HOARE’S LOGIC FOR NONDETERMINISTIC REGULAR PROGRAMS: A NONSTANDARD APPROACH

M. Teresa HORTALÁ-GONZÁLEZ and Mario RODRÍGUEZ-ARTALEJO

Sección Departamental de Informática y Automática, Facultad de Matemáticas,
Universidad Complutense de Madrid, Spain

Communicated by C. Böhm
Received September 1985
Revised April 1988

Abstract. This paper studies Hoare’s logic for nondeterministic regular programs (with unbounded nondeterminism) from the point of view of nonstandard dynamic logic. We define a so-called continuous semantics which allows certain “infinitely long computations” and compare it with usual semantics, proving among other things the equivalence of both over “reasonable data types”. We also establish a completeness theorem for Hoare’s calculus relative to continuous semantics, thereby generalizing a previous result of Csirmaz. The proof makes use of a normal form for regular programs which is perhaps interesting in its own right.

1. Introduction

The issue of formal program verification has given rise to many logical formalisms which serve to state and prove different kinds of program properties.

Hoare-like partial correctness calculi (cf. [20]) use finitary proof rules, but the (usually highly undecidable) theory of an expressive interpretation must be taken as an axiom set in order to obtain relative completeness in Cook’s sense (cf. [11]). Total correctness calculi must take termination into account, and this is known to involve higher order induction principles (cf. [23]). Algorithmic and dynamic logic, which go beyond partial and total correctness properties, also use either highly undecidable axiom sets (as in Harel’s arithmetical dynamic logic, cf. [18]) or infinitary proof rules (as in Mirkowska’s algorithmic logic, cf. [24]) to achieve completeness.

First-order formalisms are appealing because of their simplicity and well understood model theory. Moreover, decidable axiom sets seem most appropriate in the context of computer science. The nonstandard approach to program logic maintains both of these advantages without having to pay for it with incompleteness. This is achieved through nonstandard semantics which allow computations to proceed in a nonstandard time, internal to the interpretation. Although this seems strange and counterintuitive at first sight, it can be justified by the following considerations:

(a) Sufficiently rich first-order axioms are enough to capture the temporal properties needed for program verification.
(b) This allows program verification methods to be characterized, compared and
discovered in a uniform framework, by considering the different temporal assump-
tions on which they implicitly rely.

(c) Moreover, nonstandard models are useful (even necessary) to supply seman-
tical characterizations of standard unprovability results. (By this we mean that a
nonstandard model is sometimes needed to show an instance of bad behaviour of
a program whose good behaviour cannot be proved in some standard verification
system; cf. Example 5.5).

Andréka, Németi and Sain [3, 28, 29, 30] have worked out a very fruitful approach
to nonstandard dynamic logic; a quite detailed account of their work can be found
in [26]. Some more or less related approaches, placed in different contexts, are
those in [7, 8, 10, 12, 13, 14, 17], among others. In these papers, one finds different
definitions of nonstandard semantics, where the internal time we referred to above
is not always handled explicitly. Nevertheless, they all share a definitely first-order
spirit and allow for the kind of applications described in (b) above. At a deeper
level, some of them are connected through transfer principles which give a method
of mapping models into models; for an example of this, see the proof of Theorem
3.1 in [26].

This paper investigates a nonstandard approach to Hoare’s logic for a simple,
but computationally powerful, programming language with regular control struc-
tures and unbounded nondeterminism. We introduce a standard Hoare-like partial
correctness calculus and define continuous semantics, which is a relational nonstan-
dard semantics closely related to ideas in [1, 2, 8, 12]. We compare continuous
semantics with the standard one and show that both coincide in most “reasonable”
situations (cf. Theorem 2.10). These results are related to the work of Bergstra and
Tucker [7]. In Theorem 3.1, we show that the partial correctness calculus is sound
relative to continuous semantics. The reciprocal completeness result is established
in Theorem 5.1, for the case of programs without nested loops. The proof combines
techniques adapted from [12] and [29] (these authors deal with deterministic, un-
structured programs) with an application of Craig’s interpolation theorem (cf. [31]),
which is the key to coping with the sequential composition operator. To extend the
completeness result to arbitrary programs, we introduce an algorithm which trans-
lates them to a normal form without nested loops. The idea behind the normal form
is essentially the same as in Böhm-Jacopini’s theorem [9], but the precise analysis
of the equivalence between programs and their normal forms needs some care (see
Theorem 4.3).

As mentioned above, different nonstandard semantics can serve as the basis for
deriving completeness results, as well for Hoare’s logic as for other verification
methods and/or logics of programs. Our choice of continuous semantics was
motivated mainly by its conceptual simplicity: no explicit mention of time and no
multisorted structures are needed. Nevertheless, the work by Andréka, Néméti and
Sain shows the possibility of characterizing it in a temporal framework; this is partly
what we meant by our comments on transfer principles.
This paper is a substantially improved and extended version of a communication presented at the 12th ICALP [21]. Our main objective in writing it was to investigate the behaviour of nonstandard semantics in the case of structured programs. For unstructured programs, the verification of partial correctness is more akin to Floyd's method [16] and can be reduced to the consideration of a unique global invariant. This is no longer the case for our language; in fact, we do not know if the completeness Theorem 5.1. holds for programs which include nested loops. We have considered unbounded nondeterminism because the notion had been recognized as useful (for instance, to model fair termination; cf. [4, 19]) and did not introduce any major complications. This would not have been the case in an investigation of total correctness properties. Indeed, total correctness presents difficulties of its own from the point of view of nonstandard semantics, as shown in [30].

The rest of the paper is organized in the following way: in Section 2 we introduce the language of nondeterministic regular programs and study their standard and continuous semantics. Section 3 is devoted to Hoare's calculus; its axioms and rules are introduced, and the proofs of some auxiliary proof theoretical lemmas are sketched. In Section 4 we state and prove the normal form theorem, which serves as a basis for the soundness and completeness results in Section 5. Section 6 is a brief conclusion.

2. Nondeterministic regular programs

We shall use the following basic notation from mathematical logic.

Let $\tau$ be a finite signature, consisting of

- $C(\tau) = \{c, d, e, \ldots \}$ set of constant symbols
- $F(\tau) = \{f, g, h, \ldots \}$ set of ranked function symbols
- $R(\tau) = \{P, Q, R, \ldots \}$ set of ranked relation symbols.

We assume $x, y, z, \ldots \in V$ individual variables, $t = t(x) \in T^\tau(x)$ ($\tau$-terms) using at most the variables $x_1, \ldots, x_n$ and $\varphi = \varphi(x) \in L^\tau(x)$ ($\tau$-formulae) with all its free variables among $x_1, \ldots, x_n$. For $\varphi \in L^\tau(x)$ and $t = t_1, \ldots, t_n$, simultaneous substitution will be denoted by $\varphi[t/x]$ or simply $\varphi(t)$.

**Definition 2.1.** The set $RP^\tau(x)$ of nondeterministic regular programs of type $\tau$ in variables $x_1, \ldots, x_n$ is recursively defined by

- $x := ?y. \rho(x, y) \in RP^\tau(x)$ for every $\rho \in L^\tau(x, y)$ (nondeterministic assignment)
- $x \in RP^\tau(x)$ for every $x \in L^\tau(x)$ (test)
- $\alpha, \beta \in RP^\tau(x) \Rightarrow (\alpha \cup \beta), (\alpha; \beta), \alpha^* \in RP^\tau(x)$ (union, composition and iteration).
The intended meaning of \( x := ?y.p(x,y) \) is "nondeterministically replace state \( x \) by a new state \( y \) such that \( p(x,y) \)". This may fail (in the sense of giving rise to no computation) if there is no \( y \) satisfying \( p(x,y) \). Notice also that random assignment and ordinary assignment can be simulated as \( x_j := ? \) shorthand for \( x_j := ?y.A_k \{ y_k = x_j | 1 \leq k \leq n, k \neq j \} \) and \( x_j := t(x) \) shorthand for \( x_j := ?y. \{ y_j = x_j | 1 \leq k \leq n, k \neq j \} \), respectively.

The meaning of the other constructs is well known from dynamic logic; they allow the expression of if, while, Dijkstra’s guarded commands [15], etc. (see [18]), as well as the programs of Csirmaz [12].

The test \( \chi(x) \) is of course highly ineffective, since \( \chi(x) \) stands for an arbitrary first-order formula. We allow this because it does not involve any additional mathematical difficulty.

In the sequel, we sometimes write \( \alpha(x) \) for \( \alpha \in \mathcal{RP}^\tau(x) \).

We are going to interpret programs \( \alpha \in \mathcal{RP}^\tau(x) \) over \( \tau \)-structures of the form

\[
\mathcal{A} = (A, (e^*_c)_{c \in C(\tau)}, (f^*_f)_{f \in F(\tau)}, (P^*_p)_{p \in R(\tau)})
\]

where the domain \( A \) is any nonempty set and the equality symbol \( = \) is implicitly interpreted as the identity on \( A \).

The cartesian product \( S^* = A^n \) represents the set of all possible computation states for programs \( \alpha(x) \) over \( \mathcal{A} \).

Let \( \tau_n = \tau \cup \{ R_\alpha | \alpha \in \mathcal{RP}^\tau(x) \} \) be the infinitary signature which results by adding to \( \tau \) a new \( 2n \)-ary relation symbol for each program. \( R_\alpha \) is intended to reflect the relational semantics of \( \alpha \) as in dynamic logic [18].

**Definition 2.2. (Relational standard semantics).** The standard interpretation \( \mathcal{A}^\text{st} \) of \( \mathcal{RP}^\tau(x) \) over \( \mathcal{A} \) is the \( \tau_n \)-structure

\[
\mathcal{A}^\text{st} = (\mathcal{A}, (R^*_\alpha)_{\alpha \in \mathcal{RP}^\tau(x)})
\]

where the relations \( R^*_\alpha \leq S^*_n \times S^*_n = A^{2n} \) are univocally determined by the axioms:

\[
(= ?) \quad \forall x \forall y (R_{x := y}(x,y) \leftrightarrow R(x,y)),
\]

\[
(?) \quad \forall x \forall y (R_{x:=?}(x,y) \leftrightarrow \chi(x) \land x = y),
\]

(\( \lor \)) \quad \forall x \forall y (R_{a \lor b}(x,y) \leftrightarrow R_a(x,y) \lor R_b(x,y)),

(\( \land \)) \quad \forall x \forall y (R_{a \land b}(x,y) \leftrightarrow \exists z(R_a(x,z) \land R_b(z,y))),

(\( \ast \)) \quad R_{a \ast} = (R_\alpha)^\ast (Kleene’s closure).

