Applied Mathematics & Information Sciences
*An International Journal*



# Formalization of the Complex Number Theory in HOL4

*Zhiping Shi*[1,2,3], *Liming Li*[1], *Yong Guan*[1], *Xiaoyu Song*[4], *Minhua Wu*[1] *and Jie Zhang*[5]

[1] Beijing Key Laboratory of Electronic System Reliability Technology, Capital Normal University, Beijing, China
[2] State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[3] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China
[4] Electrical and Computer Engineering, Portland State University, USA
[5] College of Information Science & Technology, Beijing University of Chemical Technology, Beijing, China

Received: 21 Jan. 2012; Revised 27 Aug. 2012; Accepted 28 Aug. 2012
Published online: 1 Jan. 2013

**Abstract:** In this paper, the theory of complex numbers is formalized and the theorem library of complex numbers is embedded in HOL4, the theorem prover of High Order Logics. The theorem library introduces a data type $\mathbb{C}$ by an $\mathbb{R} \times \mathbb{R}$ type abbreviation, and defines arithmetic operations of complex numbers in terms of group and field theories. Moreover, the polar and exponential forms are provided for simplifying the applications in control theory and signal analysis. We define the scalar multiplication of complex numbers and prove some properties about $\mathbb{R}$-module of complex numbers. The theorem library extends the scope of application of HOL4. The developed complex number theory has been released in HOL4 Kananaskis-7.

**Keywords:** complex number theory, HOL, theorem proving.

## 1. Introduction

Theorem proving has been considered as one of effective formal verification methods [1]. Systems need to be modeled formally before they are verified by theorem provers, and theorem provers work based on logic theorem libraries of mathematics. The more the mathematic theorem libraries are, the wider the scope of application of the theorem provers becomes. The complex number theory is the basis of control theory, signal analysis, quantum mechanics, relativity, fluid dynamics and so on. It is significant to construct the logic theorem library of complex numbers for theorem provers.

Almost all the theorem provers have natural numbers and real numbers [3,4]. The complex numbers have been constructed in several theorem provers. Mizar is the earliest prover of constructing complex numbers [6] and many research results related to complex numbers have been published [7,8]. A completely constructive proof has also been formalized in Coq [5]. Moreover, the PVS dump file provides a complex numbers library [10]. John Harrison introduced the complex numbers in HOL-Light [2]. There are definitions of the complex numbers and their arithmetic operators in ProofPower-HOL [9] and Isabelle/HOL [11].

HOL4 is the latest version of High Order Logic (HOL), featuring a number of novelties compared to its predecessors. HOL4 is also the version of the system supported by the international HOL community. The system provides a wide collection of theorem libraries: Booleans, pairs, sums, options, numbers ($\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, fixed point, floating point, n-bit words), lists, lazy lists, character strings, partial orders, monad instances, predicate sets, multisets, finite maps, polynomials, probability, abstract algebra, elliptic curves, lambda calculus, program logics (Hoare logic, separation logic), machine models (ARM, PPC, and IA32), temporal logics ($\omega$-automata, CTL, $\mu$-calculus, PSL) and so on [1].

However, until now, there hasn't been any theorem library of complex numbers, which affects the scope of application of HOL4. Moreover, the polar and exponential forms are provided for simplifying the applications in control theory and signal analysis. Actually both the forms are presented as multiplying a real number with a complex number. Thus we define the scalar multiplication of complex numbers for the theorem prover. To the best of our knowledge, it is the first time to have this result.

* Corresponding author: e-mail: shizhiping@gmail.com

In this paper, we construct the theorem library of complex numbers in HOL4. We start from defining the complex data type. Then we systematically give the formal logic description and proof of relevant theorems following group, field and module algebraic systems. And then we discuss the concept of conjugate complex and the related properties. And also we introduce the polar form and the exponential form of complex numbers, which are particularly common in some areas.

## 2. the Basic Definitions of Complex Numbers

A complex number is in the form of $x + iy$, where $x$ and $y$ are real numbers and $i$ is a constant postulated to satisfy $i^2 = -1$. Thus, the complex numbers are isomorphic to the pairs of real numbers. Pair and real theories have already been developed in HOL4, so it is easy to define the complex number type $\mathbb{C}$ by just using an $\mathbb{R} \times \mathbb{R}$ type abbreviation. A complex number $a + ib$ can be represented as a pair of (a, b).

```
val _= type_abbrev (complex,″:real # real ″);
```

The real part and the imaginary part are the basic elements of a complex number. Their definitions are as follows:

**Definition 1.** $\forall z \in \mathbb{C}, RE\ z = FST\ z$

**Definition 2.** $\forall z \in \mathbb{C}, IM\ z = SND\ z$

The projection functions return the corresponding components of a pair. The function $FST$ returns the first component, and $SND$ returns the second. Thus $RE$ and $IM$ are both the type:$\mathbb{C} \to \mathbb{R}$.

