ang (An x z1 y)) add Emp) (Geos (Ang (An x z2 y)) add Emp)"
proof -
```

from assms have P1 : "∃p. Bet_Point (Se z1 z2) p ∧ Line_on (Li x y) p
∧ ¬ Line_on (Li x y) z1 ∧ ¬ Line_on (Li x y) z2" by (simp add:Plane_diffside_def)
from assms obtain pn :: Point where P2 : "Bet_Point (Se z1 z2) pn ∧ Line_on (Li x y) pn ∧ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp)" by blast
from P2 have P3 : "Bet_Point (Se z1 z2) pn" by simp
then have P4 : "Line_on (Li z1 z2) pn" by (simp add:Line_Bet_on)
from P2 P4 have P5 : "Line_on (Li z1 z2) x" by (blast intro:Eq_rev Point_Eq)
from assms P3 have P6 : "Bet_Point (Se z1 z2) x" by (blast intro:Eq_rev Point_Eq)
have P7 : "Line_on (Li x y) x" by (simp add:Line_on_rule)
have P8 : "Line_on (Li x y) y" by (simp add:Line_on_rule)
from assms P5 P7 P8 have P9 : "Line_on (Li z1 z2) y ⟹ Eq (Geos (Lin (Li z1 z2)) add Emp) (Geos (Lin (Li x y)) add Emp)" by (simp add:Line_unique)
have P10 : "Line_on (Li z1 z2) z1" by (simp add:Line_on_rule)
from P9 P10 have P11 : "Line_on (Li z1 z2) y ⟹ Line_on (Li x y) z1" by (simp add:Line_on_trans)
from P1 P11 have P12 : "¬ Line_on (Li z1 z2) y" by blast
from P3 have P13 : "¬ Eq (Geos (Poi z1) add Emp) (Geos (Poi z2) add Emp)" by (simp add:Bet_Point_def)
from P8 have P14 : "Eq (Geos (Poi z2) add Emp) (Geos (Poi y) add Emp) ⟹ Line_on (Li x y) z2" by (blast intro:Eq_rev Point_Eq)
from P1 P14 have P15 : "¬ Eq (Geos (Poi z2) add Emp) (Geos (Poi y) add Emp)" by blast
from P8 have P16 : "Eq (Geos (Poi y) add Emp) (Geos (Poi z1) add Emp) ⟹ Line_on (Li x y) z1" by (blast intro:Eq_rev Point_Eq)
from P1 P16 have P17 : "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi z1) add Emp)" by blast
from P12 P13 have P18 : "Def (Tri (Tr z1 z2 y))" by (simp add:Ang_sinple_def Ang_to_Tri)
then have P19 : "Def (Tri (Tr y z1 z2))" by (blast intro:Tri_def_trans)
from assms P19 have P20 : "Cong (Geos (Ang (An y z1 z2)) add Emp) (Geos (Ang (An y z2 z1)) add Emp)" by (simp add:Tri_isosceles)
from P18 have P21 : "Def (Ang (An z1 z2 y))" by (simp add:Tri_to_Ang)
from P6 have P22 : "Line_on (Li z2 z1) x" by (simp add:Line_Bet_on)
from P6 have P23 : "Inv (Bet_Point (Se z2 x) z1) ∧ Inv (Bet_Point (Se x z1) z2)" by (simp add:Bet_iff)
then have "¬ Bet_Point (Se x z1) z2" by (simp add:Inv_def)
then have P24 : "¬ Bet_Point (Se z1 x) z2" by (blast intro:Bet_rev)
have P25 : "Line_on (Li z2 y) y" by (simp add:Line_on_rule)
have P26 : "¬ Bet_Point (Se y y) z2" by (simp add:Bet_end_Point)
from P6 have P27 : "¬ Eq (Geos (Poi z2) add Emp) (Geos (Poi x) add Emp)" by (simp add:Bet_Point_def)
from P15 P21 P22 P24 P25 P26 P27 have P28 :
"Eq (Geos (Ang (An z1 z2 y)) add Emp) (Geos (Ang (An x z2 y)) add Emp) ∧ Def (Ang (An x z2 y))" by (simp add:Ang_Point_swap)
have P29 : "Eq (Geos (Ang (An y z2 z1)) add Emp) (Geos (Ang (An z1 z2 y)) add Emp)" by (simp add:Ang_roll)
from P20 P29 have P30 : "Cong (Geos (Ang (An y z1 z2)) add Emp) (Geos (Ang (An z1 z2 y)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P28 P30 have P31 : "Cong (Geos (Ang (An y z1 z2)) add Emp) (Geos (Ang (An x z2 y)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P19 have P32 : "Def (Ang (An z2 z1 y))" by (blast intro:Tri_to_Ang Tri_def_rev)
from P6 have P33 : "Line_on (Li z1 z2) x" by (simp add:Line_Bet_on)
from P23 have P34 : "¬ Bet_Point (Se z2 x) z1" by (simp add:Inv_def)
have P35 : "Line_on (Li z1 y) y" by (simp add:Line_on_rule)
have P36 : "¬ Bet_Point (Se y y) z1" by (simp add:Bet_end_Point)
from P6 have "¬ Eq (Geos (Poi x) add Emp) (Geos (Poi z1) add Emp)" by (simp add:Bet_Point_def)
then have P37 : "¬ Eq (Geos (Poi z1) add Emp) (Geos (Poi x) add Emp)" by (blast intro:Eq_rev)
from P17 have P38 : "¬ Eq (Geos (Poi z1) add Emp) (Geos (Poi y) add Emp)" by (blast intro:Eq_rev)
from P32 P33 P34 P35 P36 P37 P38 have P39 :
"Eq (Geos (Ang (An z2 z1 y)) add Emp) (Geos (Ang (An x z1 y)) add Emp) ∧ Def (Ang (An x z1 y))" by (simp add:Ang_Point_swap)
have P40 : "Eq (Geos (Ang (An y z1 z2)) add Emp) (Geos (Ang (An z2 z1 y)) add Emp)" by (simp add:Ang_roll)
from P31 P40 have P41 : "Cong (Geos (Ang (An z2 z1 y)) add Emp) (Geos (Ang (An x z2 y)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P39 P41 show "Cong (Geos (Ang (An x z1 y)) add Emp) (Geos (Ang (An x z2 y)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
qed

theorem (in Congruence_Rule) Tri_week_SSS :
assumes
"Plane_diffside (Li x y) z1 z2"
"¬ Eq (Geos (Poi x) add Emp) (Geos (Poi y) add Emp)"

```
    "Eq (Geos (Seg (Se x z1)) add Emp) (Geos (Seg (Se x z2)) add Emp)"
    "Eq (Geos (Seg (Se y z1)) add Emp) (Geos (Seg (Se y z2)) add Emp)"
  shows "Cong (Geos (Ang (An x y z1)) add Emp) (Geos (Ang (An x y z2)) add Emp)"
proof -
  from assms have "∃p. Bet_Point (Se z1 z2) p ∧ Line_on (Li x y) p
    ∧ ¬ Line_on (Li x y) z1 ∧ ¬ Line_on (Li x y) z2" by (simp add:Plane_diffside_def)
  then obtain pn :: Point where P1 : "Bet_Point (Se z1 z2) pn ∧ Line_on (Li x y) pn
    ∧ ¬ Line_on (Li x y) z1 ∧ ¬ Line_on (Li x y) z2" by blast
  have P2 : "Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ∧ Eq (Geos (Poi y) add Emp) (Geos
(Poi pn) add Emp) ⟹
    Eq (Geos (Poi x) add Emp) (Geos (Poi y) add Emp)" by (blast intro:Eq_trans)
  from assms P2 have "¬ (Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ∧ Eq (Geos (Poi y) add
Emp) (Geos (Poi pn) add Emp))" by blast
  then have P3 : "Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ∧ ¬ Eq (Geos (Poi y) add Emp)
(Geos (Poi pn) add Emp)
    ∨ ¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ∧ Eq (Geos (Poi y) add Emp) (Geos (Poi
pn) add Emp)
    ∨ ¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ∧ ¬ Eq (Geos (Poi y) add Emp) (Geos
(Poi pn) add Emp)" by blast
  from P1 have P4 : "Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
    ∃p. Bet_Point (Se z1 z2) p ∧ Line_on (Li x y) p ∧ Eq (Geos (Poi x) add Emp) (Geos (Poi p) add
Emp)" by blast
  from assms P4 have P5 : "Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
    Cong (Geos (Ang (An x z1 y)) add Emp) (Geos (Ang (An x z2 y)) add Emp)"
    by (simp add:Tri_week_SSS_lemma1)
  have P6 : "Line_on (Li x y) x" by (simp add:Line_on_rule)
  then have P7 : "Eq (Geos (Poi x) add Emp) (Geos (Poi z1) add Emp) ⟹ Line_on (Li x y) z1" by
(simp add:Point_Eq)
  from P1 P7 have P8 : "¬ Eq (Geos (Poi z1) add Emp) (Geos (Poi x) add Emp)" by (blast intro:Eq_rev)
  have P9 : "Line_on (Li x y) y" by (simp add:Line_on_rule)
  then have P10 : "Eq (Geos (Poi y) add Emp) (Geos (Poi z1) add Emp) ⟹ Line_on (Li x y) z1" by
(simp add:Point_Eq)
  from P1 P10 have P11 : "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi z1) add Emp)" by blast
  from assms P1 have "Def (Tri (Tr x y z1))" by (simp add:Ang_sinple_def Ang_to_Tri)
  then have P12 : "Def (Tri (Tr z1 x y))" by (simp add:Tri_def_trans)
  from P6 have P13 : "Eq (Geos (Poi x) add Emp) (Geos (Poi z2) add Emp) ⟹ Line_on (Li x y) z2"
by (simp add:Point_Eq)
  from P1 P13 have P14 : "¬ Eq (Geos (Poi z2) add Emp) (Geos (Poi x) add Emp)" by (blast in-
tro:Eq_rev)
  from P9 have P15 : "Eq (Geos (Poi y) add Emp) (Geos (Poi z2) add Emp) ⟹ Line_on (Li x y) z2"
by (simp add:Point_Eq)
  from P1 P15 have P16 : "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi z2) add Emp)" by (blast in-
tro:Eq_rev)
  from assms P1 have "Def (Tri (Tr x y z2))" by (simp add:Ang_sinple_def Ang_to_Tri)
  then have P17 : "Def (Tri (Tr z2 x y))" by (simp add:Tri_def_trans)
  have P18 : "Eq (Geos (Seg (Se x z1)) add Emp) (Geos (Seg (Se z1 x)) add Emp)" by (blast in-
tro:Seg_rev)
  have P19 : "Eq (Geos (Seg (Se x z2)) add Emp) (Geos (Seg (Se z2 x)) add Emp)" by (blast in-
tro:Seg_rev)
  from assms P18 have P20 : "Eq (Geos (Seg (Se z1 x)) add Emp) (Geos (Seg (Se x z2)) add Emp)" by
(blast intro:Eq_trans Eq_rev)
  from P19 P20 have P21 : "Eq (Geos (Seg (Se z1 x)) add Emp) (Geos (Seg (Se z2 x)) add Emp)" by
(blast intro:Eq_trans Eq_rev)
  have P22 : "Eq (Geos (Seg (Se y z1)) add Emp) (Geos (Seg (Se z1 y)) add Emp)" by (blast in-
tro:Seg_rev)
  have P23 : "Eq (Geos (Seg (Se y z2)) add Emp) (Geos (Seg (Se z2 y)) add Emp)" by (blast in-
tro:Seg_rev)
  from assms P22 have P24 : "Eq (Geos (Seg (Se z1 y)) add Emp) (Geos (Seg (Se y z2)) add Emp)" by
(blast intro:Eq_trans Eq_rev)
  from P23 P24 have P25 : "Eq (Geos (Seg (Se z1 y)) add Emp) (Geos (Seg (Se z2 y)) add Emp)" by
(blast intro:Eq_trans Eq_rev)
  from P5 P12 P17 P21 P25 have "Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
    Cong (Geos (Tri (Tr z1 x y)) add Emp) (Geos (Tri (Tr z2 x y)) add Emp)" by (simp add:Tri_SAS)
  then have P26 : "Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
    Cong (Geos (Ang (An z1 y x)) add Emp) (Geos (Ang (An z2 y x)) add Emp)"
    by (simp add:Tri_Cong_def)
```

have P27 : ”Eq (Geos (Ang (An z1 y x)) add Emp) (Geos (Ang (An x y z1)) add Emp)” by (simp add:Ang_roll)
have P28 : ”Eq (Geos (Ang (An z2 y x)) add Emp) (Geos (Ang (An x y z2)) add Emp)” by (simp add:Ang_roll)
from P26 P27 have P29 : ”Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
Cong (Geos (Ang (An x y z1)) add Emp) (Geos (Ang (An z2 y x)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P28 P29 have P30 : ”Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
Cong (Geos (Ang (An x y z1)) add Emp) (Geos (Ang (An x y z2)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from assms have P31 : ”Eq (Geos (Lin (Li x y)) add Emp) (Geos (Lin (Li y x)) add Emp)” by (simp add:Line_rev)
from assms P31 have P32 : ”Plane_diffside (Li y x) z1 z2” by (simp add:Plane_Line_diff_trans)
from assms have P33 : ”¬ Eq (Geos (Poi y) add Emp) (Geos (Poi x) add Emp)” by (blast intro:Eq_rev)
from P1 P31 have P34 : ”Line_on (Li y x) pn” by (blast intro:Line_on_trans)
from P1 P34 have P35 : ”Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
∃p. Bet_Point (Se z1 z2) p ∧ Line_on (Li y x) p ∧ Eq (Geos (Poi y) add Emp) (Geos (Poi p) add Emp)” by blast
from assms P32 P33 P35 have P36 : ”Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Cong (Geos (Ang (An y z1 x)) add Emp) (Geos (Ang (An y z2 x)) add Emp)”
by (simp add:Tri_week_SSS_lemma1)
from P12 have P37 : ”Def (Tri (Tr z1 y x))” by (blast intro:Tri_def_trans Tri_def_rev)
from P17 have P38 : ”Def (Tri (Tr z2 y x))” by (blast intro:Tri_def_trans Tri_def_rev)
from P21 P25 P36 P37 P38 have ”Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Cong (Geos (Tri (Tr z1 y x)) add Emp) (Geos (Tri (Tr z2 y x)) add Emp)” by (simp add:Tri_SAS)
then have P39 : ”Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Cong (Geos (Ang (An x y z1)) add Emp) (Geos (Ang (An x y z2)) add Emp)”
by (simp add:Tri_Cong_def)
from P1 have P40 : ”Bet_Point (Se z1 z2) pn” by simp
then have P41 : ”Line_on (Li z1 z2) pn” by (simp add:Line_Bet_on)
have ”Line_on (Li z1 z2) z1” by (simp add:Line_on_rule)
then have P42 : ”Eq (Geos (Lin (Li z1 z2)) add Emp) (Geos (Lin (Li x y)) add Emp) ⟹ Line_on (Li x y) z1” by (simp add:Line_on_trans)
from P1 P42 have P43 : ”¬ Eq (Geos (Lin (Li z1 z2)) add Emp) (Geos (Lin (Li x y)) add Emp)” by blast
from P1 P6 P41 have P44 : ”¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹ Line_on (Li z1 z2) x ⟹
Eq (Geos (Lin (Li z1 z2)) add Emp) (Geos (Lin (Li x y)) add Emp)” by (simp add:Line_unique)
from P43 P44 have P45 : ”¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
¬ Line_on (Li z1 z2) x” by blast
from P40 have P46 : ”¬ Eq (Geos (Poi z1) add Emp) (Geos (Poi z2) add Emp)”
by (simp add:Bet_Point_def)
from P45 P46 have ”¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
Def (Tri (Tr z1 z2 x))” by (simp add:Ang_sinple_def Ang_to_Tri)
then have P47 : ”¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
Def (Tri (Tr x z1 z2))” by (simp add:Tri_def_trans)
from assms P47 have P48 : ”¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
Cong (Geos (Ang (An x z1 z2)) add Emp) (Geos (Ang (An x z2 z1)) add Emp)”
by (simp add:Tri_isosceles)
have P49 : ”Line_on (Li z1 x) x” by (simp add:Line_on_rule)
have P50 : ”¬ Bet_Point (Se x x) z1” by (simp add:Bet_end_Point)
from P40 have P51 : ”Inv (Bet_Point (Se z2 pn) z1) ∧ Inv (Bet_Point (Se pn z1) z2)” by (simp add:Bet_iff)
then have P52 : ”¬ Bet_Point (Se z2 pn) z1” by (simp add:Inv_def)
from P40 have ”¬ Eq (Geos (Poi pn) add Emp) (Geos (Poi z1) add Emp)” by (simp add:Bet_Point_def)
then have P53 : ”¬ Eq (Geos (Poi z1) add Emp) (Geos (Poi pn) add Emp)” by (blast intro:Eq_rev)
from P47 have P54 : ”¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
Def (Ang (An x z1 z2))” by (simp add:Tri_to_Ang)
from P8 P41 P49 P50 P52 P53 P54 have P55 : ”¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
Eq (Geos (Ang (An x z1 z2)) add Emp) (Geos (Ang (An x z1 pn)) add Emp) ∧ Def (Ang (An x z1 pn))” by (simp add:Ang_Point_swap)
from P48 P55 have P56 : ”¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
Cong (Geos (Ang (An x z1 pn)) add Emp) (Geos (Ang (An x z2 z1)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P47 have P57 : ”¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹

Def (Ang (An x z2 z1))" by (blast intro:Tri_def_rev Tri_def_trans Tri_to_Ang)
have P58 : "Line_on (Li z2 x) x" by (simp add:Line_on_rule)
have P59 : "¬ Bet_Point (Se x x) z2" by (simp add:Bet_end_Point)
from P40 have P60 : "Line_on (Li z2 z1) pn" by (simp add:Line_Bet_on)
from P51 have "¬ Bet_Point (Se pn z1) z2" by (simp add:Inv_def)
then have P61 : "¬ Bet_Point (Se z1 pn) z2" by (blast intro:Bet_rev)
from P40 have P62 : "¬ Eq (Geos (Poi z2) add Emp) (Geos (Poi pn) add Emp)"
by (simp add:Bet_Point_def)
from P14 P57 P58 P59 P60 P61 P62 have P63 : "¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
Eq (Geos (Ang (An x z2 z1)) add Emp) (Geos (Ang (An x z2 pn)) add Emp) ∧ Def (Ang (An x z2 pn))" by (simp add:Ang_Point_swap)
from P56 P63 have P64 : "¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
Cong (Geos (Ang (An x z1 pn)) add Emp) (Geos (Ang (An x z2 pn)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P1 P9 P41 have P65 : "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹ Line_on (Li z1 z2) y ⟹
Eq (Geos (Lin (Li z1 z2)) add Emp) (Geos (Lin (Li x y)) add Emp)" by (simp add:Line_unique)
from P43 P65 have P66 : "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
¬ Line_on (Li z1 z2) y" by blast
from P46 P66 have "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Def (Tri (Tr z1 z2 y))" by (simp add:Ang_sinple_def Ang_to_Tri)
then have P67 : "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Def (Tri (Tr y z1 z2))" by (simp add:Tri_def_trans)
from assms P67 have P68 : "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Cong (Geos (Ang (An y z1 z2)) add Emp) (Geos (Ang (An y z2 z1)) add Emp)"
by (simp add:Tri_isosceles)
from P67 have P69 : "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Def (Ang (An y z1 z2))" by (simp add:Tri_to_Ang)
have P70 : "Line_on (Li z1 y) y" by (simp add:Line_on_rule)
have P71 : "¬ Bet_Point (Se y y) z1" by (simp add:Bet_end_Point)
from P11 have P72 : "¬ Eq (Geos (Poi z1) add Emp) (Geos (Poi y) add Emp)" by (blast intro:Eq_rev)
from P41 P52 P53 P69 P70 P71 P72 have P73 : "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Eq (Geos (Ang (An y z1 z2)) add Emp) (Geos (Ang (An y z1 pn)) add Emp) ∧ Def (Ang (An y z1 pn))" by (simp add:Ang_Point_swap)
from P68 P73 have P74 : "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Cong (Geos (Ang (An y z1 pn)) add Emp) (Geos (Ang (An y z2 z1)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P69 have P75 : "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Def (Ang (An y z2 z1))" by (simp add:Ang_def_inv)
have P76 : "Line_on (Li z2 y) y" by (simp add:Line_on_rule)
have P77 : "¬ Bet_Point (Se y y) z2" by (simp add:Bet_end_Point)
from P16 have P78 : "¬ Eq (Geos (Poi z2) add Emp) (Geos (Poi y) add Emp)" by (blast intro:Eq_rev)
from P60 P61 P62 P75 P76 P77 P78 have P79 : "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Eq (Geos (Ang (An y z2 z1)) add Emp) (Geos (Ang (An y z2 pn)) add Emp) ∧ Def (Ang (An y z2 pn))" by (simp add:Ang_Point_swap)
from P74 P79 have P80 : "¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Cong (Geos (Ang (An y z1 pn)) add Emp) (Geos (Ang (An y z2 pn)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
have P81 : "¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ⟹
¬ Eq (Geos (Poi pn) add Emp) (Geos (Poi x) add Emp)" by (blast intro:Eq_rev)
from assms P1 P6 P9 P81 have
"¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ∧ ¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Bet_Point (Se x pn) y ∨ Bet_Point (Se pn y) x ∨ Bet_Point (Se y x) pn" by (simp add:Bet_case)
then have P82 :
"¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ∧ ¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹
Bet_Point (Se x pn) y ∧ ¬ Bet_Point (Se pn y) x ∧ ¬ Bet_Point (Se y x) pn
∨ ¬ Bet_Point (Se x pn) y ∧ Bet_Point (Se pn y) x ∧ ¬ Bet_Point (Se y x) pn
∨ ¬ Bet_Point (Se x pn) y ∧ ¬ Bet_Point (Se pn y) x ∧ Bet_Point (Se y x) pn"
by (simp add:Bet_case_fact)
have P83 : "Bet_Point (Se x pn) y ⟹ Bet_Point (Se pn x) y" by (simp add:Bet_rev)
have P84 : "Line_on (Li z1 pn) pn" by (simp add:Line_on_rule)

have P85 : ”Line_on (Li pn x) x” by (simp add:Line_on_rule)
from P83 have P86 : ”Bet_Point (Se x pn) y ⟹ Line_on (Li pn x) y” by (simp add:Line_Bet_on)
from assms P6 P9 P85 P86 have P87 : ”Bet_Point (Se x pn) y ⟹
Eq (Geos (Lin (Li pn x)) add Emp) (Geos (Lin (Li x y)) add Emp)” by (simp add:Line_unique)
have P88 : ”Line_on (Li z1 pn) z1” by (simp add:Line_on_rule)
from P87 P88 have P89 : ”Bet_Point (Se x pn) y ⟹
Eq (Geos (Lin (Li pn x)) add Emp) (Geos (Lin (Li z1 pn)) add Emp) ⟹ Line_on (Li x y) z1” by
(blast intro:Line_on_trans Eq_rev)
from P1 P89 have P90 : ”Bet_Point (Se x pn) y ⟹
¬ Eq (Geos (Lin (Li pn x)) add Emp) (Geos (Lin (Li z1 pn)) add Emp)” by blast
from P83 P84 P90 have P91 : ”Bet_Point (Se x pn) y ⟹ Plane_sameside (Li z1 pn) y x” by (simp
add:Plane_Bet_sameside)
have P92 : ”Line_on (Li z2 pn) pn” by (simp add:Line_on_rule)
have P93 : ”Line_on (Li z2 pn) z2” by (simp add:Line_on_rule)
from P87 P93 have P94 : ”Bet_Point (Se x pn) y ⟹
Eq (Geos (Lin (Li pn x)) add Emp) (Geos (Lin (Li z2 pn)) add Emp) ⟹ Line_on (Li x y) z2” by
(blast intro:Line_on_trans Eq_rev)
from P1 P94 have P95 : ”Bet_Point (Se x pn) y ⟹
¬ Eq (Geos (Lin (Li pn x)) add Emp) (Geos (Lin (Li z2 pn)) add Emp)” by blast
from P83 P92 P95 have P96 : ”Bet_Point (Se x pn) y ⟹ Plane_sameside (Li z2 pn) y x” by (simp
add:Plane_Bet_sameside)
from P37 have ”Def (Ang (An y z1 x))” by (blast intro:Tri_to_Ang Ang_def_rev Ang_def_inv)
then have P97 : ”¬ Eq (Geos (Lin (Li z1 y)) add Emp) (Geos (Lin (Li z1 x)) add Emp)” by (simp
add:Ang_def)
from P38 have ”Def (Ang (An y z2 x))” by (blast intro:Tri_to_Ang Ang_def_rev Ang_def_inv)
then have P98 : ”¬ Eq (Geos (Lin (Li z2 y)) add Emp) (Geos (Lin (Li z2 x)) add Emp)” by (simp
add:Ang_def)
from P53 P62 P64 P80 P91 P96 P97 P98 have P99 : ”¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn)
add Emp)
∧ ¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹ Bet_Point (Se x pn) y ⟹
Cong (Geos (Ang (An y z1 x)) add Emp) (Geos (Ang (An y z2 x)) add Emp)” by (simp add:Ang_sub)
have P100 : ”Line_on (Li pn y) y” by (simp add:Line_on_rule)
from P85 have P101 : ”Bet_Point (Se pn y) x ⟹ Line_on (Li pn y) x” by (simp add:Line_Bet_on)
from assms P6 P9 P100 P101 have P102 : ”Bet_Point (Se pn y) x ⟹
Eq (Geos (Lin (Li pn y)) add Emp) (Geos (Lin (Li x y)) add Emp)” by (simp add:Line_unique)
from P88 P102 have P103 : ”Bet_Point (Se pn y) x ⟹
Eq (Geos (Lin (Li pn y)) add Emp) (Geos (Lin (Li z1 pn)) add Emp) ⟹ Line_on (Li x y) z1” by
(blast intro:Line_on_trans Eq_rev)
from P1 P103 have P104 : ”Bet_Point (Se pn y) x ⟹
¬ Eq (Geos (Lin (Li pn y)) add Emp) (Geos (Lin (Li z1 pn)) add Emp)” by blast
from P84 P104 have ”Bet_Point (Se pn y) x ⟹ Plane_sameside (Li z1 pn) x y”
by (simp add:Plane_Bet_sameside)
then have P105 : ”Bet_Point (Se pn y) x ⟹ Plane_sameside (Li z1 pn) y x”
by (simp add:Plane_sameside_rev)
from P93 P102 have P106 : ”Bet_Point (Se pn y) x ⟹
Eq (Geos (Lin (Li pn y)) add Emp) (Geos (Lin (Li z2 pn)) add Emp) ⟹ Line_on (Li x y) z2” by
(blast intro:Line_on_trans Eq_rev)
from P1 P106 have P107 : ”Bet_Point (Se pn y) x ⟹
¬ Eq (Geos (Lin (Li pn y)) add Emp) (Geos (Lin (Li z2 pn)) add Emp)” by blast
from P92 P107 have ”Bet_Point (Se pn y) x ⟹ Plane_sameside (Li z2 pn) x y”
by (simp add:Plane_Bet_sameside)
then have P108 : ”Bet_Point (Se pn y) x ⟹ Plane_sameside (Li z2 pn) y x”
by (simp add:Plane_sameside_rev)
from P53 P62 P64 P80 P97 P98 P105 P108 have P109 : ”¬ Eq (Geos (Poi x) add Emp) (Geos (Poi
pn) add Emp)
∧ ¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹ Bet_Point (Se pn y) x ⟹
Cong (Geos (Ang (An y z1 x)) add Emp) (Geos (Ang (An y z2 x)) add Emp)” by (simp add:Ang_sub)
have P110 : ”Bet_Point (Se y x) pn ⟹ Bet_Point (Se x y) pn” by (simp add:Bet_rev)
from P88 have P111 : ”Eq (Geos (Lin (Li x y)) add Emp) (Geos (Lin (Li z1 pn)) add Emp)
⟹ Line_on (Li x y) z1” by (blast intro:Line_on_trans Eq_rev)
from P1 P111 have P112 : ”¬ Eq (Geos (Lin (Li x y)) add Emp) (Geos (Lin (Li z1 pn)) add Emp)” by
blast
from P84 P110 P112 have ”Bet_Point (Se y x) pn ⟹ Plane_diffside (Li z1 pn) x y” by (simp
add:Plane_Bet_diffside)
then have P113 : ”Bet_Point (Se y x) pn ⟹ Plane_diffside (Li z1 pn) y x”
by (simp add:Plane_diffside_rev)

