Dynamic Predicate Logic[1]

Jeroen Groenendijk
Martin Stokhof
ITLI/Department of Philosophy
Department of Computational Linguistics
University of Amsterdam

July, 1989
revised April, 1990

[1]to appear in *Linguistics and Philosophy*

1 Introduction

This paper is devoted to the formulation and investigation of a dynamic semantic interpretation of the language of first-order predicate logic. The resulting system, which will be referred to as 'dynamic predicate logic', is intended as a first step towards a compositional, non-representational theory of discourse semantics.

In the last decade, various theories of discourse semantics have emerged within the paradigm of model-theoretic semantics. A common feature of these theories is a tendency to do away with the principle of compositionality, a principle which, implicitly or explicitly, has dominated semantics since the days of Frege. Therefore the question naturally arises whether non-compositionality is in any way a necessary feature of discourse semantics.

Since we subscribe to the interpretation of compositionality as constituting primarily a methodological principle, we consider this to be a methodological rather than an empirical question. As a consequence, the emphasis in the present paper lies on developing an alternative compositional semantics of discourse, which is empirically equivalent to its non-compositional brethren, but which differs from them in a principled methodological way. Hence, no attempts are made to improve on existing theories empirically.

Nevertheless, as we indicate in section 5, the development of a compositional alternative may in the end have empirical consequences, too. First of all, it can be argued that the dynamic view on interpretation developed in this paper suggests natural and relatively easy to formulate extensions which enable one to deal with a wider range of phenomena than can be dealt with in existing theories.

Moreover, the various approaches to the model-theoretic semantics of discourse that have been developed during the last decade, have constituted a 'fresh start' in the sense that much of what had been accomplished before was ignored, at least for a start. Of course, this is a justified strategy if one feels one is trying to develop a radically different approach to recalcitrant problems. However, there comes a time when such new approaches have to be compared with the older one, and when an assessment of the pros and cons of each has to be made.

One of the main problems in semantics today, we feel, is that a semantic theory such as Montague grammar, and an approach like Kamp's discourse representation theory, are hard to compare, let alone that it is possible to unify their insights and results. One of the main obstacles is that the latter lacks, or has abolished, the principle of compositionality, which is so central a feature of the former. Hence, the development of a compositional alternative to the semantics of discourse may very well have empirical import on this score as well: in the end, it may contribute to a unification of these two approaches which have largely complementary descriptive domains.

In the extension of modeltheoretic semantics from the sentential to the dis-

course level, various theories have emerged: beside discourse representation theory (Kamp [1981,1983]), we should mention Heim's file card semantics (Heim [1982,1983]), and, in a different framework, the work of Seuren (Seuren [1986]). None of these theories makes compositionality its starting point. (However, it seems that Heim [1982, Ch.3], does attach some value to compositionality.) Since the aim of this paper is restricted to showing that a compositional alternative can be developed, and since Kamp's discourse representation theory is both self-consciously non-compositional and formally most explicit, we feel justified in restricting comparison to just the latter theory.

The paper is organized as follows. In section 2, we introduce the elements of dynamic interpretation in a heuristic fashion, discussing a small number of well-known problematic cases. In section 3, we recapitulate our findings and formulate the dynamic semantics of predicate logic systematically, and study its logical properties. The resulting system is compared with ordinary predicate logic, discourse representation theory, and quantificational dynamic logic in section 4. In section 5, we indicate prospects for further developments and, in retrospect, we present our philosophical and methodological motives.

To end this introductory section, we remark that Barwise' proposal for the interpretation of anaphoric relations within the framework of situation semantics (Barwise [1987]), which in Rooth [1987] is compared with Heim's file card semantics and with Montague Grammar, are akin in spirit and content to our approach. So is Schubert & Pelletier [1989]. Equally akin in spirit, but less in content, is Zeevat [1989].

2 Elements of dynamic interpretation

2.1 Cross-sentential and donkey-anaphora

We begin this section with a brief discussion of two well-known problems: cross-sentential anaphora and anaphoric relations in donkey-sentences. We state them anew, because we want to make clear what, we feel, is the real challenge they offer.

If we use standard first-order predicate logic (henceforth, *PL*) in translating a natural language sentence or discourse, anaphoric pronouns will turn up as bound variables. In many cases, this means that in order to arrive at formulas which are good translations, i.e., which express the right meaning, we have to be pretty inventive, and should not pay too much attention to the way in which the natural language sentence or discourse is built up. Let us illustrate this with three simple examples, which nevertheless are representative for the kind of problems we meet:

(1) A man walks in the park. He whistles

(2) If a farmer owns a donkey, he beats it

(3) Every farmer who owns a donkey, beats it

In order for the pronoun *he* in the second sentence of (1) to be anaphorically linked to *a man* in the first sentence, we have to give an existential quantifier wide scope over the conjunction of the two sentences involved. Doing so, we arrive at (1a):

(1a) $\exists x[\texttt{man}(x) \wedge \texttt{walk_in_the_park}(x) \wedge \texttt{whistle}(x)]$

Now, notice that the translation of the first sentence in (1), which would be $\exists x[\texttt{man}(x) \wedge \texttt{walk_in_the_park}(x)]$, does not occur as a subformula in (1a). Apparently, we do not get from (1) to (1a) in a step-by-step, i.e., in a compositional way. If we did, we would rather translate (1) as (1b):

(1b) $\exists x[\texttt{man}(x) \wedge \texttt{walk_in_the_park}(x)] \wedge \texttt{whistle}(x)$

But this is not a proper translation of (1), at least not in standard predicate logic, since in (1b) the last occurrence of the variable x is not bound by the existential quantifier, and hence the anaphoric link in (1) is not accounted for. However, suppose we could interpret (1b) in such a way that it *is* equivalent with (1). Evidently, (1b) would be preferred to (1a) as a translation of (1), since it could be the result of a compositional procedure.

Turning to examples (2) and (3), we observe that a proper translation in *PL* for both of them is (2a):

(2a) $\forall x \forall y[[\texttt{farmer}(x) \wedge \texttt{donkey}(y) \wedge \texttt{own}(x, y)] \rightarrow \texttt{beat}(x, y)]$

These cases are more dramatic than the previous one. Although (2) and (3) contain indefinite terms, which normally translate as existentially quantified phrases, we need universal quantification to account for their meaning in these kinds of examples. And notice, moreover, that the corresponding universal quantifiers $\forall x$ and $\forall y$ have to be given wide scope over the entire formula, whereas the indefinite terms in (2) and (3) to which they correspond, appear inside the antecedent of an implication in the case of (2), and way inside the relative clause attached to the subject term *every farmer* in the case of (3). If we use *PL* as our means to represent meaning, these kinds of examples prevent us from uniformly translating indefinite terms as existentially quantified phrases. Again, this constitutes a breach of the principle of compositionality, a principle which is not only intuitively appealing, but also theoretically parsimonious and computationally plausible.

From a compositional point of view, translations like (2b) for sentence (2), and (3b) for sentence (3), are to be preferred:

(2b) $\exists x[\texttt{farmer}(x) \wedge \exists y[\texttt{donkey}(y) \wedge \texttt{own}(x, y)]] \rightarrow \texttt{beat}(x, y)$

(3b) $\forall x[[\texttt{farmer}(x) \wedge \exists y[\texttt{donkey}(y) \wedge \texttt{own}(x, y)]] \rightarrow \texttt{beat}(x, y)]$

But then again, (2b) and (3b) do not have the proper meaning in *PL*. For one thing the occurrences of the variable y in case of (3b), and of the variables x and y in case of (2b), in the respective consequents, are not bound by the existential quantifiers in the antecedents. Hence, (2b) and (3b) are not equivalent with (2a), at least not in *PL*.

Examples like (1)–(3) have been treated successfully in discourse representation theory (henceforth *DRT*), but at a cost: the problem of providing a compositional translation is not really solved, and *DRT* uses a rather non-orthodox logical language. In *DRT*, (1) would be represented as (1c), (2) and (3) as (2c):

(1c) $[x][\texttt{man}(x), \texttt{walk_in_the_park}(x), \texttt{whistle}(x)]$

(2c) $[\ ][[x, y][\texttt{farmer}(x), \texttt{donkey}(y), \texttt{own}(x, y)] \rightarrow [\ ][\texttt{beat}(x, y)]]$

We will not go into the semantics of these discourse representation structures here (cf. section 4.2), for the moment it suffices to note that (1c) and (2c) have essentially the same truth conditions as (1a) and (2a) respectively. The important thing, however, is that these representations differ in structure from the corresponding sentences in much the same way as the *PL*-translations. In fact, the structure of (1c) is essentially that of (1a), and not that of (1b). And in (2c) no representation of the relative clause *who owns a donkey* or of the intransitive verbphrase *own a donkey*—which form a constituent in (2) and (3) respectively—can be isolated as a substructure of (2c). So, from a compositional point of view, they are hardly a change for the better. For the moment we leave it at this observation, but we return to the issue in some detail in section 4.2.

In this paper we give an alternative account of the phenomena exemplified by (1)–(3); we do so by replacing the standard semantics of the language of first-order predicate logic by a dynamic semantics, which is inspired by systems of dynamic logic as they are used in the denotational semantics of programming languages. (See Harel [1984] for an overview.) The resulting system of dynamic predicate logic (henceforth, *DPL*) constitutes an improvement over *DRT* in the following sense: to the extent that this is possible in a first-order language at all, it gives a compositional semantic treatment of the relevant phenomena, while the syntax of the language used, being that of standard predicate logic, is an orthodox one. More specifically, using *DPL* it becomes possible to represent the meanings of the sentences (1), (2) and (3) by means of the formulas (1b), (2b) and (3b). As we remarked above, such representations are to be preferred from a compositional and a computational point of view. The dynamic semantics of *DPL* makes sure that (1b) comes out with the same truth conditions as (1a) is assigned in *PL*, and that (2b) and (3b) come out with the same truth conditions as (1a) and (2a) have in *PL*.

2.2 The dynamic view on meaning

The general starting point of the kind of semantics that *DPL* is an instance of, is that the meaning of a sentence does not lie in its truth conditions, but rather

in the way it changes (the representation of) the information of the interpreter. The utterance of a sentence brings us from a certain state of information to another one. The meaning of a sentence lies in the way it brings about such a transition. Although this 'procedural' dynamic view on meaning as such is not particular to semantic theories of discourse (it can also be found in theories about sentence meaning, e.g., in the work of Stalnaker), it is a view which is endorsed by all approaches to discourse semantics which we referred to above.

It should be noted, though, that in most cases one really studies only one particular aspect of the information change potential that makes up the meaning of a sentence, at a time. For example, in the standard version of *DRT*, information change is narrowed down to the (im)possibilities of subsequent anaphoric reference that sentences determine. All other information that a sentence conveys, is treated in a static, rather than in a dynamic fashion. *DPL* is like *DRT* in this respect. It, too, restricts the dynamics of interpretation to that aspect of the meaning of sentences that concerns their potential to 'pass on' possible antecedents for subsequent anaphors, within and across sentence boundaries. (See Groenendijk & Stokhof [1988] for some more discussion of this point.)

As has been observed by several authors, there is a strong correspondence between the dynamic view on meaning, and a basic idea underlying the denotational approach to the semantics of programming languages, viz., that the meaning of a program can be captured in terms of a relation between machine states. Given the restriction to antecedent-anaphor relations, the observed correspondence comes down to the following. A machine state may be identified with an assignment of objects to variables. The interpretation of a program can then be regarded as a set of ordered pairs of assignments, as the set of all its possible 'input-output' pairs. A pair $\langle g, h \rangle$ is in the interpretation of a program π, if when π is executed in state g, a possible resulting state is h.

For example, the execution of an atomic program consisting of a simple assignment statement '$x := a$' transforms a state (assignment) g into a state (assignment) h which differs from g at most with respect to the value it assigns to x, and in which the object denoted by the constant a is assigned to x.

Another simple illustration is provided by sequences of programs. The interpretation of a sequence of programs '$\pi_1 ; \pi_2$' is as follows. It can take us from state g to h, if there is some state k such that the program π_1 can take us from g to k, and π_2 from k to h. Or to put it differently, the second program is executed in a state which is (partly) created by the first.

As we intend to show in this paper, the basic idea that (certain aspects) of meaning can be described in terms of relations between states, can be applied fruitfully in natural language semantics as well. It should be remarked, though, that the aims and perspectives of systems of dynamic logic as they are used in the semantics of programming languages, are rather different from the purpose for which we want to use the system to be developed below. And consequently, there are differences between these systems as such. Some discussion of these matters can be found in section 4.3.

2.3 Dynamic conjunction and existential quantification

In the present and the next two sections, we introduce a dynamic interpretation for the language of extensional first-order predicate logic in a step-by-step fashion, deferring an explicit statement and a formal investigation of *DPL* to section 3. In the present section we introduce dynamic conjunction and existential quantification, which will enable us to deal with the first of the three examples discussed above, which concerned cross-sentential anaphora. In section 2.4, we discuss implication and existential quantification. Their dynamic treatment will give us the means to treat simple donkey-sentences, such as exemplified by the second example. And finally in section 2.5, we turn to universal quantification and negation in order to be able to deal with the more complicated donkey-sentences as exemplified by the last example.

The vocabulary of *DPL* consists of n-place predicates, individual constants and variables. They are interpreted in the usual fashion. The models that we use, are ordinary extensional first-order models, consisting of a domain D of individuals and an interpretation function F, assigning individuals to the individual constants, and sets of n-tuples of individuals to the n-place predicates. Further, we use assignments as usual, i.e., as total functions from the set of variables to the domain. They are denoted by 'g', 'h', and so on. By '$h[x]g$' we mean that assignment h differs from g at most with respect to the value it assigns to x. When in what follows we speak of the interpretation of an expression, we mean its semantic value in a suitable model. The function assigning semantic values is denoted by '$[\![\ ]\!]$'.

In the standard semantics of predicate logic, the interpretation of a formula is a set of assignments, viz., those assignments which verify the formula. In the dynamic semantics of *DPL* the semantic object expressed by a formula is a set of ordered pairs of assignments. Trading on the analogy with programming languages, such pairs can be regarded as possible 'input-output' pairs: a pair $\langle g, h\rangle$ is in the interpretation of a formula ϕ iff when ϕ is evaluated with respect to g, h is a possible outcome of the evaluation procedure. Since g and h are assignments of objects to variables, the difference between an input assignment g and an output assignment h can only be that a different object is assigned to one or more variables. This is precisely what happens when an existentially quantified formula is interpreted dynamically. Consider the formula $\exists xPx$. In the standard semantics, an assignment g is in the interpretation of $\exists xPx$ iff there is some assignment h which differs from g at most with respect to the value it assigns to x, and which is in the interpretation of Px, i.e., which assigns an object $h(x)$ to x such that $h(x) \in F(P)$. When $\exists xPx$ is treated dynamically, all assignments h such that $h[x]g$ & $h(x) \in F(P)$, are taken to be possible outputs with respect to input g. In other words:

$$[\![\exists xPx]\!] = \{\langle g, h\rangle \mid h[x]g \ \&\ h(x) \in F(P)\}$$

This will not yet do for the general case of $\exists x\phi$. We have to reckon with the

possibility that the interpretation of ϕ, too, has dynamic effects. (For example, ϕ itself might be an existentially quantified formula.) Taking this into account, the dynamic interpretation of $\exists x\phi$ will consist of those pairs of assignments $\langle g, h\rangle$ such that there is some assignment k which differs from g at most in x and which together with h forms a possible input-output pair for ϕ. The interpretation clause for existentially quantified formulas then reads as follows:

$$[\![\exists x\phi]\!] = \{\langle g, h\rangle \mid \exists k\colon k[x]g \ \&\ \langle k, h\rangle \in [\![\phi]\!]\}$$

In order to show that this interpretation of $\exists x\phi$ squares with the one given above for $\exists xPx$, we first have to state the interpretation of atomic formulas.

Unlike existentially quantified formulas, atomic formulas do not have dynamic effects of their own. Rather, they function as a kind of 'test' on incoming assignments. An atomic formula tests whether an input assignment satisfies the condition it embodies. If so, the assignment is passed on as output, if not it is rejected. So, the dynamics of an atomic formula consists in letting pass the assignments which satisfy it, and blocking those that don't. This is captured in the following definition:

$$[\![Rt_1 \ldots t_n]\!] = \{\langle g, h\rangle \mid h = g \ \&\ \langle [\![t_1]\!]_h, \ldots, [\![t_n]\!]_h\rangle \in F(R)\}$$

Here, as usual, $[\![t]\!]_h = F(t)$ if t is an individual constant, and $[\![t]\!]_h = h(t)$ if t is a variable.

We first work out our simple example of an existentially quantified formula $\exists xPx$:

$$\begin{gathered}[\![\exists xPx]\!] = \{\langle g, h\rangle \mid \exists k\colon k[x]g \ \&\ \langle k, h\rangle \in [\![Px]\!]\} = \\ \{\langle g, h\rangle \mid \exists k\colon k[x]g \ \&\ k = h \ \&\ h(x) \in F(P)\}\} = \\ \{\langle g, h\rangle \mid h[x]g \ \&\ h(x) \in F(P)\}\end{gathered}$$

This example illustrates the interpretation of the existential quantifier and that of atomic formulas. The meaning of $\exists xPx$ determines that for a given input assignment g, we get as possible outputs those assignments h which differ from g at most in x and which satisfy the condition that the individual $h(x)$ has the property $F(P)$.

