Estratto da

R. Ferro e A. Zanardo (a cura di), *Atti degli incontri di logica matematica* Volume 3, Siena 8-11 gennaio 1985, Padova 24-27 ottobre 1985, Siena 2-5 aprile 1986.

Disponibile in rete su http://www.ailalogica.it

A LOGIC OF RECURSION

V. MICHELE ABRUSCI[1], GIANFRANCO MASCARI[2]
[1]Dipartimento di Filosofia Università di Firenze
[2]I.A.C., C.N.R., Roma

Introduction

The theory of abstract recursion introduced by Moschovakis is of interest in two fields: the models of this theory include most of the generalized recursion theories studied in logic and the basic computational models of theoretical computer science (including those of the fixed point theory of programs, database theory and complexity theory). Reciprocal influence of the research in these two fields are fruitful. More specifically, recursive program schemes can be treated as particular functionals.

In this paper, structures are introduced with full generality: particular structures, the so called "standard", are those in which λ-abstraction, recursion and iteration have the intended meaning. The logic defined allows to deal with partial objects and transfinite recursion (in particular, non terminating programs) and is based on many-sorted languages with a specific sort for ordinals.

Various assertions and specification languages have been proposed which are extensions of equational ones, or extensions of first order languages. Our approach seems to provide an unified framework for such proposals. The formal systems considered could be used for "proving programs" or "programming proofs", as suggested by Kreisel.

The kind of completeness theorems that can be obtained for such logics of programs (or of specifications) is one of the basic issues which characterize the various approaches in the fields of logic of programs and the theory of specification. Accordingly the distinction between the "logical" (provable by first order means) and "mathematical" (second order properties), two basic approaches can be considered: one approach deals with finitary rules and (essentially) non standard models, another approach deals with infinitary rules and standard models. This paper can be seen as a further development of this second approach, by benefiting of the results obtained by Girard in proof theory.

Summary of the paper

1) A β-similarity type Σ consists of sorts (the sort On of ordinals, a set S_o of "basic sorts" among them the sort bool, a set S_1 of "composed sorts" i.e. the set of all the configurations $(s_o,\dots,s_n \rightarrow s)$ of elements of S_o), of constants and function symbols (among them 0, 1, Booleq, if...then...else, Ap) with their arities ($\doteq$ configurations $(s_o,\dots,s_n \rightarrow s)$ where $s \in S_o$ and $s_i \in S_o \cup S_1$), and of predicate symbols (among them $\overset{s}{=}$ for each sort s, $\leq^{On}$, and others).

2) For each β-similarity type Σ we define a language $L(\Sigma)$. The alphabet has variables for each sort in Σ, all the constants, function symbols and predicate symbols in Σ, the usual connectives and quantifiers, and the operators λ, R, I. The terms are constructed as usually by means of variables, constants and function symbols; moreover, if t is a term of basic sort s, and $v_o,\dots,v_n$ are variables of basic sorts $s_o,\dots,s_n$, and v is a variable of sort $(s_o,\dots,s_n \rightarrow s)$, and α is a variable of sort On, then $\lambda v_o\dots v_n .\, t$, $Rvv_o\dots v_n .t$ and

$I^{\alpha} vv_o\dots v_n .t$ are terms of sort $(s_o,\dots,s_n \rightarrow s)$ (where $v_o,\dots,v_n$ are bounded variables, and v is a bounded variable in the last two terms, and α is free). In the construction of the formulas, the quantification is allowed only on variables of basic sorts. Thus, $L(\Sigma)$ is obtained from an usual elementary (first-order) many-sorted language, with the addition of the operators λ, R, I .

