1991

D.J.N. van Eijck, F.-J. de Vries

A sound and complete calculus for update logic

Computer Science/Department of Software Technology Report CS-R9155 December

CWI, nationaal instituut voor onderzoek op het gebied van wiskunde en informatica

CWI is the research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a non-profit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands organization for scientific research (NWO).

A Sound and Complete Calculus for Update Logic

Jan van Eijck[1,2] & Fer-Jan de Vries[1]

[1]CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

[2]OTS, Trans 10, 3512 JK Utrecht, The Netherlands

Abstract

This paper presents a sound and complete deduction system for Veltman's update logic in the style of the Hoare calculus with two kinds of correctness statements for dynamic assignment logic in Van Eijck and De Vries [13]. The Hoare correctness statements use modal propositional logic as assertion language and connect update logic to the modal propositional logic S5. The connection with modal propositional logic provides a clear link between the dynamic and the static semantics of update logic. The fact that update logic is decidable was noted already in [10]; the connection with S5 provides an alternative proof. The S5 connection can also be used for rephrasing the validity notions of update logic and for performing consistency checks.

1985 Mathematics Subject Classification: 03B65, 68Q55, 68S10.
CR Categories: F.3.1, I.2.4, I.2.7.
Keywords and Phrases: semantics of natural language, dynamic interpretation, Hoare logic, knowledge representation languages.

1 Introduction

Recently the semantics of natural language has witnessed a shift of emphasis from static meaning (truth conditions) to dynamic meaning (information processing). In dynamic semantics (see for example Karttunen [7], Stalnaker [9], Kamp [6], Heim [4] and Barwise [1]) one thinks of the meaning of a text as the change it brings about in the information state of anyone who accepts the information conveyed by it. One perspective on dynamic semantics for natural language is to view this approach as a proposal to represent the meanings of natural language sentences not by means of formulae from static logic but by means of expressions from a dynamic action language. The action languages that have been proposed display an intriguing mix of features of programming languages and features of logical languages. It has been demonstrated by example in [1] that compositional translations become feasible for fragments of natural language which include features that have resisted compositional treatment in the static representation approach (the most notable features being the handling of 'donkey' pronouns and pronominal binding across sentence boundaries).

The move from static logic to dynamic logic raises some interesting questions. In the first place, due to this transition we seem to have lost the deduction systems that static logical languages carry with them. This was noted by Barwise in [1], one of the first papers to propose an action language as representation medium for natural language meaning. Here the quest for a complete set of axioms for the dynamic inference notion that is engendered by the action language

Report CS-R9155
ISSN 0169-118X
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

is put forward as an open problem. Next it can be asked what is the precise relation between the static semantics and the dynamic semantics of natural language. Are there systematic ways to derive the static meaning of a sentence from its dynamic representation?

These questions are intimately connected. Our contention is that they should be tackled together and moreover that theoretical computer science can guide the way. In computer science the view that the meaning of a program is a function from information states to information states is common ground. In the case of imperative programming languages this perspective has led to Hoare logic [5] as a successful means to construct deductive systems for reasoning about imperative programs.

To apply this to natural language processing, consider a reader of a text π as an agent who uses π to update her or his knowledge φ. Unless there is a consistency clash the agent will end up with more specific knowledge φ'. Taking our cue from Hoare logic we ask the following question. What is the weakest formula φ such that any knowledge implying φ remains consistent during the process of absorbing the information from text π? This weakest precondition φ for successful processing represents the static meaning of text π. It is the careful analysis of these weakest preconditions that leads to Hoare deduction systems which are sound and complete for a given dynamic semantics. In this paper we will study Veltman's [14] update logic from this perspective.

To end this introduction, here is an overview of the contents and structure of the paper. In Section 2 Veltman's update logic is presented. Section 3 consists of a brief review of the tools from propositional modal logic that we will need. Section 4 contains the semantic definitions of the basic concepts of the paper that will serve as the starting point for our adaptation of Hoare's logic to update semantics. This section discusses two kinds of Hoare correctness statements with the two kinds of weakest preconditions that they give rise to. Section 5 links weakest preconditions to the next state conditions for update logic that were proposed by Van Benthem [10]. Section 6 contains the Hoare calculus engendered by the notion of weakest preconditions of update programs. In Section 7 we prove the soundness and in Section 8 the completeness of the Hoare calculus. In Section 9 we illustrate how the weakest preconditions analysis and the link to modal propositional logic can be used for reasoning about update logic. Finally, in Section 10, we demonstrate how weakest preconditions and the Hoare calculus that is based on them can be used for reasoning about consistency of update programs.

2 Update Logic: Syntax and Semantics

The characteristic feature of Veltman's update logic (see Veltman [14]) is the epistemic modal operator **might**. Due to the presence of this operator the meanings of 'update programs' have to be phrased in terms of input information *sets*, and have to be phrased dynamically. A program π maps an information state I to a new information state $[\![\pi]\!](I)$. To see that **might** is the key feature, note that the semantics for the fragment of update logic without **might** can be given by means of a dynamic yes/no function for individual propositional valuations, which reduces the semantics immediately to ordinary static propositional logic.

Following Veltman [14] (and in fact, slightly simplifying his syntax), we can define the language of epistemic update logic over a set of proposition letters P as the smallest set L_P such that the following hold:

Definition 1 (Syntax of Update Logic L_P)

1. $\perp \in L_P$.

2. *If* $p \in P$ *then* $p \in L_P$.

3. *If* π *and* $\pi' \in L_P$, *then* $(\pi;\pi') \in L_P, (\pi \cup \pi') \in L_P$.

4. *If* $\pi \in L_P$, *then* $\neg\pi \in L_P$, $\textbf{might}\ \pi \in L_P$.

5. *Nothing else is in* L_P.

The semantics of L_P is given in terms of input-output behaviour. We take the set W of worlds over P to be the set $\mathcal{P}P$. Any subset of W is an information state. Progams are interpreted as functions from information states to information states, i.e., as functions in $\mathcal{P}W \to \mathcal{P}W$. The clauses are as follows:

Definition 2 (Semantics of Update Logic)

1. $[\![\bot]\!](I) = \emptyset$.

2. $[\![p]\!](I) = I \cap \{w \mid p \in w\}$.

3. $[\![\pi;\pi']\!](I) = [\![\pi']\!]([\![\pi]\!](I))$.

4. $[\![\pi \cup \pi']\!](I) = [\![\pi]\!](I) \cup [\![\pi']\!](I)$.

5. $[\![\neg\pi]\!](I) = I - [\![\pi]\!](I)$.

6. $[\![\textbf{might}\ \pi]\!](I) = \begin{cases} I & \textit{if } [\![\pi]\!](I) \neq \emptyset, \\ \emptyset & \textit{otherwise.} \end{cases}$

We will follow the usual conventions and drop outermost parentheses as much as possible. Also, since sequential composition is associative we will write both $\pi_1;(\pi_2;\pi_3)$ and $(\pi_1;\pi_2);\pi_3$ as $\pi_1;\pi_2;\pi_3$.

Intuitively, a program of the form **might** π does not provide information about the world but about available information. A program **might** π is acceptable, given an information state I, if there is at least one world $w \in I$ for which π is accepted in the sense that $w \in [\![\pi]\!](I)$. If such a w can be found, the output information state of **might** π is equal to its input information state; this agrees with the intuition that **might** π does not say anything at all about what the world is like. In the other case, i.e., the case were $[\![\pi]\!](I) = \emptyset$, the output information state of **might** π equals $\emptyset$.

As was mentioned already, the **might** operator is the key feature of update logic. Yet another way to see this is to note that the semantic clause for **might** π introduces an element of *non distributivity* (this terminology is taken from [3]) into the semantics, in the sense that unions of input states do not distribute over output states: (1) does not in general hold.

(1) $$[\![\pi]\!](I) = \bigcup_{i \in I} [\![\pi]\!](\{i\}).$$

More specifically, it does not in general hold that $[\![\pi]\!](I) \subseteq \bigcup_{i \in I}[\![\pi]\!](\{i\})$. Counterexample: take π equal to **might** p and let $I = \{w, w'\}$ with $p \in w$ and $p \notin w'$. Then $[\![\textbf{might}\ p]\!](\{w\}) \cup [\![\textbf{might}\ p]\!](\{w'\}) = \{w\}$, but $[\![\textbf{might}\ p]\!](I) = I = \{w, w'\}$.