The dynamic interpretation of existential quantification presented here is only one ingredient of a treatment of cross-sentential anaphoric binding as it was illustrated by the first example discussed in setion 2.1. The example consists of a sequence of two sentences, the first of which contains an indefinite term which functions as the antecedent of an anaphoric pronoun occurring in the second sentence. Although this obviously is not all there is to it, within the framework at hand, simple sentence sequencing is best represented as *conjunction*. Going about compositionally, what we get in this case is a conjunction consisting of an existentially quantified formula and a formula containing a free occurrence of

the variable corresponding to the quantifier. The simplest example of a formula that is of this form is $\exists x Px \wedge Qx$.

To get the required interpretation for this kind of formula, a dynamic interpretation of existential quantification alone does not suffice, we need a dynamic treatment of conjunction as well. For example, in order to get the required anaphoric reading we have to interpret $\exists x Px \wedge Qx$ in such a way that the second occurrence of x, which is outside the scope of the quantifier, is bound by that quantifier with the same force as the first occurrence of x, which is inside its scope.

So the first thing we require of the dynamic interpretation of conjunction is that it passes on values of variables from the first conjunct to the second. Moreover, we note that values assigned to variables in a conjunction should remain available for further conjuncts that are added. If we continue the discourse 'A man walks in the park. He meets a woman.' with 'He kisses her.', we must view this as adding another conjunct. And this newly added conjunct may contain 'free' occurrences of variables (pronouns) which nevertheless are bound by existential quantifiers (indefinite terms) which have occurred earlier on.

In fact, this is exactly what the interpretation of a sequence of programs as described above amounts to. Hence, our definition of dynamic conjunction is the following:

$$[\![\phi \wedge \psi]\!] = \{\langle g, h\rangle \mid \exists k\colon \langle g, k\rangle \in [\![\phi]\!] \ \&\ \langle k, h\rangle \in [\![\psi]\!]\}$$

According to this definition, the interpretation of $\phi \wedge \psi$ with input g may result in output h iff there is some k such that interpreting ϕ in g may lead to k, and interpreting ψ in k enables us to reach h.

We are now fully equipped to deal with the first of the three examples dicussed above, which concerned cross-sentential anaphora. Calculating the interpretation of $\exists x Px \wedge Qx$ shows that indeed the binding effects of an existential quantifier may reach further than its scope, more in particular they reach over further conjuncts:

$$\begin{gathered}[\![\exists x Px \wedge Qx]\!] = \{\langle g, h\rangle \mid \exists k\colon \langle g, k\rangle \in [\![\exists x Px]\!] \ \&\ \langle k, h\rangle \in [\![Qx]\!]\} = \\ \{\langle g, h\rangle \mid \exists k\colon k[x]g \ \&\ k(x) \in F(P) \ \&\ h = k \ \&\ h(x) \in F(Q)\} = \\ \{\langle g, h\rangle \mid h[x]g \ \&\ h(x) \in F(P) \ \&\ h(x) \in F(Q)\}\end{gathered}$$

Here, we see that the occurrence of x in the second conjunct Qx, although it is not in the scope of the quantifier $\exists x$ in the ordinary sense, is nevertheless bound by it with the same force as the occurrence of x in Px in the first conjunct, which obviously is in the scope of $\exists x$. This means that in *DPL* there is no difference in meaning between the formula $\exists x Px \wedge Qx$ and the formula $\exists x[Px \wedge Qx]$. From the latter fact it is also clear that if we continue a conjunction $\phi \wedge \psi$ with a further conjunct χ, the binding force of quantifiers in either one of the conjuncts ϕ and ψ will remain active.

Because of its power to pass on variable bindings from its left conjunct to the right one, we call conjunction an *internally dynamic* connective. And because of its capacity to keep passing on bindings to conjuncts yet to come, we call it an *externally dynamic* connective as well. For similar reasons, the existential quantifier is called both internally and externally dynamic: it can bind variables to the right, both inside and outside its scope.

It is precisely this feature of *DPL*, that it allows for existential quantifiers to bind variables yet to come which are outside their scope, that lends it the power to solve the problem of getting a compositional treatment of antecedent-anaphor relations which go across sentence boundaries. It allows us to translate the sentence containing the antecedent indefinite term, without having to look ahead at what is still to come, treating it as an ordinary existentially quantified phrase. Then we can translate a sentence which follows and which contains an anaphor, without having to re-analyze the translation so-far, regarding the anaphoric pronoun as an ordinary variable. The dynamic semantics takes care of the rest. It makes sure that the pronoun is treated as a variable bound by the quantifier which corresponds to the indefinite term.

2.4 Dynamic existential quantification and implication

The second kind of example which we introduced in section 2.1, concerns simple donkey-sentences. The main problem of donkey-sentences is the occurrence of an indefinite term in the antecedent of an implication which is anaphorically linked to a pronoun in the consequent. As we indicated above, if we are to represent the meaning of such sentences in ordinary predicate logic, which allows quantifiers to bind only those variables which occur in their syntactic scope, then we are forced to regard the indefinite term as a universal quantifier and to give it wide scope over the implication as a whole. This goes against compositionality in two ways: first of all, we cannot use the ordinary, lexically determined meaning of indefinite terms, and secondly, we must deviate from the syntactic structure by 'raising' these terms from their position in the antecedent to a position outside the implication. In order to show that this kind of example can be treated in *DPL* in a more compositional, and hence more satisfactory way, we have to say what the dynamic interpretation of implication is.

The simplest example of a formula corresponding to a donkey-sentence is $\exists x Px \rightarrow Qx$. In order for this formula to get the required interpretation, the dynamic interpretation of implication has to allow for an existential quantifier in its antecedent to bind a variable in its consequent. This means that implication is like conjunction in the following respect: it passes on values assigned to variables in its antecedent to its consequent. In other words, implication is an internally dynamic connective.

But that is not all. We also observe that the existential quantifier in the antecedent has universal force. This can be accounted for as follows. With respect to an input assignment, the antecedent of an implication results in a set of

possible output assignments. For the implication as a whole, it seems reasonable to require that *every* assignment that is a possible output of the antecedent, be a possible input for the consequent. By this we mean that an output assignment h of the antecedent, when taken as input to the consequent, should result in at least one output assignment k. In other words, the interpretation of an implication $\phi \rightarrow \psi$ should be such that for every pair $\langle g, h\rangle$ in the interpretation of ϕ there is some assignment k such that $\langle h, k\rangle$ is in the interpretation of ψ. This feature of the interpretation of implication results in universal force of an existential quantifier occurring in the antecedent. Consider $\exists xPx \rightarrow Qx$. With respect to an input assignment g, the antecedent $\exists xPx$ results in the set of assignments h such that $h[x]g$ and $h(x) \in F(P)$. If we require, as we do, that every such h should be a proper input of the consequent Qx, the result is that every h such that $h(x) \in F(P)$, also satisfies $h(x) \in F(Q)$.

This does not yet determine which pairs of assignments constitute the interpretation of $\phi \rightarrow \psi$, it only tells us with respect to which assignments $\phi \rightarrow \psi$ can be 'successfully executed'. To get at the full interpretation of $\phi \rightarrow \psi$ we need yet another observation, which is that normally an implication as a whole does not pass on values assigned to variables by quantifiers in the implication itself, to sentences yet to come. Consider the following example:

(4) *If a farmer owns a donkey, he beats it. He hates it

In this example, the pronouns *he* and *it* in the second sentence cannot be anaphorically linked to the indefinite terms in the preceding implication. And quite generally it is concluded on the basis of examples such as these that a quantifier which occurs inside an implication, be it in the antecedent or in the consequent, cannot bind variables outside the implication. (A lot more needs to be said about this, and for some of it we refer to section 5.1.) In this respect implication is unlike conjunction: it is not externally dynamic; like an atomic formula, an implication as a whole has the character of a test.

What we thus end up with as the dynamic interpretation of implication, is the following:

$$[\![\phi \rightarrow \psi]\!] = \{\langle g, h\rangle \mid h = g \ \&\ \forall k\colon \langle h, k\rangle \in [\![\phi]\!] \Rightarrow \exists j\colon \langle k, j\rangle \in [\![\psi]\!]\}$$

The interpretation of $\phi \rightarrow \psi$ accepts an assignment g iff every possible output of ϕ with respect to g leads to a successful interpretation of ψ, and it rejects g otherwise. Armed with this definition, we can now proceed to show that *DPL* assigns the required interpretation to formulas which correspond to the kind of donkey-sentences exemplified by our second example. By way of illustration, we work out the interpretation of the formula $\exists xPx \rightarrow Qx$:

$$\begin{gathered}
[\![\exists xPx \rightarrow Qx]\!] = \\
\{\langle g, h\rangle \mid h = g \ \&\ \forall k\colon \langle h, k\rangle \in [\![\exists xPx]\!] \Rightarrow \exists j\colon \langle k, j\rangle \in [\![Qx]\!]\} = \\
\{\langle g, g\rangle \mid \forall k\colon \langle g, k\rangle \in [\![\exists xPx]\!] \Rightarrow \exists j\colon \langle k, j\rangle \in [\![Qx]\!]\} = \\
\{\langle g, g\rangle \mid \forall k\colon k[x]g \ \&\ k(x) \in F(P) \Rightarrow k(x) \in F(Q)\}
\end{gathered}$$

This example shows that the binding effects of an existential quantifier occurring in the antecedent of an implication extend to occurrences of the corresponding variable in the consequent, and that such a quantifier occurrence has universal force. It also shows that dynamic effects are restricted to the implication as such, and are not passed on to any formulas which might follow it. In effect, as we shall see below, $\exists x Px \rightarrow Qx$ is equivalent in *DPL* to $\forall x[Px \rightarrow Qx]$.

2.5 Universal quantification, negation and disjunction

For a treatment of the second, more complicated kind of donkey-sentences, exemplified by our third example, we need to state the interpretation of the universal quantifier. One aspect of this interpretation is illustrated by the following two examples:

(5) *Every man walks in the park. He whistles

(6) *Every farmer who owns a donkey beats it. He hates it

The pronoun *he* occurring in the second sentence of (5), cannot be interpreted as being anaphorically linked to the universal term in the sentence preceding it. Nor can the pronouns *he* and *it* in the second sentence of (6) be anaphorically linked to the terms *every farmer* and *a donkey* in the first sentence of (6). Generally, from examples such as these it is concluded that the universal quantifier shares with implication the characteristic of being externally static. Neither a universal quantifier itself, nor any existential quantifier inside its scope can bind variables outside the scope of that universal quantifier. (Again, we refer to section 5.1 for some discussion of this point.) But, of course, inside the scope of a universal quantifier dynamic effects may very well occur, as the donkey-sentence (3) shows. This leads to the following definition of the interpretation of universal quantification:

$$[\![\forall x\phi]\!] = \{\langle g, h\rangle \mid h = g \ \&\ \forall k\colon k[x]h \Rightarrow \exists m\colon \langle k, m\rangle \in [\![\phi]\!]\}$$

So, a universally quantified formula $\forall x\phi$, too, functions as a test. An input assignment g is passed on iff every assignment that differs at most from g in x is a proper input for ϕ, otherwise it is blocked. An output assignment is always identical to the corresponding input.

That the dynamic interpretation of the universal quantifier, together with that of the existential quantifier and implication, allows us to deal with the donkey-sentence (3) in the manner discussed at the beginning of this section, is shown by working out the interpretation of a formula that exhibits the relevant structure:

$$[\![\forall x[[Px \wedge \exists y[Qy \wedge Rxy]] \rightarrow Sxy]]\!] =$$
$$\{\langle g, h\rangle \mid h = g \ \&\ \forall k\colon k[x]h \Rightarrow \exists m\colon \langle k, m\rangle \in [\![[Px \wedge \exists y[Qy \wedge Rxy]] \rightarrow Sxy]\!]\} =$$

$$\{\langle g,g\rangle \mid \forall k\colon k[x]g \Rightarrow (\forall j\colon \langle k,j\rangle \in$$
$$[\![Px \wedge \exists y[Qy \wedge Rxy]]\!] \Rightarrow \exists z\colon \langle j,z\rangle \in [\![Sxy]\!])\} =$$
$$\{\langle g,g\rangle \mid \forall k\colon k[x]g \ \&\ k(x) \in F(P) \Rightarrow$$
$$(\forall j\colon j[y]k \ \&\ j(y) \in F(Q) \ \&\ \langle j(x),j(y)\rangle \in F(R) \Rightarrow \langle j(x),j(y)\rangle \in F(S))\} =$$
$$\{\langle g,g\rangle \mid \forall h\colon h[x,y]g \ \&\ h(x) \in F(P) \ \&$$
$$h(y) \in F(Q) \ \&\ \langle h(x),h(y)\rangle \in F(R) \Rightarrow \langle h(x),h(y)\rangle \in F(S)\}$$

This example illustrates that the dynamic semantics of *DPL* enables us to treat the more complicated type of donkey-sentences, too, in a straightforward, intuitive and compositional manner. *DPL* allows us to translate an indefinite term uniformly as an existentially quantified phrase *in situ*, i.e., when and where we encounter it in a structure, without any need of re-analysis. We can treat a pronoun which is anaphorically linked to such a term simply as a variable corresponding to the quantifier. The dynamic interpretation of the existential quantifier and of the implication, ensures that the proper bindings result, and that the indefinite term has the required universal force.

Within the limits set by a first-order language, the account we have given above of cross-sentential anaphora and donkey-sentences, is as compositional as can be. Using *DPL* as our semantic representation language, we can proceed to obtain representations of the meanings of simple natural language discourses in an on-line, more or less left-to-right manner, guided by the ordinary syntactic structures and the usual lexical meanings of the phrases we encounter.

We conclude this section by stating the interpretation of negation and disjunction. Negation is like implication and universal quantification in that it, too, normally blocks anaphoric links between a term that occurs in its scope, and a pronoun outside of it, i.e., negation is static. (More on this in section 5.1.) The following two examples illustrate this:

(7) It is not the case that a man walks in the park. *He whistles.

(8) No man walks in the park. *He whistles.

Hence, the interpretation of a negation $\neg\phi$ will be of the type of a test: it returns an input assignment g iff ϕ can not be succesfully processed. If ϕ can be succesfully processed with respect to g as input, g is blocked by $\neg\phi$:

$$[\![\neg\phi]\!] = \{\langle g,h\rangle \mid h = g \ \&\ \neg\exists k\colon \langle h,k\rangle \in [\![\phi]\!]\}$$

The following example, which has the structure of such sequences of sentences as (7) and (8), illustrates how negation works:

$$[\![\neg\exists xPx \wedge Qx]\!] =$$
$$\{\langle g,h\rangle \mid \exists k\colon \langle g,k\rangle \in [\![\neg\exists xPx]\!] \ \&\ \langle k,h\rangle \in [\![Qx]\!]\} =$$
$$\{\langle g,h\rangle \mid \exists k\colon \langle g,k\rangle \in [\![\neg\exists xPx]\!] \ \&\ h = k \ \&\ h(x) \in F(Q)\} =$$

$$\{\langle g,h\rangle \mid \langle g,h\rangle \in [\![\neg\exists xPx]\!] \ \&\ h(x) \in F(Q)\} =$$
$$\{\langle g,h\rangle \mid h = g \ \&\ \neg\exists k\colon \langle h,k\rangle \in \{\langle g,h\rangle \mid h[x]g \ \&\ h(x) \in F(P)\} \ \&\ h(x) \in F(Q)\} =$$
$$\{\langle g,h\rangle \mid h = g \ \&\ \neg\exists k\colon k[x]h \ \&\ k(x) \in F(P) \ \&\ h(x) \in F(Q)\} =$$
$$\{\langle g,g\rangle \mid \neg\exists k\colon k[x]g \ \&\ k(x) \in F(P) \ \&\ g(x) \in F(Q)\}$$

As we can see, the first conjunct, being a negation, does not change the assignment with respect to which the second conjunct is interpreted. The test-like character of a negation leaves the occurrence of x in the second conjunct unbound by the existential quantifier which occurs within its scope in the first conjunct. This means that, whereas $\exists xPx \wedge Qx$ and $Qx \wedge \exists xPx$ differ in meaning, $\neg\exists xPx \wedge Qx$ is equivalent to $Qx \wedge \neg\exists xPx$.

As for disjunction, it shares the feature of being externally static with implication, negation and the universal quantifier. It, too, tests an input assignment g, and the condition it embodies is that at least one of its disjuncts be interpretable successfully with g as input. Only if this condition is met, g is returned as output:

$$[\![\phi \vee \psi]\!] = \{\langle g,h\rangle \mid h = g \ \&\ \exists k : \langle h,k\rangle \in [\![\phi]\!] \vee \langle h,k\rangle \in [\![\psi]\!]\}$$

According to this interpretation of disjunction, no antecedent-anaphor relations are possible between the disjuncts, i.e., disjunction is not only externally, but also internally static. We will come back to this in sections 4.3 and 5.1.

This concludes our introduction of the ingredients of *DPL*. In the next section, we will present *DPL* more systematically, and investigate some of the logical facts touched upon above in somewhat more detail.

