Journal of Logic, Language, and Information **7:** 21–52, 1998.

Contexts in Dynamic Predicate Logic

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

(Received in final form 19 February 1997)

Abstract. In this paper we introduce a notion of context for Groenendijk & Stokhof's Dynamic Predicate Logic DPL. We use these contexts to give a characterization of the relations on assignments that can be generated by composition from tests and random resettings in the case that we are working over an infinite domain. These relations are precisely the ones expressible in DPL if we allow ourselves arbitrary tests as a starting point. We discuss some possible extensions of DPL and the way these extensions interact with our notion of context.

Key words: Context, dynamics, Predicate Logic, relation, resetting, register, variable, expressability, definability

> Contexts manifest themselves even when not explicitly definitionally introduced.

1. Introduction

1.1. Metamathematics of Dynamic Predicate Logic

Dynamic Predicate Logic (DPL) was invented by Groenendijk and Stokhof (1991), see also our Section 2) as a specification language (or better: as a module for a specification language) of meanings for fragments of natural language. Most of the research concerning DPL has gone into integrating it with versions of Montague Grammar (see Muskens, 1991) and into integrating it with Veltman's Update Semantics (see Veltman, 1996; Groenendijk et al. 1996).

DPL is a theory of testing and resetting of variables/registers. These are fundamental operations in computer science. Thus, apart from its use in Logical Semantics, DPL is a simple theory of these basic operations.*

DPL is a natural variant of Predicate Logic. It mainly differs in the treatment of the scope of the existential quantifier. Certain basic truths about variables in Predicate Logic, however, fail in DPL (see Section 2, see also Hollenberg and Vermeulen, 1994) for similar observations on DPLE). The study of DPL and

* DPL is just one theory in a family of alternatives to Predicate Logic. In other alternatives "resetting a register" is replaced by other related actions, like "create a new register"(see Appendix B for references).

its kin makes the dependence of these truths on the specific choice of scoping mechanisms in standard Predicate Logic visible.

In the light of the varied interest of DPL, it seems a good idea to make a closer study of its metamathematical properties. We focus on the closely related questions:

- ⇔ Which relations between assignments are expressible in DPL (in a sense we will specify later)?
- ⇔ How does DPL treat its variables?

To throw some light on both questions a good notion of context in DPL is indispensable. In the next subsection we present our notion of context and discuss its connections and differences with notions of context in the literature.

1.2. CONTEXTS

Contexts appear in the literature in bewildering variety. We briefly discuss three meanings of *context*, viz.

1. contexts as syntactical objects with “holes,”
2. contexts as assignments/environments/anchors,
3. contexts as (structured) declarations of variables.

The last notion is the one that we are concerned with in this paper.

A common occurrence of *context* is as a “hole” in a syntactical object. This notion is important in term-rewriting and proof theory. See Troelstra and van Dalen (1988a: 60), Barendregt (1984: 29, 375), Klop (1992: 12) for standard references. The contexts studied in this paper have *nothing* to do with *this* notion of context.

The second notion of context closer – but not equal – to the notion we are after is a context as an environment or assignment. This usage is well established in Linguistics. E.g., in the Heim tradition one speaks about “context change potential” for the potential to change assignments. Using *context* for assignment is consonant to the way David Kaplan and the Situation Semanticists employ the word. Here a context is the speaker situation: this situation provides values for *I, here, now,* . . . in the same way as an assignment provides values for the variables. For the use, in Computer Science, of *environment* for assignment, see, e.g., Mitchell (1991: 383). In Situation Semantics sometimes *anchor* is used for assignment/environment.

The notion of context that will emerge in the present paper is best viewed as in harmony with a tradition in Proof Theory, Type Theory, Category Theory and Computer Science in which a context is *a declaration of variables*. Some references for this usage are: Troelstra and van Dalen (1988b: 577); Jacobs (1991: 29); Nederpelt (1994: 103, 104, 233, 253), etc. A context in this sense has also been called *a basis*. See, e.g., Barendregt (1991: 351) or Barendregt (1992: 149). In this last paper (Barendregt, 1992) the word *context* is used for a *linearly ordered* set of declarations as opposed to such a set simpliciter: see p. 199. Contexts are called *referent systems*, in Vermeulen (1995). Contexts in our present, third, sense are clearly connected with contexts in the second sense: contexts in the third sense provide the domains for contexts in the second sense.

We give a brief discussion to introduce some of the ingredients of contexts in the third sense, which from now on we simply call *contexts*. Variables are usually declared as objects of a certain *type*. Since, however, we are thinking about *First Order* Predicate Logic here, we may ignore the types for the purposes of this paper. As we will see, what is left is still not completely trivial, especially not in a dynamic setting.

The simplest form of a context is the context assigned to a formula ϕ of Predicate Logic, viz. *the set of its free variables*. In the "indexed" treatment of the semantics of Predicate Logic (see Jacobs, 1991) this context also occurs at the semantical level: the meaning of ϕ in a given model $\mathcal{M}$ is the pair $\langle X, F\rangle$, where X is $\mathsf{FV}(\phi)$, the set of free variables occurring in ϕ, and F is a set of assignments from X to the domain of $\mathcal{M}$. Note that on the indexed perspective an assignment can be viewed as a pair $\langle X, f\rangle$, where X is the domain of f. Thus, assignments/environments are objects "in context."

To develop more dynamical kinds of context, we consider two example sentences. Consider the sentences *he is a man* and *a man comes in*. In the Predicate Logical specification language of DPL and Kamp's Discourse Representation Theory, DRT, we will render these sentences, e.g., as

$$\mathit{MAN}(x) \quad \text{and} \quad \exists x.\mathit{MAN}(x).\mathit{COMES\text{-}IN}(x).$$

In the style of the "indexed" semantics we would give the first sentence as context $\{x\}$ and the second one $\emptyset$. Viewed dynamically we can interpret the first context as embodying the information that a value for x is asked from the incoming assignment. This is analogous to the way in which the anaphor *he* "asks for" a value from (the interpretation of) the previous text, and, also, to the way in which *the present king of France* signals that a king of France is presupposed. Thus, an "indexed" style context signals the presence of a kind of presupposition. The second context, $\emptyset$, signals that no previous value for any variable is demanded.

However, variables occur also in other ways in our formulas. $\exists x$ newly introduces an x. Consider the Discourse Representation Structures or DRSs associated to our sentences. We can represent these as:

$$\langle\emptyset, \{MAN(x)\}\rangle \quad \text{and} \quad \langle\{x\}, \{MAN(x), \mathit{COMES\text{-}IN(x)}\}\rangle.$$

It seems reasonable to consider the first components of DRSs – these are the top boxes of the box representation – also as contexts. These first components signal which variables are *newly introduced* or *declared*. Reflecting on the "indexed" contexts and the DRT-contexts, it is a small step to form combined contexts $\langle X, Y\rangle$, where X and Y are finite sets of variables. Here X will tell us which variables are imported from outside and Y will inform us which variables are newly introduced. For example,

$$\mathit{FARMER}(x).\exists y.\mathit{OWNS}(x,y).\mathit{DONKEY}(y)$$

will have context $\langle\{x\},\{y\}\rangle$. Following up this idea it is not too difficult to develop versions of DRT that are compatible with DPL, i.e., versions of DRT in which resetting actions are associated to DRSs in a compositional way.[*] See, for example, (Visser, 1995) for an implementation of this idea. The DPL-contexts of this paper are much like such pairs $\langle X,Y\rangle$. However, they have a third component: the future directed mirror image of the presupposition set. That such a mirror image should exist follows from the time symmetry of relation composition, the fundamental operation of DPL. So we get contexts of the form $\langle X,Y,Z\rangle$, where X is the set of free or presupposed or input-constrained variables, where Y is the set of variables that may be reset and where Z is the set of output-constrained variables.

To define contexts as objects of a certain sort is a rather empty exercise if we do not specify certain operations on contexts. The most important operation is without doubt the *merge* or *simple addition of contexts*. We represent the merge by "$\bullet$." For example, the indexed contexts for Predicate Logic were just sets of free variables. The corresponding merge is set theoretical union, consonant with the familiar fact that $\mathsf{FV}(\phi\wedge\psi) = \mathsf{FV}(\phi)\cap\mathsf{FV}(\psi)$. What would be the merge on pairs $\langle X,Y\rangle$ as discussed above?[**] Here more than one choice is possible. One such choice is:

$$\langle X,Y\rangle \bullet \langle X',Y'\rangle = \langle X\cup(X'\setminus Y), Y\cup(Y'\setminus X)\rangle.$$

What does this choice mean? To gain better understanding, we must digress for a moment on the subject of variable clashes.

In dynamic semantics for Predicate Logic, we have to agree upon conventions to handle clashes of variables. For example, what happens when x is already introduced and assigned a value and we try to introduce it or a new value for it again? A wide range of solutions is available. We may decide that the new introduction will bounce. Nothing will happen and x will keep its old value. This is the strategy of classical DRT (see also Visser, 1995). We may also opt for the contrary strategy of throwing away the old value and making the new value the actual one. This is the strategy of classical DPL. A third possibility is to stack the new value on top of the old one (see Vermeulen, 1993; Hollenberg and Vermeulen, 1994; Visser and Vermeulen, 1996). A fourth possibility is to distinguish between the variable-qua-identifier, or variable-as-syntactic-object, and the variable-as-underlying-file. If we make this, philosophically plausible, distinction, we can view the second introduction as a *renaming*: the original file named "x" loses its name; a newly introduced file is baptized "x" (see Vermeulen, 1995; Groenendijk et al., 1996; Visser and Vermeulen, 1996).

[*] Ordinary DRT with sets as contexts does not have this property.

[**] To be precise, we assume at this point, locally, that the intersection of the first and the second component is empty.

Strategies introduced to avoid clashes of variables are reflected by the definitions of contexts and their merge. Consider the merge that we introduced for pairs of sets of variables. If x is both in X and in Y', then x will be in $X \cup (X' \setminus Y)$, but not in $Y \cup (Y' \setminus X)$. The reason is that our example corresponds to a "bouncing" strategy: if x is already present, any attempt to newly introduce it will fail. Another example: the contexts, called *referent systems*, studied in Vermeulen (1995) and Groenendijk et al. (1996) contain both files and file names. Thus, the fundamental distinction is already present at the contextual level. In Visser and Vermeulen (1996) it is attempted to handle the variable clashes fully at the level of the contexts.

As this point, sufficiently many ideas have been touched upon to sketch a general picture of the role that in my view contexts have to play in dynamics. This picture is more extensively described in Visser and Vermeulen (1996). A context is essentially a set of files or discourse referents. These files are best thought of as featureless objects. Also part of the contexts is machinery that "programmes" what happens to files when two contexts are merged: whether various files are fused, whether a given file is aborted, whether a file is renamed, whether a file is relocated in a structure, etc. So the context of a formula ϕ will contain the files used in ϕ plus the information about how the files are "handled" by ϕ: are they *free, fresh* . . . ?

In dynamic semantics contexts will be an important part of the larger picture. They may be enriched with contents that could be as varied things as sets of assignments, pictures or relational databases. The contexts determine what happens to the contents in merging full meaning objects. Thus, contexts are "demi-mondain:" they can be directly assigned to syntactical objects like terms and formulas, but are also naturally paired with the semantical objects, like assignments, sets of assignments, resetting relations, update functions. All these semantical objects are objects "in context."

Updating does not just result in growth of contentual information, but also in extension of context. In this sense, information growth is many dimensional. It is important to realize what this means for improved versions of DPL. The assignments that are modified will be not assignments on a rigid set of variables but on a changing set of files. These files will be enriched with certain information to regulate their interactions in merging. Such an "assignment in context" can be viewed as a special case of an information state in update semantics, which is a set of assignments or a set of assignment-world pairs in context.[⋆]

We mention two successes of the use of contexts in dynamics. The first success is the clarification of the relationship between DPL-style semantics and DRT-style semantics. See, e.g., Vermeulen (1995) and Visser (1995). The second is the solution of the problem to integrate DPL and Update Semantics, including the semantics for *maybe*, by Groenendijk et al. (1996).

⋆ DRT-meanings can be viewed as information states in context. Remember that DRT-meanings can be interpretations of formulas with free variables. Thus their contexts will witness a presuppositional structure. DRT-meanings are partial states or states under presuppositions. (See Visser, 1994, for a first attempt to come to grips with the idea of a partial state.)

Contexts in this paper will be partly atypical. In various versions of DPL and DRT as introduced in Vermeulen (1993, 1995), Visser (1995), Visser and Vermeulen (1996), Groenendijk et al. (1996), contexts make their appearance in the very definition of the semantics: they are part of the design of the semantics. Moreover, they admit a realistic interpretation: they could model real mechanisms of handling files. In this paper, however, we study classical DPL as introduced in (Groenendijk and Stokhof, 1991). No contexts occur in the definition of classical DPL. Thus, in this paper, contexts can only appear as *objets trouvés*: they emerge as a useful device to characterize expressibility in DPL. Contexts result from "abstracting away" or "erasing" certain properties of the predicate logical language (both vocabulary and structure), thus yielding an underspecified language. Underspecification simply means here that the denotations or meanings of the contexts are properties of relations, rather than relations.

