UvA-DARE (Digital Academic Repository)

Dynamic Relation Logic Is the Logic of DPL-Relations

Visser, A.

DOI

[10.1023/A:1008228122853](https://doi.org/10.1023/A:1008228122853)

Publication date

1997

Document Version

Final published version

Published in

Journal of Logic, Language and Information

[Link to publication](#)

Citation for published version (APA):

Visser, A. (1997). Dynamic Relation Logic Is the Logic of DPL-Relations. *Journal of Logic, Language and Information*, 6(4), 441-452. <https://doi.org/10.1023/A:1008228122853>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Dynamic Relation Logic Is the Logic of DPL-Relations

ALBERT VISSER

Department of Philosophy, Utrecht University, Heidelberglaan 8, 3584 CS Utrecht, the Netherlands
E-mail: albert.visser@phil.ruu.nl

(Received 20 January 1997)

Abstract. In this paper we prove that the principles in the language with relation composition and dynamic implication, valid for all binary relations, are the same ones as the principles valid when we restrict ourselves to DPL-relations, i.e. relations generated from conditions (tests) and resettings.

Key words: dynamics, Predicate Logic, Relation Algebra, relation, resetting, bisimulation

1. Introduction

In studying a complex language it is often useful to abstract away from some carefully selected details and to try to characterize its valid “schemes.” A scheme can be considered as a formula of a simpler language. The scheme is valid if all substitutions of formulas of the original language for the “schematic variables” of the simpler language are valid. In this paper we will study the complex language of Dynamic Predicate Logic, DPL, a variant of Predicate Logic invented by Groenendijk and Stokhof (1991). Our simple language will be a relational analogue of the language of Propositional Logic: the language of Dynamic Relation Logic, DRL. We will show that the principles valid for all DPL-substitutions are precisely the principles valid for the intended interpretation of DRL. One variant of our result provides a positive answer to a conjecture of van Benthem and Ceparrello (1994: conjecture 9).

We briefly describe DRL. Consider a domain, i.e. a non-empty set of objects D . Consider some set $\mathcal{R}$ of binary relations over D . There are two operations on relations, we want to focus on. The first is ordinary relation composition and the second is dynamic implication (to be defined below). We close our relations under these operations. The relations so obtained, together with the operations form a structure. Here we count the relations of our original set $\mathcal{R}$ as “constants.” We call such structures DR-models. The first question to ask is: *which sequents and identities are valid for DR-models?* This question was solved by Hollenberg for identities. See his paper (Hollenberg, 1997), which axiomatizes the identities valid in general DR-structures.

Now, suppose our domains have a more specific character: suppose they are sets of assignments, i.e. $D = E^{\text{Var}}$, where E is some non-empty domain and Var is some set of variables. Moreover, let us restrict our set of basic relations to relations that are generated by composition from *finitely restricted tests* (or: *finitely restricted conditions*) and *random resettings* of single variables. Here a *test* or *condition* is a subrelation C of the identity relation – in other words: a relation without “dynamic powers.” A test is *finitely restricted* if there is a finite set of variables W such that:

$$f C f \text{ and } f \upharpoonright W = g \upharpoonright W \Rightarrow g C g.$$

We often confuse C with the set $\{g \in D^{\text{Var}} \mid g C g\}$. Thus, the conditions represent classical sets of assignments in the dynamic setting. Resettings of variables are relations of the form $[\exists x]$, where

$$f[\exists x]g : \Leftrightarrow f \text{ and } g \text{ coincide on all variables except, possibly, on } x.$$

The relations thus generated by tests and resettings are DPL-relations, i.e. relations typically definable in the language of DPL, a variant of predicate logic invented by Groenendijk and Stokhof (1991). Our question is: *which DR-sequents and identities are valid for the restricted set of relations?*

In this paper we will make this question precise and prove that the same sequents and identities are valid both for DR-structures in general and for DPL-models.

Here is a sketch of the argument. One first shows that if two structures are bisimilar, then they satisfy the same sequents. This is immediate by the well known fact that composition and dynamic implication are *safe* for bisimulation. Then one shows that for every structure $\mathfrak{N}$, there is a structure $\mathfrak{D}$ which is bisimilar to $\mathfrak{N}$ and in which every relation can be generated from conditions and resettings. The domain of $\mathfrak{D}$ is the cartesian product of the domain of $\mathfrak{N}$ with itself. A relation R of $\mathfrak{N}$ is mapped to a relation $\hat{R}$ of $\mathfrak{D}$, where:

$$\langle a, b \rangle \hat{R} \langle c, d \rangle : \Leftrightarrow a R c \text{ and } c = d.$$

These relations $\hat{R}$ can be specified in the DPL-language by:

$$\exists y. R(x, y). \exists x. (x = y).$$

This formula means: reset y arbitrarily and then test whether it gives an appropriate output for R . If so, reset x to the value of y . The presence of this formula shows that DPL has the resources to transform a *static* representation of a relation to a *dynamic* one. To do this we need the *auxiliary variable* y . We will see that this is unavoidable. It is typical for DPL that it does not have truly *local variables*, i.e. variables that can be used for temporary storage and then can be restored to their original content – unless we use some other variable to store this content, but then this variable

The work in the present paper illustrates the usefulness of notions like bisimulation and safety in thinking about relational languages. Moreover, the paper shows

one way to think about the use of *auxiliary variables in DPL*: they are things to be divided out modulo bisimulation.

