ONE VERSION OF PROGRAMMING ACCORDING TO LOGIC τ_{SR} LANGUAGE

Kh. Rukhaia, L. Tibua

I. Vekua Institute of Applied Mathematics
I. Javakhishvili Tbilisi State University

In the last thirty years of the twentieth century a new field of logical program has been produced and developed in programming [1]. We can say that at the end of the century logical programming was a success. Inspite of this success there are some difficulties which are caused by the fact that up to now there is not constructed programming language based on the ideology of Sh. Pkhakadze [2]. One of the basis of such a program language can be logic τ_{sr}. Language symbols of this logic are:

I. Basic symbols.

1) Logical symbols: $\neg$, $\bigvee$, $\bigwedge$, $\rightarrow$, $\leftrightarrow$,=.

2) Logical operative mark τ.

3) Subject letters

4) Predicate letters

5) Functional letters

6) Special operative mark of substitution S.

7) logical-special operative mark of substitution R.

II. Marks, which are introduced by I end II' definitions.

Definitions are numbered in that way $D_m[n]$, where m is a natural number and n shows type of the definition. On the right side of the definition there are only the fundamental operators or operators which have been defined earlier.

For example.

First type of definition

$\exists xA - R_x\tau_xAA$

Second type of definition

$\langle\pi_p\rangle_xUT - \exists y = US_{xy}T$

Where y isn't variable in U and T.

Besides, let us remark that the axiomatic of τ_{SR} logic provides invariance of all its operators [3].

We shall show some of our ideas by the concrete example of logic program.

Discuss the program [1].

parent(niko, mary)

parent(lali, mary)

parent (lali, kety)

mother (lali, mary)

female (mary)

$$child(x, y) : -parent(y, x) \qquad (1)$$

$$sisterof(x, y) : -parent(z, x) \wedge parent(z, y) \wedge parent(x, y) \wedge female(x) \wedge diff(x, y) \qquad (2)$$

$$mother(x,y) : -parent(x,y) \wedge female(x) \tag{3}$$

?-female (lali)

Our question consists of one goal and we must reach this goal.

While analyzing the program it was found out that there is no such a clause whose head would be match our goal. That's why answer to this question will be "no".

If in the program we add the following rules

$$female(x) : -mother(x,-) \tag{4}$$

and ask the same question

?-female (lali)

When during the analysis the program we will found out a rule whose head matches the goal. This rule is the following:

female (x):-mother (x,-)

So our goal will be changed with a new goal which is the following

mother (lali,-)

This goal is successful because there exists a fact in program. So the answer to the question will be "yes".

When comparing the third and the fourth rules we see that predicate **mother** is defined with the help of predicate **female** and on the contrary, which is the defined defect of this program.

Now let's discuss the following alternative program of the given program.

We must make difference between the basic predicates and predicates given by definition in the alternative program.

The definition has the following form

$A \leftrightarrow B_1 \wedge \cdots \wedge B_n$

Here we mean that in $B_1, B_2, \cdots B_n$ meet only basic or predicates defined earlier and A includes predicate defined by this definition.

We shall use such definitions as our alternative program rules.

The alternative program is

Basic operators	Operators wich are introduced by definitions.
parent,female	sister,mother
Basic facts	Derived operators
parent(niko,mary) parent(niko,mary) female(mary)	mother

$D_1[I]child(x,y) \leftrightarrow parent(y,x)$

$D_2[II]sister(x,y) \leftrightarrow \exists z(mother(z,x) \wedge mother(x,y) \wedge female(x) \wedge deff(x,y))$

$D_3[I]mother(x,y) \leftrightarrow mother(x,y) \wedge female(x)$

?-female(lali)

Since the question starts with predicate **female**, first we must see if it is a basic or a derived predicate. As we see it is in the list of the basic predicates that's why we have to find out it in the basic facts.

First we try to match our goal to the basic facts. It is impossible. The difference is from the given predicate that we don't try to match the rules in the head because we are sure that this match will not be realized. We are starting to search first predicate whose characteristic includes predicate **female** and try to match it to the proper part in suitable definition. After that our goal is the left part of this definition. While working at this goal we shall use only derived facts. We stop working in the case of success.But when we are not successful we look for the next derived operator which includes operator **female** and we shall repeat the same procedure before we exhaust the proper abilities.

In our case we shall receive answer in the following way. The first derived operator where we meet **female** is sister. Then we move to the difinition $D_2[II]$. Here x becomes instantised to lali. There is no fact **sister** in derived facts that's why we move to the next derived operator which includes **female**. It is mother. We shall disuss the definition $D_3[I]$.Then x becomes instantised to lali and as we meet mother lali in derived facts that's why the answer of our program will be "yes".

References

[1] Bratko I., Prolog Programing for Artificial Inteligence, 1986.

[2] Pkhakadze Sh., Some Problems of the Notation Theory. 1977, Tbilisi.

[3] Rukhaia Kh., Description of the Formal Mathematical τ_{SR} Theory with Substitucion Operators. 1995, Tbilisi.