Notice that all axioms except (\( \ast \)) are \( \tau_n \)-sentences.

Any program \( \alpha(x) \) may fail at a given state \( a \) if there is no state \( b \) such that \( R^*_\alpha (a,b) \). In particular, this may happen for \( x := ?y.p(x,y) \) or \( \chi(x) \). This kind of failure is of no concern for partial correctness, but must be taken into account in the following definition, because after a failure of the program, no further composition with another program is semantically possible.
Definition 2.3. (Operational standard semantics). Let $\alpha \in \mathrm{RP}^T(x)$ and $\tau_n$-structure $\mathcal{A}$ be given.

The standard computation tree $T^\mathcal{A}_\alpha(a)$ over $\mathcal{A}$ at the state $a \in S^n$ is recursively defined by:

1. For $\alpha = x := ?y. \rho(x, y)$ the tree is as shown in Fig. 1, with (left-hand side) one leaf for every state $b$ such that $\mathcal{A} \models \rho(a, b)$ and (right-hand side) if there is no $b$ such that $\mathcal{A} \models \rho(a, b)$.

2. For $\alpha = ?x. \chi(a)$ the tree is as shown in Fig. 2, with left-hand side if $\mathcal{A} \models \chi(a)$ and the right-hand side if $\mathcal{A} \models \neg \chi(a)$.

3. For $\alpha = (\beta \cup \gamma)$ the tree is as shown in Fig. 3.

4. For $\alpha = (\beta; \gamma)$ the tree is as shown in Fig. 4; that is, $T^\mathcal{A}_\gamma(b)$ is grafted at the leaves of $T^\mathcal{A}_\beta(a)$ labeled by a state $b \in S^n$ and the leaves of $T^\mathcal{A}_\beta(a)$ labeled by fail remain as leaves in $T^\mathcal{A}_\alpha(a)$.

5. For $\alpha = \beta^*$, the tree is as shown in Fig. 5, where $\beta^1 = \beta, \beta^{t+1} = (\beta^1, \beta)$ and $T^\mathcal{A}_\beta(a)$ stands for the one node tree $\cdot a$ (root labeled by $a$).

Following Smullyan’s terminology [32], we mean by a path in a given tree any finite or denumerable sequence of nodes, beginning with the root, such that each term of the sequence (except the last, if there is one) is the predecessor of the next.

![Fig. 1](image1.png)

![Fig. 2](image2.png)

![Fig. 3](image3.png)
By a maximal path or branch we shall mean a path whose last node is a leaf of the tree, or an infinite path. Obviously, branches (resp. paths) in $T^\mathfrak{g}(a)$ correspond to computations (resp. unfinished computations) of $a$ in $\mathfrak{A}$ starting at $a$, under the standard interpretation.

More precisely, the following fact is easily checked.

**Proposition 2.4.** Let $\mathfrak{A}$ be a $\tau$-structure and $a \in RP^\tau(x)$. For any $a, b \in S_n^\mathfrak{g}$ we have

$$R^\mathfrak{g}_\alpha(a, b) \text{ iff there is some leaf labeled by } b \text{ in } T^\mathfrak{g}_\alpha(a).$$

We agree to put $S(T^\mathfrak{g}_\alpha(a))$ for the set of all states $b \in S_n^\mathfrak{g}$ labeling some node of $T^\mathfrak{g}_\alpha(a)$.

We now turn our attention to nonstandard semantics.

**Definition 2.5.** A continuous interpretation $\mathfrak{A}^{ct}$ of $RP^\tau(x)$ over $\mathfrak{A}$ is any $\tau$-structure

$$\mathfrak{A}^{ct} = (\mathfrak{M}, (R^\mathfrak{g}_\alpha)_{\alpha \in RP^\tau(x)}),$$

where the relations $R^\mathfrak{g}_\alpha \subseteq S_n^\mathfrak{g} \times S_n^\mathfrak{g} = A^{2n}$ are chosen in such a way that $\mathfrak{A}^{ct}$ becomes a model of the set $C_n \subseteq L^\tau$ formed by all axioms ($=$ ?)-(;), from Definition 2.2, and instead of (*), the first-order sentences

- $\text{refl}_{\alpha}^* = \forall x R_{\alpha}^*(x, x)$
- $\text{ext}_{\alpha}^* = \forall x \forall y (R_{\alpha}(x, y) \rightarrow R_{\alpha}^*(x, y))$
- $\text{trn}_{\alpha}^* = \forall x \forall y \forall z (R_{\alpha}^*(x, y) \wedge R_{\alpha}^*(y, z) \rightarrow R_{\alpha}^*(x, z))$
A nonstandard approach to Hoare's logic

\[ \text{ind}_\alpha(\xi) = \forall x \forall y \ (R_\alpha(x, y) \land \xi(x) \land \\
\land \forall u \forall v \ (R_\alpha(x, u) \land \xi(u) \land R_\alpha(u, v) \rightarrow \xi(v)) \rightarrow \xi(y)) \]

(for each \( \xi \in L^\tau(x) \))

Notice that different continuous interpretations over one and the same \( \mathcal{A} \) may exist; \( \mathcal{A}^{st} \) is always one of them. The idea is that a continuous interpretation allows \( R_\alpha^* \) to be any reflexive and transitive extension of \( R_\alpha \) which obeys the induction axioms \( \text{ind}_\alpha^*(\xi) \). If the set of all states \( y \) such that \( R_\alpha^*(x, y) \) is imagined as the (nondeterministic) “trace” of starting at state \( x \), we are guaranteed that the first-order assertions true at the beginning of a trace and invariant on it must hold all along the trace. The notion of nonstandard time is implicit here, as well as in Csirmaz’s runs [12] and in the continuous traces of Andráka, Németh and Sain [1, 2].

Standard and continuous semantics for partial correctness assertions (pcas) can now be defined.

**Definition 2.6.** The set \( \text{PCA}^\tau(x) \) of pcas for programs in \( \text{RP}^\tau(x) \) consists of all formal expressions \( \pi = \{\phi\}\alpha\{\psi\} \) with \( \phi, \psi \in L^\tau(x) \) (called pre- and postconditions, respectively) and \( \alpha \in \text{RP}^\tau(x) \).

Of course, \( \{\phi\}\alpha\{\psi\} \) is intended to mean: “All computations which start at states satisfying \( \phi \) always halt (if they halt at all) in states satisfying \( \psi \)”. This is not meant to exclude the existence of nonhalting computations.

**Definition 2.7.** Let a specification \( \text{Ax} \subseteq L^\tau \) and a pca \( \pi = \{\phi\}\alpha\{\psi\} \in \text{PCA}^\tau(x) \) be given.

(a) \( \pi \) is true in \( \mathcal{A}^{ct} \) (in symbols, \( \mathcal{A}^{ct} \models \pi \)) iff
\[
\mathcal{A}^{ct} \models \forall x \forall y \ (\phi(x) \land R_\alpha(x, y) \rightarrow \psi(y))
\]
in the sense of first-order logic. This is meant in particular for \( \mathcal{A}^{st} \).

(b) \( \pi \) is a logical consequence of \( \text{Ax} \) with respect to standard semantics \( (\text{Ax} \models^{st} \pi) \) iff \( \mathcal{A}^{st} \models \pi \) for every model \( \mathcal{A} \models \text{Ax} \).

(c) \( \pi \) is a logical consequence of \( \text{Ax} \) with respect to continuous semantics \( (\text{Ax} \models^{ct} \pi) \) iff \( \mathcal{A}^{ct} \models \pi \) for every continuous interpretation \( \mathcal{A}^{ct} \) over an arbitrary model \( \mathcal{A} \models \text{Ax} \).

The continuous consequence \( \models^{ct} \) will be proved equivalent to Hoare-derivability in Section 5. For the rest of this section, we try to clarify the relationship between standard and continuous semantics.

**Lemma 2.8.** Let \( \alpha \in \text{RP}^\tau(x) \) and a continuous interpretation \( \mathcal{A}^{ct} \) over a \( \tau \)-structure \( \mathcal{A} \) be given. Then \( R_\alpha^{st} \supseteq R_\alpha^{ct} \). Furthermore, if \( \alpha \) is a star free program, then \( R_\alpha^{st} = R_\alpha^{ct} \) and there is a formula \( \rho_\alpha \in L^\tau(x, y) \), independent of \( \mathcal{A} \), which defines \( R_\alpha^{ct} \) in \( \mathcal{A} \).
Proof. \( R_\alpha^\mathfrak{st} \supseteq R_\alpha^\mathfrak{nt} \) is established by induction on \( \alpha \). The induction base holds because of the axioms (\( := ? \)) and (\( ? \)) and the induction step follows easily from (\( \cup \)), (\( ; \)), (\( * \)) and (\( \gamma \)). For star free \( \alpha \), the same induction allows \( R_\alpha^\mathfrak{nt} = R_\alpha^\mathfrak{st} \) and \( \rho_\alpha(x, y) \) to be constructed. For instance, axiom (\( ; \)) justifies to set
\[
\rho_{(\alpha; \beta)}(x, y) = \exists z(\rho_\alpha(x, z) \land \rho_\beta(z, y)),
\]
provided that \( \rho_\alpha, \rho_\beta \) have already been constructed. \( \square \)

Definition 2.9. Let \( \mathfrak{A} \) be a \( \tau \)-structure.

(a) \( \mathfrak{A} \) is expressible iff \( \mathfrak{A}^\mathfrak{nt} \) is definable by a formula \( \rho_\alpha \in L^\tau(x, y) \) for every program \( \alpha \in \mathbb{RP}^\tau(x) \); see [25] for other equivalent characterizations.

