Semantical Analysis of Specification Logic, 3
An Operational Approach

Dan R. Ghica*

Oxford University Computing Laboratory, Oxford, U.K. OX1 3QD
drg@comlab.ox.ac.uk

Abstract. We are presenting a semantic analysis of Reynolds's specification logic of Idealized Algol using the parametric operational techniques developed by Pitts. We hope that this more elementary account will make the insights of Tennent and O'Hearn, originally formulated in a functor-category denotational semantics, more accessible to a wider audience. The operational model makes clearer the special nature of term equivalence in the logical setting, identifies some problems in the previous interpretation of negation and also proves the soundness of two new axioms of specification logic. Using the model we show that even a very restricted fragment of specification logic is undecidable.

1 Introduction

The specification logic of Idealized Algol (IA) [1,2] is perhaps the best attempt to extend Hoare-style programming logic to a language with procedures. It is quite general and its usability has been clearly demonstrated. The semantics initially proposed by Reynolds is classical, with models of specifications consisting of environments of meanings for its free identifiers (universal specifications). Subsequently, Reynolds realized that making specifications stateful, i.e. including both environment and state in the model, would lead to more powerful and more useful programming axioms.

The soundness of specification logic with stateful specifications has been the object of important semantic research. Tennent [3] had the first insight that this logic is intuitionistic and requires a Kripke-style possible worlds model, with sets of states as worlds. The principal semantic innovation of his approach was the restriction of the behaviour of IA commands so that no states outside of the current world are visited in the course of the command's execution. The way this restriction is imposed mathematically in the denotational setting is quite striking, based on the functor-categorical construction initially developed by Reynolds and Oles as a denotational model for IA [4,5]. This mathematical construction is remarkable because the restriction it imposes is prima facie incompatible with denotational semantics, which is extensional rather than intensional; in addition, the semantic model of the specification logic connects seamlessly with that of the programming language. This semantic model was

* The author was partly funded by Canadian NSERC and UK EPSRC.

D.A. Schmidt (Ed.): ESOP 2004, LNCS 2986, pp. 264–278, 2004.

further refined by Tennent and O'Hearn, allowing the validation of several new axioms [6].

The semantic ideas of the research mentioned before carried over into further work on the semantics of IA, leading to several improved models. Much of this work is collected in [7]. One trait this line of research shares is the technical reliance on the functor-categorical construction of Reynolds and Oles. The presentation therefore demands a rather high level of mathematical sophistication from the reader. In addition, the denotational model of IA it leads to is not fully abstract, which is, to some extent, unsatisfactory. In an operational model, this issue would not arise.

A more elementary account of the semantic properties of IA, profoundly inspired by its denotational model, is given by Pitts, using parametric operational techniques [8]. The mathematical apparatus of this model is elementary, yet the model is powerful enough to prove many non-trivial properties of IA. In this paper we further develop Pitts's model to deal with specification logic. This should make the interesting ideas used to prove the soundness of specification logic more accessible in the same way as Pitts's work made accessible the semantic properties of the programming language. To illustrate the usefulness of the model we prove two additional new axioms of specification logic. One of the axioms has been proposed before by O'Hearn [9], but without a soundness proof. The other axiom is interesting because it relates Hoare triples and equivalence.

Although IA is not a programming language used in practice, it has been an influential prototypical language. In fact it is not far from Haskell with state monadic extensions [10]. A better understanding of IA's specification logic should help with the design of a similar logic for Haskell.

Finally, using the operational model of specification logic we show that even the very restricted fragment of specification logic consisting of first-order recursion-free boolean Algol with first-order quantifiers and without term equivalence is not decidable. This is despite the fact that observational equivalence of IA terms of first [11] and second [12] order IA is decidable.

Contributions. We present a new model for the specification logic of IA, based on the operational semantics of the language. Compared to the previous models, based on the denotational semantics of IA, ours is mathematically elementary and it does not suffer from a lack of full-abstraction. Two new axioms are added to the logic, and the special nature of equivalence and negation is better explained. We conclude by giving an undecidability result about a restricted fragment of specification logic.

2 Operational Semantics of IA

In this section we briefly present Pitts's parametric operational model of IA [8].

IA is simply-typed lambda calculus over booleans, integers, variables, expressions and commands. Variables, storing integers, form the state while commands can change the state. In addition to abstraction ($\lambda x : \sigma . M$) and application (FA), other terms of the language are conditionals, uniformly applied to any

$$w \vdash s; R \Downarrow_\sigma s; R, \quad \text{where } R ::= \mathit{tt} \mid \mathit{ff} \mid v \mid n \mid \mathbf{skip} \mid \lambda x:\sigma.M$$

$$\frac{w \vdash s; B \Downarrow_{\mathbf{bool}} s'; b \qquad w \vdash s'; M_b \Downarrow_\sigma s''; R}{w \vdash s; (\mathbf{if}\ B\ \mathbf{then}\ M_{\mathit{tt}}\ \mathbf{else}\ M_{\mathit{ff}}) \Downarrow_\sigma s''; R}$$

$$\frac{w \vdash s; N_1 \Downarrow_{\mathbf{int}} s'; n_1 \qquad w \vdash s'; N_2 \Downarrow_{\mathbf{int}} s''; n_2}{w \vdash s; (N_1 * N_2) \Downarrow_{\mathbf{int}} s''; n}\ n = n_1 * n_2$$

$$\frac{w \vdash s; F \Downarrow_{\sigma\to\sigma'} s'; \lambda x:\sigma.M \qquad w \vdash s'; M[A/x] \Downarrow_{\sigma'} s''; R}{w \vdash s; FA \Downarrow_{\sigma'} s''; R}$$

$$\frac{w \vdash s; M[\mathbf{fix}\, x:\sigma.M/x] \Downarrow_\sigma s'; R}{w \vdash s; \mathbf{fix}\, x:\sigma.M \Downarrow_\sigma s'; R} \qquad \frac{w \vdash s; V \Downarrow_{\mathbf{var}} s'; v}{w \vdash s; !V \Downarrow_{\mathbf{int}} s'; n}\ s'(v) = n$$

$$\frac{w \vdash s; V \Downarrow_{\mathbf{var}} s'; v \qquad w \vdash s'; N \Downarrow_{\mathbf{int}} s''; n}{w \vdash s; V := N \Downarrow_{\mathbf{comm}} (s'' \mid v \mapsto n); \mathbf{skip}}$$

$$\frac{w \vdash s; C \Downarrow_{\mathbf{comm}} s'; \mathbf{skip} \qquad w \vdash s'; C' \Downarrow_{\mathbf{comm}} s''; \mathbf{skip}}{w \vdash s; (C; C') \Downarrow_{\mathbf{comm}} s''; \mathbf{skip}}$$

$$\frac{w \vdash s; N \Downarrow_{\mathbf{int}} s'; n \qquad wv \vdash s' \otimes (v \mapsto n); C[v/x] \Downarrow_{\mathbf{comm}} s'' \otimes (v \mapsto n'); \mathbf{skip}}{w \vdash s; (\mathbf{new}\, x := N\, \mathbf{in}\, C) \Downarrow_{\mathbf{comm}} s''; \mathbf{skip}}$$