On the other hand, a simple induction on the complexity of π shows that Lemma 1 holds for all $\pi \in L_P$ and all information states I. In the terminology of Groenendijk & Stokhof [3]: epistemic update logic is *eliminative*.

Lemma 1 (Elimination Lemma) *For all* I: $[\![\pi]\!](I) \subseteq I$.

3 Modal Logic

The presence of the modal **might** operator in a dynamic setting which is otherwise fully propositional strongly suggests the use a modal propositional logic as language to make static assertions about update programs in, i.e., as assertion language. The syntax of our assertion language ml_P is as follows:

Definition 3 (Syntax of ml_P)

1. *$\bot \in ml_P$.*
2. *If $p \in P$ then $p \in ml_P$.*
3. *If $\varphi, \psi \in ml_P$, then $(\varphi \wedge \psi), \neg\pi, \Box\varphi \in ml_P$.*
4. *Nothing else is in ml_P.*

As is customary, we abbreviate $\neg\bot$ as $\top$, $\neg(\neg\varphi \wedge \neg\psi)$ as $(\varphi \vee \psi)$, $\neg(\varphi \wedge \neg\psi)$ as $(\varphi \rightarrow \psi)$, and $\neg\Box\neg\varphi$ as $\Diamond\varphi$. Also, we omit outermost parentheses for readability.

We consider information states $I \in \mathcal{P}W$ as universal Kripke models; thus, I is considered as the Kripke model with accessibility relation $I \times I$. Recall from the literature (see e.g. [2]) that the modal logic determined by the class of finite universal frames is S5. Moreover, for any finite universal model (a universal frame with valuations assigned to all of its worlds) there is a finite subset I of W validating the same formulae. I can be got by throwing away the extra copies of the worlds with identical valuations: because of the universal accessibility this makes no difference to validity.

The notion $I, w \Vdash \varphi$ (world w forces formula φ in information state I) is defined in the usual way. It is convenient to also define the interpretation of a formula with respect to an information state.

Definition 4 (Interpretation of φ with respect to I) $[\varphi]_I = \{w \in I : I, w \Vdash \varphi\}$.

In Hoare style reasoning about update logic the notions of relativisation and localisation of modal formulae play an important role. Localisations of modal formulae are defined in Kracht [8]. If φ, ψ are in ml_P, then $\varphi \downarrow \psi$, the localisation of φ to ψ, is given by the following definition.

Definition 5 (Localised modal formulae $\varphi \downarrow \psi$)

$$\begin{array}{lcl}
\bot \downarrow \psi & = & \bot \\
p \downarrow \psi & = & p \wedge \psi \\
(\varphi_1 \wedge \varphi_2) \downarrow \psi & = & (\varphi_1 \downarrow \psi) \wedge (\varphi_2 \downarrow \psi) \\
(\neg\varphi) \downarrow \psi & = & \psi \wedge \neg(\varphi \downarrow \psi) \\
(\Box\varphi) \downarrow \psi & = & \psi \wedge \Box(\psi \rightarrow (\varphi \downarrow \psi))
\end{array}$$

Localisation is closely related to the usual notion of a relativised modal formula.

Definition 6 (Relativised modal formulae φ^ψ)

$$\begin{array}{lcl}
\bot^\psi & = & \bot \\
p^\psi & = & \psi \rightarrow p \\
(\varphi_1 \wedge \varphi_2)^\psi & = & \varphi_1^\psi \wedge \varphi_2^\psi \\
(\neg\varphi)^\psi & = & \neg(\varphi^\psi) \\
(\Box\varphi)^\psi & = & \Box(\psi \rightarrow \varphi^\psi)
\end{array}$$

The connection between the two notions is given by the following lemma.

Lemma 2 (Van Benthem) *$\varphi \downarrow \psi$ iff $\varphi^{\psi} \wedge \psi$.*

Proof: Induction on the structure of φ. For example, in the case of negation the reasoning is as follows.

$$\begin{array}{rcll} \neg\varphi \downarrow \psi & = & \downarrow \text{def} & \psi \wedge \neg(\varphi \downarrow \psi) \\ & = & \text{ind hyp} & \psi \wedge \neg(\varphi^{\psi} \wedge \psi) \\ & = & \text{prop logic} & \psi \wedge \neg(\varphi^{\psi}) \\ & = & \downarrow \text{def} & \psi \wedge (\neg\varphi)^{\psi}. \end{array}$$

∎

Given this connection, the following lemma will not come as a surprise (the first two items are from the modal folklore, the third and the fourth from Kracht [8]).

Lemma 3 (Relativisation and Localisation)

1. *$I, w \Vdash \varphi^{\psi}$ iff $I, w \Vdash \psi$ implies $\{u \in I : I, u \Vdash \psi\}, w \Vdash \varphi$.*
2. $[\varphi^{\psi}]_I = (I - [\psi]_I) \cup [\varphi]_{[\psi]_I}$.
3. *$I, w \Vdash \varphi \downarrow \psi$ iff $I, w \Vdash \psi$ and $\{u \in I : I, u \Vdash \psi\}, w \Vdash \varphi$.*
4. $[\varphi \downarrow \psi]_I = [\varphi]_{[\psi]_I}$.

Proof: First note that 2 is a reformulation of 1 and 4 a reformulation of 3. Both 1 and 3 are proved by induction on the complexity of φ. ∎

4 Correctness Statements for Update Logic

As in the calculus for dynamic predicate logic presented in Van Eijck and De Vries [13], the deductive system for update logic is a hybrid calculus, with statements characterising information states (we call these: statements of the assertion language), plus two kinds of correctness statements, which we call *disjunctive* and *conjunctive* correctness statements. Because of the presence of a modal operator **might** in the language, we need modal operators in the assertion language. The following clauses define the language of H_P of Hoare statements over P.

Definition 7 (Hoare Logic over P: Syntax of H_P)

1. *If φ is a formula of mlp, then φ is a formula of H_P.*
2. *If φ, ψ are formulae of mlp and π is a program of L_P, then $\langle\varphi\rangle\ \pi\ \langle\psi\rangle$ and $\{\varphi\}\ \pi\ \{\psi\}$ are formulae of H_P.*
3. *Nothing else is a formula of H_P.*

The H_P statements of the form φ are used for making assertions about information states I (viewed as universal Kripke models). The H_P statements of the form $\{\varphi\}\ \pi\ \{\psi\}$ are *disjunctive* correctness statements. The statement $\{\varphi\}\ \pi\ \{\psi\}$ expresses that all information states I and all $w \in W$ such that $I, w \Vdash \varphi$ have the property that either $w \notin [\![\pi]\!](I)$ or $[\![\pi]\!](I), w \Vdash \psi$. The H_P statements of the form $\langle\varphi\rangle\ \pi\ \langle\psi\rangle$ are *conjunctive* correctness statements. The statement $\langle\varphi\rangle\ \pi\ \langle\psi\rangle$ expresses that all input information states I and all $w \in I$ such that $I, w \Vdash \varphi$ have the property that $w \in [\![\pi]\!](I)$ and $I, w \Vdash \psi$. Formally:

Definition 8 (Validity of H_P formulae)

1. *If F has the form $\varphi \in mlp$ then $\models F$ if for all $I \subseteq W$ and all $w \in I$ it holds that $I, w \Vdash F$.*

2. *If F has the form $\{\varphi\}\ \pi\ \{\psi\}$ then $\models F$ if for all $I \subseteq W$ and all $w \in I$ such that $I, w \Vdash \varphi$ it holds that $w \notin [\![\pi]\!](I)$ or $[\![\pi]\!](I), w \Vdash \psi$.*

3. *If F has the form $\langle\varphi\rangle\ \pi\ \langle\psi\rangle$ then $\models F$ if for all $I \subseteq W$ and all $w \in I$ such that $I, w \Vdash \varphi$ it holds that $w \in [\![\pi]\!](I)$ and $[\![\pi]\!](I), w \Vdash \psi$.*

The next lemma gives a useful reformulation.

