ON THE EQUIVALENCE OF WEAK SECOND ORDER AND NON-STANDARD TIME SEMANTICS FOR VARIOUS PROGRAM VERIFICATION SYSTEMS

by

J.A. Makowsky and I. Sain

Technical Report #395
January 1986

1. Most of the contribution of the first author was made while he was visiting Budapest during an extended stay at the Forschungsinstitut fur Mathematik, ETH-Zentrum, 8092 Zurich, Switzerland.

On the equivalence of weak second order and nonstandard time semantics for various program verification systems.

J.A. Makowsky
Department of Computer Science, Technion - Israel Institute of Technology, Haifa, Israel
and
I. Sain
Mathematical Institute, Hungarian Academy of Sciences, Budapest, Hungary

Abstract: We show how to derive Leivant's characterization of Floyd-Hoare Logic in weak second order logic ([Le85]) directly from Csirmaz's characterization of Floyd-Hoare logic in Nonstandard Logics of Programs ([Cs80]). Our method allows us to spell out the precise role of the comprehension axiom in weak second order logic. We then prove similar results for other program verification systems (suggested by Burstall and Pnueli) and identify exactly the comprehension axioms corresponding to those systems.

Introduction.

In a recent publication ([Le85]) Leivant proposes a differentiation between the "logical" and the "mathematical" part of program verification. "Logical" here means provable by first order means, whereas "mathematical" refers to second order properties of Arithmetic or fragments thereof. Previous attempts to make this difference precise include Cook completeness (cf. [Apt81]) and the use

1 Most of the contribution of the first author was made while he was visiting Budapest during an extended stay at the Forschungsinstitut für Mathematik, ETH-Zentrum, 8092 Zürich, Switzerland.
The results of the Hungarian School come in many variations. The variations concern either the class of programs under consideration or the semantics chosen. The earliest predecessor of theorem 2 may be found in [AN78]. A simple proof of theorem 2 for deterministic programs may be found in [Sa85, Sa85a]. The original proof for deterministic programs appears in [Cs81]. It has an equivalent form in theorem 9 of [ANS82]. The semantics can be given either using time structures and program traces in time or by considering relational traces of the programs. We follow here the time approach. The relational approach is discussed in [Cs81, Sa85, Sa85a, Sa85b]. In [Sa85b] the analogue of theorem 2 for relational semantics is also proved. It is equally easy as in section 4 to prove the equivalence of weak second order semantics of Floyd-Hoare logic to the relational semantics.
There is a second point to be made: It is questionable whether the equivalence of weak second order logic and Floyd-Hoare logic make the latter so special. In [Ne82, Sa83, Sa85a, SM87] characterizations in terms of nonstandard-time semantics are given for temporal oriented program verification systems such as the one proposed by R.Burstall [Bu74] and A. Pnueli [Pnu81]. These characterizations show that Burstall’s system has strictly greater program verifying power than the Floyd-Hoare system, and that Pnueli’s system is strictly stronger than Burstall’s. It is an interesting open problem to identify that first order approximation of second order logic which corresponds exactly to Burstall’s and Pnueli’s system.

Our paper is organized as follows:

In section 1 we give a very simplified definition of Floyd-Hoare logic. It is simple enough to allow a discussion of the relevant points but it is rich enough to derive from it the general case.

In section 2 we give a similar treatment to weak second order logic and state Leivant’s theorem.

In section three we present nonstandard-time semantics in its rudimentary form and state Csirmaz’s theorem. In section 4 we show how one can obtain Leivant’s theorem from Csirmaz’s theorem.

In section 5, finally, we discuss further directions of research suggested by the equivalence of weak second order semantics and nonstandard-time semantics.

Acknowledgments: The first author is indebted to N. Frances, S. Katz and A. Pnueli for valuable discussions.
1. Floyd-Hoare Logic.

In this note we use Csirmaz's general notion of a program as defined e.g. in [Cs80,Cs81]. We now recall its definition.