2. Dynamic Relation Logic and Dynamic Relation Algebra

We first introduce the necessary relational notions.

DEFINITION 2.1. Let X be any non-empty set.

1. $Rel(X)$ is the set of binary relations on X , i.e. $Rel(X) := \wp(X \times X)$
2. Let $R, S \in Rel(X)$. The composition $R \circ S$ of R and S is defined by:

$$x(R \circ S)y :\Leftrightarrow \exists z \, xRzSy$$

3. Let $R, S \in Rel(X)$. The dynamic implication $(R \rightarrow S)$ between R and S is defined by:

$$x(R \rightarrow S)y :\Leftrightarrow x = y \text{ and } \forall z (xRz \Rightarrow \exists u \, zSu).$$

Our use of $\rightarrow$ here overloads the symbol, since we also use it for implication in the object language.

4. id_X is the identity relation. A relation R is a *condition* if $R \subseteq \text{id}_X$.

□

The notion of dynamic implication was first introduced by Kamp in his pioneering paper (1981).

DEFINITION 2.2. We define the language of *Dynamic Relation Logic/Algebra*, DRL/A, as follows.

1. A DR-language $\mathcal{K}$ is a structure $\langle \text{Rels} \rangle$, where Rels is a, possibly empty, set of atomic relation symbols.
2. The set of $\mathcal{K}$ -formulas, $\text{For}_{\mathcal{K}}$, is the smallest set such that:
 - $\Leftrightarrow R$ is in $\text{For}_{\mathcal{K}}$, for $R \in \text{Rels}$
 - $\Leftrightarrow \perp, \delta$ are in $\text{For}_{\mathcal{K}}$
 - $\Leftrightarrow$ If $\phi, \psi \in \text{For}_{\mathcal{K}}$, then so are $\phi.\psi$ and $(\phi \rightarrow \psi)$.*
 - We use $\neg(\phi)$ as abbreviation of $(\phi \rightarrow \perp)$.
3. A DR-model $\mathfrak{M}$ for a DR-language $\mathcal{K}$ is a structure $\langle D, I \rangle$, where D is a non-empty set, the *domain* of $\mathfrak{M}$; I is a function which assigns to each atomic relation symbol R of $\text{Rels}_{\mathcal{K}}$ a binary relation on D .

* I feel that it is more faithful to the semantics to leave out the brackets in the formation rule for the dot *officially*, but nothing important hangs on this choice in this paper. We get an ambiguous syntax, but still unique meanings, since the operation of composition is associative.

4. Let a DR-model $\mathfrak{M}$ for a DR-language $\mathcal{K}$ be given. We define an interpretation function $[\cdot]_{\mathfrak{M}} : \text{For}_{\mathcal{K}} \rightarrow \text{Rel}(D)$ as follows*: $[R] := I(R)$, $[\perp] := \emptyset$, $[\delta] := \text{id}_D$, $[\phi.\psi] := [\phi] \circ [\psi]$, $[(\phi \rightarrow \psi)] := ([\phi] \rightarrow [\psi])$.
5. We define validity in DR by:

$$\phi \models_{\mathfrak{M}} \psi \Leftrightarrow \forall f, g (f[\phi]_{\mathfrak{M}} g \Rightarrow \exists h g[\psi]_{\mathfrak{M}} h).$$

Let $\phi \models \psi$ iff $\phi \models_{\mathfrak{M}} \psi$ for all models $\mathfrak{M}$ for the appropriate language.

6. We write $\phi =_{\mathfrak{M}} \psi$ for: $[\phi]_{\mathfrak{M}} = [\psi]_{\mathfrak{M}}$, and $\phi = \psi$ for: for all $\mathfrak{M}$: $\phi =_{\mathfrak{M}} \psi$.

□

We will speak of *Dynamic Relation Logic*, DRL, when we are thinking about the validity of sequents and about Dynamic Relation Algebra, DRA, when we are thinking about the validity of identities. We discuss the relationship between $\models$ and $=$ in the appendix. The valid identities of DRA are axiomatized by Hollenberg (1997). The valid sequents of the $\rightarrow$ -fragment of DRL are axiomatized by Blackburn and Venema (1995).

3. Dynamic Predicate Logic and Dynamic Predicate Algebra

In this section we introduce the language and model theory of DPL/A

DEFINITION 3.1.

