Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 218492, 6 pages
http://dx.doi.org/10.1155/2013/218492

Research Article

Formalization of Linear Space Theory in the Higher-Order Logic Proving System

Jie Zhang,[1] Danwen Mao,[1] and Yong Guan[2]

[1] *College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China*
[2] *College of Information Engineering, Capital Normal University, Beijing 100048, China*

Correspondence should be addressed to Jie Zhang; jzhang@mail.buct.edu.cn

Received 8 March 2013; Accepted 1 April 2013

Academic Editor: Xiaoyu Song

Theorem proving is an important approach in formal verification. Higher-order logic is a form of predicate logic that is distinguished from first-order logic by additional quantifiers and stronger semantics. Higher-order logic is more expressive. This paper presents the formalization of the linear space theory in HOL4. A set of properties is characterized in HOL4. This result is used to build the underpinnings for the application of higher-order logic in a wider spectrum of engineering applications.

1. Introduction

Linear space is a core theory of linear algebra. It has a wide spectrum of applications, such as cryptography, pattern recognition, and signal processing and communications. In order to formally model and analyze designs with linear space in a formal logic, it is necessary to achieve the formalization of linear space. The HOL theorem prover was developed in 1984, and has a wide range of applications in various fields [1–4]. HOL4 is the latest version of HOL, which provides a wide collection of theories and libraries. However, there is no formalization of linear space in HOL4. This paper fills this gap in the formalization of the linear space in HOL4.

2. Preliminaries in HOL

In HOL4, the theories and libraries are categorized as boolean logic, temporal logic, natural numbers, real numbers, lists, and so forth. Each theory consists of types, definitions, and theorems. Theorems are established based on rigorous mathematical derivation. A library is usually a collection of theories, proof tools (such as tactics, tactical and simplification sets) and proof procedures [5].

In HOL4, the following steps are involved in the creation of a new theory.

(1) *New Types*. The HOL4 system is based on higher-order logic. Any variable in the higher-order logic has a type [6]. When establishing a theory, one has to define new types of variables that do not exist in the system.

(2) *Formal Definitions*. Definition is a process of modeling. Formal definitions affect the proof of properties and theorems.

(3) *Formal Proof of Properties and Theorems*. A proof is related to the choice of appropriate proof structures, inference rules, and tactics. The final proof result will be saved as a theorem of type ":thm". These rigorous steps of mathematical derivation are checked in HOL.

3. Formalization of Linear Space Theory in HOL4

3.1. Definitions (Field). Let S be a nonempty subset of the complex number set. S is a field if

(1) 0 and $1 \in S$,

(2) closure property: for $a \in S$ and $b \in S$, $a+b \in S$, $a-b \in S$, $ab \in S$ and $b/a \in S (a \neq 0)$ [7].

The definition of a field is associated with the existing theories of HOL4. For example, "a nonempty subset of the

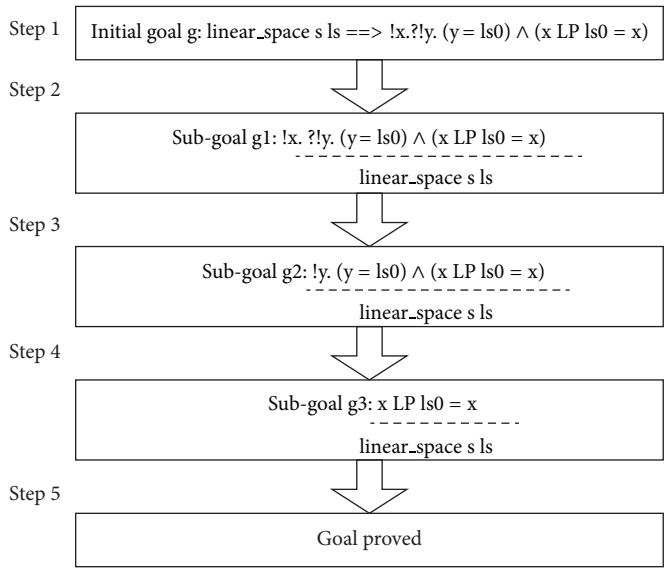

Figure 1: Proof of Property 1.

complex number set" is associated with the Set Theory and the Complex Theory of HOL4. Therefore, the formalization of a field in HOL4 will be based on these two existing theories of HOL4.