Fig. 1. IA evaluation relation

type, (**if** B **then** M **else** N), fix-point recursion (**fix** $x:\sigma.M$), constants (integers, booleans) and arithmetic-logic operators ($M * N$); we also have command-type terms which are the standard imperative operators: dereferencing (explicit in the syntax, $!V$), assignment ($V := N$), sequencing ($C; C'$), no-op (**skip**) and local variable block (**new** $x := N$ **in** C).

Let Γ be a function from identifiers to types. We write $\Gamma \vdash M : \sigma$ to indicate that term M has type σ and (typed) free identifiers in Γ.

The terms for abstraction, fix-point and local-variable block are identifier binders; IA terms are identified up to α-conversion. For any term M we denote the set of free identifiers and their type assignment $\mathit{fi}(M)$. By $IA_\sigma(w) = \{M : \sigma \mid \mathit{fi}(M) = \emptyset \text{ and } \mathit{gv}(M) \subseteq w\}$, we denote the set of closed IA terms of type σ with global variables in the set w.

The operational semantics of IA is defined inductively on the syntax of terms by an evaluation relation of the form $w \vdash s; M \Downarrow_\sigma s'; R.$. We call the set of global variables w *the world*, with terms $M, R \in IA_\sigma(w)$ and states $s, s' \in \mathit{States}(w) = \{f \mid f : w \to \mathbb{N}\}$. The interpretation of the rules is the following: term M executed in state s evaluates to term R (called a *canonical form*) and changes the state to s'. The evaluation relation is defined in Fig. 1.

The following notations are used: wv denotes the set w with a new element v added; $s \otimes (v \mapsto n) \in \mathit{States}(wv)$ denotes the state properly extending s by mapping v to n; $(s \mid v \mapsto n)$ denotes the state mapping v to n and otherwise acting like s. This notation is generalized to $s \otimes s' \in \mathit{States}(ww')$, $w \cap w' = \emptyset$, in the obvious way.

The evaluation rules are more general than really necessary. It is straightforward to show that the evaluation of non-command terms does not change the state. Conversely, command terms can only evaluate to **skip**. Therefore, the

evaluation relation can be abbreviated to $w \vdash s; M \Downarrow_\sigma R$ if $\sigma \neq \mathbf{comm}$ and to $w \vdash s; C \Downarrow s'$, if $C : \mathbf{comm}$. If $w \vdash s; M \Downarrow_\sigma s'; R$ does not hold for any R and s' we write $w \vdash s; M \Uparrow_\sigma$.

A context $\mathcal{C}[-_\sigma]$ is an IA term in which a sub-term of type σ is replaced by a "hole" $-_\sigma$. The term resulting from filling the hole with term $M : \sigma$ is denoted by $\mathcal{C}[M]$. We write $traps(\mathcal{C}[-_\sigma])$ for the set of bound identifiers the scope of which includes the hole of $\mathcal{C}[-_\sigma]$, and their type assignments.

Terms or contexts without free identifiers are called *closed*. A key distinction is being made between *global variables* and *free identifiers of type* **var**. A closed term may have zero, one or more global variables. This distinction is also logically important because *contraction* of free identifier preserves equivalence whereas that of global variables does not.

Definition 1 (Contextual equivalence). *If $M_i : \sigma$, with $fi(M_i) \subseteq \Gamma$ and $gv(M_i) \subseteq w$ for $i = 1, 2$, we write $w, \Gamma \vdash M_1 \cong_\sigma M_2$ to indicated that the terms are* contextually equivalent*. By definition, this means that for all worlds $w' \supseteq w$, all closed contexts $\mathcal{C}[-_\sigma] : \mathbf{comm}$ with $gv\big(\mathcal{C}[-_\sigma]\big) \subseteq w'$ and $\Gamma \subseteq traps\big(\mathcal{C}[-_\sigma]\big)$, and all states $s, s' \in States(w')$ we have $w' \vdash \mathcal{C}[M_1]; s \Downarrow s'$ iff $w' \vdash \mathcal{C}[M_2]; s \Downarrow s'$.*

Two terms are contextually equivalent if their occurrences can be interchanged in any closed command without affecting the meaning of the command, as a partial function from states to states.

The quantification over contexts in the definition of contextual equivalence makes the definition unwieldy for the proof of program properties. A more usable form of equivalence is *extensional equivalence*.

Definition 2 (Extensional equivalence). *If $M_i \in IA_\sigma(w)$, $i = 1, 2$, we write $w \vdash M_1 \equiv_\sigma M_2$ to indicate that the terms are* extensionally equivalent*, which is defined inductively on the structure of σ as follows:*

1. *If $\sigma = \mathbf{bool}, \mathbf{int}$ then $w \vdash M_1 \equiv_\sigma M_2$ if $\forall s \in States(w)$ and all constants c, $w \vdash M_1; s \Downarrow_\sigma c$ iff $w \vdash M_2; s \Downarrow_\sigma c$;*
2. *$w \vdash M_1 \equiv_{\mathbf{comm}} M_2$ if $\forall s, s' \in States(w)$, $w \vdash M_1; s \Downarrow s'$ iff $w \vdash M_2; s \Downarrow s'$;*
3. *$w \vdash M_1 \equiv_{\mathbf{var}} M_2$ if $w \vdash M_1 := n \equiv_{\mathbf{comm}} M_2 := n$ for all $n \in \mathbb{N}$;*
4. *$w \vdash M_1 \equiv_{\sigma_1 \to \sigma_2} M_2$ if $\forall w' \supseteq w$ and $A \in IA_{\sigma_1}(w')$, $w' \vdash M_1 A \equiv_{\sigma_2} M_2 A$.*

Extensional equivalence is extended to open terms via closed instantiations.