Lemma 4

1. $\models \{\varphi\}\ \pi\ \{\psi\}$ *iff for all I:* $[\varphi]_I \subseteq (I - [\![\pi]\!](I)) \cup [\psi]_{[\![\pi]\!](I)}$.

2. $\models \langle\varphi\rangle\ \pi\ \langle\psi\rangle$ *iff for all I:* $[\varphi]_I \subseteq [\psi]_{[\![\pi]\!](I)}$.

Proof: Immediate from Definition 4, Definition 8 and Lemma 1. ■

Next, let us explain why we need both kinds of correctness statements in our calculus. Suppose we want to express in Hoare logic that for all I, $[\varphi]_I$ is equal to $[\![\pi]\!](I)$. Then the conjunction of $\models \langle\varphi\rangle\ \pi\ \langle\top\rangle$ and $\models \{\neg\varphi\}\ \pi\ \{\bot\}$ does the job, witness the following lemma.

Lemma 5 $\models \langle\varphi\rangle\ \pi\ \langle\top\rangle$ *and* $\models \{\neg\varphi\}\ \pi\ \{\bot\}$ *iff for all I:* $[\varphi]_I = [\![\pi]\!](I)$.

Proof: Note the following equivalences:

$\models \langle\varphi\rangle\ \pi\ \langle\top\rangle$	iff	Lemma 4	for all I: $[\varphi]_I \subseteq [\top]_{[\![\pi]\!](I)}$
	iff	$[\ ]_I$ def	for all I: $[\varphi]_I \subseteq [\![\pi]\!](I)$.

Also:

$\models \{\neg\varphi\}\ \pi\ \{\bot\}$	iff	Lemma 4	for all I: $[\neg\varphi]_I \subseteq (I - [\![\pi]\!](I)) \cup [\bot]_I$
	iff	$[\ ]_I$ def	for all I: $I - [\varphi]_I \subseteq I - [\![\pi]\!](I)$
	iff	Lemma 1	for all I: $[\varphi]_I \supseteq [\![\pi]\!](I)$.

Combine these results and the lemma follows. ■

Corresponding with the two kinds of correctness statements we have two notions of weakest precondition. We will call them respectively weakest conjunctive precondition and weakest disjunctive precondition.

Definition 9 *A formula $\varphi \in mlp$ is a weakest disjunctive precondition (WDP) of the program $\pi \in L_P$ and the formula $\psi \in mlp$ if for all I and all $w \in I$:*

$$I, w \Vdash \varphi \text{ iff } w \notin [\![\pi]\!](I) \text{ or } [\![\pi]\!](I), w \Vdash \psi.$$

A formula $\varphi \in mlp$ is a weakest conjunctive precondition (WCP) of the program $\pi \in L_P$ and the formula $\psi \in mlp$ if for all I and all $w \in I$:

$$I, w \Vdash \varphi \text{ iff } w \in [\![\pi]\!](I) \text{ and } [\![\pi]\!](I), w \Vdash \psi.$$

The following lemma gives a useful reformulation of this definition.

Lemma 6

1. *φ is a WDP of π and ψ iff for all I: $[\varphi]_I = (I - [\![\pi]\!](I)) \cup [\psi]_{[\![\pi]\!](I)}$.*

2. *φ is a WCP of π and φ iff for all I: $[\varphi]_I = [\psi]_{[\![\pi]\!](I)}$.*

Proof: Immediate from Lemma 4 and Definition 9. ■

We can say a bit more about the relation between WCP and WDP.

Lemma 7 *φ is a WCP of π and ψ iff $\neg\psi$ is a WDP of π and $\neg\varphi$.*

Proof:

φ is a WCP of π and φ

iff	Lemma 6	for all I: $[\varphi]_I = [\psi]_{[\![\pi]\!](I)}$
iff		for all I: $I - [\varphi]_I = I - [\psi]_{[\![\pi]\!](I)}$
iff	Lemma 1	for all I: $I - [\varphi]_I = (I - ([\![\pi]\!](I)) \cup ([\![\pi]\!](I) - [\psi]_{[\![\pi]\!](I)})$
iff	$[\ ]_I$ def	for all I: $[\neg\varphi]_I = (I - ([\![\pi]\!](I)) \cup [\neg\psi]_{[\![\pi]\!](I)}$
iff	Lemma 6	$\neg\varphi$ is a WDP of π and $\neg\psi$.

■

It is not obvious at first sight that WDPs and WCPs of an L_P program π and an ml_P formula ψ always exist (as formulae of ml_P). We will demonstrate now that they do, by first inductively defining a function $\mathbf{wcp}(\pi, \psi)$, of which we will show that it expresses a WCP of π and ψ, and then defining a function $\mathbf{wdp}(\pi, \psi)$ in terms of $\mathbf{wcp}(\pi, \psi)$ which expresses a WDP of π and ψ.

Definition 10 (wcp)

1. $\mathbf{wcp}(\bot, \psi) = \bot$.

2. $\mathbf{wcp}(p, \psi) = \psi \downarrow p$.

3. $\mathbf{wcp}(\pi_1; \pi_2, \psi) = \mathbf{wcp}(\pi_1, \mathbf{wcp}(\pi_2, \psi))$.

4. $\mathbf{wcp}(\pi_1 \cup \pi_2, \psi) = \psi \downarrow (\mathbf{wcp}(\pi_1, \top) \vee \mathbf{wcp}(\pi_2, \top))$.

5. $\mathbf{wcp}(\neg\pi, \psi) = \psi \downarrow \neg\mathbf{wcp}(\pi, \top)$.

6. $\mathbf{wcp}(\mathbf{might}\ \pi, \psi) = \Diamond\mathbf{wcp}(\pi, \top) \wedge \psi$.

Lemma 8 shows that the function $\mathbf{wcp}(\pi, \psi)$ does indeed express a WCP of a program π and a modal propositional formula ψ.

Lemma 8 (wcp adequacy) $[\mathbf{wcp}(\pi, \psi)]_I = [\psi]_{[\![\pi]\!](I)}$.

Proof: We prove the claim with induction on the structure of π.

$$\begin{array}{llll} [\mathbf{wcp}(\bot, \psi)]_I & = & \text{wcp def} & [\bot]_I \\ & = & [\ ]_I \text{ def} & \emptyset \\ & = & [\![\ ]\!] \text{ def} & [\psi]_{[\![\bot]\!](I)}. \end{array}$$

$$\begin{array}{llll} [\mathbf{wcp}(p, \psi)]_I & = & \text{wcp def} & [\psi \downarrow p]_I \\ & = & \text{loc lemma} & [\psi]_{[p]_I} \\ & = & [\![\ ]\!] \text{ def} & [\psi]_{[\![p]\!](I)}. \end{array}$$

$$
\begin{aligned}
[\mathbf{wcp}(\pi_1;\pi_2,\psi)]_I &= \text{wcp def} & & [\mathbf{wcp}(\pi_1,\mathbf{wcp}(\pi_2,\psi))]_I \\
&= \text{ind hyp} & & [\mathbf{wcp}(\pi_2,\psi)]_{[\![\pi_1]\!](I)} \\
&= \text{ind hyp} & & [\psi]_{[\![\pi_2]\!]([\![\pi_1]\!](I))} \\
&= [\![\ ]\!] \text{ def} & & [\psi]_{[\![\pi_1;\pi_2]\!](I)}.
\end{aligned}
$$

$$
\begin{aligned}
[\mathbf{wcp}(\pi_1 \cup \pi_2,\psi)]_I &= \text{wcp def} & & [\psi \downarrow (\mathbf{wcp}(\pi_1,\top) \vee \mathbf{wcp}(\pi_2,\top))]_I \\
&= \text{loc lemma} & & [\psi]_{[\mathbf{wcp}(\pi_1,\top)\vee\mathbf{wcp}(\pi_2,\top)]_I} \\
&= \text{ind hyp , } [\ ]_I \text{ def} & & [\psi]_{[\![\pi_1]\!](I)\cup[\![\pi_2]\!](I)} \\
&= [\![\ ]\!] \text{ def} & & [\psi]_{[\![\pi_1\cup\pi_2]\!](I)}.
\end{aligned}
$$