1. A DPL-language $\mathcal{L}$ is a structure $\langle \text{Pred}, \text{Ar}, \text{Var} \rangle$, where Pred is a set of predicate symbols; Ar is a function from Pred to the natural numbers (including 0); Var is a, possibly empty, set of variables. We assume that $=$ is in Pred and that $\text{Ar}(=) = 2$.**
2. Tests is the set of $P(v_1, \dots, v_n)$, for $P \in \text{Pred}$ with $\text{Ar}(P) = n$ and $v_1, \dots, v_n \in \text{Var}$.
3. Resets is the set of $\exists v$, for $v \in \text{Var}$
4. The formulas $\text{For}_{\mathcal{L}}$ of $\mathcal{L}$ are the formulas of the DR-language $\mathcal{K} = \mathcal{K}(\mathcal{L})$, that we get by taking $\text{Rels}_{\mathcal{K}} := \text{Tests}_{\mathcal{L}} \cup \text{Resets}_{\mathcal{L}}$. We will call $\mathcal{K}$ the *associated DR-language*. We use $\forall v(\phi)$ as abbreviations of $(\exists v \rightarrow \phi)$. If v and w are distinct variables, we write $[v := w]$ for: $\exists v.v = w$. An alternative notation for $\exists v$, is $[v := ?]$ (random reset).
5. A DPL-model $\mathfrak{M}$ for a DPL-language $\mathcal{L}$ is a structure $\langle D, I \rangle$, where D is a non-empty set, the *domain* of $\mathfrak{M}$; I is a function which assigns to each predicate symbol P of $\text{Pred}_{\mathcal{L}}$ an $\text{Ar}(P)$ -ary relation on D . We stipulate that $I(=) := \text{id}_D$.
6. Let a DPL-model $\mathfrak{M}$ be given. $\text{Ass}_{\mathfrak{M}}$, the set of *assignments* for $\mathfrak{M}$, is D^{Var} . The interpretation function associated to $\mathfrak{M}$ is the interpretation function of the DR-model $\mathfrak{M} = \mathfrak{M}(\mathfrak{M}) = \langle \text{Ass}_{\mathfrak{M}}, J \rangle$, where

* We suppress the subscripts if no confusion is possible.

** I plead guilty of cheating here. “=” should evidently be a *logical constant*. We can, however, economise on clauses by putting it in Pred .

$$f J(P(v_1, \dots, v_n)) g :\Leftrightarrow f = g \text{ and } \langle f(v_1), \dots, f(v_n) \rangle \in I(P)$$

and

$$f J(\exists v) g :\Leftrightarrow \forall w \in \text{Var} \setminus \{v\} f(w) = g(w).$$

We will call $\mathfrak{N}$ the *associated DR-model*. We write $[\cdot]_{\mathfrak{M}}$ for $[\cdot]_{\mathfrak{N}}$.

7. The notion of satisfaction for a DPL-model $\mathfrak{M}$ is the one of its associated DR-model $\mathfrak{N}$. We write $\models_{\mathfrak{M}}$ for $\models_{\mathfrak{N}}$. $\phi \models \psi$ if $\phi \models_{\mathfrak{M}} \psi$ for all DPL-models $\mathfrak{M}$ for the appropriate language. Identity of meanings is treated similarly.
8. A binary relation R is *definable* in a DPL-model $\mathfrak{M}$ for a language $\mathcal{L}$ if there is an $\mathcal{L}$ -formula ϕ , which defines R , i.e. $R = [\phi]_{\mathfrak{M}}$.

□

We could extend the DPL-language with function symbols by copying the way this is done in ordinary predicate logic. However, for the kind of result we are after such an extension is immaterial, since the usual trick to eliminate function symbols works also in DPL – with a small twist. E.g. $P(f(g(x)))$ will be translated to: $\neg \neg (\exists u. G(x, u). \exists v. F(u, v). P(v))$.

Example 3.2 (with Kees Vermeulen). We show that in *no* DPL-model with at least two elements in D can we define the relation $\neq$ on $D^{\text{Var}} \times D^{\text{Var}}$ as a resetting relation. Suppose we could. Say, ϕ is the defining formula. Since $\neq$ is not a condition ϕ must contain at least one $\exists v$ (not confined in an implication). Let $\exists x$ be the first such. So, ϕ has the form $C. \exists x. \chi$, where C stands for a condition. Since every f can be successfully continued via $\neq$, no f is weeded out in advance. Hence, C must be equivalent to δ . So we may write ϕ as $\exists x. \chi$. Consider any f and g with $f \neq g$ and $f[\exists x]g$. (Since our domain has at least two elements such f and g exist.) We have $f[\exists x. \chi]g$. But, then, we also have $g[\exists x. \chi]g$, contradicting the assumption that our formula defines $\neq$. (For a more extensive discussion of the question which relations are DPL-definable see Visser, 1995.)

□

A typical feature of DPL is that the predicate symbols do not take all possible DPL-meanings as possible values. As a consequence, we do not generally have: $\phi \models \psi \Rightarrow \phi[P := \chi] \models \psi[P := \chi]$, where P is a predicate symbol and where no free variables in χ are bound in ϕ, ψ . The simplest example is as follows. We *do* have $P \models P$, but *not* $Qx. \exists x. \neg Qx \models Qx. \exists x. \neg Qx$.

We now introduce the schematic way of looking at DPL/A.

DEFINITION 3.3.

1. Consider a DR-model $\mathfrak{N} = \langle D, I \rangle$. $\mathfrak{N}$ is **DPL-definable**, if there is a DPL-model $\mathfrak{M}$, such that D is (isomorphic to) $Ass_{\mathfrak{M}}$ and each $I(R)$ is definable in $\mathfrak{M}$.

2. $\phi \models_{dpl} \psi$ iff $\phi \models_{\mathfrak{N}} \psi$ for all DPL-definable models $\mathfrak{N}$, appropriate for the given language.
3. $\phi =_{dpl} \psi$ iff $\phi =_{\mathfrak{N}} \psi$ for all DPL-definable models $\mathfrak{N}$, appropriate for the given language.