```
  from P93 have P114 : "Eq (Geos (Lin (Li x y)) add Emp) (Geos (Lin (Li z2 pn)) add Emp)
    ⟹ Line_on (Li x y) z2" by (blast intro:Line_on_trans Eq_rev)
  from P1 P114 have P115 : "¬ Eq (Geos (Lin (Li x y)) add Emp) (Geos (Lin (Li z2 pn)) add Emp)" by
blast
  from P92 P110 P115 have "Bet_Point (Se y x) pn ⟹ Plane_diffside (Li z2 pn) x y" by (simp
add:Plane_Bet_diffside)
  then have P116 : "Bet_Point (Se y x) pn ⟹ Plane_diffside (Li z2 pn) y x"
    by (simp add:Plane_diffside_rev)
  from P53 P62 P64 P80 P97 P98 P113 P116 have P117 : "¬ Eq (Geos (Poi x) add Emp) (Geos (Poi
pn) add Emp)
    ∧ ¬ Eq (Geos (Poi y) add Emp) (Geos (Poi pn) add Emp) ⟹ Bet_Point (Se y x) pn ⟹
    Cong (Geos (Ang (An y z1 x)) add Emp) (Geos (Ang (An y z2 x)) add Emp)" by (simp add:Ang_add)
  from P82 P99 P109 P117 have P118 :
    "¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ∧ ¬ Eq (Geos (Poi y) add Emp) (Geos
(Poi pn) add Emp) ⟹
    Cong (Geos (Ang (An y z1 x)) add Emp) (Geos (Ang (An y z2 x)) add Emp)" by blast
  from P21 P25 P37 P38 P118 have
    "¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ∧ ¬ Eq (Geos (Poi y) add Emp) (Geos
(Poi pn) add Emp) ⟹
    Cong (Geos (Tri (Tr z1 y x)) add Emp) (Geos (Tri (Tr z2 y x)) add Emp)" by (simp add:Tri_SAS)
  then have P119 :
    "¬ Eq (Geos (Poi x) add Emp) (Geos (Poi pn) add Emp) ∧ ¬ Eq (Geos (Poi y) add Emp) (Geos
(Poi pn) add Emp) ⟹
    Cong (Geos (Ang (An x y z1)) add Emp) (Geos (Ang (An x y z2)) add Emp)"
    by (simp add:Tri_Cong_def)
  from P3 P30 P39 P119 show "Cong (Geos (Ang (An x y z1)) add Emp) (Geos (Ang (An x y z2)) add
Emp)" by blast
qed

theorem (in Congruence_Rule) Tri_SSS :
  assumes
    "Def (Tri (Tr A1 B1 C1))" "Def (Tri (Tr A2 B2 C2))"
    "Eq (Geos (Seg (Se A1 B1)) add Emp) (Geos (Seg (Se A2 B2)) add Emp)"
    "Eq (Geos (Seg (Se B1 C1)) add Emp) (Geos (Seg (Se B2 C2)) add Emp)"
    "Eq (Geos (Seg (Se C1 A1)) add Emp) (Geos (Seg (Se C2 A2)) add Emp)"
  shows "Cong (Geos (Tri (Tr A1 B1 C1)) add Emp) (Geos (Tri (Tr A2 B2 C2)) add Emp)"
proof -
  from assms have "Def (Tri (Tr C2 B2 A2))" by (simp add:Tri_def_rev)
  then have P1 : "¬ Line_on (Li A2 C2) B2" by (simp add:Tri_def_Line)
  from assms have P2 : "Def (Ang (An B1 A1 C1))" by (blast intro:Tri_def_rev Tri_def_trans Tri_to_Ang)
  from P1 P2 have "∃p. Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An p A2 C2)) add
Emp) ∧ Plane_sameside (Li A2 C2) p B2" by (simp add:Ang_move_sameside)
  then obtain B21 :: Point where P3 : "Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An
B21 A2 C2)) add Emp)
    ∧ Plane_sameside (Li A2 C2) B21 B2" by blast
  then have P4 : "¬ Line_on (Li A2 C2) B21" by (simp add:Plane_sameside_def)
  from P2 P4 have "∃p. Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An p A2 C2)) add
Emp) ∧ Plane_diffside (Li A2 C2) p B21" by (simp add:Ang_move_diffside)
  then obtain B22 :: Point where P5 : "Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An
B22 A2 C2)) add Emp)
    ∧ Plane_diffside (Li A2 C2) B22 B21" by blast
  have P6 : "Line_on (Li A2 B21) A2" by (simp add:Line_on_rule)
  have P7 : "Line_on (Li A2 B21) B21" by (simp add:Line_on_rule)
  have P8 : "Line_on (Li A2 C2) A2" by (simp add:Line_on_rule)
  then have P9 : "Eq (Geos (Poi A2) add Emp) (Geos (Poi B21) add Emp) ⟹ Line_on (Li A2 C2)
B21" by (simp add:Point_Eq)
  from P4 P9 have P10 : "¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi B21) add Emp)" by blast
  from assms have P11 : "¬ Eq (Geos (Poi A1) add Emp) (Geos (Poi B1) add Emp)" by (simp
add:Tri_def)
  from P6 P7 P10 P11 have "∃p. Eq (Geos (Seg (Se A1 B1)) add Emp) (Geos (Seg (Se A2 p)) add Emp)
    ∧ ¬ Bet_Point (Se p B21) A2 ∧ Line_on (Li A2 B21) p ∧ ¬ Eq (Geos (Poi A2) add Emp) (Geos
(Poi p) add Emp)" by (simp add:Seg_move_sameside)
  then obtain B211 :: Point where P12 : "Eq (Geos (Seg (Se A1 B1)) add Emp) (Geos (Seg (Se A2
B211)) add Emp)
    ∧ ¬ Bet_Point (Se B211 B21) A2 ∧ Line_on (Li A2 B21) B211 ∧ ¬ Eq (Geos (Poi A2) add Emp)
(Geos (Poi B211) add Emp)" by blast
```

have P13 : "Line_on (Li A2 B22) A2" by (simp add:Line_on_rule)
have P14 : "Line_on (Li A2 B22) B22" by (simp add:Line_on_rule)
from P8 have P15 : "Eq (Geos (Poi A2) add Emp) (Geos (Poi B22) add Emp) ⟹ Line_on (Li A2 C2) B22" by (simp add:Point_Eq)
from P5 have P16 : "∃p. Bet_Point (Se B22 B21) p ∧ Line_on (Li A2 C2) p
∧ ¬ Line_on (Li A2 C2) B22 ∧ ¬ Line_on (Li A2 C2) B21" by (simp add:Plane_diffside_def)
from P15 P16 have P17 : "¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi B22) add Emp)" by blast
from P11 P13 P14 P17 have "∃p. Eq (Geos (Seg (Se A1 B1)) add Emp) (Geos (Seg (Se A2 p)) add Emp)
∧ ¬ Bet_Point (Se p B22) A2 ∧ Line_on (Li A2 B22) p ∧ ¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi p) add Emp)" by (simp add:Seg_move_sameside)
then obtain B221 :: Point where P18 : "Eq (Geos (Seg (Se A1 B1)) add Emp) (Geos (Seg (Se A2 B221)) add Emp)
∧ ¬ Bet_Point (Se B221 B22) A2 ∧ Line_on (Li A2 B22) B221 ∧ ¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi B221) add Emp)" by blast
from assms have P19 : "¬ Eq (Geos (Poi C2) add Emp) (Geos (Poi A2) add Emp)" by (simp add:Tri_def)
then have P20 : "¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi C2) add Emp)" by (blast intro:Eq_rev)
from P4 P20 have P21 : "Def (Ang (An B21 A2 C2))" by (blast intro:Ang_sinple_def Ang_def_rev Ang_def_inv)
from P12 have "¬ Bet_Point (Se B211 B21) A2" by blast
then have P22 : "¬ Bet_Point (Se B21 B211) A2" by (blast intro:Bet_rev)
have P23 : "Line_on (Li A2 C2) C2" by (simp add:Line_on_rule)
have P24 : "¬ Bet_Point (Se C2 C2) A2" by (simp add:Bet_end_Point)
from P12 P20 P21 P22 P23 P24 have P25 :
"Eq (Geos (Ang (An B21 A2 C2)) add Emp) (Geos (Ang (An B211 A2 C2)) add Emp) ∧ Def (Ang (An B211 A2 C2))" by (simp add:Ang_Point_swap)
from P3 P25 have P26 : "Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An B211 A2 C2)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from assms have P27 : "Def (Tri (Tr A1 B1 C1))" by (simp add:Ang_to_Tri)
from P25 have P28 : "Def (Tri (Tr A2 B211 C2))" by (blast intro:Ang_to_Tri Tri_def_trans Tri_def_rev)
from assms have P29 : "Eq (Geos (Seg (Se A1 C1)) add Emp) (Geos (Seg (Se A2 C2)) add Emp)" by (blast intro:Seg_rev Eq_rev Eq_trans)
from P12 P26 P27 P28 P29 have "Cong (Geos (Tri (Tr A1 B1 C1)) add Emp) (Geos (Tri (Tr A2 B211 C2)) add Emp)" by (simp add:Tri_SAS)
then have P30 : "Eq (Geos (Seg (Se B1 C1)) add Emp) (Geos (Seg (Se B211 C2)) add Emp)" by (simp add:Tri_Cong_def)
from P16 P20 have P31 : "Def (Ang (An B22 A2 C2))" by (blast intro:Ang_sinple_def Ang_def_rev Ang_def_inv)
from P18 have "¬ Bet_Point (Se B221 B22) A2" by blast
then have P32 : "¬ Bet_Point (Se B22 B221) A2" by (blast intro:Bet_rev)
from P18 P20 P23 P24 P31 P32 have P33 :
"Eq (Geos (Ang (An B22 A2 C2)) add Emp) (Geos (Ang (An B221 A2 C2)) add Emp) ∧ Def (Ang (An B221 A2 C2))" by (simp add:Ang_Point_swap)
from P5 P33 have P34 : "Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An B221 A2 C2)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P33 have P35 : "Def (Tri (Tr A2 B221 C2))" by (blast intro:Ang_to_Tri Tri_def_trans Tri_def_rev)
from P18 P27 P29 P34 P35 have "Cong (Geos (Tri (Tr A1 B1 C1)) add Emp) (Geos (Tri (Tr A2 B221 C2)) add Emp)" by (simp add:Tri_SAS)
then have P36 : "Eq (Geos (Seg (Se B1 C1)) add Emp) (Geos (Seg (Se B221 C2)) add Emp)" by (simp add:Tri_Cong_def)
from P12 have P37 : "Eq (Geos (Seg (Se A1 B1)) add Emp) (Geos (Seg (Se A2 B211)) add Emp)" by simp
from P18 have P38 : "Eq (Geos (Seg (Se A1 B1)) add Emp) (Geos (Seg (Se A2 B221)) add Emp)" by simp
from P37 P38 have P39 : "Eq (Geos (Seg (Se A2 B221)) add Emp) (Geos (Seg (Se A2 B211)) add Emp)" by (blast intro:Eq_trans Eq_rev)
from assms P38 have P40 : "Eq (Geos (Seg (Se A2 B221)) add Emp) (Geos (Seg (Se A2 B2)) add Emp)" by (blast intro:Eq_trans Eq_rev)
from P30 P36 have P41 : "Eq (Geos (Seg (Se B221 C2)) add Emp) (Geos (Seg (Se B211 C2)) add Emp)" by (blast intro:Eq_trans Eq_rev)
from assms P36 have P42 : "Eq (Geos (Seg (Se B221 C2)) add Emp) (Geos (Seg (Se B2 C2)) add Emp)" by (blast intro:Eq_trans Eq_rev)
from P5 have P43 : "Plane_diffside (Li A2 C2) B22 B21" by simp
then have P44 : "Eq (Geos (Poi B21) add Emp) (Geos (Poi B211) add Emp) ⟹
Plane_diffside (Li A2 C2) B22 B211" by (blast intro:Point_Eq Eq_rev)

from P6 P8 P20 P23 have P45 : "Line_on (Li A2 B21) C2 ⟹
Eq (Geos (Lin (Li A2 B21)) add Emp) (Geos (Lin (Li A2 C2)) add Emp)" by (simp add:Line_unique)
from P7 P45 have P46 : "Line_on (Li A2 B21) C2 ⟹ Line_on (Li A2 C2) B21"
by (simp add:Line_on_trans)
from P4 P46 have P47 : "¬ Line_on (Li A2 B21) C2" by blast
from P6 P7 P10 P12 P22 P47 have P48 : "Plane_sameside (Li C2 A2) B21 B211 ∨ Eq (Geos (Poi B21) add Emp) (Geos (Poi B211) add Emp)" by (simp add:Seg_Plane_sameside)
from P20 have P49 : "Plane_sameside (Li C2 A2) B21 B211 ⟹
Plane_sameside (Li A2 C2) B21 B211" by (blast intro:Line_rev Plane_Line_trans Eq_rev)
from P43 have P50 : "Plane_diffside (Li A2 C2) B21 B22" by (simp add:Plane_diffside_rev)
from P49 P50 have P51 : "Plane_sameside (Li C2 A2) B21 B211 ⟹
Plane_diffside (Li A2 C2) B22 B211" by (simp add:Plane_trans Plane_diffside_rev)
from P44 P48 P51 have P52 : "Plane_diffside (Li A2 C2) B22 B211" by blast
then have "Plane_diffside (Li A2 C2) B211 B22" by (blast intro:Plane_diffside_rev)
then have P53 : "Eq (Geos (Poi B22) add Emp) (Geos (Poi B221) add Emp) ⟹
Plane_diffside (Li A2 C2) B211 B221" by (blast intro:Point_Eq Eq_rev)
from P8 P13 P20 P23 have P54 : "Line_on (Li A2 B22) C2 ⟹
Eq (Geos (Lin (Li A2 B22)) add Emp) (Geos (Lin (Li A2 C2)) add Emp)" by (simp add:Line_unique)
from P14 P54 have P55 : "Line_on (Li A2 B22) C2 ⟹ Line_on (Li A2 C2) B22"
by (simp add:Line_on_trans)
from P16 P55 have P56 : "¬ Line_on (Li A2 B22) C2" by blast
from P31 have "¬ Eq (Geos (Poi B22) add Emp) (Geos (Poi A2) add Emp)" by (simp add:Ang_def)
then have P57 : "¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi B22) add Emp)" by (blast intro:Eq_rev)
from P13 P14 P18 P32 P56 P57 have P58 : "Plane_sameside (Li C2 A2) B22 B221 ∨ Eq (Geos (Poi B22) add Emp) (Geos (Poi B221) add Emp)" by (simp add:Seg_Plane_sameside)
from P20 have P59 : "Plane_sameside (Li C2 A2) B22 B221 ⟹ Plane_sameside (Li A2 C2) B22 B221" by (blast intro:Line_rev Plane_Line_trans Eq_rev)
from P52 P59 have P60 : "Plane_sameside (Li C2 A2) B22 B221 ⟹
Plane_diffside (Li A2 C2) B211 B221" by (simp add:Plane_trans Plane_diffside_rev)
from P53 P58 P60 have "Plane_diffside (Li A2 C2) B211 B221" by blast
then have P61 : "Plane_diffside (Li A2 C2) B221 B211" by (simp add:Plane_diffside_rev)
from P20 P61 have P62 : "Plane_diffside (Li C2 A2) B221 B211"
by (blast intro:Line_rev Plane_Line_diff_trans)
have P63 : "Eq (Geos (Seg (Se B221 C2)) add Emp) (Geos (Seg (Se C2 B221)) add Emp)" by (simp add:Seg_rev)
have P64 : "Eq (Geos (Seg (Se B211 C2)) add Emp) (Geos (Seg (Se C2 B211)) add Emp)" by (simp add:Seg_rev)
from P41 P63 P64 have P65 : "Eq (Geos (Seg (Se C2 B221)) add Emp) (Geos (Seg (Se C2 B211)) add Emp)" by (blast intro:Eq_rev Eq_trans)
from P19 P39 P62 P65 have P66 : "Cong (Geos (Ang (An C2 A2 B221)) add Emp) (Geos (Ang (An C2 A2 B211)) add Emp)" by (simp add:Tri_week_SSS)
have P67 : "Eq (Geos (Ang (An C2 A2 B211)) add Emp) (Geos (Ang (An B211 A2 C2)) add Emp)" by (simp add:Ang_roll)
from P66 P67 have P68 : "Cong (Geos (Ang (An C2 A2 B221)) add Emp) (Geos (Ang (An B211 A2 C2)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P3 have P69 : "Plane_sameside (Li A2 C2) B21 B2" by simp
from P50 P69 have P70 : "Plane_diffside (Li A2 C2) B2 B22" by (simp add:Plane_trans)
then have P71 : "Plane_diffside (Li A2 C2) B22 B2" by (simp add:Plane_diffside_rev)
from P70 have P72 : "Eq (Geos (Poi B22) add Emp) (Geos (Poi B221) add Emp) ⟹
Plane_diffside (Li A2 C2) B221 B2" by (blast intro:Point_Eq Plane_diffside_rev)
from P59 P71 have P73 : "Plane_sameside (Li C2 A2) B22 B221 ⟹
Plane_diffside (Li A2 C2) B221 B2" by (simp add:Plane_trans)
from P58 P72 P73 have P74 : "Plane_diffside (Li A2 C2) B221 B2" by blast
from P20 P74 have P75 : "Plane_diffside (Li C2 A2) B221 B2"
by (blast intro:Line_rev Plane_Line_diff_trans)
have P76 : "Eq (Geos (Seg (Se B221 C2)) add Emp) (Geos (Seg (Se C2 B221)) add Emp)" by (simp add:Seg_rev)
have P77 : "Eq (Geos (Seg (Se B2 C2)) add Emp) (Geos (Seg (Se C2 B2)) add Emp)" by (simp add:Seg_rev)
from P42 P76 P77 have P78 : "Eq (Geos (Seg (Se C2 B221)) add Emp) (Geos (Seg (Se C2 B2)) add Emp)" by (blast intro:Eq_rev Eq_trans)
from P19 P40 P75 P78 have P79 : "Cong (Geos (Ang (An C2 A2 B221)) add Emp) (Geos (Ang (An C2 A2 B2)) add Emp)" by (simp add:Tri_week_SSS)
have P80 : "Eq (Geos (Ang (An C2 A2 B2)) add Emp) (Geos (Ang (An B2 A2 C2)) add Emp)" by (simp add:Ang_roll)
from P79 P80 have P81 : "Cong (Geos (Ang (An C2 A2 B221)) add Emp) (Geos (Ang (An B2 A2

```
C2)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from P52 P71 have P82 : "¬ Eq (Geos (Poi B2) add Emp) (Geos (Poi B211) add Emp) ⟹
    Plane_sameside (Li A2 C2) B2 B211" by (blast intro:Plane_trans_inv)
  from P68 P81 P82 have P83 : "¬ Eq (Geos (Poi B2) add Emp) (Geos (Poi B211) add Emp) ⟹
    Eq (Geos (Lin (Li B2 A2)) add Emp) (Geos (Lin (Li B211 A2)) add Emp) ∧ ¬ Bet_Point (Se B2
B211) A2" by (simp add:Ang_move_unique)
  from assms have P84 : "Def (Ang (An B2 A2 C2))" by (blast intro:Tri_to_Ang Ang_def_rev Ang_def_inv)
  then have "¬ Eq (Geos (Poi B2) add Emp) (Geos (Poi A2) add Emp)" by (simp add:Ang_def)
  then have P85 : "Eq (Geos (Lin (Li B2 A2)) add Emp) (Geos (Lin (Li A2 B2)) add Emp)" by (simp
add:Line_rev)
  have P86 : "Line_on (Li B211 A2) B211" by (simp add:Line_on_rule)
  from P83 P85 P86 have P87 : "¬ Eq (Geos (Poi B2) add Emp) (Geos (Poi B211) add Emp) ⟹
    Line_on (Li A2 B2) B211" by (blast intro:Eq_rev Line_on_trans)
  have "Line_on (Li A2 B2) B2" by (simp add:Line_on_rule)
  then have P88 : "Eq (Geos (Poi B2) add Emp) (Geos (Poi B211) add Emp) ⟹ Line_on (Li A2 B2)
B211" by (simp add:Point_Eq)
  from P87 P88 have P89 : "Line_on (Li A2 B2) B211" by blast
  have P90 : "¬ Bet_Point (Se B2 B2) A2" by (simp add:Bet_end_Point)
  have P91 : "Eq (Geos (Poi B2) add Emp) (Geos (Poi B211) add Emp) ⟹
    Bet_Point (Se B211 B2) A2 ⟹ Bet_Point (Se B2 B2) A2" by (blast intro:Eq_rev Bet_Point_Eq)
  from P90 P91 have P92 : "Eq (Geos (Poi B2) add Emp) (Geos (Poi B211) add Emp) ⟹ ¬ Bet_Point
(Se B2 B211) A2" by (blast intro:Bet_rev)
  from P83 P92 have P93 : "¬ Bet_Point (Se B2 B211) A2" by blast
  from P12 P20 P23 P24 P84 P89 P93 have P94 :
    "Eq (Geos (Ang (An B2 A2 C2)) add Emp) (Geos (Ang (An B211 A2 C2)) add Emp) ∧ Def (Ang
(An B211 A2 C2))" by (simp add:Ang_Point_swap)
  from P25 P94 have P95 : "Eq (Geos (Ang (An B21 A2 C2)) add Emp) (Geos (Ang (An B2 A2 C2))
add Emp)" by (blast intro:Eq_rev Eq_trans)
  from P3 P95 have P96 : "Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An B2 A2 C2))
add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  have P97 : "Eq (Geos (Seg (Se C1 A1)) add Emp) (Geos (Seg (Se A1 C1)) add Emp)" by (simp
add:Seg_rev)
  have P98 : "Eq (Geos (Seg (Se C2 A2)) add Emp) (Geos (Seg (Se A2 C2)) add Emp)" by (simp
add:Seg_rev)
  from assms P97 P98 have P99 : "Eq (Geos (Seg (Se A1 C1)) add Emp) (Geos (Seg (Se A2 C2)) add
Emp)" by (blast intro:Eq_rev Eq_trans)
  from assms have P100 : "Def (Tri (Tr A1 B1 C1))" by (simp add:Ang_to_Tri)
  from assms have P101 : "Def (Tri (Tr A2 B2 C2))" by (simp add:Ang_to_Tri)
  from assms P96 P99 P100 P101 show "Cong (Geos (Tri (Tr A1 B1 C1)) add Emp) (Geos (Tri (Tr A2
B2 C2)) add Emp)" by (simp add:Tri_SAS)
qed

theorem (in Congruence_Rule) Ang_trans :
  assumes
    "Def (Ang (An A1 B1 C1))" "Def (Ang (An A2 B2 C2))" "Def (Ang (An A3 B3 C3))"
    "Cong (Geos (Ang (An A2 B2 C2)) add Emp) (Geos (Ang (An A1 B1 C1)) add Emp)"
    "Cong (Geos (Ang (An A3 B3 C3)) add Emp) (Geos (Ang (An A1 B1 C1)) add Emp)"
  shows "Cong (Geos (Ang (An A2 B2 C2)) add Emp) (Geos (Ang (An A3 B3 C3)) add Emp)"
proof -
  from assms have P1 : "Cong (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An A2 B2 C2)) add
Emp)" by (simp add:Ang_rev)
  from assms P1 have "∃p q. Cong (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An p B2 q))
add Emp)
    ∧ Eq (Geos (Ang (An A2 B2 C2)) add Emp) (Geos (Ang (An p B2 q)) add Emp)
    ∧ Eq (Geos (Seg (Se B1 A1)) add Emp) (Geos (Seg (Se B2 p)) add Emp)
    ∧ Line_on (Li B2 A2) p ∧ ¬ Bet_Point (Se p A2) B2
    ∧ Eq (Geos (Seg (Se B1 C1)) add Emp) (Geos (Seg (Se B2 q)) add Emp)
    ∧ Line_on (Li B2 C2) q ∧ ¬ Bet_Point (Se q C2) B2 ∧ Def (Ang (An p B2 q))" by (simp
add:Ang_replace)
  then obtain A21 C21 :: Point where P2 : "Cong (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang
(An A21 B2 C21)) add Emp)
    ∧ Eq (Geos (Ang (An A2 B2 C2)) add Emp) (Geos (Ang (An A21 B2 C21)) add Emp)
    ∧ Eq (Geos (Seg (Se B1 A1)) add Emp) (Geos (Seg (Se B2 A21)) add Emp)
    ∧ Line_on (Li B2 A2) A21 ∧ ¬ Bet_Point (Se A21 A2) B2
    ∧ Eq (Geos (Seg (Se B1 C1)) add Emp) (Geos (Seg (Se B2 C21)) add Emp)
    ∧ Line_on (Li B2 C2) C21 ∧ ¬ Bet_Point (Se C21 C2) B2 ∧ Def (Ang (An A21 B2 C21))" by blast
```

from assms have P3 : ”Cong (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An A3 B3 C3)) add Emp)” by (simp add:Ang_rev)
from assms P3 have ”∃p q. Cong (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An p B3 q)) add Emp)
∧ Eq (Geos (Ang (An A3 B3 C3)) add Emp) (Geos (Ang (An p B3 q)) add Emp)
∧ Eq (Geos (Seg (Se B1 A1)) add Emp) (Geos (Seg (Se B3 p)) add Emp)
∧ Line_on (Li B3 A3) p ∧ ¬ Bet_Point (Se p A3) B3
∧ Eq (Geos (Seg (Se B1 C1)) add Emp) (Geos (Seg (Se B3 q)) add Emp)
∧ Line_on (Li B3 C3) q ∧ ¬ Bet_Point (Se q C3) B3 ∧ Def (Ang (An p B3 q))” by (simp add:Ang_replace)
then obtain A31 C31 :: Point where P4 : ”Cong (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An A31 B3 C31)) add Emp)
∧ Eq (Geos (Ang (An A3 B3 C3)) add Emp) (Geos (Ang (An A31 B3 C31)) add Emp)
∧ Eq (Geos (Seg (Se B1 A1)) add Emp) (Geos (Seg (Se B3 A31)) add Emp)
∧ Line_on (Li B3 A3) A31 ∧ ¬ Bet_Point (Se A31 A3) B3
∧ Eq (Geos (Seg (Se B1 C1)) add Emp) (Geos (Seg (Se B3 C31)) add Emp)
∧ Line_on (Li B3 C3) C31 ∧ ¬ Bet_Point (Se C31 C3) B3 ∧ Def (Ang (An A31 B3 C31))” by blast
from assms have P5 : ”Def (Tri (Tr B1 A1 C1))” by (blast intro:Tri_def_rev Tri_def_trans Ang_to_Tri)
from P2 have P6 : ”Def (Tri (Tr B2 A21 C21))” by (blast intro:Tri_def_rev Tri_def_trans Ang_to_Tri)
from P2 P5 P6 have ”Cong (Geos (Tri (Tr B1 A1 C1)) add Emp) (Geos (Tri (Tr B2 A21 C21)) add Emp)” by (simp add:Tri_SAS)
then have P7 : ”Eq (Geos (Seg (Se A1 C1)) add Emp) (Geos (Seg (Se A21 C21)) add Emp)” by (simp add:Tri_Cong_def)
from P4 have P8 : ”Def (Tri (Tr B3 A31 C31))” by (blast intro:Tri_def_rev Tri_def_trans Ang_to_Tri)
from P4 P5 P8 have ”Cong (Geos (Tri (Tr B1 A1 C1)) add Emp) (Geos (Tri (Tr B3 A31 C31)) add Emp)” by (simp add:Tri_SAS)
then have P9 : ”Eq (Geos (Seg (Se A1 C1)) add Emp) (Geos (Seg (Se A31 C31)) add Emp)” by (simp add:Tri_Cong_def)
from P6 have P10 : ”Def (Tri (Tr A21 C21 B2))” by (blast intro:Tri_def_trans)
from P8 have P11 : ”Def (Tri (Tr A31 C31 B3))” by (blast intro:Tri_def_trans)
from P7 P9 have P12 : ”Eq (Geos (Seg (Se A21 C21)) add Emp) (Geos (Seg (Se A31 C31)) add Emp)” by (blast intro:Eq_trans)
from P2 P4 have P13 : ”Eq (Geos (Seg (Se B2 A21)) add Emp) (Geos (Seg (Se B3 A31)) add Emp)” by (blast intro:Eq_trans)
from P2 P4 have P14 : ”Eq (Geos (Seg (Se B2 C21)) add Emp) (Geos (Seg (Se B3 C31)) add Emp)” by (blast intro:Eq_trans)
have P15 : ”Eq (Geos (Seg (Se B2 C21)) add Emp) (Geos (Seg (Se C21 B2)) add Emp)” by (simp add:Seg_rev)
have P16 : ”Eq (Geos (Seg (Se B3 C31)) add Emp) (Geos (Seg (Se C31 B3)) add Emp)” by (simp add:Seg_rev)
from P14 P15 P16 have P17 : ”Eq (Geos (Seg (Se C21 B2)) add Emp) (Geos (Seg (Se C31 B3)) add Emp)” by (blast intro:Eq_trans Eq_rev)
from P10 P11 P12 P13 P17 have ”Cong (Geos (Tri (Tr A21 C21 B2)) add Emp) (Geos (Tri (Tr A31 C31 B3)) add Emp)” by (simp add:Tri_SSS)
then have P18 : ”Cong (Geos (Ang (An A21 B2 C21)) add Emp) (Geos (Ang (An A31 B3 C31)) add Emp)” by (simp add:Tri_Cong_def)
from P2 P18 have P19 : ”Cong (Geos (Ang (An A2 B2 C2)) add Emp) (Geos (Ang (An A31 B3 C31)) add Emp)” by (blast intro:Ang_weektrans Ang_rev)
from P4 P19 show ”Cong (Geos (Ang (An A2 B2 C2)) add Emp) (Geos (Ang (An A3 B3 C3)) add Emp)” by (blast intro:Ang_weektrans Ang_rev)
qed

lemma (in Congruence_Rule) Ang_move_unique_inv :
assumes
”Def (Ang (An p1 p2 p3))” ”Def (Ang (An p4 p2 p3))”
”Plane_sameside (Li p2 p3) p1 p4”
”Eq (Geos (Lin (Li p2 p1)) add Emp) (Geos (Lin (Li p2 p4)) add Emp)”
shows
”Cong (Geos (Ang (An p1 p2 p3)) add Emp) (Geos (Ang (An p4 p2 p3)) add Emp)”
proof -
have P1 : ”Line_on (Li p2 p4) p4” by (simp add:Line_on_rule)
from assms P1 have P2 : ”Line_on (Li p2 p1) p4” by (blast intro:Line_on_trans Eq_rev)
have P3 : ”Line_on (Li p2 p3) p2” by (simp add:Line_on_rule)
from assms have P4 : ”¬ Line_on (Li p2 p3) p1” by (simp add:Plane_sameside_def)
from assms have P5 : ”¬ Line_on (Li p2 p3) p4” by (simp add:Plane_sameside_def)
from P3 P4 P5 have

```
    "Bet_Point (Se p1 p4) p2 ⟹ ∃p. Bet_Point (Se p1 p4) p ∧ Line_on (Li p2 p3) p
    ∧ ¬ Line_on (Li p2 p3) p1 ∧ ¬ Line_on (Li p2 p3) p4" by blast
  then have "Bet_Point (Se p1 p4) p2 ⟹ Plane_diffside (Li p2 p3) p1 p4"
    by (simp add:Plane_diffside_def)
  then have P6 : "Bet_Point (Se p1 p4) p2 ⟹ ¬ Plane_sameside (Li p2 p3) p1 p4"
    by (simp add:Plane_diffside_not_sameside)
  from assms P6 have P7 : "¬ Bet_Point (Se p1 p4) p2" by blast
  have P8 : "Line_on (Li p2 p3) p3" by (simp add:Line_on_rule)
  have P9 : "¬ Bet_Point (Se p3 p3) p2" by (simp add:Bet_end_Point)
  from assms have "¬ Eq (Geos (Poi p4) add Emp) (Geos (Poi p2) add Emp)" by (simp add:Ang_def)
  then have P10 : "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p4) add Emp)" by (blast intro:Eq_rev)
  from assms have P11 : "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp)"
    by (simp add:Ang_def)
  from assms P2 P7 P8 P9 P10 P11 have "Eq (Geos (Ang (An p1 p2 p3)) add Emp) (Geos (Ang (An p4
p2 p3)) add Emp)
    ∧ Def (Ang (An p4 p2 p3))" by (simp add:Ang_Point_swap)
  thus "Cong (Geos (Ang (An p1 p2 p3)) add Emp) (Geos (Ang (An p4 p2 p3)) add Emp)" by (blast
intro:Ang_weektrans)
qed

theorem (in Congruence_Rule) Ang_move_Greater :
  assumes
    "Def (Ang (An h1 o1 k1))" "Def (Ang (An h2 o2 l2))"
    "Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 k2)) add Emp)"
    "Plane_sameside (Li o2 h2) k2 l2"
    "Cong (Geos (Ang (An h2 o2 l2)) add Emp) (Geos (Ang (An h1 o1 l1)) add Emp)"
    "Plane_sameside (Li o1 h1) k1 l1"
    "Ang_inside (An h2 o2 l2) k2"
  shows
    "¬ Ang_inside (An h1 o1 k1) l1"
    "¬ Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp)"
proof -
  from assms have P1 : "¬ Line_on (Li o2 h2) k2" by (simp add:Plane_sameside_def)
  from assms have "¬ Eq (Geos (Poi h2) add Emp) (Geos (Poi o2) add Emp)" by (simp add:Ang_def)
  then have P2 : "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi h2) add Emp)" by (blast intro:Eq_rev)
  from P1 P2 have "Def (Ang (An o2 h2 k2))" by (simp add:Ang_sinple_def)
  then have P3 : "Def (Ang (An h2 o2 k2))" by (blast intro:Ang_def_rev Ang_def_inv)
  from assms P3 have "Ang_inside (An h1 o1 k1) l1 ⟹ ∃p. Ang_inside (An h2 o2 k2) p
    ∧ Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An h2 o2 p)) add Emp)
    ∧ Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An k2 o2 p)) add Emp)" by (simp
add:Ang_split)
  then obtain l21 :: Point where P4 : "Ang_inside (An h1 o1 k1) l1 ⟹ Ang_inside (An h2 o2 k2) l21
    ∧ Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An h2 o2 l21)) add Emp)
    ∧ Cong (Geos (Ang (An k1 o1 l1)) add Emp) (Geos (Ang (An k2 o2 l21)) add Emp)" by blast
  then have P5 : "Ang_inside (An h1 o1 k1) l1 ⟹
    Plane_sameside (Li o2 k2) h2 l21 ∧ Plane_sameside (Li o2 h2) k2 l21" by (simp add:Ang_inside_def)
  from assms have P6 : "Plane_diffside (Li o2 k2) h2 l2" by (simp add:Ang_inside_Planeside)
  from P5 P6 have "Ang_inside (An h1 o1 k1) l1 ⟹
    Plane_diffside (Li o2 k2) l21 l2" by (blast intro:Plane_trans)
  then have "Ang_inside (An h1 o1 k1) l1 ⟹ ∃p. Bet_Point (Se l21 l2) p ∧ Line_on (Li o2 k2) p
    ∧ ¬ Line_on (Li o2 k2) l21 ∧ ¬ Line_on (Li o2 k2) l2" by (simp add:Plane_diffside_def)
  then obtain pn :: Point where P7 : "Ang_inside (An h1 o1 k1) l1 ⟹
    Bet_Point (Se l21 l2) pn ∧ Line_on (Li o2 k2) pn
    ∧ ¬ Line_on (Li o2 k2) l21 ∧ ¬ Line_on (Li o2 k2) l2" by blast
  then have P8 : "Ang_inside (An h1 o1 k1) l1 ⟹ Bet_Point (Se l21 l2) pn" by simp
  then have P9 : "Ang_inside (An h1 o1 k1) l1 ⟹
    ¬ Eq (Geos (Poi l21) add Emp) (Geos (Poi l2) add Emp)" by (simp add:Bet_Point_def)
  from assms have P10 : "Plane_sameside (Li o2 h2) l2 k2" by (simp add:Plane_sameside_rev)
  from P5 P9 P10 have P11 : "Ang_inside (An h1 o1 k1) l1 ⟹ Plane_sameside (Li o2 h2) l2 l21" by
(blast intro:Plane_sameside_trans)
  from P8 have P12 : "Ang_inside (An h1 o1 k1) l1 ⟹
    Eq (Geos (Poi pn) add Emp) (Geos (Poi o2) add Emp) ⟹ Bet_Point (Se l21 l2) o2" by (simp
add:Point_Eq)
  have P13 : "Line_on (Li o2 h2) o2" by (simp add:Line_on_rule)
  have P14 : "Line_on (Li l21 l2) l2" by (simp add:Line_on_rule)
  from P14 have P15 : "Eq (Geos (Lin (Li l21 l2)) add Emp) (Geos (Lin (Li o2 h2)) add Emp) ⟹
```