(b) \( \mathfrak{A} \) is discrete iff each element \( \alpha \in A \) is definable by a formula \( \delta_\alpha \in L^\tau(z) \).

The following theorem summarizes the main relationship between both kinds of semantics.

Theorem 2.10. Let \( \mathfrak{A} \) be expressive and discrete. Then \( \mathfrak{A}^\mathfrak{nt} \) is the only continuous interpretation of \( \mathbb{RP}^\tau(x) \) over \( \mathfrak{A} \). Moreover, both hypotheses are necessary to guarantee the conclusion.

Proof. Let \( \mathfrak{A} \) be expressive and discrete. The standard interpretation \( \mathfrak{A}^\mathfrak{nt} \) is trivially continuous. Given an arbitrary continuous interpretation \( \mathfrak{A}^\mathfrak{st} \), we prove \( R_\alpha^\mathfrak{st} = R_\alpha^\mathfrak{nt} \) by induction on \( \alpha \). The basis step and the induction steps for (\( \cup \)) and (\( ; \)) are handled as in Lemma 2.8. For the case \( \alpha = \beta^* \), the induction hypotheses tell us that \( R^\mathfrak{nt}_\beta = R^\mathfrak{nt}_\beta \), and \( R^\mathfrak{nt}_\beta \supseteq R^\mathfrak{nt}_\beta \) can be assumed by Lemma 2.8. To prove the opposite inclusion, we fix \( \alpha \in S^\mathfrak{nt}_n \) and build the formulae
\[
\delta_\alpha(z) = \bigwedge \{ \delta_\alpha(z_j) \mid 1 \leq j \leq n \}; \quad \xi_\alpha(x) = \exists z(\delta_\alpha(z) \land \rho_{\beta^*}(z, x)),
\]
where \( \rho_{\beta^*} \) exists by expressiveness. Due to \( R^\mathfrak{nt}_\beta = R^\mathfrak{nt}_\beta \), we have
\[
\mathfrak{A}^\mathfrak{st} \models \xi_\alpha(a) \land \forall u \forall v (R_{\beta^*}(a, u) \land \xi_\alpha(u) \land R_{\beta^*}(u, v) \rightarrow \xi_\alpha(v))
\]
and this together with axiom \( \text{ind}_{\beta^*}(\xi_m) \) yields
\[
\mathfrak{A}^\mathfrak{st} \models \forall y (R_{\beta^*}(a, y) \rightarrow \xi_\alpha(y)),
\]
which, holding for every \( a \), means that \( R^\mathfrak{nt}_\beta \subseteq R^\mathfrak{st}_\beta \). This proves the first part of the theorem.

For the second part, let us define two structures in a graphical way (Fig. 6). \( \mathfrak{A} \) consists of a domain of cardinality 4 and a function \( f^\mathfrak{nt} \) which acts as suggested by the arrows. \( \mathfrak{B} \) is formed by the domain \( B = N \cup Z \) (disjoint union), a predecessor function \( f^\mathfrak{nt} \) acting separately on both parts of the domain, and distinguished elements \( c^\mathfrak{nt} = 0^N \), \( d^\mathfrak{nt} = 0^Z \) named by two constants.
As a finite structure, $\mathcal{A}$ is expressive. For $\alpha = x := f(x) \in \text{RP}^T(x)$, the standard interpretation of $\alpha^*$ over $\mathcal{A}$ is

$$R_{\alpha^*}^{\mathcal{A}} = \{(a, a), (a, b), (b, b), (c, c), (c, d), (d, d)\}$$

and a continuous nonstandard interpretation of $\alpha^*$ over $\mathcal{A}$ may be taken as

$$R_{\alpha^*}^{\mathcal{A}} = R_{\alpha^*}^{\mathcal{A}} \cup \{(a, c), (a, d), (b, d), (c, a), (c, b), (d, b)\}.$$ 

In fact, $\mathcal{A}$ is not discrete, and the induction axioms needed to justify the continuity of $\mathcal{A}^\mathbb{N}$ hold because of the nontrivial automorphism of $\mathcal{A}$ which permutes $a$ with $c$ and $b$ with $d$.

On the other side, $\mathcal{B}$ is a discrete structure because of the formulae:

$$\delta_n(z) = f^{(m)}(z) \models c \land \{\neg f^{(m)}(z) \models c \mid m < n\}(n \in \mathbb{N}),$$

$$\delta_i(z) = f^{(i)}(z) \models d \quad (i \in \mathbb{Z}_+),$$

$$\delta_j(z) = f^{(j)}(d) \models z \quad (j \in \mathbb{Z}_-).$$

For the same $\alpha$ as before, taken now as a program of $\text{RP}^T(x)$, the standard interpretation of $\alpha^*$ over $\mathcal{B}$ is

$$R_{\alpha^*}^{\mathcal{B}} = \{(m, n) \mid m \geq n \in \mathbb{N}\} \cup \{(i, j) \mid i \geq j \in \mathbb{Z}\}$$

and there exist a continuous, nonstandard interpretation of $\alpha^*$ over $\mathcal{B}$, namely

$$R_{\alpha^*}^{\mathcal{B}} = R_{\alpha^*}^{\mathcal{B}} \cup \mathbb{Z} \times \mathbb{B}.$$ 

In this interpretation, $\alpha^*$ retains its standard computations and is additionally able to connect points of the $\mathbb{Z}$-part with arbitrary points of $\mathbb{B}$. To justify continuity, let us consider two arbitrary points $i \in \mathbb{Z}, k \in \mathbb{B}$ and an arbitrary formula $\xi(x) \in L^{\mathbb{N}}(x)$.

In order to prove $\mathcal{B}^\mathbb{N} \models \text{ind}_{\alpha^*}(\xi)$ we assume

(1) $\mathcal{B} \models \xi(i)$ and $\mathcal{B} \models \forall u \forall v \left(R_{\alpha^*}(i, u) \land \xi(u) \land R_{\alpha^*}(u, v) \rightarrow \xi(v)\right)$

or equivalently,

(2) $\mathcal{B} \models \forall u \forall v \left(R_{\alpha^*}(i, u) \land \xi(u) \rightarrow \xi(f(u))\right)$

and have to show that

(3) $\mathcal{B} \models \xi(k)$.

But (1) and (2) imply that $\mathcal{B} \models \xi(j)$ for every $j \in \mathbb{Z}, j \leq i$; in particular $\xi(x)$ holds in $\mathcal{B}$ for infinitary many values of $x$. It can be shown that $\mathcal{B}$'s first-order theory
admits elimination of quantifiers (see [31]), and quantifier-free formulae of \( L^{\infty}(x) \) only allow the definition of finite and co-finite subsets of \( B \). Consequently, \( \xi(x) \) must hold for almost every value of \( x \), and this together with (2) does imply (3).

Of course, we can conclude that \( \mathcal{V} \) is not expressive. We shall come to the same conclusion in a different way in Section 5. \( \square \)

**Definition 2.11.** A given \( \tau \)-structure \( \mathcal{V} \) is **locally expressive** for the program \( \alpha(x) \) at \( a \in S^\mathcal{V}_n \) iff \( R^{\mathcal{V}_a}_{\alpha} \upharpoonright b \) is definable in \( \mathcal{V} \) for every subtree \( T^\mathcal{V}_a(b) \) of \( T^\mathcal{V}_a(a) \). By \( R^{\mathcal{V}_a}_{\beta^*} \upharpoonright b \) we mean the set \( \{ c \in S^\mathcal{V}_n | R^{\mathcal{V}_a}_{\beta^*}(b, c) \} \).

**Theorem 2.12.** Let \( \mathcal{V} \) be locally expressive for \( \alpha(x) \) at \( a \in S^\mathcal{V}_n \). Then \( R^{\mathcal{V}_a}_{\alpha} \upharpoonright a = R^{\mathcal{V}_a}_{\beta^*} \upharpoonright a \) for every continuous interpretation \( \mathcal{V}_c \) over \( \mathcal{V} \).

**Proof.** Let us assume the hypothesis. By Lemma 2.8, we know that \( R^{\mathcal{V}_a}_{\alpha} \upharpoonright a \supseteq R^{\mathcal{V}_a}_{\beta^*} \upharpoonright a \). We prove the opposite by induction on \( \alpha \). The induction base, where \( \alpha \) is a nondeterministic assignment or a test, is trivial, as well as the induction step for the case \( \alpha = (\beta \cup \gamma) \).

Let \( \alpha \) be \( (\beta; \gamma) \) and assume \( R^{\mathcal{V}_a}_{\alpha} \upharpoonright (a, b) \) is definable in \( \mathcal{V} \) for every subtree \( T^\mathcal{V}_a(b) \) of \( T^\mathcal{V}_a(a) \). By \( R^{\mathcal{V}_a}_{\beta^*} \upharpoonright b \) we mean the set \( \{ c \in S^\mathcal{V}_n | R^{\mathcal{V}_a}_{\beta^*}(b, c) \} \).

Theorem 2.12 can be seen as a corollary of Theorem 2.10, because an expressive and discrete structure is locally expressive for every program at each state. Sufficient conditions for local expressiveness can still be found in other ways.

**Definition 2.13.** Let \( \mathcal{V} \) be a \( \tau \)-structure and \( \alpha \in \text{RP}^{\infty}(x) \).

(a) \( \mathcal{V} \) is **locally finite** iff the substructure \( [a]^\mathcal{V} \) generated by \( \{a_1, \ldots, a_n\} \) is finite for each state \( a \in S^\mathcal{V}_n \).

(b) \( \alpha \) is **algebraic** over \( \mathcal{V} \) iff for every nondeterministic assignment \( x := ?y. \rho(x, y) \) appearing as a subprogram in \( \alpha \) and for arbitrary states \( a, b \in S^\mathcal{V}_n \) with \( \mathcal{V} \models \rho(a, b) \), it holds that \( \{b_1, \ldots, b_n\} \subseteq [a]^\mathcal{V} \).