□

A DPL-definable DR-model $\mathfrak{N}$ is just a DR-model in which the atomic relations can be generated from conditions and resettings. Note that it follows that *all* relations of $\mathfrak{N}$ can be generated from conditions and resettings.

Finally, we have reached the stage, where we can officially state our problem:

Do we have $\models_{dpl} = \models$ for DR? Do we have: $=_{dpl} = =$ for DR?

We end this section by describing an equivalent way of introducing $\models_{dpl}$. Let a DR-language $\mathcal{K}$ be given. A translation $(.)^*$ from $\mathcal{K}$ to a DPL-language $\mathcal{L}$ is a function from the formulas of $\mathcal{K}$ to the formulas of $\mathcal{L}$, such that $(.)^*$ commutes with $\perp$, δ , $.$ and $\rightarrow$. We have for formulas ϕ, ψ in $\mathcal{K}$: $\phi \models_{dpl} \psi \Leftrightarrow$ for all $(.)^*$ (with domain $\mathcal{K}$) $\phi^* \models \psi^*$.

4. Transition Systems

To solve our problem we will need some facts about transition systems and bisimulations. These are presented in the present section.

DEFINITION 4.1.

1. A transition system $\mathfrak{G}$ is a structure $\langle S, A, \Leftrightarrow \rangle$. Here, S is a non-empty set, the set of *states*. A is a (possibly empty) set, the set of *labels*. $\Leftrightarrow$ is a function from A to the binary relations on S . We write $s \xrightarrow{a} t$ for: $\langle s, t \rangle \in \Leftrightarrow(a)$. Note that our transition systems are “unrooted.”
2. Consider two transition systems $\mathfrak{G}$ and $\mathfrak{H}$. Suppose $A_{\mathfrak{G}} = A_{\mathfrak{H}}$. A *bisimulation* $\mathcal{B}$ between $\mathfrak{G}$ and $\mathfrak{H}$ is a relation between $S_{\mathfrak{G}}$ and $S_{\mathfrak{H}}$, such that whenever $s \mathcal{B} s'$ and $s \xrightarrow{a}_{\mathfrak{G}} t$, there is a t' with $t \mathcal{B} t'$ and $s' \xrightarrow{a}_{\mathfrak{H}} t'$ (*the zig property*) and such that whenever $s \mathcal{B} s'$ and $s' \xrightarrow{a}_{\mathfrak{H}} t'$, there is a t with $t \mathcal{B} t'$ and $s \xrightarrow{a}_{\mathfrak{G}} t$ (*the zag property*). $\mathcal{B}$ is *total* if $\text{dom}(\mathcal{B}) = S_{\mathfrak{G}}$. $\mathcal{B}$ is *surjective* if $\text{range}(\mathcal{B}) = S_{\mathfrak{H}}$. $\mathcal{B}$ is *full* if $\mathcal{B}$ is total and surjective.
3. Let $\mathfrak{G}$ and $\mathfrak{H}$ have the same set of labels. Define:
 - $\Leftrightarrow \mathcal{B} : \mathfrak{G} \preceq \mathfrak{H} :\Leftrightarrow \mathcal{B}$ is a total bisimulation between $\mathfrak{G}$ and $\mathfrak{H}$
 - $\Leftrightarrow \mathfrak{G} \preceq \mathfrak{H} \Leftrightarrow$ for some $\mathcal{B} \mathcal{B} : \mathfrak{G} \preceq \mathfrak{H}$
 - $\Leftrightarrow \mathfrak{G} \equiv \mathfrak{H} \Leftrightarrow \mathfrak{G} \preceq \mathfrak{H}$ and $\mathfrak{H} \preceq \mathfrak{G}$

□

It is easy to check that – for transition systems on a fixed set of labels – $\preceq$ is a partial pre-ordering. Bisimulations are closed under union and converse.

So when $\mathfrak{G} \equiv \mathfrak{H}$, then there is a full bisimulation between $\mathfrak{G}$ and $\mathfrak{H}$.

Fix a DR-language $\mathcal{K}$. Consider a DR model $\mathfrak{N}$. We associate two transition systems $\mathfrak{G} := T_0(\mathfrak{N})$ and $\mathfrak{H} := T_1(\mathfrak{N})$ to $\mathfrak{N}$ by taking:

$$\begin{aligned} \Leftrightarrow S_{\mathfrak{G}} &:= S_{\mathfrak{H}} := D_{\mathfrak{N}} \\ \Leftrightarrow A_{\mathfrak{G}} &:= \text{Rels}_{\mathcal{K}}, A_{\mathfrak{H}} := \text{For}_{\mathcal{K}} \\ \Leftrightarrow \Leftrightarrow_{\mathfrak{G}} &:= I_{\mathfrak{N}}, \Leftrightarrow_{\mathfrak{H}} := [\cdot]_{\mathfrak{N}} \end{aligned}$$

We state some simple facts.

