ON ALGORITHMIC LOGIC WITH PARTIAL OPERATIONS

Uwe Petermann
Karl-Marx-Universität Leipzig, Sektion Mathematik
DDR 7010 Leipzig, Karl-Marx-Platz 1

Introduction
In theoretical considerations on algorithmic logic there is assumed as usual, that models of algorithmic theories are realizations assigning a full operation to every operator. In applications (for instance describing data structures by algorithmic theories) it is useful to allow realizations with partial operations too.

In the present paper we prove, that the deductive system of algorithmic logic (see /BCP/, /Mir/, /Ban/) with some slight modifications remains available in a larger class of realizations. There are considered many sorted algorithmic languages with partial operations, where the domain of each operation has a certain syntactical characterization. Namely to every operator \( \varphi \) corresponds a predicate with the same arity as \( \varphi \). There are considered only such realizations, in which the realization of the predicate \( \varphi^* \) corresponding to the operator \( \varphi \) coincides with the characteristic function of the domain of the operation \( \varphi^*_R \). This assumption enables us to prove a completeness theorem.

Furthermore there is given a syntactical characterization of the programme property "there will be no failure during the execution of the programme". Because of lack of space the results are given without proof.
The terminology corresponds to that used in /Ra-Si/ and /BCP/. The examples are suggested by /Sal1/.

1. Many sorted algorithmic languages with partial operations
1.1. Let \( \text{Srt} \) be an at most countable set (not empty), called in the following the set of sorts. By \( \text{Srt}^* \) we shall denote the set of all words over \( \text{Srt} \), 0 the empty word and by \( \text{Srt}^+ \) the set of all not empty words.

As the alphabet of an \( \text{Srt} \)-sorted algorithmic language with partial operations (shortly: alphabet) we understand the ordered pair \( A=(\Sigma,D) \) where \( \Sigma \) is the disjoint union of the following sets:
V_6$, for each sort $\mathfrak{e}$, where $\text{card}(V_6) = \aleph_0$;

the set of individual variables of the sort $\mathfrak{e}$,

$V_0$, where $\text{card}(V_0) = \aleph_0$; the set of propositional variables,

$\Phi_{\mathfrak{e},\mathfrak{e}}$ for each ordered pair $(\mathfrak{e}, \mathfrak{e}) \in \mathfrak{Srt} \times \mathfrak{Srt}$, where $\text{card}(\Phi_{\mathfrak{e},\mathfrak{e}}) \leq \aleph_0$;

the set of operators of the sort $\mathfrak{e}$ with the arity $\mathfrak{e}$,

$P_{\mathfrak{e}}$ for every $\mathfrak{e} \in \mathfrak{Srt}^*$, where $\text{card}(P_{\mathfrak{e}}) \leq \aleph_0$;

the set of predicates of the sort $\mathfrak{e}$,

$L_0$, a two-element set containing propositional constants $0$ and $1$,

$L_1$, a one-element set containing the unary logical connective $\neg$ (negation),

$L_2$, a three-element set containing the binary connectives $\land$ (conjunction), $\lor$ (disjunction), $\Rightarrow$ (implication),

$Q$, a four-element set containing the iterational quantifiers $\bigcap$, $\bigcup$ (general and existential) and the classical quantifiers $\forall$, $\exists$,

$\pi$, a three-element set containing the programme connectives sequential composition, if-then-else, while-do, denoted by $\cdot$, $\cdot$, $\cdot$ respectively,

$U$, a set of auxiliary signs $[ ]$, $\{ \}$, $\langle \rangle$, $/$ and $\backslash$,

$D$ is a family of 1-1 mappings $D = \{ D_{\mathfrak{e}} : \mathfrak{Srt} \rightarrow \mathfrak{P} \}_{\mathfrak{e} \in \mathfrak{Srt}}$.

If $\mathfrak{Srt}$ is a one-element set, the first part of the definition above is equivalent to the definition of an alphabet of an algorithmic language given in /Ban/. The family of mappings $D$ has been introduced in order to assign to each operator a predicate characterizing its domain.

To a given alphabet an $\mathfrak{Srt}$-sorted algorithmic language with partial operations will be constructed in a similar way as presented in /Mir/ or /Ban/ respecting the modifications caused by the occurrence of different sorts of individuals.

1.2. Let $A = (\Sigma, D)$ be an alphabet in the sense of 1.1. Then by the $\mathfrak{Srt}$-sorted algorithmic language with partial operations for $A$ we understand the system of subsets of $\Sigma^*$ ($T, F, S, FS, FST, FSF$) where:

The set of classical terms $T$ is the disjoint union of the sets $T_{\mathfrak{e}}$ of terms of sort $\mathfrak{e}$, the set $F$ is the set of classical open formulas, both constructed as usual (see /Ass/, for the one-sorted case /Ra-Si/). The occurrence of terms of different sorts must be respected as follows:

if $\varphi(t_1, \ldots, t_n) \in T$ (resp. $F$), where for $i = 1, \ldots, n$ $t_i \in T_{\mathfrak{e}_i}$, then the operator $\varphi$ (resp. predicate $\varphi$) has the arity $\mathfrak{e}_1 \cdots \mathfrak{e}_n$.

The set $S$ of substitutions is the set of all words over $\Sigma^*$ of the form $[x_1/t_1 \cdots x_n/t_n a_1/\alpha_1 \cdots a_m/\alpha_m]$, where $x_1, \ldots, x_n$ (resp. $a_1, \ldots, a_m$) denote pairwise different individual (resp. propositional) variables ($n$ as well as $m$ may be equal to zero), $\alpha_1, \ldots, \alpha_m$ are open formulas and for each $i = 1, \ldots, n$ holds: if $x_i \in V_{\mathfrak{e}_i}$, then $t_i \in T_{\mathfrak{e}_i}$. 