3.2. Axiomatization of a Field. A field S can be characterized by the following axioms.

Axiom 1 (Complex Number). S is a subset of the complex number set.

Axiom 2 (Nonempty). S is a nonempty set.

Axiom 3 (Membership). S includes element 0 and element 1, and all the elements of S meet the closure property of four operations.

The formalization of Axiom 3 in HOL4 is given as shown in Box 1.

The definition of a field can be expressed by the formal description of the above three axioms. The formalization of Axioms 1 and 2 are "is_complex_subset" and "nf_not_empty". The formalization of a field in HOL4 is given as shown in Box 2.

3.3. Definitions (Linear Space). Let V be a nonempty set and S a field. An addition operation is defined on the elements of V. For two arbitrary elements x and y of V, there is a unique element z in V, denoted as $z = x + y$. A scalar multiplication operation is defined on the numbers of S with the elements of V. For an arbitrary element k of S and an arbitrary element x of V, there is a unique element y in V, denoted as $y = kx$. If addition and multiplication operations meet the following eight rules, then V is called a linear space on the field S.

(1) $x + y = y + x$

(2) $(x + y) + z = x + (y + z)$

(3) There is an element in V called "zero", denoted as θ. For an arbitrary element x, $x + \theta = x$

(4) For an arbitrary element x, there is a element y in V so that $x + y = \theta$

(5) If $1 \in S$, then $1x = x$

(6) If $k \in S$, $l \in S$ and $x \in V$, then $k(lx) = (kl)x$

(7) If $k \in S$, $l \in S$ and $x \in V$, then $(k + l)x = kx + lx$

(8) If $k \in S$, $x \in V$ and $y \in V$, then $k(x + y) = kx + ky$ [8].

A linear space is composed of a nonempty set V and a field S under addition and multiplication operations with eight algebraic rules. The Set Theory in HOL4 is used to complete the formulation of the new theory.

3.4. Axiomatization of Linear Space. A linear space M can be characterized by the following axioms.

Axiom 1. M is a nonempty set.

Axiom 2. M is a linear space on a field S.

Axiom 3. The elements of M meet the uniqueness and closure property of addition.

Axiom 4. The elements of M meet the uniqueness and closure property of multiplication.

Axiom 5. The elements of M meet the associative law of addition.

Axiom 6. The elements of M meet the commutative law of addition.

Axiom 7. M contains the special element zero.

```
-g "linear_space ^s^ls ==> !x:'a.?!y:'a. (y = ls0) ∧ (x LP ls0 = x)";
> val it =
      Proof manager status: 1 proof.
      1. Incomplete goalstack:
          Initial goal:
          linear_space s ls ==> !x. ?!y. (y = ls0) ∧ (x LP ls0 = x)
       : proofs
```

Algorithm 1: Initial goal g.

```
- e (DISCH_TAC)
OK..
1 subgoal:
> val it =
      !x. ?!y. (y = ls0) ∧ (x LP ls0 = x)

        linear_space s ls
       : proof
```

Algorithm 2: The subgoal g1.

```
- e (GEN_TAC);
OK..
1 subgoal:
> val it =
      ?!y. (y = ls0) ∧ (x LP ls0 = x)

        linear_space s ls
       : proof
```

Algorithm 3: The subgoal g2.

Axiom 8. M contains the defined negative elements.

Axiom 9. The elements of M meet the rule of multiplication with element 1 in S.

Axiom 10. The elements of M meet the associative law of multiplication.

Axiom 11. The elements of M meet the left distributive law of multiplication.

Axiom 12. The elements of M meet the right distributive law of multiplication.

Let LM, LP, and LN be the addition, multiplication, and negation operators of linear space. The formal description of the definition of linear space can be converted into the formal descriptions of the above 12 axioms. For example, consider Axiom 3; its formal description in HOL4 is as shown in Box 3.

The unique existing quantifier "?!" is used in the above description to indicate the uniqueness of the results of addition.

This axiom is named "ls_add". The formal descriptions of the other 11 axioms are given in a similar manner. The 12 axioms are then connected by "∧" to indicate that they are required at the same time.

The formal description of the definition of linear space in HOL4 is described as shown in Box 4.