$$
\begin{aligned}
[\mathbf{wcp}(\neg\pi,\psi)]_I &= \text{wcp def} & & [\psi \downarrow \neg\mathbf{wcp}(\pi,\top)]_I \\
&= \text{loc lemma} & & [\psi]_{[\neg\mathbf{wcp}(\pi,\top)]_I} \\
&= [\ ]_I \text{ def} & & [\psi]_{I-[\mathbf{wcp}(\pi,\top)]_I} \\
&= \text{ind hyp, } [\ ]_I \text{ def} & & [\psi]_{I-[\![\pi]\!](I)} \\
&= [\![\ ]\!] \text{ def} & & [\![\psi]_{[\![\neg\pi]\!](I)}.
\end{aligned}
$$

$$
\begin{aligned}
[\mathbf{wcp}(\mathbf{might}\ \pi,\psi)]_I &= \text{wcp def} & & [\Diamond\mathbf{wcp}(\pi,\top)\wedge\psi]_I \\
&= [\ ]_I \text{ def} & & [\Diamond\mathbf{wcp}(\pi,\top)]_I \cap [\psi]_I \\
&= \Diamond \text{ def in S5} & & \begin{cases} [\psi]_I & \text{if } [\mathbf{wcp}(\pi,\top)]_I \neq \emptyset, \\ \emptyset & \text{otherwise} \end{cases} \\
&= \text{ind hyp, } [\ ]_I \text{ def} & & \begin{cases} [\psi]_I & \text{if } [\![\pi]\!](I) \neq \emptyset, \\ \emptyset & \text{otherwise} \end{cases} \\
&= [\![\ ]\!] \text{ def} & & [\psi]_{[\![\mathbf{might}\ \pi]\!](I)}.
\end{aligned}
$$

This completes the proof of the lemma. ∎

Definition 11 $\mathbf{wdp}(\pi,\psi) \stackrel{\text{def}}{=} \neg\mathbf{wcp}(\pi,\neg\psi)$.

Lemma 9 (wdp adequacy) $[\mathbf{wdp}(\pi,\psi)]_I = (I - [\![\pi]\!](I)) \cup [\psi]_{[\![\pi]\!](I)}$.

Proof: Immediate from Lemma 7 and the **wcp** adequacy lemma. ∎

Lemma 10

1. $[\mathbf{wcp}(\pi,\top)]_I = [\![\pi]\!](I)$.
2. $[\mathbf{wdp}(\pi,\bot)]_I = I - [\![\pi]\!](I)$.
3. $\models \langle\varphi\rangle\ \pi\ \langle\top\rangle$ *and* $\models \{\neg\varphi\}\ \pi\ \{\bot\}$ *iff* *for all* I*:* $[\mathbf{wcp}(\pi,\top)]_I = [\varphi]_I$.

Proof:
The first item:

$$
\begin{aligned}
[\mathbf{wcp}(\pi,\top)]_I &= \text{wcp adeq} & & [\top]_{[\![\pi]\!](I)} \\
&= [\ ]_I \text{ def} & & [\![\pi]\!](I).
\end{aligned}
$$

The second item:

$$
\begin{aligned}
[\mathbf{wdp}(\pi,\bot)]_I &= \text{wdp def} & & [\neg\mathbf{wcp}(\pi,\top)]_I \\
&= [\ ]_I \text{ def} & & I - [\mathbf{wcp}(\pi,\top)]_I \\
&= \text{previous item} & & I - [\![\pi]\!](I).
\end{aligned}
$$

The third item: immediate from Lemma 5 and the first item. ■

Lemma 11 $[\mathbf{wcp}(\pi, \psi)]_I = [\psi \downarrow \mathbf{wcp}(\pi, \top)]_I$.

Proof:

$$\begin{array}{rcll} [\mathbf{wcp}(\pi,\psi)]_I & = & \text{wcp adeq} & [\psi]_{[\![\pi]\!](I)} \\ & = & \text{Lemma 10} & [\psi]_{[\mathbf{wcp}(\pi,\top)]} \\ & = & \text{loc lemma} & [\psi \downarrow \mathbf{wcp}(\pi,\top)]_I. \end{array}$$

■

5 Weakest Preconditions Versus Next State Conditions

In [10] and [11], Van Benthem has studied update logic by looking at update programs π as functions of the form $\lambda I \cdot \text{NEXT STATE}(I, \pi)$, were NEXT STATE is the function producing the information state which results from processing π in information state I, i.e., π is considered as $\lambda I \cdot [\![\pi]\!](I)$. The investigation in [10, 11] was carried out in semantic terms, without reference to a specific assertion language, but it can easily be transposed in a setting of assertions from modal propositional logic. Some illuminating conversations between Johan van Benthem and the authors, backed up by an exchange of letters of explanation and consecutive drafts of the present paper, have fully cleared up the connection between his perspective and ours. His generous help in clarifying the issues raised in this section is herewith gratefully acknowledged.

Definition 12 *A formula $\psi \in mlp$ is a next state condition (NSC) of the formula $\varphi \in mlp$ and the program $\pi \in L_P$ if for all I and all $w \in I$:*

$$I, w \Vdash \psi \text{ iff } w \in [\![\pi]\!]([\varphi]_I).$$

Again, it is helpful to reformulate this. Lemma 12 follows immediately from the definition.

Lemma 12 *ψ is a NSC of φ and π iff for all I:* $[\psi]_I = [\![\pi]\!]([\varphi]_I)$.

The following function **nsc** is a reformulation in modal logic of Van Benthem's characterisation of the next state function.

Definition 13 (nsc)

1. $\mathbf{nsc}(\varphi, \bot) = \bot$.
2. $\mathbf{nsc}(\varphi, p) = p \wedge \varphi$.
3. $\mathbf{nsc}(\varphi, \pi_1; \pi_2) = \mathbf{nsc}(\mathbf{nsc}(\varphi, \pi_1), \pi_2)$.
4. $\mathbf{nsc}(\varphi, \pi_1 \cup \pi_2) = \mathbf{nsc}(\varphi, \pi_1) \vee \mathbf{nsc}(\varphi, \pi_2)$.
5. $\mathbf{nsc}(\varphi, \neg\pi) = \neg\mathbf{nsc}(\varphi, \pi) \wedge \varphi$.
6. $\mathbf{nsc}(\varphi, \mathbf{might}\ \pi) = \Diamond\mathbf{nsc}(\varphi, \pi) \wedge \varphi$.

Lemma 13 (nsc adequacy) $[\mathbf{nsc}(\varphi, \pi)]_I = [\![\pi]\!]([\varphi]_I)$.

Induction on the structure of π.

$$\begin{array}{rcll}
[\mathbf{nsc}\,(\varphi,\bot)]_I & = & \text{nsc def} & [\bot]_I \\
& = & [\;]_I \text{ def} & \emptyset \\
& = & [\![\;]\!] \text{ def} & [\![\bot]\!]([\varphi]_I).
\end{array}$$

$$\begin{array}{rcll}
[\mathbf{nsc}\,(\varphi,p)]_I & = & \text{nsc def} & [p\wedge\psi]_I \\
& = & [\;]_I \text{ def} & [p]_I\cap[\varphi]_I \\
& = & [\![\;]\!] \text{ def} & [\![p]\!]([\varphi]_I).
\end{array}$$

$$\begin{array}{rcll}
[\mathbf{nsc}\,(\varphi,\pi_1;\pi_2)]_I & = & \text{nsc def} & [\mathbf{nsc}\,(\mathbf{nsc}\,(\varphi,\pi_1),\pi_2)]_I \\
& = & \text{ind hyp} & [\![\pi_2]\!]([\mathbf{nsc}\,(\varphi,\pi_1)]_I) \\
& = & \text{ind hyp} & [\![\pi_2]\!]([\![\pi_1]\!]([\varphi]_I)) \\
& = & [\![\;]\!] \text{ def} & [\![\pi_1;\pi_2]\!]([\varphi]_I).
\end{array}$$