Real numbers are subset of complex numbers, i.e., any real number can be transformed into a complex number, the imaginary part of which equals $0$.

**Definition 3.** $\forall x \in \mathbb{R}$, *complex_of_real x = (x, 0r)*

Note: The suffix r means $\mathbb{R}$. The type of this function is $\mathbb{C} \to \mathbb{R}$.

Specifically the imaginary unit $i$ is defined as

**Definition 4.** *i = (0r, 1r)*

Two complex numbers are equal if and only if both the real parts and the imaginary parts are equal respectively, which is the most basic decision theorem.

**Theorem 1.** *COMPLEX_RE_IM_EQ:*$\forall z, w \in \mathbb{C}$, $(z = w) \Leftrightarrow (RE\ z = RE\ w) \wedge (IM\ z = IM\ w)$

Theorem 1 can be proven using the definitions of $RE$ and $IM$.

```
⊢ val COMPLEX_RE_IM_EQ = store_thm(
COMPLEX_RE_IM_EQ ,
″!z w. (z=w)¡=¿(RE z=RE w)∧(IM z=IM w)″,
REWRITE_TAC [RE, IM, PAIR_FST_SND_EQ ]);
```

(We start a theorem with " ⊢ " symbol in rectangles.)

## 3. The Field of Complex Numbers

In this section, the complex numbers are characterized as the field algebraic structure.

### 3.1. The Definitions of Basic Operation

We overload the common operators for $\mathbb{C}$ and $\mathbb{R}$.

Addition:

**Definition 5.** $\forall z, w \in \mathbb{C}$, $z + w = (RE\ z + RE\ w, IM\ z + IM\ w)$

Additive inverse (negation):

**Definition 6.** $\forall z \in \mathbb{C}, -z = (-RE\ z, -IM\ z)$

Multiplication:

**Definition 7.** $\forall z, w \in \mathbb{C}$, $z * w = (RE\ z * RE\ w - IM\ z * IM\ w, RE\ z * IM\ w + IM\ z * RE\ w)$

Multiplicative inverse (reciprocal):

**Definition 8.** $\forall z \in \mathbb{C}$, $inv\ z = (RE\ z/((RE\ z)^2+(IM\ z)^2), -IM\ z/((RE\ z)^2+(IM\ z)^2))$

Note: In the existing realTheory library of HOL4, $inv\ 0r = 0r$, so here we set $inv\ 0c = 0c$. The suffix $c$ means $\mathbb{C}$.

### 3.2. $\langle \mathbb{C}, + \rangle$ is an Abelian Group

Based on the definition of addition of complex numbers, we verify properties of the Abelian Group.

**Theorem 2.** *COMPLEX_ADD_CLOSURE:* $\forall z, w \in \mathbb{C}, z + w \in \mathbb{C}$

**Theorem 3.** *COMPLEX_ADD_ASSOC:*$\forall z, w, v \in \mathbb{C}$, $z + (w + v) = (z + w) + v$

**Theorem 4.** *COMPLEX_ADD_RID:*$\forall z \in \mathbb{C}, z + 0c = z$

**Theorem 5.** *COMPLEX_ADD_LID:*$\forall z \in \mathbb{C}, 0c + z = z$

**Theorem 6.** *COMPLEX_ADD_RINV:*$\forall z \in \mathbb{C}$, $z + -z = 0c$

**Theorem 7.** *COMPLEX_ADD_LINV:*$\forall z \in \mathbb{C}$, $-z + z = 0c$

The complex number $0c$ is the identical element of addition. The theorems above show that ¡$\mathbb{C}$, +¿ is a group. Easy to prove that it also satisfies commutative law, namely:

**Theorem 8.** *COMPLEX_ADD_COMM:*$\forall z, w \in \mathbb{C}$, $z + w = w + z$

It shows that ¡$\mathbb{C}$, +¿ is an Abelian group.

### *3.3. $\langle C - \{0c\}, \rangle$ is an Abelian Group*

Similarly, the following properties of multiplication are established:

**Theorem 9.** *COMPLEX_MUL_CLOSURE:* $\forall z, w \in \mathbb{C}, z * w \in \mathbb{C}$

**Theorem 10.** *COMPLEX_MUL_ASSOC:* $\forall z, w, v \in \mathbb{C}, z * (w * v) = (z * w) * v$

**Theorem 11.** *COMPLEX_MUL_RID:* $\forall z \in \mathbb{C}, z * 1c = z$

**Theorem 12.** *COMPLEX_MUL_LID:* $\forall z \in \mathbb{C}, 1c * z = z$

**Theorem 13.** *COMPLEX_MUL_RINV:* $\forall z \in \mathbb{C}, z \neq 0c \Rightarrow z * inv\ z = 1c$

**Theorem 14.** *COMPLEX_MUL_LINV:* $\forall z \in \mathbb{C}, z \neq 0c \Rightarrow inv\ z * z = 1c$

The complex number $1c$ is the unit element of the multiplication, namely, unitary element. The theorems above show that ¡$\mathbb{C}$ -{$0c$},*¿ is a group. And easy to prove the commutative law:

**Theorem 15.** *COMPLEX_MUL_COMM:* $\forall z, w \in \mathbb{C}, z * w = w * z$

So ¡$\mathbb{C}$ -{$0c$},* ¿ is an Abelian group.

