Theoretical Computer Science 612 (2016) 23–28

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Completeness of Hoare logic with inputs over the standard model

Zhaowei Xu [a,*], Yuefei Sui [b], Wenhui Zhang [a]

[a] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
[b] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Article history:
Received 18 September 2014
Received in revised form 5 March 2015
Accepted 3 August 2015
Available online 7 August 2015
Communicated by M. Hofmann

Keywords:
Hoare logic
Peano arithmetic
The standard model
Computation
Arithmetical definability
Logical completeness

ABSTRACT

Hoare logic for the set of while-programs with the first-order logical language L and the first-order theory $T \subset L$ is denoted by $HL(T)$. Bergstra and Tucker have pointed out that the complete number theory $Th(N)$ is the only extension T of Peano arithmetic PA for which $HL(T)$ is logically complete. The completeness result is not satisfying, since it allows inputs to range over nonstandard models. The aim of this paper is to investigate under what circumstances $HL(T)$ is logically complete when inputs range over the standard model N. PA^+ is defined by adding to PA all the unprovable Π_1-sentences that describe the nonterminating computations. It is shown that each computable function in N is uniformly Σ_1-definable in all models of PA^+, and that PA^+ is arithmetical. Finally, it is established, based on the reduction from $HL(T)$ to T, that PA^+ is the minimal extension T of PA for which $HL(T)$ is logically complete when inputs range over N. This completeness result has an advantage over Bergstra's and Tucker's one, in that PA^+ is arithmetical while $Th(N)$ is not.

1. Introduction

Hoare logic is a formal system for the manipulation of statements about the correctness of while-programs [1,2], which has had a significant impact upon the methods of both designing and verifying programs [3,4]. Hoare logic for the set WP of while-programs with the first-order logical language L and the first-order theory $T \subset L$ is denoted by $HL(T)$ [5]. In what follows, let L be the logical language of Peano arithmetic PA, and let N be the standard model of PA [6].

In [7], Bergstra and Tucker have pointed out that the complete number theory $Th(N)$ is the only extension T of PA for which $HL(T)$ is logically complete. Closer scrutiny of their argument reveals that the incompleteness of $HL(T)$ with $PA \subseteq T \subsetneqq Th(N)$ results from allowing inputs to range over nonstandard models. (For more details, we refer to Theorem 2.1 and Corollary 3.1.4.) However, Tennenbaum's theorem [8] says that addition and multiplication are not computable in nonstandard models. For practical purposes, it would be meaningless to consider computations over nonstandard models. Without further declaration, for a while-program $S \in WP$, the vector $(x_1, x_2, \ldots, x_m)$ of all m program variables $x_1, x_2, \ldots, x_m$ occurring in S will be denoted by $\vec{x}$; the vector $(n_1, n_2, \ldots, n_m)$ of m natural numbers $n_1, n_2, \ldots, n_m \in N$ will be denoted by $\vec{n}$; the connectives will be assumed to distribute over the components of the vectors (for instance, $\vec{n} \in N$ means $n_1, n_2, \ldots, n_m \in N$, and $\vec{x} = \vec{n}$ means $\bigwedge_{i=1}^{m} x_i = n_i$). The aim of this paper is to investigate under what circumstances $HL(T)$ is logically complete when inputs range over N:

* Corresponding author at: University of Chinese Academy of Sciences, China.
E-mail addresses: xuzw@ios.ac.cn (Z. Xu), yfsui@ict.ac.cn (Y. Sui), zwh@ios.ac.cn (W. Zhang).

http://dx.doi.org/10.1016/j.tcs.2015.08.004
0304-3975/

Definition 1.1. $HL(T)$ is logically complete when inputs range over N if for every $S \in WP$ with program variables $\vec{x}$, every $p, q \in L$ (p, q could contain other first-order variables than those in $\vec{x}$), and every $\vec{n} \in N$, $HL(T) \vdash \{p \wedge \vec{x} = \vec{n}\}S\{q\}$ iff $HL(T) \models \{p \wedge \vec{x} = \vec{n}\}S\{q\}$.