3 *DPL*, a system of dynamic predicate logic

This section is devoted to a formal study of the *DPL*-system. In 3.1, we present its syntax and semantics systematically. Section 3.2 contains definitions of some basic semantic notions, such as truth and equivalence. In section 3.3, we turn to the subject of scope and binding, and in section 3.4, we state some logical facts. Section 3.5 is concerned with the notion of entailment.

3.1 Syntax and semantics

The non-logical vocabulary of *DPL* consists of: n-place predicates, individual constants, and variables. Logical constants are negation $\neg$, conjunction $\wedge$, disjunction $\vee$, implication $\rightarrow$, the existential and universal quantifiers $\exists$ and $\forall$, and identity $=$.

Definition 1 (Syntax)

1. If $t_1, \ldots, t_n$are individual constants or variables, R is an n-place predicate, then $Rt_1 \ldots t_n$ is a formula

2. If t_1 and t_2 are individual constants or variables, then $t_1 = t_2$ is a formula
3. If ϕ is a formula, then $\neg\phi$ is a formula
4. If ϕ and ψ are formulas, then $[\phi \wedge \psi]$ is a formula
5. If ϕ and ψ are formulas, then $[\phi \vee \psi]$ is a formula
6. If ϕ and ψ are formulas, then $[\phi \rightarrow \psi]$ is a formula
7. If ϕ is a formula, and x is a variable, then $\exists x\phi$ is a formula
8. If ϕ is a formula, and x is a variable, then $\forall x\phi$ is a formula
9. Nothing is a formula except on the basis of 1–8

So, the syntax of *DPL* is that of ordinary predicate logic.

A model M is a pair $\langle D, F\rangle$, where D is a non-empty set of individuals, F an interpretation function, having as its domain the individual constants and predicates. If α is an individual constant, then $F(\alpha) \in D$; if α is an n-place predicate, then $F(\alpha) \subseteq D^n$. An assignment g is a function assigning an individual to each variable: $g(x) \in D$. G is the set of all assignment functions. Next, we define $[\![t]\!]_g = g(t)$ if t is a variable, and $[\![t]\!]_g = F(t)$ if t is an individual constant. Finally, we define the interpretation function $[\![\ ]\!]_M^{DPL} \subseteq G \times G$ as follows. (As usual, we suppress subscripts and superscripts whenever this does not give rise to confusion.)

Definition 2 (Semantics)

1. $[\![Rt_1 \ldots t_n]\!] = \{\langle g, h\rangle \mid h = g \ \&\ \langle [\![t_1]\!]_h \ldots [\![t_n]\!]_h\rangle \in F(R)\}$
2. $[\![t_1 = t_2]\!] = \{\langle g, h\rangle \mid h = g \ \&\ [\![t_1]\!]_h = [\![t_2]\!]_h\}$
3. $[\![\neg\phi]\!] = \{\langle g, h\rangle \mid h = g \ \&\ \neg\exists k\colon \langle h, k\rangle \in [\![\phi]\!]\}$
4. $[\![\phi \wedge \psi]\!] = \{\langle g, h\rangle \mid \exists k\colon \langle g, k\rangle \in [\![\phi]\!] \ \&\ \langle k, h\rangle \in [\![\psi]\!]\}$
5. $[\![\phi \vee \psi]\!] = \{\langle g, h\rangle \mid h = g \ \&\ \exists k\colon \langle h, k\rangle \in [\![\phi]\!] \vee \langle h, k\rangle \in [\![\psi]\!]\}$
6. $[\![\phi \rightarrow \psi]\!] = \{\langle g, h\rangle \mid h = g \ \&\ \forall k\colon \langle h, k\rangle \in [\![\phi]\!] \Rightarrow \exists j\colon \langle k, j\rangle \in [\![\psi]\!]\}$
7. $[\![\exists x\phi]\!] = \{\langle g, h\rangle \mid \exists k\colon k[x]g \ \&\ \langle k, h\rangle \in [\![\phi]\!]\}$
8. $[\![\forall x\phi]\!] = \{\langle g, h\rangle \mid h = g \ \&\ \forall k\colon k[x]h \Rightarrow \exists j\colon \langle k, j\rangle \in [\![\phi]\!]\}$

Besides the clauses that were discussed in the previous section, definition 2 also contains a clause which gives the interpretation of identity statements. It will come as no surprise that such statements are interpreted as tests.

3.2 Meaning, truth and equivalence

The notion of the interpretation of a formula that the semantics of *DPL* specifies, differs from the one we are familiar with from *PL*. The latter can be given in the form of a recursive specification of a set of assignments, those which satisfy a formula, whereas the semantics stated above defines a recursive notion of a set of pairs of assignments, those which are proper input-output pairs.

The notion of interpretation of *PL* brings along a notion of truth with respect to an assignment which is defined as follows: ϕ is true with respect to g iff g is an element of the set denoted by ϕ. In the present, essentially richer scheme a similar notion can be defined. We call a formula true with respect to an assignment g in a model M iff with g as input, it has an output:

Definition 3 (Truth) ϕ is *true with respect to g in M* iff $\exists h\colon \langle g, h\rangle \in [\![\phi]\!]_M$

In terms of this notion, we define when a formula is *valid* and when it is a *contradiction*:

Definition 4 (Validity) ϕ is *valid* iff $\forall M \forall g$: ϕ is true with respect to g in M

Definition 5 (Contradictoriness) ϕ is a *contradiction* iff $\forall M \forall g$: ϕ is false with respect to g in M

Notice that the interpretation of any contradiction is always the empty set. For valid formulas things are different: no unique semantic object serves as their interpretation. They either denote the identity relation on G, or a certain extension of this. For example, $Px \vee \neg Px$ always denotes the set of all pairs $\langle g, g\rangle$, but $\exists x[Px \vee \neg Px]$ denotes the set of all pairs $\langle g, h\rangle$ such that h differs at most with respect to x from g. Both formulas are valid according to definition 4, since both are true with respect to any g in any M. So, whereas semantically there is only one contradiction, there are many different tautologies. What distinguishes these can be expressed in terms of the variables they bind.

The set of all assignments with respect to which a formula is true, we call its *satisfaction set*, and we denote it by '$\backslash\ \ \backslash_M$':

Definition 6 (Satisfaction set) $\backslash\phi\backslash_M = \{g \mid \exists h\colon \langle g, h\rangle \in [\![\phi]\!]_M\}$

So, truth with respect to g in M can also be defined as $g \in \backslash\phi\backslash_M$, validity as $\backslash\phi\backslash_M = G$ for every M, and contradictoriness as $\backslash\phi\backslash_M = \emptyset$ for every M.

The notion of a satisfaction set is of the same type as the notion of interpretation in *PL*. But truth conditions do not exhaust dynamic meaning. The satisfaction set of a compound formula can not always be defined in terms of the satisfaction sets of its compounds, it is determined by its compositional interpretation in terms of the notion $[\![\ \ ]\!]$. The latter gives the building blocks of meaning. And meaning in its turn determines, globally but not locally, what the truth conditions of compound expressions are.

These considerations make clear that the notion of equivalence of standard logic, that of two formulas having the same truth conditions, although definable in *DPL* in terms of the notion of a satisfaction set, has only a marginal role to play. We call it *s-equivalence*, and denote it by '$\simeq_s$':

Definition 7 (s-equivalence) $\phi \simeq_s \psi \Leftrightarrow \forall M\colon \backslash\phi\backslash_M = \backslash\psi\backslash_M$

Full equivalence of two formulas requires that their interpretations be the same in every model. We call it *equivalence* simpliciter, and denote it by '$\simeq$':

Definition 8 (Equivalence) $\phi \simeq \psi \Leftrightarrow \forall M\colon [\![\phi]\!]_M = [\![\psi]\!]_M$

Of course, if two formulas are equivalent they will also have the same satisfaction set, i.e. they will be s-equivalent:

Fact 1 $\phi \simeq \psi \Rightarrow \phi \simeq_s \psi$

The reverse does not hold. For example, $\exists x Px$ and $\exists y Py$ have the same satisfaction sets, G or $\emptyset$, but they differ in meaning, since they produce different output assignments, viz., $\{h \mid h(x) \in F(P)\}$ and $\{h \mid h(y) \in F(P)\}$ respectively. The first formula has the potential to bind free occurrences of x in formulas to come, and the second has the potential to bind occurrences of y. We can formulate this in terms of the notion of the *production set*of a formula, the set consisting of those assignments which are its possible outputs, which we write as '$/\ \ /_M$':

Definition 9 (Production set) $/\phi/_M = \{h \mid \exists g\colon \langle g, h\rangle \in [\![\phi]\!]_M\}$

Whereas the satisfaction sets of $\exists x Px$ and $\exists y Py$ are the same, their production sets are different. If two formulas always have the same production set, we call them *p-equivalent*, denoted by '$\simeq_p$':

Definition 10 (p-equivalence) $\phi \simeq_p \psi \Leftrightarrow \forall M\colon /\phi/_M = /\psi/_M$

Of course, analogous to the previous fact, we have:

Fact 2 $\phi \simeq \psi \Rightarrow \phi \simeq_p \psi$

So, if two formulas have the same meaning, they always have the same satisfaction set and the same production set. However, the reverse does not hold:

Fact 3 $\phi \simeq_s \psi \ \&\ \phi \simeq_p \psi \not\Rightarrow \phi \simeq \psi$

If two formulas always have the same satisfaction set and always the same production set, this does not imply that they have the same meaning. So, meaning can not be defined in terms of satisfaction and production sets. Consider the following simple example. The two tautologies $Px \vee \neg Px$ and $\exists x[Px \vee \neg Px]$ both have the total set of assignments G as their satisfaction set and as their production set. But, as we have seen above, their meanings are different. The interpretation of the former is $\{\langle g, h\rangle \mid g = h\}$, and that of the latter is $\{\langle g, h\rangle \mid h[x]g\}$.

We end this section with the definitions of two other notions that will prove useful for what is to come.

As we have seen in the previous section, various kinds of *DPL*-formulas have the characteristic that they do not pass on bindings created by expressions which occur in them. They function as a kind of 'test' in this sense that they examine whether an input assignment meets a certain condition, return it as output if it does, and reject it otherwise. Semantically, they can be characterized as follows:

Definition 11 (Test) ϕ is a *test* iff $\forall M \forall g \forall h\colon \langle g, h\rangle \in [\![\phi]\!]_M \Rightarrow g = h$

Notice that for a test ϕ the definition of truth with respect to g given above boils down to $\langle g, g\rangle \in [\![\phi]\!]$. Also, we observe that for tests equivalence, s-equivalence and p-equivalence coincide.

Fact 4 If ϕ and ψ are tests, then: $\phi \simeq_s \psi \Leftrightarrow \phi \simeq \psi \Leftrightarrow \phi \simeq_p \psi$

The notion of a test is a semantic one. A partial syntactic characterization can be given as follows. In view of their semantic interpretation, atomic formulas, negations, implications, disjunctions, and universally quantified formulas are tests. Further, it holds that a conjunction of tests is a test. We will refer to this syntactically delineated class of formulas as *conditions*:

Definition 12 (Conditions)

1. If ϕ is an atomic formula, a negation, a disjunction, or an implication, then ϕ is a condition;
2. If ϕ and ψ are conditions, then $[\phi \wedge \psi]$ is a condition;
3. Nothing is a condition except on the basis of 1 or 2.

And we note the following fact:

Fact 5 If ϕ is a condition, then ϕ is a test

With the exception of contradictions, which have the empty set as their interpretation, and hence are tests, the syntactic notion of a condition characterizes the semantic notion of a test:

Fact 6 ϕ is a test iff ϕ is a condition or a contradiction

3.3 Scope and binding

A distinctive feature of *DPL* is that it allows for existential quantifiers to bind variables which are outside their syntactic scope. In this section we give a syntactic characterization of when an occurrence of a variable is bound by an occurrence of a quantifier. This characterization will consist of a simultaneous recursive definition of three notions:

- $\mathbf{bp}(\phi)$, the set of *binding pairs* in ϕ;
- $\mathbf{aq}(\phi)$, the set of *active quantifier occurrences* in ϕ;
- $\mathbf{fv}(\phi)$, the set of *free occurrences of variables* in ϕ.

A binding pair consist of a quantifier occurrence and a variable occurrence such that the first binds the second. An active quantifier occurrence is one which has the potential to bind occurrences of the corresponding variable further on. A free occurrence of a variable is one which is not in any binding pair. The definition, which is a bit sloppy since we have refrained from explicitly introducing a notation for occurrences, is as follows:

Definition 13 (Scope and binding)

1. $\mathbf{bp}(Rt_1, \ldots, t_n) = \emptyset$
   $\mathbf{aq}(Rt_1, \ldots, t_n) = \emptyset$
   $\mathbf{fv}(Rt_1, \ldots, t_n) = \{t_i \mid t_i \text{ a variable}\}$
2. $\mathbf{bp}(\neg\phi) = \mathbf{bp}(\phi)$
   $\mathbf{aq}(\neg\phi) = \emptyset$
   $\mathbf{fv}(\neg\phi) = \mathbf{fv}(\phi)$
3. $\mathbf{bp}(\phi \wedge \psi) = \mathbf{bp}(\phi) \cup \mathbf{bp}(\psi) \cup \{\langle \exists x, x\rangle \mid \exists x \in \mathbf{aq}(\phi) \ \& \ x \in \mathbf{fv}(\psi)\}$
   $\mathbf{aq}(\phi \wedge \psi) = \mathbf{aq}(\psi) \cup \{\exists x \in \mathbf{aq}(\phi) \mid \exists x \notin \mathbf{aq}(\psi)\}$
   $\mathbf{fv}(\phi \wedge \psi) = \mathbf{fv}(\phi) \cup \{x \in \mathbf{fv}(\psi) \mid \exists x \notin \mathbf{aq}(\phi)\}$
4. $\mathbf{bp}(\phi \vee \psi) = \mathbf{bp}(\phi) \cup \mathbf{bp}(\psi)$
   $\mathbf{aq}(\phi \vee \psi) = \emptyset$
   $\mathbf{fv}(\phi \vee \psi) = \mathbf{fv}(\phi) \cup \mathbf{fv}(\psi)$
5. $\mathbf{bp}(\phi \rightarrow \psi) = \mathbf{bp}(\phi) \cup \mathbf{bp}(\psi) \cup \{\langle \exists x, x\rangle \mid \exists x \in \mathbf{aq}(\phi) \ \& \ x \in \mathbf{fv}(\psi)\}$
   $\mathbf{aq}(\phi \rightarrow \psi) = \emptyset$
   $\mathbf{fv}(\phi \rightarrow \psi) = \mathbf{fv}(\phi) \cup \{x \in \mathbf{fv}(\psi) \mid \exists x \notin \mathbf{aq}(\phi)\}$
6. $\mathbf{bp}(\exists x\phi) = \mathbf{bp}(\phi) \cup \{\langle \exists x, x\rangle \mid x \in \mathbf{fv}(\phi)\}$
   $\mathbf{aq}(\exists x\phi) = \mathbf{aq}(\phi) \cup \{\exists x\}$, if $\exists x \notin \mathbf{aq}(\phi)$, $= \mathbf{aq}(\phi)$ otherwise
   $\mathbf{fv}(\exists x\phi) = \mathbf{fv}(\phi)$ minus the occurrences of x in ϕ
7. $\mathbf{bp}(\forall x\phi) = \mathbf{bp}(\phi) \cup \{\langle \forall x, x\rangle \mid x \in \mathbf{fv}(\phi)\}$
   $\mathbf{aq}(\forall x\phi) = \emptyset$
   $\mathbf{fv}(\forall x\phi) = \mathbf{fv}(\phi)$ minus the occurrences of x in ϕ

The 'test'-like character of atomic formulas, negations, disjunctions, implications and universally quantified formulas, is reflected in the above definition by the fact that for such formulas ϕ, $\mathbf{aq}(\phi) = \emptyset$, i.e., no quantifier occurring in such a formula is able to bind occurrences of the corresponding variable further on. Notice that if we conjoin two formulas which each have an empty set

of active quantifier occurrences, the resulting conjunction has no active occurrences either. This reminds us of the notion of a condition defined above. In fact, the requirement that $\mathbf{aq}(\phi) = \emptyset$ characterizes those ϕ which are conditions, and hence it also characterizes those ϕ which are tests, with the exception of contradictions.

The extra(-ordinary) binding power of the existential quantifier, the fact that it is externally dynamic, is reflected in clause 6. The occurrence of $\exists x$ in $\exists x\phi$ is added to the active occurrences of ϕ, unless, of course, there is already an active occurrence of that same quantifier in ϕ, in which case the latter remains the active occurrence. It is precisely in this respect that the binding properties of existential and universal quantification differ. Only existential quantifiers can have active occurrences, and for any formula, only one occurrence of a quantifier in that formula can be active.

That disjunction is internally static is reflected in the first line of clause 4. The set of binding pairs of a disjunction is simply the union of the binding pairs of its disjuncts. So, no binding relations are possible across the disjuncts of a disjunction. In contrast to this, the binding pairs of a conjunction or an implication are not simply obtained by putting together the binding pairs of the constituent formulas; what is further added are pairs consisting of active occurrences of quantifiers in the first conjunct or in the antecedent, together with free occurrences of the corresponding variables in the second conjunct or in the consequent.