```
    Line_on (Li o2 h2) l2" by (simp add:Line_on_trans)
  from assms have "Def (Tri (Tr o2 h2 l2))" by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
  then have P16 : "¬ Line_on (Li o2 h2) l2" by (simp add:Tri_def_Line)
  from P15 P16 have P17 : "¬ Eq (Geos (Lin (Li l21 l2)) add Emp) (Geos (Lin (Li o2 h2)) add Emp)"
by blast
  from P12 P13 P17 have "Ang_inside (An h1 o1 k1) l1 ⟹
    Eq (Geos (Poi pn) add Emp) (Geos (Poi o2) add Emp) ⟹
    Plane_diffside (Li o2 h2) l2 l21" by (simp add:Plane_Bet_diffside Plane_diffside_rev)
  then have P18 : "Ang_inside (An h1 o1 k1) l1 ⟹
    Eq (Geos (Poi pn) add Emp) (Geos (Poi o2) add Emp) ⟹
    ¬ Plane_sameside (Li o2 h2) l2 l21" by (simp add:Plane_diffside_not_sameside)
  from P11 P18 have P19 : "Ang_inside (An h1 o1 k1) l1 ⟹
    ¬ Eq (Geos (Poi pn) add Emp) (Geos (Poi o2) add Emp)" by blast
  have P20 : "Line_on (Li o2 k2) o2" by (simp add:Line_on_rule)
  from P8 have P21 : "Ang_inside (An h1 o1 k1) l1 ⟹ Line_on (Li l21 l2) pn" by (simp add:Line_Bet_on)
  from P7 P19 P20 P21 have P22 : "Ang_inside (An h1 o1 k1) l1 ⟹ Line_on (Li l21 l2) o2 ⟹
    Eq (Geos (Lin (Li l21 l2)) add Emp) (Geos (Lin (Li o2 k2)) add Emp)" by (simp add:Line_unique)
  from P14 P22 have P23 : "Ang_inside (An h1 o1 k1) l1 ⟹
    Line_on (Li l21 l2) o2 ⟹ Line_on (Li o2 k2) l2" by (simp add:Line_on_trans)
  from P7 P23 have P24 : "Ang_inside (An h1 o1 k1) l1 ⟹ ¬ Line_on (Li l21 l2) o2" by blast
  from P9 P24 have "Ang_inside (An h1 o1 k1) l1 ⟹ Def (Ang (An l21 l2 o2))"
    by (simp add:Ang_sinple_def)
  then have P25 : "Ang_inside (An h1 o1 k1) l1 ⟹ Def (Ang (An l21 o2 l2))" by (blast intro:Ang_def_rev
Ang_def_inv)
  then have P26 : "Ang_inside (An h1 o1 k1) l1 ⟹
    ¬ Eq (Geos (Lin (Li o2 l21)) add Emp) (Geos (Lin (Li o2 l2)) add Emp)" by (simp add:Ang_def)
  have P27 : "Eq (Geos (Ang (An h2 o2 l21)) add Emp) (Geos (Ang (An l21 o2 h2)) add Emp)" by
(simp add:Ang_roll)
  from P4 P27 have P28 : "Ang_inside (An h1 o1 k1) l1 ⟹
    Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An l21 o2 h2)) add Emp)" by (blast in-
tro:Ang_weektrans Ang_rev Eq_rev)
  have P29 : "Eq (Geos (Ang (An h2 o2 l2)) add Emp) (Geos (Ang (An l2 o2 h2)) add Emp)" by (simp
add:Ang_roll)
  from assms P29 have P30 : "Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An l2 o2 h2))
add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from P11 P28 P30 have P31 : "Ang_inside (An h1 o1 k1) l1 ⟹
    Eq (Geos (Lin (Li l2 o2)) add Emp) (Geos (Lin (Li l21 o2)) add Emp)" by (simp add:Ang_move_unique)
  from P25 have P32 : "Ang_inside (An h1 o1 k1) l1 ⟹ Def (Tri (Tr l21 o2 l2))" by (simp add:Ang_to_Tri)
  then have "Ang_inside (An h1 o1 k1) l1 ⟹ ¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi l2) add Emp)"
by (simp add:Tri_def)
  then have P33 : "Ang_inside (An h1 o1 k1) l1 ⟹
    Eq (Geos (Lin (Li o2 l2)) add Emp) (Geos (Lin (Li l2 o2)) add Emp)" by (simp add:Line_rev)
  from P32 have "Ang_inside (An h1 o1 k1) l1 ⟹
    ¬ Eq (Geos (Poi l21) add Emp) (Geos (Poi o2) add Emp)" by (simp add:Tri_def)
  then have P34 : "Ang_inside (An h1 o1 k1) l1 ⟹
    Eq (Geos (Lin (Li l21 o2)) add Emp) (Geos (Lin (Li o2 l21)) add Emp)" by (simp add:Line_rev)
  from P31 P33 have P35 : "Ang_inside (An h1 o1 k1) l1 ⟹
    Eq (Geos (Lin (Li o2 l2)) add Emp) (Geos (Lin (Li l21 o2)) add Emp)" by (blast intro:Eq_rev
Eq_trans)
  from P34 P35 have P36 : "Ang_inside (An h1 o1 k1) l1 ⟹
    Eq (Geos (Lin (Li o2 l21)) add Emp) (Geos (Lin (Li o2 l2)) add Emp)" by (blast intro:Eq_rev
Eq_trans)
  from P26 P36 show "¬ Ang_inside (An h1 o1 k1) l1" by blast
  from assms have P37 : "Def (Ang (An k1 o1 h1))" by (blast intro:Ang_def_rev)
  from assms have P38 : "¬ Line_on (Li o1 h1) l1" by (simp add:Plane_sameside_def)
  from P37 have P39 : "¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi h1) add Emp)" by (simp add:Ang_def)
  from P38 P39 have "Def (Ang (An o1 h1 l1))" by (simp add:Ang_sinple_def)
  then have P40 : "Def (Ang (An l1 o1 h1))" by (blast intro:Ang_def_rev Ang_def_inv)
  from assms P37 P40 have P41 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp)
⟹
    Cong (Geos (Ang (An k1 o1 h1)) add Emp) (Geos (Ang (An l1 o1 h1)) add Emp)" by (simp
add:Ang_move_unique_inv)
  have P42 : "Cong (Geos (Ang (An k1 o1 h1)) add Emp) (Geos (Ang (An h1 o1 k1)) add Emp)" by
(simp add:Ang_roll)
  from assms P37 P40 P41 P42 have P43 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1))
add Emp) ⟹
```

```
    Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An l1 o1 h1)) add Emp)" by (blast in-
tro:Ang_trans Ang_rev)
  have P44 : "Cong (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An h1 o1 l1)) add Emp)" by
(simp add:Ang_roll)
  from P40 have P45 : "Def (Ang (An h1 o1 l1))" by (blast intro:Ang_def_rev)
  from assms P40 P43 P44 P45 have P46 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1))
add Emp) ⟹
    Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h1 o1 l1)) add Emp)" by (blast in-
tro:Ang_trans Ang_rev)
  from assms P45 P46 have P47 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp)
⟹
    Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 l2)) add Emp)" by (blast in-
tro:Ang_trans Ang_rev)
  from assms P3 P47 have P48 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp)
⟹
    Cong (Geos (Ang (An h2 o2 k2)) add Emp) (Geos (Ang (An h2 o2 l2)) add Emp)" by (blast in-
tro:Ang_trans Ang_rev)
  from P29 P48 have P49 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹
    Cong (Geos (Ang (An h2 o2 k2)) add Emp) (Geos (Ang (An l2 o2 h2)) add Emp)" by (blast in-
tro:Ang_weektrans Ang_rev Eq_rev)
  have P50 : "Cong (Geos (Ang (An h2 o2 k2)) add Emp) (Geos (Ang (An k2 o2 h2)) add Emp)" by
(simp add:Ang_roll)
  from assms P49 P50 have P51 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp)
⟹
    Eq (Geos (Lin (Li k2 o2)) add Emp) (Geos (Lin (Li l2 o2)) add Emp)
    ∧ ¬ Bet_Point (Se k2 l2) o2" by (simp add:Ang_move_unique)
  from assms have "Def (Ang (An h2 o2 l2)) ∧ Plane_sameside (Li o2 h2) l2 k2
    ∧ Plane_sameside (Li o2 l2) h2 k2" by (simp add:Ang_inside_def)
  then have P52 : "¬ Line_on (Li o2 l2) k2" by (simp add:Plane_sameside_def)
  from assms have "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi l2) add Emp)" by (simp add:Ang_def)
  then have P53 : "Eq (Geos (Lin (Li o2 l2)) add Emp) (Geos (Lin (Li l2 o2)) add Emp)" by (simp
add:Line_rev)
  from P51 P53 have P54 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹
    Eq (Geos (Lin (Li k2 o2)) add Emp) (Geos (Lin (Li o2 l2)) add Emp)" by (blast intro:Eq_trans)
  have P55 : "Line_on (Li k2 o2) k2" by (simp add:Line_on_rule)
  from P54 P55 have P56 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹
    Line_on (Li o2 l2) k2" by (simp add:Line_on_trans)
  from P52 P56 show "¬ Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp)" by blast
qed

theorem (in Congruence_Rule) Ang_move_Smaller :
  assumes
    "Def (Ang (An h1 o1 k1))" "Def (Ang (An h2 o2 l2))"
    "Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 k2)) add Emp)"
    "Plane_sameside (Li o2 h2) k2 l2"
    "Cong (Geos (Ang (An h2 o2 l2)) add Emp) (Geos (Ang (An h1 o1 l1)) add Emp)"
    "Plane_sameside (Li o1 h1) k1 l1"
    "¬ Ang_inside (An h2 o2 l2) k2"
    "¬ Eq (Geos (Lin (Li o2 k2)) add Emp) (Geos (Lin (Li o2 l2)) add Emp)"
  shows "Ang_inside (An h1 o1 k1) l1"
proof -
  have P1 : "Ang_inside (An l2 o2 h2) k2 ⟹ Ang_inside (An h2 o2 l2) k2" by (simp add:Ang_inside_def
Ang_def_rev)
  from assms P1 have P2 : "¬ Ang_inside (An l2 o2 h2) k2" by blast
  from assms have P3 : "¬ Line_on (Li o2 h2) k2" by (simp add:Plane_sameside_def)
  from assms have "¬ Eq (Geos (Poi h2) add Emp) (Geos (Poi o2) add Emp)" by (simp add:Ang_def)
  then have P4 : "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi h2) add Emp)" by (blast intro:Eq_rev)
  from P3 P4 have "Def (Ang (An o2 h2 k2))" by (simp add:Ang_sinple_def)
  then have P5 : "Def (Ang (An k2 o2 h2))" by (blast intro:Ang_def_rev Ang_def_inv)
  from assms have P6 : "Def (Ang (An l2 o2 h2))" by (blast intro:Ang_def_rev)
  from assms P5 P6 have P7 : "Ang_inside (An k2 o2 h2) l2 ∧ ¬ Ang_inside (An l2 o2 h2) k2
    ∨ ¬ Ang_inside (An k2 o2 h2) l2 ∧ Ang_inside (An l2 o2 h2) k2" by (simp add:Ang_inside_case)
  from P2 P7 have "Ang_inside (An k2 o2 h2) l2" by blast
  then have P8 : "Ang_inside (An h2 o2 k2) l2" by (simp add:Ang_inside_def Ang_def_rev)
  from assms have P9 : "¬ Line_on (Li o1 h1) l1" by (simp add:Plane_sameside_def)
  from assms have "¬ Eq (Geos (Poi h1) add Emp) (Geos (Poi o1) add Emp)" by (simp add:Ang_def)
```

then have P10 : "¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi h1) add Emp)" by (blast intro:Eq_rev)
from P9 P10 have "Def (Ang (An o1 h1 l1))" by (simp add:Ang_sinple_def)
then have P11 : "Def (Ang (An h1 o1 l1))" by (blast intro:Ang_def_rev Ang_def_inv)
from P5 have P12 : "Def (Ang (An h2 o2 k2))" by (blast intro:Ang_def_rev Ang_def_inv)
from assms have P13 : "Cong (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An h2 o2 l2)) add Emp)" by (blast intro:Ang_rev)
from assms have P14 : "Plane_sameside (Li o2 h2) l2 k2" by (simp add:Plane_sameside_rev)
from assms have P15 : "Cong (Geos (Ang (An h2 o2 k2)) add Emp) (Geos (Ang (An h1 o1 k1)) add Emp)" by (blast intro:Ang_rev)
from assms have P16 : "Plane_sameside (Li o1 h1) l1 k1" by (simp add:Plane_sameside_rev)
from P8 P11 P12 P13 P14 P15 P16 have P17 : "¬ Ang_inside (An h1 o1 l1) k1"
by (simp add:Ang_move_Greater)
have P18 : "Ang_inside (An l1 o1 h1) k1 ⟹ Ang_inside (An h1 o1 l1) k1" by (simp add:Ang_inside_def Ang_def_rev)
from P17 P18 have P19 : "¬ Ang_inside (An l1 o1 h1) k1" by blast
from assms have P20 : "Def (Ang (An k1 o1 h1))" by (blast intro:Ang_def_rev)
from P11 have P21 : "Def (Ang (An l1 o1 h1))" by (blast intro:Ang_def_rev)
have "Line_on (Li o1 k1) k1" by (simp add:Line_on_rule)
then have P22 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹
Line_on (Li o1 l1) k1" by (simp add:Line_on_trans)
have P23 : "Line_on (Li o1 h1) o1" by (simp add:Line_on_rule)
have "Line_on (Li l1 k1) l1" by (simp add:Line_on_rule)
then have P24 : "Eq (Geos (Lin (Li l1 k1)) add Emp) (Geos (Lin (Li o1 h1)) add Emp) ⟹ Line_on (Li o1 h1) l1" by (simp add:Line_on_trans)
from P9 P24 have P25 : "¬ Eq (Geos (Lin (Li l1 k1)) add Emp) (Geos (Lin (Li o1 h1)) add Emp)" by blast
from P23 P25 have "Bet_Point (Se l1 k1) o1 ⟹ Plane_diffside (Li o1 h1) l1 k1"
by (simp add:Plane_Bet_diffside)
then have P26 : "Bet_Point (Se l1 k1) o1 ⟹ ¬ Plane_sameside (Li o1 h1) l1 k1"
by (simp add:Plane_diffside_not_sameside)
from P16 P26 have P27 : "¬ Bet_Point (Se l1 k1) o1" by blast
have P28 : "Line_on (Li o1 h1) h1" by (simp add:Line_on_rule)
have P29 : "¬ Bet_Point (Se h1 h1) o1" by (simp add:Bet_end_Point)
from assms have P30 : "¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi k1) add Emp)"
by (simp add:Ang_def)
from P10 P21 P22 P27 P28 P29 P30 have P31 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹
Eq (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An k1 o1 h1)) add Emp) ∧ Def (Ang (An k1 o1 h1))" by (simp add:Ang_Point_swap)
have P32 : "Eq (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An h1 o1 l1)) add Emp)" by (simp add:Ang_roll)
from P31 P32 have P33 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹
Eq (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An k1 o1 h1)) add Emp)" by (blast intro:Eq_trans Eq_rev)
have P34 : "Eq (Geos (Ang (An k1 o1 h1)) add Emp) (Geos (Ang (An h1 o1 k1)) add Emp)" by (simp add:Ang_roll)
from P33 P34 have P35 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹
Eq (Geos (Ang (An h1 o1 l1)) add Emp) (Geos (Ang (An h1 o1 k1)) add Emp)" by (blast intro:Eq_trans Eq_rev)
from assms P35 have P36 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹
Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An h2 o2 l2)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
have P37 : "Eq (Geos (Ang (An h2 o2 k2)) add Emp) (Geos (Ang (An k2 o2 h2)) add Emp)" by (simp add:Ang_roll)
from assms P37 have P38 : "Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An k2 o2 h2)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
have P39 : "Eq (Geos (Ang (An h2 o2 l2)) add Emp) (Geos (Ang (An l2 o2 h2)) add Emp)" by (simp add:Ang_roll)
from P36 P39 have P40 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹
Cong (Geos (Ang (An h1 o1 k1)) add Emp) (Geos (Ang (An l2 o2 h2)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from assms P38 P40 have P41 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹
Eq (Geos (Lin (Li k2 o2)) add Emp) (Geos (Lin (Li l2 o2)) add Emp) ∧ ¬ Bet_Point (Se k2 l2) o2" by (simp add:Ang_move_unique)

```
  from P12 have "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi k2) add Emp)" by (simp add:Ang_def)
  then have P42 : "Eq (Geos (Lin (Li o2 k2)) add Emp) (Geos (Lin (Li k2 o2)) add Emp)" by (simp
add:Line_rev)
  from P41 P42 have P43 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹
    Eq (Geos (Lin (Li o2 k2)) add Emp) (Geos (Lin (Li l2 o2)) add Emp)" by (blast intro:Eq_rev
Eq_trans)
  from assms have "¬ Eq (Geos (Poi o2) add Emp) (Geos (Poi l2) add Emp)" by (simp add:Ang_def)
  then have P44 : "Eq (Geos (Lin (Li o2 l2)) add Emp) (Geos (Lin (Li l2 o2)) add Emp)" by (simp
add:Line_rev)
  from P43 P44 have P45 : "Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹
    Eq (Geos (Lin (Li o2 k2)) add Emp) (Geos (Lin (Li o2 l2)) add Emp)" by (blast intro:Eq_rev
Eq_trans)
  from assms P45 have P46 : "¬ Eq (Geos (Lin (Li o1 k1)) add Emp) (Geos (Lin (Li o1 l1)) add Emp)"
by blast
  from assms P20 P21 P46 have P47 : "Ang_inside (An k1 o1 h1) l1 ∧ ¬ Ang_inside (An l1 o1 h1) k1
    ∨ ¬ Ang_inside (An k1 o1 h1) l1 ∧ Ang_inside (An l1 o1 h1) k1" by (simp add:Ang_inside_case)
  from P19 P47 have "Ang_inside (An k1 o1 h1) l1" by blast
  thus "Ang_inside (An h1 o1 k1) l1" by (simp add:Ang_inside_def Ang_def_rev)
qed

lemma (in Congruence_Rule) Ang_not_Gr_Eq_rev :
  assumes
    "Def (Ang (An p11 p12 p13))" "Def (Ang (An p21 p22 p23))"
    "¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)"
  shows
    "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
    ∨ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)"
proof -
  from assms have "¬ Line_on (Li p12 p13) p11" by (simp add:Ang_to_Tri Tri_def_Line)
  then have "∃p. Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p p12 p13)) add Emp)
    ∧ Plane_sameside (Li p12 p13) p p11" using assms by (simp add:Ang_move_sameside)
  then obtain p4 :: Point where P1 : "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An
p4 p12 p13)) add Emp)
    ∧ Plane_sameside (Li p12 p13) p4 p11" by blast
  from assms P1 have P2 : "¬ Ang_inside (An p11 p12 p13) p4 ∧ ¬ Eq (Geos (Lin (Li p12 p11)) add
Emp) (Geos (Lin (Li p12 p4)) add Emp) ⟹
    Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by (blast
intro:Ang_less_def)
  from assms P2 have P3 : "Ang_inside (An p11 p12 p13) p4 ∨ Eq (Geos (Lin (Li p12 p11)) add Emp)
(Geos (Lin (Li p12 p4)) add Emp)" by blast
  from P1 have P4 : "Plane_sameside (Li p12 p13) p11 p4" by (simp add:Plane_sameside_rev)
  then have P5 : "¬ Line_on (Li p12 p13) p4" by (simp add:Plane_sameside_def)
  from assms have P6 : "¬ Eq (Geos (Poi p12) add Emp) (Geos (Poi p13) add Emp)" by (simp
add:Ang_def)
  from P5 P6 have "Def (Ang (An p12 p13 p4))" by (simp add:Ang_sinple_def)
  then have P7 : "Def (Ang (An p4 p12 p13))" by (blast intro:Ang_def_inv Ang_def_rev)
  from assms P4 P7 have P8 : "Eq (Geos (Lin (Li p12 p11)) add Emp) (Geos (Lin (Li p12 p4)) add
Emp) ⟹
    Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p4 p12 p13)) add Emp)" by (simp
add:Ang_move_unique_inv)
  from assms P1 P7 P8 have P9 : "Eq (Geos (Lin (Li p12 p11)) add Emp) (Geos (Lin (Li p12 p4)) add
Emp) ⟹
    Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by (blast
intro:Ang_trans Ang_rev)
  from P1 have P10 : "Ang_inside (An p11 p12 p13) p4 ⟷
    Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by (simp
add:Ang_greater_def)
  from P3 P9 P10 show "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23))
add Emp)
    ∨ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by blast
qed

lemma (in Congruence_Rule) Ang_not_Eq_Gr :
  assumes
    "Def (Ang (An p11 p12 p13))" "Def (Ang (An p21 p22 p23))"
    "¬ Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)"
```

```
  shows
    "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
    ∨ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)"
proof -
  from assms have "¬ Line_on (Li p12 p13) p11" by (simp add:Ang_to_Tri Tri_def_Line)
  then have "∃p. Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p p12 p13)) add Emp)
    ∧ Plane_sameside (Li p12 p13) p p11" using assms by (simp add:Ang_move_sameside)
  then obtain p4 :: Point where P1 : "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An
p4 p12 p13)) add Emp)
    ∧ Plane_sameside (Li p12 p13) p4 p11" by blast
  from P1 have P2 : "Plane_sameside (Li p12 p13) p11 p4" by (simp add:Plane_sameside_rev)
  then have P3 : "¬ Line_on (Li p12 p13) p4" by (simp add:Plane_sameside_def)
  from assms have P4 : "¬ Eq (Geos (Poi p12) add Emp) (Geos (Poi p13) add Emp)" by (simp
add:Ang_def)
  from P3 P4 have "Def (Ang (An p12 p13 p4))" by (simp add:Ang_sinple_def)
  then have P5 : "Def (Ang (An p4 p12 p13))" by (blast intro:Ang_def_inv Ang_def_rev)
  from assms P2 P5 have P6 : "Eq (Geos (Lin (Li p12 p11)) add Emp) (Geos (Lin (Li p12 p4)) add
Emp) ⟹
    Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p4 p12 p13)) add Emp)" by (simp
add:Ang_move_unique_inv)
  from assms P1 P5 P6 have P7 : "Eq (Geos (Lin (Li p12 p11)) add Emp) (Geos (Lin (Li p12 p4)) add
Emp) ⟹
    Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by (blast
intro:Ang_trans Ang_rev)
  from assms P7 have P8 : "¬ Eq (Geos (Lin (Li p12 p11)) add Emp) (Geos (Lin (Li p12 p4)) add
Emp)" by blast
  from assms P2 P5 P8 have P9 :
    "Ang_inside (An p11 p12 p13) p4 ∧ ¬ Ang_inside (An p4 p12 p13) p11
    ∨ ¬ Ang_inside (An p11 p12 p13) p4 ∧ Ang_inside (An p4 p12 p13) p11"
    by (simp add:Ang_inside_case)
  from P1 have P10 : "Ang_inside (An p11 p12 p13) p4 ⟷
    Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by (simp
add:Ang_greater_def)
  from P1 P8 have P11 : "¬ Ang_inside (An p11 p12 p13) p4 ⟹
    Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by (blast
intro:Ang_less_def)
  from P9 P10 P11 show "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23))
add Emp)
    ∨ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by blast
qed

lemma (in Congruence_Rule) Ang_relation_case :
  assumes
    "Def (Ang (An p11 p12 p13))" "Def (Ang (An p21 p22 p23))"
  shows
    "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
    ∨ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
    ∨ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)"
proof -
  from assms have P1 : "¬ Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22
p23)) add Emp) ⟹
    Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
    ∨ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by (simp
add:Ang_not_Eq_Gr)
  then have P2 : "¬ Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add
Emp) ⟹
    ¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp) ⟹
    Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by blast
  from P1 have P3 : "¬ Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23))
add Emp) ⟹
    ¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp) ⟹
    Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by blast
  from assms have "¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add
Emp) ⟹
    Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
    ∨ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by (simp
```

```
add:Ang_not_Gr_Eq_rev)
  then have P4 : "¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add
Emp) ⟹
    ¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp) ⟹
    Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by blast
  from P2 P3 P4 show "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23))
add Emp)
    ∨ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
    ∨ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by blast
qed

lemma (in Congruence_Rule) Ang_not_Gr_lemma1 :
  assumes
    "Def (Ang (An p11 p12 p13))" "Def (Ang (An p21 p22 p23))"
    "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)"
  shows
    "¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)"
proof -
  from assms have "¬ Line_on (Li p12 p13) p11" by (simp add:Ang_to_Tri Tri_def_Line)
  then have "∃p. Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p p12 p13)) add Emp)
    ∧ Plane_sameside (Li p12 p13) p p11" using assms by (simp add:Ang_move_sameside)
  then obtain p14 :: Point where P1 : "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An
p14 p12 p13)) add Emp)
    ∧ Plane_sameside (Li p12 p13) p14 p11" by blast
  from P1 have P2 : "Plane_sameside (Li p12 p13) p11 p14" by (simp add:Plane_sameside_rev)
  then have P3 : "¬ Line_on (Li p12 p13) p14" by (simp add:Plane_sameside_def)
  from assms have P4 : "¬ Eq (Geos (Poi p12) add Emp) (Geos (Poi p13) add Emp)" by (simp
add:Ang_def)
  from P3 P4 have "Def (Ang (An p12 p13 p14))" by (simp add:Ang_sinple_def)
  then have P5 : "Def (Ang (An p14 p12 p13))" by (blast intro:Ang_def_inv Ang_def_rev)
  from assms P1 P5 have P6 : "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p14
p12 p13)) add Emp)" by (blast intro:Ang_rev Ang_trans)
  have P7 : "Cong (Geos (Ang (An p13 p12 p11)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)"
by (simp add:Ang_roll)
  from assms have P8 : "Def (Ang (An p13 p12 p11))" by (simp add:Ang_def_rev)
  from assms P5 P6 P7 P8 have P9 : "Cong (Geos (Ang (An p13 p12 p11)) add Emp) (Geos (Ang (An
p14 p12 p13)) add Emp)" by (blast intro:Ang_trans Ang_rev)
  from P2 P7 P9 have P10 : "Eq (Geos (Lin (Li p11 p12)) add Emp) (Geos (Lin (Li p14 p12)) add Emp)
∧ ¬ Bet_Point (Se p11 p14) p12" by (simp add:Ang_move_unique)
  from P1 have "Ang_inside (An p11 p12 p13) p14 ⟷
    Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by (simp
add:Ang_greater_def)
  then have "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
⟹
    Ang_inside (An p11 p12 p13) p14" by blast
  then have "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
⟹
    Plane_sameside (Li p12 p11) p13 p14" by (simp add:Ang_inside_def)
  then have P11 : "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add
Emp) ⟹
    ¬ Line_on (Li p12 p11) p14" by (simp add:Plane_sameside_def)
  from assms have "¬ Eq (Geos (Poi p11) add Emp) (Geos (Poi p12) add Emp)" by (simp add:Ang_def)
  then have P12 : "Eq (Geos (Lin (Li p11 p12)) add Emp) (Geos (Lin (Li p12 p11)) add Emp)" by (simp
add:Line_rev)
  from P10 P12 have P13 : "Eq (Geos (Lin (Li p14 p12)) add Emp) (Geos (Lin (Li p12 p11)) add Emp)"
by (blast intro:Eq_rev Eq_trans)
  have P14 : "Line_on (Li p14 p12) p14" by (simp add:Line_on_rule)
  from P13 P14 have P15 : "Line_on (Li p12 p11) p14" by (simp add:Line_on_trans)
  from P11 P15 show "¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23))
add Emp)" by blast
qed

lemma (in Congruence_Rule) Ang_not_Gr :
  assumes
    "Def (Ang (An p11 p12 p13))" "Def (Ang (An p21 p22 p23))"
    "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)"
```

```
  shows
    "¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)"
    "¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)"
proof -
  from assms show P1 : "¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by (simp add:Ang_not_Gr_lemma1)
  from assms have P2 : "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by (simp add:Ang_rev)
  from assms P2 show "¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by (simp add:Ang_not_Gr_lemma1)
qed

lemma (in Congruence_Rule) Ang_Gr_not_Eq_rev :
  assumes
    "Def (Ang (An p11 p12 p13))" "Def (Ang (An p21 p22 p23))"
    "Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)"
  shows
    "¬ Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)"
    "¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)"
proof -
  from assms have "¬ Line_on (Li p12 p13) p11" by (simp add:Ang_to_Tri Tri_def_Line)
  then have "∃p. Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p p12 p13)) add Emp)
    ∧ Plane_sameside (Li p12 p13) p p11" using assms by (simp add:Ang_move_sameside)
  then obtain p14 :: Point where P1 : "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p14 p12 p13)) add Emp)
    ∧ Plane_sameside (Li p12 p13) p14 p11" by blast
  from P1 have P2 : "Plane_sameside (Li p12 p13) p11 p14" by (simp add:Plane_sameside_rev)
  then have P3 : "¬ Line_on (Li p12 p13) p14" by (simp add:Plane_sameside_def)
  from assms have P4 : "¬ Eq (Geos (Poi p12) add Emp) (Geos (Poi p13) add Emp)" by (simp add:Ang_def)
  from P3 P4 have "Def (Ang (An p12 p13 p14))" by (simp add:Ang_sinple_def)
  then have P5 : "Def (Ang (An p14 p12 p13))" by (blast intro:Ang_def_inv Ang_def_rev)
  from assms have "¬ Line_on (Li p22 p23) p21" by (simp add:Ang_to_Tri Tri_def_Line)
  then have "∃p. Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p p22 p23)) add Emp)
    ∧ Plane_sameside (Li p22 p23) p p21" using assms by (simp add:Ang_move_sameside)
  then obtain p24 :: Point where P6 : "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p24 p22 p23)) add Emp)
    ∧ Plane_sameside (Li p22 p23) p24 p21" by blast
  then have P7 : "¬ Line_on (Li p22 p23) p24" by (simp add:Plane_sameside_def)
  from assms have P8 : "¬ Eq (Geos (Poi p22) add Emp) (Geos (Poi p23) add Emp)" by (simp add:Ang_def)
  from P7 P8 have "Def (Ang (An p22 p23 p24))" by (simp add:Ang_sinple_def)
  then have P9 : "Def (Ang (An p24 p22 p23))" by (blast intro:Ang_def_inv Ang_def_rev)
  from P6 have P10 : "Plane_sameside (Li p22 p23) p21 p24" by (simp add:Plane_sameside_rev)
  from assms have P11 : "Def (Ang (An p13 p12 p11))" by (simp add:Ang_def_rev)
  from assms have P12 : "Def (Ang (An p23 p22 p21))" by (simp add:Ang_def_rev)
  from P5 have P13 : "Def (Ang (An p13 p12 p14))" by (simp add:Ang_def_rev)
  from P9 have P14 : "Def (Ang (An p23 p22 p24))" by (simp add:Ang_def_rev)
  have P15 : "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p13 p12 p11)) add Emp)" by (simp add:Ang_roll)
  have P16 : "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p23 p22 p21)) add Emp)" by (simp add:Ang_roll)
  have P17 : "Cong (Geos (Ang (An p14 p12 p13)) add Emp) (Geos (Ang (An p13 p12 p14)) add Emp)" by (simp add:Ang_roll)
  have P18 : "Cong (Geos (Ang (An p24 p22 p23)) add Emp) (Geos (Ang (An p23 p22 p24)) add Emp)" by (simp add:Ang_roll)
  from assms P6 P9 P11 P15 have P19 :
    "Cong (Geos (Ang (An p13 p12 p11)) add Emp) (Geos (Ang (An p24 p22 p23)) add Emp)" by (blast intro:Ang_rev Ang_trans)
  from P9 P11 P14 P18 P19 have P20 :
    "Cong (Geos (Ang (An p13 p12 p11)) add Emp) (Geos (Ang (An p23 p22 p24)) add Emp)" by (blast intro:Ang_rev Ang_trans)
  from assms P1 P5 P12 P16 have P21 :
    "Cong (Geos (Ang (An p23 p22 p21)) add Emp) (Geos (Ang (An p14 p12 p13)) add Emp)" by (blast intro:Ang_rev Ang_trans)
  from P5 P12 P13 P17 P21 have P22 :
```

