An information like this gives us a better picture of what is going on and makes the possibilities of contacts between various researchers and institutions more feasible. In this way our feeling of forming a community of people interested in the development of (theoretical) computer science should be strengthened.

Thank you very much for your cooperation.

G. Rozenberg (Leiden)

P.S. Please notice that (on p.163) we have a list of papers to be presented at ICALP 82 in July in Aarhus. It looks like we are going to have (again!) a very interesting meeting. See you there!

TECHNICAL CONTRIBUTIONS

A Propositional version of Hoare's Logic with modal operators

Jan Bergstra, Department of Mathematics, University of Leiden, July 1981.

Abstract. We present an extension of the propositional Hoare Logic in [1] to a system in which the assertion language contains model operators corresponding to universal and existential quantification over the domain. This enables us to model the important deduction property in a propositional setting.

Introduction

In [1] we developed a sound and complete proof system PHL for a propositional version of Hoare's Logic. In section 1 we will repeat this system, without the completeness proof which is in fact rather straightforward and bears

technical similarity with the methods used in KOZEN & PARIKH [5]. In section 2 we will explain an extended propositional system which includes operators $\Box$ and $\Diamond$ having the following intended meaning:

$\Box A \equiv$ for all states σ, A holds;

$\Diamond A \equiv$ for some state σ, A holds.

Thus obtaining an extended assertion language we have to augment PHL with various rules to obtain MPHL, (modal-PHL) which is sound and complete; the completeness proof is presented in section 3.

It may be useful to spend some words on the motivation of this kind of work. Our primary interest is to work in the direction of a proof theory for Hoare's Logic. This proof theory should work for while programs, of course, as well as for program languages involving recursion, nested procedure declarations, function procedures, goto's, concurrent systems with diverse synchronization mechanisms and systems involving abstract datatypes.

We were able to work ([2],[3]) on while programs and their proof theory by removing one obstacle: Cook's notion of relative completeness (see [4]). Of course this concept has shown quite a positive effect on the search for proof systems for complex languages. Nevertheless it obscures the fact that a specification is likely to have more than one model.

Trying to work on recursion and concurrency however, we came to the conclusion that the sheer complexity of these proof systems defeats an attempt to prove more than just soundness and relative completeness.

Therefore, we hope to develop a propositional framework in which proof systems are essentially easier first. As is apparent in [1] the propositional approach solves the discrepancy between relative completeness and logical completeness at once, essentailly because all models are expressive in the propositional situation.

In forthcoming work we intend to involve more complex language features in the system as well.

1. PHL

1.1 SYNTACS: L is built from finitely many names for objects of the following three kinds:

- Atomic actions $A_0, A_1, \ldots, A_{n1}$.
- Boolean atoms $b_0, b_1, \ldots, b_{n2}$.
- Primitive assertions $p_0, p_1, \ldots, p_{n3}$.

Booleans are built up from the b_i using $\vee$ and $\neg$. Programs over L, WP(L) are inductively defined as follows:

$A_i \in WP(L)$ and if $S_0, S_1 \in WP(L)$ and b is a boolean then the following constructs are programs in WP(L) as well:

- $\underline{\text{if}}\ b\ \underline{\text{then}}\ S_0\ \underline{\text{else}}\ S_1\ \underline{\text{fi}}$
- $S_0 ; S_1$
- $\underline{\text{while}}\ b\ \underline{\text{do}}\ S_0\ \underline{\text{od}}$

Assertions are built from both boolean atoms and primitive assertions using $\vee$ and $\neg$. ASS(L) is the set of assertions over L. Note that each boolean is an assertion but not conversely.

An asserted program, (correctness assertion) is a construct of the form

$$\{P\}\ S\ \{Q\}$$

with $P,Q \in ASS(L)$, $S \in WP(L)$.

1.2 SEMANTICS: A structure α for L consists of a set V^α plus unary relations b_i^α $1 \le i \le n2$, p_i^α $1 \le i \le n3$ and binary relations A_i^α $1 \le i \le n1$ on it.

$$\alpha = \langle V^\alpha, b_i^\alpha, p_i^\alpha, A_i^\alpha \rangle$$

A_i^α is the input output graph of the nondeterministic action A_i in α . For a state σ in V^α one inductively defines

$$\alpha, \sigma \models P$$

for $P \in ASS(L)$ in the usual way, starting with $\alpha, \sigma \models b_i$ (p_i) iff $\sigma \in b_i^\alpha$, p_i^α .