In addition to the three notions just introduced, we define a fourth one, which will turn out to be convenient when we compare *DPL* and *PL* in section 4.1. It is the set of *scope pairs*, $\mathbf{sp}(\phi)$, of a formula ϕ:

Definition 14 (Scope pairs)

1. $\mathbf{sp}(Rt_1, \ldots, t_n) = \emptyset$
2. $\mathbf{sp}(\neg\phi) = \mathbf{sp}(\phi)$
3. $\mathbf{sp}(\phi \wedge \psi) = \mathbf{sp}(\phi) \cup \mathbf{sp}(\psi)$
4. $\mathbf{sp}(\phi \vee \psi) = \mathbf{sp}(\phi) \cup \mathbf{sp}(\psi)$
5. $\mathbf{sp}(\phi \rightarrow \psi) = \mathbf{sp}(\phi) \cup \mathbf{sp}(\psi)$
6. $\mathbf{sp}(\exists x\phi) = \mathbf{sp}(\phi) \cup \{\langle \exists x, x\rangle \mid x \in \mathbf{fv}(\phi)\}$
7. $\mathbf{sp}(\forall x\phi) = \mathbf{sp}(\phi) \cup \{\langle \forall x, x\rangle \mid x \in \mathbf{fv}(\phi)\}$

We note two things about this definition. First, if we replace the notion $\mathbf{fv}(\phi)$, the notion of a free variable in *DPL* as defined above, by the notion $\mathbf{fv}_{PL}(\phi)$, the notion of a free variable in *PL*, we end up with the notion of binding in *PL*. Secondly, concerning *DPL* itself again, the following fact can be proved by simple induction:

Fact 7 $\mathbf{sp}(\phi) \subseteq \mathbf{bp}(\phi)$

So, it is sufficient but not necessary for a variable to be bound by a quantifier that it occurs in its scope. Of course, in some cases, all variables bound by a quantifier are also inside its scope. This holds for example for $\exists x[Px \wedge Qx]$: $\mathbf{bp}(\exists x[Px \wedge Qx]) = \mathbf{sp}(\exists x[Px \wedge Qx])$. In section 3.6, we will show that for any formula ϕ there is a formula ϕ' which is equivalent in *DPL* to ϕ, such that $\mathbf{bp}(\phi') = \mathbf{sp}(\phi')$.

For some purposes it is convenient to talk about the free variables of a formula, rather than about their occurrences. We will write the set of free variables in ϕ as '$FV(\phi)$':

Definition 15 $x \in FV(\phi)$ iff there is an occurrence of x in $\mathbf{fv}(\phi)$

For similar reasons, we introduce the notion of the set of variables x such that there is an active occurrence of $\exists x$ in ϕ, and denote it as '$AQV(\phi)$':

Definition 16 $x \in AQV(\phi)$ iff $\exists x \in \mathbf{aq}(\phi)$

It is useful to point out the following two facts, which both can be proven by simple induction on the complexity of ϕ (by $g =_{FV(\phi)} h$ we mean that for all $x \in FV(\phi)\colon g(x) = h(x)$):

Fact 8 If $g =_{FV(\phi)} h$, then $\forall M\colon g \in \backslash\phi\backslash_M \Leftrightarrow h \in \backslash\phi\backslash_M$

Fact 9 If $\exists M\colon \langle g, h\rangle \in [\![\phi]\!]_M$ & $g(x) \neq h(x)$, then $x \in AQV(\phi)$

Fact 8 says that if two assignments differ in that the one is in the satisfaction set of a formula ϕ, whereas the other is not, they should also differ in the value they assign to at least one of the free variables of ϕ. And fact 9 says that if two assignments which assign a different value to a certain variable x form an input-output pair in the interpretation of a formula ϕ, then there is an active occurrence of the quantifier $\exists x$ in ϕ.

3.4 Some logical facts

Let us now turn to an exposition of some basic logical facts, which will illustrate various properties of *DPL*.

We start with the interdefinablity of the logical constants. A simple calculation with the relevant clauses of definition 2 shows that negation, conjunction and existential quantification can be used as our basic logical constants, the others being definable in terms of them in the usual way:

$$\phi \rightarrow \psi \simeq \neg[\phi \wedge \neg\psi]$$

$$\phi \vee \psi \simeq \neg[\neg\phi \wedge \neg\psi]$$

$$\forall x\phi \simeq \neg\exists x\neg\phi$$

It should be noted, though, that contrary to what is the case in ordinary predicate logic, a different choice of basic constants is not possible. The following facts show that we cannot do with the universal quantifier and disjunction, nor with the universal quantifier and implication:

$$\phi \wedge \psi \not\simeq \neg[\phi \rightarrow \neg\psi]$$

$$\phi \wedge \psi \not\simeq \neg[\neg\phi \vee \neg\psi]$$

$$\exists x\phi \not\simeq \neg\forall x\neg\phi$$

The reason for this is, of course, that the expressions on the right are tests, which lack the dynamic binding properties of the expressions on the left. In the first and in the last case, the satisfaction sets, i.e., the truth conditions, of the expressions on the right and of those on the left indeed are the same: they are s-equivalent. This does not hold in the second case, because of the fact that disjunction is not only externally, but also internally static.

$$\phi \wedge \psi \simeq_s \neg[\phi \rightarrow \neg\psi]$$

$$\phi \wedge \psi \not\simeq_s \neg[\neg\phi \vee \neg\psi]$$

$$\exists x\phi \simeq_s \neg\forall x\neg\phi$$

Furthermore, we may note that, whereas disjunction can be defined in terms of implication, the reverse does not hold:

$$\phi \vee \psi \simeq \neg\phi \rightarrow \psi$$

$$\phi \rightarrow \psi \not\simeq \neg\phi \vee \psi$$

Disjunctions and implications are both tests, they are externally static. However, an implication is internally dynamic, i.e., an existential quantifier in the antecedent can bind variables in the consequent. But no such binding relations are possible between the disjuncts of a disjunction, the latter being also internally static. This is also the reason why in the last case not even the truth conditions of the expressions on the right and on the left are the same: no s-equivalence obtains in this case:

$$\phi \rightarrow \psi \not\simeq_s \neg\phi \vee \psi$$

In some special cases, some of these non-equivalences do hold. For example, if (and only if) $\phi \wedge \psi$ is a test, it is equivalent with $\neg[\phi \rightarrow \neg\psi]$. And if ϕ is a test, or more generally if no binding relations exist between ϕ and ψ, i.e., if $AQV(\phi) \cap FV(\psi) = \emptyset$, then $\phi \rightarrow \psi$ is equivalent with $\neg\phi \vee \psi$. Similarly, $\exists x\phi$ is equivalent with $\neg\forall x\neg\phi$ iff $\exists x\phi$ is a test, which it is only if ϕ is a contradiction.

A relationship between the existential and the universal quantifier that does hold unconditionally is:

$$\neg\exists x\phi \simeq \forall x\neg\phi$$

Of course, this follows from the fact that negation turns anything into a test.

From the latter observation, we may conclude that the law of double negation will not hold unconditionally. Consider a formula ϕ that is not a test. Negating ϕ results in the test $\neg\phi$, and a second negation, which gives $\neg\neg\phi$, does not reverse this effect. And this seems correct, since a doubly negated sentence in general does not allow subsequent pronouns to refer back to elements in the scope of the negations. (But see section 5.1 for some further discussion.) Precisely in this respect, ϕ and $\neg\neg\phi$ may differ in meaning. However, as far as their truth conditions are concerned, the two coincide, so ϕ and $\neg\neg\phi$ are s-equivalent. We can formulate the following restricted versions of the law of double negation:

$$\phi \simeq_s \neg\neg\phi$$

$$\neg\neg\phi \simeq \phi \text{ iff } \phi \text{ is a test}$$

Hence, double negation is not in general eliminable. The effect of applying double negation is that the meaning of a formula is restricted, so to speak, to its truth conditions. It is useful to introduce an operator, $\Diamond$, which performs this function:

Definition 17 (Closure) $[\![\Diamond\phi]\!] = \{\langle g, h\rangle \mid g = h \;\&\; \exists k\colon \langle h, k\rangle \in [\![\phi]\!]\}$

One can look upon $\Diamond$ as a kind of assertion or closure operator. It can be used to close off a piece of discourse, blocking any further anaphoric reference, stating: this is how things stand.

We notice that the following hold:

$$\Diamond\phi \simeq \neg\neg\phi \simeq \phi \text{ iff } \phi \text{ is a test}$$

$$\Diamond\phi \simeq \Diamond\psi \Leftrightarrow \phi \simeq_s \psi$$

$$\Diamond\Diamond\phi \simeq \Diamond\phi \simeq \neg\neg\phi$$

$$\Diamond\neg\phi \simeq \neg\phi \simeq \neg\Diamond\phi$$

In terms of the operator $\Diamond$ we can also state the restricted versions of the interdefinability of the logical constants discussed above:

$$\Diamond[\phi \wedge \psi] \simeq \neg[\phi \rightarrow \neg\psi]$$

$$\Diamond\phi \wedge \Diamond\psi \simeq \neg[\neg\phi \vee \neg\psi]$$

$$\Diamond\exists x\phi \simeq \neg\forall x\neg\phi$$

$$\Diamond\phi \rightarrow \psi \simeq \neg\phi \vee \psi$$

Let us now turn to some properties of conjunction. First of all, it can be noticed that conjunction is associative:

$$[\phi \wedge \psi] \wedge \chi \simeq \phi \wedge [\psi \wedge \chi]$$

Notice that associativity holds despite the increased binding power of the existential quantifier. This is so because if two conjuncts each contain an active occurrence of the same quantifier, it is the rightmost one which is active in the conjunction as a whole, the left one being 'de-activated'. Compare $[\exists x Px \wedge \psi] \wedge \exists x Qx$ with $\exists x Px \wedge [\psi \wedge \exists x Qx]$. The last occurrence of $\exists x$ is the active one. Hence, it is this occurrence that binds the x in Hx, both in $[[\exists x Px \wedge \psi] \wedge \exists x Qx] \wedge Hx$, and in $[\exists x Px \wedge [\psi \wedge \exists x Qx]] \wedge Hx$. The structure of the respective conjunctions is irrelevant in this respect.

Conjunction is not unconditionally commutative, however, as the simple example of $\exists x Px \wedge Qx$ and $Qx \wedge \exists x Px$ shows. In fact the latter of these two formulas is a counterexample against idempotency of conjunction as well.

$$\phi \wedge \psi \not\simeq \psi \wedge \phi$$

$$\phi \not\simeq \psi \wedge \phi$$

Of course, if both ϕ and ψ are tests, commutativity holds, and if ϕ is a test idempotency holds:

$$\Diamond\phi \wedge \Diamond\psi \simeq \Diamond\psi \wedge \Diamond\phi$$

$$\Diamond\phi \simeq \Diamond\phi \wedge \Diamond\phi$$

That ϕ is a test is not a necessary condition for idempotency of conjunction to hold. It is sufficient that active occurrences of a quantifier in ϕ are unable to bind free variables in ϕ:

$$AQV(\phi) \cap FV(\phi) = \emptyset \Rightarrow \phi \simeq \phi \wedge \phi$$

This condition isn't a necessary one either, e.g. $Px \wedge \exists x Px \simeq [Px \wedge \exists x Px] \wedge [Px \wedge \exists x Px]$.

Similarly, ϕ and ψ need not necessarily be both tests for commutativity of conjunction to hold. An example of a conjunction which does not consist of tests, but which nevertheless is commutative is $\exists x Px \wedge Qy$, which has the same meaning as $Qy \wedge \exists x Px$. Commuting this conjunction does not interfere with its binding pattern. In general, if commuting the conjuncts does not change in the binding pairs, nor the active occurrences of quantifiers in the conjunction, commutativity holds:

$$\left.\begin{array}{l} AQV(\phi) \cap FV(\psi) = \emptyset \\ AQV(\psi) \cap FV(\phi) = \emptyset \\ AQV(\phi) \cap AQV(\psi) = \emptyset \end{array}\right\} \Rightarrow \phi \wedge \psi \simeq \psi \wedge \phi$$

In this case, too, the conditions are sufficient but not necessary. A case in point is the contradiction $[Px \wedge \neg Px] \wedge \exists x Qx$.

As is to be expected, disjunction, being both internally and externally static, is unconditionally idempotent, commutative and associative:

$$\phi \simeq \phi \vee \phi$$

$$\phi \vee \psi \simeq \psi \vee \phi$$

$$\phi \vee [\psi \vee \chi] \simeq [\phi \vee \psi] \vee \chi$$

Idempotency and commutativity of disjunction reflect that there cannot be any anaphoric relations across disjuncts. (But see sections 4.3 and 5.1 for some discussion.)

As for the classical de Morgan laws, *DPL* validates the following:

$$\Diamond[\phi \wedge [\psi \vee \chi]] \simeq [\phi \wedge \psi] \vee [\phi \wedge \chi]$$

$$\phi \vee [\Diamond\psi \wedge \chi] \simeq [\phi \vee \psi] \wedge [\phi \vee \chi]$$

The latter is a special instance of:

$$AQ(\psi) \cap FV(\chi) = \emptyset \Rightarrow \phi \vee [\psi \wedge \chi] \simeq [\phi \vee \psi] \wedge [\phi \vee \chi]$$

Turning to implication, we may observe that the following form of contraposition goes through unconditionally:

$$[\neg\phi \rightarrow \psi] \simeq [\neg\psi \rightarrow \phi]$$

But for the general case we need, again, the condition that no binding pairs are distorted:

$$[\Diamond\phi \rightarrow \psi] \simeq [\neg\psi \rightarrow \neg\phi]$$

$$AQV(\phi) \cap FV(\psi) = \emptyset \Rightarrow [\phi \rightarrow \psi] \simeq [\neg\psi \rightarrow \neg\phi]$$

The reason that we need a condition here, is that quantifiers in the antecedent of an implication may bind variables in the consequent. Implications, as we noted repeatedly, are internally dynamic. But no outside binding effects are permitted, they are externally static, and this is reflected in the following two equivalences:

$$[\phi \rightarrow \psi] \simeq \Diamond[\phi \rightarrow \psi]$$

$$[\phi \rightarrow \psi] \simeq [\phi \rightarrow \Diamond\psi]$$

The first equivalence is another way of saying that an implication is a test, the second expresses that an implication turns its consequent into a test.

A last fact concerning implication that we want to note, is the following:

$$\phi \rightarrow [\psi \rightarrow \chi] \simeq [\phi \wedge \psi] \rightarrow \chi$$

Finally, we notice some facts concerning the interplay of quantifiers and connectives:

$$\exists x\phi \wedge \psi \simeq \exists x[\phi \wedge \psi]$$

$$x \notin (FV(\phi) \cup AQV(\phi)) \Rightarrow \phi \wedge \exists x\psi \simeq \exists x[\phi \wedge \psi]$$

The first fact illustrates the dynamics of the existential quantifier: its binding power extends indefinitely to the right. This is what makes *DPL* a suitable instrument for the representation of antecendent-anaphor relations across sentence boundaries. The second fact states under which condition the scope of an existential quantifier may be extended to the left in a conjunction: under the usual condition that the left conjunct has no free occurrences of x, and further that the active occurrence of $\exists x$ is not 'de-activated' by an occurrence of that same quantifier in the first conjunct.

The following equivalence is important for the analysis of 'donkey'- like cases of anaphora:

$$\exists x\phi \rightarrow \psi \simeq \forall x[\phi \rightarrow \psi]$$

Existential quantifiers in the antecedent of an implication may bind occurrences of variables in the consequent, and they have 'universal' force.

One final observation:

$$\exists x\phi \not\simeq \exists y[y/x]\phi$$

Here, $[y/x]\phi$ denotes, as usual, the result of replacing all free occurrences of x in ϕ by y. This non-equivalence illustrates the fact that bound variables in *DPL* are 'more meaningful' expressions than in *PL*. Notice that $\exists x\phi$ and $\exists y[y/x]\phi$ are s-equivalent if no occurrence of y that is free in $\exists x\phi$ is bound in $\exists y[y/x]\phi$:

$$y \notin FV(\phi) \Rightarrow \exists x\phi \simeq_s \exists y[y/x]\phi$$

The above observations mark some of the ways in which the dynamic semantics of *DPL* differs from the ordinary, static interpretation of *PL*. A more detailed comparison can be found in section 4.1.

3.5 Entailment

In standard logic, ϕ entails ψ iff whenever ϕ is true, ψ is true as well. Since we have defined a notion of truth in *DPL*, we can also define an analogue of this notion of entailment for *DPL*. We will refer to it as *s-entailment*, and write it as '$\models_s$':

Definition 18 (s-entailment) $\phi \models_s \psi$ iff $\forall M \forall g$: if ϕ is true with respect to g in M, then ψ is true with respect to g in M

In other words, ϕ s-entails ψ iff $\forall M$: $\backslash\phi\backslash_M \subseteq \backslash\psi\backslash_M$. Obviously, s-equivalence as it was defined above, is mutual s-entailment.

Unlike the notion of entailment in *PL*, in *DPL* the notion of s-entailment does not coincide with that of meaning inclusion, which is denoted by '$\preceq$':

Definition 19 (Meaning inclusion) $\phi \preceq \psi$ iff $\forall M$: $[\![\phi]\!]_M \subseteq [\![\psi]\!]_M$

In *DPL*, meaning is a richer notion than in *PL*, where interpretation and satisfaction coincide. Meaning inclusion implies s-entailment, but not the other way around:

Fact 10 $\phi \preceq \psi \Rightarrow \phi \models_s \psi$

The notion of equivalence $\simeq$ defined in section 3.2 is nothing but mutual meaning inclusion.