(c) \( \alpha \) is of **finite type** over \( \mathcal{V} \) at state \( a \in S^\mathcal{V}_n \) iff \( R^{\mathcal{V}_a}_{\beta^*} \upharpoonright b \) is finite for every subtree \( T^\mathcal{V}_a(b) \) of \( T^\mathcal{V}_a(a) \). Notice that this is guaranteed to hold if \( \alpha \) is a deterministic while-program (written \( \alpha \) as a regular program in the usual way) which converges at \( a \) under the standard interpretation.

**Corollary 2.14.** Let \( \alpha \in \text{RP}^{\infty}(x) \) and a continuous interpretation \( \mathcal{V}_c \) over a \( \tau \)-structure \( \mathcal{V} \) be given. Each of the following conditions implies \( R^{\mathcal{V}_a}_{\alpha} \upharpoonright a = R^{\mathcal{V}_a}_{\alpha^*} \upharpoonright a \), where \( a \in S^\mathcal{V}_n \).
(a) \( \mathcal{A} \) is discrete and \( \alpha \) is of finite type over \( \mathcal{A} \) at \( a \) (which is ensured if \( \alpha \) is a deterministic while-program converging at \( a \) in the standard sense).

(b) \( \mathcal{A} \) is discrete, \( \alpha \) is algebraic over \( \mathcal{A} \) and \( [a]^\mathcal{A} \) is finite (which, of course, happens whenever \( \mathcal{A} \) is locally finite).

**Proof.** It suffices to notice that (a) is implied by (b) and implies the local expressiveness of \( \mathcal{A} \) for \( \alpha \) at \( a \), then apply Theorem 2.12. □

The hypotheses of Corollary 2.14 can hold for an inexpressive structure, as shown by \( \mathcal{R} = (\mathbb{N}, 0^\mathbb{N}, \text{pred}^\mathbb{N}) \) which is obviously discrete and locally finite and can be shown to be inexpressive by means of the techniques from [27]. By Corollary 2.14, we are guaranteed that all continuous interpretations of algebraic programs over \( \mathcal{R} \) must be standard.

To summarize the results of this section, if we are ready to accept the view that all "reasonable data types" should be at least discrete structures, then continuous semantics is not so strange over a reasonable data type. Indeed, it gives their usual meaning to all deterministic, total while-programs, and even to all nondeterministic regular programs if the data type is also expressive. In spite of this, the notion of logical consequence must appeal to arbitrary structures, and hence some kind of nonstandard semantics is needed to derive truly general completeness theorems.

3. A Hoare's calculus for regular programs

The following calculus is intended to derive pcas \( \pi \in \text{PCA}_\mathcal{R}(x) \) from an arbitrary specification \( \text{Ax} \subseteq L^\mathcal{R} \).

**Axioms**

\[
\begin{array}{l}
\text{Assignment} \\
\{ \varphi(x) \} x := ?y. \rho(x, y) \exists z (\varphi(z) \land \rho(z, x)) \\
\hline
\text{Test} \\
\{ \varphi(x) \} \chi(x) ? \{ \varphi(x) \land \chi(x) \}
\end{array}
\]

**Rules**

\[
\begin{array}{l}
\text{Choice} \\
\{ \varphi \} \alpha \{ \psi \} \\
\{ \varphi \} \beta \{ \psi \} \\
\{ \varphi \} (\alpha \cup \beta) \{ \psi \}
\end{array}
\quad
\begin{array}{l}
\text{Composition} \\
\{ \varphi \} \alpha \{ \eta \} \\
\{ \eta \} \beta \{ \psi \} \\
\{ \varphi \} (\alpha ; \beta) \{ \psi \}
\end{array}
\]

\[
\begin{array}{l}
\text{Iteration} \\
\{ \eta \} \alpha \{ \eta \} \\
\{ \eta \} \alpha^* \{ \eta \}
\end{array}
\quad
\begin{array}{l}
\text{Consequence} \\
\phi \rightarrow \phi' \\
\{ \phi' \} \alpha \{ \psi' \} \\
\psi' \rightarrow \psi \\
\{ \phi \} \alpha \{ \psi \}
\end{array}
\]

By a *Hoare derivation* of pca \( \pi \) from a specification \( \text{Ax} \) we mean any finite sequence of pcas and first-order assertions whose last member is \( \pi \) and having the
property that any of its members is either an implication \( \varphi \rightarrow \varphi' \) which is derivable from \( \text{Ax} \) in first-order logic, and eventually used as a premise for the rule of consequence at some later point, or a pca which can be inferred from some previous members of the sequence by means of some of the rules or axioms just stated.

If there is some Hoare derivation of \( \pi \) from \( \text{Ax} \), we write \( \text{Ax} \vdash \pi \) and say that \( \pi \) is Hoare derivable from \( \text{Ax} \).

**Lemma 3.1** (Soundness lemma). Our Hoare's calculus is sound with respect to both standard and continuous semantics. It holds:

\[
\text{Ax} \vdash \pi \Rightarrow \text{Ax} \models \pi \Rightarrow \text{Ax} \models ^{st} \pi.
\]

**Proof.** The second implication holds by the continuity of standard interpretations. The first one can be proved by induction on the program \( \alpha \) of \( \pi \). The only nontrivial case, \( \alpha = \beta * \), uses the fact that a continuous interpretation must satisfy \( \text{ind}_{\beta *}(\eta) \) for every \( \eta \in L^*(x) \).

**Lemma 3.2.** (Completeness theorem for star free programs). Given \( \varphi, \psi \in L^*(x) \), \( \text{Ax} \subseteq L^* \) and a star free \( \alpha \in \text{RP}^*(x) \), the following statements are equivalent:

(a) \( \text{Ax} \vdash \{ \varphi \} \alpha \{ \psi \} \),
(b) \( \text{Ax} \models ^{ct} \{ \varphi \} \alpha \{ \psi \} \),
(c) \( \text{Ax} \models ^{st} \{ \varphi \} \alpha \{ \psi \} \),
(d) \( \text{Ax} \vdash \forall x \forall y (\varphi(x) \land \rho_{\alpha}(x, y) \rightarrow \psi(y)) \),

where \( \rho_{\alpha} \) is the formula from Lemma 2.8.

**Proof.** \((\ast) \Rightarrow (b) \Rightarrow (c)\) is guaranteed by the soundness lemma. \((c) \Rightarrow (d)\) holds because \( \rho_{\alpha} \) defines \( R_{\alpha}^{\text{st}} \) in \( \mathfrak{A} \) and first-order logic is complete. \((d) \Rightarrow (a)\) can be proved by induction on \( \alpha \). We treat here only the case \( \alpha = (\beta; \gamma) \). The hypothesis \((d)\) and the form of \( p_{\beta; \gamma} \) mean that:

\[
\text{Ax} \vdash \forall x \forall z \forall y (\varphi(x) \land \rho_{\beta}(x, z) \land \rho_{\gamma}(z, y) \rightarrow \psi(y)).
\]

Taking \( \eta(x) = \exists u (\varphi(u) \land \rho_{\beta}(u, x)) \in L^*(x) \), it follows that:

\[
\text{Ax} \vdash \forall x \forall y (\varphi(x) \land \rho_{\beta}(x, y) \rightarrow \eta(y)),
\]

\[
\text{Ax} \vdash \forall x \forall y (\eta(x) \land \rho_{\gamma}(x, y) \rightarrow \psi(y)).
\]

By induction hypotheses, we can conclude that

\[
\text{Ax} \vdash \{ \varphi \} \beta \{ \eta \} \quad \text{and} \quad \text{Ax} \vdash \{ \eta \} \gamma \{ \psi \}
\]

and hence \( \text{Ax} \vdash \{ \varphi \} \alpha \{ \psi \} \), because of the composition rule.

**Theorem 3.3** (Cook's completeness theorem). Our Hoare's calculus is complete in the sense of Cook; that is, for any \( \mathfrak{A} \) expressive \( \tau \)-structure and any \( \pi = \{ \varphi \} \alpha \{ \psi \} \in \text{PCA}^*(x) \),

\[
\mathfrak{A}^{\text{st}} \vdash \pi \leftrightarrow \text{Th}(\mathfrak{A}) \vdash \pi.
\]
Proof. \((\Leftarrow)\) follows from the soundness lemma.

\((\Rightarrow)\) We reason by induction on \(\alpha\), assuming a formula \(\rho_\alpha \in L^*(x, y)\) which defines \(R^\alpha\) in \(\mathcal{H}\) for every \(\alpha \in \mathbb{P}'(x)\). The case \(\alpha = (\beta; \gamma)\) is handled in Theorem 3.2. For the case \(\alpha = \beta^*\), the formula \(\eta(x) = \exists u (\varphi(u) \land \rho_{\beta^*}(u, x))\) can be used as invariant. The other cases are trivial. \(\square\)

Hoare's calculus is known to be incomplete with respect to standard semantics (cf. [22, 27]). Nonstandard completeness results are known for unstructured programming languages (cf. e.g. [26]). An especially clear and elegant result was obtained by Csirmaz [12] (the proof has been subsequently simplified by Sain [29]). Csirmaz's result holds for quite abstract programs which can be viewed as an unstructured loop of the form

\[
\text{while } \neg x = f(x) \text{ do } x := f(x) \text{ od},
\]

where \(f\) stands for a first-order definable deterministic state transformer. Well-known normal form results guarantee that this includes, up to equivalence, all deterministic while-programs. But two equivalent while-programs can behave differently with respect to derivability in Hoare's logic, as shown in [5]. To understand in what sense Csirmaz's result applies to arbitrary while-programs, a more detailed examination of the proof theoretical properties of normal forms is needed.

In Section 5 we shall generalize Csirmaz's result by establishing a completeness theorem of Hoare's calculus with respect to continuous semantics. According to the comments above, we shall use a proof theoretical analysis of normal forms for programs. We now state some proof theoretical properties of our Hoare's calculus. Most of them have been obtained already by Bergstra and Tucker [6] for deterministic while-programs and translate without any difficulties to our formalism. Consequently, we only give an outline of the proofs.

We start by stating a technical definition which is needed for some of the lemmas.