A second feature of contexts will be lacking in this paper: assignments will have no associated context. Classical DPL is a theory of *total* assignments. Thus, the only possible context these assignments could have is the fixed set of all variables. Explicitly carrying this context along is pointless. Thus, our contexts will be exclusively associated to the resetting relations.

The present paper illustrates that contexts are useful, not just for their philosophical plausibility, but also because they are technically fruitful.

1.3. The Cornerstone Argument

Consider the sentence $Px.\exists x.Qx$. The leftmost occurrence of x is in some sense "used" in a different way from the rightmost occurrence. This fact gives rise to the problem how these two occurrences have to be treated in the semantics.

It is often argued that one of the most salient and exciting aspects of theories like DPL and DRT, viz. the treatment of the problem of multiple uses of variables, is irrelevant for linguistics. We give the argument (for the specific case of DPL) as it occurs in an anonymous reviewers comments.

> Consider a typical example of multiple use, say, $Px.\exists x.Qx$. Such formulas are never needed when one translates natural language sentences into DPL formulas. When translating an indefinite noun phrase, one can always use a fresh variable, so that in any subformula of the translation, no variable occurs both free and bound. Thus, instead of a formula like the above, one could use $Px.\exists y.Qy$ with no loss. The existence of a formula like the first one is an artifact of the formalization, and the study of such formulas does not reveal anything about natural language.

I call this *the cornerstone argument*, since it denies a place to what I consider one of the central issues in the development of dynamic versions of Predicate Logic. I have two objections to the argument, a minor and a major one. Before giving these objections, let me point out that not all sources of interest of theories like DPL derive from the study of Language. The treatment of variables is, e.g., an important subject in Computer Science. The "overloading" of variables in large programmes is often nearly unavoidable. Moreover, the precise treatment of variables is almost definitionally of *logical* interest.

We turn to my minor objection. Our objector is certainly right that "one can always use a fresh variable, so that in any subformula of the translation, no variable occurs both free and bound." But this is only obvious for cases where the variable-qua-syntactic-object is really extrinsic to natural language. I am thinking here of cases where natural language clearly employs different mechanisms for referent linking than the linking by label of DPL. Natural language simply does not employ variables-qua-syntactic-objects to realize the linking of the referents associated to anaphors. However, in Visser and Vermeulen (1996) the idea is explored to treat roles/casus/items-of-feature-information as dynamic variables. Here re-using the same "variable," like subject is quite plausible. Now, there must be something in the semantics to stop the unification of all subjects . . . . Our counterargument leaves open the possibility that the cornerstone argument convincingly shows that the study of multiple uses of variables is unilluminating for the study of *anaphors*.

My main objection is that the cornerstone argument presumes that the only way in which theories like DPL could reveal anything about natural language is by being a specification language whose formulas are assigned to natural language texts by a certain translation procedure. I submit that having such a specification language and such a translation procedure (for a reasonable fragment) is not something we must rest content with. The cornerstone argument itself indicates that the variables of the specification language (in the case of the treatment of anaphors) are extrinsic to the natural language. The variables are added in the translation procedure by the translator. This means that the translation procedure could very well produce the correct meaning objects, but can never be considered as a model of the process of understanding, since the resolution of anaphors is clearly partially based on contentual considerations. The ordinary regulative principle of compositionality as employed in the Montague tradition already shows that not just the end result of translation is important, but also the way the result is obtained. The programme of Dynamics, as I see it, adds a further constraint. Dynamics is directed at providing models of interpretation processes. Thus, we want our semantical interpretation to reflect the interpretation process studied. The translation procedure relied upon in the cornerstone argument does not provide any insight in the dynamics of interpretation.

So what is the role of theories like DPL in the study of natural language? The alternative way, of viewing the relationship of dynamical systems and natural language that I would like to push, is this. I want to view these systems as *analogues* of natural language. The systems (languages with their given semantics) are analogous to natural language in certain respects. E.g., they allow long distance linking of variables/anaphors. We study these systems as fruitflies to receive hints about what a semantics for (a reasonable fragment of) natural language should look like. DPL and natural language have long distance linking, but the detailed mechanisms seem to be quite different. The one, DPL, has linking by label and the other .... Still the careful study of the linking machinery of DPL could help us understand other linking strategies. Analogy is helpful not only by providing points of agreement, but also by giving us a good understanding of points where the analogues differ.

Let us look at the matter from a different perspective. What is the business of Semantics? Well, it is to associate objects from a semantical algebra to a syntactical algebra. The semantical objects and the syntactical objects differ in some salient respects. This is what gives the exercise a point. The semantical objects are both richer and poorer than the syntactical ones. E.g., a DRS lacks the linear organisation of the linguistic string. E.g., a set of possible worlds contains all kinds of stuff not present in a string of letters. Now it seems to me that we should be able to understand the semantical algebra as a thing on its own. Part of the essence of the semantical objects is revealed by the natural interactions between these objects, in other words, what makes the algebra into an algebra. The cornerstone argument asks us to leave a substantial part of these interactions unspecified or arbitrarily specified, since they are supposed not to matter.

Let me end with a question: how can one expect to understand natural language, other than as one way of doing things *among others*?

1.4. Contents of the Paper

Section 2 is a straightforward introduction to Dynamic Predicate Logic. It contains all the *technical* material the reader needs to know. For a discussion of the applications to discourse phenomena, however, the reader should consult Groenendijk and Stokhof (1991). Section 3 presents our theory of contexts in DPL. We study contexts as mathematical objects in their own right and establish their connections to language and semantics. Some materials concerning the information ordering on contexts are placed in an appendix. In Section 4, we treat the Switching Property. This property is characteristic for the DPL-expressible relations. Section 5 contains the main result of the paper: a relation over an infinite domain is DPL-expressible iff it "has" a context and satisfies the Switching Property. In the next section, we touch on the subject of extending the DPL-language with new operations such as conjunction and disjunction. We consider the question whether such extensions support a good theory of contexts. We will produce two extensions that are *complete* for all relations that have a context. In other words: all such relations are expressible in those extensions. In Appendix B we have a brief look at the idea of making the context *part of* the DPL-semantics.

2. What Is Dynamic Predicate Logic?

We provide the basic definitions of DPL. Nothing in this section pretends to be original. We start by introducing some basic relational notions.

DEFINITION 2.1. Let X be any non-empty set. $\mathsf{Rel}(X)$ is the set of binary relations on X, i.e., $\mathsf{Rel}(X) := \wp(X \times X)$. Let $R, S \in \mathsf{Rel}(X)$. We define:

1. The composition $R \circ S$ of R and S is defined by: $x(R \circ S)y :\Leftrightarrow \exists z\ xRzSy$. Note that composition is in the order of application.
2. The dynamic implication $(R \to S)$ between R and S is defined by:

   $x(R \to S)y :\Leftrightarrow x = y$ *and* $\forall z(xRz \Rightarrow \exists u\ zSu)$.

   Our use of $\to$ here overloads the symbol, since we also use it for implication in the object language. We write $\neg(R)$ for $(R \to \emptyset)$
3. id_X is the identity relation. R is a *condition* or *test* if $R \subseteq \mathsf{id}_X$.
4. Consider $Y \subseteq X$. We define $\mathsf{diag}(Y) := \{\langle y, y\rangle \in X \times X \mid y \in Y\}$.
5. $\mathsf{dom}(R) := \{x \in X \mid \exists y\ xRy\}$ and $\mathsf{cod}(R) := \{y \in X \mid \exists x\ xRy\}$.

The notion of dynamic implication was first introduced by Kamp in his pioneering paper (1981). Note that: $\neg\neg(R) = (\mathsf{id}_X \to R) = \mathsf{dom}(R)$. The relations in the range of diag are precisely the conditions. Writing $(Y \to Z) := (X \setminus Y) \cup Z$, we have:

- $\Leftrightarrow \mathsf{diag}(X) = \mathsf{id}_X$
- $\Leftrightarrow \mathsf{diag}(\emptyset) = \emptyset$
- $\Leftrightarrow \mathsf{diag}(Y \cap Z) = \mathsf{diag}(Y) \circ \mathsf{diag}(Z) = \mathsf{diag}(Y) \cap \mathsf{diag}(Z)$
- $\Leftrightarrow \mathsf{diag}(Y \cup Z) = \mathsf{diag}(Y) \cup \mathsf{diag}(Z)$
- $\Leftrightarrow \mathsf{diag}(Y \to Z) = \mathsf{diag}(Y) \to \mathsf{diag}(Z)$.

Thus, diag embeds the structure $\langle \wp X; X, \emptyset, \cap, \cap, \cup, \to\rangle$ homomorphically into the structure $\langle \mathsf{Rel}(X); \mathsf{id}_X, \emptyset, \circ, \cap, \cup, \to\rangle$. We will sometimes confuse, in the relational context, the set X with the relation $\mathsf{diag}(X)$. We need some further relational notions specifically concerned with relations between assignments.

DEFINITION 2.2. Let D be a non-empty domain and let Var be a set of variables. Let $R \in \mathsf{Rel}(D^{\mathsf{Var}})$, $Y \subseteq D^{\mathsf{Var}}$, $f \in D^{\mathsf{Var}}$ and $V \subseteq \mathsf{Var}$. We define:

- $\Leftrightarrow f^{d_1,\ldots,d_n}_{v_1,\ldots,v_n}$ is the result of changing the values of f on the v_i to d_i.
- $\Leftrightarrow f\mathcal{I}_V g :\Leftrightarrow$ for all $v \in V$ $\ f(v) = g(v)$.
- $\Leftrightarrow [V] := \mathcal{I}_{\mathsf{Var}\setminus V}$. We write $[v]$ for: $[\{v\}]$.
- $\Leftrightarrow Y$ is a $\langle V\rangle$-*set* if $f \in Y$ and $f\mathcal{I}_V g \Rightarrow g \in Y$. R is a $\langle V\rangle$-*condition* if it is the diag-image of a $\langle V\rangle$-set. Y is *finitely restricted* if Y is a $\langle I\rangle$-set for some *finite* I. A condition is finitely restricted iff it is the image under diag of a finitely restricted set.

If we want to make the dependence of $\mathcal{I}$ or [.] on Var or D visible, we add them as subscripts.

We collect some simple facts concerning these notions. We have:

- $\mathcal{I}_\emptyset = [\mathsf{Var}] = D^{\mathsf{Var}} \times D^{\mathsf{Var}}$, $\mathcal{I}_{\mathsf{Var}} = [\emptyset] = \mathsf{id}_{D^{\mathsf{Var}}}$
- $\mathcal{I}_V \circ \mathcal{I}_W = \mathcal{I}_{V \cap W}$, $[V] \circ [W] = [V \cup W]$
- $\mathcal{I}_V \cap \mathcal{I}_W = \mathcal{I}_{V \cup W}$, $[V] \cap [W] = [V \cap W]$
- $\mathcal{I}_V \cup \mathcal{I}_W \subseteq \mathcal{I}_{V \cap W}$, $[V] \cup [W] \subseteq [V \cup W]$.

Note that the classical meaning of the existential quantifier as a "cylindrification" can be given as: $\Sigma x(Y) := \mathsf{dom}([x] \circ \mathsf{diag}(Y))$. We turn to the definition of DPL.

DEFINITION 2.3. We define a DPL-language $\mathcal{L}$ as follows. $\mathcal{L}$ is a structure $\langle \mathsf{Pred}, \mathsf{Ar}, \mathsf{Var}, \mathsf{Con} \rangle$. Here 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 and Con is a, possibly empty, set of constants. Let $\mathsf{Ref} := \mathsf{Var} \cup \mathsf{Con}$ be the set of *referents*. We will use $v, w, \ldots$ for variables, $c, c', \ldots$ for constants and $r, s, \ldots$ for referents. The set of *$\mathcal{L}$-formulas*, $\mathsf{For}_{\mathcal{L}}$, is the smallest set such that:

- $P(r_1, \cdots, r_n) \in \mathsf{For}_{\mathcal{L}}$, for $P \in \mathsf{Pred}$, $\mathsf{Ar}(P) = n$ and $r_1, \ldots, r_n \in \mathsf{Ref}$
- $\top$, $\bot$, $r = s$, $\exists v$ are in $\mathsf{For}_{\mathcal{L}}$ for $r, s \in \mathsf{Ref}$ and $v \in \mathsf{Var}$
- If $\phi, \psi \in \mathsf{For}_{\mathcal{L}}$, then so are $\phi.\psi$ and $(\phi \to \psi)$.

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 – the semantic counterpart of "." – is associative. An alternative notation for $\exists v$, is $[v :=?]$ (random reset). We use $\neg(\phi)$ and $\forall v(\phi)$ as abbreviations of, respectively, $(\phi \to \bot)$ and $(\exists v \to \phi)$. If $x \in \mathsf{Var}$ and $r \in \mathsf{Ref}$ and x and r are distinct, we write $[x := r]$ for: $\exists x.x = r$.

