JOURNAL OF COMPUTATIONAL INNOVATION AND ANALYTICS
http://e-journal.uum.edu.my/index.php/jcia

How to cite this article:
Iwama, F., & Takahashi, T. (2022). On the proof of the theorems of foundations of geometry using Isabelle/HOL. *Journal of Computational Innovation and Analytics, 1*(2), 45-69. https://doi.org/10.32890/jcia2022.1.2.3

ON THE PROOF OF THE THEOREMS OF FOUNDATIONS OF GEOMETRY USING ISABELLE/HOL

[1]Fumiya Iwama & [2]Tadashi Takahashi
Graduate School of Natural Science,
Konan University, Japan

[2]*Corresponding Author: takahasi@konan-u.ac.jp*

Received: 4/9/2021 Revised: 27/4/2022 Accepted: 25/5/2022 Published: 31/7/2022

ABSTRACT

Isabelle/HOL is a generic proof assistant. Using Isabelle/HOL requires insight into procedures as well as into the concepts involved. In addition, how a computer manages procedures can affect mathematical concepts. Use of Isabelle/HOL can correct a current weakness in mathematical studies. The advantage of the theorem proving support system represented by Isabelle/HOL is that it mechanically guarantees the "correctness" of both human-written programs and mathematical proofs. It can allow us to clearly understand mathematical concepts and can minimize the burden of operation opportunities. However, in order to take advantage of its high versatility and reliability, the problem that all certification procedures must be clearly formalized when creating certification must be overcome. "Foundations of Geometry" is a book on mathematics written by Hilbert in 1899. The book is famous as the most rigorous study of the axiom system of Euclidean geometry by axioms and formalism. When we tried to implement Hilbert's axioms in Isabelle/HOL, the proofs based on human cognition hindered the implementation. The purpose of this paper is "correctly" reconstruct the proofs as automated theorem proving. We are aiming to implement them "accurately" on Isabelle/HOL and have done so for many of them. This is the originality of this study.

Keywords: ATP, Foundations of Geometry, Hilbert's axioms, Isabelle/HOL.

INTRODUCTION

Isabelle/HOL is a generic system for implementing logical formalisms and is a specialization of Isabelle for higher-order logic (HOL). HOL can express most mathematical concepts, with functional programming being just one particularly simple and ubiquitous instance (Nipkow et al., 2021). Research on formalizing abstract algebra in Isabelle/HOL is based on Kobayashi et al. (2005). This research focuses on teaching mathematics to mathematics students, and in particular, on training students in the art of proving (Kobayashi et al., 2005). The following example combines methods of automated theorem proving and also integrates programming in a natural way (Takahashi & Kobayashi, 2006).

Example (Kobayashi et al., 2005)

A set of homomorphisms from a polynomial ring **R** = **S**[X] to a polynomial ring **A** = **B**[Y] is defined as Polyn_Hom::"[('a, 'm) RingType_scheme, ('a, 'm1) RingType_scheme, 'a, ('b, 'n) RingType_scheme, ('b, 'n1) RingType_scheme, 'b] => ('a => 'b) set"
(" (pHom _ _ _ , _ _ _)" [67, 67, 67, 67, 67, 68] 67)
"(pHom R S X, A B Y == {f. f ∈ rHom R A ∧ f' (carrier S) ⊆ carrier B ∧ f X = Y"}

Using ordinary mathematical expressions, we can write this lemma as follows.

Lemma pHom_mem;

Let **R** be a polynomial ring **S**[X] and let **A** be a polynomial ring **B**[Y]. If f is a ring homomorphism of **S** to **B**, then f is uniquety extended to a homomorphism F of **S**[X] to **B**[Y] such that
$F (a_0 + a_1 X + . . . + a_n Xn) = (fa_0) + (fa_1) Y + . . . + (fa_n) Yn.$

This kind of research aims at extending current computer systems using facilities for supporting mathematical proving. The system consists of a general higher-order predicate logic prover and a collection of special provers. The individual provers imitate the proof style of human mathematicians and produce human-readable proofs in natural language presented in nested cells. In contrast, in this article, we tried to reproduce Hilbert's axiom on Isabelle/HOL. As a result, we found that there are some small parts that are not clearly proven, relying on human recognition. Even if it is obvious to human perception, there are some parts that require detailed construction to be implemented in automated theorem proving.

COMPOSITION

Hilbert (1899/1902) made a rigorous reconstruction of Euclidean geometry in Chapter 1 of his work. There, five types of axioms are listed and 32 theorems are proved. The axiom group consists of 15 axioms: Incidence (I1 to I3), Order (II1–II4), Congruence (III1–III5), Parallels (IV) and Continuity (V1&V2) (Coupling axioms related to space geometry I4 to I8 are excluded from the present study). These axioms and theorems are referred to as HaI-1 to HaV-2 and Ht-1 to Ht-32, respectively, herein. In Hilbert's axiom system, basic concepts such as points and lines are treated as undefined terms, and only their relationships are defined by the axioms. In addition, HaV-2 and Ht-32 stipulate that the Euclidean plane is essentially equivalent to the real plane R2, ensuring that the axiom system is categorical (Nishimura, 2016). In this paper, we show some derived lemmas, axioms and theorems necessary for them. This is close to the actual Isabelle/HOL implementation file composition. The following axioms, theorems and definitions are taken from Nakamura (1930/1969).