Then there is a meaning function $M_{\mathcal{A}}$ which describes the semantics of programs in WP(L). Working inductively from $M_{\mathcal{A}}(A_i) = A_i^{\mathcal{A}}$, $M_{\mathcal{A}}$ produces just the operational (relational) semantics of <u>while</u> programs over $\mathcal{A}$. So $M_{\mathcal{A}}(S) \subseteq V^{\mathcal{A}} \times V^{\mathcal{A}}$.

Then one writes $\mathcal{A} \models P$ if for all σ $\mathcal{A}, \sigma \models P$ and $\mathcal{A} \models \{P\}\ S\ \{Q\}$ if for all $(\sigma,\tau) \in M_{\mathcal{A}}(S)$, $(\mathcal{A}, \sigma \models P) \Rightarrow (\mathcal{A}, \tau \models Q)$.

Assumptions are either assertions (P) or asserted atomic programs ($\{P\}\ A_i\ \{Q\}$). Γ will allways denote a finite set of assumptions. We write

$$\Gamma \models \{P\}\ S\ \{Q\}$$

$$(\Gamma \models P)$$

if in all models $\mathcal{A}$, with $\mathcal{A} \models \Gamma$, $\mathcal{A} \models \{P\}\ S\ \{Q\}$ $(\mathcal{A} \models P)$.

It is possible to restrict one's interest to models in which the $M_{\mathcal{A}}(A_i)$ are singlevalued relations, which assign to A_i a partial deterministic state transformation as meaning. Let us write $\models_D$ if only such deterministic structures are allowed.

From [1] there is this lemma.

1.3 LEMMA. *For all* Γ, $\{P\}\ S\ \{Q\}$ *the following are equivalent:*

(i) $\Gamma \models_D \{P\}\ S\ \{Q\}$

(ii) $\Gamma \models \{P\}\ S\ \{Q\}$

It turns out that nondeterministic semantics is technically easyer and in view of LEMMA 1.3 we adopt it without much loss of generality.

1.4 Before indicating a proof system we must explain several notational conventions. For instance T abbreviates $b_0 \vee \neg b_0$ and F abbreviates $b_0 \wedge \neg b_0$; *of course* $\wedge$ *and* $\supset$ *are introduced as abbreviations as well.*

P,Q,R, will denote assertions and b denotes a boolean. As we want to focus on proving asserted programs rather than assertions, the latter being the task of ordinary logic, we are satisfied with deriving $\Gamma \vdash P$ immediately from $\Gamma \models P$.

It must be noted that $\Gamma \models P$ iff $\Gamma^{SP} \models P$ where Γ^{SP}, the specification part of Γ, contains just the assertions in Γ. Clearly standard propositional logic provides us with many proof systems for $\Gamma^{SP} \models P$.

The following rules constitute an inductive definition of $\Gamma \vdash \{P\}\ S\ \{Q\}$ (and $\Gamma \vdash P$). A rule of the form

$$\frac{X_1,\ldots,X_k}{X}$$

should be read as follows:

if $\Gamma \vdash X_1,\ldots,\Gamma \vdash X_K$ then also $\Gamma \vdash X$.

Subset rule

$$\frac{\Gamma' \supseteq \Gamma,\ \Gamma \vdash \{P\}\ S\ \{Q\}}{\Gamma' \vdash \{P\}\ S\ \{Q\}}$$

Assertion rule

$$\Gamma \vdash P \quad \text{provided } \Gamma \models P$$

Selection rule

$$\Gamma \vdash \{P\}\ A_i\ \{Q\} \quad \text{provided } \{P\}\ A_i\ \{Q\} \in \Gamma$$