According to the classic recursion theory [9], a while-program S produces in N a partial recursive (or recursive for short) function $\vec{y} = f_S^N(\vec{x})$, where $\vec{y}$ is disjoint from $\vec{x}$ and has the same length as $\vec{x}$. By the arithmetical definability of recursive functions [10, Chapter 16], there exists a Σ_1-formula $\alpha_S(\vec{x}, \vec{y}) \in L$ that defines $\vec{y} = f_S^N(\vec{x})$ in N (cf. Definition 3.1.1 and Lemma 3.1.2). Defining $SP(p, S)$ by $\exists \vec{u}(p(\vec{u}/\vec{x}) \wedge \alpha_S(\vec{u}/\vec{x}, \vec{x}/\vec{y}))$, it follows from Theorem 2.2 that for every $PA \subseteq T \subseteq Th(N)$, every $p, q \in L$, and every $S \in WP$, $HL(T) \vdash \{p\}S\{q\}$ iff $T \vdash p(\vec{x}) \wedge \alpha_S(\vec{x}, \vec{y}) \rightarrow q(\vec{y}/\vec{x})$ (cf. Theorem 3.1.3). Observe that if, for every $S \in WP$, f_S^N was defined by α_S in every model M of PA (i.e., for every $\vec{n} \in N$, $f_S^N(\vec{n}) = \vec{y}$ iff $M \models \alpha_S(\vec{n}, \vec{y})$), then $HL(PA)$ would be logically complete when inputs range over N (for p, q, S and $\vec{n}$ as defined in Definition 1.1, $HL(PA) \models \{p \wedge \vec{x} = \vec{n}\}S\{q\}$ iff $PA \models p(\vec{x}) \wedge \vec{x} = \vec{n} \wedge \alpha_S(\vec{x}, \vec{y}) \rightarrow q(\vec{y}/\vec{x})$; moreover, $HL(PA) \vdash \{p \wedge \vec{x} = \vec{n}\}S\{q\}$ iff $PA \vdash p(\vec{x}) \wedge \vec{x} = \vec{n} \wedge \alpha_S(\vec{x}, \vec{y}) \rightarrow q(\vec{y}/\vec{x})$; by the soundness and completeness of the first order logic, $HL(PA) \models \{p \wedge \vec{x} = \vec{n}\}S\{q\}$ iff $HL(PA) \vdash \{p \wedge \vec{x} = \vec{n}\}S\{q\}$). However, there exist $S \in WP$ and $\vec{n} \in N$ such that $N \models \forall \vec{y} \neg \alpha_S(\vec{n}, \vec{y})$ and $PA \nvdash \forall \vec{y} \neg \alpha_S(\vec{n}, \vec{y})$ (cf. Theorem 3.2.1), which together with the completeness of the first order logic implies that f_S^N is not defined by α_S in some model of PA (i.e., for some $M \models PA$, $M \models \exists \vec{y} \alpha_S(\vec{n}, \vec{y})$, but f_S^N is not defined for $\vec{x} = \vec{n}$). Hence PA^+ will be defined by adding to PA all such Π_1-sentences $\forall \vec{y} \neg \alpha_S(\vec{n}, \vec{y})$ that $N \models \forall \vec{y} \neg \alpha_S(\vec{n}, \vec{y})$ and $PA \nvdash \forall \vec{y} \neg \alpha_S(\vec{n}, \vec{y})$. It will be proved that for every $S \in WP$, f_S^N is defined by α_S in all models of PA^+, and that PA^+ is arithmetical. Finally, it will be established, based on the reduction from $HL(T)$ to T, that PA^+ is the minimal extension T of PA for which $HL(T)$ is logically complete when inputs range over N.

Related work. Cook [11] considered the relative completeness of Hoare logic with the expressiveness condition: $Th(N)$ is the only extension T of PA for which $HL(T)$ is complete relative to N. Kozen and Tiuryn [12] investigated the completeness of propositional Hoare logic with assertions and programs abstracted to propositional symbols.

The rest of this paper is structured as follows: the basic preliminary results are presented in Section 2; the definition of α_S is shown in Section 3.1; the definition of PA^+ and its properties are shown in Section 3.2; the strong completeness of $HL(PA^+)$ is shown in Section 3.3; concluding remarks are given in Section 4.

2. Preliminaries

First some notations are introduced: in syntax, we write $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$ to denote the negation, conjunction, disjunction, conditional, biconditional connectives and the universal, existential quantifiers; in semantics, we write $\sim, \&, |, \Rightarrow, \Leftrightarrow, \mathbf{A}, \mathbf{E}$ to denote the corresponding connectives and quantifiers.