DEFINITION 2.4. A DPL-model $\mathcal{M}$ for a DPL-language $\mathcal{L}$ is a structure $\langle D, I \rangle$, where D is a non-empty set, the *domain* of $\mathcal{M}$; I is a function which assigns to each predicate symbol P of $\mathsf{Pred}_{\mathcal{L}}$ an $\mathsf{Ar}(P)$-ary relation on D and to each constant c an element of D. $\mathsf{Ass}_{\mathcal{M}}$, the set of *assignments* for $\mathcal{M}$, is D^{Var}. Consider $r \in \mathsf{Ref}$. We define:

$$|r|_{\mathcal{M},f} := \begin{cases} f(r) & \text{if } r \in \mathsf{Var} \\ I(r) & \text{if } r \in \mathsf{Con} \end{cases}$$

The interpretation function $[.]_{\mathcal{M}} : \mathsf{For}_{\mathcal{L}} \to \mathsf{Rel}(\mathsf{Ass}_{\mathcal{M}})$ is given as follows.

- $[P(r_1, \cdots, r_n)]_{\mathcal{M}} := \mathsf{diag}(\{f \in D^{\mathsf{Var}} \mid \langle |r_1|_{\mathcal{M},f}, \cdots, |r_n|_{\mathcal{M},f} \rangle \in I(P)\})$
- $[\top]_{\mathcal{M}} := \mathsf{id}_{D^{\mathsf{Var}}}$, $[\bot]_{\mathcal{M}} := \emptyset$
- $[r = s]_{\mathcal{M}} := \mathsf{diag}(\{f \in D^{\mathsf{Var}} \mid |r|_{\mathcal{M},f} = |s|_{\mathcal{M},f}\})$

⇔ $[\exists v]_{\mathcal{M}} := [v]_D$
⇔ $[\phi.\psi]_{\mathcal{M}} := [\phi]_{\mathcal{M}} \circ [\psi]_{\mathcal{M}}$
⇔ $[(\phi \to \psi)]_{\mathcal{M}} := ([\phi]_{\mathcal{M}} \to [\psi]_{\mathcal{M}})$.

We write $\phi \equiv_{\mathcal{M}} \psi$ for $[\phi]_{\mathcal{M}} = [\psi]_{\mathcal{M}}$. We define validity in DPL by:

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

As usual, $\phi \models \psi$ iff $\phi \models_{\mathcal{M}} \psi$ for all models $\mathcal{M}$ appropriate for the given language.

A binary relation R is *definable* in a DPL-model $\mathcal{M}$ for a language $\mathcal{L}$ if there is an $\mathcal{L}$-formula ϕ which defines R, i.e., $R = [\phi]_{\mathcal{M}}$.

We will often suppress the subscript $\mathcal{M}$, when the model is clear from the context. 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: $\neg\neg(\exists u.G(x,u).\exists v.F(u,v).P(v))$.

We remind the reader of Geach's Donkey Sentence: *If a farmer owns a donkey, he beats it*. This sentence can be translated into DPL in a compositional way as:

$$(\exists x.\textit{farmer}(x).\exists y.\textit{donkey}(y).\textit{owns}(x,y) \to \textit{beats}(x,y)).$$

One striking feature of DPL is that it is not "structural:" the values the predicate symbols may assume are not all the possible meaning objects provided by the semantics; we only allow tests. A second striking feature is the *time symmetry* of resetting and composition, which contrast strongly with our *time asymmetric intuition* about, say, the meaning of $P(x).\exists x.Q(x)$. The asymmetry of our intuition may be explained by the fact that we tend to think more in terms of *successful* resetting, i.e., $\mathcal{M} \models P(x).\exists x.Q(x)$, than just in terms of what the resetting relation is.

Ordinary Predicate Logic can be interpreted in DPL as follows. We suppose that the predicate logical language has as connectives and quantifiers: $\top$, $\bot$, $\wedge$, $\to$, $\exists x$. We translate as follows:

⇔ $(.)^*$ commutes with atomic formulas and with $\to$
⇔ $(\phi \wedge \psi)^* = \phi^*.\psi^*$
⇔ $(\exists x(\phi))^* = \neg\neg(\exists x.\phi^*)$.

We find: $[\phi^*]_{\mathcal{M}} = \mathsf{diag}(\llbracket \phi \rrbracket_{\mathcal{M}})$, where $\llbracket . \rrbracket$ is the usual valuation function of Predicate Logic. Our translation is compositional. It shows that we may consider Predicate Logic as a subsystem of DPL.

3. Contexts for Dynamic Predicate Logic

In this section, we study the notion of *context* and its connections with relations and language. We placed some materials concerning the information ordering on contexts in an appendix, since, on the one hand, they are conceptually relevant and have a clear place in the total picture, but, on the other hand, they have no direct bearing upon the main results of the paper.

3.1. Introductory Remarks

To motivate our notion of contexts, we first give an intuitive discussion of substitution and kinds of variable occurrences in DPL. In Predicate Logic variables may occur in a formula in two ways: freely and bound. The free variables admit (under certain conditions) substitution. The bound variables may be renamed salva significatione (α-conversion). Let us write $\sigma_x^t(\phi)$ for: the result of substituting t for x in ϕ. In Predicate Logic we have, e.g.,

$$f_x^{I(c)} \in [\![\phi]\!]_{\mathcal{M}} \Leftrightarrow f \in [\![\sigma_x^c(\phi)]\!]_{\mathcal{M}}.$$

What is the proper analogue of this fact for DPL? To simplify the discussion we will only treat a special case and refrain from giving official definitions. Consider the DPL-formula $P(x).\exists x.Q(x).\exists x.R(x)$. We have:

1. $f_x^{I(c)}[P(x).\exists x.Q(x).\exists x.R(x)]_{\mathcal{M}}g \Leftrightarrow f[P(c).\exists x.Q(x).\exists x.R(x)]_{\mathcal{M}}g$
2. $f[P(x).\exists x.Q(x).\exists x.R(x)]_{\mathcal{M}}g_x^{I(c)} \Leftrightarrow f[P(x).\exists x.Q(x).\exists x.R(c)]_{\mathcal{M}}g.$

Meditation upon (1) and (2) suggests, that, in DPL, we have to distinguish two kinds of substitution *left* substitution and *right* substitution and corresponding to these kinds two kinds of "free occurrence:" *left* free and *right* free. We also speak of *input* occurrences and *output* occurrences. Following temporal intuitions – ignoring the essentially time-symmetric character of resetting and composition – we may also call the left free occurrences simply *free* and the right free occurrences *actively bound*. Now consider the following formula, call it ϕ_0, in which we have tagged occurrences of x with superscript numerals.

$$P(x^1).\exists x^2.Q(x^3).\exists x^4.\neg\neg(\exists x^5.R(x^6)).S(x^7).$$

We see that x^1 is a (left) free or input occurrence. Left substitution for x will cause it to be replaced. If we form $T(x^0).\phi_0$, in the semantics the values assigned to x^0 and x^1 will be unified. If we form $\exists x.\phi_0$, x^1 will be "bound" or "initialized" by the new "quantifier." Symmetrically, x^7 is right free or actively bound. It will be in the scope of right substitution. If we form $\phi_0.T(x^0)$, the values of x^7 and x^0 will be unified, If we form $\phi_0.\exists x$, x^7 will be "aborted." Neither x^1 nor x^7 are open to α-conversion *salva significatione*. x^3 is not accessible for substitution, nor is it α-convertible: replacing x^2, x^3 and x^4 by, say, y, will result in a formula that resets y, which ϕ_0 does not do. We call x^3 a *garbage* occurrence: it is something that "exists," but is no longer "used."⋆ x^6 is also inaccessible for substitution, but in addition it can be α-converted: replacing x^5 and x^6 by y does not change the meaning of ϕ_0. We say that x^6 is *classically* bound. Finally, we consider x^2, x^4 and x^5. These are "occurrences" in a purely syntactical sense only: they do not represent "files" carrying information, but just signal that incoming files labeled x should not be "unified" with outgoing files labeled x. We say that these "occurrences" are *blockers*. x^5 is not a blocker in ϕ_0 as a whole.

Contexts, in our present set-up, signal the presence of input occurrences, of blockers and of output occurrences.⋆⋆ They are abstract – in comparison with formulas – in the sense that they contain no information about the number or the place of these occurrences. Contexts can be studied independently from their connection with the logical language.

Contexts are familiar from Predicate Logic. There the context associated with a formula ϕ is simply the set X of free variables of ϕ.‡ A salient property of contexts in Predicate Logic is as follows. Suppose X is a context for ϕ, then $[\![\phi]\!]_{\mathcal{M}}$ is X-restricted, i.e., $f \in [\![\phi]\!]_{\mathcal{M}}$ and $f\mathcal{I}_X g$, implies $g \in [\![\phi]\!]_{\mathcal{M}}$. Most of the work in this section will be devoted to proving the appropriate DPL-analogue of this property of Predicate Logic.

3.2. Contexts, Considered by Themselves

In this subsection, we treat contexts as mathematical objects in their own right. The natural connection with DPL will surface in the subsequent subsections.

DEFINITION 3.1. A DPL-context is a triple $\langle I, B, O\rangle$, where I, B and O are finite sets of variables and where $I\backslash B = O\backslash B$, or, equivalently, $I \cup B = O \cup B$. The set I is the *input set*, i.e., the set on which the incoming assignments are constrained. The set O is the *output set*, i.e., the set on which the outgoing assignments are constrained. Finally, the set B is the set of blocks. This is the set of variables for which the identity between input and output value is cut through. The "block" is a barrier between past and future, breaking the link between input- and output-value. We write $\downarrow$ for *is defined* or *converges*, and $\uparrow$ for *is undefined* or *diverges*. Define:

$\Leftrightarrow \mathfrak{id} := \langle \emptyset, \emptyset, \emptyset\rangle$
$\Leftrightarrow \langle I, B, O\rangle \bullet \langle I', B', O'\rangle := \langle I \cup (I'\backslash B), B \cup B', (O\backslash B') \cup O'\rangle$
$\Leftrightarrow \langle I, B, O\rangle \to \langle I', B', O'\rangle := \langle I \cup (I'\backslash B), \emptyset, I \cup (I'\backslash B)\rangle$

⋆ The notion of *garbage* is studied in Vermeulen (1995) and Visser and Vermeulen (1996).

⋆⋆ In fact, there are good reasons also to put witnesses of *garbage* into the contexts. We will not do this in the present paper, since it is not necessary for our results here. Moreover, adding garbage leads to considerable complication of the framework and it necessitates bringing in Category Theory. We refer the reader further to Visser and Vermeulen (1996).

‡ The reader is referred to Jacobs (1991) for a category-theoretical framework appropriate for the study of contexts in Predicate Logic.

$\Leftrightarrow \langle I,B,O\rangle \le \langle I',B',O'\rangle \Leftrightarrow I \subseteq I',\ O \subseteq O',\ B \subseteq B' \subseteq B \cup (I' \cap O')$
$\Leftrightarrow$ $\cap$ is a partial operation on contexts, defined by:

$$\langle I,B,O\rangle \cap \langle I',B',O'\rangle := \begin{cases} \langle I\cap I', B\cap B', O\cap O'\rangle & \text{if} \\ & B \subseteq B' \cup (I\cap O), \\ & B' \subseteq B \cup (I'\cap O') \\ \uparrow & \text{otherwise} \end{cases}$$

We will use $\mathfrak{c}, \mathfrak{d}, \ldots$ as variables over contexts. We write $I_{\mathfrak{c}}$ for the first component of $\mathfrak{c}$, etc.

The meanings of these objects, relations and operations will become apparent in Section 3.3.

LEMMA 3.2. *The operations* $\bullet$, $\rightarrow$ *and* $\cap$ *are well defined.*

Proof. To see that $\bullet$ is well defined, note that:

$$\begin{aligned} (I \cup (I'\backslash B)) \cup (B \cup B') &= (I \cup B) \cup (I' \cup B') \\ &= (O \cup B) \cup (O' \cup B') \\ &= (O\backslash B') \cup O' \cup (B \cup B'). \end{aligned}$$

It is trivial that $\rightarrow$ is well defined. For the proof that $\cap$ is well defined we refer the reader to the Appendix. □

THEOREM 3.3. *The contexts with* $\mathfrak{id}$ *and* $\bullet$ *form a* monoid. *Moreover,* $\le$ *is a partial ordering.*

The proof of the theorem is easy. In the appendix we will show that $\cap$, if defined, is the infimum, w.r.t. $\le$.

Consider the monoid of contexts. It can be represented in an alternative way, as follows. The monoid $\mathcal{A}$ is the monoid on two generators $\mathfrak{a}$ and $\mathfrak{b}$, given by the equations: $\mathfrak{a} \bullet \mathfrak{a} = \mathfrak{a}$, $\mathfrak{b} \bullet \mathfrak{b} = \mathfrak{b}$ and $\mathfrak{b} \bullet \mathfrak{a} \bullet \mathfrak{b} = \mathfrak{b}$. The table of the monoidal operation $\bullet$ is as follows.