### *3.4. $\langle C+, \rangle$ is a Field*

**Theorem 16.** *COMPLEX_ADD_LDISTRIB:* $\forall z, w, v \in \mathbb{C}, z * (w + v) = z * w + z * v$

**Theorem 17.** *COMPLEX_ADD_RDISTRIB:* $\forall z, w, v \in \mathbb{C}, (z + w) * v = z * v + w * v$

From what has been mentioned above, ¡$\mathbb{C}$ ,+,* ¿ is a field.

In addition, the subtraction operation of complex numbers can be defined using the addition and negation operators:

**Definition 9.** $\forall z, w \in \mathbb{C}, z - w = z + (-w)$

Similarly, the division operation of complex numbers can be defined using the multiplication and reciprocal operations:

**Definition 10.** $\forall z, w \in \mathbb{C}, z/w = z * inv\ w$

Based on these basic definitions and properties, many other operations and properties can be reduced. As an example, here illustrates the property of the addition of two ratios, $\frac{z}{w} + \frac{u}{v} = \frac{zv+uw}{wv}$ , denoted as COMPLEX_ADD_RAT, It is necessary to prove the following theorems before reaching the goal. The distributivity of division over addition, denoted as COMPLEX_DIV_ADD, is proved by rewriting the definition of division and the distributivity of multiplication over addition.

⊢ val COMPLEX_DIV_ADD = store_thm(
COMPLEX_DIV_ADD,
″!z w v :complex. z / v + w / v = (z + w) / v″,
REWRITE_TAC[complex_div,
GSYM COMPLEX_ADD_RDISTRIB]);

The properties about canceling the same factor of the operands of the division, denoted as COMPLEX_DIV_RMUL_CANCEL and COMPLEX_DIV_LMUL_CANCEL, can be implemented as follows.

⊢ val COMPLEX_DIV_RMUL_CANCEL = store_thm(
COMPLEX_DIV_RMUL_CANCEL,
″!v:complex z w. ∼(v = 0) ==¿ ((z * v) / (w * v) = z / w)″,
RW_TAC bool_ss [complex_div] THEN
Cases_on ′w = 0′ THEN
RW_TAC bool_ss [COMPLEX_MUL_LZERO,
COMPLEX_INV_0, COMPLEX_INV_MUL,
COMPLEX_MUL_RZERO, COMPLEX_EQ_LMUL,
GSYM COMPLEX_MUL_ASSOC] THEN
DISJ2_TAC THEN
ONCE_REWRITE_TAC [COMPLEX_MUL_COMM] THEN
ONCE_REWRITE_TAC [GSYM COMPLEX_MUL_ASSOC]
THEN RW_TAC bool_ss [COMPLEX_MUL_LINV,
COMPLEX_MUL_RID]);

⊢ val COMPLEX_DIV_LMUL_CANCEL = store_thm(
COMPLEX_DIV_LMUL_CANCEL,
″!v:complex z w. ∼(v = 0) ==¿ ((v * z) / (v * w) = z / w)″,
METIS_TAC [COMPLEX_DIV_RMUL_CANCEL,
COMPLEX_MUL_COMM]);

So, the property COMPLEX_ADD_RAT can be proved using these theorems.

⊢ val COMPLEX_ADD_RAT = store_thm (
COMPLEX_ADD_RAT,
″!z:complex w u v.
∼(w=0)∧ ∼(v=0)==¿(z/w+u/v=(z*v+w*u)/(w*v))″,
RW_TAC bool_ss [GSYM COMPLEX_DIV_ADD,
COMPLEX_DIV_RMUL_CANCEL,
COMPLEX_DIV_LMUL_CANCEL]);

## 4. R-module

Practically, real numbers often act as coefficients of complex numbers such as $r(\cos\theta + i\sin\theta)$ and $re^{i\theta}$. This is the scalar multiplication of complex numbers. The operands are $\mathbb{R}$ and $\mathbb{C}$, and the result is $\mathbb{C}$. The scalar multiplication of complex numbers has two styles, the left coefficient and

the right coefficient. The symbol "*" is overloaded to represent the operation.