**Definition 3.4.** Let \(\alpha = \alpha(x)\) and \(x_j (1 \leq j \leq n)\) be given:

(a) \(\alpha\) does not affect \(x_j\) iff \(\forall x \forall y (\rho \rightarrow y_j = x_j)\) for any subprogram \(x := ?y. \rho(x, y)\) of \(\alpha\).

(b) \(\alpha\) does not use \(x_j\) iff \(\forall x \forall y (\rho \rightarrow \exists v \varphi v[u/x_j, v/y_j])\) for any subprogram \(x := ?y. \rho(x, y)\) of \(\alpha\) and \(\forall x (\chi \rightarrow \exists v \xi v[u/x_j])\) for any subprogram \(\chi(x)\) of \(\alpha\).

(c) \(\alpha\) neither affects nor uses any variable \(z \in \{x_1, \ldots, x_n\}\).

**Lemma 3.5 (Proof decomposition lemma).** For any \(Ax \leq L^*\), any \(\varphi, \psi \in L^*(x)\) and programs in \(\mathbb{P}'(x)\) one has:

(a) Assignment

\[
Ax \vdash \{\varphi(x)\} x := ?y. \rho(x, y) \{\psi(x)\} \iff
Ax \vdash \forall x \forall y (\varphi(x) \land \rho(x, y) \rightarrow \psi(y)),
\]
(b) **Test**

\[
\text{Ax} \vdash \{\varphi(x)\} \chi(x) ? \{\psi(x)\} \quad \text{iff}
\]

\[
\text{Ax} \vdash \forall x (\varphi(x) \land \chi(x) \rightarrow \psi(x));
\]

(c) **Choice**

\[
\text{Ax} \vdash \{\varphi\}(\alpha \cup \beta)\{\psi\} \quad \text{iff}
\]

\[
\text{Ax} \vdash \{\varphi\}\alpha\{\psi\} \quad \text{and} \quad \text{Ax} \vdash \{\varphi\}\beta\{\psi\};
\]

(d) **Composition**

\[
\text{Ax} \vdash \{\varphi\}(\alpha; \beta)\{\psi\}
\]

iff for some intermediate assertion \( \eta \in L^r(x) \),

\[
\text{Ax} \vdash \{\varphi\}\alpha\{\eta\} \quad \text{and} \quad \text{Ax} \vdash \{\eta\}\beta\{\psi\};
\]

(e) **Iteration**

\[
\text{Ax} \vdash \{\varphi\}\alpha^*\{\psi\}
\]

iff for some invariant assertion \( \eta \in L^r(x) \),

\[
\text{Ax} \vdash \forall x (\varphi(x) \rightarrow \eta(x)), \quad \text{Ax} \vdash \{\eta\}\alpha\{\eta\}
\]

and

\[
\text{Ax} \vdash \forall x (\eta(x) \rightarrow \psi(x)).
\]

**Proof.** This follows easily from the form of the axioms and rules of the calculus. □

**Lemma 3.6 (Disjunctions and conjunctions lemma).** Let \( \text{Ax} \subseteq L^r \), \( \alpha \in \text{RP}^r(x) \) and \( \varphi, \psi_i \in L^r(x) \) be such that

\[
\text{Ax} \vdash \{\varphi_i\}\alpha\{\psi_i\} \quad \text{for} \quad 1 \leq i \leq k.
\]

Then

\[
\text{Ax} \vdash \{\lor \{\varphi_i| 1 \leq i \leq k\}\} \alpha\{\lor \{\psi_i| 1 \leq i \leq k\}\}
\]

and

\[
\text{Ax} \vdash \{\land \{\varphi_i| 1 \leq i \leq k\}\} \alpha\{\land \{\psi_i| 1 \leq i \leq k\}\}.
\]

**Proof.** This follows by easy induction on \( \alpha \). □

**Lemma 3.7 (Preservation lemma).** Let \( \alpha \in \text{RP}^r(x) \), \( \varphi \in L^r(x) \) and assume that \( \alpha \) does not affect any variable occurring free in \( \varphi \). Then

\[
\vdash \{\varphi\}\alpha\{\varphi\} \quad \text{(from the empty specification)}.
\]

**Proof.** Proof is by induction on \( \alpha \).

The fact that \( \alpha \) does not affect the free variables of \( \varphi \) is used in the case \( \alpha = x := y. \rho(x, y) \). The other cases are straightforward. □
Lemma 3.8 (Particularization lemma). Let $\alpha \in \mathbf{RP}(x)$, $\varphi, \psi \in L^*(x)$ and assume that $z$ is not used by $\alpha$ and does not occur free in $\psi$. Then, for any $\text{Ax} \subseteq L^*$,

$$\text{Ax} \vdash \{\varphi\} \alpha \{\psi\} \Rightarrow \text{Ax} \vdash \exists z \varphi \alpha \{\psi\}.$$ 

Proof. We assume $\text{Ax} \vdash \{\varphi\} \alpha \{\psi\}$ and reason by induction on $\alpha$. If $\alpha = x := ?y. \rho(x, y)$, Lemma 3.5 yields

$$\text{Ax} \vdash \forall x \forall y (\varphi(x) \land \rho(x, y) \rightarrow \psi(y)).$$

As $\alpha$ does not use $z$, we can infer

$$\text{Ax} \vdash \forall x \forall y (\exists z \varphi(x) \land \rho(x, y) \rightarrow \exists z \psi(y)),$$

and as $z$ does not appear free in $\psi$, also,

$$\text{Ax} \vdash \forall x \forall y (\exists z \varphi(x) \land \rho(x, y) \rightarrow \neg \psi(y)),$$

which implies $\text{Ax} \vdash \{\exists z \varphi\} \alpha \{\psi\}$ by Lemma 3.5. The case $\alpha = \chi(x)$? is handled similarly, and the other cases are easy. \(\square\)

Lemma 3.9 (Deduction lemma). Let $\alpha \in \mathbf{RP}(x)$, $\varphi, \psi \in L^*(x)$ and a sentence $\sigma \in L^*$ be given. Then, for any $\text{Ax} \subseteq L^*$,

$$\text{Ax} \cup \{\sigma\} \vdash \{\varphi\} \alpha \{\psi\} \Leftrightarrow \text{Ax} \vdash \{\varphi \land \sigma\} \alpha \{\psi\}.$$ 

Proof. The proof is by induction on $\alpha$. \(\square\)

Lemma 3.10 (Guard’s lemma). Given $\alpha \in \mathbf{RP}(x)$ and $\chi, \varphi, \psi \in L^*(x)$, it holds for any $\text{Ax} \subseteq L^*$ that

$$\text{Ax} \vdash \{\varphi\}(\chi?; \alpha) \{\psi\} \Leftrightarrow \text{Ax} \vdash \{\varphi \land \chi\} \alpha \{\psi\}.$$ 

Proof. Assume $\text{Ax} \vdash \{\varphi\}(\chi?; \alpha) \{\psi\}$. Applying Lemma 3.5 twice, it follows that

$$\text{Ax} \vdash \varphi \land \chi \rightarrow \eta, \quad \text{Ax} \vdash \{\eta\} \alpha \{\psi\}$$

for certain $\eta \in L^*(x)$. By the consequence rule, we have then

$$\text{Ax} \vdash \{\varphi \land \chi\} \alpha \{\psi\},$$

proving the left-to-right implication. The other follows by the test axiom and the composition rule. \(\square\)

Lemma 3.11 (Simplification lemma). Assume $\alpha \in \mathbf{RP}(x)$, $\text{Ax} \subseteq L^*$, $\varphi, \psi \in L^*(x)$ and $t \in T^*(x)$. Then

(a) If $z$ does not occur free in $\psi$ and $\alpha$ does not use $z$,

$$\text{Ax} \vdash \{\varphi\}(z := t; \alpha) \{\psi\} \Rightarrow \text{Ax} \vdash \{\varphi\} \alpha \{\psi\},$$
(b) If \( z \) does not occur free in \( \psi \),
\[ \text{Ax} \vdash \{ \varphi \}(\alpha; z := t)(\psi) \Rightarrow \text{Ax} \vdash \{ \varphi \} \alpha(\psi), \]

(c) If \( z \) has no free occurrences in \( \varphi, \psi \) or \( t \) and \( \alpha \) does not use \( z \),
\[ \text{Ax} \vdash \{ \varphi \}(z = t; \alpha)(\psi) \Rightarrow \text{Ax} \vdash \{ \varphi \} \alpha(\psi). \]

**Proof.** To prove (a), assume \( \text{Ax} \vdash \{ \varphi \}(x := t; \alpha)(\psi) \). By Lemma 3.5, there is some \( \eta \in L^\tau(x) \) such that
\[ \text{Ax} \vdash \forall x (\varphi \rightarrow \eta(t/z)) \text{ and } \text{Ax} \vdash \{ \eta \} \alpha(\psi). \]
Applying first-order reasoning and Lemma 3.8, we can infer that
\[ \text{Ax} \vdash \forall x (\varphi \rightarrow \exists z \eta) \text{ and } \text{Ax} \vdash \exists z \eta \alpha(\psi), \]
and hence \( \text{Ax} \vdash \{ \varphi \} \alpha(\psi) \). The proofs for (b) and (c) are similar. \( \Box \)

4. A normal form theorem

In this section we prepare the completeness result by showing that every nondeterministic regular program is semantically and proof theoretically equivalent to a normalized program which uses the iteration operator only once. The idea behind the normal form is essentially the same as in the Böhm-Jacopini's theorem [9], but Proposition 4.2 shows that the equivalence between programs and their normal forms must be handled with care in the present context.

**Proposition 4.2.** There exist specifications \( \text{Ax}_1 \subseteq L^\tau \) and very simple deterministic while-programs \( \alpha_i, \hat{\alpha}_i \in \text{RP}^\tau(x) \) \( (i = 1, 2) \) such that

(a) \( \text{Ax}_1 \vdash \text{st} \alpha_1 = x \hat{\alpha}_1 \) but not \( \text{Ax}_1 \vdash \alpha_1 = x \hat{\alpha}_1 \),

(b) \( \text{Ax}_2 \vdash \text{st} \alpha_2 = x \hat{\alpha}_2 \) but not \( \text{Ax}_2 \vdash \text{st} \alpha_2 = x \hat{\alpha}_2 \).
Proof. See Bergstra and Klop [5]. Examples 7.2 (slightly modified) and 7.3. □

Theorem 4.3 (Normal form theorem). Assume that $T^*$ includes two syntactically different variable free terms 0, 1. For every $\alpha \in \text{RP}^*(x)$ it is possible to construct $m \in \mathbb{N}$ and $\hat{\alpha} \in \text{RP}^*(x, u)$ (with $u = u_1, \ldots, u_m$) such that

(a) $\hat{\alpha}$ is of the form $\alpha_1; \alpha_2^*; \alpha_3$ where $\alpha_1, \alpha_2, \alpha_3$ are star free.