In an important sense, the notion of s-entailment is not a truly dynamic notion of entailment. One way to illustrate this, is to point out the fact that the notion of s-entailment does not correspond in the usual way to implication. For example, although it holds that $\models_s \exists x Px \rightarrow Px$, we have $\exists x Px \not\models_s Px$. Whereas in an implication an existential quantifier in the antecedent can bind variables in the consequent, the notion of s-entailment does not account for similar binding relations between premiss and conclusion. However, in natural language, such relations do occur. From *A man came in wearing a hat*, we may conclude *So, he wore a hat*, where the pronoun in the conclusion is anaphorically linked to the indefinite term in the premiss. As we have just seen, if we want to account for this, the notion of s-entailment is not the one we are after. For similar reasons, meaning inclusion is not what we are looking for either. It is too strict: $\exists x Px \not\preceq Px$. And it is also not strict enough. For $\preceq$ is reflexive, but, as is argued below, dynamic entailment is not: $Px \wedge \exists x Qx$ does not entail $Px \wedge \exists x Qx$.

Hence, we have to find another, an inherently dynamic notion of entailment. Taking up our processing metaphor once more, which means looking at sentences as a kind of programs, a reasonably intuitive notion is the following. We say that ϕ entails ψ if every successful execution of ϕ guarantees a succesful execution of ψ. Or, to put it slightly differently, ϕ entails ψ iff every assignment that is a possible output of ϕ is a possible input for ψ. This is captured in the following definition of dynamic entailment:

Definition 20 (Entailment)

$$\phi \models \psi \text{ iff } \forall M\, \forall g \forall h\colon \langle g, h\rangle \in [\![\phi]\!]_M \Rightarrow \exists k\colon \langle h, k\rangle \in [\![\psi]\!]_M$$

Using the notions of satisfaction set and production set, we can write this more economically as:

$$\phi \models \psi \text{ iff } \forall M\colon /\phi/_M \subseteq \backslash\psi\backslash_M$$

As requested, the notion of dynamic entailment corresponds in the usual way to the interpretation of implication:

Fact 11 (Deduction theorem) $\phi \models \psi$ iff $\models \phi \rightarrow \psi$

Entailment is related to s-entailment in the following way:

Fact 12 $\phi \models_s \psi$ iff $\Diamond\phi \models \psi$

More generally, entailment and s-entailment coincide if no binding relations exist between premiss and conclusion:

Fact 13 If $AQV(\phi) \cap FV(\psi) = \emptyset$, then: $\phi \models_s \psi \Leftrightarrow \phi \models \psi$

We note further that mutual entailment of ϕ and ψ does not mean that ϕ and ψ are equivalent. For example, $\exists x Px$ and Px do entail each other, but they are not equivalent. The same pair of formulas illustrates that entailment does not imply meaning inclusion. And the reverse does not hold in general either. For example, the meaning of $Qx \wedge \exists x Px$ includes the meaning of $Qx \wedge \exists x Px$, but the latter does not entail the former. Meaning inclusion does imply entailment if there are no binding relations between premiss and conclusion:

Fact 14 If $AQV(\phi) \cap FV(\psi) = \emptyset$, then: $\phi \preceq \psi \Rightarrow \phi \models \psi$

In the proof of this fact, the two facts 8 and 9 stated in section 3.3 play a central role. Suppose $AQV(\phi) \cap FV(\psi) = \emptyset$ and $\phi \preceq \psi$. Let $h \in /\phi/_M$, that is $\exists g\colon \langle g, h\rangle \in [\![\phi]\!]_M$. Since, if $\langle g, h\rangle \in [\![\phi]\!]_M$, then $h[AQV(\phi)]g$ (fact 9), and $AQV(\phi) \cap FV(\psi) = \emptyset$, it holds for all $x \in FV(\psi)$ that $g(x) = h(x)$. Since $\phi \preceq \psi$, it also holds that $\langle g, h\rangle \in [\![\psi]\!]_M$, and hence that $g \in \backslash\psi\backslash_M$. From $g(x) = h(x)$ for all $x \in FV(\psi)$, and $g \in \backslash\psi\backslash_M$, we may conclude on the basis of fact 8 that $h \in \backslash\psi\backslash_M$ as well.

Precisely because the notion of entailment is truly dynamic in the sense that it allows active quantifiers in a premiss to bind variables in the conclusion, it lacks some properties which more orthodox notions, such as s-entailment, do have, notably the properties of reflexivity and transitivity.

We already encountered a typical counterexample to reflexivity of dynamic entailment in the formula $Px \wedge \exists x Qx$, which does not entail itself. The reason is that in the occurrence of this formula as a conclusion, the variable x in the first conjunct gets bound by the quantifier in the occurrence of the formula as a premiss, whereas in the occurrence of the formula as a premiss it is free. The following restricted fact about reflexivity, however, does hold as an immediate consequence of fact 14 and the reflexivity of $\preceq$:

Fact 15 (Reflexivity) If $AQV(\phi) \cap FV(\phi) = \emptyset$, then $\phi \models \phi$

The condition on reflexivity given here is a sufficient, but not a necessary one. For example, the formulas $Px \wedge \exists x Px$, and $Px \wedge \exists x Px \wedge Qx$, both do entail themselves. Conditions similar to the one on reflexivity can be laid upon other facts about entailment known from ordinary predicate logic, in order to accommodate them to *DPL*. An example is the following:

$$\text{If } AQV(\psi) \cap FV(\psi) = \emptyset, \text{ then } \phi \wedge \psi \models \psi$$

We mention in passing that if one would use *DPL* for practical purposes, one would certainly choose active quantifiers and free variables in such a way that these troublesome cases are avoided.

Not only reflexivity, but transitivity, too, may fail when free occurrences of variables in a conclusion, are bound by a premiss. If we arrive at a conclusion χ

in several steps $\psi_1 \ldots \psi_n$ from an initial premiss ϕ, we cannot simply omit these intermediary steps, and conclude immediately from ϕ to χ. Roughly speaking, we first have to make sure that there are no antecedent-anaphor relations between one of the intermediate steps and the conclusion which are not due to a similar relation between premiss and conclusion. For example, although $\neg\neg\exists xPx \models \exists xPx$, and $\exists xPx \models Px$, we notice that $\neg\neg\exists xPx \not\models Px$.

The cases which present problems for transitivity can be characterized as follows. Suppose $\phi \models \psi$ and $\psi \models \chi$. If we want to conclude from this that $\phi \models \chi$, then problems may arise if $x \in FV(\chi)$ and $x \in AQV(\psi)$. Consider again $\neg\neg\exists xPx$, $\exists xPx$, and Px. Clearly, the first entails the second, and the second entails the third, without the first entailing the third. On the other hand, consider $\exists xPx$, $\exists xPx$, and Px, or $\exists xPx \wedge Qx$, $\exists xPx$, and Px. These are two cases where nothing goes wrong. So, not all cases where χ contains a free occurrence of x, and ψ contains an active ocurrence of $\exists x$ are to be excluded. Evidently, what also matters is what ϕ 'says' about x, in the dynamic sense of what constraint it puts on whatever free occurrences of x that are still to come. Roughly speaking, what ϕ says about variables which occur freely in χ and which are bound by ψ, should be at least as strong as what ψ says about them. So, what is needed is a stronger version of the notion of entailment that covers the condition that the premiss puts at least as strong a condition on certain variables as the conclusion does. This notion can be defined as follows, where by '$h =_{x_1 \ldots x_n} g$' we mean '$h(x_1) = g(x_1)$ & $\ldots$ & $h(x_n) = g(x_n)$':

Definition 21 **($x_1 \ldots x_n$-Entailment)**

$$\phi \models_{x_1 \ldots x_n} \psi \text{ iff } \forall M \forall g\colon g \in /\phi/_M \Rightarrow \exists h\colon \langle g, h\rangle \in [\![\psi]\!]_M \ \&\ h =_{x_1 \ldots x_n} g$$

Notice that $\phi \models_{x_1 \ldots x_n} \psi$ implies that $\phi \models \psi$, and that if $n = 0$, then $\phi \models_{x_1 \ldots x_n} \psi$ collapses into $\phi \models \psi$.

Now we are ready to state the following fact:

Fact 16 (Transitivity) $\phi \models_{AQV(\psi) \cap FV(\chi)} \psi \ \&\ \psi \models \chi \Rightarrow \phi \models \chi$

The proof of this fact runs as follows. Suppose $g \in /\phi/_M$. Then $\exists h\colon \langle g, h\rangle \in [\![\psi]\!]_M$, and for all $x \in AQV(\chi) \cap FV(\psi)$, it holds that $g(x) = h(x)$. Since $\psi \models \chi$, it holds that $h \in \backslash\chi\backslash_M$. For any variable $x \in FV(\chi)$, if $x \in AQV(\psi)$ then $g(x) = h(x)$ by assumption; but if $x \notin AQV(\psi)$, then also $g(x) = h(x)$, since $\langle g, h\rangle \in [\![\psi]\!]_M$ (use fact 9). Hence, we have $g(x) = h(x)$ for all variables $x \in FV(\chi)$, and hence $g \in \backslash\chi\backslash_M$ (fact 8).

The above shows that there are some complications inherent in the notion of dynamic entailment. These do pay off, however. We can translate 'natural language' reasonings in which pronouns are introduced in intermediary steps, directly into *DPL*. Consider the following, admittedly stylized, example and its translation into *DPL*:

1. It is not the case that nobody walks and talks. ($\neg\neg\exists x[Px \wedge Qx]$)

2. So, somebody walks and talks. ($\exists x[Px \wedge Qx]$)

3. So, he walks. (Px)

4. So, somebody walks. ($\exists xPx$)

5. So, it is not the case that nobody walks. ($\neg\neg\exists xPx$)

The interesting bit is the step from 2 to 3. The pronoun *he* occurring in 3 is bound by *somebody* in 2. So, although 1 implies 2, and 2 implies 3, 1 does not imply 3, precisely because 1 cannot, and should not, bind the pronoun in 3. But in the transition from 2 via 3 to 4, 3 can be omitted. And the same holds for all other intermediate steps. So, in the end, 5 is a consequence of 1.

Up to now, we have only discussed entailment with respect to a single premiss. It makes sense to generalize the definition of entailment given above in the following way:

Definition 22 (Entailment, general form)

$$\begin{aligned}\phi_1, \ldots, \phi_n \models \psi \quad \text{iff} \quad & \forall M \forall h \forall g_1 \ldots g_n \colon \langle g_1, g_2 \rangle \in [\![\phi_1]\!]_M \ \& \ \ldots \\ & \& \ \langle g_n, h \rangle \in [\![\phi_n]\!]_M \Rightarrow \exists k \colon \langle h, k \rangle \in [\![\psi]\!]_M\end{aligned}$$

Notice that it is not a set, but a sequence of formulas, a *discourse*, that can be said to entail a formula. In view of the above, this is not surprising. What holds, of course, is:

$$\phi_1, \ldots, \phi_n \models \psi \text{ iff } \phi_1 \wedge \ldots \wedge \phi_n \models \psi \text{ iff } \models [\phi_1 \wedge \ldots \wedge \phi_n] \rightarrow \psi$$

Since the order of the conjuncts matters for the interpretation of a conjunction, so will the order of the premisses matter for entailment. For example, although $\exists xPx, \exists xQx \models Qx$, we have $\exists xQx, \exists xPx \not\models Qx$. Further, we note that in a certain sense dynamic entailment is not monotonic. Whereas it holds unconditionally that if $\phi \models \psi$, then $\chi, \phi \models \psi$, we may not always conclude from $\phi \models \psi$ that $\phi, \chi \models \psi$. The reason for this being, again, that χ may interfere with bindings between ϕ and ψ, For example, it does hold that $\exists xPx \models Px$, but we have $\exists xPx, \exists xQx \not\models Px$.

Again, for practical purposes these complications can be evaded by a suitable choice of active quantifiers and free variables. For example, in adding a premiss which contains an active quantifier, we better choose one which does not already occur actively in one of the other premisses.

Such practical considerations are particularly important in designing a proof system. In cooperation with Roel de Vrijer, a sound and complete system of natural deduction for *DPL* has been developed. It will be presented in a separate paper.

4 Comparisons

In section 4.1, we compare *DPL* with ordinary predicate logic. In section 4.2, the relation between *DPL* and *DRT* is discussed. And in section 4.3, we turn to a comparison of *DPL* with quantificational dynamic logic.

4.1 *DPL* and *PL*

In discussing some basic logical facts concerning *DPL*, we have noticed a number of differences between *DPL* and *PL*, all arising from the essentially richer notion of binding of the former. In this section we show that this is indeed exactly the point at which the two systems differ. First we show that for any formula ϕ there is a formula ϕ' which is *DPL*-equivalent to ϕ, in which all variables bound by a quantifier are brought under its scope, i.e., for which it holds that $\mathbf{bp}(\phi') = \mathbf{sp}(\phi')$. Then we show that for any formula ϕ such that $\mathbf{bp}(\phi) = \mathbf{sp}(\phi)$, the truth conditions of ϕ in *DPL* and in *PL* coincide. If we put these two facts together, it follows that for any formula ϕ there is a formula ϕ' which is equivalent to it, and for which it holds that its truth conditions in *DPL* and *PL* are the same.

We already noticed above that the satisfaction set of a *DPL*-formula, the set of assigments with respect to which it is true, is the same type of semantic object as the *PL*-interpretation of a formula. Because *PL* and *DPL* have the same syntax, we may speak of the *PL*- and the *DPL*-interpretation of one and the same formula.

We first define the semantics of *PL* in the same kind of format we used for the semantics of *DPL*. *PL*-models are the same as *DPL*-models, as are assignments and the interpretation of terms. The definition of the interpretation function $[\![\ ]\!]^{PL}_M \subseteq G$ is as follows. (We drop subscripts whenever this does not lead to confusion, and we continue to use '$[\![\ ]\!]$' without a superscript to denote interpretation in *DPL*.)

Definition 23 (*PL*-semantics)

1. $[\![Rt_1 \ldots t_n]\!]^{PL} = \{g \mid \langle [\![t_1]\!]_g \ldots [\![t_n]\!]_g \rangle \in F(R)\}$
2. $[\![t_1 = t_2]\!]^{PL} = \{g \mid [\![t_1]\!]_g = [\![t_2]\!]_g\}$
3. $[\![\neg\phi]\!]^{PL} = \{g \mid g \notin [\![\phi]\!]^{PL}\}$
4. $[\![\phi \vee \psi]\!]^{PL} = \{g \mid g \in [\![\phi]\!]^{PL} \vee g \in [\![\psi]\!]^{PL}\}$
5. $[\![\phi \rightarrow \psi]\!]^{PL} = \{g \mid g \in [\![\phi]\!]^{PL} \Rightarrow g \in [\![\psi]\!]^{PL}\}$
6. $[\![\phi \wedge \psi]\!]^{PL} = \{g \mid g \in [\![\phi]\!]^{PL} \ \&\ g \in [\![\psi]\!]^{PL}\}$
7. $[\![\exists x\phi]\!]^{PL} = \{g \mid \exists k\colon k[x]g \ \&\ k \in [\![\phi]\!]^{PL}\}$

8. $[\![\forall x\phi]\!]^{PL} = \{g \mid \forall k\colon k[x]g \Rightarrow k \in [\![\phi]\!]^{PL}\}$

The set of assignments which is the interpretation of a formula, consists of those assignments which satisfy the formula: we call ϕ *true with respect to* g *in* M iff $g \in [\![\phi]\!]^{PL}_M$.

The satisfaction set $\backslash\phi\backslash$ of a formula in *DPL*, and its interpretation $[\![\phi]\!]^{PL}$ in *PL* are both sets of assignments. But the satisfaction set of a formula need not be the same as its *PL*-interpretation. For example, the satisfaction set of $\exists xPx \wedge Qx$ is not identical to its *PL*-interpretation. However, for the formula $\exists x[Px \wedge Qx]$, which is equivalent to $\exists xPx \wedge Qx$ in *DPL*, it does hold that its satisfaction set and its *PL*-interpretation are the same. The difference between the two is that in the latter all occurrences of x which are bound by the existential quantifier are also brought in its scope. Similarly, the satisfaction set and the *PL*-interpretation of $\exists xPx \rightarrow Qx$ are different, whereas the satisfaction set and the *PL*-interpretation of $\forall x[Px \rightarrow Qx]$, which is equivalent in *DPL* to $\exists xPx \rightarrow Qx$, are the same. Again, the difference between the two is that in the latter case all bound variables are brought under the scope of a quantifier.

