A FORMALIZED PROOF SYSTEM FOR TOTAL CORRECTNESS OF WHILE PROGRAMS

J.A. Bergstra
Department of Computer Science
University of Leiden
Wassenaarseweg 80
2333 AL Leiden

J.H. Klop
Department of Computer Science
Mathematical Centre
Kruislaan 413
1098 SJ Amsterdam

ABSTRACT

We introduce datatype specifications based on schemes, a slight generalization of first order specifications. For a schematic specification \((\Sigma, \Xi)\), Hoare's Logic \(HL(\Sigma, \Xi)\) for partial correctness is defined as usual and on top of it a proof system \((\Sigma, \Xi) \vdash p \rightarrow S +\) for termination assertions is defined. The system is first order in nature, but we prove it sound and complete w.r.t. a second order semantics. We provide a translation of a standard proof system \(HL_T(A)\) for total correctness on a structure \(A\) into our format.

O. INTRODUCTION

In this note we will present a formalized proof system for total correctness of while-programs. Its merits should be first of all that it acts as a first order proof system (although we can, at this moment, only prove a soundness result w.r.t. a second order semantics which allows fewer models for a specification than the usual first order semantics would do). The advantage of having a formalized proof system \((\Sigma, \Xi) \vdash p \rightarrow S +\) for program termination which is just as first order as Hoare's logic \(HL(\Sigma, \Xi) \vdash \{p\}S\{q\}\) for partial correctness is both the possibility of mechanisation and the effect of giving a firm basis for a logical (proof theoretic) investigation of the system.

An essential point is that we want to base our proof system on a specification \((\Sigma, \Xi)\) rather than on a structure \(A\), which is done by most authors. For Hoare's Logic there is no strict need either to consider \(HL(A)\) for a fixed datastructure \(A\), and the more general case of \(HL(\Sigma, \Xi)\) is clearly of substantial importance.

In various fairly standard approaches to total correctness, such as in HAREL [7] and [8] for deterministic sequential processes and in APT & OLDEROG [1] and GRÜMBERG
et al. [6] for fair parallel computation the essence of using a fixed domain A is in
the assumption that certain parts of A, as a many-sorted algebra, are well-ordered.
This gives rise to quite natural proof rules like the system $\text{HL}_T(A)$ that we explain
in section 1.1 in order to compare it with our system.

Instead we will develop a device called \textit{schemes} which constitutes a slight gene-
ralization of the first order predicate logic. For a specification with schemes we
write $(\Sigma, E)$ (whereas $(\Sigma, E)$ denotes a specification with $E \subseteq \mathcal{L}(\Sigma)$). Using schemes we
can work in quite a flexible way with signature extensions, a method that proved to
be useful and to be of first order character in BERGSTRA & KLOP [2]. Thus we obtain
a proof system for termination assertions $(\Sigma, E) \vdash p \rightarrow S \downarrow$ on top of a logic for par-
tial correctness, in the same way as in BERGSTRA & KLOP [2] proof systems for program
inclusion are obtained from a partial correctness logic.

We will now sum up the main notations and results.

For a specification $(\Sigma, E)$ with $\Sigma$ a set of schemes, the logic of partial correct-
ness $\text{HL}(\Sigma, E)$ brings nothing new. A proof system $(\Sigma, E) \vdash p \rightarrow S \downarrow$ is then defined such
that soundness can be shown for a semantics $\models_S$ in Lemma 5.

As a relation of $(\Sigma, E)$, $p$ and $S$, $\vdash$ is recursively enumerable, thus deserving its
denotation as a proof system.

Given a fixed $A$ let $E_A$ be the set of all schemes $\Phi$ over $\Sigma_A$ that are true in $A$ in
the sense of $\models_S$. There is the following completeness result:

\textbf{THEOREM (9.2)} $(\Sigma_A, E_A) \vdash p \rightarrow S \downarrow \iff A \models p \rightarrow S \downarrow$.

In order to compare our system with a usual formalism using well-ordered sets we take
the notation $[p] \subseteq S [q]$ for total correctness (i.e. $[p] \subseteq S [q] \iff (p) \subseteq S (q) \& p \rightarrow S \downarrow$) and define a system $\text{HL}_T(A) \vdash [p] \subseteq S [q]$ for datastructures $A$ with a fixed well-ordering
$\leq$ on it. Then we define a canonical specification $(\Sigma_A, E_A^\subseteq)$ of such $A$ and state the
following result:

\textbf{THEOREM (11.1)} $\text{HL}_T(A) \vdash [p] \subseteq S [q] \Rightarrow \text{HL}(\Sigma_A, E_A^\subseteq) \vdash (p) \subseteq S (q)$ and $(\Sigma_A, E_A^\subseteq) \vdash p \rightarrow S \downarrow$.

This result says that the proposed formalism can be used to represent methods using
well-ordered sets.

Some final remarks should be made. First of all it would be nice to have a logic
for total correctness which is of a first order nature and which is sound and complete
for a semantics of specifications and programs which is of first order nature as well.
For partial correctness the corresponding problem was solved in BERGSTRA & TUCKER [5].
There a so called axiomatic semantics for while-programs is given such that $\text{HL}$ is
sound and complete for it in a most general and first order way. It is not clear to
us whether or not a similar result can be obtained for total correctness. Anyhow, if we
consider simultaneously first order semantics for specifications and the operational
semantics (which is not first order) for programs, a proof system $\vdash$ for $(\Sigma, E) \vdash p \rightarrow S \downarrow$