**Notation:** Throughout, let $\tau$ be an arbitrary but fixed similarity type (a set of relation, function and constant symbols). By a $\tau$-formula we understand a first order formula with equality using as non-logical symbols only those in $\tau$. A $\tau$-structure ($\tau$-model) is a first order structure (model) for the similarity type $\tau$.

For every natural number $n \in \omega$, let $F^v_{\tau,n}$ denote the set of all $\tau$-formulas which have their free variables among $\{x_i: i < n\}$. By a $\tau$-theory $Th$ we understand a set of $\tau$-formulas in $F^v_{\tau,0}$ which is consistent (which has a first order model).

**Definition 1.1:** Let $Th$ be a $\tau$-theory and $\Pi \in F^v_{\tau,2k}$ be a formula with $2k$ free variables. We say that $\Pi$ defines a (non-deterministic, $k$-ary) $Th$-program if

$$Th \models \forall \bar{x} \exists \bar{y} \Pi(\bar{x},\bar{y}).$$

Here $\bar{x}$ stands for $x_0, x_1, \ldots, x_{k-1}$ and $\bar{y}$ stands for $x_k, x_{k+1}, \ldots, x_{2k}$. Throughout this note $k$ remains fixed.

Intuitively, $\Pi(\bar{x},\bar{y})$ defines the state transition relation of our program (and not the input/output relation). Sometimes $\Pi$ is called a state transducer. The input/output relation of $\Pi$ is the transitive closure of the relation defined by $\Pi$. This approach is taken here to simplify technicalities. It is justified by the fact that "every program" is equivalent to a program with one while-loop only, cf [Ha80]. More precisely, in [Cs80,Cs81] it is shown that all nondeterministic block diagram programs can be written in this form. Also the regular programs in dynamic logic and the while programs in [Le85] fall into this class. Therefore a proof of Leivant's theorem for our class of programs is equivalent to the theorem for its original class of while programs in [Le85].

**Definition 1.2:** Let $Th$ be a $\tau$-theory and $k \in \omega$. Let $\varphi, \psi \in F^v_{\tau,k}$ and $\Pi$ is a $k$-ary $Th$-program.
(i) A \( \text{Th-partial correctness assertion (Th-p.c.a.)} \) is a formula of the form \( \varphi \rightarrow [\Pi] \psi \).

(ii) We say that \( \varphi \rightarrow [\Pi] \psi \) is provable in Floyd-Hoare logic from a (data) theory \( \text{Th} \). In symbols, \( \text{Th} \vdash_{FH} \varphi \rightarrow [\Pi] \psi \), iff there is a \( \tau \)-formula \( \chi \in F_{\tau}^k \) such that

\[
\text{Th} \vdash \varphi(x) \rightarrow \chi(x)
\]

\[
\text{Th} \vdash \chi(x) \land \Pi(x,y) \rightarrow \psi(y)
\]

and

\[
\text{Th} \vdash \chi(x) \land \Pi(x,x) \rightarrow \psi(x).
\]

Here \( \vdash \) denotes first order provability.

2. Weak second order logic.

Following the terminology of [Le85] we introduce now (Henkin type) weak second order logic.

Definition 2.1:

(i) Let \( D \) be a \( \tau \)-structure with universe \( D \). Let \( S \) be a subclass of all the finitary relations over \( D \), i.e. \( S \subseteq \bigcup_{n \in \omega} P(D^n) \). Let \( \in \) be the standard membership relation between elements of \( D \) and \( S \). We call the two-sorted structure \( <D,S,\in> \) a weak second order model over \( D \) if it satisfies the first order comprehension scheme, i.e. for every \( \tau \)-formula \( \varphi(x) \) there is a set \( X \in S \) such that for every \( \bar{a} \in D^n \), \( \bar{a} \in X \) iff \( D \models \varphi(x)[\bar{a}] \).

(ii) Let \( \text{Th} \) be a \( \tau \)-theory and \( \varphi\) a \( \tau \)-formula. We denote by \( \text{Th} \models_{w1} \varphi \) the (semantic) consequence relation for weak second order logic, i.e. that \( \varphi \) is true in every weak second order model satisfying \( \text{Th} \).