**Definition 11.** $\forall k \in \mathbb{R}, z \in \mathbb{C}$,
$k * z = (k * RE\ z, k * IM\ z)$

**Definition 12.** $\forall k \in \mathbb{R}, z \in \mathbb{C}$,
$z * k = (RE\ z * k, IM\ z * k)$

Based on the definitions 11 and 12, the following theorems are verified:

**Theorem 18.** *COMPLEX_ADD_SCALAR_LMUL:*
$\forall a \in \mathbb{R}, z, w \in \mathbb{C}, a * (z + w) = a * z + a * w$

**Theorem 19.** *COMPLEX_SCALAR_LMUL_ADD:*
$\forall a, b \in \mathbb{R}, z \in \mathbb{C}, (a + b) * z = a * z + b * z$

**Theorem 20.** *COMPLEX_SCALAR_LMUL:*
$\forall a, b \in \mathbb{R}, z \in \mathbb{C}, a * (b * z) = (a * b) * z$

**Theorem 21.** *COMPLEX_SCALAR_LMUL_ONE:*
$\forall z \in \mathbb{C}, 1r * z = z$

Because the real number set $\mathbb{R}$ is a ring containing unitary element 1, and the Section 3.2 shows ($\mathbb{C}$, +) is an Abelian group, the theorems presented above educe that the complex number set $\mathbb{C}$ is a left $\mathbb{R}$-module. The two operands of the scalar multiplication are commutative, namely:

**Theorem 22.** *COMPLEX_SCALAR_MUL_COMM:*
$\forall k \in \mathbb{R}, z \in \mathbb{C}, z * k = k * z$

So the complex set $\mathbb{C}$ is a right $\mathbb{R}$-module. Therefore, $\mathbb{C}$ is $\mathbb{R}$-module.

The scalar multiplication of complex numbers is the base of the polar form of complex numbers, and many applications will occur in the later chapters. Now, we talk about an interesting property about the reciprocal of the scalar multiplication.

⊢ val COMPLEX_INV_SCALAR_LMUL = store_thm(
COMPLEX_INV_SCALAR_LMUL,
″!k:real z:complex.
∼(k = 0)∧ ∼(z = 0) ==¿ (inv (k*z) = inv k * inv z)″,
REWRITE_TAC [COMPLEX_0_THM, complex_inv,
complex_scalar_lmul,RE,IM, POW_MUL,
GSYM REAL_ADD_LDISTRIB, real_div,
REAL_INV_MUL] THEN
REPEAT STRIP_TAC THEN
′ ∼(k pow 2 = 0)′ by RW_TAC real_ss[POW_2,
REAL_ENTIRE] THEN
RW_TAC real_ss[REAL_INV_MUL] THEN
′inv (k pow 2) = inv k * inv k′ by
RW_TAC real_ss[POW_2,REAL_INV_MUL] THEN
ASM_REWRITE_TAC[REAL_MUL_ASSOC] THEN
REWRITE_TAC[REAL_ARITH
″!a b c:real. a * b * c * c = c * a * b * c″] THEN
RW_TAC real_ss[REAL_MUL_LINV,
REAL_MUL_COMM]);

## 5. Complex Conjugate

If the real parts of two complex numbers are equal and the imaginary parts are opposite, then the two complex numbers are mutually conjugate complex numbers. Conjugate is an important concept as to complex numbers. Complex conjugate is used in the rationalization of complex numbers and for finding the amplitude of the polar form of a complex number. Also it is involved when discussing many important properties of complex numbers. For example, conjugate complex numbers always are used to solve the dot product of complex vector and the unitary transformations. Complex conjugate is defined as follows:

**Definition 13.** $\forall z \in \mathbb{C}, conj\ z = (RE\ z, -IM\ z)$

Complex conjugate represents that two coordinates of the complex plane are symmetric to the real axis. There are many interesting properties about complex conjugate. First several computing properties of complex conjugate are given.

**Theorem 23.** *CONJ_ADD:*
$\forall z, w \in \mathbb{C}, conj\ (z + w) = conj\ z + conj\ w$

**Theorem 24.** *CONJ_NEG:*
$\forall z \in \mathbb{C}, conj\ (-z) = -conj\ z$

**Theorem 25.** *CONJ_SUB:*
$\forall z, w \in \mathbb{C}, conj\ (z - w) = conj\ z - conj\ w$

**Theorem 26.** *CONJ_MUL:*
$\forall z, w \in \mathbb{C}, conj\ (z * w) = conj\ z * conj\ w$

**Theorem 27.** *CONJ_INV:*
$\forall z \in \mathbb{C}, conj\ (inv\ z) = inv\ (conj\ z)$

**Theorem 28.** *CONJ_DIV:*
$\forall z, w \in \mathbb{C}, conj\ (z/w) = conj\ z/conj\ w$

**Theorem 29.** *CONJ_SCALAR_LMUL:*
$\forall k \in \mathbb{R}, z \in \mathbb{C}, conj\ (k * z) = k * conj\ z$

The theorems above represent that for all of algebra operating f, the conjugate value is commutative, i.e. $f(conj\ z) = conj\ f(z)$.

Other two complex conjugate properties are fairly important, which will be used frequently in some problems of proof.

**Theorem 30.** *CONJ_CONJ:*$\forall z \in \mathbb{C}, conj\ (conj\ z) = z$

**Theorem 31.** *COMPLEX_MUL_RCONJ:* $\forall z \in \mathbb{C}$,
$z * conj\ z = complex\_of\_real((RE\ z)^2 + (IM\ z)^2)$

Their implementations in HOL4 are as follows.