THEOREM 4.2. *Consider two $\mathcal{K}$ -models $\mathfrak{N}$ and $\mathfrak{D}$. Suppose $T_0(\mathfrak{N}) \preceq T_0(\mathfrak{D})$. Then, $T_1(\mathfrak{N}) \preceq T_1(\mathfrak{D})$.*

Proof. The verification is immediate. The basic insight is that “adding the identity transition,” “adding the empty transition,” composition and dynamic implication are *safe* for bisimulations. (The fact that, e.g., $\circ$ is safe, means that if $\mathcal{B}$ is a bisimulation for the transitions α and β , then it also a bisimulation for $\alpha \circ \beta$.) Safety is studied and characterized in (van Benthem, 1993) and (Hollenberg, 1995). $\square$

We write $\mathfrak{N} \preceq \mathfrak{D}$ for: $T_0(\mathfrak{N}) \preceq T_0(\mathfrak{D})$.

THEOREM 4.3. *Consider two $\mathcal{K}$ -models $\mathfrak{N}$ and $\mathfrak{D}$. Suppose $\mathfrak{N} \preceq \mathfrak{D}$. Then, $\models_{\mathfrak{D}} \subseteq \models_{\mathfrak{N}}$.*

Proof. Let $\mathcal{B}$ witness $\mathfrak{N} \preceq \mathfrak{D}$. Note that we have: $\mathcal{B} : T_1(\mathfrak{N}) \preceq T_1(\mathfrak{D})$. Suppose $\phi \models_{\mathfrak{D}} \psi$ and $d[\phi]_{\mathfrak{N}} e$. Since $\mathcal{B}$ is a total bisimulation, we can find d', e' with: $d\mathcal{B}d'$, $e\mathcal{B}e'$ and $d'[\phi]_{\mathfrak{D}} e'$. Since $\phi \models_{\mathfrak{D}} \psi$, there is an f' with $e'[\psi]_{\mathfrak{D}} f'$. We have $e\mathcal{B}e'$ and $e'[\psi]_{\mathfrak{D}} f'$, so, by the zag property, there is an f with $f\mathcal{B}f'$ and $e[\psi]_{\mathfrak{N}} f$. $\square$

5. The Problem Solved

We show that $\models_{dpl} = \models$ and that $=_{dpl} = =$, i.o.w., DRA is precisely the algebra of DPL-definable relations. We show that for every $\mathcal{K}$ -model $\mathfrak{N}$ there is a DPL-definable model $\mathfrak{D}$ such that $\mathfrak{N} \equiv \mathfrak{D}$. By Theorem 4.3 it is immediate that $\models_{dpl} = \models$. The result on identity will follow by inspecting the bisimulation between $\mathfrak{N}$ and $\mathfrak{D}$.

Consider a DR-model $\mathfrak{N}$ for $\mathcal{K}$. We write $R_{\mathfrak{N}}$ for $I_{\mathfrak{N}}(R)$, etc. We construct $\mathfrak{D}$ as follows:

$$\begin{aligned} \Leftrightarrow D_{\mathfrak{D}} &:= D_{\mathfrak{N}} \times D_{\mathfrak{N}}. \text{ We conveniently “identify” } D_{\mathfrak{D}} \text{ with } D_{\mathfrak{N}}^{\{x,y\}}, \text{ where } x \text{ and } \\ &\quad y \text{ are distinct variables.} \\ \Leftrightarrow R_{\mathfrak{D}} &:= \widehat{R_{\mathfrak{N}}}, \text{ where } \langle a, b \rangle \widehat{R_{\mathfrak{N}}} \langle c, d \rangle : \Leftrightarrow aR_{\mathfrak{N}}c \text{ and } c = d. \end{aligned}$$

Let $a\mathcal{B}\langle b, c \rangle : \Leftrightarrow a = b$. Clearly $\mathcal{B}$ is a full bisimulation witnessing $\mathfrak{N} \equiv \mathfrak{D}$. Finally, we show that $\mathfrak{D}$ is DPL-definable. Consider the DPL-language $\mathcal{L}$ and the $\mathcal{L}$ -model $\mathfrak{M}$ given by:

$$\begin{aligned} \Leftrightarrow \text{Pred}_{\mathcal{L}} &:= \text{Rels}_{\mathcal{K}} \\ \Leftrightarrow \text{Ar}_{\mathcal{L}}(R) &:= 2 \text{ for all } R \in \text{Pred}_{\mathcal{L}} \end{aligned}$$

$$\begin{aligned}
&\Leftrightarrow \text{Var}_{\mathcal{L}} = \{x, y\} \\
&\Leftrightarrow D_{\mathfrak{M}} := D_{\mathfrak{N}} \\
&\Leftrightarrow I_{\mathfrak{M}}(R) := I_{\mathfrak{N}}(R).
\end{aligned}$$

Note the crucial subtlety that R stands for a *set of pairs* in $\mathfrak{M}$, but for a resetting relation in $\mathfrak{N}$. We can now define the resetting relation $R_{\mathfrak{D}}$ in $\mathfrak{M}$, by: $\exists y.R(x, y). \exists x.x = y$, or, using an abbreviation: $\exists y.R(x, y).[x := y]$. We leave the easy check that this formula does deliver the promised goods to the reader.*

Note that $\mathcal{B}$ is an inverse p-morphism, i.o.w. its converse is functional. Thus, if $\phi =_{\mathfrak{D}} \psi$, then also $\phi =_{\mathfrak{N}} \psi$. Thus, whenever ϕ and ψ define different relations in $\mathfrak{N}$, they will also define different relations in $\mathfrak{D}$. Thus it follows that: $\phi =_{\text{dpl}} \psi \Rightarrow \phi = \psi$. The converse is trivial.