Definition 2.2 ([Le85] - section 2.1.): Let \( \Pi \) be a \( \text{Th} \)-program. We define the weak second order approximation to the input/output relation of \( \Pi \) to be the second order formula.
Makowsky-Sain

$M_{\Pi}(\bar{u}, \bar{v}) \overset{\text{def.}}{=} (\Pi(\bar{u}, \bar{v}) \land \forall R[R(\bar{u}) \land \text{Prog}_{\Pi}[R] \Rightarrow R(\bar{v})])$

with

$\text{Prog}_{\Pi}[R] \overset{\text{def.}}{=} \forall \bar{z} \forall \bar{w}[R(\bar{z}) \land \Pi(\bar{z}, \bar{w}) \Rightarrow R(\bar{w})]$.

In $M_{\Pi}$ the only second order variable is $R$.

We are now in a position to state the main theorem of [Le85]:

**THEOREM 1** (Leivant's completeness of Floyd-Hoare Logic):

Let $\varphi \rightarrow [\Pi] \psi$ be a $\text{Th}$-partial correctness assertion. Then the following are equivalent:

1. $\text{Th} \models_{FH} \varphi \rightarrow [\Pi] \psi$
2. $\text{Th} \models_{\Pi} \forall \bar{u} \forall \bar{v}[\varphi(\bar{u}) \land M_{\Pi}(\bar{u}, \bar{v}) \Rightarrow \psi(\bar{v})]$.

3. Nonstandard-time semantics.

We give here a very brief description of the semantics used in [ANS82, Cs80, Cs81, Pa85 e]. It is based on a model of time, such as the integers with successor, and its nonstandard models. Therefore it is called **nonstandard-time semantics**.

Let $T$ be an arbitrary first order model elementarily equivalent to $\langle \omega, 0, \text{suc} \rangle$. We call such a $T$ a **time structure**.

**Definition 3.1**: (cf. [Cs80], Definition 1.1)

Let $\text{Th}$ be a $\tau$-theory, $D$ be a model of $\text{Th}$, $\Pi$ be a $\text{Th}$-program and $T$ a time structure. A function $Q : T \rightarrow D^k$ is called a $T$-**trace of** $\Pi$ in $D$ iff the following two (infinitary) formulas hold in $D$.

1. $D \models \Pi(Q(i), Q(\text{suc}(i)))$ and
2. for every $\tau$-formula $\gamma$

   
   $D \models (\gamma(Q(0)) \land \Pi \overset{\tau}{\Pi}(\gamma(Q(i)) \rightarrow \gamma(Q(\text{suc}(i)))) \rightarrow \Pi \overset{\tau}{\Pi}(\gamma(Q(i))))$. 


We are now in a position to state the main theorem of [Cs80]:

**THEOREM 2** (Csirmaz's completeness of Floyd-Hoare Logic):

Let $\varphi \to [\Pi]\psi$ be a $\mathcal{Th}$-partial correctness assertion. Then the following are equivalent:

1. $\mathcal{Th} \vdash_{PH} \varphi \to [\Pi]\psi$
2. $\mathcal{Th} \vdash N \varphi \to [\Pi]\psi$

4. The equivalence of nonstandard-time semantics and weak second order semantics.

We now proceed to give a short proof of theorem 1 using only theorem 2. We state more precisely:

**THEOREM 3**: Let $\varphi \to [\Pi]\psi$ be a $\mathcal{Th}$-partial correctness assertion. Then the following are equivalent:

1. $\mathcal{Th} \vdash_{PH} \varphi \to [\Pi]\psi$
2. $\mathcal{Th} \vdash \forall u \forall v [\varphi(u) \land M_\Pi(u,v) \to \psi(v)]$
3. $\mathcal{Th} \vdash N \varphi \to [\Pi]\psi$

**Proof**: We prove only (2) $\Rightarrow$ (3), since (1) $\Rightarrow$ (2) is trivial ([Le85]) and (3) $\Rightarrow$ (1) is theorem 2.