In fact, for every formula ϕ there is a formula ϕ' which is equivalent to ϕ in *DPL* such that in ϕ' all variables which are bound by a quantifier, occur in its scope. We define a recipe $\flat$ which provides us with such a variant for every formula. We will call $\flat\phi$ the *normal binding form* of ϕ:

Definition 24 (*DPL* normal binding form)

1. $\flat Rt_1 \ldots t_n = Rt_1 \ldots t_n$
2. $\flat(t_1 = t_n) = (t_1 = t_n)$
3. $\flat\neg\psi = \neg\flat\psi$
4. $\flat[\psi_1 \vee \psi_2] = [\flat\psi_1 \vee \flat\psi_2]$
5. $\flat\exists x\psi = \exists x\flat\psi$
6. $\flat\forall x\psi = \forall x\flat\psi$
7. $\flat[\psi_1 \wedge \psi_2] =$
   (a) $\flat[\chi_1 \wedge [\chi_2 \wedge \psi_2]]$ if $\psi_1 = [\chi_1 \wedge \chi_2]$
   (b) $[\exists x\flat[\chi \wedge \psi_2]]$ if $\psi_1 = \exists x\chi$
   (c) $[\flat\psi_1 \wedge \flat\psi_2]$ otherwise
8. $\flat[\psi_1 \rightarrow \psi_2] =$
   (a) $\flat[\chi_1 \rightarrow [\chi_2 \rightarrow \psi_2]]$ if $\psi_1 = [\chi_1 \wedge \chi_2]$
   (b) $[\forall x\flat[\chi \rightarrow \psi_2]]$ if $\psi_1 = \exists x\chi$
   (c) $[\flat\psi_1 \rightarrow \flat\psi_2]$ otherwise

The interesting bit in this definition are clauses 7 and 8. Clause 7(a) rebrackets complex conjunctions in such a way that all closing brackets are moved to the right end side. For example, $[[Px \wedge Qx] \wedge Rx]$ is turned into $[Px \wedge [Qx \wedge Rx]]$, and $[[[Px \wedge Qx] \wedge Rx] \wedge Sx]$ is first turned into $[[Px \wedge Qx] \wedge [Rx \wedge Sx]]$, and then into $[Px \wedge [Qx \wedge [Rx \wedge Sx]]]$. Clause 7(b) moves existential quantifiers which are inside the first conjunct of a conjunction, outside that conjunction. For example, $\flat[[\exists x Px \wedge \exists y Qy] \wedge Rxy] = \flat[\exists x Px \wedge [\exists y Qy \wedge Rxy]] = \exists x \flat[Px \wedge [\exists y Qy \wedge Rxy]]$ $= \exists x[\flat Px \wedge \flat[\exists y Qy \wedge Rxy]] = \exists x[Px \wedge \exists y[Qy \wedge Rxy]]$. The workings of 7(a) and 7(b) make sure that after repeated application, one will always end up with a conjunction of which the first conjunct is neither a conjunction, nor an existentially quantified formula, i.e., it will be a condition. That is when clause 7(c) applies. Clause 8 defines an analogous procedure for implications. Notice that all clauses leave the length of the formula unchanged. A proof that the recipe will always terminate can easily be given.

Now, we prove the following fact:

Fact 17 For all formulas ϕ: $\phi \simeq \flat\phi$

The proof is by induction on the length of ϕ. For the cases which concern the clauses 1-6, 7(c) and 8(c), the proof is trivial. For the clauses 7(a) and (b), and 8(a) and (b), it suffices to point out the following four *DPL*-equivalences:

$$\begin{array}{rcl} [\phi \wedge \psi] \wedge \chi & \simeq & \phi \wedge [\psi \wedge \chi] \\ \exists x \phi \wedge \psi & \simeq & \exists x[\phi \wedge \psi] \\ [\phi \wedge \psi] \rightarrow \chi & \simeq & \phi \rightarrow [\psi \rightarrow \chi] \\ \exists x \phi \rightarrow \psi & \simeq & \forall x[\phi \rightarrow \psi] \end{array}$$

Next, we show that when a formula is brought in normal binding form, all variables bound by a quantifier are in its scope:

Fact 18 $\mathbf{bp}(\flat\phi) = \mathbf{sp}(\flat\phi)$

The proof is by induction on the length of ϕ.

Clauses 1–6 are simple. For 7(a) we only need to remark that $\mathbf{bp}([\chi_1 \wedge \chi_2] \wedge \psi_2) = \mathbf{bp}(\chi_1 \wedge [\chi_2 \wedge \psi_2])$, as can be seen from definition 13. Similar observations can be made for 7(b), 8(a) and 8(b).

The crucial clauses are 7(c) and 8(c). Consider case 7(c); i.e. $\phi = \psi_1 \wedge \psi_2$ and $\flat\phi = \flat\psi_1 \wedge \flat\psi_2$. We have $\mathbf{bp}(\flat\psi_1 \wedge \flat\psi_2) = \mathbf{bp}(\flat\psi_1) \cup \mathbf{bp}(\flat\psi_2) \cup \{\langle \exists x, x\rangle \mid \exists x \in \mathbf{aq}(\flat\psi_1)\ \&\ x \in \mathbf{fv}(\flat\psi_2)\}$. Since in this case $\flat\psi_1$ can not be a conjunction or an existentially quantified formula, it holds that $\mathbf{aq}(\flat\psi_1) = \emptyset$. This means that in this case $\mathbf{bp}(\flat\psi_1 \wedge \flat\psi_2) = \mathbf{bp}(\flat\psi_1) \cup \mathbf{bp}(\flat\psi_2)$. By induction, $\mathbf{bp}(\flat\psi_1) = \mathbf{sp}(\flat\psi_1)$ and $\mathbf{bp}(\flat\psi_2) = \mathbf{sp}(\flat\psi_2)$. Hence, $\mathbf{bp}(\flat\psi_1 \wedge \flat\psi_2) = \mathbf{sp}(\flat\psi_1) \cup \mathbf{sp}(\flat\psi_2)$. And according to the definition of $\mathbf{sp}$, the latter is the same as $\mathbf{sp}(\flat\psi_1 \wedge \flat\psi_2)$. For 8(c) a similar reasoning can be given.

Now we show that for any formula in which all variables which are bound by a quantifier are inside its scope, it holds that its satisfaction set and its *PL*-interpretation coincide:

Fact 19 If $\mathbf{bp}(\phi) = \mathbf{sp}(\phi)$, then $\forall M\colon \backslash\phi\backslash_M = [\![\phi]\!]_M^{PL}$

The proof proceeds by induction on the length of ϕ.

Obviously, it holds for atomic formulas. And for all but the internally dynamic connectives $\rightarrow$ and $\wedge$, the result follows by a straightforward induction. For example: let $\phi = \exists x\psi$, and suppose $\mathbf{bp}(\exists x\psi) = \mathbf{sp}(\exists x\psi)$. This is the case iff $\mathbf{bp}(\psi) = \mathbf{sp}(\psi)$. Now,

$$\backslash\exists x\psi\backslash = \{g \mid \exists k\colon k[x]g \ \&\ \exists h\colon \langle k, h\rangle \in [\![\psi]\!]\} = \{g \mid \exists k\colon k[x]g \ \&\ k \in \backslash\psi\backslash\}$$

By induction the latter is the same as:

$$\{g \mid \exists k\colon k[x]g \ \&\ k \in [\![\psi]\!]^{PL}\}$$

which in turn equals:

$$[\![\exists x\psi]\!]^{PL}$$

The case of $\rightarrow$ is slightly more complex. Let $\phi = \psi \rightarrow \chi$. Suppose $\mathbf{bp}(\psi \rightarrow \chi) = \mathbf{sp}(\psi \rightarrow \chi)$. In other words, $\mathbf{bp}(\psi) = \mathbf{sp}(\psi)$, $\mathbf{bp}(\chi) = \mathbf{sp}(\chi)$, and if $\exists x \in \mathbf{aq}(\psi)$, then $x \notin \mathbf{fv}(\chi)$. We also know that:

$$\backslash\psi \rightarrow \chi\backslash = \{g \mid \forall h\colon \langle g, h\rangle \in [\![\psi]\!] \rightarrow h \in \backslash\chi\backslash\}$$

It follows by facts 8 and 9 from section 3.3 that this is equal to:

$$\{g \mid \forall h\colon \langle g, h\rangle \in [\![\psi]\!] \rightarrow g \in \backslash\chi\backslash\}$$

For, by fact 9, g and h differ only in variables which have a corresponding active occurrence in ψ. By our assumption that $\mathbf{bp}(\psi \rightarrow \chi) = \mathbf{sp}(\psi \rightarrow \chi)$, these variables do not occur freely in χ, whence it follows by fact 8 that if $h \in \backslash\chi\backslash$, then $g \in \backslash\chi\backslash$.

The above, in its turn, is equal to:

$$\{g \mid g \in \backslash\psi\backslash \rightarrow g \in \backslash\chi\backslash\}$$

Applying induction, we see that this is the same as:

$$\{g \mid g \in [\![\psi]\!]^{PL} \Rightarrow g \in [\![\chi]\!]^{PL}\}$$

which equals:

$$[\![\psi \rightarrow \chi]\!]^{PL}$$

We end the proof by noting that for the remaining case of $\phi = \psi \wedge \chi$, the proof proceeds in a similar fashion.

From facts 18 and 19 it now follows that:

Fact 20 $\forall M\colon \backslash\flat\phi\backslash = [\![\flat\phi]\!]^{PL}$

And putting the latter fact together with fact 17 we get:

Fact 21 For any formula ϕ there is a formula ϕ' such that $\forall M\colon [\![\phi]\!]_M = [\![\phi']\!]_M$, and $\backslash\phi'\backslash_M = [\![\phi']\!]^{PL}_M$

Moreover, we also have that:

Fact 22 For any formula ϕ there is a formula ϕ' such that $\forall M\colon [\![\phi]\!]^{PL}_M = [\![\phi']\!]^{PL}_M$, and $[\![\phi']\!]^{PL}_M = \backslash\phi'\backslash_M$

It is easy to see that this holds. In *PL* any formula ϕ is equivalent to a formula $\sharp\phi$ in which all occurrrences of $\wedge$, $\rightarrow$ and $\exists x$ are eliminated in favour of $\vee$, $\neg$ and $\forall$. Clearly, $\mathbf{bp}(\sharp\phi) = \mathbf{sp}(\sharp\phi)$. Hence, by fact 19, $\backslash\sharp\phi\backslash_M = [\![\sharp\phi]\!]^{PL}_M$, for all M.

Finally, we note the following:

Fact 23 If $\mathbf{bp}(\phi) = \mathbf{sp}(\phi)$, then $\models_{PL} \phi$ iff $\models_{DPL} \phi$

This follows directly from fact 19.

Summing up:

Fact 24 For any formula ϕ:

1. there is a formula ϕ' such that $\models_{DPL} \phi$ iff $\models_{DPL} \phi'$ iff $\models_{PL} \phi'$
2. there is a formula ϕ'' such that $\models_{PL} \phi$ iff $\models_{PL} \phi''$ iff $\models_{DPL} \phi''$

4.2 *DPL* and *DRT*

In this section we compare the language of *DPL* with what corresponds to it in Kamp's *DRT*, viz., the language in which the discourse representation structures (*DRS's*) are formulated. *DPL* is intended to be 'empiricallly equivalent' to *DRT*: it was designed to deal with roughly the same range of natural language facts. The difference between the two approaches is primarily of a methodological nature and compositionality is the watershed between the two. Therefore, besides giving a formal comparison of the logical languages of the two systems in the present section, we shall also discuss the matter of compositionality in somewhat more detail in section 5.2.

The *DRS*-language and the language of *DPL* differ in several respects. First of all, in the *DRS*-language, a syntactic distinction is made between *conditions* and *DRS's*. It is by means of the latter, that natural language sentences and discourses are represented; conditions are elements out of which *DRS's* are constructed. In other words, conditions occur as subexpressions of *DRS's*. Corresponding to this syntactic distinction, there is a semantic one. Conditions are interpreted in terms of their truth conditions, *DRS's* are interpreted in terms of their *verifying embeddings*.

A second difference between the *DRS*-language and the language of *DPL* is that the former contains negation, implication and disjunction, but not conjunction and no quantifiers. The basis of the *DRS*-language is formed by a set of atomic conditions. Further, there is a single, non-iterative rule which has *DRS's* as output: *DRS's* are formed by prefixing a number of variables to a number of conditions. This rule is to compensate *DRT's* lack of conjunction and quantifiers. The prefixed variables function as *DRT's* quantification mechanism, and the conditions to which they are prefixed can be viewed as the conjunction of those conditions. These conditions can be either atomic or complex. Complex conditions are in turn built from *DRS's* by means of the connectives. Negation turns a *DRS* into a condition, implication and disjunction take two *DRS's* and deliver a condition.

Choosing a format that resembles as closely as possible that of *DPL*, the syntax and semantics of *DRT* can be defined as follows. The non-logical vocabulary of consists of: n-place predicates, individual constants, and variables. Logical constants are negation $\neg$, disjunction $\vee$, implication $\rightarrow$, and identity $=$. The syntactic rules are as follows:

Definition 25 (*DRT*-syntax)

1. If $t_1 \ldots t_n$are individual constants or variables, R is an n-place predicate, then $Rt_1 \ldots t_n$ is a condition
2. If t_1 and t_2 are individual constants or variables, then $t_1 = t_2$ is a condition
3. If ϕ is a *DRS*, then $\neg\phi$ is a condition
4. If ϕ and ψ are *DRS's*, then $[\phi \vee \psi]$ is a condition
5. If ϕ and ψ are *DRS's*, then $[\phi \rightarrow \psi]$ is a condition
6. If $\phi_1 \ldots \phi_n$ $(n \geq 0)$ are conditions, and $x_1 \ldots x_k$ are variables $(k \geq 0)$, then $[x_1 \ldots x_k][\phi_1 \ldots \phi_n]$ is a *DRS*
7. Nothing is a condition or a *DRS* except on the basis of 1–6

Models for the *DRS*-language are the same as those for *DPL*, as are assignments and the interpretation of terms. Parallel to the syntactic distinction between conditions and *DRS's*, the semantics defines two notions of interpretation. First of all, we define an interpretation function $[\![\ ]\!]_M^{DRS} \subseteq G \times G$, for *DRS's*. Here, '$\langle g, h\rangle \in [\![\phi]\!]^{DRS}$' corresponds to the *DRT*-notion 'h is a verifying embedding of ϕ with respect to g'. Since *DRS's* are built up from conditions, we also need to define a notion of interpretation of conditions: $[\![\ ]\!]_M^{Cond} \subseteq G$, where '$g \in [\![\phi]\!]^{Cond}$' corresponds to the *DRT*-notion 'ϕ is true with respect to g'. So, *DRS's* receive the same type of interpretation as *DPL*-formulas. In one respect our definition of these notions differs from the one given in *DRT*: we prefer assignments to be total functions rather than partial ones. This is no matter of

principle. Just as is usually done in *DRT*, we could rephrase the semantics of *DPL* in terms of partial assignments.

The simultaneous recursive definition of the notions $[\![\ ]\!]_M^{Cond}$ and $[\![\ ]\!]_M^{DRS}$ runs as follows (where we drop subscripts again, whenever this does not lead to confusion):

Definition 26 (*DRT*-semantics)

1. $[\![Rt_1 \ldots t_n]\!]^{Cond} = \{g \mid \langle [\![t_1]\!]_g \ldots [\![t_n]\!]_g \rangle \in F(R)\}$
2. $[\![t_1 = t_2]\!]^{Cond} = \{g \mid [\![t_1]\!]_g = [\![t_2]\!]_g\}$
3. $[\![\neg\phi]\!]^{Cond} = \{g \mid \neg\exists h\colon \langle g, h\rangle \in [\![\phi]\!]^{DRS}\}$
4. $[\![\phi \vee \psi]\!]^{Cond} = \{g \mid \exists h\colon \langle g, h\rangle \in [\![\phi]\!]^{DRS} \vee \langle g, h\rangle \in [\![\psi]\!]^{DRS}\}$
5. $[\![\phi \rightarrow \psi]\!]^{Cond} = \{g \mid \forall h\colon \langle g, h\rangle \in [\![\phi]\!]^{DRS} \Rightarrow \exists k\colon \langle h, k\rangle \in [\![\psi]\!]^{DRS}\}$
6. $[\![[x_1 \ldots x_k][\phi_1 \ldots \phi_n]]\!]^{DRS} =$
   $\{\langle g, h\rangle \mid h[x_1 \ldots x_k]g \ \&\ h \in [\![\phi_1]\!]^{Cond} \ \&\ \ldots \ \&\ h \in [\![\phi_n]\!]^{Cond}\}$

In order to make clear in what sense the set of variables introduced in clause 6 functions as *DRT's* quantification mechanism, we first define the notion of a *DRS* being true with respect to an assignment in a model:

Definition 27 (Truth in *DRT*) A *DRS* ϕ is *true with respect to g in M* iff $\exists h\colon \langle g, h\rangle \in [\![\phi]\!]_M^{DRS}$

So, the notion of truth for *DRS's* is the same as the notion of truth in *DPL*. And from definition 27 we see that the variable set in a *DRS* behaves like existential quantification over these variables. A simple *DRS* like $[x][Px, Qx]$ has the same truth conditions as the formula $\exists x[Px \wedge Qx]$ in *PL* and *DPL*. Moreover, the interpretations of this *DRS* and of the *DPL*-formula are also the same. To give another example, the *DRS* $[x, y][Px, Qx, Rxy]$ has the same meaning as the *DPL*-formula $\exists x \exists y[Px \wedge Qy \wedge Rxy]$.