(b) For every $Ax \subseteq L^*$ such that $Ax \vdash \neg 0 = 1$, one has

$$Ax \vdash \alpha = x \hat{\alpha} \quad \text{and} \quad Ax \vdash \alpha = x \hat{\alpha}.$$

Proof. Let $\varepsilon$ be the trivial program $x_1 := x_1$, which has no effect. Define $\hat{\alpha} = \alpha_1; \alpha_2^*; \alpha_3$ recursively on the structure of $\alpha$. If $\alpha$ is a nondeterministic assignment or a test, we put $\hat{\alpha} = \alpha; \varepsilon^*; \varepsilon$. If $\hat{\alpha} = \alpha_1; \alpha_2^*; \alpha_3$ and $\hat{\beta} = \beta_1; \beta_2^*; \beta_3$ have been already constructed, we take new variables $u, v$ not appearing in $\hat{\alpha}$ or $\hat{\beta}$ and put

$$\alpha \cup \beta)^\wedge = ((u := 0; \alpha_1) \cup (u := 1; \beta_1)), ((u := 0; \alpha_2) \cup (u := 1; \beta_2))^\wedge;$$

$$(u := 0; \alpha_3) \cup (u := 1; \beta_3),$$

$$(\alpha; \beta)^\wedge = u := 0; v := 0; \alpha_1; (((u := 0 \land v := 0); (\alpha_2 \cup (\alpha_3; v := 0))) \cup$$

$$(u := 0 \land v := 1) ?; (\beta_1; u := 1)) \cup ((u := 1 \land v := 1) ?; (\beta_2))^\wedge;$$

$$(u := 1 \land v := 1) ?; \beta_3,$$

$$(\alpha)^\wedge = u := 0; ((u := 0; (\alpha_1; u := 1)) \cup (u := 1; (\alpha_2 \cup (\alpha_3; u := 0))))^\wedge; u := 0.$$

The idea is to combine guards and boolean variables to control the flow of the computation. Part (a), as well as the semantical equivalence in (b), are easy to check by structural induction on $\alpha$. In particular, notice that if $\hat{\alpha}$ is assumed to satisfy the induction hypotheses, the construction of $(\alpha^*)^\wedge$ guarantees that there are no nested occurrences of the iteration operator, since $\alpha_1, \alpha_2$ and $\alpha_3$ will be star free. For the proof theoretical equivalence, we use induction on $\alpha$ and apply the proof theoretical Lemmas 3.5–3.11 to translate intermediate and invariant assertions from $\alpha$ to $\hat{\alpha}$ and vice versa. Going into all the details would be quite long, but let us sketch the composition case.

Given arbitrary $\varphi, \psi \in L^*(x)$ and $Ax \subseteq L^*$ with $Ax \vdash \neg 0 = 1$ we must prove

$$Ax \vdash \{\varphi\}(\alpha; \beta; \psi) \iff Ax \vdash \{\varphi\}(\alpha; \beta)^\wedge; \psi.$$
To establish the right-hand side, and in view of the form of \((\alpha; \beta)^{\gamma}\), we construct the formula
\[
\sigma = (u \rightleftharpoons 0 \land v \rightleftharpoons 0 \land \eta) \lor (u \rightleftharpoons 0 \land v \rightleftharpoons 1 \land \xi) \lor (u \rightleftharpoons 1 \land v \rightleftharpoons 1 \land \chi)
\]
and prove that

(7) \(AxA(\varphi)u := 0; v := 0; \alpha_1(\sigma)\),

(8) \(AxA(\sigma)((u \rightleftharpoons 0 \land v \leftleftharpoons 0)^{?}; (\alpha_2 \cup (\alpha_3; v := 1))) \lor
\quad ((u \rightleftharpoons 0 \land v \leftleftharpoons 1)^{?}; (\beta_1; u := 1)) \lor ((u \rightleftharpoons 1 \land v \leftleftharpoons 1)^{?}; \beta_2)(\sigma)\),

(9) \(AxA(\sigma)(u \rightleftharpoons 1 \land v \leftleftharpoons 1)^{?}; \beta_3(\psi)\).

In fact, Lemmas 3.5-3.11 allow (7) to be proved from (1), (8) from (2)-(5) and (9) from (6).

\((\Leftarrow):\) Assuming the right-hand side, the form of \((\alpha; \beta)^{\gamma}\) and the proof decomposition lemma allow us to also assume (7)-(9) for a certain formula \(\sigma\). We construct the new formulae
\[
\eta = \sigma[0/u, 0/v], \quad \xi = \sigma[0/u, 1/v], \quad \chi = \sigma[1/u, 1/v]
\]
and prove that (1)-(6) hold for them. This can be done with the help of Lemmas 3.5-3.11, using the fact that
\[

\vdash (u \rightleftharpoons 0 \land v \leftleftharpoons 0 \lor \sigma \rightarrow \eta)^{\forall}, \quad \vdash (\eta \rightarrow \exists u \exists v (u \rightleftharpoons 0 \land v \leftleftharpoons 0 \land \sigma))^{\forall}
\]
and analogously for \(\xi\) and \(\chi\). \(\Box\)

5. A nonstandard completeness theorem

Let us say that a program \(\alpha \in \text{RP}^{*}(x)\) has no nested loops iff for every subprogram \(\beta^*\) of \(\alpha, \beta\) is star free. Notice that programs in the normal form of Theorem 4.3 have no nested loops.

The main result of the present section is the following.

**Theorem 5.1.** Given \(AxA \subseteq L^{\forall}, \varphi, \psi \in L^{\forall}(x)\) and a program \(\alpha \in \text{RP}^{*}(x)\) without nested loops,

\[
AxA \{\varphi\}^\alpha(\psi) \iff AxA \{\varphi\}^\alpha(\psi).
\]

This, together with the normal form Theorem 4.3, yields the following.

**Theorem 5.2** (Nonstandard completeness theorem). Let \(AxA \subseteq L^{\forall}\) be such that \(AxA \vdash \neg 0 \rightleftharpoons 1\). Hoare's calculus is complete with respect to \(AxA\) end continuous semantics in the sense that

\[
AxA \{\varphi\}^\alpha(\psi) \iff AxA \{\varphi\}^\alpha(\psi)
\]

for every \(\alpha \in \text{RP}^{*}(x), \varphi, \psi \in L^{\forall}(x),\) where \(\hat{\alpha}\) is the normal form as in Theorem 4.3.
Proof. The proof is immediate from Theorems 4.3 and 5.1.

The rest of the section is almost completely devoted to the proof of Theorem 5.1, which proceeds by structural induction on $\alpha$. We present the skeleton of the proof first, leaving the heavy work in hands of two auxiliary lemmata.

Proof of Theorem 5.1 (skeleton). The implication from left to right holds by the soundness Lemma 3.1.

For the opposite direction, we reason by induction on $\alpha$.

Induction base. Let $\alpha$ be star free. Then the result holds by the completeness theorem for star free programs, Lemma 3.2.

Induction step. Let $\alpha$ have loops, but without nesting. One of the three following cases must apply:

(a) Choice: $\alpha$ is the form $(\beta \cup \gamma)$. Then $Ax \vdash^c \{\varphi\} \beta \{\psi\}$ and $Ax \vdash^c \{\varphi\} \gamma \{\psi\}$, by the semantics of $(\cup)$, and $Ax \vdash \{\varphi\} \alpha \{\psi\}$ follows by the induction hypotheses and the choice rule.

(b) Composition: follows from the composition Lemma 5.3.

(c) Iteration: see the iteration Lemma 5.4.

Lemma 5.3 (Composition lemma). Assume that the Hoare's calculus is ct-complete for $\alpha_1 \in \mathbb{RP}^i(x) (i = 1, 2)$ in the sense that

$$Ax \vdash^c \{\varphi\} \alpha_i \{\psi\} \Rightarrow Ax \vdash \{\varphi\} \alpha_i \{\psi\}$$

holds for arbitrary $Ax \subseteq L^i, \varphi, \psi \in L^i(x)$. Then Hoare's calculus is also ct-complete for $(\alpha_1; \alpha_2)$.

(Notice that no assumption about stars is made here).

Proof. Assume that the hypotheses of the lemma hold for $\alpha_1, \alpha_2$ and consider arbitrary $Ax, \varphi, \psi$ such that $Ax \vdash^c \{\varphi\} \alpha_1; \alpha_2 \{\psi\}$. If we remember Definition 2.7 and the set $CT_n \subseteq L^i$ from Definition 2.5, this amounts to saying that the following consequence holds in first-order logic.

(1) $Ax \cup CT_n \models \forall x \forall y \forall z (\varphi(x) \land R_{\alpha_1}(x, z) \land R_{\alpha_2}(z, y) \rightarrow \psi(y))$

Let $\tau_i^n (i = 1, 2)$ be two copies of the signature $\tau_n$, obtained by replacing each relation symbol $R_\beta (\beta \in \mathbb{RP}^i(x))$ by $R^i_\beta$, where $R^i_\beta, R^j_\gamma$ are different symbols whenever $i \neq j$. Let $CT_n (i = 1, 2)$ be the result of replacing all occurrences of $R_\beta (\beta \in \mathbb{RP}^i(x))$ in $CT_n$ by $R^i_\beta$. From (1) we are able to infer the following.

(2) $Ax \cup CT_n^1 \cup CT_n^2 \models \forall x \forall y \forall z (\varphi(x) \land R^i_{\alpha_1}(x, z) \land R^j_{\alpha_2}(z, y) \rightarrow \psi(y))$.