HaI-1

For two points *A* and *B*, there is always at least one line connecting to each of these two points.

HaI-2

For two points A and B, there is no more than one line connecting to each of these two points.

We write $AB = a$ or $BA = a$. Instead of "contains", we may also employ other forms of expression; for example, we could say "*A* lies upon *a*", "*A* is a point of *a*", "*a* goes through *A* and through *B*", "*a* joins *A* to *B*", etc. If A lies upon a and at the same time upon another line b, we also make use of expressions like the following: "The lines *a* and *b* have the point *A* in common." The above two axioms show the uniqueness of a line. Therefore, consider the following lemma.

Lemma 1

When two different points A, B exist on two lines a, b, then a, b are the same line. Also, when two lines are the same line, if a point C exists on one, C also exists on the other.

By HaI-2, it is clear that a line that shares two different points A and B is unique, so the lemma is easily derived.

HaI-3

There are always at least two points on a line. There are at least three points that are not on a line.

Definition

Points on a line have a certain relationship with each other. The expression "between points" is used to describe this situation.

HaII-1

If point B is between points A and C, then A, B, and C are three different points on one line, and B is also between C and A.

HaII-2

For two points A and C, there is always at least one point B on the line AC, and C is between A and B.

In this paper, "point *B* is between points *A* and *C*" is hereafter written as "*Bet* (*A, C*) *B*". This axiom guarantees the three points are different, their existence on the same line, and that if *Bet* (*A, C*) *B*, then *Bet* (*C, A*) *B*.

HaII-3

Of any three points on one line, there are no more than one that can be between the other two points.

Lemma 2

Given three different points *A, B* and *C* on a line, if *Bet* (*A, C*) *B* then ¬Bet (*A, B*) *C* ∧ ¬Bet (*B, C*) *A*.

Proof

If *Bet* (*A, B*) *C*, then each of *B* and *C* is between the other two points, and there is more than one such point. Therefore, it cannot be *Bet* (*A, B*) *C*. The same is true for *Bet* (*B, C*) *A*. Therefore, the lemma is derived.

Definition

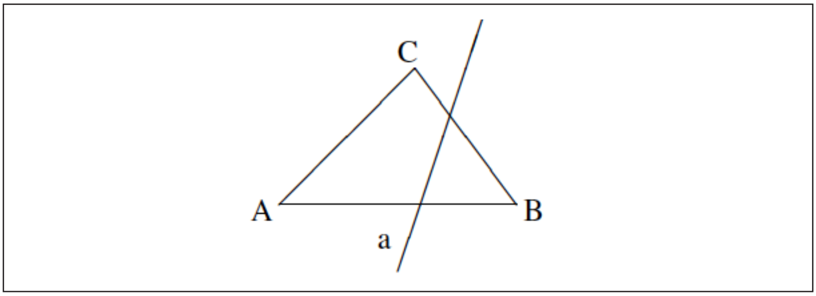
Consider two points *A, B* on a line *a*, and refer to the combination of these two points as a "segment", expressing it as *AB* or *BA*. A point between *A* and *B* is a point of the segment *AB*, also called an interior point of *AB*, and points *A, B* are the endpoints of *AB* rather than interior points of *AB*. Let all other points be the outer points of *AB*.

HaII-4

For three points *A*, *B* and *C* are not on one line, and line *a* does not pass through any of *A*, *B* and *C*, if *a* passes through the point of segment *AB* then *a* passes through the point of segment *AC* or segment *BC*.

Figure 1

HaII-4



This axiom can also be expressed as follows: "If a line passes through the inside of a triangle, it enters the outside again. Also, it does not pass through more than two sides." (However, since polygons have not been defined yet at this point, this expression is only provided for intuitive understanding.) Some theorems have been derived from the axioms presented so far. Among them, the ones related to this paper are given below.

Ht-4

Of any three points *A, B and C* on a line, there is always one point that is between the other two points.

Ht-5

Given any four points on a line, expressed as *A, B, C* and *D*, it is always possible to have "*B* is between *A* and *C*, and *A* and *D*" or "*C* is between *A* and *D*, and *B* and *D*".

Ht-5 is a theorem that guarantees that when four different points exist on a line, ordering them is possible. In the proof, it is stated that the positions of these four points can be distinguished by appealing to HaII-3 and Ht-4 as follows. When there are four different points on a line, first focus on three of them. From Ht-4, there is always one point that is between the other two points, so let *Q* be such a point, and let *P* and *R* be the other two points. If the fourth given point is *S*, the position of each point is distinguished into one of the following five cases.