```
    "Cong (Geos (Ang (An p23 p22 p21)) add Emp) (Geos (Ang (An p13 p12 p14)) add Emp)" by
(blast intro:Ang_rev Ang_trans)
  from P6 have P23 : "Ang_inside (An p21 p22 p23) p24 ⟷
    Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by (simp
add:Ang_greater_def)
  from assms P23 have P24 : "Ang_inside (An p21 p22 p23) p24" by blast
  from P12 P24 have P25 : "Ang_inside (An p23 p22 p21) p24" by (simp add:Ang_inside_def)
  from P2 P6 P11 P12 P20 P22 P25 have P26 : "¬ Ang_inside (An p13 p12 p11) p14
    ∧ ¬ Eq (Geos (Lin (Li p12 p11)) add Emp) (Geos (Lin (Li p12 p14)) add Emp)"
    by (simp add:Ang_move_Greater)
  from P1 have "Ang_inside (An p11 p12 p13) p14 ⟷
    Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by (simp
add:Ang_greater_def)
  then have P27 : "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add
Emp) ⟹
    Ang_inside (An p11 p12 p13) p14" by blast
  from P11 P27 have P28 : "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22
p23)) add Emp) ⟹
    Ang_inside (An p13 p12 p11) p14" by (simp add:Ang_inside_def)
  from P26 P28 show "¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23))
add Emp)" by blast
  from assms P1 P5 have P29 : "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21
p22 p23)) add Emp) ⟹
    Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p14 p12 p13)) add Emp)" by (blast
intro:Ang_rev Ang_trans)
  have P30 : "Cong (Geos (Ang (An p13 p12 p11)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)"
by (simp add:Ang_roll)
  from assms P5 P11 P29 P30 have P31 : "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang
(An p21 p22 p23)) add Emp) ⟹
    Cong (Geos (Ang (An p13 p12 p11)) add Emp) (Geos (Ang (An p14 p12 p13)) add Emp)" by (blast
intro:Ang_rev Ang_trans)
  from P2 P30 P31 have P32 : "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21
p22 p23)) add Emp) ⟹
    Eq (Geos (Lin (Li p11 p12)) add Emp) (Geos (Lin (Li p14 p12)) add Emp) ∧ ¬ Bet_Point (Se p11
p14) p12" by (simp add:Ang_move_unique)
  from assms have "¬ Eq (Geos (Poi p11) add Emp) (Geos (Poi p12) add Emp)" by (simp add:Ang_def)
  then have P33 : "Eq (Geos (Lin (Li p11 p12)) add Emp) (Geos (Lin (Li p12 p11)) add Emp)" by (simp
add:Line_rev)
  from P32 P33 have P34 : "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22
p23)) add Emp) ⟹
    Eq (Geos (Lin (Li p12 p11)) add Emp) (Geos (Lin (Li p14 p12)) add Emp)" by (blast intro:Eq_trans
Eq_rev)
  from P5 have "¬ Eq (Geos (Poi p14) add Emp) (Geos (Poi p12) add Emp)" by (simp add:Ang_def)
  then have P35 : "Eq (Geos (Lin (Li p14 p12)) add Emp) (Geos (Lin (Li p12 p14)) add Emp)" by (simp
add:Line_rev)
  from P34 P35 have P36 : "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22
p23)) add Emp) ⟹
    Eq (Geos (Lin (Li p12 p11)) add Emp) (Geos (Lin (Li p12 p14)) add Emp)" by (blast intro:Eq_trans
Eq_rev)
  from P26 P36 show "¬ Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23))
add Emp)" by blast
qed

lemma (in Congruence_Rule) Ang_relation_case_fact :
  assumes
    "Def (Ang (An p11 p12 p13))" "Def (Ang (An p21 p22 p23))"
  shows
    "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
    ∧ ¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
    ∧ ¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)
    ∨ ¬ Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
    ∧ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
    ∧ ¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)
    ∨ ¬ Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
    ∧ ¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
    ∧ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)"
```

```
proof -
  from assms have P1 : "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
      ∨ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
      ∨ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by (simp add:Ang_relation_case)
  from assms have P2 : "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp) ==¿
      ¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
      ∧ ¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by (simp add:Ang_not_Gr)
  from assms have P3 : "Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp) ⟹
      ¬ Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
      ∧ ¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by (simp add:Ang_Gr_not_Eq_rev)
  from assms have "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp) ⟹
      ¬ Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)
      ∧ ¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by (simp add:Ang_Gr_not_Eq_rev)
  then have P4 : "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp) ⟹
      ¬ Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
      ∧ ¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by (blast intro:Ang_rev)
  from P1 P2 P3 P4 show "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
      ∧ ¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
      ∧ ¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)
      ∨ ¬ Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
      ∧ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
      ∧ ¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)
      ∨ ¬ Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
      ∧ ¬ Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)
      ∧ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by blast
qed

lemma (in Congruence_Rule) Ang_Gr_trans_Eq_Gr :
  assumes
    "Def (Ang (An p11 p12 p13))" "Def (Ang (An p21 p22 p23))" "Def (Ang (An p31 p32 p33))"
    "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)"
    "Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p31 p32 p33)) add Emp)"
  shows
    "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p31 p32 p33)) add Emp)"
proof -
  from assms have P1 : "¬ Line_on (Li p22 p23) p21" by (simp add:Ang_to_Tri Tri_def_Line)
  from assms P1 have "∃p. Cong (Geos (Ang (An p31 p32 p33)) add Emp) (Geos (Ang (An p p22 p23)) add Emp)
      ∧ Plane_sameside (Li p22 p23) p p21" by (simp add:Ang_move_sameside)
  then obtain p24 :: Point where P2 : "Cong (Geos (Ang (An p31 p32 p33)) add Emp) (Geos (Ang (An p24 p22 p23)) add Emp)
      ∧ Plane_sameside (Li p22 p23) p24 p21" by blast
  then have P3 : "Cong (Geos (Ang (An p31 p32 p33)) add Emp) (Geos (Ang (An p24 p22 p23)) add Emp)" by simp
  from P2 have P4 : "Plane_sameside (Li p22 p23) p24 p21" by simp
  from assms P3 P4 have "Ang_inside (An p21 p22 p23) p24" by (simp add:Ang_greater_def)
  then have P5 : "Ang_inside (An p23 p22 p21) p24" by (simp add:Ang_inside_def Ang_def_rev)
  from assms have P6 : "¬ Line_on (Li p32 p33) p31" by (simp add:Ang_to_Tri Tri_def_Line)
  from assms P6 have "∃p. Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p p32 p33)) add Emp)
      ∧ Plane_sameside (Li p32 p33) p p31" by (simp add:Ang_move_sameside)
  then obtain p34 :: Point where P7 : "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p34 p32 p33)) add Emp)
      ∧ Plane_sameside (Li p32 p33) p34 p31" by blast
  then have P8 : "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p34 p32 p33)) add
```

```
Emp)" by simp
  from P7 have P9 : "Plane_sameside (Li p32 p33) p34 p31" by simp
  from assms have P10 : "Def (Ang (An p33 p32 p31))" by (blast intro:Ang_def_rev)
  from assms have P11 : "Def (Ang (An p23 p22 p21))" by (blast intro:Ang_def_rev)
  from P4 have P12 : "¬ Line_on (Li p22 p23) p24" by (simp add:Plane_sameside_def)
  from assms have P13 : "¬ Eq (Geos (Poi p22) add Emp) (Geos (Poi p23) add Emp)" by (simp
add:Ang_def)
  from P12 P13 have "Def (Ang (An p22 p23 p24))" by (simp add:Ang_sinple_def)
  then have P14 : "Def (Ang (An p24 p22 p23))" by (blast intro:Ang_def_rev Ang_def_inv)
  then have P15 : "Def (Ang (An p23 p22 p24))" by (blast intro:Ang_def_rev)
  have P16 : "Cong (Geos (Ang (An p24 p22 p23)) add Emp) (Geos (Ang (An p23 p22 p24)) add Emp)"
by (simp add:Ang_roll)
  from assms P3 P14 P15 P16 have P17 :
    "Cong (Geos (Ang (An p31 p32 p33)) add Emp) (Geos (Ang (An p23 p22 p24)) add Emp)" by
(blast intro:Ang_trans Ang_rev)
  have P18 : "Cong (Geos (Ang (An p31 p32 p33)) add Emp) (Geos (Ang (An p33 p32 p31)) add Emp)"
by (simp add:Ang_roll)
  from assms P10 P15 P17 P18 have P19 :
    "Cong (Geos (Ang (An p33 p32 p31)) add Emp) (Geos (Ang (An p23 p22 p24)) add Emp)"  by
(blast intro:Ang_trans Ang_rev)
  from P9 have P20 : "¬ Line_on (Li p32 p33) p34" by (simp add:Plane_sameside_def)
  from assms have P21 : "¬ Eq (Geos (Poi p32) add Emp) (Geos (Poi p33) add Emp)" by (simp
add:Ang_def)
  from P20 P21 have "Def (Ang (An p32 p33 p34))" by (simp add:Ang_sinple_def)
  then have P22 : "Def (Ang (An p34 p32 p33))" by (blast intro:Ang_def_rev Ang_def_inv)
  then have P23 : "Def (Ang (An p33 p32 p34))" by (blast intro:Ang_def_rev)
  have P24 : "Cong (Geos (Ang (An p34 p32 p33)) add Emp) (Geos (Ang (An p33 p32 p34)) add Emp)"
by (simp add:Ang_roll)
  from assms P8 P22 P23 P24 have P25 :
    "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p33 p32 p34)) add Emp)" by
(blast intro:Ang_trans Ang_rev)
  have P26 : "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p23 p22 p21)) add Emp)"
by (simp add:Ang_roll)
  from assms P11 P23 P25 P26 have P27 :
    "Cong (Geos (Ang (An p23 p22 p21)) add Emp) (Geos (Ang (An p33 p32 p34)) add Emp)" by
(blast intro:Ang_trans Ang_rev)
  from P9 have P28 : "Plane_sameside (Li p32 p33) p31 p34" by (simp add:Plane_sameside_rev)
  from P4 P5 P10 P11 P19 P27 P28 have P29 : "¬ Ang_inside (An p33 p32 p31) p34
    ∧ ¬ Eq (Geos (Lin (Li p32 p31)) add Emp) (Geos (Lin (Li p32 p34)) add Emp)"
    by (simp add:Ang_move_Greater)
  have P30 : "Ang_inside (An p31 p32 p33) p34 ⟹ Ang_inside (An p33 p32 p31) p34" by (simp
add:Ang_inside_def Ang_def_rev)
  from P29 P30 have P31 : "¬ Ang_inside (An p31 p32 p33) p34" by blast
  from assms P8 P22 have P32 : "Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p34
p32 p33)) add Emp)" by (blast intro:Ang_trans Ang_rev)
  from P9 P29 P31 P32 show "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p31 p32
p33)) add Emp)" by (simp add:Ang_less_def)
qed

lemma (in Congruence_Rule) Ang_Gr_trans_Gr_Eq :
  assumes
    "Def (Ang (An p11 p12 p13))" "Def (Ang (An p21 p22 p23))" "Def (Ang (An p31 p32 p33))"
    "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)"
    "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p31 p32 p33)) add Emp)"
  shows
    "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p31 p32 p33)) add Emp)"
proof -
  from assms have P1 : "¬ Line_on (Li p12 p13) p11" by (simp add:Ang_to_Tri Tri_def_Line)
  from assms P1 have "∃p. Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p p12 p13))
add Emp)
    ∧ Plane_sameside (Li p12 p13) p p11" by (simp add:Ang_move_sameside)
  then obtain p14 :: Point where P2 : "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An
p14 p12 p13)) add Emp)
    ∧ Plane_sameside (Li p12 p13) p14 p11" by blast
  then have P3 : "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p14 p12 p13)) add
Emp)" by simp
```

```
  from P2 have P4 : "Plane_sameside (Li p12 p13) p14 p11" by simp
  from P3 P4 have P5 : "Ang_inside (An p11 p12 p13) p14 ⟷
    Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by (simp
add:Ang_greater_def)
  from assms P5 have P6 : "Ang_inside (An p11 p12 p13) p14" by simp
  from P4 have P7 : "¬ Line_on (Li p12 p13) p14" by (simp add:Plane_sameside_def)
  from assms have P8 : "¬ Eq (Geos (Poi p12) add Emp) (Geos (Poi p13) add Emp)" by (simp
add:Ang_def)
  from P7 P8 have "Def (Ang (An p12 p13 p14))" by (simp add:Ang_sinple_def)
  then have P9 : "Def (Ang (An p14 p12 p13))" by (blast intro:Ang_def_rev Ang_def_inv)
  from assms P3 P9 have P10 : "Cong (Geos (Ang (An p31 p32 p33)) add Emp) (Geos (Ang (An p14
p12 p13)) add Emp)" by (blast intro:Ang_trans Ang_rev)
  from P4 P10 have P11 : "Ang_inside (An p11 p12 p13) p14 ⟷
    Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p31 p32 p33)) add Emp)" by (simp
add:Ang_greater_def)
  from P6 P11 show "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p31 p32 p33)) add
Emp)" by simp
qed

lemma (in Congruence_Rule) Ang_Eq_Point :
  assumes
    "Def (Ang (An p1 p2 p3))"
    "Eq (Geos (Poi p1) add Emp) (Geos (Poi p4) add Emp)"
  shows
    "Eq (Geos (Ang (An p1 p2 p3)) add Emp) (Geos (Ang (An p4 p2 p3)) add Emp) ∧ Def (Ang (An
p4 p2 p3))"
proof -
  have "Line_on (Li p2 p1) p1" by (simp add:Line_on_rule)
  then have P1 : "Line_on (Li p2 p1) p4" using assms by (simp add:Point_Eq)
  from assms have P2 : "Bet_Point (Se p1 p4) p2 ⟹ Bet_Point (Se p4 p4) p2" by (simp add:Bet_Point_Eq)
  have P3 : "¬ Bet_Point (Se p4 p4) p2" by (simp add:Bet_end_Point)
  from P2 P3 have P4 : "¬ Bet_Point (Se p1 p4) p2" by blast
  have P5 : "Line_on (Li p2 p3) p3" by (simp add:Line_on_rule)
  have P6 : "¬ Bet_Point (Se p3 p3) p2" by (simp add:Bet_end_Point)
  from assms have P7 : "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)" by (simp add:Ang_def)
  from assms have P8 : "Eq (Geos (Poi p2) add Emp) (Geos (Poi p4) add Emp) ⟹
    Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)" by (blast intro:Eq_trans Eq_rev)
  from P7 P8 have P9 : "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p4) add Emp)" by blast
  from assms have P10 : "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp)" by (simp add:Ang_def)
  from assms P1 P4 P5 P6 P9 P10 show
    "Eq (Geos (Ang (An p1 p2 p3)) add Emp) (Geos (Ang (An p4 p2 p3)) add Emp) ∧ Def (Ang (An
p4 p2 p3))" by (simp add:Ang_Point_swap)
qed

lemma (in Congruence_Rule) Planeside_wrong_relation :
  assumes
    "Plane_diffside (Li p1 p2) p3 p4"
    "Plane_diffside (Li p1 p3) p2 p4"
    "Plane_sameside (Li p1 p5) p3 p2"
    "Plane_sameside (Li p1 p5) p4 p2"
  shows False
proof -
  from assms have "∃p. Bet_Point (Se p3 p4) p ∧ Line_on (Li p1 p2) p
    ∧ ¬ Line_on (Li p1 p2) p3 ∧ ¬ Line_on (Li p1 p2) p4" by (simp add:Plane_diffside_def)
  then obtain p6 :: Point where P1 : "Bet_Point (Se p3 p4) p6 ∧ Line_on (Li p1 p2) p6
    ∧ ¬ Line_on (Li p1 p2) p3 ∧ ¬ Line_on (Li p1 p2) p4" by blast
  from assms have "∃p. Bet_Point (Se p2 p4) p ∧ Line_on (Li p1 p3) p
    ∧ ¬ Line_on (Li p1 p3) p2 ∧ ¬ Line_on (Li p1 p3) p4" by (simp add:Plane_diffside_def)
  then obtain p7 :: Point where P2 : "Bet_Point (Se p2 p4) p7 ∧ Line_on (Li p1 p3) p7
    ∧ ¬ Line_on (Li p1 p3) p2 ∧ ¬ Line_on (Li p1 p3) p4" by blast
  then have P3 : "Bet_Point (Se p2 p4) p7" by simp
  then have P4 : "Line_on (Li p2 p4) p7" by (blast intro:Line_Bet_on)
  from P3 have P5 : "¬ Eq (Geos (Poi p4) add Emp) (Geos (Poi p7) add Emp)" by (simp add:Bet_Point_def)
  have P6 : "Line_on (Li p2 p4) p4" by (simp add:Line_on_rule)
  have P7 : "Line_on (Li p3 p4) p4" by (simp add:Line_on_rule)
  from P4 P5 P6 P7 have P8 : "Line_on (Li p3 p4) p7 ⟹
```

Eq (Geos (Lin (Li p2 p4)) add Emp) (Geos (Lin (Li p3 p4)) add Emp)" by (simp add:Line_unique)
have P9 : "Line_on (Li p2 p4) p2" by (simp add:Line_on_rule)
from P8 P9 have P10 : "Line_on (Li p3 p4) p7 ⟹ Line_on (Li p3 p4) p2" by (simp add:Line_on_trans)
from P7 have P11 : "Eq (Geos (Lin (Li p3 p4)) add Emp) (Geos (Lin (Li p1 p2)) add Emp) ⟹
Line_on (Li p1 p2) p4" by (simp add:Line_on_trans)
from P1 P11 have P12 : "¬ Eq (Geos (Lin (Li p3 p4)) add Emp) (Geos (Lin (Li p1 p2)) add Emp)" by
blast
from P1 have P13 : "Bet_Point (Se p3 p4) p6" by simp
then have P14 : "Line_on (Li p3 p4) p6" by (blast intro:Line_Bet_on)
have P15 : "Line_on (Li p1 p2) p2" by (simp add:Line_on_rule)
from P1 P10 P12 P14 P15 have P16 : "Line_on (Li p3 p4) p7 ⟹
Eq (Geos (Poi p6) add Emp) (Geos (Poi p2) add Emp)" by (simp add:Line_unique_Point)
from P13 P16 have P17 : "Line_on (Li p3 p4) p7 ⟹ Bet_Point (Se p3 p4) p2" by (simp add:Point_Eq)
from P7 have P18 : "Eq (Geos (Lin (Li p3 p4)) add Emp) (Geos (Lin (Li p1 p3)) add Emp) ⟹
Line_on (Li p1 p3) p4" by (simp add:Line_on_trans)
from P2 P18 have P19 : "¬ Eq (Geos (Lin (Li p3 p4)) add Emp) (Geos (Lin (Li p1 p3)) add Emp)" by
blast
have P20 : "Line_on (Li p1 p3) p3" by (simp add:Line_on_rule)
from P17 P19 P20 have "Line_on (Li p3 p4) p7 ⟹ Plane_sameside (Li p1 p3) p2 p4" by (simp
add:Plane_Bet_sameside)
then have "Line_on (Li p3 p4) p7 ⟹ ¬ Plane_diffside (Li p1 p3) p2 p4"
by (simp add:Plane_sameside_not_diffside)
then have P21 : "¬ Line_on (Li p3 p4) p7" using assms by blast
from P3 have P22 : "¬ Eq (Geos (Poi p7) add Emp) (Geos (Poi p2) add Emp)"
by (simp add:Bet_Point_def)
from P4 P9 P15 P22 have P23 : "Line_on (Li p1 p2) p7 ⟹
Eq (Geos (Lin (Li p2 p4)) add Emp) (Geos (Lin (Li p1 p2)) add Emp)" by (simp add:Line_unique)
from P6 P23 have P24 : "Line_on (Li p1 p2) p7 ⟹ Line_on (Li p1 p2) p4" by (simp add:Line_on_trans)
from P1 P24 have P25 : "¬ Line_on (Li p1 p2) p7" by blast
from P1 P13 P21 P25 have P26 : "Line_on_Seg (Li p1 p2) (Se p3 p7) ∧ ¬ Line_on_Seg (Li p1 p2) (Se
p4 p7)
∨ Line_on_Seg (Li p1 p2) (Se p4 p7) ∧ ¬ Line_on_Seg (Li p1 p2) (Se p3 p7)" by (simp
add:Pachets_axiom)
have "Line_on_Seg (Li p1 p2) (Se p4 p7) ⟹ ∃p. Line_on (Li p1 p2) p ∧ Bet_Point (Se p4 p7) p" by
(simp add:Line_on_Seg_rule)
then obtain p8 :: Point where P27 : "Line_on_Seg (Li p1 p2) (Se p4 p7) ⟹
Line_on (Li p1 p2) p8 ∧ Bet_Point (Se p4 p7) p8" by blast
then have P28 : "Line_on_Seg (Li p1 p2) (Se p4 p7) ⟹ Line_on (Li p4 p7) p8" by (blast in-
tro:Line_Bet_on)
from P3 have P29 : "Line_on (Li p4 p7) p2" by (blast intro:Line_Bet_on)
from P27 have "Line_on_Seg (Li p1 p2) (Se p4 p7) ⟹ Bet_Point (Se p4 p7) p8" by simp
then have P30 : "Line_on_Seg (Li p1 p2) (Se p4 p7) ⟹
Eq (Geos (Poi p8) add Emp) (Geos (Poi p2) add Emp) ⟹ Bet_Point (Se p4 p7) p2" by (simp
add:Point_Eq)
from P3 have "Inv (Bet_Point (Se p4 p7) p2)" by (simp add:Bet_iff)
then have P31 : "¬ Bet_Point (Se p4 p7) p2" by (simp add:Inv_def)
from P30 P31 have P32 : "Line_on_Seg (Li p1 p2) (Se p4 p7) ⟹
¬ Eq (Geos (Poi p8) add Emp) (Geos (Poi p2) add Emp)" by blast
from P15 P27 P28 P29 P32 have P33 : "Line_on_Seg (Li p1 p2) (Se p4 p7) ⟹
Eq (Geos (Lin (Li p4 p7)) add Emp) (Geos (Lin (Li p1 p2)) add Emp)" by (simp add:Line_unique)
have P34 : "Line_on (Li p4 p7) p4" by (simp add:Line_on_rule)
from P33 P34 have P35 : "Line_on_Seg (Li p1 p2) (Se p4 p7) ⟹ Line_on (Li p1 p2) p4" by (simp
add:Line_on_trans)
from P1 P35 have P36 : "¬ Line_on_Seg (Li p1 p2) (Se p4 p7)" by blast
from P26 P36 have "Line_on_Seg (Li p1 p2) (Se p3 p7)" by blast
then have "∃p. Line_on (Li p1 p2) p ∧ Bet_Point (Se p3 p7) p" by (simp add:Line_on_Seg_rule)
then obtain p8 :: Point where P37 : "Line_on (Li p1 p2) p8 ∧ Bet_Point (Se p3 p7) p8" by blast
have "Line_on (Li p3 p4) p3" by (simp add:Line_on_rule)
then have P38 : "Eq (Geos (Poi p3) add Emp) (Geos (Poi p7) add Emp) ⟹ Line_on (Li p3 p4) p7"
by (simp add:Point_Eq)
from P21 P38 have P39 : "¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p7) add Emp)" by blast
have P40 : "Line_on (Li p3 p7) p3" by (simp add:Line_on_rule)
have P41 : "Line_on (Li p3 p7) p7" by (simp add:Line_on_rule)
from P2 have P42 : "Line_on (Li p1 p3) p7" by simp
from P20 P39 P40 P41 P42 have P43 :
"Eq (Geos (Lin (Li p1 p3)) add Emp) (Geos (Lin (Li p3 p7)) add Emp)" by (simp add:Line_unique)

```
  have P44 : ”Line_on (Li p1 p3) p1” by (simp add:Line_on_rule)
  from P43 P44 have P45 : ”Line_on (Li p3 p7) p1” by (simp add:Line_on_trans)
  from P37 have P46 : ”Line_on (Li p3 p7) p8” by (blast intro:Line_Bet_on)
  have P47 : ”Line_on (Li p1 p2) p1” by (simp add:Line_on_rule)
  from P40 have P48 : ”Eq (Geos (Lin (Li p3 p7)) add Emp) (Geos (Lin (Li p1 p2)) add Emp) ⟹
    Line_on (Li p1 p2) p3” by (simp add:Line_on_trans)
  from P1 P48 have P49 : ”¬ Eq (Geos (Lin (Li p3 p7)) add Emp) (Geos (Lin (Li p1 p2)) add Emp)” by
blast
  from P37 P45 P46 P48 P47 P49 have P50 : ”Eq (Geos (Poi p8) add Emp) (Geos (Poi p1) add Emp)”
by (simp add:Line_unique_Point)
  from P37 have P51 : ”Bet_Point (Se p3 p7) p8” by simp
  from P50 P51 have P52 : ”Bet_Point (Se p3 p7) p1” by (simp add:Point_Eq)
  from P44 have P53 : ”Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp) ⟹ Line_on (Li p1 p3)
p2” by (simp add:Point_Eq)
  from P2 P53 have P54 : ”¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)” by blast
  have P55 : ”Line_on (Li p1 p2) p2” by (simp add:Line_on_rule)
  from P45 P47 P54 P55 have P56 : ”Line_on (Li p3 p7) p2 ⟹
    Eq (Geos (Lin (Li p3 p7)) add Emp) (Geos (Lin (Li p1 p2)) add Emp)” by (simp add:Line_unique)
  from P49 P56 have P57 : ”¬ Line_on (Li p3 p7) p2” by blast
  from assms have P58 : ”¬ Line_on_Seg (Li p1 p5) (Se p3 p2) ∧ ¬ Line_on (Li p1 p5) p3
    ∧ ¬ Line_on (Li p1 p5) p2 ∧ ¬ Eq (Geos (Poi p3) add Emp) (Geos (Poi p2) add Emp)” by (simp
add:Plane_sameside_def)
  from P52 have P59 : ”¬ Eq (Geos (Poi p7) add Emp) (Geos (Poi p1) add Emp)”
    by (simp add:Bet_Point_def)
  have P60 : ”Line_on (Li p1 p5) p1” by (simp add:Line_on_rule)
  from P41 P45 P59 P60 have P61 : ”Line_on (Li p1 p5) p7 ⟹
    Eq (Geos (Lin (Li p3 p7)) add Emp) (Geos (Lin (Li p1 p5)) add Emp)” by (simp add:Line_unique)
  from P40 P61 have P62 : ”Line_on (Li p1 p5) p7 ⟹ Line_on (Li p1 p5) p3” by (simp add:Line_on_trans)
  from P58 P62 have P63 : ”¬ Line_on (Li p1 p5) p7” by blast
  from P58 have P64 : ”¬ Line_on (Li p1 p5) p3” by simp
  from P58 have P65 : ”¬ Line_on (Li p1 p5) p2” by simp
  from P52 P57 P60 P63 P64 P65 have P66 : ”Line_on_Seg (Li p1 p5) (Se p3 p2) ∧ ¬ Line_on_Seg (Li
p1 p5) (Se p7 p2)
          ∨ Line_on_Seg (Li p1 p5) (Se p7 p2) ∧ ¬ Line_on_Seg (Li p1 p5) (Se p3 p2)” by (simp
add:Pachets_axiom)
  from P58 P66 have ”Line_on_Seg (Li p1 p5) (Se p7 p2)” by blast
  then have ”∃p. Line_on (Li p1 p5) p ∧ Bet_Point (Se p7 p2) p” by (simp add:Line_on_Seg_rule)
  then obtain p9 :: Point where P67 : ”Line_on (Li p1 p5) p9 ∧ Bet_Point (Se p7 p2) p9” by blast
  then have P68 : ”Bet_Point (Se p2 p7) p9” by (simp add:Bet_rev)
  from P3 P68 have ”Bet_Point (Se p2 p4) p9” by (blast intro:Bet_swap_134_124)
  then have P69 : ”Bet_Point (Se p4 p2) p9” by (simp add:Bet_rev)
  from P67 P69 have ”∃p. Line_on (Li p1 p5) p ∧ Bet_Point (Se p4 p2) p” by blast
  then have P70 : ”Line_on_Seg (Li p1 p5) (Se p4 p2)” by (simp add:Line_on_Seg_rule)
  from assms have P71 : ”¬ Line_on_Seg (Li p1 p5) (Se p4 p2)” by (simp add:Plane_sameside_def)
  from P70 P71 show False by blast
qed