3) For each β-similarity type Σ we define the structures for $L(\Sigma)$. Roughly, a structure $\mathfrak{M}$ for $L(\Sigma)$ is a structure for the underlying usual many-sorted language of $L(\Sigma)$ together with a partial function $EVAL_{\mathfrak{M}}(\underline{t},\underline{e})$ (evaluation of a term $\underline{t}$ under a valuation $\underline{e}$ of the variables) and a relation $\mathfrak{M},\underline{e} \overset{*}{\models} \underline{A}$ (the formula $\underline{A}$ is satisfied under the valuation $\underline{e}$ of the variables), s.t.:

- the function symbols are interpreted as partial functions, so that if t is a term and e is a valuation $EVAL_{\mathfrak{M}}(t,e)$ may be undefined;
- $EVAL_{\mathfrak{M}}(t,e)$ is defined as usually, if t does not contain λ, R, I;
- $\mathfrak{M},e \overset{*}{\models} t_1 \overset{s}{=} t_2$ iff $EVAL_{\mathfrak{M}}(t_1,e)$ and $EVAL_{\mathfrak{M}}(t_2,e)$ are defined and are in the relation $\overset{s}{=}_{\mathfrak{M}}$;
- $\mathfrak{M},e \overset{*}{\models} A$ is defined as usually, when A is a molecular formula.

Remark that $\mathfrak{M}(s_o,\dots,s_n \rightarrow s)$ needs not to be the set $\mathfrak{M}^+(s_o,\dots,s_n \rightarrow s)$ of all the partial functions from $\mathfrak{M}(s_o) \times\dots\times \mathfrak{M}(s_n)$ to $\mathfrak{M}(s)$; but, for each $x \in \mathfrak{M}(s_o,\dots,s_n \rightarrow s)$ (a "program") we may consider the partial function EXT(x) from $\mathfrak{M}(s_o)\times\dots\times\mathfrak{M}(s_n)$ to $\mathfrak{M}(s)$:

$$x_o,\dots,x_n \rightsquigarrow \mathfrak{M}(Ap)(x,x_o,\dots,x_n).$$

Thus, in general, to each function symbol of $L(\Sigma)$ we may associate a partial functional on $\mathfrak{M}$. We say that two objects of composed sort x,y are s.t. $x \leq_{\mathfrak{M}} y$, iff $EXT(x) \leq_{\mathfrak{M}} EXT(y)$; thus we may say that a function symbol is interpreted in $\mathfrak{M}$ as a "monotone" partial function; $EXT(x) \equiv_{\mathfrak{M}} EXT(y)$ means that the values of EXT(x) and

EXT(y) are in the relation $\overset{s}{=}_{\mathfrak{M}}$ when the functions are applied to the same arguments.

4) A structure $\mathfrak{M}$ for L(Σ) is standard iff $\mathfrak{M}$ is a β-model, all the relations $\overset{s}{=}_{\mathfrak{M}}$ are "congruence relations", all the function symbols are interpreted as "monotone" partial functions, and the interpretations of 0,1,Booleq, Ap, if...then...else, λ ,R and I are the standard ones. This means, in particular:

- $EVAL_{\mathfrak{M}}(\lambda v_0 \ldots v_n . t, e)$ is an object, s.t.
  $\mathfrak{M}(Ap)(EVAL_{\mathfrak{M}}(\lambda v_0 \ldots v_n . t, e), x_0, \ldots, x_n) \overset{s}{\simeq}_{\mathfrak{M}} EVAL_{\mathfrak{M}}(t, e \begin{bmatrix} v_0 \ldots v_n \\ x_0 \ldots x_n \end{bmatrix})$;
- if t is a term of basic sort s, and $v_0, \ldots, v_n$ are variables of basic sorts $s_0, \ldots, s_n$, and v is a variable of sort $(s_0, \ldots, s_n \to s)$, then the partial function
  $x, x_0, \ldots, x_n \leadsto EVAL_{\mathfrak{M}}(t, e \begin{bmatrix} v & v_0 & \ldots & v_n \\ x & x_0 & \ldots & x_n \end{bmatrix})$
  is a monotone function (in the variable x) from $\mathfrak{M}(s_0, \ldots, s_n \to s) \times \mathfrak{M}(s_0) \times \ldots \times \mathfrak{M}(s_n)$ to $\mathfrak{M}(s)$ and induces a monotone partial functional Φ from $\mathfrak{M}^+(s_0, \ldots, s_n \to s) \times \mathfrak{M}(s_0) \times \ldots \times \mathfrak{M}(s_n)$ to $\mathfrak{M}(s)$; then:
  $EXT(EVAL_{\mathfrak{M}}(I^{\alpha} v v_0 \ldots v_n . t, e)) \equiv_{\mathfrak{M}} \Phi^{e(\alpha)}$
  (where Φ^{μ} is the μ-th iteration of Φ)
  $EXT(EVAL_{\mathfrak{M}}(R v v_0 \ldots v_n . t, e)) \equiv_{\mathfrak{M}} \Phi^{\infty}$
  (where ∞ is the closure ordinal of the iteration).