Assume (2). To prove (3) let $D \models \mathcal{Th}$ and let $Q: T \to D^k$ be a $T$-trace of $\Pi$ in $D$. 

\(D \models \forall i \in T \left[ Q(i) = Q(\text{suc}(i)) \to \psi(Q(i)) \right].\)

We are now in a position to state the main theorem of [Cs80]:
Assume $D \models \varphi(Q(0)) \land Q(j) = Q(suc(j))$ for some fixed $j \in T$. It suffices to prove that $D \models \psi(Q(j))$.

Let $S$ be the set of all (finitary) relations definable by some $\tau$-formula in $D$. Clearly, $M = <D, S, \varepsilon>$ is a weak second order model over $D$.

Claim: $M \models M_{\Pi}(Q(0), Q(j))$.

By $Q(j) = Q(suc(j))$ and definition 3.1 (i) we have $M \models \Pi(Q(j), Q(j))$.

Now assume $R \in S$ is such that $M \models R(Q(0)) \land Prog[R]$. Then $D \models R(Q(0))$ and $D \models \bigwedge_{t \in T} [R(Q(t)) \rightarrow R(Q(suc(t)))].$ Therefore, by the induction scheme of 3.1.(ii), we have $D \models R(Q(j))$. So we conclude that $M \models M_{\Pi}(Q(0), Q(j))$, which proves our claim.

So we have $M \models \varphi(Q(0)) \land M_{\Pi}(Q(0), Q(j))$ and $M \models \psi(Q(j))$ by our assumption. Thus $D \models \psi(Q(j))$. *

5. Comparing the Expressive Power of Other Programming Logics

In the framework of nonstandard-time semantics one can also model approaches to program verification as advocated by Burstall [Bu74] and Pnueli [Pnu81]. They are based originally on modal and temporal logic respectively. In [Ne82, Sa8Sc] those systems were analyzed in terms of to nonstandard-time semantics. There are three degrees of freedom here:

One can vary the time structure by augmenting it with a linear order and adding various axioms;

one can vary the induction scheme ((2) in definition 3.1 and one can also vary the first order description of the partial correctness statement of definition 3.2.

One can then compare various such systems in several ways. They all have in common that one singles out a class of program assertions $\mathcal{X}$ (partial correct-
ness, total correctness, safety or liveness properties) by their first order approximations. In [Ne82,Sa85c] such a comparative study was made in great detail for the class of partial correctness statements. It is obvious how (and possibly tedious) to carry out such a study for other classes of program assertions. For a more formal treatment of safety and liveness properties, the reader should consult [AD83].

The point we wish to make here is the following: The same degrees of freedom are also inherent in Leivant’s approach, though they reduce to only two:

One can vary the first order approximation to second order logic by modifying the comprehension scheme and by possibly adding other axioms which the family of sets (relations) should satisfy; and one can again vary the first order approximation to the partial correctness assertion. Having this in mind one is led to question the canonicity of Leivant’s theorem. Rather than telling us that the Floyd-Hoare method is “right” because it is equivalent weak second order program verification, Leivant’s theorem says merely that the Floyd-Hoare method is equivalent to weak second order program verification with a specific comprehension scheme. The approach sketched above suggests the following host of additional questions:

1) Identify other classes of program assertions which are expressible in weak second order logic (with various comprehension schemes and possibly other natural axioms).

2) Characterize other proof systems of program logics by their exact counterparts in weak second order logic.

To carry out such a program one merely needs a way to translate the results of [Ne82,Sa85c] into the formalism of weak second order logic. In the last section we show how Leivant’s original result can readily be obtained from Csirmaz’s theorem. This should give the reader enough insight to understand
the connection between nonstandard-time semantics and weak second order semantics. For lack of space we have to refrain from defining in detail other reasonable program logics and their proof systems. For some of the systems introduced in [Ne74], in particular the one called Mod (in [Sa85c] called Burstall) the weak second order logic of the partial correctness statements has been successfully determined and shown equivalent to nonstandard-time semantics augmented but various induction schemes.

References.


Makowsky-Sain

[Sa85c]