Our formula $\exists y.R(x, y). \exists x.x = y$ effects a *static-dynamic conversion*. It does this by employing an auxiliary variable y . Note that the original contents stored under y are irrevocably lost. We proceed to illustrate, that the use of an auxiliary variable cannot be avoided. We have considered the DPL-definable relations. This class can be refined in a natural way by counting the number of variables in the DPL-language we are considering. Thus we can talk about the DPL-1-definable relations (*1dpl*), the DPL- $\aleph_0$ -definable relations ($\aleph_0\text{dpl}$), etc. Clearly, we have shown that: $\models_{2\text{dpl}} = \models$. The result is preserved if we look at numbers bigger than 2, since for any $n \geq 2$: $\models \subseteq \models_{n\text{dpl}} \subseteq \models_{2\text{dpl}} \subseteq \models$. We do not have $\models_{1\text{dpl}} = \models$, as is shown by the following theorem.

THEOREM 5.1. *We have $\neg\neg(\phi.\phi) \models_{1\text{dpl}} \phi.\phi.\phi$, but not $\neg\neg(\phi.\phi) \models \phi.\phi.\phi$.*

Proof. To prove the first half of the result, we work in a language with one variable x and in a model for this language. Suppose ψ stands for a condition. By a remark of Kees Vermeulen, we have: $[\exists x.\psi.\exists x] = [\neg\neg(\exists x.\psi).\exists x]$. It follows that ϕ can – *salva significatione* – be rewritten as $C.\exists x.C'$, where C and C' stand for conditions. Now it is easy to see that both $\phi.\phi$ and $\phi.\phi.\phi$ are equivalent to $\phi.\neg\neg(\exists x.C'.C)$. From this insight, the first half of our result is immediate. The second half is proved by considering the countermodel: $\bullet \xleftrightarrow{\phi} \bullet \xleftrightarrow{\phi} \bullet$. $\square$

Our result illustrates that *modulo bisimulation* is a good way of thinking about *auxiliary variables* in DPL.

6. Extensions of the Language

In this section we will consider several ways of extending the DR/DPL language, preserving our result.

* An alternative route would have been to take: $\langle a, b \rangle \widehat{R_{\mathfrak{N}}} \langle c, d \rangle : \Leftrightarrow aR_{\mathfrak{N}}c$ and define this relation by: $\exists y.R(x, y). \exists x.x = y.\exists y$.

6.1. SAFE EXTENSIONS

Our result still works for any extension of DR with bisimulation safe operations. Thus we could extend our language with $\vee$, interpreted as $\cup$ (union of relations, indeterministic choice). The operations that are safe for bisimulations were characterized by van Benthem (1993). We have a bit more here, since *full* bisimulation is the relevant notion. For example, “adding the universal relation” is safe for full bisimulation, but not for (partial) bisimulation.* So we could add $\top$, where $\top$ stands for the universal relation, to the DR-language, preserving our result.

6.2. EXTENSION WITH VARIABLES FOR TESTS

We can enrich the DR-language with variables $P, Q, \dots$ for tests. The model $\mathfrak{D}$ of the previous section has to be extended as follows: $P_{\mathfrak{D}} := \widehat{P_{\mathfrak{N}}}$, where $\langle a, b \rangle \widehat{P_{\mathfrak{N}}} \langle c, d \rangle : \Leftrightarrow a \in P_{\mathfrak{N}}$ and $a = c$ and $b = d$. The DPL-language $\mathcal{L}$ we have been considering and the $\mathcal{L}$ -model $\mathfrak{M}$ are given by:

$$\begin{aligned} &\Leftrightarrow \text{Pred}_{\mathcal{L}} := \text{Rels}_{\mathcal{K}} \cup \text{Tests}_{\mathcal{K}} \\ &\Leftrightarrow \text{Ar}_{\mathcal{L}}(R) := 2, \text{ for } R \in \text{Rels}, \text{Ar}_{\mathcal{L}}(P) := 1, \text{ for } P \in \text{Tests} \\ &\Leftrightarrow \text{Var}_{\mathcal{L}} = \{x, y\} \\ &\Leftrightarrow D_{\mathfrak{M}} := D_{\mathfrak{N}} \\ &\Leftrightarrow I_{\mathfrak{M}}(R) := I_{\mathfrak{N}}(R), I_{\mathfrak{M}}(P) := I_{\mathfrak{N}}(P). \end{aligned}$$

We can now define the test $P_{\mathfrak{D}}$ in $\mathfrak{M}$, by: $P(x)$. Note that in $\mathfrak{N}$ the variables R and P might have been assigned the same relation, but that still not generally $R_{\mathfrak{D}} = P_{\mathfrak{D}}$.

6.3. ADDING CONJUNCTION

We can add $\wedge$ to the DR/DPL-languages, interpreting it as $\cap$. Since intersection is not safe for bisimulation, our earlier result does not extend automatically. However, the language with $\wedge$ is more expressive and, in it, we can implement a way of setting back the auxiliary variable employed in our argument to its original value.