For all $M_i : \sigma_i$ with $fi(M_i) \subseteq \Gamma = \{x_1 : \sigma_1, \ldots, x_n : \sigma_n\}$, $gv(M_i) \subseteq w$, $i = 1, 2$, we define $w, \Gamma \vdash M_1 \equiv_\sigma M_2$ as $w' \vdash M_1[\vec{A}/\vec{x}] \equiv_\sigma M_2[\vec{A}/\vec{x}]$ for all $w' \supseteq w$ and all $A_j \in IA_{\sigma_j}(w')$, $j = 1, \ldots, n$.

The key result of [8] is the following (OE):

Theorem 1 (Operational Extensionality). *IA contextual equivalence coincides with extensional equivalence: $w, \Gamma \vdash M_1 \cong_\sigma M_2$ iff $w, \Gamma \vdash M_1 \equiv_\sigma M_2$.*

The proof technique used in [8] is perhaps as interesting as the result itself, involving an operational version of parametric logical relations for IA.

A useful ancillary property of IA proved in the same paper, and useful in proving the monotonicity of identity is Weakening and Strengthening (WS).

Lemma 1 (Weakening and Strengthening).
Suppose $w = w_1 w_2$, $s_i \in States(w_i)$, $i = 1, 2$. Given $M \in IA_\sigma(w_1)$, then $w \vdash s_1 \otimes s_2; M \Downarrow_\sigma s'; R$ if and only if $R \in IA_\sigma(w_1)$ and $s' = s'_1 \otimes s_2$ for some $s'_1 \in States(w_1)$ with $w \vdash s_1; M \Downarrow_\sigma s'_1; R$.

2.1 Identity Logic of IA

One important direct consequence of the OE Theorem, pointed out by Pitts, is that it allows one to establish familiar functional extensionality properties such as: $F \cong_{\sigma\to\sigma'} F' \quad \text{iff} \quad \forall A : \sigma.\, FA \cong_{\sigma'} F'A$.

Extensionality is not an universal property of programming languages. In call-by-value languages with more complex computational features such as references or exceptions, such as Scheme or ML, this property fails. IA, which is a conservative extension of PCF [13], inherits extensionality as a generalised version of Milner's Context Lemma [14] for PCF.

We take this point further and we notice that the definitions of extensional equivalence for functions and open terms is strongly suggestive of the interpretation of the universal quantifier in an intuitionistic logic, with a Kripke structure of worlds w and domains $IA_\sigma(w)$, cf. [15, Chap. 5].

Let us define the (intuitionistic) identity logic of IA as its many-sorted predicate logic with identity.

Definition 3 (Identity logic). *The set of formulas Φ of the logic consists of:*

1. ***equality**: if $\Gamma \vdash M_i : \sigma$, $i = 1, 2$, then $\Gamma \vdash M_1 \doteq M_2 \in \Phi$;*
2. ***connectives**: if $\Gamma \vdash \phi_i \in \Phi$, $i = 1, 2$, then $\Gamma \vdash \phi_1 \square \phi_2 \in \Phi$, where $\square \in \{\wedge, \vee, \to\}$;*
3. ***quantifiers**: if $\Gamma, x : \sigma \vdash \phi \in \Phi$, then $\Gamma \vdash \forall x : \sigma.\phi \in \Phi$, $\Gamma \vdash \exists x : \sigma.\phi \in \Phi$.*

We use the symbol $\doteq$ to signify equality *in the logic* (as opposed to in the meta-language or in IA). If no confusion is possible in context we may use just $=$ instead.

A formula with an empty environment is called *closed*.

The identity logic of IA has a standard Kripke-style semantics.

Definition 4 (Kripke model). *For any world w and closed formula ϕ we define $w \Vdash \phi$, read "w forces ϕ," inductively on the structure of ϕ:*

1. *$w \Vdash \phi_1 \vee \phi_2$ if $w \Vdash \phi_1$ or $w \Vdash \phi_2$;*
2. *$w \Vdash \phi_1 \wedge \phi_2$ if $w \Vdash \phi_1$ and $w \Vdash \phi_2$;*
3. *$w \Vdash \phi_1 \to \phi_2$ if for all $w' \supseteq w$, if $w' \Vdash \phi_1$ then $w' \Vdash \phi_2$;*
4. *$w \Vdash \forall x : \sigma.\, \phi$ if for all $w' \supseteq w$ and for all $A \in IA_\sigma(w')$, $w' \Vdash \phi[A/x]$;*
5. *$w \Vdash \exists x : \sigma.\, \phi$ if there is $A \in IA_\sigma(w)$ such that $w \Vdash \phi[A/x]$;*
6. *$w \Vdash M_1 \doteq M_2$ if $w \vdash M_1 \cong_\sigma M_2$.*

Theorem 2 (Soundness). *If ϕ is a formula of the identity logic of IA derivable using the (standard) axioms of intuitionistic logic and identity then $\Vdash \phi$.*

3 Specification Logic

The logic of the previous section can be augmented with two more predicate symbols, Hoare triples and non-interference:

Definition 5 (Specification logic of IA). *The set of formulas Σ (specifications) of the logic consists of:*

1. ***equality**: if $\Gamma \vdash M_i : \sigma$, $i = 1, 2$, then $\Gamma \vdash M_1 \doteq M_2 \in \Sigma$;*
2. ***Hoare triple**: if $\Gamma \vdash P : \mathbf{assert}$, $\Gamma \vdash Q : \mathbf{assert}$ and $\Gamma \vdash C : \mathbf{comm}$, then $\Gamma \vdash \{P\}\ C\ \{Q\} \in \Sigma$;*
3. ***non-interference**: if $\Gamma \vdash P : \sigma$, $\Gamma \vdash Q : \sigma'$ then $\Gamma \vdash P \# Q \in \Sigma$;*
4. ***connectives**: if $\Gamma \vdash \phi_i \in \Sigma$, $i = 1, 2$, then $\Gamma \vdash \phi_1 \,\square\, \phi_2 \in \Sigma$, where $\square \in \{\wedge, \vee, \rightarrow\}$;*
5. ***quantifiers**: if $\Gamma, x : \sigma \vdash \phi \in \Sigma$, then $\Gamma \vdash \forall x : \sigma.\phi \in \Sigma$, $\Gamma \vdash \exists x : \sigma.\phi \in \Sigma$.*

The *pre-condition* P and *post-condition* Q of a Hoare triple belong to a special type, called *assertions*. The language of assertions is the language of boolean phrases augmented with universal and existential quantifiers. To keep the current presentation focused we only consider a reduced assertion language which consists of finitary boolean phrases (constructed without the fix-point operator). The introduction of assertion quantifiers or non-termination raises some technical complications, surveyed in [16]. It is common to use $\{R\}$ for $\{R\}$ **skip** $\{\mathbf{true}\}$.