$\bullet$	$\mathfrak{e}$	$\mathfrak{a}$	$\mathfrak{b}$	$\mathfrak{ab}$	$\mathfrak{ba}$	$\mathfrak{aba}$
$\mathfrak{e}$	$\mathfrak{e}$	$\mathfrak{a}$	$\mathfrak{b}$	$\mathfrak{ab}$	$\mathfrak{ba}$	$\mathfrak{aba}$
$\mathfrak{a}$	$\mathfrak{a}$	$\mathfrak{a}$	$\mathfrak{ab}$	$\mathfrak{ab}$	$\mathfrak{aba}$	$\mathfrak{aba}$
$\mathfrak{b}$	$\mathfrak{b}$	$\mathfrak{ba}$	$\mathfrak{b}$	$\mathfrak{b}$	$\mathfrak{ba}$	$\mathfrak{ba}$
$\mathfrak{ab}$	$\mathfrak{ab}$	$\mathfrak{aba}$	$\mathfrak{ab}$	$\mathfrak{ab}$	$\mathfrak{aba}$	$\mathfrak{aba}$
$\mathfrak{ba}$	$\mathfrak{ba}$	$\mathfrak{ba}$	$\mathfrak{b}$	$\mathfrak{b}$	$\mathfrak{ba}$	$\mathfrak{ba}$
$\mathfrak{aba}$	$\mathfrak{aba}$	$\mathfrak{aba}$	$\mathfrak{ab}$	$\mathfrak{ab}$	$\mathfrak{aba}$	$\mathfrak{aba}$

The monoid of contexts is now given as the set of functions from Var to $\mathcal{A}$ that are on all, but finitely many arguments equal to $\mathfrak{e}$. We put: $(f \bullet g)(v) := f(v) \bullet g(v)$. A triple $\langle I, B, O\rangle$ "translates" to a function f with, e.g., $f(x) = \mathfrak{ab}$ iff $x \in I$, $x \in B$ and $x \notin O$, etc. A function f translates to a triple $\langle I, B, O\rangle$ with, e.g., $x \in I$ iff $f(x) \in \{\mathfrak{a}, \mathfrak{ab}, \mathfrak{aba}\}$, etc. It is easily seen that these "translations" give us an isomorphism of monoids between the representations. $\mathfrak{A}$ is in fact isomorphic to the monoid of contexts in the case that $\mathsf{Var} = \{x\}$. The alternative representation is possible by the fact that in our monoidal operation treats all variables "independently." It is not difficult to extend the structure on $\mathfrak{a}, \mathfrak{b}, \ldots$, to get a function representation also for $\rightarrow$, $\leq$ and $\cap$.

In this paper we will stick to the set representation, since this representation is closest to the relational notions we will need to formulate our theorem on contexts and relations – the theorem that tells us what the contexts *do*. The function representation, however, has two advantages. First, it is easier to use for doing computations "in the head." Secondly, its connection with the framework developed in (Visser and Vermeulen, 1996) to study contexts, is more perspicuous.

3.3. Contexts and Relations

We turn to the connection between contexts and relations. We show that this connection "commutes" w.r.t. $\bullet/\circ$, $\rightarrow / \rightarrow$ and $\leq/\subseteq$. We fix a non-empty domain D.

DEFINITION 3.4. Consider a relation R on D^{Var}. R is an $\langle I, B, O\rangle$*-relation* if $R = (\mathcal{I}_I \circ R \circ \mathcal{I}_O) \cap [B]$. We say that $\mathfrak{c}$ is a context *for* R, if R is a $\mathfrak{c}$-relation. R is an *IBO-relation* if R is an $\langle I, B, O\rangle$-relation for some $\langle I, B, O\rangle$. We assign to each context $\mathfrak{c}$ the property or "meaning" $[\![\mathfrak{c}]\!]_D$, the set of all $\mathfrak{c}$-relations on D^{Var}. (We will often suppress the subscript D.)

The heuristic for this property is as follows. R is $\langle I, B, O\rangle$ means that R is only concerned with the values of the incoming assignment on I; R only cares about the values of the outgoing assingment on O; all this under the constraint that in going from input to output only values of variables in B are changed. Before proving some facts about the notion introduced above, we sample some immediate insights.

$\Leftrightarrow \{\emptyset, [B_{\mathfrak{c}}]\} \subseteq [\![\mathfrak{c}]\!]$
$\Leftrightarrow [\![\langle \emptyset, B, \emptyset\rangle]\!] = \{\emptyset, [B]\}$
$\Leftrightarrow$ R is a $\langle I, \emptyset, I\rangle$-relation precisely if R is an $\langle I\rangle$-condition.

We show that $\leq$ on contexts describes the "information ordering" on contexts. The idea is simply that $\mathfrak{c}$ is *more* informative than $\mathfrak{d}$ if $[\![\mathfrak{c}]\!] \subseteq [\![\mathfrak{d}]\!]$.

THEOREM 3.5.
1. $\mathfrak{c} \leq \mathfrak{d} \Rightarrow [\![\mathfrak{c}]\!] \subseteq [\![\mathfrak{d}]\!]$.

2. *Suppose* $|D| \geq 2$. *Then:* $[\![\mathfrak{c}]\!] \subseteq [\![\mathfrak{d}]\!] \Rightarrow \mathfrak{c} \leq \mathfrak{d}$.

Proof. Let $\mathfrak{c} = \langle I, B, O\rangle$ and $\mathfrak{d} = \langle I', B', O'\rangle$. We prove (1). Let $\mathfrak{c} \leq \mathfrak{d}$ and $R \in [\![\mathfrak{c}]\!]$. We want to show: $R = (\mathcal{I}_{I'} \circ R \circ \mathcal{I}_{O'}) \cap [B']$. Trivially, R is contained in $(\mathcal{I}_{I'} \circ R \circ \mathcal{I}_{O'}) \cap [B']$. We show $(\mathcal{I}_{I'} \circ R \circ \mathcal{I}_{O'}) \cap [B'] \subseteq R$. Suppose $f'\mathcal{I}_{I'} f R g \mathcal{I}_{O'} g'$ and $f'[B']g'$. It is immediate, that $f'\mathcal{I}_I f R g \mathcal{I}_O g'$. We show that $f'[B]g'$. Suppose $v \in B' \backslash B$. Then, $v \in I' \cap O'$. We find: $f'(v) = f(v)$, since $v \in I'$. Moreover, $f(v) = g(v)$, since fRg, $R \subseteq [B]$ and $v \notin B$. Finally, $g(v) = g'(v)$, since $v \in O'$. Since R is a $\mathfrak{c}$-relation, we may conclude that $f'Rg'$.

We prove (2). We write X^c for $\mathsf{Var} \backslash X$. Let $|D| \geq 2$ and $[\![\mathfrak{c}]\!] \subseteq [\![\mathfrak{d}]\!]$. Since, $[B] \in [\![\mathfrak{c}]\!]$, we have $[B] \in [\![\mathfrak{d}]\!]$. Hence, using the facts of Section 2, we find:

$$\begin{aligned} [B] &= (\mathcal{I}_{I'} \circ [B] \circ \mathcal{I}_{O'}) \cap [B'] \\ &= [(I'^c \cup B \cup O'^c) \cap B']. \end{aligned}$$

Since $|D| \geq 2$, it follows that: $B = (I'^c \cup B \cup O'^c) \cap B'$. Now it is immediate that: $B \subseteq B' \subseteq B \cup (I' \cap O')$.

The arguments that $I \subseteq I'$ and that $O \subseteq O'$ are analogous to one another. We give the argument for the I-case. Suppose, for a reductio, that $v \in I \backslash I'$. Let d and e be distinct elements of D. We write $[v = d]$ for the test: *is f(v)=d?*. Consider the relation $R := [v = d] \circ [B]$. Clearly, $R \in [\![\mathfrak{c}]\!]$ and, so, $R \in [\![\mathfrak{d}]\!]$. Consider f with $f(v) = d$. In case $v \in B'$, we have: $f_v^e \mathcal{I}_{I'} f R g \mathcal{I}_{O'} g$. Since, as we have already shown, $B \subseteq B'$, we find: $f_v^e[B']g$. Hence, since $R \in [\![\mathfrak{d}]\!]$, we have $f_v^d R g$. Quod non. We turn to the case that $v \notin B'$. Since $I' \backslash B' = O' \backslash B'$, we find: $v \notin O'$. So: $f_v^e \mathcal{I}_{I'} f R g \mathcal{I}_{O'} g_v^e$. Since $B \subseteq B'$, we find $f_v^e[B']g_v^e$. Thus, since $R \in [\![\mathfrak{d}]\!]$, we have $f_v^d R g_v^d$. Quod non. □

Theorem 3.5 shows clearly that contexts stand in a many-many relation to relations. Thus, the question *What is the context of* R*?* has no definite answer. In the appendix we show that, but for one notable exception, every relation has a most informative context. The next lemma may be used in some cases to simplify the verification that a relation is $\langle I, B, O\rangle$.

LEMMA 3.6. *R is a $\langle I, B, O\rangle$-relation iff* $R = (\mathcal{I}_I \circ R \circ \mathcal{I}_{O \cap B}) \cap [B]$.*

Proof. It is clearly sufficient to show that for *any* $R \subseteq [B]$:

$$(\mathcal{I}_I \circ R \circ \mathcal{I}_O) \cap [B] = (\mathcal{I}_I \circ R \circ \mathcal{I}_{O \cap B}) \cap [B].$$

From left to right is immediate, since $\mathcal{I}_O \subseteq \mathcal{I}_{O \cap B}$. For the converse, suppose $f'\mathcal{I}_I f R g \mathcal{I}_{O \cap B} g'$ and $f'[B]g'$. We have to show: $g\mathcal{I}_O g'$. Consider $v \in O$. In case $v \in B$, we have $g(v) = g'(v)$. Suppose $v \in O \backslash B$. Then, also $v \in I \backslash B$. We have: $g'(v) = f'(v)$, since $v \notin B$. Moreover, $f'(v) = f(v)$, since $v \in I$. Also

* Note that, due to our design choice that in contexts $I \setminus B = O \setminus B$, the triple $\langle I, B, O \cap B\rangle$ is not generally a context.

$f(v) = g(v)$, since fRg, $R \subseteq [B]$ and $v \notin B$. Putting the identities together, we find $g'(v) = g(v)$. □

The following lemma is quite useful in applications (see Example 3.14). We write $A := I \cup B$.

LEMMA 3.7. *Suppose R is an $\langle I, B, O\rangle$-relation and $f'\mathcal{I}_I fRg$. Then there is a unique g', such that $f'Rg'\mathcal{I}_B g$. This g' has the following property: for any set of variables J, if $f'\mathcal{I}_J f$, then $g'\mathcal{I}_J g$. As a consequence, we find that $g'\mathcal{I}_I g$ and, hence, $g'\mathcal{I}_A g$. So, also, $g'\mathcal{I}_O g$.*

Proof. Let R be an $\langle I, B, O\rangle$-relation with $f'\mathcal{I}_I fRg$. Any g' with $f'Rg'\mathcal{I}_B g$, must satisfy: $f'[B]g'\mathcal{I}_B g$. So the only possible choice of such a g' is: $f' \restriction (\mathsf{Var}\backslash B) \cup g \restriction B$. We verify that g', thus defined, satisfies $f'Rg'$. It is sufficient to show that $g\mathcal{I}_O g'$. This, in its turn, follows immediately from the property in the last part of this lemma.

Consider any set J such that $f'\mathcal{I}_J f$. Suppose $v \in J$. In case $v \in B$, we have $g(v) = g'(v)$. In case $v \notin B$, we have, $g(v) = f(v)$, since $v \notin B$, Moreover, $f(v) = f'(v)$, since $v \in J$. Finally $f'(v) = g'(v)$, since $v \notin B$. Putting things together, we find $g(v) = g(v')$, as desired. □

Note that in the lemma O plays no significant role. Due to the "forward looking" and time asymmetric nature, however, of the definitions of implication and validity in DPL, it is sufficient for most applications. An immediate consequence of the lemma is that if R is an $\langle I, B, O\rangle$-relation, then $\mathsf{dom}(R)$ is an $\langle I\rangle$-condition and (by symmetry) $\mathsf{cod}(R)$ is an $\langle O\rangle$-condition.

THEOREM 3.8. *Suppose R is a $\mathfrak{c}$-relation and S is a $\mathfrak{d}$-relation. Then $R \circ S$ is a $\mathfrak{c} \bullet \mathfrak{d}$-relation.*

Proof. Let $\mathfrak{c} = \langle I, B, O\rangle$, $\mathfrak{d} = \langle I', B', O'\rangle$ and $\mathfrak{c} \bullet \mathfrak{d} = \langle I,'' B,'' O''\rangle$. It is easy to see that: $R \circ S \subseteq (\mathcal{I}_{I''} \circ (R \circ S) \circ \mathcal{I}_{O''}) \cap [B'']$ For the converse, suppose that $f'[B'']g'$, $f'\mathcal{I}_{I''} f$, $f(R \circ S)g$ and $g\mathcal{I}_{O''} g'$. We have to show: $f'(R \circ S)g'$. For some h, we have $fRhSg$. We partition Var into three sets $X_1 := O \cup I'$, $X_2 := B\backslash(O \cup I' \cup B')$ and $X_3 := \mathsf{Var}\backslash((B\backslash B') \cup O \cup I')$. Define: $h' := h\restriction X_1 \cup g'\restriction X_2 \cup f'\restriction X_3$. We show that $f'Rh'Sg'$. We first prove that $f'Rh'$. We check the conditions for applying the fact that R is an $\langle I, B, O\rangle$-relation.