Notice that *DRS's* can also be built from conditions by means of empty *DRS*-quantification. For example, $[\ ][Px]$ is a *DRS*, and its interpretation according to definition 26, is $\{\langle g, h\rangle \mid h[\ ]g \ \&\ h(x) \in F(P)\}$. Now, $h[\ ]g$ means the same as $h = g$, so the 'atomic *DRS*' $[\ ][Px]$ and the atomic *DPL*-formula Px have the same interpretation. In fact, this procedure can be applied to turn any *DRT*-condition into a *DRS*, giving it structurally the same interpretation as the corresponding *DPL*-conditions.

The interpretation of a *DRS*, being the same kind of object as the interpretation of formulas in *DPL*, is of a dynamic nature. The dynamics of *DRS's* is put to use in the interpretation of implications (and nowhere else, by the way). For example, the *DRT*-condition $[x][Px] \rightarrow [\ ][Qx]$ has the same truth conditions as the *DPL*-formula $\exists xPx \rightarrow Qx$. This, of course, is the key to *DRT's* successful treatment of donkey-sentences.

Having made these observations, we now turn to the definition of a translation from the *DRS*-language into that of *DPL*. We translate both *DRS's* and *DRT*-conditions into *DPL*-formulas. Blurring the syntactic and semantic distinction between *DRS's* and conditions in this way is justified, since *DRT*-conditions will translate into *DPL*-conditions, and the latter are tests, i.e., their meaning and truth conditions in *DPL* are one-to-one related. The translation $\dagger\phi$ of a *DRS* or a condition ϕ is defined as follows:

Definition 28 (*DRT*-to-*DPL* translation)

1. $\dagger Rt_1 \ldots t_n = Rt_1 \ldots t_n$

2. $\dagger(t_1 = t_n) = (t_1 = t_n)$

3. $\dagger\neg\psi = \neg \dagger \psi$

4. $\dagger[\psi_1 \vee \psi_2] = [\dagger\psi_1 \vee \dagger\psi_2]$

5. $\dagger[\psi_1 \rightarrow \psi_2] = [\dagger\psi_1 \rightarrow \dagger\psi_2]$

6. $\dagger[x_1 \ldots x_k][\psi_1 \ldots \psi_n] = \exists x_1 \ldots \exists x_n[\dagger\psi_1 \wedge \ldots \wedge \dagger\psi_n]$

We prove that our translation is meaning-preserving in the following sense:

Fact 25

1. If ϕ is a condition, then $\forall M\colon [\![\phi]\!]_M^{Cond} = \backslash \dagger\, \phi \backslash_M$
2. If ϕ is a *DRS*, then $\forall M\colon [\![\phi]\!]_M^{DRS} = [\![\dagger\phi]\!]_M$

This fact is proven by induction on the complexity of ϕ. For the clauses 1–5 of definition 25, which build *DRT*-conditions, the proof is trivial. So, what remains to be shown is that:

$$[\![[x_1 \ldots x_k][\psi_1 \ldots \psi_n]]\!]^{DRS} = [\![\exists x_1 \ldots \exists x_n[\dagger\psi_1 \wedge \ldots \wedge \dagger\psi_n]]\!]$$

By definition it holds that:

$$[\![[x_1 \ldots x_k][\psi_1 \ldots \psi_n]]\!]^{DRS} =$$
$$\{\langle g, h\rangle \mid h[x_1 \ldots x_k]g \;\&\; h \in [\![\psi_1]\!]^{Cond} \;\&\; \ldots \;\&\; h \in [\![\psi_n]\!]^{Cond}\}$$

By induction: $[\![\psi_i]\!]^{Cond} = \backslash \dagger\, \psi_i \backslash$, for $1 \leq i \leq n$. So, we may continue our equation as follows:

$$= \{\langle g, h\rangle \mid h[x_1 \ldots x_k]g \;\&\; h \in \backslash \dagger\, \psi_1 \backslash \;\&\; \ldots \;\&\; h \in \backslash \dagger\, \psi_n \backslash\}$$

Next, we note two auxiliary facts:

Fact 26 If ϕ and ψ are *DRT*-conditions, then $\dagger\phi \wedge \dagger\psi$ is a condition in *DPL* as well

From definition 28 it is easy to see that if ϕ and ψ are *DRT*-conditions, then $\dagger\phi$ and $\dagger\psi$ are conditions in *DPL* as well. By definition 12 it then follows that $\dagger\phi \wedge \dagger\psi$ is also a *DPL*-condition.

Fact 27 If ϕ and ψ are *DPL*-conditions, then $\backslash\phi \wedge \psi\backslash = \backslash\phi\backslash \cap \backslash\psi\backslash$

This can be proven by a simple calculation.

Now we return to our proof of fact 25. On the basis of our auxiliary facts, we arrive at the following continuation of our equation:

$$= \{\langle g, h\rangle \mid h[x_1 \ldots x_k]g \;\&\; h \in \backslash \dagger\, \psi_1 \wedge \ldots \wedge \dagger\psi_n \backslash\}$$

Since $\dagger\psi_1 \wedge \ldots \wedge \dagger\psi_n$ is a *DPL*-condition, it holds that $h \in \backslash \dagger\, \psi_1 \wedge \ldots \wedge \dagger\psi_n \backslash$ iff $\langle h, h\rangle \in [\![\dagger\psi_1 \wedge \ldots \wedge \dagger\psi_n]\!]$. This implies that we can continue as follows:

$$= \{\langle g, h\rangle \mid \exists k k[x_1 \ldots x_k]g \;\&\; \langle k, h\rangle \in [\![\dagger\psi_1 \wedge \ldots \wedge \dagger\psi_n]\!]\}$$
$$= [\![\exists x_1 \ldots \exists x_k[\dagger\psi_1 \wedge \ldots \wedge \dagger\psi_n]]\!]$$

By which the proof of fact 25 is completed.

Before turning to the less urgent, but more difficult problem of translating *DPL*-formulas into *DRS's*, we return, by way of a short intermezzo, to two of the examples discussed in section 2.1, and compare, once again, the corresponding *DPL*-formulas and *DRS's*. By $\dagger$*DRT* we mean the translation of the *DRT*-representation in *DPL*.

(1) A man walks in the park. He whistles.

(1a) $\exists x[\texttt{man}(x) \wedge \texttt{walk_in_the_park}(x) \wedge \texttt{whistle}(x)]$ *PL*/†*DRT*

(1b) $\exists x[\texttt{man}(x) \wedge \texttt{walk_in_the_park}(x)] \wedge \texttt{whistle}(x)$ *DPL*

(1c) $[x][\texttt{man}(x), \texttt{walk_in_the_park}(x), \texttt{whistle}(x)]$ *DRT*

(3) Every farmer who owns a donkey beats it

(3a) $\forall x \forall y[[\texttt{farmer}(x) \wedge \texttt{donkey}(y) \wedge \texttt{own}(x,y)] \rightarrow \texttt{beat}(x,y)]$ *PL*

(3b) $\forall x[[\texttt{farmer}(x) \wedge \exists y[\texttt{donkey}(y) \wedge \texttt{own}(x,y)]] \rightarrow \texttt{beat}(x,y)]$ *DPL*

(3c) $[\ ][[x,y][\texttt{farmer}(x), \texttt{donkey}(y), \texttt{own}(x,y)] \rightarrow [\ ][\texttt{beat}(x,y)]]$ *DRT*

(3d) $\exists x \exists y[\texttt{farmer}(x) \wedge \texttt{donkey}(y) \wedge \texttt{own}(x,y)] \rightarrow \texttt{beat}(x,y)$ †*DRT*

Consider the first example. If our diagnosis of the problem that such a sequence poses, viz., that the real problem is to provide a compositional translation of such sequences of sentences into a logical representation language, is correct, then the *DRT*-representation has as little to offer as the *PL*-translation. The two component sentences cannot be retrieved from (1c), neither can they be isolated from (1a) as subformulas. The *DPL*-representation differs precisely at this point.

As for the second example, it is not possible to retrieve the component parts of the sentence from either the *PL*- or the *DRT*-representation, nor from the †*DRT*-translation. But in the *DPL*-formula, we do find an open formula which corresponds to the complex noun *farmer who owns a donkey*, viz., $\texttt{farmer}(x) \wedge \exists y[\texttt{donkey}(y) \wedge \texttt{own}(x,y)]$. But no such sub-expression is to be found in the *DRT*-representation or in the corresponding †*DRT*-formula.

In fact, each of the two examples can be used to make a point in favour of *DPL*. The preferable *DPL*-translation of the first example is available precisely because *DPL* has dynamic conjunction, whereas such a concept is lacking in *DRT*. As for the second example, here *DPL* fares better because, unlike *DRT*-quantification, *DPL*-quantification is iterative. It are precisely these two concepts of *DPL* which enable us to give a compositional treatment of the cases at hand. In fact, if both conjunction of *DRS's* and iterative quantification were added to *DRT*, it would simply collapse into *DPL*.

This is a rather surprising result, since one of the trademarks of theories such as those of Kamp and Heim is the *non-quantificational* analysis of indefinite terms, whereas it is characteristic of *DPL* that it does allow us to treat such terms as existentially quantified expressions.

A host of arguments have been presented against a quantificational analysis of indefinites, see in particular in Heim [1982]. However, these arguments now appear to be directed, not against a quantificational analysis as such, but only

against the traditional quantificational analysis, which is *static*. The dynamic *DPL*-approach is not affected by these.

We do not discuss the various arguments here, with one exception. This argument concerns what Heim calls the 'chameleontic' nature of the quantificational force of indefinites. What follows is in essence, but not in all details, taken from Heim [1982].

Consider the following variants of our example (2):

(9) If a farmer owns a donkey, he always/usually/sometimes/never beats it

These variants of the donkey-example (2) have readings which can be paraphrased as in (10):

(10) In all/most/some/no cases in which a farmer owns a donkey, he beats it

Examples such as those in (9) are taken to support the view that what appears to be the quantificational force of an indefinite term, is in fact either due to a different expression, a so-called 'adverb of quantification', as in (9), or is implicit in the construction, as for example in the original donkey-sentence (2). Following Lewis [1975], it is assumed that the sentences in (9) are to be analyzed along the following lines. The main operator is the adverb of quantification, which takes two arguments, the antecedent and the consequent. The indefinite terms are treated as free variables, which are unselectively bound by the main operator, which determines the quantificational force. The antecedent serves as a restriction on the unselective quantification. For the original donkey-sentence, which lacks an overt adverb of quantification, it is assumed that the construction itself acts as a universal adverb of quantification.

At first sight, this argument seems not to be restricted to the traditional, static quantificational analysis, but seems to apply to any quantificational approach which associates a specific quantificational force with indefinite terms. However, in view of our observations concerning the relationship between *DPL* and *DRT*, this can not really be correct. In fact, it is not difficult to incorporate the 'adverbs-of-quantification' analysis of sentences such as (9) in *DPL*, thus showing its compatibility with a quantificational treatment of indefinites.

Recall that in the interpretation clause of implication, there is universal quantification over the output assignments of the antecedent:

$$[\![\phi \rightarrow \psi]\!] = \{\langle g, h\rangle \mid h = g \ \& \ \ \textit{for all } k\colon \langle h, k\rangle \in [\![\phi]\!], \exists j\colon \langle k, j\rangle \in [\![\psi]\!]\}$$

What we can do is simply generalize this to other quantificational forces, and index the implication accordingly:

$$[\![\phi \rightarrow_Q \psi]\!] = \{\langle g, h\rangle \mid h = g \ \& \ \ \textit{for } Q\, k\colon \langle h, k\rangle \in [\![\phi]\!], \exists j\colon \langle k, j\rangle \in [\![\psi]\!]\}$$

This has the required effects. Consider the following example:

(11) If a farmer owns a donkey, he never beats it

If we translate this using $\rightarrow_{no}$, the result is:

$$\exists x[\texttt{farmer}(x) \wedge \exists y[\texttt{donkey}(y) \wedge \texttt{own}(x,y)]] \rightarrow_{no} \texttt{beat}(x,y)$$

This denotes the following set of pairs of assignments:

$$\{\langle g,g\rangle \mid \neg\exists h\colon h[x,y]g \ \&\ h(x) \in F(\texttt{farmer}) \ \&\ h(y) \in F(\texttt{donkey}) \ \& \\ \langle h(x),h(y)\rangle \in F(\texttt{own}) \ \&\ \langle h(x),h(y)\rangle \in F(\texttt{beat})\}$$

Notice that this analysis works precisely because indefinite terms are analyzed as dynamically existentially quantified expressions. For this has the effect that the quantification in the general scheme is restricted to the variables which correspond to indefinite terms.

We do not intend this as a final analysis of adverbs of quantification, since, for one thing, such an analysis has to be higher-order and intensional, and *DPL* is only first-order and extensional. But we do take the above to show that an 'adverbs-of-quantification' analysis is perfectly compatible with a quantificational analysis of indefinite terms, provided this is a dynamic one.

After this intermezzo, we return to the formal comparison of *DPL* and *DRT*. As we already remarked above, the formulation of a translation from the *DPL*-language to the *DRS*-language, is more difficult than the other way around. In fact, no strict interpretation-preserving translation is possible, though one which preserves truth conditions is. We point out the main features of such a translation, written as '§', without, however, going into details.

Notice that the fact that *DRT* distinguishes between conditions and *DRS's*, presents no problem. Defining a translation from *DPL*-formulas into *DRS's* is sufficient, for as we saw above, any *DRT*-condition can easily be turned into a *DRS* by means of empty *DRT*-quantification.

Now, there are three complications. The first concerns universal quantification, which is lacking in *DRT*. We can either use the definition of $\forall x\phi$ in terms of $\neg\exists x\neg\phi$, or turn $\S\forall x\phi$ directly into the condition $[x][\ ] \rightarrow \S\phi$.

The remaining two complications, not surprisingly, stem from the two essential differences between *DRT* and *DPL* which we noticed above. Because *DRT* lacks a notion of *DRS*-conjunction, we cannot compositionally translate a *DPL*-conjunction. Something like $\S[\phi \wedge \psi] = [\ ][\S\phi, \S\psi]$ would work only if both conjuncts are *DPL*-conditions, which, of course, they need not be.

Similarly, no compositional translation of existentially quantified formulas is possible either. Again, $\S\exists x\phi = [x]\S\phi$ works only if ϕ is a *DPL*-condition. Suppose that ϕ in its turn is $\exists yRxy$. Then the resulting translation would be $[x][y]Rxy$. But this is not a well-formed *DRS*, since $[y]Rxy$ is not a *DRT*-condition.

To get things to work, we first need to define a special format for *DPL*-formulas which enables us to translate them in a non-compositional, global manner into *DRS's*. In order to arrive at the required format, any *DPL*-formula

ϕ should be turned into a formula ϕ', such that any subformula of ϕ' is of the form $\exists x_1 \ldots \exists x_n \psi (n \geq 0)$, where ψ is a *DPL*-condition.

It is possible to give an algorithm that has the required effect, but it is not strictly meaning-preserving. The following two examples may serve to illustrate this. Consider the formula $Px \wedge \exists x Qx$. In order to give it the right format, the existential quantifier in the second conjunct has to be moved outside of the conjunction. But this can't be done, since there is a free occurrence of x in the first conjunct. So, we have to resort to an alphabetic variant: $\exists y[Px \wedge Qy]$. As a second example, consider $\exists x Px \wedge \exists x Qx$. In this case, too, both quantifiers have to be moved outside the conjunction, and then again, we need an alphabetic variant: $\exists x \exists y[Px \wedge Qy]$. The use of alphabetic variants implies that the algorithm is not meaning preserving, for in *DPL* such variants have different meanings: $\exists x \phi \not\simeq \exists y[y/x]\phi$.

These features of *DPL*-to-*DRT*-translation illustrate once more, we think, what exactly makes *DPL* a more suitable instrument for semantic analysis. Its dynamic notion of conjunction, and its dynamic and iterative concept of existential quantification allow us to deal with various phenomena in a simple, intuitive and compositional manner.

4.3 *DPL* and quantificational dynamic logic

Although there are important resemblances between *DPL* and certain systems of dynamic logic as they are discussed in Harel [1984], there are also major differences. To begin with, there is an important difference in perspective and over-all aims. Dynamic logic is meant to be used in the formalization of reasoning about computer programs, about the effects of their execution, their soundness and correctness, and so on. Typically, the formulas of a system of dynamic logic are interpreted as assertions about programs. And the semantic interpretation of these formulas is an ordinary static interpretation. However, in order to be able to talk about programs in a logical language, one also needs expressions that refer to these programs. And that is where dynamic interpretation comes in. The expressions which are the logical stand-in's for programs, do receive a dynamic interpretation. However, they only appear as sub-expressions in the formulas that make up the logical system, they are not themselves formulas of the logic.

We note in passing, that in this respect we find precisely the opposite situation in *DRT*. There as well, we have a distinction between statically interpreted expressions, *DRT's* conditions, and dynamically interpreted ones, the *DRS's*. But in *DRT* it are the dynamic expressions that play first fiddle, and the static conditions only occur as subexpressions in these.