Let us postpone the proof of this fact for a moment. If we introduce new constants $a, b, c$ to avoid the universal quantifiers and apply the compactness of first-order
logic, we obtain finite conjunctions $\sigma^i$ of sentences from $\mathrm{CT}_n$ ($i = 1, 2$) such that

$$\vdash (\sigma^1 \land \varphi(a) \land R^{1}_{\alpha}(a, c)) \rightarrow (\sigma^2 \land R^{2}_{\alpha}(c, b) \rightarrow \psi(b)).$$

As the two members of the implication share symbols from the signature $\tau \cup \{c\}$ only, we may apply Craig's interpolation theorem (cf. [31]) and obtain some $\eta \in L^{\tau}(x)$ such that

$$\vdash \sigma^1 \land \varphi(a) \land R^{1}_{\alpha}(a, c) \rightarrow \eta(c),$$

$$\vdash \eta(c) \land \sigma^2 \land R^{2}_{\alpha}(c, b) \rightarrow \psi(b).$$

Returning to the universal quantifiers and dropping superscripts again, we clearly obtain

$$\forall x \vdash \sigma^1 \alpha_1(x) \land \forall x \vdash \sigma^2 \alpha_2(x).$$

Hence, $\forall x \vdash \{\varphi\} \alpha_1 \land \forall x \vdash \{\psi\} \alpha_2$ follows by the hypotheses of the lemma and the composition rule.

We still have to give a justification for (2) on the basis of (1). Clearly, it suffices to show that any model $\mathcal{M}'$ of $\forall x \vdash \mathrm{CT}_n \land \mathrm{CT}_n$ gives rise to a model $\mathcal{M}$ of $\forall x \vdash \mathrm{CT}_n$ over the same domain and such that

$$\mathcal{M} \models \tau = \mathcal{M}' \models \tau \quad \text{and} \quad R^{\mathcal{M}}_{\beta} \supseteq R^{\mathcal{M}'}_{\beta} \quad \text{for } i = 1, 2 \text{ and any } \beta \in \mathrm{RP}^\tau(x).$$

We show this by giving an explicit construction for $\mathcal{M}$. Notice that $\mathcal{M}'$ will be a $(\tau^{1}_n \cup \tau^{2}_n)$-structure, while $\mathcal{M}$ must be a $\tau_n$-structure. Instead of being completely formal, we shall write $R^{\beta}_{\mathcal{M}}$ for $(R^{\beta}_{\mathcal{M}})^{\mathcal{M}'}_i$ ($i \in \{1, 2\}$; $\beta \in \mathrm{RP}^\tau(x)$) and $R^{\beta}_{\mathcal{M}}$ for $(R^{\beta}_{\mathcal{M}})^{\mathcal{M}'}_i$ ($\beta \in \mathrm{RP}^\tau(x)$) in what follows. The relations $R^{1}_{\mathcal{M}}$ are given by $\mathcal{M}'$, and $R^{1}_{\mathcal{M}}$ must be defined. We do this by induction on $\beta$. For star free $\beta$, we take as $R^{1}_{\mathcal{M}}$ the relation given by the meaning of the $\tau$-formula $\rho_{\beta}$ in $\mathcal{M}'$. According to Lemma 2.8, this is the only possibility. For programs $\beta$ of the form $(\beta_1 \cup \beta_2)$ or $(\beta_1 ; \beta_2)$ we construct $R^{1}_{\mathcal{M}}$ in such a way that the axioms $(\cup), (\cdot)$ from Definition 2.5 are satisfied. Finally, for any $\beta \in \mathrm{RP}^\tau(x)$, we define $R^{1*}_{\mathcal{M}}$ by the condition

$$R^{1*}_{\mathcal{M}}(x, y) : \iff \text{there are } k \in \mathbb{N} \text{ and states } z_0, \ldots, z_k \text{ over } \mathcal{M}' \text{ such that:}$$

(i) $z_0 = x$ and $z_k = y$ and

(ii) for any $0 \leq j < k$: either $R^{1}_{\beta}(z_j, z_{j+1})$ or $R^{1}_{\beta}(z_j, z_{j+1})$ or $R^{2}_{\beta}(z_j, z_{j+1})$

where $x, y, z_j$ are informally assumed to range over the domain of $\mathcal{M}'$.

The condition $R^{1}_{\mathcal{M}} \supseteq R^{\mathcal{M}'}_{\beta}$ ($i = 1, 2$) is easy to check by induction on $\beta$. The construction of $\mathcal{M}$ clearly guarantees that all axioms of the form $(\vdash ?), (\vdash \cdot)$ and $(\vdash \cdot)$ are true in $\mathcal{M}$. It remains only to show that $\mathcal{M}$ satisfies all axioms $(\forall \cdot)^{\mathcal{M}}$ for any $\beta \in \mathrm{RP}^\tau(x)$. $R^{1*}_{\mathcal{M}}$ is by construction a reflexive and transitive relation extending $R^{\mathcal{M}}_{\beta}$. We prove that any induction axiom $\text{ind}_{\beta}^*(\xi), \xi \in L^{\tau}(x)$, is satisfied. For this purpose we consider arbitrarily states $x, y$ over $\mathcal{M}$ such that

(3) $\xi(x)$

(4) $\forall u \forall v \ (R^{1*}_{\mathcal{M}}(x, u) \land \xi(u) \land R^{\mathcal{M}}_{\beta}(u, v) \rightarrow \xi(v))$

(5) $R^{\mathcal{M}}_{\beta}(x, y)$
and try to prove $\xi(y)$. By (5), (i) and (ii) above will hold for some number $k$ and intermediate states $z_j$. We show $\xi(z_j)$ ($0 \leq j \leq k$) by induction on $j$. The base $j=0$ is given by (3). Assume $j < k$ and $\xi(z_j)$. Trivially, we can assert

(6) $R_{\alpha^*}(x, z_j)$.

Moreover, according to (ii) above we either have $R_{\alpha}(z_j, z_{j+1})$ or $R_{\alpha^*}(z_j, z_{j+1})$ (with $i = 1$ or $i = 2$). In the first case, $\xi(z_{j+1})$ follows from $\xi(z_j)$, (4) and (6). In the second case, we reason as follows: (4), (6) and the transitivity of $R_{\alpha^*}$ imply

(7) $\forall u \forall v (R_{\alpha^*}(z_j, u) \land \xi(u) \land R_{\alpha}(u, v) \rightarrow \xi(v))$,

which, together with the inclusions $R_{\alpha^*} \supseteq R_{\alpha}^i$, yields:

(8) $\forall u \forall v (R_{\alpha^*}(z_j, u) \land \xi(u) \land R_{\alpha}^i(u, v) \rightarrow \xi(v))$.

As $\forall \alpha \models CT_n$, we know that $\text{ind}_{\alpha^*}(\xi)$ holds for $R_{\alpha^*}^i$, $R_{\alpha}^i$, and (8) together with the induction hypothesis $\xi(z_j)$ implies that

(9) $\forall y (R_{\alpha^*}(z_j, y) \rightarrow \xi(y))$,

which in particular means $\xi(z_{j+1})$ because we are assuming the case $R_{\alpha^*}(z_j, z_{j+1})$. 

Lemma 5.4 (Iteration lemma). Let $\alpha \in \mathcal{R}^\alpha(x)$ be star free. Then Hoare’s calculus is ct-complete for $\alpha^*$ in the sense that

$\forall x \in L^\alpha \models \alpha^*{\{0\}}$ \iff $\forall x \in L^\alpha \models \alpha^*{\{\psi\}}$

holds for arbitrary $Ax \subseteq L^\alpha$, $\varphi, \psi \in L^\alpha(x)$.

Proof. Instead of adapting Csirmaz’s method, as done in our previous work [21], we are going to use the much easier technique of Sain [29] which fits with only minor adaptations to our setting.

Let us substitute the first-order formula $\rho_\alpha$ for $R_\alpha$ in the axioms (*)_ct for $\alpha^*$ and call $E \cup \text{IND}$ to the resulting set, where $E$ includes the three axioms corresponding to $\text{refl}_{\alpha^*}, \text{ext}_{\alpha^*}$ and $\text{trn}_{\alpha^*}$ and IND corresponds to the induction axioms $\text{ind}_{\alpha^*}(\xi)$. By abuse of language, we shall retain the same names $\text{refl}_{\alpha^*}, \text{ext}_{\alpha^*}, \text{trn}_{\alpha^*}, \text{ind}_{\alpha^*}(\xi)$ for the members of $E \cup \text{IND}$. The hypothesis $Ax \models \forall \alpha \models \{\varphi\} \models \alpha^*{\{\psi\}}$ implies

$Ax \cup E \cup \text{IND} \models \forall x \forall y (\alpha(x) \land R_{\alpha^*}(x, y) \rightarrow \psi(y))$

because every model of $Ax \cup E \cup \text{IND}$ can be expanded to a model of $Ax \cup CT_n$ (a formal proof of this would be similar to the construction of $\text{M}$ in Lemma 5.3, and even easier). By the completeness of first-order logic, there is a finite set $\Phi = \{\varphi_0, \ldots, \varphi_m\} \subseteq L^\alpha(x)$ such that

(1) $Ax \cup E \cup \{\text{ind}_{\alpha^*}(\varphi_i) \mid 0 \leq i \leq m\} \models \forall x \forall y (\alpha(x) \land R_{\alpha^*}(x, y) \rightarrow \psi(y))$.

Let $\Phi^+ = \{\bigwedge \Phi_0 \mid \Phi_0 \subseteq \Phi\}$ be the finite set of all finite conjunctions of formulae from
\(\Phi\), chosen without repetitions in some canonical way. Notice that the empty conjunction \textit{true} belongs to \(\Phi^+\). For each \(\chi \in \Phi^+\), we take

\[\text{inv}_\chi(x) = \chi(x) \land \forall u \forall v (\chi(u) \land \rho_\alpha(u, v) \rightarrow \chi(v))\]

and using these formulae we build

\[\theta(x, y) = \bigwedge \{(\text{inv}_\chi(x) \rightarrow \chi(y))| \chi \in \Phi^+\}\].