Let L be the logical language of Peano arithmetic PA with the signature $\Sigma = \{0, 1, +, \cdot, <\}$. The distinguished axiom of PA is the induction axiom scheme, i.e. $\varphi(0, \vec{y}) \wedge \forall x(\varphi(x, \vec{y}) \rightarrow \varphi(x+1, \vec{y})) \rightarrow \forall x \varphi(x, \vec{y})$ with $\varphi(x, \vec{y}) \in L$. Theorem 16.13 in [10] says that for each $\exists$-rudimentary (or alternatively Σ_1) sentence $\varphi \in L$, $N \models \varphi$ iff $PA \vdash \varphi$. For simplicity, the sum of 1 with itself n times is abbreviated as n. We use n to denote both a closed term and a natural number, and use M to denote both a model and its domain, which will be clear from the context. Besides the standard model N, PA has nonstandard models, among which only the countable M will be considered: the order relation of M is linear [10, Section 25.1]; M has a standard part N^M which is isomorphic to N; each element of N^M is denoted by n as well.

Let $\langle x, y \rangle$, $L(z)$ and $R(z)$ be the recursive functions with $\langle L(z), R(z) \rangle = z$, $L(\langle x, y \rangle) = x$ and $R(\langle x, y \rangle) = y$ [13, Theorem 2.1]. For notational convenience, we denote $(L(z), R(z))$ by $\overline{z}$. The functions $\langle x, y \rangle$ and $\overline{z}$ can be extended to n-tuples (for each $n \in N$) by setting $\langle x_1, x_2, \dots, x_n \rangle = \langle x_1, \langle x_2, \dots, x_n \rangle \rangle$ and $\overline{\langle x_1, x_2, \dots, x_n \rangle} = (x_1, \overline{\langle x_2, \dots, x_n \rangle})$. Let $(x)_i$ be the recursive function such that for each finite sequence $a_0, a_1, \dots, a_n$ of natural numbers, there exists a natural number w such that $(w)_i = a_i$ for all $i \le n$ [13, Theorem 2.4]. Note that these functions are all arithmetically definable (or arithmetical for short) by Σ_1-formulas of L [10, Chapter 16].

Based on the first-order logical language L, together with the program constructs (:=, ;, *if*, *then*, *else*, *fi*, *while*, *do*, *od*), a while-program S is defined by $S ::= x := E \mid S_1; S_2 \mid \textit{if } B \textit{ then } S_1 \textit{ else } S_2 \textit{ fi} \mid \textit{while } B \textit{ do } S_0 \textit{ od}$, where an expression E is defined by $E ::= 0 \mid 1 \mid x \mid E_1 + E_2 \mid E_1 \cdot E_2$, and a boolean expression B is defined by $B ::= E_1 < E_2 \mid \neg B_1 \mid B_1 \rightarrow B_2$. The set of all such while-programs is denoted WP. Unless otherwise stated, let the program variables considered below occur among $\vec{x}$, the vector of all program variables of the target program. For a model M of L, let v be an assignment over M for all the first order variables (including $\vec{x}$), let $v(\vec{x})$ be the vector of elements of M assigned to $\vec{x}$ at v, and let $v(\vec{y}/\vec{x})$ be an assignment as v except that $v(\vec{y}/\vec{x})(\vec{x}) = \vec{y}$. For every $S \in WP$ and every model M of L, the input–output relation R_S^M is a binary relation on the set of assignments over M inductively defined as follows:

- $(v, v') \in R^M_{x:=E} \Leftrightarrow v' = v(E^{M,v}/x)$, where $E^{M,v}$ receives the standard meaning;
- $(v, v') \in R^M_{S_1;S_2} \Leftrightarrow (v, v') \in R^M_{S_1} \circ R^M_{S_2}$, where $(z, z') \in R_1 \circ R_2 \Leftrightarrow \mathbf{E}z''((z, z'') \in R_1 \ \& \ (z'', z') \in R_2)$;
- $(v, v') \in R^M_{\textit{if } B \textit{ then } S_1 \textit{ else } S_2 \textit{ fi}} \Leftrightarrow M, v \models B \ \& \ (v, v') \in R^M_{S_1} \mid M, v \not\models B \ \& \ (v, v') \in R^M_{S_2}$, where $M, v \models B$ and $M, v \not\models B$ receive the standard meanings;
- $(v, v') \in R^M_{\textit{while } B \textit{ do } S_0 \textit{ od}} \Leftrightarrow \mathbf{E}i \in N, \mathbf{E}\vec{x}_0, \dots, \vec{x}_i \in M \ (v(\vec{x}) = \vec{x}_0 \ \& \ \mathbf{A}j < i(M, v(\vec{x}_j/\vec{x}) \models B \ \& \ (v(\vec{x}_j/\vec{x}), v(\vec{x}_{j+1}/\vec{x})) \in R^M_{S_0}) \ \& \ v' = v(\vec{x}_i/\vec{x}) \ \& \ M, v' \not\models B)$.