(The condition " $\mathfrak{M}$ is a β-model" assures that the iteration is standard).

5) We give a set $Ax_{REC(\Sigma)}$ of axioms in the language L(Σ); and we prove that if $\mathfrak{M}$ is a structure for L(Σ) and $\mathfrak{M}$ is a β-model then

$\mathfrak{M} \overset{*}{\models} Ax_{REC(\Sigma)}$ iff $\mathfrak{M}$ is standard.

Therefore, if A is a formula of L(Σ) and M is a set of formulas of L(Σ), then are equivalent:

- "A is true in every standard structure $\mathfrak{M}$ for L(Σ) which is model of M" , $M \overset{*S}{\models} A$
- "A is true in every $\mathfrak{M}$ β-model of $Ax_{REC(\Sigma)}$ and M, $\mathfrak{M}$ structure for L(Σ) ", $M, Ax_{REC(\Sigma)} \overset{*\beta}{\models} A$.

6) We define - following Girard - the concept of " β-proof " in the langnage L(Σ). Remark that "recursive β -proofs" in the language L(Σ) are syntactical objects.

We prove the following completeness theorem:

For every(recursive) set of formulas M of L(Σ), for every formula A of L(Σ):
$M, Ax_{REC(\Sigma)} \overset{*\beta}{\models} A$ iff
there is a (recursive) β - proof of A from $M \cup Ax_{REC(\Sigma)}$ in L(Σ) .

(Remark that the Girard's β-completeness theorem cannot applied directly, because $\overset{*\beta}{\models}$ is not the usual semantic relation).

Therefore, if t_1 and t_2 are terms of sort s of L(Σ), then:

$\overset{*S}{\models} t_1 \overset{s}{=} t_2$ iff there is a recursive β-proof of $t_1 \overset{s}{=} t_2$ from $Ax_{REC(\Sigma)}$.

7) Moschovakis introduced the concept of "recursion structures of signature σ ", and the language REC(σ) (in fact, a programming language) together with a partial function Val($\mathfrak{A}, \alpha$,t) ($\mathfrak{A}$ recursion structure, α valuation of variables of REC(σ) , t term of REC(σ)), in order to define for each recursion structure $\mathfrak{A}$ the class of the functionals "recursive on $\mathfrak{A}$ ". Moschovakis raised the question:

"If $\mathcal{K}$ is a class of recursion structures of signature σ and t is a term of sort bool of REC(σ), we put
$\mathcal{K} \models t$ iff for all $\mathfrak{A}$ in $\mathcal{K}$ and all valuations α in $\mathfrak{A}$,
Val($\mathfrak{A}, \alpha$,t) $\simeq$ 1.

We would like to find natural axioms and rules of inferenee which will prove $\mathcal{K} \models t$ when this holds, at least for special cases of $\mathcal{K}$ and t ."

We investigate the relationships between "signatures" and " β-similarity types", languages REC(σ) and languages L(Σ),"recursion structures" and "standard structures for L(Σ)"; so that our completeness theorem gives a partial positive answer to the question raised by Moschovakis.

REFERENCES

ABRUSCI Vito Michele, MASCARI Gianfranco, A logic of recursion (in preparation)

ERSHOV A.P., Abstract Computability on Algebraic Structures, L.N.C.S., 122

GIRARD J.-Y., Proof theory and logical complexity, Bibliopolis, Napoli, (to appear)

KREISEL G., Proof theory and the synthesis of programs: potential and limitations, L.N.C.S., 203

MASCARI G., VENTURINI ZILLI M., While-Programs with Non-deterministic Assignments and the logic ALNA, Theoretical Computer Science, 40 (1985),211-235

MOSCHOVAKIS Y.N., Abstract recursion as a foundation for the theory of algorithms, in Computation and Proof Theory, L.N.M. 1104, Springer, 1984, pp.289-362.

TRAKHTENBROT B.A., Recursive Program Schemes and Computable Functionals, L.N.C.S. 45