1. $f'\mathcal{I}_{I''} f$ and, hence, since $I \subseteq I''$, $f'\mathcal{I}_I f$.
2. fRh.
3. $h\mathcal{I}_{O \cup I'} h'$ and, hence, $h\mathcal{I}_O h'$.
4. We show that $f'[B]h'$. Consider a variable v not in B. We have to show $f'(v) = h'(v)$. We can only run into trouble in case v is not in X_3, i.e., if v is in $B\backslash B'$ or in $O \cup I'$. The first possibility is excluded, by the fact that v is not in B. Suppose v is in $O \cup I'$. Then: $h'(v) = h(v)$, by definition. R is

an $\langle I, B, O\rangle$-relation, so $R \subseteq [B]$. We may conclude that $h(v) = f(v)$, since $v \notin B$. In case $v \in O$, we find: $v \in O\backslash B = I\backslash B \subseteq I \subseteq I''$. In case $v \in I'$, we find: $v \in I'\backslash B \subseteq I''$. So in both cases: $v \in I''$. Since $f'\mathcal{I}_{I''}f$, it follows that $f(v) = f'(v)$. Composing the identities, we find $h'(v) = f'(v)$, as desired.

By (1–4), we may conclude that $f'Rh'$. Next, we check the conditions for applying the fact that S is an $\langle I', B', O'\rangle$-relation.

1. We have $h\mathcal{I}_{O\cup I'}h'$ and, hence, $h'\mathcal{I}_{I'}h$.
2. hSg.
3. We have $g\mathcal{I}_{O''}g'$ and, hence, since $O' \subseteq O''$, $g\mathcal{I}_{O'}g'$.
4. We show that $h'[B']g'$. Consider a variable v not in B'. We have to show $h'(v) = g'(v)$. Inspecting the definition of h', we see that our desired identity can only fail if either $v \notin B$ or $v \in B$ and $v \in O \cup I' \cup B'$. We consider the case that $v \notin B$. We already showed that $f'[B]h'$. We assumed $f'[B'']g'$. Hence: $h'[B]f'[B \cup B']g'$. So, $h'[B \cup B']g'$. Since $v \notin B \cup B'$, we find: $h'(v) = g'(v)$. Next, we consider the case that $v \in B$ and $v \in O \cup I' \cup B'$. Since, we choose v outside of B', we need only consider the possibility that $v \in O \cup I'$. We have, by definition, $h'(v) = h(v)$. Since $v \notin B'$, we find $h(v) = g(v)$. By our early assumption, $g\mathcal{I}_{O''}g'$, where $O'' = (O\backslash B') \cup O'$. If $v \in O$, then clearly $v \in O\backslash B'$, and we find $g(v) = g'(v)$. If $v \in I'$, we have $v \in I'\backslash B' = O'\backslash B' \subseteq O'$, hence, again, $g(v) = g'(v)$. Putting the identities together, we find $h'(v) = g'(v)$, as desired.

By (1–4), we have: $h'Sg'$. □

THEOREM 3.9. *Suppose R is a $\mathfrak{c}$-relation and S is a $\mathfrak{d}$-relation. Then $R \to S$ is a $\mathfrak{c} \to \mathfrak{d}$-relation.*

Proof. Suppose R is a $\mathfrak{c}$-relation and S is a $\mathfrak{d}$-relation. Let $\mathfrak{c} = \langle I, B, O\rangle$, $\mathfrak{d} = \langle I', B', O'\rangle$ and $(\mathfrak{c} \to \mathfrak{d}) = \langle I,'' \emptyset, I''\rangle$. Trivially, $(R \to S) \subseteq [\emptyset]$. Moreover, since $\mathsf{id} \subseteq \mathcal{I}_{I''}$, $(R \to S) \subseteq (\mathcal{I}_{I''} \circ (R \to S) \circ \mathcal{I}_{I''})$

To prove the converse, suppose $f'\mathcal{I}_{I''}f$ and $f(R \to S)f$. We have to show that $f'(R \to S)f'$. Suppose $f'Rg'$. By Lemma 3.7, there is a g such that $fRg\mathcal{I}_{I''\cup B}g'$. It follows that $g'\mathcal{I}_{I'}g$. Since $f(R \to S)f$, we can find an h, such that gSh. Again applying Lemma 3.7, we find an h' with $g'Sh'$. □

We close this subsection with a language-free soundness result.

DEFINITION 3.10. A relation on D^{Var} is DPL-expressible over D iff it can be generated by composition from resettings $[v]$ and finitely restricted conditions over D.

THEOREM 3.11. *Every DPL-expressible relation over D is an IBO-relation.*

The proof is an obvious induction on the way the relation is generated.

3.4. CONTEXTS AND LANGUAGE

We turn to our discussion of how contexts are connected to formulas.

DEFINITION 3.12. We assign to every DPL-formula ϕ a context $\mathfrak{c}_\phi$. Define:

- $\Leftrightarrow \mathfrak{c}_{P(r_1,\cdots,r_n)} = \langle V, \emptyset, V\rangle$, where $V = \{r_1, \cdots, r_n\} \cap \mathsf{Var}$
- $\Leftrightarrow \mathfrak{c}_\top = \mathfrak{c}_\bot = \langle \emptyset, \emptyset, \emptyset\rangle$, $\mathfrak{c}_{r=s} = \langle \{r, s\} \cap \mathsf{Var}, \emptyset, \{r, s\} \cap \mathsf{Var}\rangle$, $\mathfrak{c}_{\exists v} = \langle \emptyset, \{v\}, \emptyset\rangle$
- $\Leftrightarrow \mathfrak{c}_{\phi.\psi} = \mathfrak{c}_\phi \bullet \mathfrak{c}_\psi$ and $\mathfrak{c}_{(\phi\to\psi)} = (\mathfrak{c}_\phi \to \mathfrak{c}_\psi)$.

We write I_ϕ for $I_{\mathfrak{c}_\phi}$, etc.

Note that the definition correctly defines a function, by the associativity of $\bullet$. We now prove the main theorem of this section.

THEOREM 3.13. *For every formula ϕ, $[\phi]$ is a $\mathfrak{c}_\phi$-relation on D^{Var}.*

Proof. The proof is by induction on ϕ using Theorems 3.8 and 3.9. The atomic cases are easy. □

We obtain the following picture of the way contexts work: ϕ is mapped to $\mathfrak{c}_\phi$ by abstracting both from part of the vocabulary and part of the structure. $\mathfrak{c}_\phi$ is mapped to $[\![\mathfrak{c}_\phi]\!]_D$, a property of relations. Via a different route ϕ is mapped to the relation $[\phi]_\mathcal{M}$. The two routes are connected by the theorem that $[\phi]_\mathcal{M} \in [\![\mathfrak{c}_\phi]\!]_D$.

Note that $\mathfrak{c}_\phi$ is not always the $\leq$-minimal context of $[\phi]_\mathcal{M}$, as is witnessed by the fact that $\mathfrak{c}_{x=x} = \langle \{x\}, \emptyset, \{x\}\rangle$ and that $[x = x]_\mathcal{M}$ is the identity on D^{Var}, thus admitting the context $\langle \emptyset, \emptyset, \emptyset\rangle$.

Example 3.14. We provide two examples of how Theorem 3.13 in combination with Lemma 3.7 can be used to verify a valid principle for DPL. We first prove:

$$\chi.\phi \models \phi, \text{ if } B_\phi \cap I_\phi = \emptyset.$$

Suppose $B_\phi \cap I_\phi = \emptyset$ and $f[\chi.\phi]g$. We have to produce an h with $g[\phi]h$. Since $f[\chi.\phi]g$, we can find a j with $j[\phi]g$. By Theorem 3.13: $j[B_\phi]g$. Since $B_\phi \cap I_\phi = \emptyset$, we find: $g\mathcal{I}_{I_\phi}j$. So, $g\mathcal{I}_{I_\phi}j[\phi]g$. By Theorem 3.13 and Lemma 3.7 we can find an h with $g[\phi]h$.

As a second example we prove:

$$\chi \models \phi.\psi \Rightarrow \chi \models \psi, \text{ if } B_\phi \cap I_\psi = \emptyset.$$

Suppose $B_\phi \cap I_\psi = \emptyset$, $\chi \models \phi.\psi$ and $f[\chi]g$. We have to produce an h with $f[\psi]h$. By our assumptions, there are i and j such that $f[\phi]i[\psi]j$. Hence, $f[B_\phi]i$ and so $f\mathcal{I}_{I_\psi}i$. Thus, $f\mathcal{I}_{I_\psi}i[\psi]j$. We may conclude that there is an h with $f[\psi]h$.

The examples demonstrate the role contexts must play in the formulation of schematic principles for DPL.

4. The Switching Property

In Section 3 we introduced contexts or *IBO*s as properties of relations and showed that every $[\phi]$ is an *IBO*-relation. A first conjecture for characterizing the DPL-expressible relations would be that these are precisely the *IBO*-relations. We will see, however, that this conjecture is false. To characterize the DPL-expressible relations we need one extra property: the Switching Property. In the present section we will prove that the DPL-expressible relations *do* have the Switching Property (*soundness*). In Section 5 we will show that every *IBO*-relation on an infinite domain that has the Switching Property is DPL-expressible (*completeness*).

DEFINITION 4.1. A relation R on D^{Var} has the *switching property* if it is either a condition or there are variables x and y (not necessarily distinct), such that $R = \mathsf{dom}(R) \circ [x] \circ R \circ [y] \circ \mathsf{cod}(R)$. If the second case obtains, we call the variables x and y involved a pair of *switching variables*. There might be more than one pair of switching variables.

There are various other ways to define the Switching Property, but, I submit, the one presented here is the most natural one. In the lemma below, we collect some helpful insights.

LEMMA 4.2. *Suppose R is a relation on D^{Var}*

1. $R = \mathsf{dom}(R) \circ R \circ \mathsf{cod}(R)$.
2. *Suppose R is $C \circ T \circ C'$, where C and C' are conditions and where T is a relation. Then* $\mathsf{dom}(R) \circ C = \mathsf{dom}(R)$ *and* $C' \circ \mathsf{cod}(R) = \mathsf{cod}(R)$.
3. *Suppose C is a condition. Then,* $C \circ [x] \circ C \circ [x] = C \circ [x]$ *and* $[x] \circ C \circ [x] \circ C = [x] \circ C$.

The easy proof is left to the reader.

THEOREM 4.3. *Every* DPL-*expressible relation over D has the switching property.*

Proof. In case R is a condition, we are done. Suppose R is not a condition. As is easily seen, R must be of the form $C \circ [x] \circ S \circ [y] \circ C'$, for some variables x, y, some conditions C, C', and some relation S. (In a formula ϕ defining R, x would correspond to the first existential quantifier occurring in ϕ, y to the last. Note that we allow x and y to be the same variable and even the first and last existential quantifier occurrence to be the same occurrence.) We have, using Lemma 4.2:

$$\begin{aligned}
\mathsf{dom}(R)\circ[x]\circ C\circ[x]\circ S\circ[y]\circ C'\circ[y]\circ\mathsf{cod}(R) &= \\
\mathsf{dom}(R)\circ C\circ[x]\circ C\circ[x]\circ S\circ[y]\circ C'\circ[y]\circ C'\circ\mathsf{cod}(R) &= \\
\mathsf{dom}(R)\circ C\circ[x]\circ S\circ[y]\circ C'\circ\mathsf{cod}(R) &= \\
\mathsf{dom}(R)\circ R\circ\mathsf{cod}(R) &= R.
\end{aligned}$$

□

The results of Section 3 and of this section combine to the obvious "soundness"-result:

THEOREM 4.4. *Every* DPL*-expressible relation over D is an IBO-relation with the Switching Property.*

Example 4.5. We show how to use the Switching Property to prove that certain relations are not DPL-expressible. Suppose $|D| \geq 2$.