The axioms of specification logic are for the most part those of Hoare logic, with several new axioms to handle procedures and interfering side-effects [1].

A straightforward interpretation of the new predicate symbols is enough to show the soundness of much of Reynolds's original, universal, specification logic [1]:

1. $w \Vdash \{P\}\ C\ \{Q\}$ if $\forall s, s' \in \mathit{States}(w)$, $w \vdash s; P \Downarrow_{\mathbf{assert}} \mathit{tt}$ and $w \vdash s; C \Downarrow s'$ implies $w \vdash s'; Q \Downarrow_{\mathbf{assert}} \mathit{tt}$
2. $w \Vdash M \# N$ if $\forall s_0, s_1, s_2 \in \mathit{States}(w)$, and canonical forms R, R', if $w \vdash s_0; M \Downarrow_{\sigma'} s_1; R$ and $w \vdash s_0; N \Downarrow_{\sigma} s_2; R'$ then there is a state $s_3 \in \mathit{States}(w)$ such that $w \vdash s_1; N \Downarrow_{\sigma} s_3; R'$

To wit, if N evaluates to R' *before* the execution of M, then *after* its execution it also evaluates to R', although the state might have changed. For example, if i, j are distinct global variables then $i := 0 \,\#\, !j + 2$.

The stateful specification logic requires a more sophisticated approach. The reason we want to make specifications stateful is to "factor out" assumptions about the state in judgements of the form $\{R\} \rightarrow \{P\}\ C\ \{Q\}$, to be interpreted as "specification $\{P\}\ C\ \{Q\}$ is true in all states in which assertion R is true." Reynolds shows several examples where this is useful. The role of the world must be now played by a set of states, which we shall call W.

The new accessibility relation on worlds $\precsim$ reflects the fact that worlds, as sets of states, can change in two ways: they can expand through the introduction of new variables or they must "shrink" because of implication.

$$W \vdash s; R \Downarrow_\sigma s; R, \text{ where } R ::= tt \mid ff \mid v \mid n \mid \mathbf{skip} \mid \lambda x:\sigma.M$$

$$\frac{W \vdash s; B \Downarrow_{\mathbf{bool}} s'; b \qquad W \vdash s'; M_b \Downarrow_\sigma s''; R}{W \vdash s; \mathbf{if}\ B\ \mathbf{then}\ M_{tt}\ \mathbf{else}\ M_{ff} \Downarrow_\sigma s''; R}$$

$$\frac{W \vdash s; N_1 \Downarrow_{\mathbf{int}} s'; n_1 \qquad W \vdash s'; N_2 \Downarrow_{\mathbf{int}} s''; n_2}{W \vdash s; N_1 * N_2 \Downarrow_{\mathbf{int}} s''; n}\ n = n_1 * n_2$$

$$\frac{W \vdash s; F \Downarrow_{\sigma\to\sigma'} s'; \lambda x:\sigma.M \qquad W \vdash s'; M[A/x] \Downarrow_{\sigma'} s''; R}{W \vdash s; FA \Downarrow_{\sigma'} s''; R}$$

$$\frac{W \vdash s; M[\mathbf{fix}\, x:\sigma.M/x] \Downarrow_\sigma s'; R}{W \vdash s; \mathbf{fix}\, x:\sigma.M \Downarrow_\sigma s'; R} \qquad \frac{W \vdash s; V \Downarrow_{\mathbf{var}} s'; v}{W \vdash s; !V \Downarrow_{\mathbf{int}} s'; n}\ s'(v) = n$$

$$\frac{W \vdash s; V \Downarrow_{\mathbf{var}} s'; v \qquad W \vdash s'; N \Downarrow_{\mathbf{int}} s''; n}{W \vdash s; V := N \Downarrow_{\mathbf{comm}} (s'' \mid v \mapsto n); \mathbf{skip}}\ (s'' \mid v \mapsto n) \in W$$

$$\frac{W \vdash s; N \Downarrow_{\mathbf{int}} s'; n \qquad W \times States(v) \vdash s' \otimes (v \mapsto n); C[v/x] \Downarrow_{\mathbf{comm}} s'' \otimes (v \mapsto n'); \mathbf{skip}}{W \vdash s; (\mathbf{new}\, x := N\, \mathbf{in}\, C) \Downarrow_{\mathbf{comm}} s''; \mathbf{skip}}$$

Fig. 2. IA confined evaluation relation

Definition 6 (World accessibility). *We define the relation $W_1 \precsim W_2$, for $W_i \subseteq States(w_i)$, as the transitive and reflexive closure of $W_1 \precsim W_2$ if $W_2 = W_1 \times States(w_2 \setminus w_1)$ and $W_1 \precsim W_2$ if $W_2 \subseteq W_1$.*

It is straightforward to show that $\precsim$ is a partial order on sets of states.

The interpretation of the logical connectives is now given by:

Definition 7 (Stateful Kripke semantics).

1. *$W \Vdash \phi_1 \wedge \phi_2$ if $W \Vdash \phi_1$ and $W \Vdash \phi_2$*
2. *$W \Vdash \phi_1 \vee \phi_2$ if $W \Vdash \phi_1$ or $W \Vdash \phi_2$*
3. *$W \Vdash \phi_1 \to \phi_2$ if for all $W' \succsim W$, $W' \Vdash \phi_1$ implies $W' \Vdash \phi_2$*
4. *$W \Vdash \forall x:\sigma.\phi$ if for all $W' \succsim W$, $W' \in States(ww')$, for all $A \in IA_\sigma(ww')$, $W' \Vdash \phi[A/x]$,*
5. *$W \Vdash \exists x:\sigma.\phi$ if there is $A \in IA_\sigma(\mathrm{dom}\, W)$, such that $W \Vdash \phi[A/x]$.*

The straightforward interpretation of the Hoare triple is no longer satisfactory in this context, because the axiom of command composition is no longer sound. We need to apply Tennent's key idea that execution of commands must be restricted to the current world, not only in terms of its final state but also in terms of all the intermediary states that are visited [3].

We do this by introducing a more refined evaluation relation for IA, which we shall call *confined execution* and denote by $W \vdash s; M \Downarrow_\sigma s'; R$.