Rules for atomic programs

$$\Gamma \vdash \{F\}\ A_i\ \{F\}$$

$$\Gamma \vdash \{T\}\ A_i\ \{T\}$$

$$\frac{\{P\}\ A_i\ \{Q\}\ \{R\}\ A_i\ \{S\}}{\{P \wedge R\}\ A_i\ \{Q \wedge S\}}$$

$$\frac{\{P\}\ A_i\ \{Q\}\ \{R\}\ A_i\ \{S\}}{\{P \vee R\}\ A_i\ \{Q \wedge S\}}$$

Conditional rule

$$\frac{\{P \wedge b\}\ S_0\ \{Q\},\ \{P \wedge \neg b\}\ S_i\ \{Q\}}{\{P\}\ \underline{\text{if}}\ b\ \underline{\text{then}}\ S_0\ \underline{\text{else}}\ S_1\ \underline{\text{fi}}\ \{Q\}}$$

Composition rule

$$\frac{\{P\}\ S_0\ \{Q\}\ \{Q\}\ S_1\ \{R\}}{\{P\}\ S_0\ ;\ S_1\ \{R\}}$$

Iteration rule

$$\frac{\{P \wedge b\}\ S_0\ \{P\}}{\{P\}\ \underline{\text{while}}\ b\ \underline{\text{do}}\ S_0\ \underline{\text{od}}\ \{P \wedge \neg b\}}$$

Rule of consequence

$$\frac{P \supset P'\ \{P'\}\ S\ \{Q'\}\quad Q' \supset Q}{\{P\}\ S\ \{Q\}}$$

Finally we recall the theorem of soundness and completeness of $\vdash$ as stated in [1].

THEOREM. For all L and for each finite set of assumptions Γ over L and each asserted program $\{P\}\ S\ \{Q\}$ over L, $\Gamma \models \{P\}\ S\ \{Q\}$, if and only if, $\Gamma \vdash \{P\}\ S\ \{Q\}$.

2. MPHL

The next step is to augment the assertion language with a modal operator $\Box$. This leaves the set of programs WP(L) unchanged but adds one clause to the inductive definition of the set of assertions: if P is an assertion, then so is $\Box P$.

2.1 Structures are exactly the same as before and the semantics of $\Box$ is inductively laid down by defining

$$\alpha, \sigma \models \Box P \text{ if for all } \tau \in V^{\alpha} \quad \alpha, \tau \models P.$$

Then $\Diamond$ can be introduced as an abbreviation of $\neg\Box\neg$. Unambigously the meanings of $\alpha \models P$, $\alpha \models \{P\}\ S\ \{Q\}$ and $\Gamma \models \{P\}\ S\ \{Q\}$ are then defined as before.

This immediately opens the question for a sound and complete proof system. We will outline the system MPHL. It contains all rules of PHL mentioned in section 1 plus the following rules having to do with atomic programs $A = A_i$ only.

2.2 Rule of transitivity

$$\frac{\Gamma, R \vdash \{P\}\ A\ \{Q\},\ \Gamma \vdash R}{\Gamma \vdash \{P\}\ A\ \{Q\}}$$

Invariance rules

$$\Gamma \vdash \{\Box R\}\ A\ \{\Box R\}$$

$$\Gamma \vdash \{\Diamond R\}\ A\ \{\Diamond R\}$$

Deduction rules

$$\frac{\Gamma, \Box R \vdash \{P\}\ A\ \{Q\}}{\Gamma \vdash \{\Box R \wedge P\}\ A\ \{Q\}}$$

$$\frac{\Gamma, \Diamond R \vdash \{P\}\ A\ \{Q\}}{\Gamma \vdash \{\Diamond R \wedge P\}\ A\ \{Q\}}$$

2.3 THEOREM. Soundness of $\vdash$: $\Gamma \vdash \{P\}\ S\ \{Q\}$ implies $\Gamma \models \{P\}\ S\ \{Q\}$.

The proof od the soundness theorem is entirely straightforward and therefore omitted.

2.4 As a preparation for the completeness proof we provide some proof theoretic information in the form of derived rules of MPHL. We encode these facts in a theorem.