$$\begin{array}{rcll}
[\mathbf{nsc}\,(\varphi,\pi_1\cup\pi_2)]_I & = & \text{nsc def} & [\mathbf{nsc}\,(\varphi,\pi_1)\vee\mathbf{nsc}\,(\varphi,\pi_2)]_I \\
& = & \text{ind hyp} & [\![\pi_1]\!]([\varphi]_I)\cup[\![\pi_2]\!]([\varphi]_I) \\
& = & [\![\;]\!] \text{ def} & [\![\pi_1\cup\pi_2]\!]([\varphi]_I).
\end{array}$$

$$\begin{array}{rcll}
[\mathbf{nsc}\,(\varphi,\neg\pi)]_I & = & \text{nsc def} & [\neg\mathbf{nsc}\,(\varphi,\pi)\wedge\varphi]_I \\
& = & [\;]_I \text{ def} & (I-[\mathbf{nsc}\,(\varphi,\pi)]_I)\cap[\varphi]_I \\
& = & \text{ind hyp} & (I-[\![\pi]\!]([\varphi]_I))\cap[\varphi]_I \\
& = & \text{elim lemma} & ([\varphi]_I-[\![\pi]\!]([\varphi]_I))\cap[\varphi]_I \\
& = & [\![\;]\!] \text{ def} & [\![\neg\pi]\!]([\varphi]_I).
\end{array}$$

$$\begin{array}{rcll}
[\mathbf{nsc}\,(\varphi,\mathbf{might}\;\pi)]_I & = & \text{nsc def} & [\Diamond\mathbf{nsc}\,(\varphi,\pi)\wedge\varphi]_I \\
& = & [\;]_I \text{ def} & [\Diamond\mathbf{nsc}\,(\varphi,\pi)]_I\cap[\varphi]_I \\
& = & \Diamond \text{ def in S5} & \begin{cases} [\varphi]_I & \text{if } [\mathbf{nsc}\,(\varphi,\pi)]_I\neq\emptyset, \\ \emptyset & \text{otherwise} \end{cases} \\
& = & \text{ind hyp} & \begin{cases} [\varphi]_I & \text{if } [\![\pi]\!]([\varphi]_I)\neq\emptyset, \\ \emptyset & \text{otherwise} \end{cases} \\
& = & [\![\;]\!] \text{ def} & [\![\mathbf{might}\;\pi]\!]([\varphi]_I).
\end{array}$$

This completes the proof of the lemma. ∎

Lemma 14 $[\mathbf{nsc}\,(\top,\pi)]_I=[\![\pi]\!](I)$.

Proof:

$$\begin{array}{rcll}
[\mathbf{nsc}\,(\top,\pi)]_I & = & \text{nsc adeq} & [\![\pi]\!]([\top]_I) \\
& = & [\;]_I \text{ def} & [\![\pi]\!](I).
\end{array}$$

∎

The following theorem gives the precise connections between WCPs and NSCs.

Theorem 15

1. $[\mathbf{nsc}\,(\varphi,\pi)]_I=[\mathbf{wcp}(\pi,\top)\downarrow\varphi]_I$.
2. $[\mathbf{wcp}(\pi,\psi)]_I=[\psi\downarrow\mathbf{nsc}\,(\top,\pi)]_I$.

Proof:
The first item:

$$
\begin{aligned}
\mathbf{nsc}\,(\varphi, \pi)]_I &= \text{nsc adeq} && [\![\pi]\!]([\varphi]_I) \\
&= \text{Lemma 10} && [\mathbf{wcp}(\pi, \top)]_{[\varphi]_I} \\
&= \text{loc lemma} && [\mathbf{wcp}(\pi, \top) \downarrow \varphi]_I.
\end{aligned}
$$

The second item:

$$
\begin{aligned}
[\mathbf{wcp}(\pi, \psi)]_I &= \text{wcp adeq} && [\psi]_{[\![\pi]\!](I)} \\
&= \text{Lemma 14} && [\psi]_{[\mathbf{nsc}\,(\top, \pi)]_I} \\
&= \text{loc lemma} && [\psi \downarrow \mathbf{nsc}\,(\top, \pi)]_I.
\end{aligned}
$$

■

6 A Hoare Calculus for Update Logic

We now present the axioms and rules of a deduction system for update logic based on the concepts of WCPs and WDPs from Section 4. As was explained there, the need to have two kinds of correctness statements arises from the fact that we have to specify weakest preconditions for success of a program π. For example, in the rule for negation, to specify that $\neg\varphi$ is the weakest precondition for success of π we use a conjunctive premiss $\langle\neg\varphi\rangle\ \pi\ \langle\top\rangle$ together with a disjunctive premiss $\{\varphi\}\ \pi\ \{\bot\}$ (see Lemma 5).

6.1 The rules for $\bot$

$$
\frac{}{\{\top\}\ \bot\ \{\bot\}} \qquad \frac{}{\langle\bot\rangle\ \bot\ \langle\bot\rangle}
$$

6.2 The rules for p

$$
\frac{}{\{p \to (\varphi \downarrow p)\}\ p\ \{\varphi\}} \qquad \frac{}{\langle\varphi \downarrow p\rangle\ p\ \langle\varphi\rangle}
$$

6.3 The oracle rule

For every formula derivable in S5 we add the following rule (recall that derivability in S5 is decidable):

$$
\frac{}{\varphi}\ \text{oracle rule}
$$

6.4 The consequence rules

$$
\frac{\varphi \to \psi \qquad \{\psi\}\ \pi\ \{\chi\} \qquad \chi \to \xi}{\{\varphi\}\ \pi\ \{\xi\}}
$$

$$
\frac{\varphi \to \psi \qquad \langle\psi\rangle\ \pi\ \langle\chi\rangle \qquad \chi \to \xi}{\langle\varphi\rangle\ \pi\ \langle\xi\rangle}
$$

6.5 The rules for sequential composition

$$\frac{\{\varphi\}\ \pi_1\ \{\psi\} \qquad \{\psi\}\ \pi_2\ \{\chi\}}{\{\varphi\}\ (\pi_1;\pi_2)\ \{\chi\}}$$

$$\frac{\langle\varphi\rangle\ \pi_1\ \langle\psi\rangle \qquad \langle\psi\rangle\ \pi_2\ \langle\chi\rangle}{\langle\varphi\rangle\ (\pi_1;\pi_2)\ \langle\chi\rangle}$$

6.6 Rules for disjunction

$$\frac{\langle\varphi\rangle\ \pi_1\ \langle\top\rangle \qquad \{\neg\varphi\}\ \pi_1\ \{\bot\} \qquad \langle\psi\rangle\ \pi_2\ \langle\top\rangle \qquad \{\neg\psi\}\ \pi_2\ \{\bot\}}{\{(\varphi\vee\psi)\rightarrow(\chi\downarrow(\varphi\vee\psi))\}\ (\pi_1\cup\pi_2)\ \{\chi\}}$$

$$\frac{\langle\varphi\rangle\ \pi_1\ \langle\top\rangle \qquad \{\neg\varphi\}\ \pi_1\ \{\bot\} \qquad \langle\psi\rangle\ \pi_2\ \langle\top\rangle \qquad \{\neg\psi\}\ \pi_2\ \{\bot\}}{\langle\chi\downarrow(\varphi\vee\psi)\rangle\ (\pi_1\cup\pi_2)\ \langle\chi\rangle}$$

6.7 Rules for negation

$$\frac{\{\varphi\}\ \pi\ \{\bot\} \qquad \langle\neg\varphi\rangle\ \pi\ \langle\top\rangle}{\{\varphi\rightarrow(\psi\downarrow\varphi)\}\ \neg\pi\ \{\psi\}}$$

$$\frac{\{\varphi\}\ \pi\ \{\bot\} \qquad \langle\neg\varphi\rangle\ \pi\ \langle\top\rangle}{\langle\psi\downarrow\varphi\rangle\ \neg\pi\ \langle\psi\rangle}$$

6.8 Rules for might

$$\frac{\{\varphi\}\ \pi\ \{\bot\}}{\{\Box\varphi\vee\psi\}\ \mathbf{might}\ \pi\ \{\psi\}}$$

$$\frac{\langle\varphi\rangle\ \pi\ \langle\top\rangle}{\langle\Diamond\varphi\wedge\psi\rangle\ \mathbf{might}\ \pi\ \langle\psi\rangle}$$

If $F_1,\cdots,F_n$ is a sequence of H_P statements each member of which is either an axiom or a statement following by the rules of the calculus from preceding statements in the sequence, then $F_1,\cdots,F_n$ is a *derivation* in the calculus. If F equals F_n for some derivation $F_1,\cdots,F_n$, then F is a *theorem* of the calculus (notation: $\vdash F$), and n is the *length* of the derivation of F. Note that we consider only derivations without hypotheses.