This relation is defined just like the normal evaluation (Fig. 1) with the exception of assignment and local variable definition. The relation is defined, inductively on the syntax of IA, in Fig. 2.

The only way a command can stray outside the current world is through assignment, but such an assignment is prevented by confined execution. If an

illegal assignment is attempted the command cannot be executed. The world of a local variable block is expanded by all possible states involving the newly allocated variable.

Confined evaluation can be notationally simplified, similarly to standard evaluation, considering that only commands can change the state. We adapt the old notation in the straightforward way.

The interpretation of the Hoare-triple predicate is:

Definition 8 (Stateful Hoare-triple semantics). *$W \Vdash \{P\}\ C\ \{Q\}$ if and only if for all $s, s' \in W$, $W \vdash s; P \Downarrow_{\mathbf{assert}} tt$ and $W \vdash s; C \Downarrow s'$ implies $W \vdash s'; Q \Downarrow_{\mathbf{assert}} tt$.*

Using confined evaluation we can define stronger notions of contextual and extensional equivalence in the obvious way. Using exactly the same proof method as Pitts we can show that confined equivalence is also extensional.

Theorem 3 (Confined Oper. Extensionality). *IA confined contextual equivalence coincides with confined extensional equivalence: $W, \Gamma \vdash M_1 \cong_\sigma M_2$ iff $W, \Gamma \vdash M_1 \equiv_\sigma M_2$.*

It is easy to show, by induction on derivation of evaluation, that confined evaluation also satisfies a WS Lemma.

Lemma 2 (Conf. Weakening–Strengthening.). *Suppose that $w = w_1 w_2$, $W_i \subseteq States(w_i)$, and $s_i \in W_i$, $i = 1, 2$. Given $M \in IA_\sigma(w_1)$, then it follows that $W_1 \times W_2 \vdash s_1 \otimes s_2; M \Downarrow_\sigma s_1' \otimes s_2; R$ iff $R \in IA_\sigma(w_1)$ and $s_1' \in W_1$ with $W_1 \vdash s_1; M \Downarrow_\sigma s_1'; R$.*

Confined evaluation is a conservative extension of IA evaluation, and confined equivalence is strictly stronger than contextual equivalence, which corresponds to equivalence 'confined' to the set of all states.

We give the following interpretation for identity:

Definition 9 (Hyperfine identity semantics). *$W \Vdash M_1 \doteq M_2$ if and only if for all $W' \gtrsim W$, $W' \vdash M_1 \cong_\sigma M_2$.*

We call this new notion of equivalence *hyperfine* because, as we shall see, it is significantly stricter than observational equivalence.

3.1 Non-interference

Finally, we have come to the interpretation of the non-interference predicate, $M \# N$. As in the previous section, our approach is heavily influenced by concepts first introduced in the denotational setting.

Intuitively, non-interference is interpreted as follows. The term on the right-hand-side, typically an expression, is used to partition the world into smaller worlds in which the term evaluates to the same result. Then, we require any evaluation of the term on the left-hand-side to be confined to any set of equivalence classes. In other words, any execution of M must not change the possible values of N *even temporarily.*

Definition 10 (State equivalence). *We call* state-equivalence *any IA term-indexed family of equivalence relations* $[M]_W \subseteq W^2$, $W \subseteq States(w)$, *with the property that:*

1. *if* $\sigma = \mathbf{int}$ *(or* $\mathbf{bool}$*) then* $s_1 \, [E]_W \, s_2$ *if and only if for all* $n \in \mathbb{N}$ *(or* $\{tt, ff\}$*)* $W \vdash s_1; E \Downarrow_\sigma n$ *iff* $W \vdash s_2; E \Downarrow_\sigma n$*;*
2. *if* $\sigma = \mathbf{var}$ *then* $s_1 \, [V]_W \, s_2$ *if and only if* $s_1 \, [!V]_W \, s_2$*;*
3. *if* $\sigma = \mathbf{comm}$ *then* $s_1 \, [C]_W \, s_2$ *if and only if* $W \vdash s_1; C \Downarrow$ *iff* $W \vdash s_2; C \Downarrow$ *;*
4. *if* $\sigma = \sigma_1 \to \sigma_2$ *then* $s_1 \, [F]_W \, s_2$ *implies for all* w', $A \in IA_{\sigma_1}(ww')$ *and states* $s' \in States(w')$ *such that* $s_1 \otimes s' \, [A]_{W \times States(w')} \, s_2 \otimes s'$ *we have*

$$s_1 \otimes s' \, [FA]_{W \times States(w')} \, s_2 \otimes s'.$$

Proposition 1 (Existence & Admissibility).

1. *The family* $[M]_W \subseteq W^2$ *is non-empty.*
2. *State equivalence is preserved by confined equivalence: for all* $s_1, s_2 \in W$, $s_1 \, [M]_W \, s_2$ *and* $W \vdash M \equiv M'$ *implies* $s_1 \, [M']_W \, s_2$*.*
3. *There exists a state equivalence* $[M]^{\sim}_W \subseteq W^2$*, which we shall call an* admissible *state equivalence, such that whenever* $M_0 \in IA(\emptyset)$, $[M_0]^{\sim}_W = W^2$*.*

Proof. At ground types **exp** and **comm**, M_0 is a constant so the definition of state equivalence gives W^2. For **var** there is no constant in $IA(\emptyset)$ so the definition is vacuous. At function types we can construct an admissible relation $[-]^{\sim}$ from any state equivalence $[-]$ as follows: if $M \in IA(\emptyset)$ then for all $s_1, s_2 \in$, $s_1 \, [M]^{\sim}_W \, s_2$; otherwise $s_1 \, [M]^{\sim}_W \, s_2$ iff $s_1 \, [M]_W \, s_2$. Using the fact that state equivalence is preserved by confined extensional equivalence we can show by induction on the type of M that $[-]^{\sim}$ does satisfy the definition of a state equivalence. □

In the following we will only be concerned with admissible state equivalences. Any such relation has the properties required to show the soundness of the non-interference logic. The definition above provides a sufficient framework for the operational model.

By $W/[N]$ we shall denote the set of equivalence classes of W under some admissible state equivalence $[N]_W$, which Prop. 1 established that exists. By $\mathbb{P}\big(W/[N]\big)$ we shall denote the set of unions of equivalence classes under $[N]_W$.

The interpretation of the non-interference predicate $M \# N$ is also inductive, on the structure of M's type. The definition of higher-order non-interference is subtle. Reynolds initially defined it extensionally, but Tennent has shown that this definition is incompatible with a desirable axiom concerning non-interference decomposition [3]. A model proving the soundness of this axiom was given in [6].