⊢ val CONJ_CONJ = store_thm(
"CONJ_CONJ", ″!z:complex. conj (conj z) = z″,
REWRITE_TAC[conj, RE,IM,REAL_NEGNEG]);

⊢ val COMPLEX_MUL_RCONJ = store_thm(

```
"COMPLEX_MUL_RCONJ",
"!z:complex. z*conj z =
complex_of_real ((RE z) pow 2+(IM z) pow 2)",
REWRITE_TAC [complex_mul, conj, RE, IM,
complex_of_real, REAL_MUL_RNEG,
REAL_SUB_RNEG] THEN
PROVE_TAC [POW_2,
REAL_MUL_COMM, REAL_ADD_LINV]);
```

## 6. Polar Form of Complex Numbers

### 6.1. Modulus and Argument

Periodically varying signals can be described with complex numbers conveniently in signal analysis and other fields. In this case, the polar forms of complex numbers have many advantages. For example, multiplication, division and power of complex numbers are simpler in the polar form. The polar form of a complex number is represented with a modulus and an argument.

The modulus (absolute value) of a complex number means the distance between the origin point and the point corresponding to the complex number in the complex plane. Obviously it is a non-negative real number.

**Definition 14.** $\forall z \in \mathbb{C}$,
$modu\ z = sqrt\ ((RE\ z)^2 + (IM\ z)^2)$

where $sqrt$ is the function of square root of real.

There are also some properties of the modulus dealing with complex numbers.

**Theorem 32.** *MODU_NEG:* $\forall z \in \mathbb{C}, modu\ (-z) = modu\ z$

**Theorem 33.** *MODU_CONJ:*
$\forall z \in \mathbb{C}, modu\ (conj\ z) = modu\ z$

**Theorem 34.** *MODU_MUL:*
$\forall z, w \in \mathbb{C}, modu\ (z * w) = modu\ z * modu\ w$

**Theorem 35.** *MODU_INV:*
$\forall z \in \mathbb{C}, z \neq 0 \Rightarrow modu\ (inv\ z) = inv\ (modu\ z)$

**Theorem 36.** *MODU_DIV:*
$\forall z, w \in \mathbb{C}, w \neq 0 \Rightarrow modu\ (z/w) = modu\ z/modu\ w$

**Theorem 37.** *MODU_SCALAR_LMUL:*
$\forall k \in \mathbb{R}, z, w \in \mathbb{C}, modu\ (k * z) = abs\ k * modu\ z$

where $abs\ k$ is the absolute value of real number $k$.

An interesting property is that the modulus of the sum of two arbitrary complex numbers is not bigger than the sum of their modulus, and if and only if the principal value of their argument are equal, the equality holds. This property is the famous triangle inequality.

**Theorem 38.** *MODU_TRIANGLE_INEQUALITY:*
$\forall z, w \in \mathbb{C}, modu\ (z + w) \leq modu\ z + modu\ w$

The argument or phase of $z$ is the angle to the real axis, and is written as $arg(z)$. The value of an argument keeps the same angle when adding multiples of $2\pi$ (note that radians are being used). Hence, the $arg$ function is sometimes considered as multivalued. Normally, as given above, the principal value is chosen in the interval $[0, 2\pi)$. So the inverse cosine function is used to define the principal value of the argument of complex number.

**Definition 15.** $\forall z \in \mathbb{C}$,

$$argz = \begin{cases} \arccos \frac{RE\ z}{modu\ z}, & 0 \leq IM\ z \\ -\arccos \frac{RE\ z}{modu\ z} + 2\pi, & IM\ z < 0 \end{cases}$$

Note: In the traditional definition of mathematics, the argument of complex number 0 is arbitrary; but according to this definition, $arg\ 0c$ is $\frac{\pi}{2}$ .

The follow properties reflect the translation between the polar form and the rectangular coordinate form. By calling the theorems about inverse trigonometric functions in transcTheory in HOL4, we can prove the follow properties.

**Theorem 39.** *RE_MODU_ARG:*
$\forall z \in \mathbb{C}, RE\ z = modu\ z * \cos(arg\ z)$

**Theorem 40.** *IM_MODU_ARG:*
$\forall z \in \mathbb{C}, IM\ z = modu\ z * \sin(arg\ z)$

**Theorem 41.** *COMPLEX_TRIANGLE:*
$\forall z \in \mathbb{C}, modu\ z * (\cos(arg\ z), \sin(arg\ z)) = z$

The following is the decision theorem that two complex numbers are equal in the polar form.

**Theorem 42.** *COMPLEX_MODU_ARG_EQ:* $\forall z, w \in \mathbb{C}$,
$(z = w) \Leftrightarrow ((modu\ z = modu\ w) \wedge (arg\ z = arg\ w))$

The distance from any points on the unit circle to the center of the circle is 1.