lemma (in Congruence_Rule) Ang_Gr_trans_Gr_Gr :
  assumes
    ”Def (Ang (An p11 p12 p13))” ”Def (Ang (An p21 p22 p23))” ”Def (Ang (An p31 p32 p33))”
    ”Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)”
    ”Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p31 p32 p33)) add Emp)”
  shows
    ”Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p31 p32 p33)) add Emp)”
proof -
  from assms have P1 : ”¬ Line_on (Li p12 p13) p11” by (simp add:Ang_to_Tri Tri_def_Line)
  from assms P1 have ”∃p. Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p p12 p13))
add Emp)
    ∧ Plane_sameside (Li p12 p13) p p11” by (simp add:Ang_move_sameside)
  then obtain p14 :: Point where P2 : ”Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An
p14 p12 p13)) add Emp)
    ∧ Plane_sameside (Li p12 p13) p14 p11” by blast
  from assms P1 have ”∃p. Cong (Geos (Ang (An p31 p32 p33)) add Emp) (Geos (Ang (An p p12 p13))
add Emp)
    ∧ Plane_sameside (Li p12 p13) p p11” by (simp add:Ang_move_sameside)
  then obtain p15 :: Point where P3 : ”Cong (Geos (Ang (An p31 p32 p33)) add Emp) (Geos (Ang (An
```

p15 p12 p13)) add Emp)
∧ Plane_sameside (Li p12 p13) p15 p11" by blast
from P2 have P4 : "Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p14 p12 p13)) add Emp)" by simp
from P2 have P5 : "Plane_sameside (Li p12 p13) p14 p11" by simp
from P3 have P6 : "Cong (Geos (Ang (An p31 p32 p33)) add Emp) (Geos (Ang (An p15 p12 p13)) add Emp)" by simp
from P3 have P7 : "Plane_sameside (Li p12 p13) p15 p11" by simp
from P4 P5 have "Ang_inside (An p11 p12 p13) p14 ⟷
Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by (simp add:Ang_greater_def)
then have P8 : "Ang_inside (An p11 p12 p13) p14" using assms by blast
from P5 have P9 : "¬ Line_on (Li p12 p13) p14" by (simp add:Plane_sameside_def)
from assms have P10 : "¬ Eq (Geos (Poi p12) add Emp) (Geos (Poi p13) add Emp)" by (simp add:Ang_def)
from P9 P10 have "Def (Ang (An p12 p13 p14))" by (simp add:Ang_sinple_def)
then have P11 : "Def (Ang (An p14 p12 p13))" by (blast intro:Ang_def_rev Ang_def_inv)
from assms P4 P11 have P12 : "Gr (Geos (Ang (An p14 p12 p13)) add Emp) (Geos (Ang (An p31 p32 p33)) add Emp)" by (blast intro:Ang_Gr_trans_Eq_Gr Ang_rev)
from P11 have P13 : "Eq (Geos (Poi p14) add Emp) (Geos (Poi p15) add Emp) ⟹
Eq (Geos (Ang (An p14 p12 p13)) add Emp) (Geos (Ang (An p15 p12 p13)) add Emp) ∧ Def (Ang (An p15 p12 p13))" by (simp add:Ang_Eq_Point)
then have P14 : "Eq (Geos (Poi p14) add Emp) (Geos (Poi p15) add Emp) ⟹
Cong (Geos (Ang (An p14 p12 p13)) add Emp) (Geos (Ang (An p15 p12 p13)) add Emp)" by (blast intro:Ang_weektrans)
from assms P4 P11 P13 P14 have P15 : "Eq (Geos (Poi p14) add Emp) (Geos (Poi p15) add Emp) ⟹
Cong (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p15 p12 p13)) add Emp)" by (blast intro:Ang_trans Ang_rev)
from assms P6 P13 P15 have P16 : "Eq (Geos (Poi p14) add Emp) (Geos (Poi p15) add Emp) ⟹
Cong (Geos (Ang (An p31 p32 p33)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by (blast intro:Ang_trans Ang_rev)
from assms have P17 : "¬ Cong (Geos (Ang (An p31 p32 p33)) add Emp) (Geos (Ang (An p21 p22 p23)) add Emp)" by (simp add:Ang_Gr_not_Eq_rev)
from P16 P17 have P18 : "¬ Eq (Geos (Poi p15) add Emp) (Geos (Poi p14) add Emp)" by (blast intro:Eq_rev)
from P7 have P19 : "Plane_sameside (Li p12 p13) p11 p15" by (simp add:Plane_sameside_rev)
from P5 P18 P19 have P20 : "Plane_sameside (Li p12 p13) p15 p14"
by (blast intro:Plane_sameside_trans Plane_sameside_rev)
from P6 P20 have P21 : "Ang_inside (An p14 p12 p13) p15 ⟷
Gr (Geos (Ang (An p14 p12 p13)) add Emp) (Geos (Ang (An p31 p32 p33)) add Emp)" by (simp add:Ang_greater_def)
from P12 P21 have "Ang_inside (An p14 p12 p13) p15" by simp
then have P22 : "Plane_sameside (Li p12 p14) p13 p15 ∧ Plane_sameside (Li p12 p13) p14 p15" by (simp add:Ang_inside_def)
from P8 have P23 : "Plane_sameside (Li p12 p11) p13 p14 ∧ Plane_sameside (Li p12 p13) p11 p14" by (simp add:Ang_inside_def)
then have P24 : "Plane_diffside (Li p12 p11) p13 p15 ⟹ Plane_diffside (Li p12 p11) p14 p15" by (blast intro:Plane_trans)
from P8 have P25 : "Plane_diffside (Li p12 p14) p11 p13" by (simp add:Ang_inside_Planeside)
from P22 P25 have P26 : "Plane_diffside (Li p12 p14) p11 p15"
by (blast intro:Plane_trans Plane_diffside_rev)
from P5 P7 P24 P26 have "Plane_diffside (Li p12 p11) p13 p15 ⟹ False"
by (blast intro:Planeside_wrong_relation)
then have P27 : "¬ Plane_diffside (Li p12 p11) p13 p15" by blast
from P23 have P28 : "¬ Line_on (Li p12 p11) p13" by (simp add:Plane_sameside_def)
have P29 : "Line_on (Li p12 p13) p12" by (simp add:Line_on_rule)
from P19 have P30 : "¬ Line_on (Li p12 p13) p11" by (simp add:Plane_sameside_def)
from P19 have P31 : "¬ Line_on (Li p12 p13) p15" by (simp add:Plane_sameside_def)
from P29 P30 P31 have "Bet_Point (Se p11 p15) p12 ⟹ ∃p. Bet_Point (Se p11 p15) p ∧
Line_on (Li p12 p13) p ∧ ¬ Line_on (Li p12 p13) p11 ∧ ¬ Line_on (Li p12 p13) p15" by blast
then have "Bet_Point (Se p11 p15) p12 ⟹ Plane_diffside (Li p12 p13) p11 p15"
by (simp add:Plane_diffside_def)
then have P32 : "Bet_Point (Se p11 p15) p12 ⟹ ¬ Plane_sameside (Li p12 p13) p15 p11" by (simp add:Plane_diffside_rev Plane_diffside_not_sameside)
from P7 P32 have P33 : "¬ Bet_Point (Se p11 p15) p12" by blast

```
  have P34 : "Line_on (Li p12 p13) p13" by (simp add:Line_on_rule)
  have P35 : "¬ Bet_Point (Se p13 p13) p12" by (simp add:Bet_end_Point)
  from P29 have P36 : "Eq (Geos (Poi p12) add Emp) (Geos (Poi p15) add Emp) ⟹
    Line_on (Li p12 p13) p15" by (simp add:Point_Eq)
  from P31 P36 have P37 : "¬ Eq (Geos (Poi p12) add Emp) (Geos (Poi p15) add Emp)" by blast
  from assms have P38 : "¬ Eq (Geos (Poi p12) add Emp) (Geos (Poi p13) add Emp)" by (simp
add:Ang_def)
  from assms P33 P34 P35 P37 P38 have P39 : "Line_on (Li p12 p11) p15 ⟹
    Eq (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p15 p12 p13)) add Emp) ∧ Def (Ang
(An p15 p12 p13))" by (simp add:Ang_Point_swap)
  then have P40 : "Line_on (Li p12 p11) p15 ⟹
    Cong (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p15 p12 p13)) add Emp)" by (blast
intro:Ang_weektrans)
  from assms P6 P39 P40 have P41 : "Line_on (Li p12 p11) p15 ⟹
    Cong (Geos (Ang (An p31 p32 p33)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by (blast
intro:Ang_trans Ang_rev)
  from assms P41 have P42 : "Line_on (Li p12 p11) p15 ⟹
    Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12 p13)) add Emp)" by (blast
intro:Ang_Gr_trans_Gr_Eq)
  from assms have P43 : "¬ Gr (Geos (Ang (An p21 p22 p23)) add Emp) (Geos (Ang (An p11 p12
p13)) add Emp)" by (simp add:Ang_Gr_not_Eq_rev)
  from P42 P43 have P44: "¬ Line_on (Li p12 p11) p15" by blast
  from P34 have P45 : "Eq (Geos (Poi p13) add Emp) (Geos (Poi p15) add Emp) ⟹
    Line_on (Li p12 p13) p15" by (simp add:Point_Eq)
  from P31 P45 have P46 : "¬ Eq (Geos (Poi p13) add Emp) (Geos (Poi p15) add Emp)" by blast
  from P27 P28 P44 P46 have P47 : "Plane_sameside (Li p12 p11) p13 p15"
    by (simp add:Plane_not_diffside_sameside)
  from assms P19 P47 have P48 : "Ang_inside (An p11 p12 p13) p15" by (simp add:Ang_inside_def)
  from P6 P7 have P49 : "Ang_inside (An p11 p12 p13) p15 ⟷
    Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p31 p32 p33)) add Emp)" by (simp
add:Ang_greater_def)
  from P48 P49 show "Gr (Geos (Ang (An p11 p12 p13)) add Emp) (Geos (Ang (An p31 p32 p33)) add
Emp)" by simp
qed

lemma (in Congruence_Rule) Ang_complementary_inside :
  assumes
    "Def (Ang (An p1 p2 p3))"
    "Bet_Point (Se p3 p4) p2"
    "Ang_inside (An p5 p2 p3) p1"
  shows
    "Ang_inside (An p1 p2 p4) p5"
proof -
  from assms have P1 : "Plane_sameside (Li p2 p5) p3 p1 ∧ Plane_sameside (Li p2 p3) p5 p1" by (simp
add:Ang_inside_def)
  from assms have P2 : "Line_on (Li p2 p3) p4" by (simp add:Line_Bet_on)
  have P3 : "Line_on (Li p2 p3) p2" by (simp add:Line_on_rule)
  have P4 : "Line_on (Li p2 p4) p2" by (simp add:Line_on_rule)
  have P5 : "Line_on (Li p2 p4) p4" by (simp add:Line_on_rule)
  from assms have P6 : "¬ Eq (Geos (Poi p4) add Emp) (Geos (Poi p2) add Emp)"
    by (simp add:Bet_Point_def)
  from P2 P3 P4 P5 P6 have P7 : "Eq (Geos (Lin (Li p2 p3)) add Emp) (Geos (Lin (Li p2 p4)) add
Emp)" by (simp add:Line_unique)
  from P1 P7 have P8 : "Plane_sameside (Li p2 p4) p5 p1" by (blast intro:Plane_Line_trans)
  have P9 : "Line_on (Li p2 p1) p2" by (simp add:Line_on_rule)
  have "Line_on (Li p2 p1) p1" by (simp add:Line_on_rule)
  then have P10 : "Eq (Geos (Lin (Li p3 p4)) add Emp) (Geos (Lin (Li p2 p1)) add Emp) ⟹
    Line_on (Li p3 p4) p1" by (blast intro:Eq_rev Line_on_trans)
  from P1 have P11 : "¬ Line_on (Li p2 p3) p1" by (simp add:Plane_sameside_def)
  from assms have P12 : "Line_on (Li p3 p4) p2" by (simp add:Line_Bet_on)
  have P13 : "Line_on (Li p3 p4) p3" by (simp add:Line_on_rule)
  have P14 : "Line_on (Li p2 p3) p3" by (simp add:Line_on_rule)
  from assms have P15 : "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p3) add Emp)"
    by (simp add:Bet_Point_def)
  from P3 P12 P13 P14 P15 have P16 : "Eq (Geos (Lin (Li p3 p4)) add Emp) (Geos (Lin (Li p2 p3))
add Emp)" by (simp add:Line_unique)
```

```
  from P10 P16 have P17 : "Eq (Geos (Lin (Li p3 p4)) add Emp) (Geos (Lin (Li p2 p1)) add Emp) ⟹
    Line_on (Li p2 p3) p1" by (simp add:Line_on_trans)
  from P11 P17 have P18 : "¬ Eq (Geos (Lin (Li p3 p4)) add Emp) (Geos (Lin (Li p2 p1)) add Emp)"
by blast
  from assms P9 P18 have P19 : "Plane_diffside (Li p2 p1) p3 p4" by (simp add:Plane_Bet_diffside)
  from assms have P20 : "Plane_diffside (Li p2 p1) p3 p5"
    by (simp add:Ang_inside_Planeside Plane_diffside_rev)
  from P5 have P21 : "Eq (Geos (Poi p4) add Emp) (Geos (Poi p5) add Emp) ⟹ Line_on (Li p2 p4)
p5" by (simp add:Point_Eq)
  from P8 have P22 : "¬ Line_on (Li p2 p4) p5" by (simp add:Plane_sameside_def)
  from P21 P22 have P23 : "¬ Eq (Geos (Poi p4) add Emp) (Geos (Poi p5) add Emp)" by blast
  from P19 P20 P23 have P24 : "Plane_sameside (Li p2 p1) p4 p5" by (simp add:Plane_trans_inv)
  then have P25 : "¬ Line_on (Li p2 p1) p4" by (simp add:Plane_sameside_def)
  from assms have "¬ Eq (Geos (Poi p1) add Emp) (Geos (Poi p2) add Emp)" by (simp add:Ang_def)
  then have P26 : "¬ Eq (Geos (Poi p2) add Emp) (Geos (Poi p1) add Emp)" by (blast intro:Eq_rev)
  from P25 P26 have "Def (Ang (An p2 p1 p4))" by (simp add:Ang_sinple_def)
  then have P27 : "Def (Ang (An p1 p2 p4))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P8 have P28 : "Plane_sameside (Li p2 p4) p1 p5" by (simp add:Plane_sameside_rev)
  from P24 P27 P28 show "Ang_inside (An p1 p2 p4) p5" by (simp add:Ang_inside_def)
qed

theorem (in Congruence_Rule) Ang_Right_angle_Cong :
  assumes
    "Right_angle (An l1 o1 h1)" "Right_angle (An l2 o2 h2)"
  shows
    "Cong (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An l2 o2 h2)) add Emp)"
proof -
  from assms have "∃p. Cong (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An l1 o1 p)) add Emp)
          ∧ Bet_Point (Se h1 p) o1 ∧ Def (Ang (An l1 o1 h1)) ∧ Def (Ang (An l1 o1 p))" by (simp
add:Ang_Right_angle_def)
  then obtain k1 :: Point where P1 : "Cong (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An l1 o1
k1)) add Emp)
          ∧ Bet_Point (Se h1 k1) o1 ∧ Def (Ang (An l1 o1 h1)) ∧ Def (Ang (An l1 o1 k1))" by blast
  from assms have "∃p. Cong (Geos (Ang (An l2 o2 h2)) add Emp) (Geos (Ang (An l2 o2 p)) add Emp)
          ∧ Bet_Point (Se h2 p) o2 ∧ Def (Ang (An l2 o2 h2)) ∧ Def (Ang (An l2 o2 p))" by (simp
add:Ang_Right_angle_def)
  then obtain k2 :: Point where P2 : "Cong (Geos (Ang (An l2 o2 h2)) add Emp) (Geos (Ang (An l2 o2
k2)) add Emp)
          ∧ Bet_Point (Se h2 k2) o2 ∧ Def (Ang (An l2 o2 h2)) ∧ Def (Ang (An l2 o2 k2))" by blast
  from P1 have P3 : "¬ Line_on (Li o1 h1) l1" by (simp add:Ang_to_Tri Tri_def_Line)
  from P1 P2 P3 have "∃p. Cong (Geos (Ang (An l2 o2 h2)) add Emp) (Geos (Ang (An p o1 h1)) add
Emp)
    ∧ Plane_sameside (Li o1 h1) p l1" by (simp add:Ang_move_sameside)
  then obtain l11 :: Point where P4 : "Cong (Geos (Ang (An l2 o2 h2)) add Emp) (Geos (Ang (An l11
o1 h1)) add Emp)
    ∧ Plane_sameside (Li o1 h1) l11 l1" by blast
  then have P5 : "¬ Line_on (Li o1 h1) l11" by (simp add:Plane_sameside_def)
  from P1 have P6 : "¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi h1) add Emp)" by (simp add:Ang_def)
  from P5 P6 have "Def (Ang (An o1 h1 l11))" by (simp add:Ang_sinple_def)
  then have P7 : "Def (Ang (An l11 o1 h1))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P2 P4 P7 have P8 : "Cong (Geos (Ang (An l2 o2 k2)) add Emp) (Geos (Ang (An l11 o1 h1))
add Emp)" by (blast intro:Ang_rev Ang_trans)
  from P2 have P9 : "Def (Ang (An h2 o2 l2))" by (simp add:Ang_def_rev)
  from P7 have P10 : "Def (Ang (An h1 o1 l11))" by (simp add:Ang_def_rev)
  have P11 : "Cong (Geos (Ang (An l2 o2 h2)) add Emp) (Geos (Ang (An h2 o2 l2)) add Emp)" by
(simp add:Ang_roll)
  have P12 : "Cong (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An h1 o1 l11)) add Emp)" by
(simp add:Ang_roll)
  from P2 P4 P7 P9 P11 have P13 : "Cong (Geos (Ang (An h2 o2 l2)) add Emp) (Geos (Ang (An l11
o1 h1)) add Emp)" by (blast intro:Ang_rev Ang_trans)
  from P7 P9 P10 P12 P13 have P14 : "Cong (Geos (Ang (An h2 o2 l2)) add Emp) (Geos (Ang (An h1
o1 l11)) add Emp)" by (blast intro:Ang_rev Ang_trans)
  from P1 P2 P9 P10 P14 have P15 : "Cong (Geos (Ang (An l2 o2 k2)) add Emp) (Geos (Ang (An l11
o1 k1)) add Emp)" by (simp add:Ang_complementary)
  from P1 P4 P7 have P16 : "Eq (Geos (Lin (Li o1 l11)) add Emp) (Geos (Lin (Li o1 l1)) add Emp) ⟹
    Cong (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l1 o1 h1)) add Emp)" by (simp
```

```
add:Ang_move_unique_inv)
  from P1 P2 P4 P7 P16 have "Eq (Geos (Lin (Li o1 l11)) add Emp) (Geos (Lin (Li o1 l1)) add Emp)
⟹
    Cong (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An l2 o2 h2)) add Emp)" by (blast in-
tro:Ang_trans Ang_rev)
  then have P17 : "¬ Cong (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An l2 o2 h2)) add Emp)
⟹
    ¬ Eq (Geos (Lin (Li o1 l11)) add Emp) (Geos (Lin (Li o1 l1)) add Emp)" by blast
  from P1 P4 P7 P17 have P18 : "¬ Cong (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An l2 o2
h2)) add Emp) ⟹
    Ang_inside (An l11 o1 h1) l1 ∧ ¬ Ang_inside (An l1 o1 h1) l11
    ∨ ¬ Ang_inside (An l11 o1 h1) l1 ∧ Ang_inside (An l1 o1 h1) l11" by (simp add:Ang_inside_case)
  from P4 have P19 : "Plane_sameside (Li o1 h1) l1 l11" by (simp add:Plane_sameside_rev)
  have P20 : "Cong (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An l1 o1 h1)) add Emp)" by
simp
  from P19 P20 have P21 : "Ang_inside (An l11 o1 h1) l1 ⟹
    Gr (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l1 o1 h1)) add Emp)"
    by (simp add:Ang_greater_def)
  from P1 have P22 : "Line_on (Li o1 h1) k1" by (simp add:Line_Bet_on)
  have P23 : "Line_on (Li o1 h1) o1" by (simp add:Line_on_rule)
  have P24 : "Line_on (Li o1 k1) k1" by (simp add:Line_on_rule)
  have P25 : "Line_on (Li o1 k1) o1" by (simp add:Line_on_rule)
  from P1 have P26 : "Bet_Point (Se h1 k1) o1" by simp
  then have P27 : "¬ Eq (Geos (Poi k1) add Emp) (Geos (Poi o1) add Emp)" by (simp add:Bet_Point_def)
  from P22 P23 P24 P25 P27 have P28 : "Eq (Geos (Lin (Li o1 h1)) add Emp) (Geos (Lin (Li o1 k1))
add Emp)" by (simp add:Line_unique)
  from P19 P28 have P29 : "Plane_sameside (Li o1 k1) l1 l11" by (simp add:Plane_Line_trans)
  then have P30 : "Plane_sameside (Li o1 k1) l11 l1" by (simp add:Plane_sameside_rev)
  then have P31 : "¬ Line_on (Li o1 k1) l11" by (simp add:Plane_sameside_def)
  from P1 have P32 : "¬ Eq (Geos (Poi o1) add Emp) (Geos (Poi k1) add Emp)" by (simp add:Ang_def)
  from P31 P32 have "Def (Ang (An o1 k1 l11))" by (simp add:Ang_sinple_def)
  then have P33 : "Def (Ang (An l11 o1 k1))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P2 P7 P8 P15 P33 have P34 : "Cong (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l11
o1 k1)) add Emp)" by (blast intro:Ang_rev Ang_trans)
  from P1 have P35 : "Def (Ang (An l1 o1 h1))" by simp
  from P26 P35 have P36 : "Ang_inside (An l11 o1 h1) l1 ⟹ Ang_inside (An l1 o1 k1) l11" by (simp
add:Ang_complementary_inside)
  have P37 : "Cong (Geos (Ang (An l11 o1 k1)) add Emp) (Geos (Ang (An l11 o1 k1)) add Emp)" by
simp
  from P30 P36 P37 have P38 : "Ang_inside (An l11 o1 h1) l1 ⟹
    Gr (Geos (Ang (An l1 o1 k1)) add Emp) (Geos (Ang (An l11 o1 k1)) add Emp)"
    by (simp add:Ang_greater_def)
  from P1 P7 P21 have P39 : "Ang_inside (An l11 o1 h1) l1 ⟹
    Gr (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l1 o1 k1)) add Emp)" by (blast in-
tro:Ang_Gr_trans_Gr_Eq)
  from P1 P7 P33 P38 P39 have P40 : "Ang_inside (An l11 o1 h1) l1 ⟹
    Gr (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l11 o1 k1)) add Emp)" by (blast in-
tro:Ang_Gr_trans_Gr_Gr)
  from P7 P33 have P41 :
    "Cong (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l11 o1 k1)) add Emp)
    ∧ ¬ Gr (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l11 o1 k1)) add Emp)
    ∧ ¬ Gr (Geos (Ang (An l11 o1 k1)) add Emp) (Geos (Ang (An l11 o1 h1)) add Emp)
    ∨ ¬ Cong (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l11 o1 k1)) add Emp)
    ∧ Gr (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l11 o1 k1)) add Emp)
    ∧ ¬ Gr (Geos (Ang (An l11 o1 k1)) add Emp) (Geos (Ang (An l11 o1 h1)) add Emp)
    ∨ ¬ Cong (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l11 o1 k1)) add Emp)
    ∧ ¬ Gr (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l11 o1 k1)) add Emp)
    ∧ Gr (Geos (Ang (An l11 o1 k1)) add Emp) (Geos (Ang (An l11 o1 h1)) add Emp)" by (simp
add:Ang_relation_case_fact)
  from P40 P41 have P42 : "Ang_inside (An l11 o1 h1) l1 ⟹
    ¬ Cong (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l11 o1 k1)) add Emp)" by blast
  from P34 P42 have P43 : "¬ Ang_inside (An l11 o1 h1) l1" by blast
  have P44 : "Cong (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l11 o1 h1)) add Emp)" by
simp
  from P4 P44 have P45 : "Ang_inside (An l1 o1 h1) l11 ⟹
    Gr (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An l11 o1 h1)) add Emp)"
```

```
    by (simp add:Ang_greater_def)
  from P1 P7 have P46 : "Ang_inside (An l1 o1 h1) l11 ⟹ Ang_inside (An l11 o1 k1) l1" by (simp
add:Ang_complementary_inside)
  have P47 : "Cong (Geos (Ang (An l1 o1 k1)) add Emp) (Geos (Ang (An l1 o1 k1)) add Emp)" by
simp
  from P29 P46 P47 have P48 : "Ang_inside (An l1 o1 h1) l11 ⟹
    Gr (Geos (Ang (An l11 o1 k1)) add Emp) (Geos (Ang (An l1 o1 k1)) add Emp)"
    by (simp add:Ang_greater_def)
  from P1 P33 P48 have P49 : "Ang_inside (An l1 o1 h1) l11 ⟹
    Gr (Geos (Ang (An l11 o1 k1)) add Emp) (Geos (Ang (An l1 o1 h1)) add Emp)" by (blast in-
tro:Ang_Gr_trans_Gr_Eq Ang_rev)
  from P1 P7 P33 P45 P49 have P50 : "Ang_inside (An l1 o1 h1) l11 ⟹
    Gr (Geos (Ang (An l11 o1 k1)) add Emp) (Geos (Ang (An l11 o1 h1)) add Emp)" by (blast in-
tro:Ang_Gr_trans_Gr_Gr)
  from P41 P50 have P51 : "Ang_inside (An l1 o1 h1) l11 ⟹
    ¬ Cong (Geos (Ang (An l11 o1 h1)) add Emp) (Geos (Ang (An l11 o1 k1)) add Emp)" by blast
  from P34 P51 have P52 : "¬ Ang_inside (An l1 o1 h1) l11" by blast
  from P18 P43 P52 show "Cong (Geos (Ang (An l1 o1 h1)) add Emp) (Geos (Ang (An l2 o2 h2)) add
Emp)" by blast
qed

lemma (in Congruence_Rule) Ang_external_Gr_lemma1 :
  assumes N :
    "Def (Tri (Tr A B C))"
    "Bet_Point (Se B D) A"
  shows "¬ Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An A C B)) add Emp)"
proof
  assume W : "Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An A C B)) add Emp)"
  from N have P1 : "Line_on (Li B D) A" by (simp add:Line_Bet_on)
  have P2 : "Line_on (Li B D) D" by (simp add:Line_on_rule)
  from N have "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi A) add Emp)" by (simp add:Bet_Point_def)
  then have P3 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)" by (blast intro:Eq_rev)
  from N have P4 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)" by (simp add:Tri_def)
  then have P5 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi B) add Emp)" by (blast intro:Eq_rev)
  from P1 P2 P3 P5 have "∃p. Eq (Geos (Seg (Se C B)) add Emp) (Geos (Seg (Se A p)) add Emp)
    ∧ ¬ Bet_Point (Se p D) A ∧ Line_on (Li B D) p ∧ ¬ Eq (Geos (Poi A) add Emp) (Geos (Poi p) add
Emp)" by (simp add:Seg_move_sameside)
  then obtain D1 :: Point where P6 : "Eq (Geos (Seg (Se C B)) add Emp) (Geos (Seg (Se A D1)) add
Emp)
    ∧ ¬ Bet_Point (Se D1 D) A ∧ Line_on (Li B D) D1 ∧ ¬ Eq (Geos (Poi A) add Emp) (Geos (Poi
D1) add Emp)" by blast
  have P7 : "Line_on (Li A D) A" by (simp add:Line_on_rule)
  from N have P8 : "Line_on (Li A D) B" by (simp add:Line_Bet_on)
  have P9 : "Line_on (Li A B) A" by (simp add:Line_on_rule)
  have P10 : "Line_on (Li A B) B" by (simp add:Line_on_rule)
  from N have P11 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)" by (simp add:Tri_def)
  from P7 P8 P9 P10 P11 have "Eq (Geos (Lin (Li A D)) add Emp) (Geos (Lin (Li A B)) add Emp)" by
(simp add:Line_unique)
  then have P12 : "Line_on (Li A D) C ⟹ Line_on (Li A B) C" by (simp add:Line_on_trans)
  from N have P13 : "¬ Line_on (Li A B) C" by (simp add:Tri_def_Line)
  from P12 P13 have P14 : "¬ Line_on (Li A D) C" by blast
  from P3 P14 have "Def (Ang (An A D C))" by (simp add:Ang_sinple_def)
  then have P15 : "Def (Ang (An C A D))" by (blast intro:Ang_def_rev Ang_def_inv)
  have P16 : "Line_on (Li A C) C" by (simp add:Line_on_rule)
  have P17 : "¬ Bet_Point (Se C C) A" by (simp add:Bet_end_Point)
  have P18 : "Line_on (Li A D) D" by (simp add:Line_on_rule)
  from P1 P2 P3 P7 P18 have P19 : "Eq (Geos (Lin (Li B D)) add Emp) (Geos (Lin (Li A D)) add
Emp)" by (simp add:Line_unique)
  from P6 P19 have P20 : "Line_on (Li A D) D1" by (simp add:Line_on_trans)
  from P6 have P21 : "¬ Bet_Point (Se D D1) A" by (blast intro:Bet_rev)
  from N have "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)" by (simp add:Tri_def)
  then have P22 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)" by (blast intro:Eq_rev)
  from P6 P15 P16 P17 P20 P21 P22 have P23 :
    "Eq (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An C A D1)) add Emp)
    ∧ Def (Ang (An C A D1))" by (simp add:Ang_Point_swap)
  have P24 : "Cong (Geos (Ang (An C A D1)) add Emp) (Geos (Ang (An C A D1)) add Emp)" by simp
```

from P23 P24 have P25 : ”Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An C A D1)) add Emp)” by (blast intro:Ang_weektrans)
from P23 have P26 : ”Def (Tri (Tr A C D1))” by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
from N have P27 : ”Def (Tri (Tr C A B))” by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
have P28 : ”Eq (Geos (Seg (Se A C)) add Emp) (Geos (Seg (Se C A)) add Emp)” by (simp add:Seg_rev)
from P6 have P29 : ”Eq (Geos (Seg (Se A D1)) add Emp) (Geos (Seg (Se C B)) add Emp)” by (simp add:Eq_rev)
from P29 P26 P27 P28 have
”Cong (Geos (Ang (An C A D1)) add Emp) (Geos (Ang (An A C B)) add Emp) ⟹
Cong (Geos (Tri (Tr A C D1)) add Emp) (Geos (Tri (Tr C A B)) add Emp)” by (simp add:Tri_SAS)
then have P29 : ”Cong (Geos (Ang (An C A D1)) add Emp) (Geos (Ang (An A C B)) add Emp) ⟹
Cong (Geos (Ang (An D1 C A)) add Emp) (Geos (Ang (An B A C)) add Emp)”
by (simp add:Tri_Cong_def)
from N have P30 : ”Def (Ang (An A C B))” by (blast intro:Tri_to_Ang Ang_def_inv)
from W P15 P23 P25 P30 have P31 : ”Cong (Geos (Ang (An C A D1)) add Emp) (Geos (Ang (An A C B)) add Emp)” by (blast intro:Ang_trans Ang_rev)
from P29 P31 have P32 : ”Cong (Geos (Ang (An D1 C A)) add Emp) (Geos (Ang (An B A C)) add Emp)” by simp
have P33 : ”Line_on (Li B C) B” by (simp add:Line_on_rule)
have P34 : ”Line_on (Li B C) C” by (simp add:Line_on_rule)
from P4 P33 P34 have ”∃p. Bet_Point (Se B p) C ∧ Line_on (Li B C) p” by (simp add:Bet_extension)
then obtain E :: Point where P35 : ”Bet_Point (Se B E) C ∧ Line_on (Li B C) E” by blast
have P36 : ”Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An D A C)) add Emp)” by (simp add:Ang_roll)
from P15 have P37 : ”Def (Ang (An D A C))” by (simp add:Ang_def_rev)
have P38 : ”Cong (Geos (Ang (An A C B)) add Emp) (Geos (Ang (An B C A)) add Emp)” by (simp add:Ang_roll)
from P30 have P39 : ”Def (Ang (An B C A))” by (simp add:Ang_def_rev)
from W P15 P30 P36 P37 P38 have P40 : ”Cong (Geos (Ang (An D A C)) add Emp) (Geos (Ang (An A C B)) add Emp)” by (blast intro:Ang_trans Ang_rev)
from P30 P37 P38 P39 P40 have P41 : ”Cong (Geos (Ang (An D A C)) add Emp) (Geos (Ang (An B C A)) add Emp)” by (blast intro:Ang_trans Ang_rev)
from N have P42 : ”Bet_Point (Se D B) A” by (simp add:Bet_rev)
from P35 P37 P39 P41 P42 have P43 : ”Cong (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An A C E)) add Emp)” by (simp add:Ang_complementary)
have P44 : ”Cong (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An C A B)) add Emp)” by (simp add:Ang_roll)
from P39 have P45 : ”Def (Ang (An B A C))” by (simp add:Ang_def_inv)
then have P46 : ”Def (Ang (An C A B))” by (simp add:Ang_def_rev)
from P23 have P47 : ”Def (Ang (An D1 C A))” by (blast intro:Ang_def_rev Ang_def_inv)
from P32 P44 P45 P46 P47 have P48 : ”Cong (Geos (Ang (An D1 C A)) add Emp) (Geos (Ang (An C A B)) add Emp)” by (blast intro:Ang_trans Ang_rev)
from P35 have P49 : ”Bet_Point (Se B E) C” by simp
then have P50 : ”¬ Eq (Geos (Poi E) add Emp) (Geos (Poi C) add Emp)” by (simp add:Bet_Point_def)
from P35 have P51 : ”Line_on (Li B C) E” by simp
from P16 P34 P50 P51 have P52 : ”Line_on (Li A C) E ⟹
Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li A C)) add Emp)” by (simp add:Line_unique)
from P33 P52 have P53 : ”Line_on (Li A C) E ⟹ Line_on (Li A C) B” by (simp add:Line_on_trans)
from N have ”Def (Tri (Tr A C B))” by (blast intro:Tri_def_rev Tri_def_trans)
then have P54 : ”¬ Line_on (Li A C) B” by (simp add:Tri_def_Line)
from P53 P54 have P55 : ”¬ Line_on (Li A C) E” by blast
from P22 P55 have P56 : ”Def (Ang (An A C E))” by (simp add:Ang_sinple_def)
from P43 P46 P47 P48 P56 have P57 : ”Cong (Geos (Ang (An D1 C A)) add Emp) (Geos (Ang (An A C E)) add Emp)” by (blast intro:Ang_trans Ang_rev)
then have P58 : ”Cong (Geos (Ang (An A C E)) add Emp) (Geos (Ang (An D1 C A)) add Emp)” by (simp add:Ang_rev)
have P59 : ”Cong (Geos (Ang (An A C E)) add Emp) (Geos (Ang (An E C A)) add Emp)” by (simp add:Ang_roll)
have P60 : ”Line_on (Li C A) A” by (simp add:Line_on_rule)
have ”Line_on (Li B D) B” by (simp add:Line_on_rule)
then have P61 : ”Eq (Geos (Lin (Li B D)) add Emp) (Geos (Lin (Li C A)) add Emp) ⟹ Line_on (Li C A) B” by (simp add:Line_on_trans)
from N have P62 : ”¬ Line_on (Li C A) B” by (simp add:Tri_def_Line)
from P61 P62 have P63 : ”¬ Eq (Geos (Lin (Li B D)) add Emp) (Geos (Lin (Li C A)) add Emp)” by blast
from N P60 P63 have P64 : ”Plane_diffside (Li C A) B D” by (simp add:Plane_Bet_diffside)