An important technical role is played by the property of *confined termination*, which is a generalisation of the usual notion of termination to confined evaluation.

Definition 11 (Confined termination). $M \in IA(w)$ *is said to* confinedly terminate*, denoted by* $W \vdash M \Downarrow_\sigma$ *if*

1. $\sigma = \mathbf{comm}, \mathbf{int}, \mathbf{bool}$*: for all* $s \in W$*, there exists* $s' \in W$ *and canonical form* R *such that* $W \vdash s; M \Downarrow_\sigma s'; R$
2. $\sigma = \mathbf{var}$*: for all* $n \in \mathbb{N}$*,* $W \vdash M := n \Downarrow_{\mathbf{comm}}$
3. $\sigma = \sigma_1 \to \sigma_2$ *and for all* w' *and* $M' \in IA_{\sigma_1}(ww')$*, if* $W \times States(w') \vdash M' \Downarrow_{\sigma_1}$ *then* $W \times States(w') \vdash MM' \Downarrow_{\sigma_2}$.

We can now define higher-order non-interference.

Definition 12 (Non-interference). *If* $F \in IA_\sigma(w)$*, then* $W \Vdash F \# M$ *if and only if,* $W \vdash F \Downarrow_\sigma$ *implies that for all* $W_0 \in \mathbb{P}\big(W/[M]\big)$*, we have that* $W_0 \vdash F \Downarrow_\sigma$.

Restricting the current world to any smaller world which is a union of equivalence classes $W/[M]$, should not change the confined termination status of F.

The knowing reader might notice that the operational definition of non-interference is not exactly an operational reproduction of the denotational interpretation. Whereas in the denotational model non-interference was interpreted as invariance under the application of an appropriately defined restriction (endo)morphism *at the same world*, in the operational model we allow a change of world but we interpret non-interference as the invariance of the confined termination property. The difference is purely of technical presentation: worlds W and W_0 are over the same set of locations w, and the meaning of F in the two worlds, if defined, is actually the same.

We can see how this definition works by examining the typical example of a non-interference specification which is validated by the Reynolds model but, correctly, invalidated by the Tennent-O'Hearn model:

$$\lambda c : \mathbf{comm}.\, \mathbf{if}\, !i = 1\, \mathbf{then}\, (c; \mathbf{if}\, !i > 1\, \mathbf{then}\, i := 0) \# i.$$

Let us call the λ-term above F. Consider $M \stackrel{\text{def}}{=} \mathbf{if}\, !i > 0\, \mathbf{then}\, i := !i + 1$. If we take $W_0 = \{(i \mapsto n) \mid n > 0\}$, which is indeed a union of equivalences classes of $W/[i]$: $\bigcup_{k>0}\{(i \mapsto k)\}$, it can be easily seen that $W_0 \vdash M \Downarrow_{\mathbf{comm}}$ but it is not the case that $W_0 \vdash FM \Downarrow_{\mathbf{comm}}$, since for $s = (i \mapsto 1)$, FM cannot be confinedly evaluated.

Theorem 4 (Soundness). *The specification logic of IA is sound.*

3.2 Proof of the Soundness Theorem

The key property that our modeling framework must satisfy is monotonicity.

Lemma 3 (Monotonicity of Non-interference). *If* $W \Vdash F \# M$ *then* $W' \Vdash F \# M$*, for all* $W' \gtrsim W$.

For the logical connectives and quantifiers we have a standard Kripke semantics. Using the Confined OE Theorem we can also show that the interpretation of identity satisfies the standard axioms.

For the identity, "hyperfine," semantics is truly required. Contextually equivalent IA phrases are not necessarily equivalent in non-interference contexts. Consider for example the simple case $x \vdash x := 0; x := 1 \equiv_{\mathbf{comm}} x := 1$. Although contextually equivalent, when confined to world $(x \mapsto 1)$ the LHS diverges but the

RHS does not, so they are not logically identical. The logical contexts in the presence of non-interference specification are highly intensional.

The Hoare-like axioms are proved in a standard way, and we have already showed the interesting bit regarding command composition for stateful specifications.

In the following we show the soundness of the most interesting axioms of Specification Logic.

Strong constancy. The proof for the constancy axiom:

$$\forall c : \mathbf{comm}. \forall pqr : \mathbf{assert}.\ c \# r \wedge (\{r\} \to \{p\}\ c\ \{q\}) \to \{p\ \mathbf{and}\ r\}\ c\ \{q\ \mathbf{and}\ r\},$$

is straightforward. The non-interference condition ensures that command c does not satisfy $\{p\}\ c\ \{q\}$ trivially, by (even temporarily) breaking assertion $\{r\}$ and causing a (confined) divergence.

Left-side non-interference decomposition. We give a slightly different version of this axiom, which is equivalent with the traditional one but easier to deal with:

1. if $C \in IA_\sigma(\emptyset)$ then $\forall e : \sigma'.\, C \# e$;
2. $\forall f : \sigma \to \sigma'. \forall c : \sigma. \forall e : \sigma''.\, f \# e \wedge c \# e \to f(c) \# e$.

Closed terms without global variables cannot cause interference. Also, non-interference is compositional to the left.

For the first item, we need to prove that $W \Vdash C \# E$ for all $E \in IA_{\sigma'}(w)$. Using the WS Lemma we can show that for any $W_0 \subseteq W$, $W_0 \vdash C \Downarrow_\sigma$ if and only if $States(\emptyset) \vdash C \Downarrow_\sigma$, because C is a closed term.

The second item reduces, using the semantics of the universal quantifier at any arbitrary world W, to $W \Vdash F \# E$, $C \# E$ implies $W \Vdash FC \# E$, which follows directly from the definition of confined termination at function types.

Right-side non-interference decomposition. Again, we give a different but equivalent formulation for the axioms:

1. if $E \in IA_\sigma(\emptyset)$ then $\forall c : \sigma'.\, c \# E$;
2. $\forall f : \sigma \to \sigma'. \forall e : \sigma. \forall c : \sigma''.\, c \# f \wedge c \# e \to c \# f(e)$.

A closed term without global variables cannot be interfered with. Non-interference is also compositional to the right.

For the first item we can show using induction on the structure of σ and the WS Lemma that for any two states $s, s' \in W$, $s[E]_W s'$, i.e., the state-set is 'partitioned' in just one large partition. This follows from the admissibility property in Prop. 1.