**Theorem 43.** *MODU_UNIT:*
$\forall x \in \mathbb{R}, modu\ (\cos x, \sin x) = 1$

The following two theorems are the multiplication and division rules of complex numbers in polar form. They are of important significance in geometry and are useful in dealing with the geometric issues about rotation.

**Theorem 44.** *COMPLEX_MUL_ARG:* $\forall x, y \in \mathbb{R}$,
$(\cos x, \sin x) * (\cos y, \sin y) = (\cos(x+y), \sin(x+y))$

**Theorem 45.** *COMPLEX_DIV_ARG:* $\forall x, y \in \mathbb{R}$,
$(\cos x, \sin x)/(\cos y, \sin y) = (\cos(x-y), \sin(x-y))$

Now, we prove DE MOIVRE Theorem, as an example, and demonstrate the application of complex numbers in polar form in HOL4. At first, we define the operation of nature numbers power of complex numbers, and overload the operator pow. The induction definition is used here:

**Definition 16.** $\forall n \in \mathbb{N}, z \in \mathbb{C}$,
$z\ pow\ 0 = 1c \wedge z\ pow\ (n+1) = z * (z\ pow\ n))$

Next, we prove the following two theorems using the method of mathematical induction:

**Theorem 46.** *COMPLEX_POW_SCALAR_LMUL:*
$\forall n \in \mathbb{N}, k \in \mathbb{R}, z \in \mathbb{C}, (k * z)\ pow\ n = (k\ pow\ n) * (z\ pow\ n)$

**Theorem 47.** *DE_MOIVRE_LEMMA:*$\forall n \in \mathbb{N}, k \in \mathbb{R}$,
$(\cos x, \sin x)\ pow\ n = (\cos(\&n * x), \sin(\&n * x))$

⊢ val COMPLEX_POW_SCALAR_LMUL = store_thm(
”COMPLEX_POW_L”,
″!n:num k:real z:complex.
(k * z) pow n = (k pow n) * (z pow n)″,
INDUCT_TAC THEN
REWRITE_TAC[complex_pow, pow,
COMPLEX_SCALAR_LMUL_ONE] THEN
REPEAT GEN_TAC THEN
ASM_REWRITE_TAC[] THEN
REWRITE_TAC[COMPLEX_MUL_SCALAR_LMUL2]);

⊢ val DE_MOIVRE_LEMMA = store_thm(
”DE_MOIVRE_LEMMA”,
″!x:real n:num.
(cos x, sin x) pow n = (cos (&n * x), sin(&n * x))″,
GEN_TAC THEN INDUCT_TAC THEN
ASM_REWRITE_TAC [complex_pow, COS_0,
REAL_MUL_LZERO, SIN_0,
complex_of_num, complex_of_real,
COMPLEX_MUL_ARG] THEN
ONCE_REWRITE_TAC [REAL_ADD_COMM] THEN
REWRITE_TAC[REAL, REAL_ADD_RDISTRIB,
REAL_MUL_LID]);

DE_MOIVRE_THM can be proved with the two theorems above:

**Theorem 48.** *DE_MOIVRE_THM:* $\forall n \in N, z \in C$,
$(modu\ z * (\cos(argz), \sin(argz)))\ pow\ n =$
$(modu\ z)\ pow\ n * (\cos(n * argz), \sin(n * argz))$

⊢ val DE_MOIVRE_THM = store_thm(
”DE_MOIVRE_THM”,
″!z:complex n:num.
(modu z * (cos (arg z),sin (arg z))) pow n
= modu z pow n * (cos (&n * arg z),sin(&n * arg z))″,
REWRITE_TAC[COMPLEX_POW_SCALAR_LMUL,
DE_MOIVRE_LEMMA]);

### *6.2. Exponential Form of Complex Numbers*

With the definitions of the modulus and the principal values of arguments of complex numbers, in practice, we often express complex numbers in exponential form. The exponential form is the basic of many mathematical methods in digital signal processing so that the multiplication, division and power operations of complex numbers become very simple. Overloading the exp operator symbols in HOL4 is to define the $e^z$.

**Definition 17.** $\forall z \in \mathbb{C}$,
$\exp(z) = \exp(RE\ z) * (\cos(IM\ z), \sin(IM\ z))$

Euler formula can be proved using previous definition of imaginary unit $i$.

**Theorem 49.** *EXP_IMAGINARY:*
$\forall x \in \mathbb{R}, \exp(i * x) = (\cos x, \sin x)$

Especially, for x=$\pi$, holds $\exp(i * \pi) = -1c$, thus the two transcendental numbers $\pi$ and $e$ are linked together.