```
  then have P65 : "Eq (Geos (Poi D) add Emp) (Geos (Poi D1) add Emp) ⟹
    Plane_diffside (Li C A) B D1" by (simp add:Point_Eq)
  from P6 have P66 : "Line_on (Li B D) D1" by simp
  from P6 have P67 : "¬ Eq (Geos (Poi D1) add Emp) (Geos (Poi A) add Emp)" by (blast intro:Eq_rev)
  from P1 P2 P3 P66 P67 have P68 : "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi D1) add Emp) ⟹
    Bet_Point (Se D A) D1 ∨ Bet_Point (Se A D1) D ∨ Bet_Point (Se D1 D) A" by (simp add:Bet_case)
  from P19 have P69 : "Eq (Geos (Lin (Li A D)) add Emp) (Geos (Lin (Li C A)) add Emp) ⟹
    Eq (Geos (Lin (Li B D)) add Emp) (Geos (Lin (Li C A)) add Emp)" by (blast intro:Eq_trans)
  from P63 P69 have P70 : "¬ Eq (Geos (Lin (Li A D)) add Emp) (Geos (Lin (Li C A)) add Emp)" by
blast
  from P60 P70 have P71 : "Bet_Point (Se A D) D1 ⟹ Plane_sameside (Li C A) D D1" by (simp
add:Plane_Bet_sameside Plane_sameside_rev)
  have "Line_on (Li A D1) D1" by (simp add:Line_on_rule)
  then have P72 : "Eq (Geos (Lin (Li A D1)) add Emp) (Geos (Lin (Li C A)) add Emp) ⟹
    Line_on (Li C A) D1" by (simp add:Line_on_trans)
  from P23 have "Def (Tri (Tr C A D1))" by (simp add:Ang_to_Tri)
  then have P73 : "¬ Line_on (Li C A) D1" by (simp add:Tri_def_Line)
  from P72 P73 have P74 : "¬ Eq (Geos (Lin (Li A D1)) add Emp) (Geos (Lin (Li C A)) add Emp)" by
blast
  from P60 P74 have P75 : "Bet_Point (Se A D1) D ⟹ Plane_sameside (Li C A) D D1" by (simp
add:Plane_Bet_sameside)
  from P6 P64 P68 P71 P75 have P76 : "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi D1) add Emp) ⟹
    Plane_diffside (Li C A) B D1" by (blast intro:Plane_trans Plane_diffside_rev Bet_rev)
  from P65 P76 have P77 : "Plane_diffside (Li C A) B D1" by blast
  have P78 : "Line_on (Li C A) C" by (simp add:Line_on_rule)
  have "Line_on (Li B E) B" by (simp add:Line_on_rule)
  then have P79 : "Eq (Geos (Lin (Li B E)) add Emp) (Geos (Lin (Li C A)) add Emp) ⟹ Line_on (Li
C A) B" by (simp add:Line_on_trans)
  from P62 P79 have P80 : "¬ Eq (Geos (Lin (Li B E)) add Emp) (Geos (Lin (Li C A)) add Emp)" by
blast
  from P49 P78 P80 have P81 : "Plane_diffside (Li C A) B E" by (simp add:Plane_Bet_diffside)
  from P51 have P82 : "Eq (Geos (Poi D1) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li B C)
D1" by (blast intro:Eq_rev Point_Eq)
  from P77 have "∃p. Bet_Point (Se B D1) p ∧ Line_on (Li C A) p
    ∧ ¬ Line_on (Li C A) B ∧ ¬ Line_on (Li C A) D1" by (simp add:Plane_diffside_def)
  then obtain F :: Point where "Bet_Point (Se B D1) F" by blast
  then have P83 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D1) add Emp)" by (simp add:Bet_Point_def)
  from P8 P20 P33 P82 P83 have P84 : "Eq (Geos (Poi D1) add Emp) (Geos (Poi E) add Emp) ⟹
    Eq (Geos (Lin (Li A D)) add Emp) (Geos (Lin (Li B C)) add Emp)" by (simp add:Line_unique)
  from P7 P84 have P85 : "Eq (Geos (Poi D1) add Emp) (Geos (Poi E) add Emp) ⟹ Line_on (Li B C)
A" by (simp add:Line_on_trans)
  from N have P86 : "¬ Line_on (Li B C) A" by (simp add:Tri_def_Line)
  from P85 P86 have P87 : "¬ Eq (Geos (Poi D1) add Emp) (Geos (Poi E) add Emp)" by blast
  from P77 P81 P87 have P88 : "Plane_sameside (Li C A) D1 E" by (simp add:Plane_trans_inv)
  from P58 P59 P88 have P89 : "Eq (Geos (Lin (Li D1 C)) add Emp) (Geos (Lin (Li E C)) add Emp) ∧
¬ Bet_Point (Se D1 E) C" by (simp add:Ang_move_unique)
  from P49 have P90 : "Line_on (Li E C) B" by (simp add:Line_Bet_on)
  from P89 P90 have P91 : "Line_on (Li D1 C) B" by (blast intro:Eq_rev Line_on_trans)
  have P92 : "Line_on (Li D1 C) D1" by (simp add:Line_on_rule)
  from P8 P20 P83 P91 P92 have P93 : "Eq (Geos (Lin (Li A D)) add Emp) (Geos (Lin (Li D1 C)) add
Emp)" by (simp add:Line_unique)
  from P89 P93 have P94 : "Eq (Geos (Lin (Li A D)) add Emp) (Geos (Lin (Li E C)) add Emp)" by
(blast intro:Eq_trans)
  from P7 P94 have P95 : "Line_on (Li E C) A" by (simp add:Line_on_trans)
  from P56 have "Def (Tri (Tr E C A))" by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
  then have P96 : "¬ Line_on (Li E C) A" by (simp add:Tri_def_Line)
  from P95 P96 show False by blast
qed

lemma (in Congruence_Rule) Ang_external_Gr_lemma2 :
  assumes N :
    "Def (Tri (Tr A B C))"
    "Bet_Point (Se B D) A"
  shows "¬ Gr (Geos (Ang (An A C B)) add Emp) (Geos (Ang (An C A D)) add Emp)"
proof
  assume W : "Gr (Geos (Ang (An A C B)) add Emp) (Geos (Ang (An C A D)) add Emp)"
```

from N have "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi A) add Emp)" by (simp add:Bet_Point_def)
then have P1 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)" by (blast intro:Eq_rev)
have P2 : "Line_on (Li A D) A" by (simp add:Line_on_rule)
from N have P3 : "Line_on (Li A D) B" by (simp add:Line_Bet_on)
have P4 : "Line_on (Li A B) A" by (simp add:Line_on_rule)
have P5 : "Line_on (Li A B) B" by (simp add:Line_on_rule)
from N have P6 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)" by (simp add:Tri_def)
from P2 P3 P4 P5 P6 have "Eq (Geos (Lin (Li A D)) add Emp) (Geos (Lin (Li A B)) add Emp)" by (simp add:Line_unique)
then have P7 : "Line_on (Li A D) C ⟹ Line_on (Li A B) C" by (simp add:Line_on_trans)
from N have P8 : "¬ Line_on (Li A B) C" by (simp add:Tri_def_Line)
from P7 P8 have P9 : "¬ Line_on (Li A D) C" by blast
from P1 P9 have "Def (Ang (An A D C))" by (simp add:Ang_sinple_def)
then have P10 : "Def (Ang (An C A D))" by (blast intro:Ang_def_rev Ang_def_inv)
have P11 : "Cong (Geos (Ang (An A C B)) add Emp) (Geos (Ang (An B C A)) add Emp)" by (simp add:Ang_roll)
from N have P12 : "Def (Ang (An A C B))" by (blast intro:Tri_to_Ang Ang_def_inv)
then have P13 : "Def (Ang (An B C A))" by (simp add:Ang_def_rev)
from W P10 P11 P12 P13 have P14 : "Gr (Geos (Ang (An B C A)) add Emp) (Geos (Ang (An C A D)) add Emp)" by (blast intro:Ang_Gr_trans_Eq_Gr Ang_rev)
from N have P15 : "¬ Line_on (Li C A) B" by (simp add:Tri_def_Line)
from N have P16 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)" by (simp add:Tri_def)
from P10 P15 have "∃p. Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An p C A)) add Emp)
∧ Plane_sameside (Li C A) p B" by (simp add:Ang_move_sameside)
then obtain B1 :: Point where P17 : "Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An B1 C A)) add Emp)
∧ Plane_sameside (Li C A) B1 B" by blast
then have P18 : "¬ Line_on (Li C A) B1" by (simp add:Plane_sameside_def)
from P16 P18 have "Def (Ang (An C A B1))" by (blast intro:Ang_sinple_def Eq_rev)
then have P19 : "Def (Ang (An B1 C A))" by (blast intro:Ang_def_rev Ang_def_inv)
from N have P20 : "Def (Ang (An A C B))" by (blast intro:Tri_def_rev Tri_def_trans Tri_to_Ang)
from P17 have P21 : "Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An B1 C A)) add Emp)" by simp
from P17 have P22 : "Plane_sameside (Li C A) B1 B" by simp
from P10 P14 P19 P20 P21 P22 have "Ang_inside (An B C A) B1" by (simp add:Ang_greater_def)
then have "Plane_diffside (Li C B1) B A" by (simp add:Ang_inside_Planeside)
then have "∃p. Bet_Point (Se B A) p ∧ Line_on (Li C B1) p ∧ ¬ Line_on (Li C B1) B ∧ ¬ Line_on (Li C B1) A" by (simp add:Plane_diffside_def)
then obtain B2 :: Point where P23 : "Bet_Point (Se B A) B2 ∧ Line_on (Li C B1) B2
∧ ¬ Line_on (Li C B1) B ∧ ¬ Line_on (Li C B1) A" by blast
then have "Line_on (Li B A) B2" by (simp add:Line_Bet_on)
then have P24 : "Eq (Geos (Poi B2) add Emp) (Geos (Poi C) add Emp) ⟹ Line_on (Li B A) C" by (simp add:Point_Eq)
from N have "Def (Tri (Tr B A C))" by (blast intro:Tri_def_rev Tri_def_trans)
then have P25 : "¬ Line_on (Li B A) C" by (simp add:Tri_def_Line)
from P24 P25 have P26 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi B2) add Emp)" by (blast intro:Eq_rev)
from P23 have P27 : "Bet_Point (Se A B) B2" by (simp add:Bet_rev)
have P28 : "Line_on (Li C A) A" by (simp add:Line_on_rule)
from P5 have P29 : "Eq (Geos (Lin (Li A B)) add Emp) (Geos (Lin (Li C A)) add Emp) ⟹
Line_on (Li C A) B" by (simp add: Line_on_trans)
from P15 P29 have P30 : "¬ Eq (Geos (Lin (Li A B)) add Emp) (Geos (Lin (Li C A)) add Emp)" by blast
from P27 P28 P30 have P31 : "Plane_sameside (Li C A) B B2" by (simp add:Plane_Bet_sameside Plane_sameside_rev)
have P32 : "Line_on (Li C A) C" by (simp add:Line_on_rule)
have "Line_on (Li B1 B2) B1" by (simp add:Line_on_rule)
then have P33 : "Eq (Geos (Lin (Li B1 B2)) add Emp) (Geos (Lin (Li C A)) add Emp) ⟹
Line_on (Li C A) B1" by (simp add: Line_on_trans)
from P18 P33 have P34 : "¬ Eq (Geos (Lin (Li B1 B2)) add Emp) (Geos (Lin (Li C A)) add Emp)" by blast
from P32 P34 have "Bet_Point (Se B1 B2) C ⟹ Plane_diffside (Li C A) B1 B2"
by (simp add:Plane_Bet_diffside)
then have P35 : "Bet_Point (Se B1 B2) C ⟹ ¬ Plane_sameside (Li C A) B1 B2"
by (simp add:Plane_diffside_not_sameside)
from P22 P31 have P36 : "¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi B2) add Emp) ⟹

```
    Plane_sameside (Li C A) B1 B2" by (blast intro:Plane_sameside_trans Eq_rev)
  from P35 P36 have P37 : "¬ Eq (Geos (Poi B1) add Emp) (Geos (Poi B2) add Emp) ⟹
    ¬ Bet_Point (Se B1 B2) C" by blast
  have P38 : "Eq (Geos (Poi B1) add Emp) (Geos (Poi B2) add Emp) ⟹
    Bet_Point (Se B1 B2) C ⟹ Bet_Point (Se B2 B2) C" by (simp add:Bet_Point_Eq)
  have P39 : "¬ Bet_Point (Se B2 B2) C" by (simp add:Bet_end_Point)
  from P38 P39 have P40 : "Eq (Geos (Poi B1) add Emp) (Geos (Poi B2) add Emp) ⟹
    ¬ Bet_Point (Se B1 B2) C" by blast
  from P37 P40 have P41 : "¬ Bet_Point (Se B1 B2) C" by blast
  have P42 : "¬ Bet_Point (Se A A) C" by (simp add:Bet_end_Point)
  from P16 P19 P23 P26 P28 P41 P42 have P43 :
    "Eq (Geos (Ang (An B1 C A)) add Emp) (Geos (Ang (An B2 C A)) add Emp) ∧ Def (Ang (An B2
C A))" by (simp add:Ang_Point_swap)
  from P21 P43 have P44 : "Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An B2 C A)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from P43 have P45 : "Def (Tri (Tr A B2 C))" by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
  from P27 have P46 : "Bet_Point (Se B A) B2" by (simp add:Bet_rev)
  from N P46 have P47 : "Bet_Point (Se B2 D) A" by (blast intro:Bet_swap_134_234)
  from P45 P47 have P48 : "¬ Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An A C B2)) add
Emp)" by (simp add:Ang_external_Gr_lemma1)
  have P49 : "Cong (Geos (Ang (An B2 C A)) add Emp) (Geos (Ang (An A C B2)) add Emp)" by (simp
add:Ang_roll)
  from P43 have P50 : "Def (Ang (An A C B2))" by (simp add:Ang_def_rev)
  from P10 P43 P44 P49 P50 have P51 : "Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An A
C B2)) add Emp)" by (blast intro:Ang_trans Ang_rev)
  from P48 P51 show False by blast
qed

theorem (in Congruence_Rule) Ang_external_Gr :
  assumes
    "Def (Tri (Tr A B C))"
    "Bet_Point (Se B D) A"
  shows
    "Gr (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An A C B)) add Emp)"
    "Gr (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An A B C)) add Emp)"
proof -
  from assms have "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi A) add Emp)" by (simp add:Bet_Point_def)
  then have P1 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)" by (blast intro:Eq_rev)
  have P2 : "Line_on (Li A D) A" by (simp add:Line_on_rule)
  from assms have P3 : "Line_on (Li A D) B" by (simp add:Line_Bet_on)
  have P4 : "Line_on (Li A B) A" by (simp add:Line_on_rule)
  have P5 : "Line_on (Li A B) B" by (simp add:Line_on_rule)
  from assms have P6 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)" by (simp add:Tri_def)
  from P2 P3 P4 P5 P6 have "Eq (Geos (Lin (Li A D)) add Emp) (Geos (Lin (Li A B)) add Emp)" by
(simp add:Line_unique)
  then have P7 : "Line_on (Li A D) C ⟹ Line_on (Li A B) C" by (simp add:Line_on_trans)
  from assms have P8 : "¬ Line_on (Li A B) C" by (simp add:Tri_def_Line)
  from P7 P8 have P9 : "¬ Line_on (Li A D) C" by blast
  from P1 P9 have "Def (Ang (An A D C))" by (simp add:Ang_sinple_def)
  then have P10 : "Def (Ang (An C A D))" by (blast intro:Ang_def_rev Ang_def_inv)
  from assms have P11 : "Def (Ang (An A C B))" by (simp add:Tri_to_Ang Ang_def_inv)
  from P10 P11 have P12 :
    "Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An A C B)) add Emp)
    ∨ Gr (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An A C B)) add Emp)
    ∨ Gr (Geos (Ang (An A C B)) add Emp) (Geos (Ang (An C A D)) add Emp)"
    by (simp add:Ang_relation_case)
  from assms have P13 : "¬ Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An A C B)) add
Emp)" by (simp add:Ang_external_Gr_lemma1)
  from assms have P14 : "¬ Gr (Geos (Ang (An A C B)) add Emp) (Geos (Ang (An C A D)) add Emp)"
by (simp add:Ang_external_Gr_lemma2)
  from P12 P13 P14 show P15 : "Gr (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An A C B)) add
Emp)" by blast
  have P16 : "Line_on (Li C A) C" by (simp add:Line_on_rule)
  have P17 : "Line_on (Li C A) A" by (simp add:Line_on_rule)
  from assms have P18 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)" by (simp add:Tri_def)
  from P16 P17 P18 have "∃p. Bet_Point (Se C p) A ∧ Line_on (Li C A) p" by (simp add:Bet_extension)
```

```
  then obtain E :: Point where P19 : "Bet_Point (Se C E) A" by blast
  from assms have P20 : "Bet_Point (Se D B) A" by (simp add:Bet_rev)
  from P10 P19 P20 have P21 : "Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An E A B)) add Emp)" by (simp add:Ang_vertical)
  have P22 : "Eq (Geos (Ang (An E A B)) add Emp) (Geos (Ang (An B A E)) add Emp)" by (simp add:Ang_roll)
  from P21 P22 have P23 : "Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An B A E)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from P19 have P24 : "Line_on (Li C E) A" by (simp add:Line_Bet_on)
  have P25 : "Line_on (Li C E) E" by (simp add:Line_on_rule)
  from P19 have P26 : "¬ Eq (Geos (Poi E) add Emp) (Geos (Poi A) add Emp)"
    by (simp add:Bet_Point_def)
  from P4 P24 P25 P26 have P27 : "Line_on (Li A B) E ⟹
    Eq (Geos (Lin (Li C E)) add Emp) (Geos (Lin (Li A B)) add Emp)" by (simp add:Line_unique)
  have P28 : "Line_on (Li C E) C" by (simp add:Line_on_rule)
  from P27 P28 have P29 : "Line_on (Li A B) E ⟹ Line_on (Li A B) C" by (simp add:Line_on_trans)
  from P8 P29 have P30 : "¬ Line_on (Li A B) E" by blast
  from P6 P30 have "Def (Ang (An A B E))" by (simp add:Ang_sinple_def)
  then have P31 : "Def (Ang (An B A E))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P11 have P32 : "Def (Tri (Tr A C B))" by (simp add:Ang_to_Tri)
  from P19 P32 have P33 : "¬ Cong (Geos (Ang (An B A E)) add Emp) (Geos (Ang (An A B C)) add Emp)" by (simp add:Ang_external_Gr_lemma1)
  from P19 P32 have P34 : "¬ Gr (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An B A E)) add Emp)" by (simp add:Ang_external_Gr_lemma2)
  from assms have P35 : "Def (Ang (An A B C))" by (simp add:Tri_to_Ang)
  from P31 P35 have P36 :
    "Cong (Geos (Ang (An B A E)) add Emp) (Geos (Ang (An A B C)) add Emp)
    ∨ Gr (Geos (Ang (An B A E)) add Emp) (Geos (Ang (An A B C)) add Emp)
    ∨ Gr (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An B A E)) add Emp)"
    by (simp add:Ang_relation_case)
  from P33 P34 P36 have P37 : "Gr (Geos (Ang (An B A E)) add Emp) (Geos (Ang (An A B C)) add Emp)" by blast
  from P10 P23 P31 P35 P37 show "Gr (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An A B C)) add Emp)" by (blast intro:Ang_Gr_trans_Eq_Gr)
qed

lemma (in Congruence_Rule) Seg_not_Eq_move :
  assumes
    "¬ Eq (Geos (Poi A1) add Emp) (Geos (Poi B1) add Emp)"
    "¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi B2) add Emp)"
    "¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi B3) add Emp)"
    "Line_on l1 A2" "Line_on l1 B2" "Line_on l1 B3"
    "¬ Bet_Point (Se B3 B2) A2"
    "Eq (Geos (Seg (Se A1 B1)) add Emp) (Geos (Seg (Se A2 B3)) add Emp)"
    "¬ Eq (Geos (Seg (Se A1 B1)) add Emp) (Geos (Seg (Se A2 B2)) add Emp)"
  shows
    "Bet_Point (Se B2 A2) B3 ∧ ¬ Bet_Point (Se A2 B3) B2
    ∨ ¬ Bet_Point (Se B2 A2) B3 ∧ Bet_Point (Se A2 B3) B2"
proof -
  from assms have P1 : "Eq (Geos (Seg (Se A2 B2)) add Emp) (Geos (Seg (Se A2 B3)) add Emp) ⟹
    Eq (Geos (Seg (Se A1 B1)) add Emp) (Geos (Seg (Se A2 B2)) add Emp)" by (blast intro:Eq_trans Eq_rev)
  from assms P1 have "¬ Eq (Geos (Seg (Se A2 B2)) add Emp) (Geos (Seg (Se A2 B3)) add Emp)" by blast
  then have P2 : "¬ Eq (Geos (Poi B2) add Emp) (Geos (Poi B3) add Emp)"
    by (simp add:Seg_not_Eq_Point)
  from assms have P3 : "¬ Eq (Geos (Poi B3) add Emp) (Geos (Poi A2) add Emp)" by (blast intro:Eq_rev)
  from assms have P4 : "Line_on l1 A2" by simp
  from assms have P5 : "Line_on l1 B2" by simp
  from assms have P6 : "Line_on l1 B3" by simp
  from assms have P7 : "¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi B2) add Emp)" by simp
  from P2 P3 P4 P5 P6 P7 have "Bet_Point (Se A2 B3) B2 ∨ Bet_Point (Se B3 B2) A2 ∨ Bet_Point (Se B2 A2) B3" by (simp add:Bet_case)
  then have P8 :
    "Bet_Point (Se A2 B3) B2 ∧ ¬ Bet_Point (Se B3 B2) A2 ∧ ¬ Bet_Point (Se B2 A2) B3
```

```
    ∨ ¬ Bet_Point (Se A2 B3) B2 ∧ Bet_Point (Se B3 B2) A2 ∧ ¬ Bet_Point (Se B2 A2) B3
    ∨ ¬ Bet_Point (Se A2 B3) B2 ∧ ¬ Bet_Point (Se B3 B2) A2 ∧ Bet_Point (Se B2 A2) B3" by (simp
add:Bet_case_fact)
  from assms P8 show "Bet_Point (Se B2 A2) B3 ∧ ¬ Bet_Point (Se A2 B3) B2
    ∨ ¬ Bet_Point (Se B2 A2) B3 ∧ Bet_Point (Se A2 B3) B2" by blast
qed

lemma (in Congruence_Rule) Tri_Seg_diagonal :
  assumes
    "Def (Tri (Tr A B C))"
    "Bet_Point (Se B C) D"
    "Eq (Geos (Seg (Se A C)) add Emp) (Geos (Seg (Se C D)) add Emp)"
  shows
    "Gr (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An A B C)) add Emp)"
proof -
  from assms have P1 : "¬ Line_on (Li A B) C" by (simp add:Tri_def_Line)
  have "Line_on (Li A C) C" by (simp add:Line_on_rule)
  then have P2 : "Eq (Geos (Lin (Li A C)) add Emp) (Geos (Lin (Li A B)) add Emp) ⟹
    Line_on (Li A B) C" by (simp add:Line_on_trans)
  from P1 P2 have P3 : "¬ Eq (Geos (Lin (Li A B)) add Emp) (Geos (Lin (Li A C)) add Emp)" by (blast
intro:Eq_rev)
  from assms have P4 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)" by (simp add:Tri_def)
  from assms have "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)" by (simp add:Tri_def)
  then have P5 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)" by (blast intro:Eq_rev)
  from assms P3 P4 P5 have P6 : "Ang_inside (An B A C) D" by (simp add:Ang_inside_Bet_Point)
  then have "Plane_sameside (Li A C) B D" by (simp add:Ang_inside_def)
  then have P7 : "Plane_sameside (Li A C) D B" by (simp add:Plane_sameside_rev)
  have P8 : "Cong (Geos (Ang (An D A C)) add Emp) (Geos (Ang (An D A C)) add Emp)" by simp
  from P6 P7 P8 have P9 : "Gr (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An D A C)) add Emp)"
by (simp add:Ang_greater_def)
  from assms have P10 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi D) add Emp)"
    by (simp add:Bet_Point_def)
  from assms have P11 : "Line_on (Li C B) D" by (simp add:Line_Bet_on)
  from assms have P12 : "Def (Tri (Tr A C B))" by (blast intro:Tri_def_rev Tri_def_trans)
  from P10 P11 P12 have "Def (Tri (Tr A C D))" by (simp add:Tri_def_extension)
  then have P13 : "Def (Tri (Tr C A D))" by (blast intro:Tri_def_rev Tri_def_trans)
  have P14 : "Eq (Geos (Seg (Se A C)) add Emp) (Geos (Seg (Se C A)) add Emp)" by (simp add:Seg_rev)
  from assms P14 have P15 : "Eq (Geos (Seg (Se C A)) add Emp) (Geos (Seg (Se C D)) add Emp)" by
(blast intro:Eq_rev Eq_trans)
  from P13 P15 have P16 : "Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An C D A)) add
Emp)" by (simp add:Tri_isosceles)
  from assms have "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi B) add Emp)" by (simp add:Bet_Point_def)
  then have P17 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)" by (blast intro:Eq_rev)
  from assms have P18 : "Line_on (Li B C) D" by (simp add:Line_Bet_on)
  from assms P17 P18 have "Def (Tri (Tr A B D))" by (simp add:Tri_def_extension)
  then have P19 : "Def (Tri (Tr D B A))" by (simp add:Tri_def_rev)
  from assms P19 have P20 : "Gr (Geos (Ang (An A D C)) add Emp) (Geos (Ang (An D B A)) add
Emp)" by (simp add:Ang_external_Gr)
  from assms have P21 : "Def (Ang (An A B C))" by (simp add:Tri_to_Ang)
  have P22 : "Line_on (Li B A) A" by (simp add:Line_on_rule)
  have P23 : "¬ Bet_Point (Se A A) B" by (simp add:Bet_end_Point)
  from assms have P24 : "Line_on (Li B C) D" by (simp add:Line_Bet_on)
  from assms have "Inv (Bet_Point (Se C D) B)" by (simp add:Bet_iff)
  then have P25 : "¬ Bet_Point (Se C D) B" by (simp add:Inv_def)
  from P4 have P26 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp)" by (blast intro:Eq_rev)
  from P17 P21 P22 P23 P24 P25 P26 have P27 :
      "Eq (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A B D)) add Emp) ∧ Def (Ang (An A
B D))" by (simp add:Ang_Point_swap)
  have P28 : "Cong (Geos (Ang (An D B A)) add Emp) (Geos (Ang (An A B D)) add Emp)" by (simp
add:Ang_roll)
  from P27 P28 have P29 : "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An D B A)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev)
  have P30 : "Eq (Geos (Ang (An C D A)) add Emp) (Geos (Ang (An A D C)) add Emp)" by (simp
add:Ang_roll)
  from P16 P30 have P31 : "Cong (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An A D C)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
```

```
  from P13 have P32 : "Def (Ang (An C A D))" by (simp add:Tri_to_Ang)
  then have P33 : "Def (Ang (An A D C))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P27 have P34 : "Def (Ang (An D B A))" by (blast intro:Ang_def_rev)
  from P20 P31 P32 P33 P34 have P35 : "Gr (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An D B
A)) add Emp)" by (blast intro:Ang_Gr_trans_Eq_Gr)
  from P21 P29 P32 P34 P35 have P36 : "Gr (Geos (Ang (An C A D)) add Emp) (Geos (Ang (An A B
C)) add Emp)" by (blast intro:Ang_Gr_trans_Gr_Eq Ang_rev)
  have P37 : "Cong (Geos (Ang (An D A C)) add Emp) (Geos (Ang (An C A D)) add Emp)" by (simp
add:Ang_roll)
  from P32 have P38 : "Def (Ang (An D A C))" by (simp add:Ang_def_rev)
  from assms have P39 : "Def (Ang (An B A C))" by (blast intro:Tri_to_Ang Ang_def_rev Ang_def_inv)
  from P9 P32 P37 P38 P39 have P40 : "Gr (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An C A
D)) add Emp)" by (blast intro:Ang_Gr_trans_Gr_Eq)
  from P21 P32 P36 P39 P40 show "Gr (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An A B C))
add Emp)" by (blast intro:Ang_Gr_trans_Gr_Gr)
qed

lemma (in Congruence_Rule) Tri_Bet_Ang_Gr :
  assumes
    "Def (Tri (Tr A B C))"
    "Bet_Point (Se A C) D"
    "Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se A C)) add Emp)"
  shows
    "Gr (Geos (Ang (An A D B)) add Emp) (Geos (Ang (An A B D)) add Emp)"
proof -
  from assms have P1 : "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A C B)) add Emp)"
by (simp add:Tri_isosceles)
  from assms have P2 : "Def (Tri (Tr B C A))" by (simp add:Tri_def_trans)
  from assms have P3 : "Line_on (Li C A) D" by (simp add:Line_Bet_on)
  from assms have P4 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi D) add Emp)" by (simp add:Bet_Point_def)
  from P2 P3 P4 have "Def (Tri (Tr B C D))" by (simp add:Tri_def_extension)
  then have P5 : "Def (Tri (Tr D C B))" by (blast intro:Tri_def_rev Tri_def_trans)
  from assms have P6 : "Bet_Point (Se C A) D" by (simp add:Bet_rev)
  from P5 P6 have P7 : "Gr (Geos (Ang (An B D A)) add Emp) (Geos (Ang (An D C B)) add Emp)" by
(simp add:Ang_external_Gr)
  from P5 have P8 : "Def (Ang (An D C B))" by (simp add:Tri_to_Ang)
  from assms have P9 : "Line_on (Li C D) A" by (simp add:Line_Bet_on)
  from assms have "Inv (Bet_Point (Se D A) C)" by (simp add:Bet_iff)
  then have P10 : "¬ Bet_Point (Se D A) C" by (simp add:Inv_def)
  have P11 : "Line_on (Li C B) B" by (simp add:Line_on_rule)
  have P12 : "¬ Bet_Point (Se B B) C" by (simp add:Bet_end_Point)
  from P6 have P13 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)" by (simp add:Bet_Point_def)
  from P5 have P14 : "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi B) add Emp)" by (simp add:Tri_def)
  from P8 P9 P10 P11 P12 P13 P14 have P15 : "Eq (Geos (Ang (An D C B)) add Emp) (Geos (Ang
(An A C B)) add Emp)
    ∧ Def (Ang (An A C B))" by (simp add:Ang_Point_swap)
  from P1 P15 have P16 : "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An D C B)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev)
  have P17 : "Cong (Geos (Ang (An B D A)) add Emp) (Geos (Ang (An A D B)) add Emp)" by (simp
add:Ang_roll)
  from P2 have P18 : "Def (Tri (Tr B A C))" by (blast intro:Tri_def_rev Tri_def_trans)
  from assms have P19 : "Line_on (Li A C) D" by (simp add:Line_Bet_on)
  from P6 have P20 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)" by (simp add:Bet_Point_def)
  from P18 P19 P20 have "Def (Tri (Tr B A D))" by (simp add:Tri_def_extension)
  then have P21 : "Def (Ang (An B D A))" by (blast intro:Tri_to_Ang Ang_def_inv)
  then have P22 : "Def (Ang (An A D B))" by (simp add:Ang_def_rev)
  from P7 P8 P17 P21 P22 have P23 : "Gr (Geos (Ang (An A D B)) add Emp) (Geos (Ang (An D C B))
add Emp)" by (blast intro:Ang_Gr_trans_Eq_Gr Ang_rev)
  from assms have P24 : "Def (Ang (An A B C))" by (simp add:Tri_to_Ang)
  from P8 P16 P22 P23 P24 have P25 : "Gr (Geos (Ang (An A D B)) add Emp) (Geos (Ang (An A B
C)) add Emp)" by (blast intro:Ang_Gr_trans_Gr_Eq Ang_rev)
  from P24 have "¬ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp)" by (simp
add:Ang_def)
  then have P26 : "¬ Eq (Geos (Lin (Li B C)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (blast
intro:Eq_rev)
  from assms have "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)" by (simp add:Tri_def)
```

