数理解析研究所講究録
986巻 1997年 73-76

Ring Story

Hidetsune Kobayashi, Dongdai Lin, Yoshihiko Sakai and Hideo Suzuki
hikoba@math.cst.nihon-u.ac.jp
Dept. of Math., Coll. of Sci. & Tech. Nihon Univ.,
1-8 Surugadai, Tokyo 101, Japan

1. Introduction

Now, we have some formula manipulating languages. Each of them gives answer when we type in a command with some source data. For example, some polynomials and a command "Groebner" give a set of polynomials called Gröbner basis. This example shows that we can treat some properties of a polynomial ring by formula manipulating language. But some fundamental properties cannot be treated in formula manipulating language. Among such properties, the most fundamental one will be Noetherian property. That is, given an ascending chain of infinite number of ideals, we have a number n such that all ideals appearing after n-th ideal coincide. This proposition is almost impossible to express in formula manipulating language, since giving a rule to specify an ascending chain is not always constructive, and we have no algorithm to determine the number from which the chain stops.

In numerical calculation, a solution is given approximately, but in abstract theory, some properties are obtained by a finite number of repeated calculations under Noetherian condition. One of such property is "spectral sequence" of a differential modules.

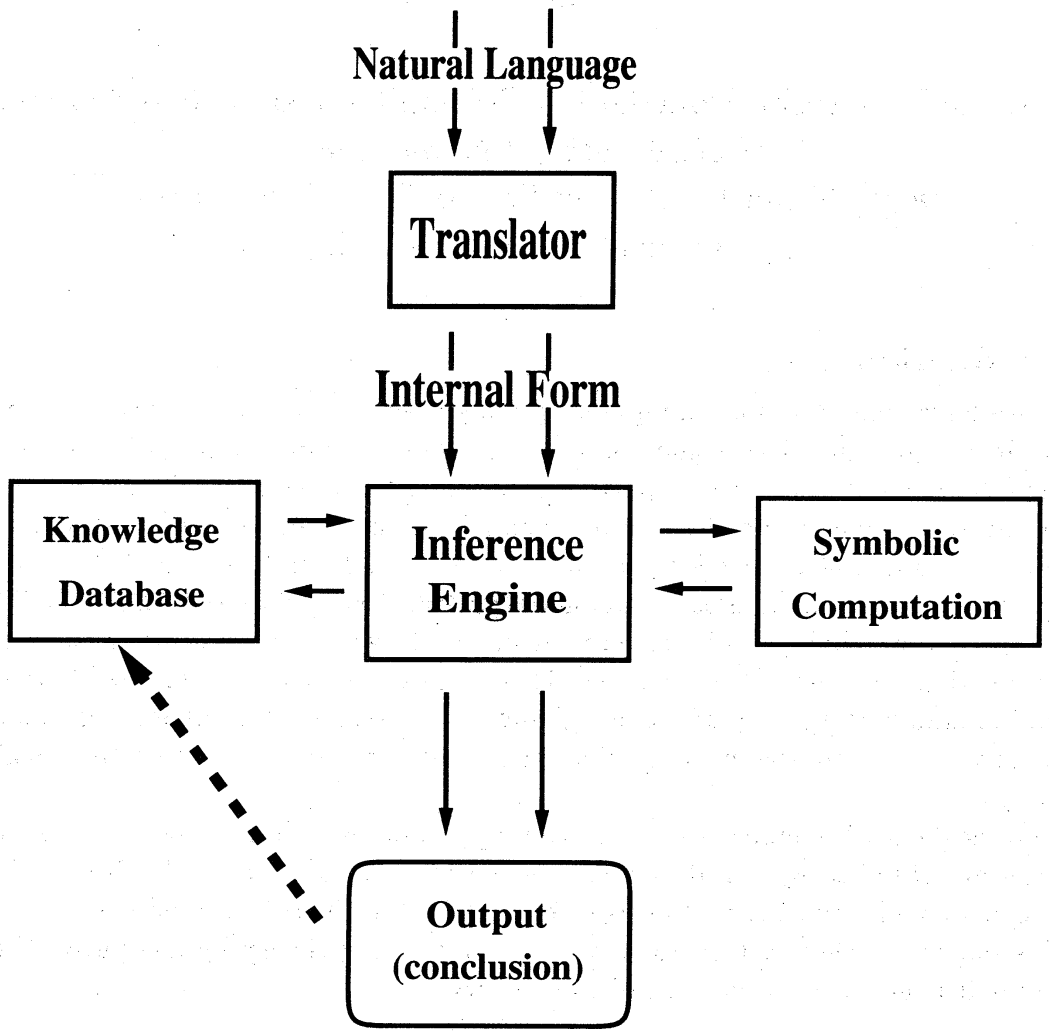
In this paper, we show a trial to treat an abstract ring theory by computer. Our title shows that it is not a research work of ring theory.