For the second item first notice that $s[E]_W s'$, and $s[F]_W s'$ implies $s[F(E)]_W s'$, from the definition of state equivalence. To wit, the partitions generated by $F(E)$ include the partitions formed by F and E. It can be easily

shown that $W/[F(E)] \in \mathbb{P}(W/[F\&E])$ where we define $s_1[F\&E]s_2$ as $s_1[F]s_2$ and $s_1[E]s_2$; therefore we can apply the hypothesis.

Note the technical importance that in the definition of non-interference the restricted world is a *collection* of equivalence classes rather than an equivalence class, since it is *not* the case in general that $W/[F(E)] \subseteq W/[F] \cap W/[E]$.

Local variable. Let M_i be arbitrary IA terms. The local variable axiom scheme is used to discard any good-variable and non-interference assumptions involving the local variable:

$$\big(\forall x : \mathbf{var}.\forall n : \mathbf{int}.\ \mathbf{good}(x) \wedge \cdots x \# M_i \cdots \wedge \cdots M_j \# x \cdots \\ \rightarrow \{P\}\ x := n; C\ \{Q\}\big) \rightarrow \{P\}\ \mathbf{new}\, x := n\, \mathbf{in}\, C\ \{Q\},$$

with $x \notin \mathit{fi}(E_i, C_j, P, Q)$, where the good variable predicate **good** is defined as:

$$\mathbf{good}(x) \stackrel{\text{def}}{=} \forall p : \mathbf{int} \rightarrow \mathbf{assert}.\forall e : \mathbf{int}.x \# p \rightarrow \{p(e)\}\ x := e\ \{p(!x)\}.$$

We first show that for any global variable v and world W, $W \Vdash \mathbf{good}(v)$. For any P, we can show that $v \notin gv(P)$ implies $W \vdash v \# P$. Then we apply the WS Lemma and the OE Theorem to show that $P(E)$ and $P(!v)$ evaluate the same value.

The second part is to show that local variable v cannot interfere or be interfered with. This also follows directly from the WS Lemma, since v does not occur in any other phrases other than the block of the command, $C[v/x]$.

3.3 Two New Axioms

In this section we show two new axioms of specification logic two be valid; they cannot be validated using the denotational model. The first one involves an existential quantifier and, therefore, one of the directions requires definability in the denotational model. The second involves equivalence and one of the directions would require the denotational model to be fully abstract. Since we work at an operational level, neither definability nor full abstraction are an issue for us.

Hoare-triple decomposition. If the programming language is finitary (only a finite set of integers can be stored in memory locations) then:

$$\forall pq : \mathbf{assert}.\forall c_1 c_2 : \mathbf{comm}.\ \{p\}\ c_1; c_2\ \{q\} \leftrightarrow \exists r : \mathbf{assert}.\ \{p\}\ c_1\ \{r\} \wedge \{r\}\ c_2\ \{q\},$$

It was suggested in [9], as a stronger tool necessary for the proof of IA equivalences.

Proof. (Soundness) The right-to-left direction is trivial and is simply the Hoare axiom for sequential composition.

The left-to-right direction requires us to prove that for all worlds $W \in \mathit{States}(w)$, if $W \Vdash \{P\}\ C_1; C_2\ \{Q\}$ holds, then $W \Vdash \exists r : \mathbf{assert}.\ \{P\}\ C_1\ \{r\} \wedge \{r\}\ C_2\ \{Q\}$ must hold, with $P, Q \in IA_{\mathbf{assert}}(w)$, $C_1, C_2 \in IA_{\mathbf{comm}}(w)$.

If the language is finitary, then W is finite. If $W \Vdash C_1; C_2, s \Downarrow s'$ then there must exist s'' such that $W \Vdash C_1, s \Downarrow s''$ and $W \Vdash C_2, s'' \Downarrow s'$. Let us denote the set of all such s'' by W'', which is also finite.

Let $R = \bigvee_{s \in W''} \bigwedge_{v \in w} !v = s(v)$. It follows that $R \in IA_{\mathbf{assert}}(w)$ and $W \Vdash \{P\}\ C_1; C_2\ \{Q\}$ implies $W \Vdash \{P\}\ C_1\ \{R\} \wedge \{R\}\ C_2\ \{Q\}$. □

Remark 1. Hoare-triple decomposition only holds for the finitary fragment of the language because the assertion language is finitary, i.e. it lacks recursion. If the assertion language is not finitary then we can prove this axiom using a weakest-precondition style argument, but a fix-point constructor is required in the assertion language. However, as we mentioned, the introduction of fix-point (and consequently divergence) into the assertion language complicates the properties of the logic.

Hoare-triple and equivalence

$$\forall c_1, c_2 : \mathbf{comm}.\big(\forall pq : \mathbf{assert}.\ \{p\}\ c\ \{q\} \leftrightarrow \{p\}\ c_2\ \{q\}\big) \leftrightarrow c_1 = c_2.$$

Proof. (Soundness) The right-to-left implication is trivial. The left-to-right is proved by contradiction. Assume that there is world W, and $C_1, C_2 \in IA_{\mathbf{comm}}(w)$ such that $W \nVdash C_1 = C_2$. We consider two cases:

1. There exists $s \in W$, $W \in \mathit{States}(w)$, such that $W \Vdash C_i, s \Downarrow$ and $W \Vdash C_{2-i}, s \Uparrow$, $i = 0, 1$. Then we take $P = \mathbf{true}$ and $Q = \mathbf{false}$.
2. There exists $s, s_1 \neq s_2 \in W$ such that $W \Vdash C_1, s \Downarrow s_1$ and $W \Vdash C_2, s \Downarrow s_2$. Then we take $P = \bigwedge_{v \in w} !v = s(v)$ and $Q = \bigwedge_{v \in w} !v = s_1(v)$. □

3.4 Computability Issues

The identity logic of full IA is undecidable, as the language is Turing-complete. What may come as a surprise is that even if we restrict IA to its boolean, first-order, recursion-free, subset, which is decidable [11], the congruence-free specification logic is still undecidable.