```
  then have P27 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp)" by (blast intro:Eq_rev)
  from assms have P28 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)" by (simp add:Tri_def)
  from P6 P26 P27 P28 have P29 : "Ang_inside (An C B A) D" by (simp add:Ang_inside_Bet_Point)
  then have P30 : "Plane_sameside (Li B A) D C" by (simp add:Ang_inside_def Plane_sameside_rev)
  have P31 : "Cong (Geos (Ang (An D B A)) add Emp) (Geos (Ang (An D B A)) add Emp)" by simp
  from P29 P30 P31 have P32 : "Gr (Geos (Ang (An C B A)) add Emp) (Geos (Ang (An D B A)) add
Emp)" by (simp add:Ang_greater_def)
  have P33 : "Cong (Geos (Ang (An C B A)) add Emp) (Geos (Ang (An A B C)) add Emp)" by (simp
add:Ang_roll)
  from P24 have P34 : "Def (Ang (An C B A))" by (simp add:Ang_def_rev)
  from P22 have P35 : "Def (Ang (An D B A))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P24 P32 P33 P34 P35 have P36 : "Gr (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An D B
A)) add Emp)" by (blast intro:Ang_Gr_trans_Eq_Gr Ang_rev)
  have P37 : "Cong (Geos (Ang (An D B A)) add Emp) (Geos (Ang (An A B D)) add Emp)" by (simp
add:Ang_roll)
  from P35 have P38 : "Def (Ang (An A B D))" by (simp add:Ang_def_rev)
  from P24 P35 P36 P37 P38 have P39 : "Gr (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A B
D)) add Emp)" by (blast intro:Ang_Gr_trans_Gr_Eq Ang_rev)
  from P22 P24 P25 P38 P39 show "Gr (Geos (Ang (An A D B)) add Emp) (Geos (Ang (An A B D))
add Emp)" by (blast intro:Ang_Gr_trans_Gr_Gr Ang_rev)
qed

theorem (in Congruence_Rule) Tri_isosceles_inv :
  assumes N :
    "Def (Tri (Tr A B C))"
    "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A C B)) add Emp)"
  shows
    "¬¬ Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se A C)) add Emp)"
proof
  assume W : "¬ Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se A C)) add Emp)"
  have P1 : "Line_on (Li A C) A" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li A C) C" by (simp add:Line_on_rule)
  from N have P3 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)" by (simp add:Tri_def)
  from N have "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)" by (simp add:Tri_def)
  then have P4 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)" by (blast intro:Eq_rev)
  from P1 P2 P3 P4 have "∃p. Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se A p)) add Emp)
    ∧ ¬ Bet_Point (Se p C) A ∧ Line_on (Li A C) p ∧ ¬ Eq (Geos (Poi A) add Emp) (Geos (Poi p) add
Emp)" by (simp add:Seg_move_sameside)
  then obtain D :: Point where P5 : "Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se A D)) add Emp)
    ∧ ¬ Bet_Point (Se D C) A ∧ Line_on (Li A C) D ∧ ¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D)
add Emp)" by blast
  from W P1 P2 P3 P4 P5 have P6 : "Bet_Point (Se C A) D ∧ ¬ Bet_Point (Se A D) C
    ∨ ¬ Bet_Point (Se C A) D ∧ Bet_Point (Se A D) C" by (simp add:Seg_not_Eq_move)
  from N have P7 : "Def (Tri (Tr B C A))" by (blast intro:Tri_def_rev Tri_def_trans)
  have P8 : "Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se B A)) add Emp)" by (simp add:Seg_rev)
  from P5 P8 have P9 : "Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se A D)) add Emp)" by (blast
intro:Eq_trans Eq_rev)
  from P7 P9 have P10 : "Bet_Point (Se C A) D ⟹
    Gr (Geos (Ang (An C B A)) add Emp) (Geos (Ang (An B C A)) add Emp)"
    by (simp add:Tri_Seg_diagonal)
  have P11 : "Cong (Geos (Ang (An C B A)) add Emp) (Geos (Ang (An A B C)) add Emp)" by (simp
add:Ang_roll)
  from assms have P12 : "Def (Ang (An A B C))" by (simp add:Tri_to_Ang)
  then have P13 : "Def (Ang (An C B A))" by (simp add:Ang_def_rev)
  then have P14 : "Def (Ang (An B C A))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P10 P11 P12 P13 P14 have P15 : "Bet_Point (Se C A) D ⟹
    Gr (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An B C A)) add Emp)"
    by (blast intro:Ang_Gr_trans_Eq_Gr Ang_rev)
  from P14 have P16 : "Def (Ang (An A C B))" by (simp add:Ang_def_rev)
  have P17 : "Cong (Geos (Ang (An B C A)) add Emp) (Geos (Ang (An A C B)) add Emp)" by (simp
add:Ang_roll)
  from P12 P14 P15 P16 P17 have P18 : "Bet_Point (Se C A) D ⟹
    Gr (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A C B)) add Emp)"
    by (blast intro:Ang_Gr_trans_Gr_Eq Ang_rev)
  from P12 P16 have P19 : "Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A C B)) add
Emp)
```

```
    ∧ ¬ Gr (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A C B)) add Emp)
    ∧ ¬ Gr (Geos (Ang (An A C B)) add Emp) (Geos (Ang (An A B C)) add Emp)
    ∨ ¬ Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A C B)) add Emp)
    ∧ Gr (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A C B)) add Emp)
    ∧ ¬ Gr (Geos (Ang (An A C B)) add Emp) (Geos (Ang (An A B C)) add Emp)
    ∨ ¬ Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A C B)) add Emp)
    ∧ ¬ Gr (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A C B)) add Emp)
    ∧ Gr (Geos (Ang (An A C B)) add Emp) (Geos (Ang (An A B C)) add Emp)"
    by (simp add:Ang_relation_case_fact)
  from P18 P19 have P20 : "Bet_Point (Se C A) D ⟹
    ¬ Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A C B)) add Emp)" by blast
  from assms have P21 : "Def (Tri (Tr B A C))" by (blast intro:Tri_def_rev Tri_def_trans)
  from P5 P21 have "Def (Tri (Tr B A D))" by (simp add:Tri_def_extension)
  then have P22 : "Def (Tri (Tr A B D))" by (blast intro:Tri_def_rev Tri_def_trans)
  from P5 P22 have P23 : "Bet_Point (Se A D) C ⟹
    Gr (Geos (Ang (An A C B)) add Emp) (Geos (Ang (An A B C)) add Emp)"
    by (simp add:Tri_Bet_Ang_Gr)
  from P19 P23 have P24 : "Bet_Point (Se A D) C ⟹
    ¬ Cong (Geos (Ang (An A B C)) add Emp) (Geos (Ang (An A C B)) add Emp)" by blast
  from assms P6 P20 P24 show False by blast
qed

lemma (in Congruence_Rule) Tri_AAS_lemma1 :
  assumes
    "Def (Tri (Tr A1 B1 C1))" "Def (Tri (Tr A2 B2 C2))"
    "Eq (Geos (Seg (Se A1 B1)) add Emp) (Geos (Seg (Se A2 B2)) add Emp)"
    "Cong (Geos (Ang (An A1 C1 B1)) add Emp) (Geos (Ang (An A2 C2 B2)) add Emp)"
    "Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An B2 A2 C2)) add Emp)"
  shows
    "Cong (Geos (Tri (Tr A1 B1 C1)) add Emp) (Geos (Tri (Tr A2 B2 C2)) add Emp)"
proof -
  have P1 : "Line_on (Li A2 C2) A2" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li A2 C2) C2" by (simp add:Line_on_rule)
  from assms have "¬ Eq (Geos (Poi C1) add Emp) (Geos (Poi A1) add Emp)" by (simp add:Tri_def)
  then have P3 : "¬ Eq (Geos (Poi A1) add Emp) (Geos (Poi C1) add Emp)" by (blast intro:Eq_rev)
  from assms have P4 : "¬ Eq (Geos (Poi C2) add Emp) (Geos (Poi A2) add Emp)" by (simp add:Tri_def)
  then have P5 : "¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi C2) add Emp)" by (blast intro:Eq_rev)
  from P1 P2 P3 P5 have "∃p. Eq (Geos (Seg (Se A1 C1)) add Emp) (Geos (Seg (Se A2 p)) add Emp)
    ∧ ¬ Bet_Point (Se p C2) A2 ∧ Line_on (Li A2 C2) p ∧ ¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi
p) add Emp)" by (simp add:Seg_move_sameside)
  then obtain C3 :: Point where P6 : "Eq (Geos (Seg (Se A1 C1)) add Emp) (Geos (Seg (Se A2 C3))
add Emp)
    ∧ ¬ Bet_Point (Se C3 C2) A2 ∧ Line_on (Li A2 C2) C3 ∧ ¬ Eq (Geos (Poi A2) add Emp) (Geos
(Poi C3) add Emp)" by blast
  from P6 have P7 : "Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
    Eq (Geos (Seg (Se A1 C1)) add Emp) (Geos (Seg (Se A2 C2)) add Emp)" by (blast intro:Eq_rev
Eq_trans)
  from assms P7 have P8 : "Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp)
⟹
    Cong (Geos (Tri (Tr A1 B1 C1)) add Emp) (Geos (Tri (Tr A2 B2 C2)) add Emp)" by (simp
add:Tri_SAS)
  from P6 have "Eq (Geos (Seg (Se A1 C1)) add Emp) (Geos (Seg (Se A2 C2)) add Emp) ⟹
    Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp)" by (blast intro:Eq_rev
Eq_trans)
  then have P9 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
    ¬ Eq (Geos (Seg (Se A1 C1)) add Emp) (Geos (Seg (Se A2 C2)) add Emp)" by blast
  from P1 P2 P3 P5 P6 P9 have P10 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3))
add Emp) ⟹
    Bet_Point (Se C2 A2) C3 ∧ ¬ Bet_Point (Se A2 C3) C2
    ∨ ¬ Bet_Point (Se C2 A2) C3 ∧ Bet_Point (Se A2 C3) C2" by (simp add:Seg_not_Eq_move)
  from assms have P11 : "Def (Tri (Tr B2 A2 C2))" by (blast intro:Tri_def_rev Tri_def_trans)
  from P6 P11 have "Def (Tri (Tr B2 A2 C3))" by (simp add:Tri_def_extension)
  then have P12 : "Def (Tri (Tr A2 B2 C3))" by (blast intro:Tri_def_rev Tri_def_trans)
  from P11 have P13 : "Def (Ang (An B2 A2 C2))" by (simp add:Tri_to_Ang)
  have P14 : "Line_on (Li A2 B2) B2" by (simp add:Line_on_rule)
  have P15 : "¬ Bet_Point (Se B2 B2) A2" by (simp add:Bet_end_Point)
```

from P6 have P16 : "¬ Bet_Point (Se C2 C3) A2" by (blast intro:Bet_rev)
from assms have P17 : "¬ Eq (Geos (Poi A2) add Emp) (Geos (Poi B2) add Emp)" by (simp add:Tri_def)
from P6 P13 P14 P15 P16 P17 have P18 :
"Eq (Geos (Ang (An B2 A2 C2)) add Emp) (Geos (Ang (An B2 A2 C3)) add Emp) ∧ Def (Ang (An B2 A2 C3))" by (simp add:Ang_Point_swap)
from assms P18 have P19 : "Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An B2 A2 C3)) add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from assms P6 P12 P19 have "Cong (Geos (Tri (Tr A1 B1 C1)) add Emp) (Geos (Tri (Tr A2 B2 C3)) add Emp)" by (simp add:Tri_SAS)
then have P20 : "Cong (Geos (Ang (An A1 C1 B1)) add Emp) (Geos (Ang (An A2 C3 B2)) add Emp)" by (simp add:Tri_Cong_def)
from assms have P21 : "Def (Ang (An A1 C1 B1))" by (blast intro:Tri_to_Ang Ang_def_inv)
from assms have P22 : "Def (Ang (An A2 C2 B2))" by (blast intro:Tri_to_Ang Ang_def_inv)
from P18 have P23 : "Def (Ang (An A2 C3 B2))" by (blast intro:Ang_def_rev Ang_def_inv)
from assms P20 P21 P22 P23 have P24 : "Cong (Geos (Ang (An A2 C2 B2)) add Emp) (Geos (Ang (An A2 C3 B2)) add Emp)" by (blast intro:Ang_trans Ang_rev)
from P22 P23 P24 have P25 : "¬ Gr (Geos (Ang (An A2 C2 B2)) add Emp) (Geos (Ang (An A2 C3 B2)) add Emp)
∧ ¬ Gr (Geos (Ang (An A2 C3 B2)) add Emp) (Geos (Ang (An A2 C2 B2)) add Emp)" by (simp add:Ang_not_Gr)
from assms have P26 : "Def (Tri (Tr B2 C2 A2))" by (blast intro:Tri_def_trans)
have P27 : "Bet_Point (Se C2 A2) C3 ⟹ ¬ Eq (Geos (Poi C3) add Emp) (Geos (Poi C2) add Emp)" by (simp add:Bet_Point_def)
have P28 : "Bet_Point (Se A2 C3) C2 ⟹ ¬ Eq (Geos (Poi C3) add Emp) (Geos (Poi C2) add Emp)" by (simp add:Bet_Point_def)
from P10 P27 P28 have P29 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
¬ Eq (Geos (Poi C2) add Emp) (Geos (Poi C3) add Emp)" by (blast intro:Eq_rev)
from P5 P6 have P30 : "Line_on (Li C2 A2) C3" by (blast intro:Line_rev Line_on_trans)
from P26 P29 P30 have "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
Def (Tri (Tr B2 C2 C3))" by (simp add:Tri_def_extension)
then have P31 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
Def (Tri (Tr C3 C2 B2))" by (blast intro:Tri_def_rev Tri_def_trans)
then have P32 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
Bet_Point (Se C2 A2) C3 ⟹ Gr (Geos (Ang (An B2 C3 A2)) add Emp) (Geos (Ang (An C3 C2 B2)) add Emp)" by (simp add:Ang_external_Gr)
have P33 : "Cong (Geos (Ang (An B2 C3 A2)) add Emp) (Geos (Ang (An A2 C3 B2)) add Emp)" by (simp add:Ang_roll)
from P31 have P34 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
Def (Ang (An C3 C2 B2))" by (simp add:Tri_to_Ang)
from P23 have P35 : "Def (Ang (An B2 C3 A2))" by (blast intro:Ang_def_rev Ang_def_inv)
from P23 P32 P33 P34 P35 have P36 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
Bet_Point (Se C2 A2) C3 ⟹ Gr (Geos (Ang (An A2 C3 B2)) add Emp) (Geos (Ang (An C3 C2 B2)) add Emp)" by (blast intro:Ang_Gr_trans_Eq_Gr Ang_rev)
have P37 : "Bet_Point (Se C2 A2) C3 ⟹ Line_on (Li C2 C3) A2" by (simp add:Line_Bet_on)
from P10 have P38 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
Bet_Point (Se C2 A2) C3 ⟹ ¬ Bet_Point (Se C3 A2) C2" by (blast intro:Bet_rev)
have P39 : "Line_on (Li C2 B2) B2" by (simp add:Line_on_rule)
have P40 : "¬ Bet_Point (Se B2 B2) C2" by (simp add:Bet_end_Point)
from P13 have P41 : "¬ Eq (Geos (Poi C2) add Emp) (Geos (Poi B2) add Emp)" by (simp add:Ang_def)
from P4 P34 P37 P38 P39 P40 P41 have "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
Bet_Point (Se C2 A2) C3 ⟹ Eq (Geos (Ang (An C3 C2 B2)) add Emp) (Geos (Ang (An A2 C2 B2)) add Emp)
∧ Def (Ang (An A2 C2 B2))" by (simp add:Ang_Point_swap)
then have P42 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
Bet_Point (Se C2 A2) C3 ⟹ Cong (Geos (Ang (An C3 C2 B2)) add Emp) (Geos (Ang (An A2 C2 B2)) add Emp)" by (blast intro:Ang_weektrans)
from P22 P23 P34 P36 P42 have P43 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
Bet_Point (Se C2 A2) C3 ⟹ Gr (Geos (Ang (An A2 C3 B2)) add Emp) (Geos (Ang (An A2 C2 B2)) add Emp)" by (blast intro:Ang_Gr_trans_Gr_Eq Ang_rev)
from P31 have "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹

```
    Def (Tri (Tr C2 C3 B2))" by (blast intro:Tri_def_rev Tri_def_trans)
  then have P44 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
    Bet_Point (Se A2 C3) C2 ⟹ Gr (Geos (Ang (An B2 C2 A2)) add Emp) (Geos (Ang (An C2 C3
B2)) add Emp)" by (simp add:Ang_external_Gr Bet_rev)
  have P45 : "Cong (Geos (Ang (An B2 C2 A2)) add Emp) (Geos (Ang (An A2 C2 B2)) add Emp)" by
(simp add:Ang_roll)
  from P26 have P46 : "Def (Ang (An B2 C2 A2))" by (simp add:Tri_to_Ang)
  from P34 have P47 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
    Def (Ang (An C2 C3 B2))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P22 P44 P45 P46 P47 have P48 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2
C3)) add Emp) ⟹
    Bet_Point (Se A2 C3) C2 ⟹ Gr (Geos (Ang (An A2 C2 B2)) add Emp) (Geos (Ang (An C2 C3
B2)) add Emp)" by (blast intro:Ang_Gr_trans_Eq_Gr Ang_rev)
  have P49 : "Bet_Point (Se A2 C3) C2 ⟹ Line_on (Li C3 C2) A2" by (simp add:Line_Bet_on)
  from P10 have P50 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
    Bet_Point (Se A2 C3) C2 ⟹ ¬ Bet_Point (Se C2 A2) C3" by (blast intro:Bet_rev)
  have P51 : "Line_on (Li C3 B2) B2" by (simp add:Line_on_rule)
  have P52 : "¬ Bet_Point (Se B2 B2) C3" by (simp add:Bet_end_Point)
  from P6 have P53 : "¬ Eq (Geos (Poi C3) add Emp) (Geos (Poi A2) add Emp)" by (blast intro:Eq_rev)
  from P47 have P54 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
    ¬ Eq (Geos (Poi C3) add Emp) (Geos (Poi B2) add Emp)" by (simp add:Ang_def)
  from P47 P49 P50 P51 P52 P53 P54 have "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se
A2 C3)) add Emp) ⟹
    Bet_Point (Se A2 C3) C2 ⟹ Eq (Geos (Ang (An C2 C3 B2)) add Emp) (Geos (Ang (An A2 C3
B2)) add Emp)
    ∧ Def (Ang (An A2 C3 B2))" by (simp add:Ang_Point_swap)
  then have P55 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3)) add Emp) ⟹
    Bet_Point (Se A2 C3) C2 ⟹ Cong (Geos (Ang (An C2 C3 B2)) add Emp) (Geos (Ang (An A2 C3
B2)) add Emp)" by (blast intro:Ang_weektrans)
  from P22 P23 P47 P48 P55 have P56 : "¬ Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2
C3)) add Emp) ⟹
    Bet_Point (Se A2 C3) C2 ⟹ Gr (Geos (Ang (An A2 C2 B2)) add Emp) (Geos (Ang (An A2 C3
B2)) add Emp)" by (blast intro:Ang_Gr_trans_Gr_Eq Ang_rev)
  from P10 P25 P43 P56 have P57 : "Eq (Geos (Seg (Se A2 C2)) add Emp) (Geos (Seg (Se A2 C3))
add Emp)" by blast
  from P8 P57 show "Cong (Geos (Tri (Tr A1 B1 C1)) add Emp) (Geos (Tri (Tr A2 B2 C2)) add Emp)"
by blast
qed

theorem (in Congruence_Rule) Tri_AAS :
  assumes
    "Def (Tri (Tr A1 B1 C1))" "Def (Tri (Tr A2 B2 C2))"
    "Eq (Geos (Seg (Se A1 B1)) add Emp) (Geos (Seg (Se A2 B2)) add Emp)"
    "Cong (Geos (Ang (An A1 C1 B1)) add Emp) (Geos (Ang (An A2 C2 B2)) add Emp)"
    "Cong (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An A2 B2 C2)) add Emp)
    ∨ Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An B2 A2 C2)) add Emp)"
  shows
    "Cong (Geos (Tri (Tr A1 B1 C1)) add Emp) (Geos (Tri (Tr A2 B2 C2)) add Emp)"
proof -
  from assms have P1 : "Def (Tri (Tr B1 A1 C1))" by (blast intro:Tri_def_rev Tri_def_trans)
  from assms have P2 : "Def (Tri (Tr B2 A2 C2))" by (blast intro:Tri_def_rev Tri_def_trans)
  have P3 : "Eq (Geos (Seg (Se A1 B1)) add Emp) (Geos (Seg (Se B1 A1)) add Emp)" by (simp
add:Seg_rev)
  have P4 : "Eq (Geos (Seg (Se A2 B2)) add Emp) (Geos (Seg (Se B2 A2)) add Emp)" by (simp
add:Seg_rev)
  from assms P3 P4 have P5 : "Eq (Geos (Seg (Se B1 A1)) add Emp) (Geos (Seg (Se B2 A2)) add
Emp)" by (blast intro:Eq_trans Eq_rev)
  have P6 : "Eq (Geos (Ang (An A1 C1 B1)) add Emp) (Geos (Ang (An B1 C1 A1)) add Emp)" by
(simp add:Ang_roll)
  from assms P6 have P7 : "Cong (Geos (Ang (An B1 C1 A1)) add Emp) (Geos (Ang (An A2 C2 B2))
add Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  have P8 : "Eq (Geos (Ang (An A2 C2 B2)) add Emp) (Geos (Ang (An B2 C2 A2)) add Emp)" by
(simp add:Ang_roll)
  from P7 P8 have P9 : "Cong (Geos (Ang (An B1 C1 A1)) add Emp) (Geos (Ang (An B2 C2 A2)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from P1 P2 P5 P9 have "Cong (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An A2 B2 C2))
```

```
add Emp) ⟹
    Cong (Geos (Tri (Tr B1 A1 C1)) add Emp) (Geos (Tri (Tr B2 A2 C2)) add Emp)" by (simp
add:Tri_AAS_lemma1)
  then have P10 : "Cong (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An A2 B2 C2)) add Emp)
⟹
    Cong (Geos (Ang (An C1 A1 B1)) add Emp) (Geos (Ang (An C2 A2 B2)) add Emp)" by (simp
add:Tri_Cong_def)
  have P11 : "Eq (Geos (Ang (An C1 A1 B1)) add Emp) (Geos (Ang (An B1 A1 C1)) add Emp)" by
(simp add:Ang_roll)
  from P10 P11 have P12 : "Cong (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An A2 B2 C2))
add Emp) ⟹
    Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An C2 A2 B2)) add Emp)" by (blast
intro:Ang_weektrans Ang_rev Eq_rev)
  have P13 : "Eq (Geos (Ang (An C2 A2 B2)) add Emp) (Geos (Ang (An B2 A2 C2)) add Emp)" by
(simp add:Ang_roll)
  from P12 P13 have "Cong (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An A2 B2 C2)) add
Emp) ⟹
    Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An B2 A2 C2)) add Emp)" by (blast
intro:Ang_weektrans Ang_rev Eq_rev)
  then have P14 : "Cong (Geos (Ang (An A1 B1 C1)) add Emp) (Geos (Ang (An A2 B2 C2)) add Emp)
    ∨ Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An B2 A2 C2)) add Emp) ⟹
    Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An B2 A2 C2)) add Emp)" by blast
  from assms have P15 : "Cong (Geos (Ang (An B1 A1 C1)) add Emp) (Geos (Ang (An B2 A2 C2))
add Emp) ⟹
    Cong (Geos (Tri (Tr A1 B1 C1)) add Emp) (Geos (Tri (Tr A2 B2 C2)) add Emp)" by (simp
add:Tri_AAS_lemma1)
  from assms P14 P15 show "Cong (Geos (Tri (Tr A1 B1 C1)) add Emp) (Geos (Tri (Tr A2 B2 C2))
add Emp)" by blast
qed