2. General Scheme

In general, a textbook of ring theory has no complicated sentences comparing with the sentences appearing in the novels. However, there are still some variations to express the same concept, and this would cause the difficulties for computer to understand and to process. Hence, we avoid to treat natural language in general, but treat the so-called standard expressions as an interface. Here we give some examples:

```
set X
set {a,b,c}
set {x;integer}
set {x;integer & even}
element x in set X
element x in set X & in set Y
element x in set X | in set Y
```

```
definition 1.1.3 X is subset of Y
statement
  if forany element x in X, x is element in Y
```



Given items expressed as above, we translate them into the so-called "internal form" by a translator. The structure of the internal form will be given in the next section.

Internal forms are thrown into the inference engine. With some related knowledge extracted from the knowledge database, the inference engine gives a conclusion to the problem. Then the conclusion expressed in the internal form will be translated back to the natural language. And at the same time, if the conclusion is something new, we can choose to store into the knowledge database.

3. The structure of Internal Form

```
Word: (NUMBER a) ;;; a is a number
      (INTEGER a) ;;; a is an integer
      (EVEN a) ;;; a is an even integer
```

```
      (ODD a) ;;; a is an odd integer
      (SET S) ;;; S is a set
      (SET S EMPTY) ;;; S is empty set
      (SET S INFINITE) ;;; S is infinite set
      (SET S FINITE) ;;; S is finite set
      (SUBSET S1 S2) ;;; S1 is subset of S2
      (ELEMENT e S) ;;; e is an element of set S
      (RING R b1 b2) ;;; set R is ring with addition b1 and multiplication b2
      (SUBRING R1 R2) ;;; R1 is subring of R2
      (IDEAL I R) ;;; I is an ideal of R
      (MODULE M R Operator)
      (SUBMODULE M1 M2) ;;; M1 is submodule of M2
      ...

Operator:  (BINARY b S) ;;; b is a binary operator of set S
           (MAP m S1 S2) ;;; m is map from set S1 to set S2
           (RINGHOMOMORPHISM f R1 R2)
           (RINGISOMORPHISM f R1 R2)
           (MODULEHOMOMORPHISM f M1 M2)
           (MODULEISOMORPHISM f M1 M2)
           ...

LogicWord: Word
           Logic

Operation: (OPERATION Operator LogicWord LogicWord)
                ;;; expression is meaningful or not and result depend on the
                ;;; the definition of Operator
           (DIFFERENCE S1 S2) ;;; the set S2\S1
           (UNION S1 S2) ;;; union of set S1 and S2
           (INTERSECTION S1 S2) ;;; intersection of set S1 and S2
           (IMAGE Map Word) ;;; Image of Word under Map
           (QUOTIENT Word Word) ;;; Quotient object Word(1)/Word(2)
           (CARTESIAN S1 S2) ;;; cartesian product of S1 and S2
           (COMPOSITION Operator Operator)
               ;;; Composition of Operator(1) and Operator(2)
           ...

Logic: (OR LogicWord LogicWord)
           ;;; true if LogicWord(1) is true or LogicWord(2) is true
       (AND LogicWord LogicWord)
           ;;; true if both LogicWord(1) and LogicWord(2) are true
       (NOT LogicWord)
           ;;; true if LogicWord is not true
       (EQUAL LogicWord LogicWord)
```

```
          ;;; true if  LogicWord(1) is equal to LogicWord(2)
       ...

Phrase:  Word
         Operator
         Operation
         Logic
         Definition
         Theorem

PhraseList: Phrase
            PhraseList Phrase

Definition: (DEFINITION PhraseList PhraseList PhraseList)
   Theorem: (THEOREM PhraseList PhraseList)
               ;;; Definition ---> Under PhraseList(1), PhraseList(2) if
                                    and only if PhraseList(3).
               ;;; Theorem ----> If PhraseList(1) then PhraseList(2).
```

4. Inference Engine

Given a problem or a proposition to be checked, we express them in internal form. Then we need a function which gives a conclusion. We call this part as "Inference Engine".

For the details about the inference engine, we will discuss in our later papers.