- ⇔ Let $R := [x := y, y := x]$, where
  $f[x := y, y := x]g :\Leftrightarrow g = f_{x,y}^{f(y),f(x)}$.
  R is an *IBO*-relation, with context $\langle\{x,y\},\{x,y\},\{x,y\}\rangle$. Suppose R has the Switching Property. R is evidently not a condition. Let v, w be a pair of switching variables. Let fRg, $f(v) = d$, and $d \neq e$. Using the fact that the domain of R is the set of all assignments, we find: $f_v^e(\mathsf{dom}(R))f_v^e[v]fRg[w]g(\mathsf{cod}(R))g$. Hence, by the switching property: f_v^eRg. But R is obviously injective. So we have a contradiction.
- ⇔ Suppose our model $\mathcal{M}$ is the usual structure of the natural numbers. Let $S := [x := x + 1]$, where $f[x := x + 1]g :\Leftrightarrow g = f_x^{f(x)+1}$. S is an *IBO*-relation with context $\langle\{x\},\{x\},\{x\}\rangle$. S does not have the Switching Property since: S is not a condition; S has as domain the set of all assignments; S is injective.
- ⇔ Let $T := [(\exists x \vee \exists y)]$, where $[(\exists x \vee \exists y)] := [\exists x] \cup [\exists y]$. T is an *IBO*-relation. The best possible context for it is $\langle\{x,y\},\{x,y\},\{x,y\}\rangle$.* Suppose that T has the Switching Property. T is not a condition, so we can find switching variables v and w. By symmetry we may assume that $v \neq x$. We can find f and g with fTg and $f(x) \neq g(x)$. Choose d with $d \neq g(v)$. Using the fact that the domain of T is the set of all assignments, we find:

  $$f_v^d(\mathsf{dom}(T))f_v^d[v]fTg[w]g(\mathsf{cod}(T))g.$$

  By the Switching Property: f_v^dTg. But we have two distinct variables x and v such that $f_v^d(x) = f(x) \neq g(x)$ and $f_v^d(v) = d \neq g(v)$. This is clearly impossible.

---

* We will discuss this phenomenon in more detail in Section 6.3.

5. The DPL-Expressible Relations on an Infinite Domain

In Sections 3 and 4 we have seen that the DPL-expressible relations are *IBO*-relations with the Switching Property. Here we show the converse – for the case that the domain, D, is infinite.

THEOREM 5.1. *Let D be an* infinite *set. Then the* DPL-*expressible relations over D are precisely the IBO-relations with the switching property on D^{Var}.*

Proof. One direction is immediate by our previous results. Let R be an $\langle I, B, O\rangle$-relation with the switching property. In case R is a condition we are done. Suppose R is not a condition and let x, y be a pair of switching variables. By the switching property

$$R = \mathsf{dom}(R) \circ [x] \circ R \circ [y] \circ \mathsf{cod}(R).$$

Note that dom(R) is an $\langle I\rangle$-condition and that cod(R) is an $\langle O\rangle$-condition. Thus, it is sufficient to show that $[x] \circ R \circ [y]$ is DPL-expressible. $[x] \circ R \circ [y]$ is an $\langle I\backslash\{x\}, B, O\backslash\{y\}\rangle$-relation, where $x, y \in B$. After renaming, we see that it is sufficient to prove that any $\langle I, B, O\rangle$-relation R, with $x, y \in B$ and $x \notin I$ and $y \notin O$ is DPL-expressible.

We will assume $x \neq y$. In case $x = y$, the proof is simpler. To increase readability, we will specify R in a DPL-language that we introduce along the way. Suppose $I = \{i_1, \cdots, i_m\}$, $B\backslash O = \{b_1, \cdots, b_n\}$ and $O \cap B = \{o_1, \cdots, o_p\}$. Here the i_k are supposed to be mutually distinct and similarly for the other sets. Since D is infinite there is a coding of finite sequences of elements of D in D. Par abus de langage, we will confuse this coding with our ordinary sequences of elements of D. Our language has an $(m+1)$-ary predicate symbol P, where:

$$\langle d_1, \cdots, d_m, e\rangle \in I(P) \Leftrightarrow \exists f, g\ \ fRg \quad \text{and} \quad f(i_1) = d_1, \ldots, f(i_m) = d_m$$
$$\text{and} \quad e = \langle g(o_1), \cdots, g(o_p)\rangle.$$

Remember that $[y := x]$ is short for $\exists y.y = x$. The formula ϕ is given by:

$$\exists x.P(i_1, \cdots, i_m, x).[y := x].\exists o_1.\cdots.\exists o_p.(y = \langle o_1, \cdots, o_p\rangle).\exists b_1.\cdots.\exists b_n.$$

Here "$y = \langle o_1, \cdots, o_p\rangle$" stands for the obvious condition. Note that $[\phi]$ is an $\langle I, B, O\rangle$-relation. We claim that $R = [\phi]$. Suppose first that fRg. Take:

$$\Leftrightarrow h_1 := f_x^{\langle g(o_1), \cdots, g(o_p)\rangle}.$$

Remember that $x \notin \{i_1, \cdots, i_m\}$. We have:

$$h_1[P(i_1, \cdots, i_m, x)]h_1 \ \Leftrightarrow\ \langle h_1(i_1), \cdots, h_1(i_m), h_1(x)\rangle \in I(P)$$
$$\Leftrightarrow\ \langle f(i_1), \cdots, f(i_m), \langle g(o_1), \cdots, g(o_p)\rangle\rangle \in I(P).$$

Clearly, f and g witness that $\langle f(i_1), \cdots, f(i_m), \langle g(o_1), \cdots, g(o_p)\rangle\rangle$ is in $I(P)$. Next we set:

$\Leftrightarrow h_2 := (h_1)_y^{\langle g(o_1),\cdots,g(o_p)\rangle}$
$\Leftrightarrow h_3 := (h_2)_{o_1,\ldots,o_n}^{g(o_1),\ldots,g(o_n)}$
$\Leftrightarrow h_4 := (h_3)_{b_1,\ldots,b_n}^{g(b_1),\ldots,g(b_n)}$.

We find (using $y \notin \{o_1, \cdots, o_p\}$):

$$f[\exists x]h_1 \;\; h_1[P(i_1, \cdots, i_m, x)]h_1 \;\; h_1[[y := x]]h_2$$
$$h_2[\exists o_1. \cdots .\exists o_p]h_3 \;\; h_3[y = \langle o_1, \cdots, o_p\rangle]h_3 \;\; h_3[\exists b_1. \cdots .\exists b_n]h_4.$$

Since $f[B]g$, it is easy to see that $g = h_4$.

For the converse, suppose $f'[\phi]g'$. Let h_1, h_2, h_3, be such that:

$$f'[\exists x]h_1 \;\; h_1[P(i_1, \cdots, i_m, x)]h_1 \;\; h_1[[y := x]]h_2$$
$$h_2[\exists o_1. \cdots .\exists o_p]h_3 \;\; h_3[y = \langle o_1, \cdots, o_p\rangle]h_3 \;\; h_3[\exists b_1. \cdots .\exists b_n]g'.$$

We are going to apply the fact that R is an $\langle I, B, O\rangle$-relation. By the fact that $h_1[P(i_1, \cdots, i_m, x)]h_1$ and by the definition of P, we can find f and g such that fRg, $f(i_1) = h_1(i_1)$,... $f(i_m) = h_1(i_m)$, and $\langle g(o_1), \cdots, g(o_p)\rangle = h_1(x)$. Since $x \notin \{i_1, \cdots, i_m\}$ and $f'[x]h_1$, it follows that $f'\mathcal{I}_I f$. Collecting what we have, we see:

$\Leftrightarrow fRg.$
$\Leftrightarrow f'[B]g'$
$\Leftrightarrow f'\mathcal{I}_I f.$

By Lemma 3.6 we need to check:

$\Leftrightarrow g\mathcal{I}_{O\cap B}g'.$

Consider $v \in O \cap B$. We have: $h_1(x) = \langle g(o_1), \cdots g(o_p)\rangle$ and, hence, $h_2(y) = \langle g(o_1), \cdots g(o_p)\rangle$. We have $y \notin O$, and, thus, we get: $h_3(y) = \langle g(o_1), \cdots g(o_p)\rangle$. Since $v \in \{o_1, \cdots, o_p\}$ and $h_3[y = \langle o_1, \cdots, o_p\rangle]h_3$, we find: $h_3(v) = g(v)$. Finally, v is not among the $b_1, \cdots, b_n$, and thus: $g'(v) = g(v)$.

Putting the itemized insights together, we may conclude: $f'Rg'$. □

6. Extensions of the DPL-Language

We consider three extensions of the DPL-language. One with conjunction interpreted as intersection of relations, one with a new quantifier ∃ and one with disjunction interpreted as union of relations. We will show that our contexts work for each of these extensions. The contexts provided for disjunction are not optimally informative and intuitively queer, however. We will give some hints on how we think this apparent defect should be repaired. We show that $\exists$ is definable using $\wedge$ and that in the system with $\exists$ all *IBO*-relations are expressible.

6.1. CONJUNCTION

We study the effect of adding intersection of relations to the DPL-repertoire. One way of of thinking about $R \cap S$ is as: *reset simultaneously via R and via S, and compare the results. If they are equal, make the output of our new relation the shared output, otherwise abort.* At the syntactical level, we reflect the new operation by extending the language of DPL by adding the clause:

$\Leftrightarrow$ If $\phi, \psi \in \mathcal{L}$, then $(\phi \wedge \psi) \in \mathcal{L}$.

We will call the new language: $\mathcal{L}(\wedge)$. The semantic clause is: $[\phi \wedge \psi] := [\phi] \cap [\psi]$. We define intersection of contexts as follows.

$\Leftrightarrow$ $\langle I, B, O\rangle \wedge \langle I', B', O'\rangle :=$
$\langle I \cup I' \cup (O \cup O')\backslash(B \cap B'), B \cap B', O \cup O' \cup (I \cup I')\backslash(B \cap B')\rangle$.

Note that:

$$\begin{aligned} I \cup I' \cup (O \cup O')\backslash(B \cap B') \cup (B \cap B') &= I \cup I' \cup O \cup O' \cup (B \cap B') \\ &= O \cup O' \cup \\ &\quad (I \cup I')\backslash(B \cap B') \cup (B \cap B'). \end{aligned}$$

So $\wedge$ is a well-defined operation on contexts. We define:

$\Leftrightarrow$ $\langle I, B, O\rangle \preceq \langle I', B', O'\rangle :\Leftrightarrow I' \subseteq I$ and$B \subseteq B'$ and $O' \subseteq O$.

Note the difference between $\preceq$ and $\leq$. It is easy to see that $\wedge$ is precisely the *infimum* with respect to $\preceq$.

THEOREM 6.1. *Let R be a $\mathfrak{c}$-relation and let S be a $\mathfrak{d}$-relation. Then, $R \cap S$ is a $(\mathfrak{c} \wedge \mathfrak{d})$-relation.*

Proof. Suppose that the conditions of the theorem are fulfilled. Let $\mathfrak{c} = \langle I, B, O\rangle$, $\mathfrak{d} = \langle I', B', O'\rangle$, $(\mathfrak{c} \wedge \mathfrak{d}) = \langle I,'' B,'' O''\rangle$. Suppose $f(R \cap S)g$. Clearly, $f\mathcal{I}_{I''}f(R \cap S)g\mathcal{I}_{O''}g$. We may conclude that $f([B] \cap [B'])g$, i.e., $f[B \cap B']g$. So: $(R \cap S) \subseteq (\mathcal{I}_{I''} \circ (R \cap S) \circ \mathcal{I}_{O''}) \cap [B'']$.

For the converse, suppose $f'\mathcal{I}_{I''}f$, $f(R \cap S)g$, $g\mathcal{I}_{O''}g'$, and $f'[B'']g'$. Evidently, $f'\mathcal{I}_I f$, fRg, $g\mathcal{I}_O g'$, and $f'[B]g'$. Ergo $f'Rg'$. Similarly $f'Sg'$. Hence $f'(R \cap S)g'$. □

We extend the definition of $\mathfrak{c}_\phi$ to the new language by adding the clause $\mathfrak{c}_{(\phi \wedge \psi)} := \mathfrak{c}_\phi \wedge \mathfrak{c}_\psi$. Theorem 6.1 immediately yields the next theorem.

THEOREM 6.2. $[\phi]$ *is a* $\mathfrak{c}_\phi$*-relation for every* $\phi \in \mathcal{L}(\wedge)$.

We consider an example. Let $\mathfrak{c}_\phi = \langle I, B, O\rangle$. Suppose $B = \{b_1, \cdots, b_n\}$. Let $\psi := (\phi.\exists b_1.\cdots.\exists b_n \wedge \top)$. We have: $[\psi] = [\neg\neg(\phi)] = [(\top \to \phi)]$. We compute $\mathfrak{c}_\psi$.

$$\begin{aligned}\mathfrak{c}_\psi &= (\langle I, B, O\rangle \bullet \langle \emptyset, B, \emptyset\rangle) \cap \langle \emptyset, \emptyset, \emptyset\rangle \\ &= \langle I, B, O\backslash B\rangle \cap \langle \emptyset, \emptyset, \emptyset\rangle \\ &= \langle I \cup \emptyset \cup (O\backslash B\backslash\emptyset), \emptyset, O\backslash B \cup (I\backslash\emptyset)\rangle \\ &= \langle I, \emptyset, I\rangle.\end{aligned}$$

Thus, in this example our conjunction on contexts gives us the "intuitive result," i.e., $[\psi]$ *is an* I*-condition.*

Let us say that the DPL($\wedge$)-relations over a given domain D are the relations on this domain generated by finitely restricted conditions and resettings using composition and intersection. The results of the present section show that the DPL($\wedge$)-relations are all *IBO*-relations. The results of the next section, will imply that, conversely, every *IBO*-relation is DPL($\wedge$).