theorem (in Congruence_Rule) Seg_bisection :
  assumes
    "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)"
  shows
    "∃p. Eq (Geos (Seg (Se A p)) add Emp) (Geos (Seg (Se p B)) add Emp) ∧ Bet_Point (Se A B) p"
proof -
  have "∃p q r. ¬ Line_on (Li A B) p ∧ ¬ Line_on (Li A B) q ∧ ¬ Line_on (Li A B) r
    ∧ ¬ Eq (Geos (Poi p) add Emp) (Geos (Poi q) add Emp) ∧ ¬ Eq (Geos (Poi q) add Emp) (Geos
(Poi r) add Emp)
    ∧ ¬ Eq (Geos (Poi r) add Emp) (Geos (Poi p) add Emp)" by (blast intro:Line_not_on_exist)
  then obtain C :: Point where P1 : "¬ Line_on (Li A B) C" by blast
  from assms P1 have P2 : "Def (Ang (An A B C))" by (simp add:Ang_sinple_def)
  then have P2 : "Def (Ang (An C A B))" by (blast intro:Ang_def_rev Ang_def_inv)
  from assms have P3 : "Eq (Geos (Lin (Li A B)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (simp
add:Line_rev)
  from P1 P3 have P4 : "¬ Line_on (Li B A) C" by (simp add:Line_not_on_trans)
  from P2 P4 have "∃p. Cong (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An p B A)) add Emp)
    ∧ Plane_diffside (Li B A) p C" by (simp add:Ang_move_diffside)
  then obtain D1 :: Point where P5 : "Cong (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An D1 B
A)) add Emp)
    ∧ Plane_diffside (Li B A) D1 C" by blast
  then have "∃p. Bet_Point (Se D1 C) p ∧ Line_on (Li B A) p ∧ ¬ Line_on (Li B A) D1 ∧ ¬ Line_on
(Li B A) C" by (simp add:Plane_diffside_def)
  then have P6 : "¬ Line_on (Li B A) D1" by blast
  from assms have P7 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp)" by (blast intro:Eq_rev)
  from P6 P7 have "Def (Ang (An B A D1))" by (simp add:Ang_sinple_def)
  then have P8 : "Def (Ang (An D1 B A))" by (blast intro:Ang_def_rev Ang_def_inv)
  from P2 P5 P8 have "∃p. Cong (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An p B A)) add Emp)
    ∧ Eq (Geos (Ang (An D1 B A)) add Emp) (Geos (Ang (An p B A)) add Emp)
    ∧ Eq (Geos (Seg (Se A C)) add Emp) (Geos (Seg (Se B p)) add Emp) ∧ Line_on (Li B D1) p
    ∧ ¬ Bet_Point (Se p D1) B ∧ Def (Ang (An p B A))" by (simp add:Ang_replace)
  then obtain D :: Point where P9 : "Cong (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An D B A))
add Emp)
    ∧ Eq (Geos (Ang (An D1 B A)) add Emp) (Geos (Ang (An D B A)) add Emp)
    ∧ Eq (Geos (Seg (Se A C)) add Emp) (Geos (Seg (Se B D)) add Emp) ∧ Line_on (Li B D1) D
    ∧ ¬ Bet_Point (Se D D1) B ∧ Def (Ang (An D B A))" by blast
```

have ”Plane_diffside (Li B A) D D1 ⟹
∃p. Bet_Point (Se D D1) p ∧ Line_on (Li B A) p ∧ ¬ Line_on (Li B A) D ∧ ¬ Line_on (Li B A) D1” by (simp add:Plane_diffside_def)
then obtain B1 :: Point where P10 : ”Plane_diffside (Li B A) D D1 ⟹
Bet_Point (Se D D1) B1 ∧ Line_on (Li B A) B1 ∧ ¬ Line_on (Li B A) D ∧ ¬ Line_on (Li B A) D1” by blast
then have P11 : ”Plane_diffside (Li B A) D D1 ⟹ Line_on (Li D D1) B1” by (simp add:Line_Bet_on)
from P10 have ”Plane_diffside (Li B A) D D1 ⟹ Bet_Point (Se D D1) B1” by simp
then have P12 : ”Plane_diffside (Li B A) D D1 ⟹
¬ Eq (Geos (Poi D) add Emp) (Geos (Poi D1) add Emp)” by (simp add:Bet_Point_def)
have P13 : ”Line_on (Li B D1) D1” by (simp add:Line_on_rule)
have P14 : ”Line_on (Li D D1) D” by (simp add:Line_on_rule)
have P15 : ”Line_on (Li D D1) D1” by (simp add:Line_on_rule)
from P9 have P16 : ”Line_on (Li B D1) D” by simp
from P12 P13 P14 P15 P16 have P17 : ”Plane_diffside (Li B A) D D1 ⟹
Eq (Geos (Lin (Li B D1)) add Emp) (Geos (Lin (Li D D1)) add Emp)” by (simp add:Line_unique)
have P18 : ”Line_on (Li B D1) B” by (simp add:Line_on_rule)
from P17 P18 have P19 : ”Plane_diffside (Li B A) D D1 ⟹ Line_on (Li D D1) B” by (simp add:Line_on_trans)
from P14 have P20 : ”Eq (Geos (Lin (Li D D1)) add Emp) (Geos (Lin (Li B A)) add Emp) ⟹
Line_on (Li B A) D” by (simp add:Line_on_trans)
from P10 P20 have P21 : ”Plane_diffside (Li B A) D D1 ⟹
¬ Eq (Geos (Lin (Li D D1)) add Emp) (Geos (Lin (Li B A)) add Emp)” by blast
have P22 : ”Line_on (Li B A) B” by (simp add:Line_on_rule)
from P10 P11 P19 P21 P22 have P23 : ”Plane_diffside (Li B A) D D1 ⟹
Eq (Geos (Poi B1) add Emp) (Geos (Poi B) add Emp)” by (simp add:Line_unique_Point)
from P10 P23 have P24 : ”Plane_diffside (Li B A) D D1 ⟹ Bet_Point (Se D D1) B” by (blast intro:Point_Eq)
from P9 P24 have P25 : ”¬ Plane_diffside (Li B A) D D1” by blast
from P5 have P26 : ”Plane_sameside (Li B A) C D ⟹ Plane_diffside (Li B A) D D1” by (simp add:Plane_diffside_rev Plane_trans)
from P25 P26 have P27 : ”¬ Plane_sameside (Li B A) C D” by blast
from P5 have P28 : ”Eq (Geos (Poi C) add Emp) (Geos (Poi D) add Emp) ⟹
Plane_diffside (Li B A) D1 D” by (blast intro:Point_Eq)
from P25 P28 have P29 : ”¬ Eq (Geos (Poi C) add Emp) (Geos (Poi D) add Emp)” by (blast intro:Plane_diffside_rev)
from P9 have ”Def (Tri (Tr B A D))” by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
then have P30 : ”¬ Line_on (Li B A) D” by (simp add:Tri_def_Line)
from P4 P27 P29 P30 have ”Plane_diffside (Li B A) C D” by (simp add:Plane_not_sameside_diffside)
then have ”∃p. Bet_Point (Se C D) p ∧ Line_on (Li B A) p ∧ ¬ Line_on (Li B A) C ∧ ¬ Line_on (Li B A) D” by (simp add:Plane_diffside_def)
then obtain E :: Point where P31 : ”Bet_Point (Se C D) E ∧ Line_on (Li B A) E
∧ ¬ Line_on (Li B A) C ∧ ¬ Line_on (Li B A) D” by blast
then have P32 : ”Bet_Point (Se C D) E” by simp
then have P33 : ”Eq (Geos (Poi E) add Emp) (Geos (Poi A) add Emp) ⟹ Bet_Point (Se D C) A” by (blast intro:Point_Eq Bet_rev)
from P9 have P34 : ”Def (Tri (Tr A D B))” by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
from P33 P34 have P35 : ”Eq (Geos (Poi E) add Emp) (Geos (Poi A) add Emp) ⟹
Gr (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An A B D)) add Emp)”
by (simp add:Ang_external_Gr)
from P32 have P36 : ”Eq (Geos (Poi E) add Emp) (Geos (Poi B) add Emp) ⟹ Bet_Point (Se C D) B” by (simp add:Point_Eq)
from P2 have P37 : ”Def (Tri (Tr B C A))” by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
from P36 P37 have P38 : ”Eq (Geos (Poi E) add Emp) (Geos (Poi B) add Emp) ⟹
Gr (Geos (Ang (An A B D)) add Emp) (Geos (Ang (An B A C)) add Emp)”
by (simp add:Ang_external_Gr)
have P39 : ”Eq (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An B A C)) add Emp)” by (simp add:Ang_roll)
from P9 P39 have P40 : ”Cong (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An D B A)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
have P41 : ”Eq (Geos (Ang (An D B A)) add Emp) (Geos (Ang (An A B D)) add Emp)” by (simp add:Ang_roll)
from P40 P41 have P42 : ”Cong (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An A B D)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P2 have P43 : ”Def (Ang (An B A C))” by (blast intro:Ang_def_rev Ang_def_inv)
from P9 have P44 : ”Def (Ang (An A B D))” by (simp add:Ang_def_rev)

from P42 P43 P44 have P45 : "¬ Gr (Geos (Ang (An B A C)) add Emp) (Geos (Ang (An A B D)) add Emp)
∧ ¬ Gr (Geos (Ang (An A B D)) add Emp) (Geos (Ang (An B A C)) add Emp)" by (simp add:Ang_not_Gr)
from P35 P45 have P46 : "¬ Eq (Geos (Poi E) add Emp) (Geos (Poi A) add Emp)" by blast
from P38 P45 have P47 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi E) add Emp)" by (blast intro:Eq_rev)
have P48 : "Line_on (Li B A) B" by (simp add:Line_on_rule)
have P49 : "Line_on (Li B A) A" by (simp add:Line_on_rule)
from assms P31 P46 P47 P48 P49 have P50 : "Bet_Point (Se A E) B ∨ Bet_Point (Se E B) A
∨ Bet_Point (Se B A) E" by (simp add:Bet_case)
have P51 : "Bet_Point (Se A E) B ⟹ Line_on (Li B E) A" by (simp add:Line_Bet_on)
have P52 : "Line_on (Li B E) B" by (simp add:Line_on_rule)
from assms P48 P49 P51 P52 have "Bet_Point (Se A E) B ⟹
Eq (Geos (Lin (Li B E)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (simp add:Line_unique)
then have P53 : "Bet_Point (Se A E) B ⟹ Line_on (Li B E) D ⟹ Line_on (Li B A) D" by (simp add:Line_on_trans)
from P30 P53 have P54 : "Bet_Point (Se A E) B ⟹ ¬ Line_on (Li B E) D" by blast
from P47 P54 have P55 : "Bet_Point (Se A E) B ⟹ Def (Tri (Tr B E D))" by (simp add:Ang_sinple_def Ang_to_Tri)
have P56 : "Bet_Point (Se A E) B ⟹ Bet_Point (Se E A) B" by (simp add:Bet_rev)
from P55 P56 have P57 : "Bet_Point (Se A E) B ⟹
Gr (Geos (Ang (An D B A)) add Emp) (Geos (Ang (An B E D)) add Emp)"
by (simp add:Ang_external_Gr)
have P58 : "Bet_Point (Se A E) B ⟹ Line_on (Li E A) B" by (simp add:Line_Bet_on)
have P59 : "Line_on (Li E A) A" by (simp add:Line_on_rule)
from assms P48 P49 P58 P59 have P60 : "Bet_Point (Se A E) B ⟹
Eq (Geos (Lin (Li E A)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (simp add:Line_unique)
then have P61 : "Bet_Point (Se A E) B ⟹ Line_on (Li E A) C ⟹ Line_on (Li B A) C" by (simp add:Line_on_trans)
from P31 P61 have P62 : "Bet_Point (Se A E) B ⟹ ¬ Line_on (Li E A) C" by blast
from P46 P62 have "Bet_Point (Se A E) B ⟹ Def (Tri (Tr E A C))" by (simp add:Ang_sinple_def Ang_to_Tri)
then have P63 : "Bet_Point (Se A E) B ⟹ Def (Tri (Tr E C A))" by (blast intro:Tri_def_rev Tri_def_trans)
from P31 P63 have P64 : "Bet_Point (Se A E) B ⟹
Gr (Geos (Ang (An A E D)) add Emp) (Geos (Ang (An E A C)) add Emp)"
by (simp add:Ang_external_Gr)
from P63 have P65 : "Bet_Point (Se A E) B ⟹ Def (Ang (An E A C))" by (blast intro:Tri_to_Ang Ang_def_inv)
have P66 : "Bet_Point (Se A E) B ⟹ Line_on (Li A E) B" by (simp add:Line_Bet_on)
have "Bet_Point (Se A E) B ⟹ Inv (Bet_Point (Se E B) A)" by (simp add:Bet_iff)
then have P67 : "Bet_Point (Se A E) B ⟹ ¬ Bet_Point (Se E B) A" by (simp add:Inv_def)
have P68 : "Line_on (Li A C) C" by (simp add:Line_on_rule)
have P69 : "¬ Bet_Point (Se C C) A" by (simp add:Bet_end_Point)
from P2 have "¬ Eq (Geos (Poi C) add Emp) (Geos (Poi A) add Emp)" by (simp add:Ang_def)
then have P70 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi C) add Emp)" by (blast intro:Eq_rev)
from assms P65 P66 P67 P68 P69 P70 have P71 : "Bet_Point (Se A E) B ⟹
Eq (Geos (Ang (An E A C)) add Emp) (Geos (Ang (An B A C)) add Emp) ∧ Def (Ang (An B A C))" by (simp add:Ang_Point_swap)
then have P72 : "Bet_Point (Se A E) B ⟹
Cong (Geos (Ang (An E A C)) add Emp) (Geos (Ang (An B A C)) add Emp)" by (blast intro:Ang_weektrans Ang_rev)
from P60 have P73 : "Bet_Point (Se A E) B ⟹ Line_on (Li E A) D ⟹ Line_on (Li B A) D" by (simp add:Line_on_trans)
from P31 P73 have P74 : "Bet_Point (Se A E) B ⟹ ¬ Line_on (Li E A) D" by blast
from P46 P74 have "Bet_Point (Se A E) B ⟹ Def (Ang (An E A D))" by (simp add:Ang_sinple_def)
then have P75 : "Bet_Point (Se A E) B ⟹ Def (Ang (An A E D))" by (blast intro:Ang_def_rev Ang_def_inv)
from P64 P65 P71 P72 P75 have P76 : "Bet_Point (Se A E) B ⟹
Gr (Geos (Ang (An A E D)) add Emp) (Geos (Ang (An B A C)) add Emp)"
by (blast intro:Ang_Gr_trans_Gr_Eq Ang_rev)
have "Bet_Point (Se A E) B ⟹ Inv (Bet_Point (Se B A) E)" by (simp add:Bet_iff)
then have "Bet_Point (Se A E) B ⟹ ¬ Bet_Point (Se B A) E" by (simp add:Inv_def)
then have P77 : "Bet_Point (Se A E) B ⟹ ¬ Bet_Point (Se A B) E" by (blast intro:Bet_rev)
have P78 : "Line_on (Li E D) D" by (simp add:Line_on_rule)
have P79 : "¬ Bet_Point (Se D D) E" by (simp add:Bet_end_Point)

from P47 have P80 : "¬ Eq (Geos (Poi E) add Emp) (Geos (Poi B) add Emp)" by (blast intro:Eq_rev)
from P32 have "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi E) add Emp)" by (simp add:Bet_Point_def)
then have P81 : "¬ Eq (Geos (Poi E) add Emp) (Geos (Poi D) add Emp)" by (blast intro:Eq_rev)
from P58 P75 P77 P78 P79 P80 P81 have P82 : "Bet_Point (Se A E) B ⟹
Eq (Geos (Ang (An A E D)) add Emp) (Geos (Ang (An B E D)) add Emp) ∧ Def (Ang (An B E D))" by (simp add:Ang_Point_swap)
then have P83 : "Bet_Point (Se A E) B ⟹
Cong (Geos (Ang (An A E D)) add Emp) (Geos (Ang (An B E D)) add Emp)" by (blast intro:Ang_weektrans)
from P9 P57 P75 P82 P83 have P84 : "Bet_Point (Se A E) B ⟹
Gr (Geos (Ang (An D B A)) add Emp) (Geos (Ang (An A E D)) add Emp)"
by (blast intro:Ang_Gr_trans_Gr_Eq Ang_rev)
from P9 P71 P75 P76 P84 have P85 : "Bet_Point (Se A E) B ⟹
Gr (Geos (Ang (An D B A)) add Emp) (Geos (Ang (An B A C)) add Emp)"
by (blast intro:Ang_Gr_trans_Gr_Gr)
from P9 P40 P71 have P86 : "Bet_Point (Se A E) B ⟹
¬ Gr (Geos (Ang (An D B A)) add Emp) (Geos (Ang (An B A C)) add Emp)"
by (simp add:Ang_not_Gr)
from P85 P86 have P87 : "¬ Bet_Point (Se A E) B" by blast
have P88 : "Bet_Point (Se E B) A ⟹ Line_on (Li E A) B" by (simp add:Line_Bet_on)
from assms P48 P49 P59 P88 have P89 : "Bet_Point (Se E B) A ⟹
Eq (Geos (Lin (Li E A)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (simp add:Line_unique)
then have P90 : "Bet_Point (Se E B) A ⟹ Line_on (Li E A) C ⟹ Line_on (Li B A) C" by (simp add:Line_on_trans)
from P31 P90 have P91 : "Bet_Point (Se E B) A ⟹ ¬ Line_on (Li E A) C" by blast
from P46 P91 have "Bet_Point (Se E B) A ⟹ Def (Ang (An E A C))" by (simp add:Ang_sinple_def)
then have P92 : "Bet_Point (Se E B) A ⟹ Def (Tri (Tr A E C))" by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
then have P93 : "Bet_Point (Se E B) A ⟹ Gr (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An A E C)) add Emp)" by (simp add:Ang_external_Gr)
have P94 : "Bet_Point (Se E B) A ⟹ Line_on (Li B E) A" by (simp add:Line_Bet_on)
from assms P48 P49 P52 P94 have "Bet_Point (Se E B) A ⟹
Eq (Geos (Lin (Li B E)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (simp add:Line_unique)
then have P95 : "Bet_Point (Se E B) A ⟹ Line_on (Li B E) D ⟹ Line_on (Li B A) D" by (simp add:Line_on_trans)
from P31 P95 have P96 : "Bet_Point (Se E B) A ⟹ ¬ Line_on (Li B E) D" by blast
from P47 P96 have "Bet_Point (Se E B) A ⟹ Def (Ang (An B E D))" by (simp add:Ang_sinple_def)
then have P97 : "Bet_Point (Se E B) A ⟹ Def (Tri (Tr E D B))" by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
from P32 have P98 : "Bet_Point (Se D C) E" by (simp add:Bet_rev)
from P97 P98 have P99 : "Bet_Point (Se E B) A ⟹
Gr (Geos (Ang (An B E C)) add Emp) (Geos (Ang (An E B D)) add Emp)"
by (simp add:Ang_external_Gr)
from P92 have P100 : "Bet_Point (Se E B) A ⟹ Def (Ang (An A E C))" by (simp add:Tri_to_Ang)
have "Bet_Point (Se E B) A ⟹ Inv (Bet_Point (Se B A) E)" by (simp add:Bet_iff)
then have "Bet_Point (Se E B) A ⟹ ¬ Bet_Point (Se B A) E" by (simp add:Inv_def)
then have P101 : "Bet_Point (Se E B) A ⟹ ¬ Bet_Point (Se A B) E" by (blast intro:Bet_rev)
have P102 : "Line_on (Li E C) C" by (simp add:Line_on_rule)
have P103 : "¬ Bet_Point (Se C C) E" by (simp add:Bet_end_Point)
from P32 have P104 : "¬ Eq (Geos (Poi E) add Emp) (Geos (Poi C) add Emp)"
by (simp add:Bet_Point_def)
from P80 P88 P100 P101 P102 P103 P104 have P105 : "Bet_Point (Se E B) A ⟹
Eq (Geos (Ang (An A E C)) add Emp) (Geos (Ang (An B E C)) add Emp) ∧ Def (Ang (An B E C))" by (simp add:Ang_Point_swap)
then have P106 : "Bet_Point (Se E B) A ⟹
Cong (Geos (Ang (An A E C)) add Emp) (Geos (Ang (An B E C)) add Emp)" by (blast intro:Ang_weektrans)
have P107 : "Bet_Point (Se E B) A ⟹ Line_on (Li B A) E" by (simp add:Line_Bet_on)
have "Bet_Point (Se E B) A ⟹ Inv (Bet_Point (Se A E) B)" by (simp add:Bet_iff)
then have P108 : "Bet_Point (Se E B) A ⟹ ¬ Bet_Point (Se A E) B" by (simp add:Inv_def)
have P109 : "Line_on (Li B D) D" by (simp add:Line_on_rule)
have P110 : "¬ Bet_Point (Se D D) B" by (simp add:Bet_end_Point)
from P44 have P111 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)" by (simp add:Ang_def)
from P44 P47 P107 P108 P109 P110 P111 have P112 : "Bet_Point (Se E B) A ⟹
Eq (Geos (Ang (An A B D)) add Emp) (Geos (Ang (An E B D)) add Emp) ∧ Def (Ang (An E B D))" by (simp add:Ang_Point_swap)

then have P113 : ”Bet_Point (Se E B) A ⟹
Cong (Geos (Ang (An A B D)) add Emp) (Geos (Ang (An E B D)) add Emp)” by (blast intro:Ang_weektrans)
from P2 P93 P100 P105 P106 have P114 : ”Bet_Point (Se E B) A ⟹
Gr (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An B E C)) add Emp)”
by (blast intro:Ang_Gr_trans_Gr_Eq)
from P44 P99 P105 P112 P113 have P115 : ”Bet_Point (Se E B) A ⟹
Gr (Geos (Ang (An B E C)) add Emp) (Geos (Ang (An A B D)) add Emp)”
by (blast intro:Ang_Gr_trans_Gr_Eq Ang_rev)
from P2 P44 P105 P114 P115 have P116 : ”Bet_Point (Se E B) A ⟹
Gr (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An A B D)) add Emp)”
by (blast intro:Ang_Gr_trans_Gr_Gr)
from P9 P41 have P117 : ”Cong (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An A B D)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P2 P44 P117 have P118 : ”¬ Gr (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An A B D)) add Emp)
∧ ¬ Gr (Geos (Ang (An A B D)) add Emp) (Geos (Ang (An C A B)) add Emp)” by (simp add:Ang_not_Gr)
from P116 P118 have P119 : ”¬ Bet_Point (Se E B) A” by blast
from P50 P87 P119 have P120 : ”Bet_Point (Se B A) E” by blast
then have P121 : ”Line_on (Li B E) A” by (simp add:Line_Bet_on)
from assms P48 P49 P52 P121 have ”Eq (Geos (Lin (Li B E)) add Emp) (Geos (Lin (Li B A)) add Emp)” by (simp add:Line_unique)
then have P122 : ”Line_on (Li B E) C ⟹ Line_on (Li B A) C” by (simp add:Line_on_trans)
from P31 P122 have P123 : ”¬ Line_on (Li B E) C” by blast
from P47 P123 have P124 : ”Def (Ang (An B E C))” by (simp add:Ang_sinple_def)
from P32 P120 P124 have P125 : ”Cong (Geos (Ang (An C E A)) add Emp) (Geos (Ang (An B E D)) add Emp)” by (simp add:Ang_vertical)
have P126 : ”Eq (Geos (Seg (Se A C)) add Emp) (Geos (Seg (Se C A)) add Emp)” by (simp add:Seg_rev)
have P127 : ”Eq (Geos (Seg (Se B D)) add Emp) (Geos (Seg (Se D B)) add Emp)” by (simp add:Seg_rev)
from P9 P126 P127 have P128 : ”Eq (Geos (Seg (Se C A)) add Emp) (Geos (Seg (Se D B)) add Emp)” by (blast intro:Eq_trans Eq_rev)
have P129 : ”Eq (Geos (Ang (An B E D)) add Emp) (Geos (Ang (An D E B)) add Emp)” by (simp add:Ang_roll)
from P125 P129 have P130 : ”Cong (Geos (Ang (An C E A)) add Emp) (Geos (Ang (An D E B)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P120 have P131 : ”Line_on (Li A B) E” by (simp add:Line_Bet_on)
from P119 have P132 : ”¬ Bet_Point (Se B E) A” by (blast intro:Bet_rev)
from P46 have P133 : ”¬ Eq (Geos (Poi A) add Emp) (Geos (Poi E) add Emp)” by (blast intro:Eq_rev)
from P2 P68 P69 P70 P131 P132 P133 have P134 :
”Eq (Geos (Ang (An C A B)) add Emp) (Geos (Ang (An C A E)) add Emp) ∧ Def (Ang (An C A E))” by (simp add:Ang_Point_swap)
then have P135 : ”Def (Tri (Tr C A E))” by (simp add:Ang_to_Tri)
from P9 P134 have P136 : ”Cong (Geos (Ang (An C A E)) add Emp) (Geos (Ang (An D B A)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P9 have P137 : ”Def (Ang (An D B A))” by simp
from P31 P47 P87 P109 P110 P111 P137 have P138 :
”Eq (Geos (Ang (An D B A)) add Emp) (Geos (Ang (An D B E)) add Emp) ∧ Def (Ang (An D B E))” by (simp add:Ang_Point_swap)
then have P139 : ”Def (Tri (Tr D B E))” by (simp add:Ang_to_Tri)
from P136 P138 have P140 : ”Cong (Geos (Ang (An C A E)) add Emp) (Geos (Ang (An D B E)) add Emp)” by (blast intro:Ang_weektrans Ang_rev Eq_rev)
from P128 P130 P135 P139 P140 have ”Cong (Geos (Tri (Tr C A E)) add Emp) (Geos (Tri (Tr D B E)) add Emp)” by (simp add:Tri_AAS)
then have P141 : ”Eq (Geos (Seg (Se A E)) add Emp) (Geos (Seg (Se B E)) add Emp)” by (simp add:Tri_Cong_def)
have P142 : ”Eq (Geos (Seg (Se B E)) add Emp) (Geos (Seg (Se E B)) add Emp)” by (simp add:Seg_rev)
from P141 P142 have P143 : ”Eq (Geos (Seg (Se A E)) add Emp) (Geos (Seg (Se E B)) add Emp)” by (blast intro:Eq_trans)
from P120 have P144 : ”Bet_Point (Se A B) E” by (simp add:Bet_rev)
from P143 P144 show ”∃p. Eq (Geos (Seg (Se A p)) add Emp) (Geos (Seg (Se p B)) add Emp) ∧ Bet_Point (Se A B) p” by blast
qed

```
theorem (in Congruence_Rule) Ang_bisection :
  assumes
    "Def (Ang (An A B C))"
  shows
    "∃p. Cong (Geos (Ang (An A B p)) add Emp) (Geos (Ang (An p B C)) add Emp)
    ∧ Ang_inside (An A B C) p ∧ Def (Ang (An A B p)) ∧ Def (Ang (An p B C))"
proof -
  have P1 : "Line_on (Li B C) B" by (simp add:Line_on_rule)
  have P2 : "Line_on (Li B C) C" by (simp add:Line_on_rule)
  from assms have P3 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C) add Emp)" by (simp add:Ang_def)
  from assms have "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi B) add Emp)" by (simp add:Ang_def)
  then have P4 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi A) add Emp)" by (blast intro:Eq_rev)
  from P1 P2 P3 P4 have "∃p. Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se B p)) add Emp)
    ∧ ¬ Bet_Point (Se p C) B ∧ Line_on (Li B C) p ∧ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi p) add
Emp)" by (simp add:Seg_move_sameside)
  then obtain C2 :: Point where P5 : "Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se B C2)) add
Emp)
    ∧ ¬ Bet_Point (Se C2 C) B ∧ Line_on (Li B C) C2 ∧ ¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C2)
add Emp)" by blast
  then have P6 : "Line_on (Li B C) C2" by simp
  then have P7 : "Eq (Geos (Poi C2) add Emp) (Geos (Poi A) add Emp) ⟹ Line_on (Li B C) A" by
(simp add:Point_Eq)
  from assms have P8 : "Def (Tri (Tr A B C))" by (simp add:Ang_to_Tri)
  then have P9 : "¬ Line_on (Li B C) A" by (simp add:Tri_def_Line)
  from P7 P9 have P10 : "¬ Eq (Geos (Poi C2) add Emp) (Geos (Poi A) add Emp)" by blast
  then have "∃p. Eq (Geos (Seg (Se C2 p)) add Emp) (Geos (Seg (Se p A)) add Emp) ∧ Bet_Point (Se
C2 A) p" by (simp add:Seg_bisection)
  then obtain D :: Point where P11 : "Eq (Geos (Seg (Se C2 D)) add Emp) (Geos (Seg (Se D A)) add
Emp)
    ∧ Bet_Point (Se C2 A) D" by blast
  have P12 : "Line_on (Li B A) B" by (simp add:Line_on_rule)
  from P1 P5 P6 P12 have P13 : "Line_on (Li B A) C2 ⟹
    Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp)" by (simp add:Line_unique)
  from assms have P14 : "¬ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C)) add Emp)" by
(simp add:Ang_def)
  from P13 P14 have P15 : "¬ Line_on (Li B A) C2" by blast
  from P11 have P16 : "Bet_Point (Se C2 A) D" by simp
  then have P17 : "Line_on (Li C2 A) D" by (simp add:Line_Bet_on)
  have P18 : "Line_on (Li C2 A) A" by (simp add:Line_on_rule)
  have P19 : "Line_on (Li B A) A" by (simp add:Line_on_rule)
  from P16 have P20 : "¬ Eq (Geos (Poi A) add Emp) (Geos (Poi D) add Emp)"
    by (simp add:Bet_Point_def)
  from P17 P18 P19 P20 have P21 : "Line_on (Li B A) D ⟹
    Eq (Geos (Lin (Li C2 A)) add Emp) (Geos (Lin (Li B A)) add Emp)" by (simp add:Line_unique)
  have P22 : "Line_on (Li C2 A) C2" by (simp add:Line_on_rule)
  from P21 P22 have P23 : "Line_on (Li B A) D ⟹ Line_on (Li B A) C2" by (simp add:Line_on_trans)
  from P15 P23 have P24 : "¬ Line_on (Li B A) D" by blast
  from P4 P24 have "Def (Ang (An B A D))" by (simp add:Ang_sinple_def)
  then have P25 : "Def (Tri (Tr A B D))" by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
  from P4 P15 have P26 : "Def (Ang (An B A C2))" by (simp add:Ang_sinple_def)
  then have "Def (Tri (Tr C2 B A))" by (blast intro:Ang_to_Tri Tri_def_rev Tri_def_trans)
  then have P27 : "¬ Line_on (Li C2 B) A" by (simp add:Tri_def_Line)
  have P28 : "Line_on (Li C2 B) C2" by (simp add:Line_on_rule)
  from P16 have P29 : "¬ Eq (Geos (Poi D) add Emp) (Geos (Poi C2) add Emp)"
    by (simp add:Bet_Point_def)
  from P17 P22 P28 P29 have P30 : "Line_on (Li C2 B) D ⟹
    Eq (Geos (Lin (Li C2 A)) add Emp) (Geos (Lin (Li C2 B)) add Emp)" by (simp add:Line_unique)
  from P18 P30 have P31 : "Line_on (Li C2 B) D ⟹ Line_on (Li C2 B) A" by (simp add:Line_on_trans)
  from P27 P31 have P32 : "¬ Line_on (Li C2 B) D" by blast
  from P5 have P33 : "¬ Eq (Geos (Poi C2) add Emp) (Geos (Poi B) add Emp)" by (blast intro:Eq_rev)
  from P32 P33 have "Def (Ang (An C2 B D))" by (simp add:Ang_sinple_def)
  then have P34 : "Def (Tri (Tr C2 B D))" by (simp add:Ang_to_Tri)
  have P35 : "Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se A B)) add Emp)" by (simp add:Seg_rev)
  have P36 : "Eq (Geos (Seg (Se B C2)) add Emp) (Geos (Seg (Se C2 B)) add Emp)" by (simp
add:Seg_rev)
  from P5 have P37 : "Eq (Geos (Seg (Se B A)) add Emp) (Geos (Seg (Se B C2)) add Emp)" by simp
```

```
  from P35 P36 P37 have P38 : "Eq (Geos (Seg (Se A B)) add Emp) (Geos (Seg (Se C2 B)) add Emp)"
by (blast intro:Eq_rev Eq_trans)
  have P39 : "Eq (Geos (Seg (Se C2 D)) add Emp) (Geos (Seg (Se D C2)) add Emp)" by (simp
add:Seg_rev)
  from P11 P39 have P40 : "Eq (Geos (Seg (Se D A)) add Emp) (Geos (Seg (Se D C2)) add Emp)" by
(blast intro:Eq_rev Eq_trans)
  from P25 P34 P38 P40 have "Cong (Geos (Tri (Tr A B D)) add Emp) (Geos (Tri (Tr C2 B D)) add
Emp)" by (simp add:Tri_SSS)
  then have P41 : "Cong (Geos (Ang (An D B A)) add Emp) (Geos (Ang (An D B C2)) add Emp)" by
(simp add:Tri_Cong_def)
  have P42 : "Eq (Geos (Ang (An D B A)) add Emp) (Geos (Ang (An A B D)) add Emp)" by (simp
add:Ang_roll)
  from P41 P42 have P43 : "Cong (Geos (Ang (An A B D)) add Emp) (Geos (Ang (An D B C2)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from P34 have P44 : "Def (Ang (An D B C2))" by (blast intro:Tri_to_Ang Ang_def_rev Ang_def_inv)
  have P45 : "Line_on (Li B D) D" by (simp add:Line_on_rule)
  have P46 : "¬ Bet_Point (Se D D) B" by (simp add:Bet_end_Point)
  from P33 have P47 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi C2) add Emp)" by (blast intro:Eq_rev)
  from P3 P6 P47 have P48 : "Line_on (Li B C2) C" by (simp add:Line_on_rev)
  from P34 have P49 : "¬ Eq (Geos (Poi B) add Emp) (Geos (Poi D) add Emp)" by (simp add:Tri_def)
  from P3 P5 P44 P45 P46 P48 P49 have P50 : "Eq (Geos (Ang (An D B C2)) add Emp) (Geos (Ang
(An D B C)) add Emp)
    ∧ Def (Ang (An D B C))" by (simp add:Ang_Point_swap)
  from P43 P50 have P51 : "Cong (Geos (Ang (An A B D)) add Emp) (Geos (Ang (An D B C)) add
Emp)" by (blast intro:Ang_weektrans Ang_rev Eq_rev)
  from P25 have P52 : "Def (Ang (An A B D))" by (simp add:Tri_to_Ang)
  from P26 have P53 : "Def (Ang (An A B C2))" by (blast intro:Ang_def_rev Ang_def_inv)
  then have P54 : "¬ Eq (Geos (Lin (Li B A)) add Emp) (Geos (Lin (Li B C2)) add Emp)" by (simp
add:Ang_def)
  from P16 have P55 : "Bet_Point (Se A C2) D" by (simp add:Bet_rev)
  from P4 P5 P54 P55 have P56 : "Ang_inside (An A B C2) D" by (simp add:Ang_inside_Bet_Point)
  have P57 : "¬ Bet_Point (Se A A) B" by (simp add:Bet_end_Point)
  from P3 P4 P5 P6 P19 P53 P56 P57 have P58 : "Ang_inside (An A B C) D"
    by (simp add:Ang_inside_trans)
  from P50 P51 P52 P58 show "∃ p. Cong (Geos (Ang (An A B p)) add Emp) (Geos (Ang (An p B C))
add Emp)
    ∧ Ang_inside (An A B C) p ∧ Def (Ang (An A B p)) ∧ Def (Ang (An p B C))" by blast
qed

end
```

謝辞

本論文は筆者が甲南大学大学院自然科学研究科知能情報学専攻博士後期課程に在籍中の研究成果をまとめたものである.

本論文の作成にあたり, 多くの方々にご指導ご鞭撻を賜りました.

元日本大学理工学部数学科教授の小林英恒先生には, 研究の初期の段階から多くの示唆を賜りました. ここに深謝の意を表します.

東邦大学理学部情報科学科の白柳潔教授には, 本論文の推敲段階において多くの助言をいただき, 感謝申し上げます.

甲南大学大学院自然科学研究科知能情報学専攻森元勘治教授には, 副査として適切なご助言を賜りました. 感謝申し上げます.

最後に, 甲南大学大学院自然科学研究科知能情報学専攻教員の皆様には, 多大なご助言を頂きました. ここに誠意の意を表します.

参考文献

[1] Frank Pfenning “Automated Theorem Proving”
https://www.cs.cmu.edu/ fp/courses/atp/handouts/atp.pdf

[2] 照井一成 「コンピュータに証明できること・できないこと」,
https://www.kurims.kyoto-u.ac.jp/ terui/suugakuseminar.pdf

[3] A. レオナルド 「定理証明支援系に基づく形式検証-近年の実例の紹介と Coq 入門-」,
http://id.nii.ac.jp/1001/00100774

[4] 「数学を教える・学ぶための GeoGebra」,
https://www.geogebra.org

[5] “The Coq Proof Assistant” https://coq.inria.fr/

[6] Fitzpatrick R, EUCLID ’S ELEMENTS OF GEOMETRY,
http://farside.ph.utexas.edu/Books/Euclid/Elements.pdf#search=%27Euclid+Elements%27

[7] “Isabelle” https://isabelle.in.tum.de/

[8] “Archive of Formal Proofs” https://www.isa-afp.org/

[9] D. ヒルベルト著, 中村幸四郎訳, 幾何学基礎論, 筑摩書房, 1969.

[10] 白柳潔, 宮本健司, 関川浩, 山本航,「人間らしい初等幾何証明における角の取り扱いについて」, 数理解析研究所講究録 1456 巻 2005 年 90-99.