View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Kyoto University Research Information Repository

KURENAI 紅
Kyoto University Research Information Repository

KYOTO UNIVERSITY

Title	Model theory and programing language(Computer Algebra and its Applications to Mathematical Studies)
Author(s)	ISHIKAWA, Shiro; NAGATA, Morio
Citation	数理解析研究所講究録 (1984), 520: 166-169
Issue Date	1984-04
URL	http://hdl.handle.net/2433/98440
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

Kyoto University

数理解析研究所講究録
520巻 1984年 166-169

Model theory and programing language

石川 史郎　　　　永田 守男
Shiro ISHIKAWA and Morio NAGATA

Fuculty of Science and Technology
Keio University
3-14-1 KOHOKUKU YOKOHAMA JAPAN 223

1, Introduction

Our purposes are to construct a programing language which is fit a model theory in mathematics and to show that its effectiveness. For this, we suppose a little knowledge on a model theory in order to read this paper (see, for example, [1]). Since we are concerned with classical analysis, it is sufficent to consider it in supper-structure U constructed by a real field R. Defining ordinarily a surjection i from a set $\mathcal{S}$ made up by finite sequences of alphabet $\mathcal{A}$ to the supper-structure U, we construct a formal language by a usual way. In this paper we try to regard the formal language as the programing language.

For this, we should show the table of the surjection $i:\mathcal{S}\to U$, which would be called the dictionary of U, in the section 2. Next we consider a little about "normal represetation of formular" in yhe section 3 and about " question and command", by which computer works effectively, in the section 4. In the section 5, we show an example of our program.

2. Dictionary of the supper-structure U

About the following words (i.e. elements of $\mathcal{S}$), we shall explain:

a) R

:: R denotes a real field.

b) $\pm 0.a_1a_2a_3a_4a_5a_6 \cdot 10{**}n$ (where n and $a_k (1 \leq k \leq 6)$ are integers such that $0 \leq a_k \leq 9$, $-100 \leq n \leq 100$)

:: which denotes a floating point of order 6.

c) $\pm a_n a_{n-1} \cdots a_1 a_0 . b_0 b_1 \ldots b_m$ (where a_i and b_j $(0 \leq i,j \leq 5)$ are integers such that $0 \leq a_i,\ b_j \leq 9$)

:: which denotes a fixed point.

d) +,-,*,/, **, sin , cos , tan , exp , log , ln , abs ,

:: which denotes an ordinary arithematic operator respectively.

e) $=, \leq, \geq,$

:: which denote an equality and inequalities

REMARK If R^n (n-dimensional Euclidean space), Mat(m,n), etc are contained in this dictionart, it seems that our language is more powerful. But in thi paper we do not mention about this extention.

3. Normal representation

For example, we consider the following formulars

(1) $x=1$

(2) $x^2=1 \wedge x \geq 0$ (where $\wedge$ means "and").

These two formulars are equivalent, but it seems that the formular (1) is simpler than (2). This implies that a kind of formulars often have a simpler representation. In this section, we consider about this when x is a real number.

The formular

$$x =' 0.a_1a_2a_3a_4a_5a_6 . 10^n$$

means that the approximation of x is $0.a_1a_2a_3a_4a_5a_6 . 10^n$.

And this formular is called an approximate normal representation of x and denoted by ANR[x].

4. Question and Command

We can input the following question to computer

"WHAT IS ANR[x] ? " (in abbrieviation "ANR[x]?")

This means the question what is an approximate normal representation of x. And the question is effective when there have been already existed a formular $x=f(x_1,x_2,...,x_n)$ where f is a composed function of arithemetic orerators in the section 2 ,d) and approximate normal representations of $x_1,x_2,...,x_n$ have been already known. Although we should define a various commands, in this paper we borrow them in FORTRAN for convienience. And the representation's methods of formulars are refered in [2].

5. Example

Under these representations, we write the program in our language obtaining an approximation 2**(1/2) by the newton's method.

```
      DIMENSION X(20)
      X(1): X(1)=2.0
      WHAT IS ANR[X(1)]?
      DO 100 n=2,20
      X(n): X(n)=(X(n-1)**2+2.0)/2*X(n-1)
      WHAT IS ANR[X(n)] ?
      IF X(n-1)-X(n) .LT. 10.0**(-5) GOTO 200
  100 CONTINUE
```

```
200 PRINT X(n)
    STOP
```

REFERENCES

[1] M. Davis. Appleid nonstandard analysis, John Wiley & Sons 1977.

[2] S.Ishikawa and M.Nagata. Note on the method for describing definitions theorems and proofs in mathematics, SURIKAISEKIKEN-KOKYUROKU 1983 "Formular Manipulation and its Application to Mathematics"