3.5. New Types. We introduce new types for the field and the linear space. In HOL4, the existing type ":complex" is used as the type of the elements of the field, and the polymorphic type ":'a" is used as the type of the elements of the linear space. Since the field and linear space are both sets, their types are defined by the use of the existing type of set. The definitions of the two types are as shown in Box 5.

The type of a number field, "num_field", is defined as a complex set ":complex -> bool". The type of the linear space, "linear_space", is defined as a polymorphic set ":'a -> bool".

3.6. Properties of Linear Space. Properties of linear space can be derived in HOL4. For example, the following properties are formalized in HOL4.

Property 1. The element "zero" in a linear space is unique.

Property 2. For an arbitrary element x in a linear space, the negative element of x is unique, and denoted as $-x$.

Property 3. For arbitrary elements x, y and z in a linear space, if $x + y = x + z$, then $y = z$.

Property 4. For arbitrary elements x, y and z in a linear space, if $x + y = z$, then $x = z - y$.

Property 5. For an arbitrary element k in a number field, $k\theta = \theta$.

Property 6. For an arbitrary element x in a linear space, $0x = \theta$.

Property 7. For an arbitrary element x in a linear space, $(-1)x = -x$.

```
- e (RW_TAC arith_ss [EXISTS_UNIQUE_CONV “?!y. (y = ls0) ∧ (x LP ls0 = x)”]);
<<HOL message: inventing new type variable names: ′a, ′b>>
OK..
1 subgoal:
> val it =
      x LP ls0 = x

       linear_space s ls
      : proof
```

Algorithm 4: The subgoal g3.

```
- e (RW_TAC arith_ss [zero_def]);
OK..
Goal proved.
  [linear_space s ls] |- x LP ls0 = x
Goal proved.
  [linear_space s ls] |- ?!y. (y = ls0) ∧ (x LP ls0 = x)
Goal proved.
  [linear_space s ls] |- !x. ?!y. (y = ls0) ∧ (x LP ls0 = x)
> val it =
      Initial goal proved.
        []|- linear_space s ls ==> !x. ?!y. (y = ls0) ∧ (x LP ls0 = x): proof
```

Algorithm 5: The proof result.

```
> val zero_unique =
      [] |- linear_space s ls ==> !x. ?!y. (y = ls0) ∧ (x LP ls0 = x): thm
```

Algorithm 6: Theorem “zero_unique”.

```
val s = “s: num_field”;
val membership = “0c IN s ∧ 1c IN s ∧
                  !a:complex b:complex. if a IN s ∧ b IN s
                  then (a + b IN s) ∧ (a − b IN s) ∧ (a ∗ b IN s) ∧ (if a < > 0 then b/a IN s else a = 0c)
                  else F”;
```

Box 1

```
val is_num_field_def = Define′
    num_field ^s = ^is_complex_subset ∧ ^nf_not_empty ∧ ^membership′.
```

Box 2

```
val ls = “ls: ′a linear_space”;
val ls_add = “!x:′a y:′a. x IN ls ∧ y IN ls ==> ?!z:′a. (z = x LP y) ∧ z IN ls ”;
```

Box 3

```
val is_linear_space_def = Define′
    linear_space ^s^ls = num_field ^s ∧ ^ls_not_empty ∧ ^ls_add ∧ ^ls_mul ∧ ^ls_plus_assoc
                         ∧ ^ls_plus_sym ∧ ^ls_zero_def ∧ ^ls_opp_def ∧ ^ls_mult_one
```

Box 4

```
val_ = type_abbrev (“num_field”, “:complex -> bool”);
val_ = type_abbrev (“linear_space”, “:′a -> bool”);
```

Box 5

Property 8. For an arbitrary element k in a number field and arbitrary elements x, y in a linear space, if $kx = y$, $k \neq 0$, then $x = (1/k)y$ [7, 8].

3.7. Proof of Properties of Linear Space. We complete the proof of 37 properties related to linear space, and create 37 theorems in the linear space theory.

Without loss of generality, we show the proof process for Property 1 (i.e., the element “zero” in a linear space is unique).

The property's proof uses the goal-oriented proof method, and its process is shown in Figure 1.

The proving process consists of five steps.