Intuitively, \(\theta(x, y)\) asserts that all invariant properties described by formulae in the set \(\Phi^+\) and satisfied by state \(x\) are also true at \(y\), which can be imagined as the assertion that \(R_{\alpha^*}(x, y)\) holds under a certain “continuous semantics”. Formally, we obtain that

\begin{align*}
(2) \quad & \forall x \theta(x, x), \\
(3) \quad & \forall x \forall y (\rho_\alpha(x, y) \rightarrow \theta(x, y)), \\
(4) \quad & \forall x \forall y \forall z (\theta(x, y) \land \theta(y, z) \rightarrow \theta(x, z)), \\
(5) \quad & \forall x \forall y \forall z (\theta(x, y) \land \rho_\alpha(y, z) \rightarrow \theta(x, z)), \\
(6) \quad & \forall x \forall y (\varphi(x) \land \theta(x, y) \rightarrow \psi(y)).
\end{align*}

In fact, (2)-(4) are easy consequences of the form of \(\theta\) and (5) follows from (3) and (4). To prove (6), let us accept for the moment that

\begin{align*}
(7) \quad & \text{for every } 0 \leq i \leq m, \\
& \forall x \forall y (\theta(x, y) \land \eta_i(x) \\
& \land \forall u \forall v (\theta(x, u) \land \varphi_i(u) \land \rho_\alpha(u, v) \rightarrow \varphi_i(v)) \rightarrow \varphi_i(y)).
\end{align*}

Then (2), (3), (4) and (7) mean that \(\theta(x, y)\) can be accepted as the definition of a continuous semantics for \(\alpha^*\), as far as the induction axioms for \(\varphi_0, \ldots, \varphi_m\) are concerned. More formally, (2), (3), (4) and (7) do imply that

\[\text{Ax} \cup \{\forall x \forall y (R_{\alpha^*}(x, y) \leftrightarrow \theta(x, y))\} \vdash \text{E} \cup \{\text{ind}_{\alpha^*}(\varphi_i)| 0 \leq i \leq m\},\]

which, together with (1), implies (6). Once (2)-(6) have been established, \(\text{Ax} \vdash \{\varphi\alpha^*\psi\}\) can be proved by using either of the two following invariants:

\[\eta_1(x) = \exists z (\varphi(z) \land \theta(z, x)) \quad \text{or} \quad \eta_2(x) = \forall z (\theta(x, z) \rightarrow \psi(z))\]

which remind us of the idea of \textit{strongest postcondition} and \textit{weakest precondition}, respectively. Indeed, (2)-(6) imply quite directly the following

\[\vdash \forall x (\varphi(x) \rightarrow \eta_1(x)) \quad \text{Ax} \vdash \forall x (\varphi(x) \rightarrow \eta_2(x))\]

\[\vdash \{\eta_1\} \alpha \{\eta_1\} \quad \vdash \{\eta_2\} \alpha \{\eta_2\}\]

\[\text{Ax} \vdash \forall x (\eta_1(x) \rightarrow \psi(x)) \quad \vdash \forall x (\eta_2(x) \rightarrow \psi(x))\]

(use Theorem 3.2 to obtain the second line and apply (6) to justify the two derivations using Ax) which proves \(\text{Ax} \vdash \{\varphi\alpha^*\psi\}\) and incidentally shows that the appeal to
A nonstandard approach to Hoare's logic

299

the specification during the partial correctness proof can be restricted in either of
two extreme ways.

It remains only to justify (7). For this purpose, assume an arbitrary \( \tau \)-structure
\( \mathcal{A} \) and fix two arbitrary states \( a, b \in S^n_\tau \) such that

\[
(8) \quad \mathcal{A} \models \varphi_i(a),
\]

\[
(9) \quad \mathcal{A} \models \forall u \forall v (\theta(a, u) \land \varphi_i(u) \land \rho_\alpha(u, v) \rightarrow \varphi_i(v)),
\]

\[
(10) \quad \mathcal{A} \models \theta(a, b).
\]

We must prove that \( \mathcal{A} \models \varphi_i(b) \). Combining (2) with (8) and (5) with (9) we obtain

\[
(11) \quad \mathcal{A} \models \theta(a, a) \land \varphi_i(a)
\]

\[
(12) \quad \mathcal{A} \models \forall u \forall v (\theta(a, u) \land \varphi_i(u) \land \rho_\alpha(u, v) \rightarrow \theta(a, v) \land \varphi_i(v))
\]

By the form of \( \theta \), \( \theta(a, y) \) is equivalent in \( \mathcal{A} \) to the conjunction of all formulae
\( \chi(y), \chi \in \Phi^+ \), such that \( \mathcal{A} \models \text{inv}_{\chi}(a) \). As \( \Phi^+ \) is (up to logical equivalence) closed
under conjunctions, we can choose a \( \chi \in \Phi^+ \) (which depends on \( a \)) in such a way that

\[
(13) \quad \mathcal{A} \models \forall y (\theta(a, y) \leftrightarrow \chi(y)).
\]

Now, \( \chi \land \varphi_i \in \Phi^+ \) (up to equivalence) and (11), (12), (13) mean that \( \mathcal{A} \models \text{inv}_{\chi \land \varphi_i}(a) \).
This together with (10) yields \( \mathcal{A} \models \chi \land \varphi_i(b) \), by the definition of \( \theta \), and in particular
\( \mathcal{A} \models \varphi_i(b) \) is established. We invite the reader to compare this construction with the
use of evasive formulae appearing in [21]. \( \square \)

Examples

As an illustration of the completeness theorem, remember structure \( \mathcal{B} \) and program
\( \alpha \) from Theorem 2.10. The continuous interpretation we built there and Theorem
5.1 show that \( \text{Th}(\mathcal{B}) \not\models \{x \equiv d\} \alpha^* \{\neg x \equiv c\} \), although this pca is true in \( \mathcal{B} \) under
standard semantics. Because of Cook's completeness Theorem 3.3, \( \mathcal{B} \) must be
inexpressive.

Here, the nondefinability of a pca from \( \text{Th}(\mathcal{B}) \) has been witnessed by a continuous
interpretation over \( \mathcal{B} \). In general, it may be necessary to go over to a nonstandard
model of the first-order theory of the given structure. To illustrate this, let us return
to structure \( \mathcal{N} \) from Section 2 and build

\[
\alpha = \text{while } \neg(x \equiv 0 \lor \text{pred}(x) \equiv 0) \text{ do } x := \text{pred}(\text{pred}(x)) \text{ od};
\]

\[
\text{while } \neg(y \equiv 0 \lor \text{pred}(y) \equiv 0) \text{ do } y := \text{pred}(\text{pred}(y)) \text{ od}.
\]

Although \( \mathcal{N} \models \{x = y\} \alpha^* \{x = y\} \), we claim that \( \text{Th}(\mathcal{N}) \not\models \{x \equiv d\} \alpha \{x \equiv c\} \). The intuitive reason is that \( \mathcal{N} \)'s
first-order language is too weak to speak about parity. Formally, we can apply Theorem 5.1. Now, Corollary 2.14(b) implies that the only continuous
interpretation of \( \alpha \) over \( \mathcal{N} \) is the standard one. But we can use the nonstandard
model \( \mathcal{N}^* \) shown in Fig. 7, where \( \mathbb{N}^* = \mathbb{N} \cup \mathbb{Z} \) (disjoint union). It can be shown that

\[
\mathcal{N}^* = (\mathbb{N}^*, 0^*, \text{pred})
\]

Fig. 7.
\( \mathcal{N}^* \models \text{Th}(\mathcal{N}) \), furthermore, there is a continuous interpretation \( \mathcal{N}^*_{\text{ct}} \) such that \( R_{\alpha}^{\mathcal{N}^*_{\text{ct}}} (a, a, 0, 1) \) for every \( a \in \mathbb{Z} \). We omit details.

6. Conclusions and related work

We have presented continuous semantics for regular programs as a natural generalization of standard semantics and have characterized Hoare's derivability of a pca \( \{ \varphi \} \alpha \{ \psi \} \) in terms of the behaviour of the normal form \( \hat{\alpha} \) in continuous interpretations. We could ask ourselves to what extent this is a satisfactory characterization. Although \( \alpha \)'s syntactic structure is very strongly reflected by \( \hat{\alpha} \) (there is even an algorithm to recover \( \alpha \) from \( \hat{\alpha} \)) it would be nice to have a completeness theorem which refers to the behaviour of \( \alpha \) directly. This question remains open and will be the subject of further research.

Of course, the methods and results of this paper belong to the nonstandard trend in program logic and are closely related to work done by Andrēka, Németi and Sain (cf. especially [2, 3, 28, 29]) and Csirmaz [12], as already discussed in the introduction. Following another approach, Bergstra and Tucker [7] have obtained a completeness theorem for Hoare's logic for deterministic while-programs. Their axiomatic semantics shares some of the properties we have established in Section 2 with continuous semantics, but there are important differences. For instance, the axiomatic meaning of the simple assignment \( x := \text{suc}(x) \) over the structure \( \langle \mathbb{Z}, \text{suc} \rangle \) is \( \mathbb{Z} \times \mathbb{Z} \), and although we have not checked it formally, we feel that the axiomatic meaning of \( (\alpha; \beta) \) can be strictly greater than the composition of the meanings. More generally, the axiomatic interpretation of a program seems to be greater than any continuous interpretation. Being closer to the general framework of nonstandard dynamic logic, our approach might be better suited for generalization to other programming languages and verification problems beyond partial correctness. Nevertheless, the relationship between axiomatic and continuous semantics seems to deserve closer investigation. As axiomatic semantics yields a completeness theorem for Hoare's partial correctness calculus, the comparison might help to answer the open question at the beginning of this section.

Acknowledgment

We thank Hajnal Andrēka, Istvan Németi and Ildiko Sain for encouraging our work and giving us many facilities to know their results. We are also grateful to two anonymous referees, whose constructive criticism helped us to improve a previous version of the paper. Last but not least, we thank Francisca Lucio-Carrasco, who held many fruitful discussions with us at the time we were starting to work on the subject.
References