6.2. A New Existential Quantifier

We define: $\exists x(R) := ([x] \circ R \circ [x]) \cap \mathcal{I}_{\{x\}}$. In case R is an $\langle I, B, O\rangle$-relation, we see that $\exists x(R) := ([x] \circ R \circ [x]) \cap [B\backslash\{x\}]$. It follows – by the result of Section 6.1 – that $\exists x(R)$ is an $\langle I\backslash\{x\}, B\backslash\{x\}, O\backslash\{x\}\rangle$-relation.

We extend the language of DPL by adding the clause:

$\Leftrightarrow$ If $\phi \in \mathcal{L}$ and $x \in \mathsf{Var}$ then $\exists x(\phi) \in \mathcal{L}$.

Note the overloading of notations. The new language will be $\mathcal{L}(\exists)$. The new semantical clause is the obvious: $[\exists x\phi]_{\mathcal{M}} = \exists x[\phi]_{\mathcal{M}}$. $\exists x$ is definable in $\mathcal{L}(\wedge)$ as follows. Suppose $\mathfrak{c}_\phi = \langle I, B, O\rangle$. Let $B\backslash\{x\} = \{b_1, \cdots, b_n\}$, then we can put $(\exists x.\phi.\exists x \wedge \exists b_1.\cdots.\exists b_n)$ for $\exists x(\phi)$.

THEOREM 6.3. *For* any *non-empty domain* D*, the* DPL*(*$\exists$*)-expressible relations are precisely the IBO-relations.*

Proof (Sketch). We have already seen that every DPL($\exists$)-expressible relation is *IBO*.

For the converse, suppose that R is an $\langle I, B, O\rangle$-relation. Let $I = \{i_1, \cdots, i_m\}$, $B\backslash O = \{b_1, \cdots, b_n\}$ and $O \cap B = \{o_1, \cdots, o_p\}$. Here the i_k are supposed to be mutually distinct and similarly for the other sets. Take a DPL($\exists$)-language with an $(m + p)$-ary predicate symbol P, where:

$$\begin{aligned}\langle d_1, \cdots, d_m, e_1, \cdots, e_p\rangle \in I(P) \ \Leftrightarrow\ \exists f, g\ \ fRg \quad &\text{and} \\ f(i_1) = d_1, \ldots, f(i_m) = d_m \quad &\text{and} \\ f(o_1) = e_1, \ldots, f(o_p) = e_p. &\end{aligned}$$

Let $u_1 \ldots u_p$ be variables disjoint from $I \cup B \cup O$. Let ϕ be given by:

$$\exists u_1(\cdots(\exists u_p(P(i_1,\cdots,i_m,u_1,\cdots,u_p).\exists o_1.\cdots.\exists o_p.$$
$$o_1 = u_1.\cdots.o_p = u_p.\exists b_1.\cdots.\exists b_n)\cdots).$$

Clearly, that $[\phi]$ is an $\langle I, B, O\rangle$-relation. The verification that $R = [\phi]$ is along the lines of the proof of Theorem 5.1. □

Example 6.4. We show how to define the three relations of Example 4.5. We do a bit more than the theorem promises, because we give explicit descriptions of the predicate "P" employed in the proof of Theorem 6.3.

⇔ $[x := y, y := x]$ can be defined by:

$$\exists u(x = u.\exists x.x = y.\exists y.y = u).$$

Note that this gives us the expected context: $\langle\{x,y\},\{x,y\},\{x,y\}\rangle$.

⇔ $[x := x+1]$ can be defined by: $\exists v(x = v.\exists x.x = v+1)$. (Strictly speaking we are working in a relational language, so $x = v+1$ is a suggestive notation for, say, $S(v,x)$.) The context produced by the formula is as expected.

⇔ We can define $[(\exists x \vee \exists y)]$ by:

$$\exists u(\exists v(\neg(\neg(x = u).\neg(y = v)).\exists x.\exists y.x = u.y = v)).$$

This gives us the context $\langle\{x,y\},\{x,y\},\{x,y\}\rangle$. We will discuss this context in the next subsection.

Since $\exists$ is definable using $\wedge$, the "expressive completeness" of $\exists$ w.r.t. the *IBO*-relations implies the "expressive completeness" of $\wedge$. Finally, we can translate Predicate Logic into DPL($\wedge$), by changing the $\exists$-clause of our earlier translation to: $(\exists x(\phi))^* := \exists x(\phi^*)$. Remarkably, the old and the new translation produce precisely the same relations at the semantical level.

Operations like $\rightarrow$, $\wedge$ and $\exists$ are not themselves *actions* in the sense of our semantics. They are transformers of actions. Yet there is a tendency to understand $\exists x(\phi)$ dynamically as a sequence of actions: *reset x; execute ϕ; set x back to its original value*. The problem with this way of viewing things is summarized with the question: where do we store the original value of x, so that we can restore it at the end? DPL-semantics does not supply the right kind of "memory" to realize $\exists x(\phi)$ as a sequence of actions. We can do that (or, rather, something very much like it) in the richer semantics of Vermeulen's DPLE (see Vermeulen, 1993), where under a *variable name* we do not store just one value, but a stack of values. Here the original value of x is simply stored "under" the new one.

6.3. Disjunction

In this section we have a brief look at the problem of adding disjunction/union to DPL. Adding disjunction/union evokes problems that are definitely beyond the scope of the present paper. So we can only offer some tantalizing remarks.

One way of of thinking about $R \cup S$ is as: *Choose between R and S, and reset via the relation chosen*. At the syntactical level, we reflect the new operation by extending the language of DPL by adding the clause:

⇔ If $\phi, \psi \in \mathcal{L}$, then $(\phi \vee \psi) \in \mathcal{L}$.

We will call the new language: $\mathcal{L}(\vee)$. The semantic clause is: $[\phi \vee \psi] := [\phi] \cup [\psi]$.

What could be a context for $[x] \cup [y]$? Some experimentation shows that the best we can do is: $\langle \{x, y\}, \{x, y\}, \{x, y\} \rangle$. This seems a wasteful way to represent the variable handling of this relation. Our intuition tells us that $[x] \cup [y]$ is a pure resetter and not something that "constrains" w.r.t. x and y. The resetting part of our contexts is somehow too crude to represent "choice" well. The example does not tell us that in any strict sense our present framework is wrong. It just suggests that, possibly, we could do better. We might try out richer notions of context. The most obvious proposal is to take as a context in the new sense a set of contexts in the old sense, where the set is given "disjunctive reading." So, e.g., we would have:

$$\mathfrak{c}_{(\exists x \vee \exists y).P(x,y)} = \{\langle \{y\}, \{x\}, \{x, y\} \rangle, \langle \{x\}, \{y\}, \{x, y\} \rangle\}.$$

Note that, e.g., the second occurrence of x in $(\exists x \vee \exists y).P(x, y)$ seems to be ambiguous between free and actively bound. So what is an ambiguous occurrence and how do we handle it theoretically? We propose to addres this question elsewhere.

7. Concluding Remarks

In this paper we introduced a notion of context and specified its connections with relational semantics and language. We used these contexts to prove a characterization of the *DPL*-expressible relations. Moreover, we illustrated the usefulness of contexts both in formulating and in verifying valid sequents of DPL. We illustrated the fact that "understanding of what is going on" is not automatically preserved if we extend the DPL-language. E.g., adding disjunction leads to ambiguous occurrences of variables. This observation tells us that the study of extensions will provide us clues regarding the question: *what is it to be a variable occurrence of a certain kind?*

So – apart from the concrete results – what general conclusions may we draw from the paper? A first one is, surely, that a study of the elementaria of DPL is both necessary and rewarding. Questions on the nature of variable occurrences, the proper notion of syntactic substitution, etcetera, appear in a new light. The fruitfulness of the study of DPL is independent of the question whether DPL is really the best choice as a medium for representing dynamic phenomena. One reason is that, in a sense, the relational semantics of DPL is very simple and that it is, therefore, easier to make progress. A second conclusion is that it is rewarding to engage in a study that stresses the *differences* between DPL and Predicate Logic. Much effort has gone into integrating DPL into the classical Montague framework. This project has unavoidably a conservative flavour. The result has been that the unfamiliarity, the strangeness of DPL has been underadvertised. Precisely mastering the strangeness provides us with better insight into the formerly familiar notions. My third conclusion is simply: contexts are essential in the study of *DPL* and its kin. We may want to vary the contexts, e.g., we may want to add "garbage elements" or to ignore the O-component, but contexts per se are there to stay. Our third conclusion points to a larger programmatic point: the study of contexts and the way they are contexts of their contents should be one of the central endeavors of the study of Information Processing and Dynamics.

Acknowledgements

I thank Kees Vermeulen for many enlightening conversations and for his careful reading of two versions of this paper. I am grateful to the students of various classes in Dynamic Logic both in Amsterdam and in Utrecht for providing a critical forum for some of my ideas on DPL and Dynamics. I thank the two referees for their acute observations.

Appendix

A. Intersection of Contexts

In this appendix we treat the properties of intersection of contexts.

LEMMA A.1. $\cap$ *is well defined.*

Proof. We have to prove: $(I \cap I')\backslash(B \cap B') = (O \cap O')\backslash(B \cap B')$ on the assumption that $B \subseteq B' \cup (I \cap O)$ and $B' \subseteq B \cup (I' \cap O')$. We prove

$$(I \cap I')\backslash(B \cap B') \subseteq (O \cap O')\backslash(B \cap B').$$

The converse direction is dual. Suppose $v \in I \cap I'$ and $v \notin B \cap B'$. We want to prove: $v \in O \cap O'$. We have: $v \notin B$ or $v \notin B'$. Suppose $v \notin B$. It follows that $v \in I\backslash B$ and, hence, $v \in O\backslash B$, so $v \in O$. If $v \notin B'$, we find $v \in O'$, and we are done. Suppose $v \in B'$. We assumed that $B' \subseteq B \cup (I' \cap O')$. Since $v \notin B$, we get: $v \in I' \cap O'$, and, hence, $v \in O'$. The case that $v \notin B'$ is similar. □

Next we prove that $\cap$ produces the $\leq$ minimum, whenever there is one.

THEOREM A.2.

1. $\mathfrak{z} \leq \mathfrak{c}$ *and* $\mathfrak{z} \leq \mathfrak{d} \Leftrightarrow \mathfrak{c} \cap \mathfrak{d} \downarrow,\ \mathfrak{z} \leq \mathfrak{c} \cap \mathfrak{d}$.

2. $\mathfrak{c} \cap \mathfrak{d}$, *whenever it exists, is the infinimum of* $\mathfrak{c}$ *and* $\mathfrak{d}$.

Proof. Let $\mathfrak{c} = \langle I, B, O\rangle$, $\mathfrak{d} = \langle I', B'O'\rangle$ and $\mathfrak{z} = \langle J, C, P\rangle$. Suppose $\mathfrak{z} \leq \mathfrak{c}$ and $\mathfrak{z} \leq \mathfrak{d}$. We have: $J \subseteq I \cap I'$ and $P \subseteq O \cap O'$ and $C \subseteq B \cap B'$ and $B \subseteq C \cup (I \cap O)$ and $B' \subseteq C \cup (I' \cap O')$. It follows that $B \subseteq (B \cap B') \cup (I \cap O)$, and, thus, $B \subseteq B' \cup (I \cap O)$. Moreover, $B' \subseteq (B \cap B') \cup (I' \cap O')$, and so $B' \subseteq B \cup (I' \cap O')$. We may conclude that $\mathfrak{c} \cap \mathfrak{d} \downarrow$, $\mathfrak{c} \cap \mathfrak{d} \leq \mathfrak{c}$ and $\mathfrak{c} \cap \mathfrak{d} \leq \mathfrak{d}$.

For the converse, it is sufficient to prove that if $\mathfrak{c} \cap \mathfrak{d} \downarrow$, then $\mathfrak{c} \cap \mathfrak{d} \leq \mathfrak{c}$ and $\mathfrak{c} \cap \mathfrak{d} \leq \mathfrak{d}$. So, suppose $\mathfrak{c} \cap \mathfrak{d} \downarrow$. We verify $\mathfrak{c} \cap \mathfrak{d} \leq \mathfrak{c}$, the case of $\mathfrak{d}$ being similar. The only problematic case to check is: $B \subseteq (B \cap B') \cup (I \cap O)$. But this is immediate from: $B \subseteq B' \cup (I \cap O)$. □

In the following theorem we show that one can always find a "best" context for a given non-empty *IBO*-relation.