Step 1. Give the initial goal of Property 1. By using the element “zero” predefined during the formal modeling of linear space, the initial goal of this property is formally described as: “linear_space ^s^ls ==> !x:′a.?!y:′a. (y = ls0) ∧ (x LP ls0 = x)”. “ls0” is the symbol of element “zero” in the expression. Algorithm 1 shows the result of the input of the initial goal in HOL4.

Step 2. Start from the initial goal, and assume that the property desired is correct. Then use the tactic DISCH_TAC to simplify the initial goal g, moving the antecedent of the implicative goal g into the assumptions to get sub-goal g1. The process of Step 2 is shown in Algorithm 2.

Step 3. Use the tactic GEN_TAC to simplify sub-goal g1, thereby stripping the outermost universal quantifier from the conclusion of sub-goal g1 to obtain sub-goal g2. Algorithm 3 shows the process of Step 3.

Step 4. For sub-goal g2, use the function EXISTS_UNIQUE_CONV existing in HOL4 to generate a theorem, that is, “(?!y· (y = ls0) ∧ (x LP ls0 = x)) <=> (?y· (y = ls0) ∧ (x LP ls0 = x)) ∧ !y y′. ((y = ls0) ∧ (x LP ls0 = x)) ∧ (y′ = ls0) ∧ (x LP ls0 = x) ==> (y = y′)”. Then apply the tactic RW_TAC to g2 by using the theorem, so as to get the simplified sub-goal g3. The process of Step 4 is shown in Algorithm 4.

Step 5. Apply the tactic RW_TAC to the sub-goal g3 by using an axiom of the definition of linear space, that is, “zero_def: [linear_space *s* ls] |- !x. x LP ls0 = x”, so as to prove the sub-goal g3 in a direct manner. The HOL4 system then returns the proved sub-goals g2 & g1 one by one, until the initial goal g is returned. The proof result is given in Algorithm 5.

Finally, this paper generates a theorem named “zero_unique” and saves it in the linear space theory by using a theorem saving tool “store_thm”. The result is shown in Algorithm 6.

According to the above processes, when carrying out a formal proof by the goal-oriented proof method, the first requirement is to accurately describe the initial goal, and then select appropriate tactics against the specific characteristics of goals in different stages and, finally, to constantly simplify the goals by using the proved theorems and tactics so as to complete the whole formal proof.

4. Conclusion

We have presented the formal modeling of the linear space and the formal proof of its properties in HOL4. Our results enriched the existing theories of HOL4, thus laying the underpinnings for theorem proving based verification with the linear space theory for a broad range of applications. Further improvements include the formalization of linear combinations, linear dependence, linear independence, and subspace. The formalization of these theories will contribute to a more powerful formal verification engine in terms of the linear space theory.

Acknowledgment

This research was supported by an international scientific and technological cooperation project (2011DFG13000) backed by The Ministry of Science and Technology of China.

References

[1] J. R. Harrison, *Theorem proving with the real numbers [Ph.D. thesis of Philosophy]*, University of Cambridge, 1996.

[2] A. Habibi, S. Tahar, and A. Ghazel, “Formal modelling of the ADSP-2100 processor using HOL,” in *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 614–619, May 2002.

[3] J. Harrison, “Floating point verification in HOL light: the exponential function,” *Formal Methods in System Design*, vol. 16, no. 3, pp. 271–305, 2000.

[4] B. Akbarpour, S. Tahar, and A. Dekdouk, “Formalization of fixed-point arithmetic in HOL,” *Formal Methods in System Design*, vol. 27, no. 1-2, pp. 173–200, 2005.

[5] K. Slind and M. Norrish, “A brief overview of HOL4,” in *Theorem Proving in Higher Order Logics*, vol. 5170 of *Lecture Notes in Computer Science*, pp. 28–32, Springer, Berlin, Germany, 2008.

[6] Cambridge Research Center of SRI International, "The HOL System TUTORIAL (for HOL Kananaskis-7)," 2011, http://cdnetworks-kr-1.dl.sourceforge.net/project/hol/hol/kananaskis-7/kananaskis-7-tutorial.pdf.

[7] W. Qiu, *Advanced Algebra*, Higher Education Press, Beijing, China, 1996.

[8] Yizhong Lan, *Simple Tutorial for Advanced Algebra*, Peking University Press, Beijing, China, 2002.