The exponential form of complex numbers can be proved:

**Theorem 50.** *EULER_FORMULE:*
$\forall z \in \mathbb{C}, modu\ z * \exp(i * arg\ z) = z$

The following rules of the operations of the complex numbers in exponential form are also trivial to prove:

**Theorem 51.** *COMPLEX_EXP_ADD:*
$\forall z, w \in \mathbb{C}, \exp(z + w) = \exp(z) * \exp(w)$

**Theorem 52.** *COMPLEX_EXP_NEG:*
$\forall z \in \mathbb{C}, \exp(-z) = inv\ (\exp(z))$

**Theorem 53.** *COMPLEX_EXP_SUB:*
$\forall z, w \in \mathbb{C}, \exp(z - w) = \exp(z)/\exp(w)$

**Theorem 54.** *COMPLEX_EXP_N:*
$\forall n \in \mathbb{N}, z \in \mathbb{C}, \exp(n * z) = \exp(z) pow\ n$

## 7. A Case Study About Sine Wave Signals

In this section, an example is presented to show application of complex number in verification of signal systems. A sine wave signal, v(t), has the general form:$v(t) = A\sin(\omega t + \theta)$ , where A is amplitude of wave, $\omega$ is frequency, t is time and  is phase shift. The sum of two sine waves of the same frequency results in another sine wave of the same frequency but having a different amplitude and phase shift. There arent enough trigonometric functions to prove the property in HOL4, but it is easy to prove this in complex number polar forms and exponential forms. Given two sine wave signals

$$v_1(t) = A_1 \sin(\omega t + \theta_1)$$
$$v_2(t) = A_2 \sin(\omega t + \theta_2)$$

, the goal is to prove that the sum of the two signals has the form:

$$v_3(t) = v_1(t) + v_2(t) = A_3 \sin(\omega t + \theta_3)$$

Euler's Identity (Theorem EULER_FORMULE) describes a relationship between polar form complex numbers and sine wave signals ,

$$Ae^{i\theta} = A\cos(\theta) + iA\sin(\theta)$$

,thus

$$v_1(t) = IM\{A_1 e^{i(\omega t+\theta_1)}\}$$
$$v_2(t) = IM\{A_2 e^{i(\omega t+\theta_2)}\}$$
$$\begin{aligned} v_3(t) &= IM\{A_1 e^{i(\omega t+\theta_1)}\} + IM\{A_2 e^{i(\omega t+\theta_2)}\} \\ &= IM\{A_1 e^{i(\omega t+\theta_1)} + A_2 e^{i(\omega t+\theta_2)}\} \\ &= IM\{e^{i(\omega t)}(A_1 e^{i\theta_1} + A_2 e^{i\theta_2}\} \end{aligned}$$

The sum of two complex numbers results in another complex number. Let

$$z_3 = A_3 e^{i\theta_3} = A_1 e^{i\theta_1} + A_2 e^{i\theta_2}$$

then

$$\begin{aligned} v_3(t) &= IM\{e^{i\omega t} A_3 e^{i\theta_3}\} = IM\{A_3 e^{i(\omega t+\theta_3)}\} \\ &= A_3 \sin(\omega t + \theta_3) \end{aligned}$$

The goal is proved. The proof goal and process in HOL4 are as follows:

This case study shows that the HOL library of complex number theory, especially the polar forms, can simplify modeling and verification of some signal processing problems.

```
⊢val SIN_ADD_SIN=store_thm(”MODU_UNIT_EXP”,
″!t omega thet1 thet2 V1 V2 V3 A1:real A2:real.
(V1(t) = A1 * sin(omega * t + thet1)) ∧
(V2(t) = A2 * sin(omega * t + thet2)) ∧
(V3(t) = V1(t) + V2(t))
=¿?A3:real thet3. V3(t) = A3 * sin(omega * t + thet3)″,
REPEAT GEN_TAC THEN
REWRITE_TAC[prove(″!x. sin x = IM (exp (i * x))″,
REWRITE_TAC [EXP_IMAGINARY,IM])] THEN
DISCH_TAC THEN ASM_REWRITE_TAC[] THEN
REWRITE_TAC [prove(″!x:real z. x * IM z = IM (x * z)″,
REWRITE_TAC[COMPLEX_SCALAR_LMUL, IM]),
prove(″!z w. IM z + IM w = IM(z + w)″,
REWRITE_TAC[COMPLEX_ADD, IM]),
COMPLEX_SCALAR_RMUL_ADD,
COMPLEX_EXP_ADD] THEN
ONCE_REWRITE_TAC[COMPLEX_MUL_COMM] THEN
REWRITE_TAC[GSYM
COMPLEX_LMUL_SCALAR_LMUL] THEN
MAP_EVERY EXISTS_TAC [
″modu(A1:real * exp (i * thet1:real) +
A2:real * exp (i * thet2:real))″,
″arg((A1:real * exp (i * thet1:real) +
A2:real * exp (i * thet2:real)))″]
THEN AP_TERM_TAC THEN
REWRITE_TAC[GSYM COMPLEX_ADD_RDISTRIB]
THEN REWRITE_TAC[COMPLEX_MODU_ARG_EQ]
THEN CONJ_TAC THEN
REWRITE_TAC[EULER_FORMULE]);
```

## 8. Conclusion

In this paper, we have formalized complex numbers in logic and produced the theorem library of complex numbers in HOL4. The theorem library includes the data type of complex numbers, the basic operations of complex numbers and the proof of fundamental theorems of complex numbers. The polar form and exponential form are expressed more directly based on the definition of the scalar multiplication of complex numbers. It becomes more convenient to solve many problems. In addition, we also have implemented many other properties in HOL4 which were not mentioned in this paper. In the light of the power of the theory of complex number, the theorem library is expected to extend the scope of application of HOL4. The developed complex number theory has been released in HOL4 Kananaskis-7.