THEOREM A.3. *Suppose* $|D| \geq 2$. *We have:*

1. *Suppose* $\emptyset \neq R \in [\![\mathfrak{c}]\!] \cap [\![\mathfrak{d}]\!]$. *Then* $\mathfrak{c} \cap \mathfrak{d} \downarrow$ *and* $R \in [\![\mathfrak{c} \cap \mathfrak{d}]\!]$.
2. $\mathfrak{c} \cap \mathfrak{d} \downarrow \Rightarrow [\![\mathfrak{c} \cap \mathfrak{d}]\!] = [\![\mathfrak{c}]\!] \cap [\![\mathfrak{d}]\!]$.
3. *Suppose* $\emptyset \neq R$. *Let* $\mathfrak{X} := \{\mathfrak{x} \mid R \in [\![\mathfrak{x}]\!]\}$. *Suppose* $\mathfrak{X} \neq \emptyset$. *Then* $\mathfrak{X}$ *has a minimum.*

Proof. Let $\mathfrak{c} = \langle I, B, O\rangle$ and $\mathfrak{d} = \langle I', B', O'\rangle$. We prove (1). Suppose $\emptyset \neq R \in [\![\mathfrak{c}]\!] \cap [\![\mathfrak{d}]\!]$. Suppose fRg. We first prove that $\mathfrak{c} \cap \mathfrak{d} \downarrow$, i.e., $B \subseteq B' \cup (I \cap O)$ and $B' \subseteq B \cup (I' \cap O')$. By symmetry, we only need to treat the first desideratum. Suppose, to obtain a contradiction, that for some v: $v \in B$, $v \notin I \cap O$ and $v \notin B'$. By the duality between I and O, we can restrict ourselves to the case that $v \notin I$. Pick d with $d \neq g(v)$. We have: $f_v^d \mathcal{I}_I f R g \mathcal{I}_O g$ and, since $R \subseteq [B]$ and $v \in B$, $f_v^d [B] g$. Hence $f_v^d R g$. It follows that $R \not\subseteq [B']$, a contradiction.

We show that $R \in [\![\mathfrak{c} \cap \mathfrak{d}]\!]$. Suppose that

$$f' \mathcal{I}_{I \cap I'} f R g \mathcal{I}_{O \cap O'} g' \quad \text{and} \quad f'[B \cap B'] g'.$$

We have to show: $f' R g'$. Define:

$\Leftrightarrow f^* := f \restriction I \ \cup \ f' \restriction (\mathsf{Var} \backslash I)$.
$\Leftrightarrow g^* := g \restriction O \ \cup \ g' \restriction (\mathsf{Var} \backslash O)$.

Clearly, $f \mathcal{I}_I f^*$, and, since $f' \mathcal{I}_{I \cap I'} f$, $f^* \mathcal{I}_{I'} f'$. Similarly we find that $g \mathcal{I}_O g^*$ and $g^* \mathcal{I}_{O'} g'$.

We show $f^*[B]g^*$. Consider $v \notin B$. In case $v \in I$, we have $v \in I \backslash B$, and, hence, $v \in O \backslash B$ and, thus, $v \in O$. We find: $f^*(v) = f(v)$, since $v \in I$. $f(v) = g(v)$, since fRg and $R \subseteq [B]$. $g(v) = g^*(v)$, since $v \in O$. So $f^*(v) = g^*(v)$, as desired. In case $v \notin I$, we also have $v \notin O$, since, otherwise $v \in O \backslash B = I \backslash B$. By the definitions of f^* and g^* we find: $f^*(v) = f'(v)$ and $g'(v) = g^*(v)$. Moreover, by the fact that $f'[B \cap B']g'$, we get $f'(v) = g'(v)$. Hence, $f^*(v) = g^*(v)$.

Since $f'[B \cap B']g'$, we have, *a fortiori*, $f'B'g'$. Collecting all previous insights, we may conclude: $f^*\mathcal{I}_I fRg\mathcal{I}_O g^*$ and $f^*[B]g^*$. Hence f^*Rg^*. It follows that $f'\mathcal{I}_{I'} f^*Rg^*\mathcal{I}_{O'} g'$ and $f'[B']g'$. Hence $f'Rg'$.

We turn to (2). Suppose $\mathfrak{c} \cap \mathfrak{d} \downarrow$. By Theorem 3.5 and the fact that $\mathfrak{c} \cap \mathfrak{d} \leq \mathfrak{c}$ and $\mathfrak{c} \cap \mathfrak{d} \leq \mathfrak{d}$, we have: $\llbracket \mathfrak{c} \cap \mathfrak{d} \rrbracket \subseteq \llbracket \mathfrak{c} \rrbracket \cap \llbracket \mathfrak{d} \rrbracket$. For the converse, apply (1). Finally to prove (3), note that, since contexts are finite, $\leq$ is well founded. Hence, $\mathfrak{X}$ has a minimal element. Moreover, (1) implies that $\mathfrak{X}$ is closed under $\cap$. So the minimal element must be the minimum. □

Our last theorem corresponds to the familiar fact of ordinary Predicate Logic that if a set of assignments F is finitely restricted, then one can find a $\subseteq$-minimal I, such that F is $\langle I \rangle$-restricted. Note that $\emptyset$ in the DPL case corresponds to many $\leq$-incomparable contexts. Thus, it is hopelessly ambiguous, in contrast to the predicate logical case.

B. Relations in Context

In DPL meanings are relations. The contexts we studied appear as properties of these relations. We could give an alternative semantics for DPL by building the context into the meaning. Thus we take as meanings pairs $\langle \mathfrak{c}, R \rangle$, where $R \in \llbracket \mathfrak{c} \rrbracket$. Let us call such a pair a $\mathfrak{c}$*-relation*. We define: $\llbracket \phi \rrbracket_{\mathcal{M}} := \langle \mathfrak{c}_\phi, [\phi]_{\mathcal{M}} \rangle$. The new domain of meanings is, on the one hand, essentially richer than the old one, since the same relation falls under several contexts. On the other hand, we threw all non-*IBO*-relations away. We can "lift" the notions intoduced in this paper to $\mathfrak{c}$-relations:

- ⇔ $\langle \mathfrak{c}, R \rangle \bullet \langle \mathfrak{d}, S \rangle = \langle \mathfrak{c} \bullet \mathfrak{d}, R \circ S \rangle$.
- ⇔ A $\mathfrak{c}$-condition is a $\mathfrak{c}$-relations of the form $\langle \langle I, \emptyset, I \rangle, R \rangle$.
- ⇔ If $\mathcal{R} = \langle \langle I, B, O \rangle, R \rangle$, then $\mathsf{dom}(\mathcal{R}) := \langle \langle I, \emptyset, I \rangle, \mathsf{diag}(\mathsf{dom}(\mathsf{R})) \rangle$. Similarly for cod.
- ⇔ $\llbracket B \rrbracket := \langle \langle \emptyset, B, \emptyset \rangle, [B] \rangle$. We write $\llbracket v \rrbracket$ for $\llbracket \{v\} \rrbracket$.
- ⇔ A $\mathfrak{c}$-relation $\mathcal{R}$ has the Switching Property if it is either a $\mathfrak{c}$-condition or there are variables v and w such that:

$$\mathcal{R} = \mathsf{dom}(\mathcal{R}) \bullet \llbracket v \rrbracket \bullet \mathcal{R} \bullet \llbracket w \rrbracket \bullet \mathsf{cod}(\mathcal{R}).$$

- ⇔ A $\mathfrak{c}$-relation is DPL-expressible (over a given domain D) if it can be generated using $\bullet$ from $\mathfrak{c}$-conditions and resettings $\llbracket v \rrbracket$.

In a similar way we can upgrade $\wedge$ and $\exists$. Inspection of the proofs in this paper shows that, in case D is infinite, the DPL-expressible $\mathfrak{c}$-relations are precisely the ones with the Switching Property. Moreover, all $\mathfrak{c}$-relations over the given domain

– infinite or not – are $\mathsf{DPL}(\exists)$-expressible. We consider an example. Remember that:

$$\langle\emptyset,\emptyset,\emptyset\rangle \leq \langle\{x\},\emptyset,\{x\}\rangle \leq \langle\{x\},\{x\},\{x\}\rangle.$$

Let a model with domain D be given. We assume that D has at least two elements. Let $\mathsf{id} := \mathsf{id}_{D^{\mathsf{Var}}}$. We consider three $\mathfrak{c}$-relations with associate relation id.

1. $[\![\top]\!] = \langle\langle\emptyset,\emptyset,\emptyset\rangle,\mathsf{id}\rangle$,
2. $[\![x = x]\!] = \langle\langle\{x\},\emptyset,\{x\}\rangle,\mathsf{id}\rangle$,
3. $[\![\exists u(u = x.\exists x.u = x)]\!] = \langle\langle\{x\},\{x\},\{x\}\rangle,\mathsf{id}\rangle$.

(1) and (2) are $\mathfrak{c}$-conditions and, hence DPL-expressible. In fact, they can be defined in the language by $\top$, respectively $x = x$. In contrast, (3) is not a condition. It is easy to see that (3) does not have the Switching Property, since the domain of its internal relation id consists of all assignments and id is injective. Hence (3) is not DPL-expressible. Note that $\langle\langle\emptyset,\{x\},\emptyset\rangle,\mathsf{id}\rangle$ is not a $\mathfrak{c}$-relation at all.

A further step in modifying our semantics is to make the assignments "local." The idea is that the context "provides" the files/discourse referents/variables on which the variables are defined. Thus, our meaning objects would be of the form $\langle\langle I,B,O\rangle,R\rangle$, where R would be a relation taking input assignments defined on I and yielding output assignments defined on O. This approach leads to a semantics very much like Vermeulen's Referent Systems (see Vermeulen, 1995). One effect of this further modification is that it leads to a somewhat different view of contexts. In the local approach, contexts are the central "engines" that manage the flow of the files in the interactions of meanings. This more dramatic view of contexts is elaborated in Visser and Vermeulen (1996).

References

Barendregt, H.P., 1984, *The Lambda Calculus*, Amsterdam: North-Holland.

Barendregt, H.P., 1991, "Functional programming and lambda calculus," pp. 321–363 in *Formal Models and Semantics, Handbook of Theoretical Computer Science*, Vol. B, J. van Leeuwen, ed., Amsterdam: Elsevier.

Barendregt, H.P., 1992, "Lambda calculi with types," pp. 117–309 in *Handbook of Logic in Computer Science*, Vol. 2, S. Ambramsky, D.M. Gabbay, and T.S.E. Maibaum, eds., Oxford: Clarendon.

Groenendijk, J. and Stokhof, J., 1991, "Dynamic predicate logic," *Linguistics and Philosophy* **14**, 39–100.

Groenendijk, J., Stokhof, M., and Veltman, F., 1996, "Coreference and modality," pp. 179–213 in *Handbook of Contemporary Semantic Theory*, S. Lappin, ed., Oxford: Blackwell.

Heim, I., 1983, "File change semantics and the familiarity theory of definiteness," pp. 164–189 in *Meaning, Use and Interpretation of Language*, R. Bäuerle, C. Schwarze, and A. von Stechow, eds., Berlin: De Gruyter.

Hollenberg, M. and Vermeulen, C.F.M., 1994, "Counting variables in a dynamic setting," Logic Group Preprint Series LGPS-125, Department of Philosophy, Utrecht University. Also in: *Journal of Logic and Computation* **6**, 1996, 725–744.

Jacobs, B., 1991, "Categorical type theory," Ph.D. Thesis, Nijmegen University.

Kamp, H., 1981, "A theory of truth and semantic representation," pp. 1–41 in *Truth, Interpretation and Information*, J. Groenendijk, T. Jansen, and M. Stokhof, eds., Dordrecht: Foris.

Klop, J.W., 1992, "Term rewriting systems," pp. 1–116 in *Handbook of Logic in Computer Science*, Vol. 2, S. Ambramsky, D.M. Gabbay, and T.S.E. Maibaum, eds., Oxford: Clarendon.

Mitchell, J., 1991, "Type systems for programming languages," pp. 365–458 in *Formal Models and Semantics, Handbook of Theoretical Computer Science*, Vol. B, J. van Leeuwen, ed., Amsterdam: Elsevier.

Muskens, R., 1991, "Anaphora and the logic of change," pp. 414–430 in *JELIA '90, European Workshop on Logics in AI*, J. van Eijck, ed., Springer Lecture Notes, Vol. 478, Berlin: Springer-Verlag.

Nederpelt, R., Geuvers, J.H., and de Vrijer, R.C., eds., 1994, *Selected Papers on Automath*, Amsterdam: North Holland.

Troelstra, A.S. and van Dalen, D., 1988a, *Constructivism in Mathematics*, Vol. 1, Studies in Logic and the Foundations of Mathematics, Vol. 121, Amsterdam: North Holland.

Troelstra, A.S. and van Dalen, D., 1988b, *Constructivism in Mathematics*, Vol. 2, Studies in Logic and the Foundations of Mathematics, Vol. 123, Amsterdam: North Holland.

Veltman, F., 1996, "Defaults in update semantics," *Journal of Philosophical Logic* **25**, 221–261.

Vermeulen, C.F.M., 1993, "Sequence semantics for dynamic predicate logic," *Journal of Logic, Language and Information* **2**, 217–254.

Vermeulen, C.F.M., 1995, "Merging without mystery, variables in dynamic semantics," *Journal of Philosophical Logic* **24**, 405–450.

Visser, A., 1994, "Actions under presuppositions," pp. 196–233, in *Logic and Information Flow*, J. van Eijck and A. Visser, eds., Cambridge, MA: MIT Press.

Visser, A., 1995, "Dynamic discourse denotations," Lecture Notes for the Department of Philosophy, University of Utrecht, and the Department of Mathematics and Computer Science, University of Amsterdam.

Visser, A. and Vermeulen, C., 1996, "Dynamic bracketing and discourse representation," *Notre Dame Journal of Formal Logic* **37**, 321–365.