7 Soundness of the Calculus

To prove that the calculus is sound, we have to prove that if $F \in H_P$ is a theorem of the calculus, then F is valid (in the sense defined in Section 4). As usual, soundness is proved by induction on the length of the derivation of F. We are done if we can show that every axiom of the calculus is valid and that the rules of the calculus preserve validity.

Lemma 16 (wcp/wdp validity)

1. $\models \{\mathbf{wdp}(\pi, \psi)\}\ \pi\ \{\psi\}$.

2. $\models \langle\mathbf{wcp}(\pi, \psi)\rangle\ \pi\ \langle\psi\rangle$.

Proof: Immediate from Lemma 6, Lemma 8 and Definition 11. ∎

Theorem 17 (Soundness) *For all $F \in H_P$: If $\vdash F$ then $\models F$.*

Proof: First, the soundness of the test axioms follows immediately from the **wcp/wdp** validity lemma, for the test axioms all have the form $\{\mathbf{wdp}(\pi, \psi)\}\ \pi\ \{\psi\}$ or $\langle\mathbf{wcp}(\pi, \psi)\rangle\ \pi\ \langle\psi\rangle$.

The soundness of the oracle rule follows from the fact that for every S5 theorem φ and every information state I it holds, by the soundness of S5, that $[\varphi]_I = I$.

For the other rules we only give the argument for the conjunctive cases, the disjunctive cases being similar. For the soundness of the conjunctive consequence rule, assume that $\vdash \varphi \rightarrow \psi$, $\vdash \langle\psi\rangle\ \pi\ \langle\chi\rangle$ and $\vdash \chi \rightarrow \xi$. By the induction hypothesis we get from these three premisses respectively the following, for every information state I:

(2) $\quad [\varphi]_I \subseteq [\psi]_I$.

(3) $\quad [\psi]_I \subseteq [\chi]_{[\![\pi]\!](I)}$.

(4) $\quad [\chi]_{[\![\pi]\!](I)} \subseteq [\xi]_{[\![\pi]\!](I)}$.

This yields conclusion (5).

(5) $\quad [\varphi]_I \subseteq [\xi]_{[\![\pi]\!](I)}$.

Because I was arbitrary, conclude that $\models \langle\varphi\rangle\ \pi\ \langle\xi\rangle$.

For the soundness of the conjunctive rule of composition, assume $\vdash \langle\varphi\rangle\ \pi_1\ \langle\psi\rangle$ and $\vdash \langle\psi\rangle\ \pi_2\ \langle\chi\rangle$. By the induction hypothesis, these premises give the following, again for every I:

(6) $\quad [\varphi]_I \subseteq [\psi]_{[\![\pi_1]\!](I)}$.

(7) $\quad [\psi]_{[\![\pi_1]\!](I)} \subseteq [\chi]_{[\![\pi_2]\!]([\![\pi_1]\!](I))}$.

Combining these two we get (8).

(8) $\quad [\varphi]_I \subseteq [\chi]_{[\![\pi_2]\!]([\![\pi_1]\!](I))}$.

From this by the semantic clause for sequential composition:

(9) $\quad [\varphi]_I \subseteq [\chi]_{[\![\pi_1;\pi_2]\!](I)}$.

Since I was arbitrary, this establishes $\models \langle\varphi\rangle\ \pi_1;\pi_2\ \langle\chi\rangle$.

For the soundness of the conjunctive rule of union, assume $\vdash \langle\varphi\rangle\ \pi_1\ \langle\top\rangle$, $\vdash \{\neg\varphi\}\ \pi_1\ \{\bot\}$, $\vdash \langle\psi\rangle\ \pi_2\ \langle\top\rangle$, $\vdash \{\neg\psi\}\ \pi_2\ \{\bot\}$. From these four premisses, with the induction hypothesis, Lemma 5 and Lemma 10:

(10) $\quad \models \varphi \leftrightarrow \mathbf{wcp}(\pi_1, \top)$.

(11) $\quad \models \psi \leftrightarrow \mathbf{wcp}(\pi_2, \top)$.

Now use the **wcp** validity lemma to conclude from this that (12).

(12) $\quad \models \langle\chi \downarrow (\varphi \vee \psi)\rangle\ \pi_1 \cup \pi_2\ \langle\chi\rangle$.

For the soundness of the conjunctive rule of negation, the reasoning follows the same pattern.

For the soundness of the conjunctive rule of **might**, assume $\vdash \langle\varphi\rangle\ \pi\ \langle\top\rangle$. By the induction hypothesis this yields (13), for every I.

(13) $\quad [\varphi]_I \subseteq [\![\pi]\!](I)$.

From (13) by the S5 semantics of $\diamond$ and the semantic clause for **might**, (14).

(14) $\quad [\diamond\varphi]_I \subseteq [\![\mathbf{might}\ \pi]\!](I)$.

From this by set theoretic reasoning and Lemma 1, (15).

(15) $\quad [\diamond\varphi]_I \cap [\psi]_I \subseteq [\psi]_{[\![\mathbf{might}\ \pi]\!](I)}$.

By modal propositional logic, (16):

(16) $\quad [\diamond\varphi \wedge \psi]_I \subseteq [\psi]_{[\![\mathbf{might}\ \pi]\!](I)}$.

Since I was arbitrary, it follows that $\models \langle\diamond\varphi \wedge \psi\rangle$ **might** $\pi\ \langle\psi\rangle$. ∎

8 Completeness of the Calculus

As a preliminary to the proof that the calculus is complete, we first show that statements of the form $\{\mathbf{wdp}(\pi,\psi)\}\ \pi\ \{\psi\}$ and $\langle\mathbf{wcp}(\pi,\psi)\rangle\ \pi\ \langle\psi\rangle$ are theorems of the calculus.

Lemma 18 (wcp/wdp derivability)

1. $\vdash \{\mathbf{wdp}(\pi,\psi\}\ \pi\ \{\psi\}$.

2. $\vdash \langle\mathbf{wcp}(\pi,\psi\rangle\ \pi\ \langle\psi\rangle$.

Proof: Induction on the structure of π, simultaneously for 1 and 2.

For atomic programs, the test axioms immediately give:

(17) $\quad \vdash \{\mathbf{wdp}(\bot,\psi)\}\ \bot\ \{\psi\}$.

(18) $\quad \vdash \langle\mathbf{wcp}(\bot,\psi)\rangle\ \bot\ \langle\psi\rangle$.

(19) $\quad \vdash \{\mathbf{wdp}(p,\psi)\}\ p\ \{\psi\}$.

(20) $\quad \vdash \langle\mathbf{wcp}(p,\psi)\rangle\ p\ \langle\psi\rangle$.

The cases of composition and union are left to the reader.

For negation, conjunctive case, we have to establish $\vdash \langle\mathbf{wcp}(\neg\pi,\psi)\rangle\ \neg\pi\ \langle\psi\rangle$. By the definition of **wcp**, this is equivalent to: $\vdash \langle\psi \downarrow \mathbf{wdp}(\pi,\bot)\rangle\ \neg\pi\ \langle\psi\rangle$. By the induction hypothesis we have:

(21) $\quad \vdash \{\mathbf{wdp}(\pi,\bot)\}\ \pi\ \{\bot\}$.

(22) $\quad \vdash \langle\mathbf{wcp}(\pi,\top)\rangle\ \pi\ \langle\top\rangle$.

Using Definition 11 and the conjunctive consequence rule we get from (22):

(23) $\quad \vdash \langle\neg\mathbf{wdp}(\pi,\bot)\rangle\ \pi\ \langle\top\rangle$.

The first negation rule can now be applied to (21) and (23) to get:

(24) $\quad \vdash \langle\psi \downarrow \mathbf{wdp}(\pi,\bot)\rangle\ \neg\pi\ \langle\psi\rangle$.