2.4.1 THEOREM. The following are derived rules in MPHL.

(i) Rule of transitivity

$$\frac{\Gamma, R \vdash \{P\}\ S\ \{Q\},\ \Gamma \vdash R}{\Gamma \vdash \{P\}\ S\ \{Q\}}$$

(ii) Deduction rules

$$\frac{\Gamma,\ \Box R \vdash \{P\}\ S\ \{Q\}}{\Gamma \vdash \{\Box R \wedge P\}\ S\ \{Q\}}$$

$$\frac{\Gamma,\ \Diamond R \vdash \{P\}\ S\ \{Q\}}{\Gamma \vdash \{\Diamond R \wedge P\}\ S\ \{Q\}}$$

(iii) $\Box$-shift rule

$$\frac{\Gamma \vdash \{\Box R \wedge P\}\ S\ \{Q\}}{\Gamma \vdash \{\Box R \wedge P\}\ S\ \{\Box R \wedge Q\}}$$

(iv) Disjunction rule

$$\frac{\{P\}\ S\ \{Q\},\ \{P'\}\ S\ \{Q'\}}{\{P \vee P'\}\ S\ \{Q \vee Q'\}}$$

(v) M-disjunction rule

$$\frac{\Gamma,\ \Box R \vdash \{P\}\ S\ \{Q\}, \Gamma,\ \Diamond \neg R \vdash \{P\}\ S\ \{Q\}}{\Gamma \vdash \{P\}\ S\ \{Q\}}$$

Proof of theorem 2.4.

(i) This is an immediate induction on proof length.

(ii) We will establish the deduction rule for $\Box$ and leave the other one to the reader. If S is an atomic program then the rule reduces to the according MPHL rule for atomic programs. In case S is not atomic we use induction on the structure of the proof of $\Gamma, \Box P \vdash \{P\}\ S\ \{Q\}$.

There are four cases and we will treat two of these: the ROC (rule of consequence) and the iteration rule case.

Assume for the ROC case that

$\Gamma, \Box R \vdash \{P\}\ S\ \{Q\}$ has been deduced from $\Gamma, \Box R \vdash P \supset P'$, $\Gamma, \Box R \vdash \{P'\}\ S\ \{Q'\}$ and $\Gamma, \Box R \vdash Q' \supset Q$.

Using the induction hypothesis and some propositional logic we may transform the initial data of this deduction into: $\Gamma \vdash (\Box R \wedge P) \supset (\Box R \wedge P')$, $\Gamma \vdash \{\Box R \wedge P'\}\ S\ \{Q'\}$, $\Gamma \vdash (\Box R \wedge Q') \supset Q$.

At this stage we use the $\Box$ shift rule (iii) to obtain $\Gamma \vdash \{\Box R \wedge P'\}\ S\ \{\Box R \wedge Q'\}$ and the ROC to obtain $\Gamma \vdash \{\Box R \wedge P\}\ S\ \{Q\}$.

The $\Box$-shift rule in turn has an entirely straightforward inductive proof.

Next we consider the case for the iteration rule:

So assume that $\Gamma, \Box R \vdash \{P\}$ <u>while</u> b <u>do</u> S <u>od</u> $\{P \wedge \neg b\}$ has been derived from $\Gamma, \Box R \vdash \{P \wedge b\}\ S\ \{P\}$. Using the induction hypothesis we obtain $\Gamma \vdash \{\Box R \wedge (P \wedge b)\}\ S\ \{P\}$. With the $\Box$-shift rule this yields $\Gamma \vdash \{\Box R \wedge (P \wedge b)\}\ S\ \{\Box R \wedge P\}$ and with ROC $\Gamma \vdash \{(\Box R \wedge P) \wedge b\}\ S\ \{\Box R \wedge P\}$. Then using the iteration rule we obtain $\Gamma \vdash \{\Box R \wedge P\}\ S\ \{(\Box R \wedge P) \wedge \neg b\}$ and with ROC once more $\Gamma \vdash \{\Box R \wedge P\}\ S\ \{P \wedge \neg b\}$ as desired.