(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*.

Hereafter, for four different points *P, Q, R* and *S* on a line such that *Bet* (*P, R*) *Q*, depending on the case, this is written as Ht-5 (1 to 5). Furthermore, the following two lemmas have been shown, and each case is covered by one of the two lemmas.

Lemma 3

Let *A, B, C* and *D* be four different points on one line.
Then, *Bet* (*A, C*) *B and Bet* (*B, D*) *C* ==> *Bet* (*A, D*) *B and Bet* (*A, D*) *C*.

Lemma 4

Let *A, B, C* and *D* be four different points on one line.
Then, *Bet* (*A, C*) *B and Bet* (*A, D*) *C* ==> *Bet* (*A, D*) *B and Bet* (*B, D*) *C*.

The proofs of these lemmas are provided in [4]. For cases Ht-5 (1 to 5), if the four points *P, Q, R* and *S* correspond to *A, B, C* and *D* as needed, the results are as follows.

(*P, Q, R, S* = *A, B, C, D* means *P* = *A*, *Q* = *B*, *R* = *C*, *S* = *D)*
[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* (By HaII-2, *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* (By HaII-2, *Bet* (*B, D*) *C* ∧ *Bet* (*A, C*) *B*).

It is shown that Ht-5(1 to 4) are covered by Lemma4, and Ht-5(5) is covered by Lemma3, so Ht-5 is proved.

IMPLEMENTATION

We contributed the Isabelle/HOL program file created in this study to the "Archive of Formal Proofs" (https://www.isa-afp.org). In the following programs, parts not related to the contents of this paper are omitted. Download the full program from https://www.isa-afp.org/entries/Foundation_of_geometry.html , and the matching points are on pp. 1-3, 5-6, 28-29, 42-46.

Theory Incidence imports Main begin

```
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"

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)"

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"

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"
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)"

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)"

The following proof corresponds to Lemma 3 of this paper.

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

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_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)

∃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)

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

```
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
```

CONCLUSION

In order to take advantage of its high versatility and reliability, the problem that all certification procedures must be clearly formalized when creating certification must be overcome (Reynald, 2014). In Hilbert’s axiom system, there are many places where it seems that define was not clearly stated because it would be clear from human recognition, such as “Are the same element and the congruent element also congruent with each other?”. Also, it is claimed that “Relationship between points in different areas” and “Guarantee of triangle” can be easily derived from other axioms and theorems, and no definite proof is shown. Moreover, there are some things that are expressed as if they were included in the axiom, such as “Nature of the large and small relationship of angles”, and are not mentioned. Currently we are aiming to implement them “accurately” on Isabelle/HOL and have already done so for many of them. An Isabelle file contributed to “Archive of Formal Proofs”. contains the existing theorems from Ht-1 to Ht-26 and more than 50 lemmas and theorems needed to implement them. Creating this file achieved many of the purpose of this study. However, we are not allowed to add definitions on our own. Therefore, the “correct” implementation of Ht-23, which presupposes the undefined concept “Large and small relationship of segments”, and Ht-24, which uses Ht-23 for its proof, is currently impossible. Regarding this, we are considering the validity of the method of deriving the proof of a theorem equivalent to Ht-24 from other axioms and theorems and paradoxically defining “Large and small relationship of segments”. The authors declare no conflict of interest.

ACKNOWLEDGMENT

This work is supported in part by MEXT, Japan.

REFERENCES

Archive of Formal Proofs. https://www.isa-afp.org

Hilbert, D. (1902). *The Foundations of Geometry.* (E. J. Townsend, Trans.). https://math.berkeley.edu/~wodzicki/160/Hilbert.pdf

Hilbert, D. (1969). *The Foundations of Geometry.* (Nakamura, K, Trans.). Tokyo: shimizukobundo (Original work published 1930)

Iwama, F. (2021, November 22). *Foundation of geometry in planes, and some complements: Excluding the parallel axioms.* https://www.isa-afp.org/entries/Foundation_of_geometry.html

Kobayashi, H., Suzuki, H., & Ono, Y. (2005). *Formalization of Henzel's Lemma.* 18th International Conference, TPHOLs, Oxford, UK.

Nipkow, L., Paulson, C., & Wenzel, M. (2021). *A Proof Assistant for Higher - Order Logic*. https://isabelle.in.tum.de/doc/tutorial.pdf

Nishimura, Y. (2016). *A reconstruction of Euclidean geometry along Elements.* https://core.ac.uk/download/pdf/76169358.pdf

Reynald, A. (2014). *Formal Verification using Proof-assistants - Survey of Recent Applications and Introduction to Coq* -. http://id.nii.ac.jp/1001/00100774

Takahashi, T. and Kobayashi, H. (2006). *The Effect of the Theorem Prover in Cognitive Science*. https://link.springer.com/chapter/10.1007/11758501_139