Theorem 5 (Undecidability of Spec. Logic). *The specification logic of recursion-free, first-order boolean IA without term equality is undecidable.*

Proof. We show that there exists a faithful encoding $\ulcorner - \urcorner : \mathbb{N}[X] \to IA$ of polynomials with integer coefficients into IA: $\ulcorner 0 \urcorner = \lambda c : \mathbf{comm}.\mathbf{skip}$, $\ulcorner n + 1 \urcorner = \lambda c : \mathbf{comm}.\ulcorner n \urcorner(c); c$, $= \lambda c : \mathbf{comm}.c^{n+1}$, $\ulcorner X_i \urcorner = x_i : \mathbf{comm} \to \mathbf{comm}$, $\ulcorner P_1 + P_2 \urcorner = \lambda c : \mathbf{comm}.\ulcorner P_1 \urcorner(c); \ulcorner P_2 \urcorner(c)$, $\ulcorner P_1 \times P_2 \urcorner = \lambda c : \mathbf{comm}.\ulcorner P_1 \urcorner\big(\ulcorner P_2 \urcorner(c)\big)$. Hilbert's tenth problem can be formulated in the intuitionistic identity logic of first-order, recursion-free, boolean IA as follows:

$$\exists x_1 : \mathbf{comm} \to \mathbf{comm} \cdots \exists x_n : \mathbf{comm} \to \mathbf{comm}.\ulcorner P_1 \urcorner = \ulcorner P_2 \urcorner, \qquad (1)$$

$P_1, P_2 \in \mathbb{N}[X]$. Hilbert's Tenth Problem, is undecidable (see [17] for a comprehensive survey). Then, using the *Hoare triple and equivalence* axiom, validated earlier, we can formulate (1) using Hoare triples instead of term equality. □

4 Concluding Remarks

The remarkable achievement the Tennent-O'Hearn denotational model of specification logic was to give a compositional semantics for the highly intensional non-interference predicate. In an operational setting this task is substantially easier. The challenge, in the operational settings, is to provide a good semantics for the underlying Kripke structure of the logic. Properties, such as monotonicity, which come "for free" in the denotational setting need more elaborate proofs here. The semantics of quantifiers and the closely related properties of substitution are also more difficult to handle using operational methods.

At the heart of the successful operational treatment of specification logic is Pitts's result that contextual equivalence in IA coincides with extensional equivalence. In a language without this property, such as ML, this treatment cannot be reproduced. Consider for example the work of Honsell, Mason, Talcott and others [18,19], which represents the best attempt to create a specification logic for a call-by-value language with assignment. The logic is not technically a first-order logic; the logical properties of quantifiers, i.e. substitution, are not the expected ones. The logic requires sophisticated syntactic side-conditions in the quantifier axioms to ensure correctness, because contextual equivalence in that language holds only in certain circumstances, detailed in the so-called *ciu* ("closed instantiation of uses") Lemma.

The question of whether the definition of specification logic presented here is in some sense definitive remains open. The essential concept of non-interference has a definition which is deeply inspired by the denotational treatment. But, arguably, our semantics of the identity and existential quantifiers is better, since the issues of definability and full abstraction do not arise. The two new axioms of Sec. 3.3 illustrate this advantage. Perhaps a more profound question to ask is whether non-interference logic is the definitive specification logic for procedural languages.

Finally, another important point of this paper is that it shows a connection between the operational extensionality property of a programming language and the fact that it has an elegant and general specification logic. This confirms Reynolds's language design intuition that *call-by-name* and *local effects* are the quintessential features if a specification and verification-friendly programming language is desired.

Acknowledgements. Andrzej Murawski helped me simplify Section 3.4; Bob Tennent and Peter O'Hearn patiently answered my numerous (and often silly) questions about the functor-category model; Andy Pitts has pointed out several errors in an earlier draft of this work. I thank all of them for the help.

References

1. Reynolds, J.C.: The Craft of Programming. Prentice-Hall Intl., London (1981)
2. Reynolds, J.C.: Idealized Algol and its specification logic. In Néel, D., ed.: Tools and Notions for Program Construction, Nice, France, Cambridge University Press, Cambridge, 1982 (1981) 121–161. Also [7, Chap. 6].
3. Tennent, R.D.: Semantical analysis of specification logic. Information and Computation **85** (1990) 135–162. Also [7, Chap. 13].
4. Reynolds, J.C.: The essence of Algol. In de Bakker, J.W., van Vliet, J.C., eds.: Algorithmic Languages, Proceedings of the International Symposium on Algorithmic Languages, Amsterdam (1981) 345–372. Also [7, Chap. 3].
5. Oles, F.J.: Functor categories and store shapes. [7, Chap. 11] 3–12
6. O'Hearn, P.W., Tennent, R.D.: Semantical analysis of specification logic, 2. Information and Computation **107** (1993) 25–57. Also [7, Chap. 19].
7. O'Hearn, P.W., Tennent, R.D., eds.: Algol-like Languages. Progress in Theoretical Computer Science. Birkhäuser, Boston (1997) Two volumes.
8. Pitts, A.M.: Reasoning about local variables with operationally-based logical relations. In: Proceedings of LICS 11, Washington (1996) 152–163. Also [7, Chap. 17].
9. O'Hearn, P.W.: The Semantics of Non-Interference: A Natural Approach. Ph.D. thesis, Queen's University, Kingston, Canada (1990)
10. Launchbury, J., Peyton Jones, S.: State in Haskell. Lisp and Symbolic Computation **8** (1995) 293–341
11. Ghica, D.R., McCusker, G.: Reasoning about Idealized algol using regular languages. In: ICALP 27. Volume 1853 of LNCS., Springer-Verlag (2000) 103–116
12. Ong, C.H.L.: Observational equivalence of third-order Idealized Algol is decidable. In: Proceedings of LICS 17, Copenhagen (2002) 22–25
13. O'Hearn, P.W.: Note on Algol and conservatively extending functional programming. J. of Functional Programming **6** (1995) 171–180. Also [7, Chap. 4].
14. Milner, R.: Fully abstract models of typed λ-calculi. Theoretical Computer Science **4** (1977) 1–22
15. van Dalen, D.: Logic and Structure. Third edn. Springer, Berlin (1994)
16. Bencivenga, E.: Free logics. In Gabbay, D., Guenthner, F., eds.: Handbook of Philosophical Logic, vol. III: Alternatives in Classical Logic. Number 166 in Synthese Library. D. Reidel, Dordrecht, Holland (1986) 373–426
17. Matiyasevich, Y.V.: Hilbert's tenth Problem. Nauka Publishers, Fizmalit (1993) English translation: MIT Press, Cambridge, MA, 1993.
18. Mason, I.A., Talcott, C.L.: References, local variables, and operational reasoning. In: Proceedings of LICS 7, Santa Cruz, California, (1992) 186–197
19. Honsell, F., Mason, I., Smith, S., Talcott, C.: A variable typed logic of effects. Information and Computation **119** (1995) 55–90