Consider a DR-model $\mathfrak{N}$ for $\mathcal{K}(\wedge)$. We construct a new $\mathfrak{D}$ as follows:

$$\begin{aligned} &\Leftrightarrow D_{\mathfrak{D}} := D_{\mathfrak{N}} \times D_{\mathfrak{N}}. \text{ We identify } D_{\mathfrak{D}} \text{ with } D_{\mathfrak{N}}^{\{x,y\}}, \text{ where } x \text{ and } y \text{ are distinct} \\ &\quad \text{variables.} \\ &\Leftrightarrow R_{\mathfrak{D}} := \widehat{R_{\mathfrak{N}}}, \text{ where } \langle a, b \rangle \widehat{R_{\mathfrak{N}}} \langle c, d \rangle : \Leftrightarrow aR_{\mathfrak{N}}c \text{ and } b = d. \end{aligned}$$

Note that $\mathfrak{D}$ is just a disjoint union of isomorphic copies of $\mathfrak{N}$. By an easy induction one shows that $\langle a, b \rangle [\phi]_{\mathfrak{N}} \langle c, d \rangle : \Leftrightarrow a[\phi]_{\mathfrak{N}}c$ and $b = d$. We show that $\mathfrak{D}$ is DPL-definable. We introduce the DPL-language $\mathcal{L}$ as before – but for the presence of $\wedge$.

* Hollenberg has proved that the operations safe for full bisimulation are obtained by adding the universal relation to the basic repertoire in van Benthems result (see Hollenberg, 1995).

We define the resetting relation R_Δ in $\mathfrak{M}$, by: $(\exists y.R(x, y). \exists x.x=y. \exists y \wedge \exists x)^*$. In the last atom, $\exists y$, of the first conjunct we reset y arbitrarily. The second conjunct $\exists x$ constrains this arbitrary reset of y to be a reset to y 's original value.

From our previous observations we find immediately: $\models_{dpl} = \models$ and $=_{dpl} = =$ for the enriched language.

Acknowledgements

I thank Marco Hollenberg and Kees Vermeulen for many enlightening conversations and for their careful reading of an earlier version of this paper. I thank Johan van Benthem for his stimulating and insightful e-mails.

Appendix

A. The Interdefinability of $=$ and $\models$

Let us first have a look at the the interdefinability of $=$ and $\models$ in DR. Clearly, we can define $\models$ in terms of $=$ even in a fixed model $\mathfrak{M}$ by putting: $\phi \models_{\mathfrak{M}} \psi \Leftrightarrow \delta =_{\mathfrak{M}} (\phi \rightarrow \psi)$. For a single model no converse of this reduction is possible, since $\models$ is preserved by bisimulation, but $=$ is not. However if we quantify over all models a kind of converse is possible. As far as I can trace it, the result is folklore. Let us write $\phi \sqsubseteq_{\mathfrak{M}} \psi$ for: $[\phi]_{\mathfrak{M}} \subseteq [\psi]_{\mathfrak{M}}$ and $\phi \sqsubseteq \psi$ for: for all $\mathfrak{M}$: $\phi \sqsubseteq_{\mathfrak{M}} \psi$. Evidently, if we can define $\sqsubseteq$, we can define $=$.

THEOREM A.1. *Consider DR-formulas ϕ and ψ . Suppose the relational atom R does not occur in ϕ, ψ . We have: $\phi \sqsubseteq \psi \Leftrightarrow \neg\neg(\phi.R) \models (\psi.R)$.*

Proof. “ $\Rightarrow$ ” Easy. “ $\Leftarrow$ ” Suppose $\neg\neg(\phi.R) \models (\psi.R)$. Consider any model $\mathfrak{M}$ for the language of ϕ, ψ . We want to show: $\phi \sqsubseteq_{\mathfrak{M}} \psi$. Suppose $a[\phi]_{\mathfrak{M}}b$. Extend $\mathfrak{M}$ to $\mathfrak{N}$, by interpreting R by: $d[R]_{\mathfrak{N}}e \Leftrightarrow d = e = b$. Clearly: $a[\neg\neg(\phi.R)]_{\mathfrak{N}}a$. Hence, by our assumption, for some c , $a[\psi.R]_{\mathfrak{N}}c$. Clearly, c must be b , so $a[\psi.R]_{\mathfrak{N}}b$. It follows that $a[\psi]_{\mathfrak{N}}b$ and, so, $a[\psi]_{\mathfrak{M}}b$. $\square$

Note that in the language with variables for conditions we could have used a condition variable instead of R . We turn to DPL/A. We show that in the DPL language we can define $=$ from $\models$ even in a single model.

THEOREM A.2. *Consider DPL-formulas ϕ and ψ . Let $V = \{v_1, \dots, v_n\}$ be the set of variables occurring in ϕ, ψ . We assume that Var is large enough, so that we can find a set of variables $W = \{w_1, \dots, w_n\}$ disjoint from V . We have:*

$$\phi \sqsubseteq \psi \Leftrightarrow \neg\neg(\phi.v_1 = w_1 \cdots v_n = w_n) \models (\psi.v_1 = w_1 \cdots v_n = w_n).$$

* Since we omit brackets for the “.”, we cannot do so for other binary connectives. Thus, “.” always binds stronger than other binary connectives. Consequently, “ $\wedge$ ” will be the main connective of our formula.