The proofs of (iii) and (iv) are entirely straightforward. The M-disjunction rule finally works as follows:

$$\frac{\dfrac{\Gamma, \Box R \vdash \{P\}\ S\ \{Q\} \qquad \Gamma, \Diamond\neg R \vdash \{P\}\ S\ \{Q\}}{\Gamma \vdash \{\Box R \wedge P\}\ S\ \{Q\} \qquad \Gamma \vdash \{\Diamond\neg R \wedge P\}\ S\ \{Q\}}}{\Gamma \vdash \{(\Box R \wedge P) \vee (\Diamond\neg R \wedge P)\}\ S\ \{Q\}}$$

$$\Gamma \vdash \{P\}\ S\ \{Q\}$$

In this proof the deduction rules and the disjunction rule as well as the ROC have been applied. This finishes the proof of theorem 2.4.1.

3. COMPLETENESS OF MPHL

We shall now establish a completeness theorem for MPHL.

3.1 THEOREM. For each L,P,Q,S and finite Γ over L, $\Gamma \models \{P\}\ S\ \{Q\}$ implies $\Gamma \vdash \{P\}\ S\ \{Q\}$.

In order to prove completeness we need three lemmas that are worked out first.

3.2 LEMMA. Suppose $\Gamma, R_1, \ldots, R_K, P, Q$ do not contain $\Box$ or $\Diamond$, and assume $\Gamma^{SP}, \Diamond R_1 \ldots \Diamond R_K \not\vdash F$ then $\Gamma, \Diamond R_1, \ldots, \Diamond R_K \models \{P\}\ S\ \{Q\}$ implies

$$\Gamma \models \{P\}\ S\ \{Q\}.$$

PROOF. Let $\alpha \models \Gamma$ and $\alpha \not\models \{P\}\ S\ \{Q\}$ for a contradiction. Because $\Gamma^{SP} \not\vdash \Box\neg R_1, \ldots, \Gamma^{SP} \not\vdash \Box\neg R_K$ one can extend α to a structure α' which contains extra points in which the R_i are satisfied.

No extra pairs are added to the $M_\alpha(A_i)$. This results in $\alpha' \models \Gamma, \Diamond R_1, \ldots, \Diamond R_K$ and $\alpha \not\models \{P\}\ S\ \{Q\}$, contradicting the assumptions.

3.3 LEMMA. Suppose $\Gamma, R_1, \ldots, R_K, P, Q$ do not contain $\Box$ and $\Diamond$ then

$\Gamma, \Diamond R_1, \ldots, \Diamond R_K \models \{P\}\ S\ \{Q\}$ implies

$\Gamma, \Diamond R_1, \ldots, \Diamond R_K \vdash \{P\}\ S\ \{Q\}$.

PROOF. There are two cases. Case (i) is that $\Gamma, \Diamond R_1, \ldots, \Diamond R_K \vdash F$. Then immediately $\Gamma, \Diamond R_1, \ldots, \Diamond R_K \vdash \{P\}\ S\ \{Q\}$. In the other case (ii) we are back in the situation of the previous lemma. We find $\Gamma \models \{P\}\ S\ \{Q\}$, and then by the completeness theorem for PHL $\Gamma \vdash \{P\}\ S\ \{Q\}$, thus $\Gamma, \Diamond R_1, \ldots, \Diamond R_K \vdash \{P\}\ S\ \{Q\}$.

3.4 LEMMA. Let Γ', P', Q' result from Γ, P, Q by replacing all occurrences of $\Box R$ by T(F) then:

$$\frac{\Gamma', \Box R \vdash \{P'\}\ S\ \{Q'\}}{\Gamma, \Box R \vdash \{P\}\ S\ \{Q\}}$$

$$\left(\frac{\Gamma', \Diamond\neg R \vdash \{P'\}\ S\ \{Q'\}}{\Gamma, \Diamond\neg R \vdash \{P\}\ S\ \{Q\}}\right)$$

We omit the straightforward proofs of both facts.

Finally we are in the position to prove the completeness result.

Let Γ be a set of assumptions. We define a division $\Gamma = \Gamma_1 \cup \Gamma_2$ as follows:

Γ_2 contains those elements of Γ that are of the form $\Box R$ or of the form $\Diamond R$ where R may contain neither $\Box$ not $\Diamond$.

We show $\Gamma \models \{P\}\ S\ \{Q\} \Rightarrow \Gamma \vdash \{P\}\ S\ \{Q\}$ with induction on the number K of occurrences of $\Box$ and $\Diamond$ in $\Gamma_1 \cup \{P\} \cup \{Q\}$.