## Acknowledgement

First and foremost we thank Shengzhen Jin for his guidance and encouragement to this work. We also thank Michael Norrish for his reviewing our complex theory library and giving many good suggestions.

This work is supported by the International S&T Cooperation Program of China (2010DFB10930, 2011DFG13000); the National Natural Science Foundation of China(60873006, 61070049, 61170304, 61104035); the Beijing Natural Science Foundation and S&R Key Program of BMEC(4122017, KZ201210028036). Open Projects of State Key Laboratory of Computer architecture and Guangxi Key Laboratory trusted software.

## References

[1] K. Slind and M. Norrish, TPHOLs ’08 Proceedings of the 21st International Conference on Theorem Proving in Higher Order Logics Springer-Verlag Berlin, Heidelberg 28 (2008).
[2] J. Harrison, TPHOLs 2001. LNCS, Springer, Heidelberg,**2152**, 159 (2001).
[3] J. Harrison, Formal Methods in System Design, **5**, 35 (1994).
[4] H. Geuvers and M. Niqui, TYPES ’00 Selected papers from the International Workshop on Types for Proofs and Programs. Springer-Verlag London, UK. 79 (2002).
[5] H. Geuvers, R. Pollack, F. Wiedijk and J. Zwanenburg, Journal of Symbolic Computation - Integrated reasoning and algebra systems. **34**, 271 (2002).
[6] C. Bylinski, Formalized Mathematics, **1**, 507 (1990).
[7] A. J. Milewska, Formalized Mathematics, **9**, 265 (2001); A. Y. Mahmoud and Alexander G. Chefranov, Inf. Sci. Lett. **1**, 91 (2012).
[8] R. Milewski, Formalized Mathematics, **9**, 455 (2001).
[9] R. Authan, Mathematical Case Studies: the Complex Numbers, Technology report, December (2006).
[10] R. W. Butler, http://shemesh.larc.nasa.gov/fm/ftp/larc/PVS-library/complex-details.htht
[11] Tobias Nipkow, Lawrence C. Paulson, Markus Wenzel. Isabelle/HOL A Proof Assistant for Higher-Order Logic. Lecture Notes in Computer Science, Springer Verlag **2283**.

**Dr. Zhiping Shi** received the Ph.D. degree in computer software and theory from Institute of Computing Technology, Chinese Academy of Sciences, China, in 2005. Currently he is an associate professor at the Capital Normal University, Beijing, China. His research interests include formal verification and visual information analysis. He has been a research staff at Institute of Computing Technology, Chinese Academy of Sciences from 2005 to 2010.

**Liming Li** is a postgraduate at Capital Normal University, China. His research focus is theorem proving in formal verification.

**Prof. Yong Guan** received the Ph.D. degree in computer science from China University of Mining and Technology, in 2004. Currently, he is a professor of Capital Normal University, China. His research interests include formal verification, PHM for power and embedded system design. Dr. Guan is a member of Chinese Institute of Electronics Embedded Expert Committee, and a member of Beijing Institute of Electronics Professional Education Committee, and Standing Council Member of Beijing Society for Information Technology in Agriculture.

**Prof. Xiaoyu Song** received the Ph.D. degree from the University of Pisa, Italy, 1991. From 1992 to 1999, he was on the faculty at the University of Montreal, Canada. In 1998, he worked as a Senior Technical Staff in Cadence, San Jose. In 1999, he joined the faculty at Portland State University. He is currently a Professor in the Department of Electrical & Computer Engineering at Portland State University, Portland, Oregon. His current research interests include formal methods, design automation, embedded system design, and emerging technologies. He has been awarded as the Intel Faculty Fellow during 2000C2005. He served as an associate editor of IEEE Transactions on Circuits and Systems and IEEE Transactions on VLSI Systems.

**Prof.Minhua Wu** received the Master degree in computer architecture from China University of Nanjing,in 1989. Currently, she is a professor of Capital Normal University, China. Her research interests include formal verification and embedded system design.

**Jie Zhang** received the Master degree in computer science from China University of Mining and Technology, in 1999. Currently, she is an associate professor of Beijing University of Chemical Technology, China. Her research interests include formal verification, PHM for power and embedded system design. She is a member of Chinese Institute of Electronics Embedded Expert Committee.