For the disjunctive case of negation the reasoning is similar.

For **might**, conjunctive case, we have to establish $\vdash \langle \mathbf{wcp}(\mathbf{might}\ \pi, \psi)\rangle\ \mathbf{might}\ \pi\ \langle\psi\rangle$. By the definition of **wcp**, this is equivalent to: $\vdash \langle \Diamond\mathbf{wcp}(\pi, \top) \wedge \psi\rangle\ \mathbf{might}\ \pi\ \langle\psi\rangle$. By the induction hypothesis we have (22), so the conjunctive rule of **might** gives the desired conclusion. The disjunctive case of **might** is similar. ∎

The completeness theorem follows almost immediately.

Theorem 19 (Completeness) *For all $F \in H_P$: If $\models F$ then $\vdash F$.*

Proof: Assume F is a formula φ from ml_P. Then $\models \varphi$ entails that φ is a theorem of S5, and therefore an axiom of the calculus, so $\vdash \varphi$.

Assume F has the form $\{\varphi\}\ \pi\ \{\psi\}$ and suppose:

(25) $\models \{\varphi\}\ \pi\ \{\psi\}$.

From this and the **wcp/wdp** validity lemma:

(26) $\models \varphi \rightarrow \mathbf{wdp}(\pi, \psi)$.

All S5 validities are axioms, so from (26):

(27) $\vdash \varphi \rightarrow \mathbf{wdp}(\pi, \psi)$.

From the **wcp/wdp** derivability lemma:

(28) $\vdash \{\mathbf{wdp}(\pi, \psi)\}\ \pi\ \{\psi\}$.

From (27) and (28) by the disjunctive consequence rule:

(29) $\vdash \{\varphi\}\ \pi\ \{\psi\}$.

The reasoning for the case of $\langle\varphi\rangle\ \pi\ \langle\psi\rangle$ is similar. ∎

9 Reasoning about Update Logic via S5

Just for the record we mention a fact about update logic which follows immediately from our 'reduction to S5' (but note that this fact was already proved in [10]).

Theorem 20 *Update logic is decidable.*

Proof: The decision problem for update logic is the question: which $\pi \in L_P$ have the property that they are valid (accepted in every input state I)? In other words: which π have the property that for all I it holds that $[\![\pi]\!](I) = I$? The decision procedure for π is as follows. Use the definition of **wcp** to find $\mathbf{wcp}(\pi, \top)$. By Lemma 10 we know that $[\![\pi]\!](I) = [\mathbf{wcp}(\pi, \top)]_I$, so the decision problem for π reduces to the question whether $\mathbf{wcp}(\pi, \top)$ is S5-valid. Use the decision procedure for S5 to settle this question. ∎

In update logic there is a distinction between acceptable and accepted information, witness the following definition.

Definition 14

1. *A program π is accepted in I if $I = [\![\pi]\!](I)$.*

2. *A program π is acceptable in I if $[\![\pi]\!](I) \neq \emptyset$.*

It is the universal version of the first of these which is taken as the notion for universal validity, but one might consider the universal version of the second one just as well.

Definition 15

1. *A program π is always accepted (or valid) if for all I it holds that $[\![\pi]\!](I) = I$.*

2. *A program π is always acceptable if for all $I \neq \emptyset$ it holds that $[\![\pi]\!](I) \neq \emptyset$.*

An obvious question suggests itself: are the notions of being always accepted and being always acceptable equivalent? Using the S5 connection it is easy to see the relation between these two notions. Now we need not concern ourselves with the case of $I = \emptyset$, for the elimination lemma forces $[\![\pi]\!](\emptyset) = \emptyset$ for every π. Thus, there is no harm in adopting the usual convention that S5 models have a non-empty set of worlds. The 'static' version of π *is always accepted* is (30).

(30) $\quad S5 \models \mathbf{wcp}(\pi, \top)$.

The 'static' version for π *is always acceptable*, on the other hand, is (31). Note that this translation hinges on the assumption of non-emptyness of S5 models.

(31) $\quad S5 \models \Diamond\mathbf{wcp}(\pi, \top)$.

So the notion of being always acceptable is decidable as well, but it does not coincide with the notion of being always accepted. Indeed, we have that $S5 \models \varphi$ implies $S5 \models \Diamond\varphi$, because of the reflexivity of accessibility, so (30) implies (31), but not the other way around. Take φ equal to $\Diamond p \rightarrow p$ for a simple counterexample. We have $S5 \not\models \Diamond p \rightarrow p$ (take a non-p world in a model containing both p and non-p worlds), but $S5 \models \Diamond(\Diamond p \rightarrow p)$. To see this latter fact, take an arbitrary w in an arbitrary universal S5 model I. If there are no p worlds, then $I, w \Vdash \Diamond(\Diamond p \rightarrow p)$; if there are p worlds, then there is a p world w' for which $I, w' \models \Diamond p \rightarrow p$, so by the fact that accessibility is universal again $I, w \Vdash \Diamond(\Diamond p \rightarrow p)$. Note, by the way, that $\Diamond(\Diamond p \rightarrow p)$ is the modal counterpart of a predicate logical sentence that philosophical logicians sometimes refer to as 'Plato's principle': $\exists x(\exists x Px \rightarrow Px)$.

The S5 counterexample can be transposed to update logic, of course: $p \cup \neg\mathbf{might}\ p$ is an example of a program which is always acceptable but not always accepted.

For a next illustration of reasoning about update logic via S5 we take a quick look at valid consequence in update logic. In his paper [14] Veltman discusses various notions of valid consequence. He distinguishes the following three definitions.

Definition 16

1. *$\pi_1 \models_1 \pi_2$ if for all I it holds that $[\![\pi_1]\!](I) = I$ implies $[\![\pi_2]\!](I) = I$.*

2. *$\pi_1 \models_2 \pi_2$ if for all I it holds that $[\![\pi_1]\!](I) = [\![\pi_2]\!]([\![\pi_1]\!](I))$.*

3. *$\pi_1 \models_3 \pi_2$ if $[\![\pi_1]\!](W) = [\![\pi_2]\!]([\![\pi_1]\!](W))$.*

The following proposition reduces these notions to S5.

Proposition 21

1. *$\pi_1 \models_1 \pi_2$ iff $S5 \vdash \Box\mathbf{wcp}(\pi_1, \top) \rightarrow \Box\mathbf{wcp}(\pi_2, \top)$.*

2. $\pi_1 \models_2 \pi_2$ *iff* $S5 \vdash \mathbf{wcp}(\pi_1, \top) \leftrightarrow \mathbf{wcp}(\pi_1; \pi_2, \top)$.

3. $\pi_1 \models_3 \pi_2$ *iff* $S5 \vdash (\bigwedge\!\!\!\bigwedge \Diamond \varphi_i) \rightarrow (\mathbf{wcp}(\pi_1, \top) \leftrightarrow \mathbf{wcp}(\pi_1; \pi_2, \top))$, *where the* φ_i *are all conjunctions of the form* $(\neg)p_1 \wedge \cdots \wedge (\neg)p_n$, *with* $p_1, \ldots, p_n$ *the list of proposition letters occurring in* π_1 *or* π_2.

Proof: The first item:

For all I: $[\![\pi_1]\!](I) = I$ implies $[\![\pi_2]\!](I) = I$
iff for all I: if for all $w \in I$: $I, w \Vdash \mathbf{wcp}(\pi_1, \top)$ then for all $w \in I$: $I, w \Vdash \mathbf{wcp}(\pi_2, \top)$
iff for all I: $I \models \Box\mathbf{wcp}(\pi_1, \top) \rightarrow \Box\mathbf{wcp}(\pi_2, \top)$
iff $S5 \vdash \Box\mathbf{wcp}(\pi_1, \top) \rightarrow \Box\mathbf{wcp}(\pi_2, \top)$.

The second item is immediate from the definitions of the validity notions, the **wcp** adequacy lemma and the completeness of S5.