Proof. “ $\Rightarrow$ ” Not too difficult. “ $\Leftarrow$ ” Suppose

$$\neg\neg(\phi.v_1 = w_1 \cdots v_n = w_n) \models (\psi.v_1 = w_1 \cdots v_n = w_n).$$

Consider any DPL-model $\mathfrak{M}$. Suppose $f[\phi]_{\mathfrak{M}}g$. We write $f_{d_1, \dots, d_n}^{w_1, \dots, w_n}$ for: the result of resetting f on w_1 to d_1 , on w_2 to d_2 , etc. Since the w_i do not occur in ϕ , we find:

$$f' := f_{g(v_1), \dots, g(v_n)}^{w_1, \dots, w_n}[\phi]_{\mathfrak{M}}g_{g(v_1), \dots, g(v_n)}^{w_1, \dots, w_n} =: g'.$$

Hence, $f'[\phi.v_1 = w_1 \cdots v_n = w_n]_{\mathfrak{M}}g'$, and, so,

$$f'[\neg\neg(\phi.v_1 = w_1 \cdots v_n = w_n)]_{\mathfrak{M}}f'.$$

By our assumption, there is an h' such that

$$f'[\psi.v_1 = w_1 \cdots v_n = w_n]_{\mathfrak{M}}h'.$$

Since the w_i do not occur in ψ , we find, for $i = 1, \dots, n$, $h(v_i) = h(w_i) = f'(w_i) = g(v_i)$. Moreover, $f'[\psi]_{\mathfrak{M}}h'$. Again, by the fact that the w_i are not in ψ and the fact that only the v_i occur in ψ , we find: $f = f_{f(v_1), \dots, f(v_n)}^{w_1, \dots, w_n}[\psi]_{\mathfrak{M}}h_{f(v_1), \dots, f(v_n)}^{w_1, \dots, w_n} = g$. $\square$

If we remove identity from the language of DPL the above result does not go through, as is shown by the following example.

Example A.3 (with Marco Hollenberg). Consider a DPL-language without identity and with at least one unary predicate symbol P . Let D be a domain with at least two elements and let $\mathcal{E}$ be a non-trivial equivalence relation on D . Suppose c and c' are distinct, but equivalent elements in D . We interpret all predicate symbols of the language in such a way that they make $\mathcal{E}$ into a congruence relation. We take $I(P)$ to be the equivalence class of c . This gives us a model $\mathfrak{M}$. Let $\mathfrak{M}_{\mathcal{E}}$ be the model obtained by dividing out $\mathcal{E}$. Consider the relation $\in^*$ between assignments on $\mathfrak{M}$ and assignments on $\mathfrak{M}_{\mathcal{E}}$, given by: $f \in^* f' :\Leftrightarrow$ for all $v \in \text{Var}$ $f(v) \in f'(v)$. It is easily verified that $\in^*$ is a bisimulation w.r.t. the atomic conditions and the $\exists x$'s. By safety, $\in^*$ will be a bisimulation w.r.t. all transitions definable in the language. Thus, the valid sequents of $\mathfrak{M}$ and $\mathfrak{M}_{\mathcal{E}}$ will be the same. However, in $\mathfrak{M}$ we have: $P(x). \exists x. P(x) \neq P(x)$, since $c[P(x). \exists x. P(x)]c'$, but not $c[P(x)]c'$. Moreover, in $\mathfrak{M}_{\mathcal{E}}$ we have: $P(x). \exists x. P(x) = P(x)$. $\square$

References

- van Benthem, J.F.A.K., 1993, “Program constructions that are safe for bisimulation,” CSLI Report 93-179, CSLI, Stanford University, Stanford, to appear in *Proceedings Logic Colloquium, Clermont Ferrant, 1994*, Studia Logica.
- van Benthem, J.F.A.K. and Ceparrello, G., 1994, “Tarskian variations. Dynamic parameters in classical semantics,” CS R9419, Amsterdam: CWI.
- Blackburn, P. and Venema, Y., 1995, “Dynamic squares,” *Journal of Philosophical Logic* **24**(5), 469–523.
- Groenendijk, J. and M. Stokhof, M., 1991, “Dynamic predicate logic,” *Linguistics and Philosophy* **14**, 39–100.

- Hollenberg, M.J., "Safety for bisimulation in general modal logic," pp. 359–376 in *Proceedings of the 10th Amsterdam Colloquium*, Part II, P. Dekker et al., eds., Amsterdam: ILLC, Department of Philosophy, University of Amsterdam.
- Hollenberg, M.J., 1997, "An equational axiomatisation of dynamic negation and relational composition," *Journal of Logic, Language, and Information*, this issue.
- Kamp, H., 1981, "A theory of truth and semantic representation," in *Formal Methods in the Study of Language*, J. Groenendijk et al., eds., Amsterdam: Mathematisch Centrum.
- Visser, A., 1995, "Contexts for dynamic predicate logic," Logic Group Preprint Series 143, Department of Philosophy, Utrecht University, Heidelberglaan 8, 3584 CS Utrecht, October 1995, to appear in *Journal of Logic, Language, and Information* special issue on Contexts.