Suppose this number is zero. Then the situation looks as follows:

$\Gamma_1, \Box R_1,\ldots,\Box R_K, \Diamond R_{K+1},\ldots,\Diamond R_{K+t} \models \{P\}\ S\ \{Q\}$.

Thus clearly

$\Gamma_1, R_1,\ldots,R_K, \Diamond R_{K+1},\ldots,\Diamond R_{K+t} \models \{P\}\ S\ \{Q\}$.

By lemma 3.2.

$\Gamma_1, R_1,\ldots,R_K, \Diamond R_{K+1},\ldots,\Diamond R_{K+t} \vdash \{P\}\ S\ \{Q\}$.

Using the subset rule

$\Gamma_1, \Box R_1,\ldots,\Box R_K, R_1\ldots R_K, \Diamond R_{K+1},\ldots,\Diamond R_{K+t} \vdash \{P\}\ S\ \{Q\}$.

Because $\Box R_i \vdash R_i$ after K applications of the rule of transitivity

$\Gamma_1, \Box R_1,\ldots,\Box R_K, \Diamond R_{K+1},\ldots,\Diamond R_{K+t} \vdash \{P\}\ S\ \{Q\}$

i.e. $\Gamma \vdash \{P\}\ S\ \{Q\}$.

Now suppose $K > 0$. Let $\Box R$ occur in $\Gamma_1 \cup \{P\} \cup \{Q\}$, such that R does not contain $\Box$ or $\Diamond$. (Here it is important to view $\Diamond U$ just as $\neg\Box\neg U$.) We know $\Gamma \models \{P\}\ S\ \{Q\}$. Thus

$\Gamma, \Box R \models \{P\}\ S\ \{Q\}$ and $\Gamma, \Diamond\neg R \models \{P\}\ S\ \{Q\}$.

Clearly also

$\Gamma', \Box R \models \{P'\}\ S\ \{Q'\}$ and $\Gamma'', \Diamond\neg R \models \{P''\}\ S\ \{Q''\}$ where Γ', P', Q' result from replacing occurrences of $\Box R$ by T and Γ'', P'', Q'' result from replacing such occurrences by F. Clearly the number K for both $\Gamma' \cup \{P', Q'\}$ and $\Gamma'' \cup \{P'', Q''\}$

decreases. Therefore the induction hypothesis may be applied to obtain $\Gamma', \Box R \vdash \{P'\}\ S\ \{Q'\}$, $\Gamma'', \Diamond\neg R \vdash \{P''\}\ S\ \{Q''\}$. Then using lemma 3.4 one finds $\Gamma, \Box R \vdash \{P\}\ S\ \{Q\}$ and $\Gamma, \Diamond\neg R \vdash \{P\}\ S\ \{Q\}$. Finally with the M-disjunction rule we obtain $\Gamma \vdash \{P\}\ S\ \{Q\}$ just as required. This finishes the proof of the completeness theorem.

References

[1] BERGSTRA, J.A. & J. TERLOUW. A propositional version of Hoare's Logic. Bulletin of the EATCS no. 13, june 1981.

[2] BERGSTRA, J.A. & J.V. TUCKER. Expressiveness and the completeness of Hoare's Logic. Math. Centre Report IW 160/81, Amsterdam 1980.

[3] BERGSTRA, J.A. & J.V. TUCKER, Two theorems on the completeness of Hoare's Logic. Math. Centre Report IW 165/81, Amsterdam 1981.

[4] COOK, S.A. Soundness and completeness of an axiom system for program verification, SIAM J. on Computing, Vol. 7 nr. 1 (1978) 70-90.

[5] KOZEN, D. & R. PARIKH. An elementary proof of the completeness of PDL, TCS 14 (1981) 113-118.

REPRESENTATION OF CONTROL IN PARALLEL PROGRAMMED GRAMMARS

HORST BUNKE

Lehrstuhl für Informatik 5 (Mustererkennung)
Martensstr. 3
8520 Erlangen

Abstract

Control diagrams for programming the application order of productions in a formal grammar have been proposed recently. Generalized