For the third item, note that $I \models \bigwedge\!\!\!\bigwedge \Diamond \varphi_i$ (where $\models$ denotes S5 validity) for precisely those information sets I that express total ignorance with respect to all proposition letters in π_1 and π_2, i.e., for the sets I that are indistinguishable from W as far as π_1 and π_2 are concerned. The S5 formula expresses that for such I, all worlds in $[\![\pi_1; \pi_2]\!](I)$ are worlds in $[\![\pi_2]\!](I)$ and vice versa. This is precisely what the validity notion $\models_3$ expresses. ∎

As Willem Groeneveld pointed out, this reduction to S5 can be simplified somewhat by defining $\mathbf{wcp}(\pi, \top)$ directly, as follows.

Definition 17

1. $\mathbf{wcp}(\bot, \top) = \bot$.
2. $\mathbf{wcp}(p, \top) = p$.
3. $\mathbf{wcp}(\pi_1; \pi_2, \top) = \mathbf{wcp}(\pi_2, \top) \downarrow \mathbf{wcp}(\pi_1, \top)$.
4. $\mathbf{wcp}(\pi_1 \cup \pi_2, \top) = \mathbf{wcp}(\pi_1, \top) \vee \mathbf{wcp}(\pi_2, \top)$.
5. $\mathbf{wcp}(\neg\pi, \top) = \neg\mathbf{wcp}(\pi, \top)$.
6. $\mathbf{wcp}(\mathbf{might}\ \pi, \top) = \Diamond\mathbf{wcp}(\pi, \top)$.

10 Calculations of Consistency

Veltman calls a program π of L_P consistent if there is some information state I for which $[\![\pi]\!](I) \neq \emptyset$. Intuitively, consistent programs are programs that can be used to convey information. By the soundness of the Hoare calculus, consistency of an update program π boils down to the question whether there is some S5 consistent φ such that $\vdash \langle\varphi\rangle\ \pi\ \langle\top\rangle$. We illustrate how to check consistency for two examples taken from Veltman [14]. We calculate with WCPs, but by virtue of the fact that WCP reasoning and NSC reasoning are equivalent (Theorem 15), calculations with NSCs work just as well.

Example 1 **might** $p; \neg p$ *is consistent.*

Proof:

$$\begin{aligned}\mathbf{wcp}(\mathbf{might}\ p; \neg p\ , \top) &= \mathbf{wcp}(\mathbf{might}\ p; \mathbf{wcp}(\neg p\ , \top))\\ &= \mathbf{wcp}(\mathbf{might}\ p; \neg\mathbf{wcp}(p, \top))\\ &= \mathbf{wcp}(\mathbf{might}\ p; \neg p)\\ &= \Diamond p \wedge \neg p\end{aligned}$$

Since $\Diamond p \wedge \neg p$ does have S5 models, so it is not S5-provably equivalent to $\perp$. ■

Example 2 $\neg p$; **might** p *is not consistent.*

Proof:

$$\begin{aligned}\mathbf{wcp}(\neg p\ ;\ \mathbf{might}\ p, \top) &= \mathbf{wcp}(\neg p; \mathbf{wcp}(\mathbf{might}\ p, \top))\\ &= \mathbf{wcp}(\neg p, \Diamond\mathbf{wcp}(p, \top))\\ &= \mathbf{wcp}(\neg p, \Diamond p)\\ &= \Diamond p \downarrow \neg\mathbf{wcp}(p, \top)\\ &= \Diamond p \downarrow \neg p\\ &= \neg p \wedge \Diamond(p \wedge (p \downarrow \neg p))\\ &= \neg p \wedge \Diamond(p \wedge \neg p))\\ &= \neg p \wedge \perp\\ &= \perp\end{aligned}$$

■

It is a fact about the Hoare calculus that it yields weakest preconditions, provided that the consequence rules are not used for precondition strengthening. For good measure, we also give the Hoare deductions of weakest preconditions for the two examples.

$$\frac{\langle \Diamond p \wedge \neg p\rangle\ \mathbf{might}\ p\ \langle \neg p\rangle \qquad \langle \neg p\rangle\ \neg p\ \langle \top\rangle}{\langle \Diamond p \wedge \neg p\rangle\ \mathbf{might}\ p;\ \neg p\ \langle \top\rangle.}$$

$$\frac{\langle \Diamond p \downarrow \neg p\rangle\ \neg p\ \langle \Diamond p\rangle \qquad \langle \Diamond p\rangle\ \mathbf{might}\ p\ \langle \top\rangle}{\langle \Diamond p \downarrow \neg p\rangle\ \neg p;\ \mathbf{might}\ p\ \langle \top\rangle.}$$

11 Conclusion

By our construction of a calculus for update logic we claim to have established a clean connection between a species of dynamic logic and good old static S5. There is scope for quite a bit of further work. In the first place, one might wish to consider extensions of update logic with new operations, such as 'downdating' (retracting information), and study how these could be linked to static modal notions. Also, a modal study of the preference relation on information states that Veltman proposes seems to be worthwhile: this would lead to a link to a trimodal system with one modal operator $\boxed{1}$ reflecting the dynamics of discarding possibilities, a modal operator $\boxed{2}$ interpreted in terms of the preference order on the set of all worlds (the 'normally' relation), and finally a modal operator $\boxed{3}$ interpreted in terms of the preference relation restricted to the current input information set (the 'presumably' relation). In a different direction, one may study the combination of the calculus given here with the calculus from [13] in a system of quantificational update logic satisfying the desiderata which Groenendijk and Stokhof list in [3]. Giving a Hoare style axiomatisation of a system of dynamic assignment logic with epistemic modalities, using modal predicate logic as the assertion language, is a feasible task now, witness Van Eijck and Cepparello [12].

Acknowledgement

This paper has benefited from helpful comments by Johan van Benthem, Tim Fernando, Willem Groeneveld, Marcus Kracht, Frank Veltman, Kees Vermeulen, Jørgen Villadsen and Albert Visser.

References

[1] J. Barwise. Noun phrases, generalized quantifiers and anaphora. In P. Gärdenfors, editor, *Generalized Quantifiers: linguistic and logical approaches*, pages 1–30. D. Reidel Publishing Company, Dordrecht, 1987.

[2] R. Goldblatt. *Logics of Time and Computation*, volume 7 of *CSLI Lecture Notes*. CSLI, Stanford, 1987. Distributed by University of Chicago Press.

[3] J. Groenendijk and M. Stokhof. Two theories of dynamic semantics. In J. van Eijck, editor, *Logics in AI—European Workshop JELIA '90*, pages 55–64, Berlin, 1991. Springer Lecture Notes in Artificial Intelligence.

[4] I. Heim. *The Semantics of Definite and Indefinite Noun Phrases*. PhD thesis, University of Massachusetts, Amherst, 1982.

[5] C.A.R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):567–580, 583, 1969.

[6] H. Kamp. A theory of truth and semantic representation. In J. Groenendijk et al., editors, *Formal Methods in the Study of Language*. Mathematisch Centrum, 1981.

[7] L. Karttunen. Discourse referents. In J. McCawley, editor, *Syntax and Semantics 7*, pages 363–385. Academic Press, 1976.

[8] M. Kracht. Splittings and the finite model property. Logic group preprint series, Department of Philosophy, University of Utrecht, 1991. To appear in the JSL.

[9] R. Stalnaker. Pragmatics. In D. Davidson and G. Harman, editors, *Semantics of Natural Language*, pages 380–397. Reidel, 1972.

[10] J. van Benthem. Semantic parallels in natural language and computation. In H.-D. Ebbinghaus et al., editors, *Logic Colloquium, Granada, 1987*, pages 331–375, Amsterdam, 1989. Elsevier.

[11] J. van Benthem. General dynamics. Technical Report LP-90-11, ITLI, University of Amsterdam, 1990.

[12] J. van Eijck and G. Cepparello. Dynamic modal predicate logic. Manuscript, CWI, Amsterdam, 1992.

[13] J. van Eijck and F.J. de Vries. Dynamic interpretation and Hoare deduction. Technical Report CS-R9115, CWI, Amsterdam, 1991. To appear in the Journal of Logic, Language and Information.

[14] F. Veltman. Defaults in update semantics. Manuscript, Department of Philosophy, University of Amsterdam, 1991. To appear in the Journal of Philosophical Logic.