In *DPL*, it are the formulas themselves that receive a dynamic interpretation, the kind of interpretation that programs receive in ordinary dynamic logic. *DPL* is a system in which certain kinds of 'programs' can be expressed, but we cannot formulate assertions about these programs in *DPL* itself. Since it is a logical

language which is designed to represent meanings of natural language sentences, one could say that it embodies the view that the meaning of a sentence is a program, an instruction to the interpreter. So, one could view *DPL* as a kind of 'programming language', rather than as a language to reason about such programs. Of course, one can reason about it in a metalanguage. And, in fact, one could use ordinary dynamic logic as a means to formalize reasoning about *DPL*.

Of course, this difference in what the systems are meant to be able to do, is reflected in various aspects of their organization. We illustrate this by presenting the syntax and semantics of a particular system of quantificational dynamic logic, referred to as '*QDL*', which proves to be intimately related to *DPL*.

Dynamic logic is related to modal logic. The models contain a set S of possible (execution) states. The formulas are interpreted as sets of states, the set of states in which a formula is true. Programs are conceived of as transformations of possible states, i.e., as relations between possible states. In this respect, the interpretation of a program is like an accessibility relation in modal logic. Each program π corresponds to an accessibility relation $[\![\pi]\!]_M \subseteq S \times S$. A pair $\langle s, s' \rangle$ is an element of $[\![\pi]\!]_M$ if when executed in s, π may lead to s'. (If the program is deterministic, $[\![\pi]\!]_M$ would be a (partial) function.)

In view of their association with accessibility relations, we can build modal operators $\langle \pi \rangle$ and $[\pi]$ around a program π. Like their counterparts in modal logic, these operators can be prefixed to a formula ϕ to form another formula, $\langle \pi \rangle \phi$ or $[\pi]\phi$. A formula $\langle \pi \rangle \phi$ is true in a state s iff execution of π in s may lead to a state s' such that ϕ is true in s'. In other words, $\langle \pi \rangle \phi$ expresses that it is possible that ϕ is true after π has been executed. Similarly, $[\pi]\phi$ is true in s iff all executions of π in s will lead to a state s' in which ϕ is true. So, $[\pi]\phi$ means that ϕ must be true after π has been executed. $[\pi]\phi$ is equivalent to $\neg\langle \pi \rangle \neg\phi$, where $\neg$ is interpreted as ordinary static negation.

In systems of quantificational dynamic logic, the set of possible states is not just a set of primitive objects, but is identified with the set of assignment functions G. The interpretation of a formula ϕ is $[\![\phi]\!]_M \subseteq G$. And the interpretation of a program π is $[\![\pi]\!]_M \subseteq G \times G$.

We now present a particular version of *QDL* that has precisely the features we need to compare it with *DPL*. We start out from the language of *PL* and add the following features. Basic programs are random assignments to variables, written as '$x :=$ `random`'. (This is a feature not present in standard *QDL*, where ordinary determistic assignments '$x := a$' figure in the language.) Further, we add an operator '?', which turns a formula ϕ into a program $?\phi$. Such a program is called a 'test', and its interpretation is indeed like that of a test in *DPL*, it is the set of identity pairs $\langle g, g \rangle$ such that ϕ is true with respect to g. Next, we add the operator ';' to form sequences of programs. (Ordinary deterministic assignments can be defined in terms of these notions as: $x :=$ `random`; $?x = a$.) Finally, we add the 'modal operators' discussed above. So, *QDL* has the following syntax:

Definition 29 (Syntax of *QDL*)

1. $\top$ is a formula
2. If $t_1 \dots t_n$ are individual constants or variables, R is an n-place predicate, then $Rt_1 \dots t_n$ is a formula
3. If t_1 and t_2 are individual constants or variables, then $t_1 = t_2$ is a formula
4. If ϕ is a formula, then $\neg\phi$ is a formula
5. If ϕ and ψ are formulas, then $[\phi \rightarrow \psi]$, $[\phi \wedge \psi]$ and $[\phi \vee \psi]$ are formulas
6. If ϕ is a formula, then $\exists x\phi$ is a formula
7. If ϕ is a formula, then $\forall x\phi$ is a formula
8. If ϕ is a formula, then $?\phi$ is a program
9. If x is a variable, then $x :=$ `random` is a program
10. If π_1 and π_2 are programs, then $[\pi_1; \pi_2]$ is a program
11. If π is a program and ϕ is a formula, then $\langle\pi\rangle\phi$ is a formula
12. If π is a program and ϕ is a formula, then $[\pi]\phi$ is a formula
13. Nothing is a formula or a program except on the basis of 1–12

Models for *QDL* are like those for *(D)PL*. Like in *DRT*, we simultaneously define two interpretation functions, one for formulas: $[\![\ ]\!]_M^{QDL} \subseteq G$; and one for programs: $[\![\ ]\!]_M^{Prog} \subseteq G \times G$ as follows (as usual, we suppress subscripts whenever this does not give rise to confusion):

Definition 30 (Semantics of *QDL*)

1. $[\![\top]\!]^{QDL} = G$
2. $[\![Rt_1 \dots t_n]\!]^{QDL} = \{g \mid \langle [\![t_1]\!]_g \dots [\![t_n]\!]_g \rangle \in F(R)\}$
3. $[\![t_1 = t_2]\!]^{QDL} = \{g \mid [\![t_1]\!]_g = [\![t_2]\!]_g\}$
4. $[\![\neg\phi]\!]^{QDL} = \{g \mid g \notin [\![\phi]\!]^{QDL}\}$
5. $[\![\phi \rightarrow \psi]\!]^{QDL} = \{g \mid g \in [\![\phi]\!]^{QDL} \Rightarrow g \in [\![\psi]\!]^{QDL}\}$, and similarly for $\wedge$,$\vee$
6. $[\![\exists x\phi]\!]^{QDL} = \{g \mid \exists k\colon k[x]g \ \&\ k \in [\![\phi]\!]^{QDL}\}$
7. $[\![\forall x\phi]\!]^{QDL} = \{g \mid \forall k\colon k[x]g \Rightarrow k \in [\![\phi]\!]^{QDL}\}$
8. $[\![?\phi]\!]^{Prog} = \{\langle g, h\rangle \mid h = g \ \&\ h \in [\![\phi]\!]^{QDL}\}$

9. $[\![x := \texttt{random}]\!]^{Prog} = \{\langle g, h\rangle \mid h[x]g\}$

10. $[\![\pi_1 \,;\, \pi_2]\!]^{Prog} = \{\langle g, h\rangle \mid \exists k\colon \langle g, k\rangle \in [\![\pi_1]\!]^{Prog} \;\&\; \langle k, h\rangle \in [\![\pi_2]\!]^{Prog}\}$

11. $[\![\langle\pi\rangle\phi]\!]^{QDL} = \{\; g \mid \exists h\colon \langle g, h\rangle \in [\![\pi]\!]^{Prog} \;\&\; h \in [\![\phi]\!]^{QDL}\}$

12. $[\![[\pi]\phi]\!]^{QDL} = \{\; g \mid \forall h\colon \langle g, h\rangle \in [\![\pi]\!]^{Prog} \Rightarrow h \in [\![\phi]\!]^{QDL}\}$

First, we note that the language defined above can be economized rather drastically. Of course, the interdefinability of the connectives and quantifiers in *PL* carries over to *QDL*. But, moreover, as appears from the following two equivalences, we can conclude that all that is characteristic of *PL* can be eliminated altogether:

$$\phi \to \psi \simeq [?\phi]\psi$$

$$\exists x\phi \simeq \langle x := \texttt{random}\rangle\phi$$

So, in terms of negation, the test-operator, random assignments and one of the two modal operators, all other logical constants can be defined.

As we noted above, *DPL*-formulas can be conceived of as a kind of programs. The following definition presents a translation '$\triangleright$' of *DPL*-formulas into *QDL*-programs:

Definition 31 (*DPL*-to-*QDL* translation)

1. $\triangleright Rt_1 \ldots t_n = ?Rt_1 \ldots t_n$

2. $\triangleright\neg\phi = ?\neg\langle\triangleright\phi\rangle\top$

3. $\triangleright[\phi \wedge \psi] = [\triangleright\phi \,;\, \triangleright\psi]$

4. $\triangleright[\phi \vee \psi] = ?[\langle\triangleright\phi\rangle\top \vee \langle\triangleright\psi\rangle\top]$

5. $\triangleright[\phi \to \psi] = ?[\triangleright\phi]\langle\triangleright\psi\rangle\top$

6. $\triangleright\exists x\phi = [x := \texttt{random} \,;\, \triangleright\phi]$

7. $\triangleright\forall x\phi = ?[x := \texttt{random}]\langle\triangleright\phi\rangle\top$

The last but one of these clauses illustrates that in *DPL* we need not introduce the existential quantifier syncategorematically: we could could take $\exists x$ itself to be a formula of *DPL*, with the same interpretation as a random assignment statement, and we could write $\exists x \wedge \phi$ instead of $\exists x\phi$.

The following fact can be proven by induction on the complexity of ϕ:

Fact 28 $\forall M\colon [\![\phi]\!]_M = [\![\triangleright\phi]\!]_M^{Prog}$

Unlike in the case of *DRT*, we can equally easily define a translation in the opposite direction. Like in our translation from *DRT* to *DPL*, we don't pay attention to the distinction between formulas and programs in *QDL*. No problems can arise from this, since *QDL*-formulas will be translated into *DPL*-conditions. We make only one small addition to *DPL*: we add $\top$ as a basic formula, and interpret it as the identity relation on G. We denote the translation function by '$\lhd$':

Definition 32 (*QDL*-to-*DPL* translation)

1. $\lhd\top = \top$
2. $\lhd Rt_1 \ldots t_n = Rt_1 \ldots t_n$
3. $\lhd\neg\phi = \neg \lhd \phi$
4. $\lhd[\phi \rightarrow \psi] = [\lhd\phi \rightarrow \lhd\psi]$, and similarly for $\vee$ and $\wedge$
5. $\lhd\exists x\phi = \Diamond\exists x \lhd \phi$
6. $\lhd\forall x\phi = \forall x \lhd \phi$
7. $\lhd ?\phi = \lhd\phi$
8. $\lhd x := \texttt{random} = \exists x\top$
9. $\lhd[\pi_1 ; \pi_2] = [\lhd\pi_1 \wedge \lhd\pi_1]$
10. $\lhd\langle\pi\rangle\phi = \Diamond[\lhd\pi \wedge \lhd\psi]$
11. $\lhd[\pi]\phi = [\lhd\pi \rightarrow \lhd\psi]$

By simultaneous induction on the complexity of the programs π and formulas ϕ of *QDL* it can be shown that the translation is meaning-preserving in the following sense:

Fact 29

1. $\forall M\colon [\![\phi]\!]^{QDL}_M = \backslash \lhd \phi \backslash_M$
2. $\forall M\colon [\![\phi]\!]^{Prog}_M = [\![\lhd\phi]\!]_M$

The redundancy of *QDL* is reflected illuminatingly in the translation. Most *DPL*-operators can be found twice on the right hand side. In some cases, they occur once in a dynamic and once in a static variant, the latter being obtained by prefixing the closure operator. Notice that since *PL* forms a fragment of *QDL*, the translation presented above is also a translation of *PL* into *DPL*.

We end this section by discussing two standard features of quantificational dynamic logics that we have left out sofar. The one is program disjunction, the other program repetition. We also discuss briefly their possible use in natural language semantics.

The syntax and semantics of program-disjunction are defined as follows:

Definition 33 (Program disjunction)

1. If π_1 and π_2 are programs, then $[\pi_1 \cup \pi_2]$ is a program
2. $[\![\pi_1 \cup \pi_2]\!]^{Prog} = [\![\pi_1]\!]^{Prog} \cup [\![\pi_2]\!]^{Prog}$

Of course, the same definition could be used to introduce a second notion of disjunction $\cup$ besides $\vee$ in *DPL*. Unlike the latter, $\cup$ is a dynamic notion, but it differs in an interesting way from dynamic implication and conjunction. Whereas implication is only internally dynamic, and conjunction is both internally and externally dynamic, $\cup$ is only externally dynamic. An existential quantifier $\exists x$ in the first disjunct cannot bind free occurrences of x in the second disjunct (nor vice versa), but an existential quantifier in either disjunct can bind variables in a further conjunct. In the formula $[\exists x Px \cup \exists x Qx] \wedge Hx$ both occurrences of $\exists x$ in the first conjunct bind the occurrence of x in the second conjunct. In fact the formula in question is equivalent to $[\exists x Px \wedge Hx] \cup [\exists x Qx \wedge Hx]$. More generally the following holds:

$$[\phi \cup \psi] \wedge \chi = [\phi \wedge \chi] \cup [\psi \wedge \chi]$$

$$[\phi \cup \psi] \rightarrow \chi = [\phi \rightarrow \chi] \wedge [\psi \rightarrow \chi]$$

Adding this kind of disjunction to our dynamic repertoire would enable us to treat the anaphoric links in examples like (12) and (13) in a completely straightforward way:

(12) A professor or an assistant professor will attend the meeting of the university board. He will report to the faculty

(13) If a professor or an assistant professor attends the meeting of the university booard, then he reports to the faculty

However, as we shall see in the next section, adding this new kind of dynamic disjunction still leaves certain dynamic feature of natural language disjunction unexplained.

We now turn to program repetition. This concept is important in the semantic analysis of program constructions like '*while* ... *do* ...' and '*repeat* ... *until* ...'. The syntax and semantics of program repetition is defined as follows:

Definition 34 (Program repetition)

1. If π is a program, then $\pi*$ is a program
2. $[\![\pi*]\!]^{Prog} = \{\langle g, h\rangle \mid \exists n \exists g_0, g_1, \ldots, g_n \colon g_0 = g \ \&\ g_n = h \ \&$ $\forall i \colon 1 \leq i \leq n \colon \langle g_{i-1}, g_i\rangle \in [\![\pi]\!]_M^{prog}\}$

According to this definition, a pair $\langle g, h\rangle$ is in the interpretation of $\pi*$ iff h can be reached from g by a repeating π a finite but non-deterministically determined number of times.

At first sight, this kind of concept seems of no use in natural language semantics. However, consider the following example (due to Schubert & Pelletier [1989]):

(14) If I've got a quarter in my pocket, I'll put in the parking meter

Notice first of all, that unlike a *DRT*/*DPL*-analysis would have it, one who utters (14), probably does not intend to spend all the quarters in his pocket on the parking meter. Now, notice that a procedural meaning of (14) could informally be paraphraseds as "Repeat getting coins out of your pocket until it is a quarter; then put it in the parking meter". So maybe after all, adding repetition to *DPL* could add to its use as a tool in natural language semantics.

As one of the referees pointed out, the Schubert & Pelletier analysis can be dealt with in *DPL* in a more straightforward way, too. It would suffice to define another notion of implication as follows:

$$\phi \hookrightarrow \psi =_{def} \neg\phi \vee [\phi \wedge \psi]$$

See also Chierchia [1988], where this conservative notion of implication is argued for.

5 Prospect and retrospect

5.1 Problems and prospects

As we have pointed out in the introduction of this paper, we are interested in developing a compositional, non-representational semantics of discourse, one which will enable us to marry the compositional framework of Montague grammar to a dynamic outlook on meaning such as can be found in *DRT* and its kin. The development of *DPL* is only a first step in achieving this over-all aim. At the empirical level, *DPL* matches *DRT*, and from a methodological point of view, it is in line with *MG*. At least at the following two points, *DPL* needs to be extended. First of all, like *DRT*, *DPL* is restricted to the resources of an extensional first-order system, whereas *MG* essentially makes use of intensional higher order logic. And secondly, *DPL* shares several empirical characteristics with *DRT* which have been disputed in the literature.

As for the first point, *DPL* offers as compositional a treatment of natural language expressions as a first-order system permits; nothing more, nothing less. However, to match *MG*, and more in particular to be able to cope with compositionality below the sentential level in the way familiar from *MG*, we do need more. In fact, we need a higher-order, intensional language with λ-abstraction, or something else that is able to do what that does. In Groenendijk & Stokhof [1990] one way to go about is presented, which uses a version of dynamic intensional logic (*DIL*) as it was developed in Janssen [1986] with the aim of providing a Montague-style semantics for programming languages. The

resulting system of 'dynamic Montague grammar' (*DMG*) is able to cope with the phenomena *DPL* deals with in a completely compositional fashion — below, on, and beyond the sentential level.

The second issue we want to touch upon here concerns certain empirical predictions that *DPL* shares with *DRT*.

As we remarked several times in the above, only conjunction and the existential quantifier are treated in a fully dynamic fashion. They are both internally and externally dynamic. All other logical constants are externally static. In this respect, *DPL* is like *DRT*. In our informal introduction to *DPL* in section 2, we motivated the interpretation clauses for the various logical constants by pointing out that they behave differently with regard to possible anaphoric relations. (Cf. examples (4)–(8).) Thus it was argued, for example, that conjunction and implication should be internally dynamic, because both allow an antecedent in their first argument to bind an anaphor in their second argument. However, it was concluded that only conjunction is also externally dynamic, since it also passes on bindings to sentences to come, whereas implication, in view of the fact that it lacks this feature, should be treated as externally static. The interpretation of the universal quantifier, negation, and disjunction was motivated in a similar fashion.

Several authors have provided examples which seem to indicate that the predictions that *DRT* and *DPL* make here, are not borne out by the facts. (See e.g. Roberts [1987,1989], Kadmon [1987].) Consider the following examples (which are (variants of) examples that can be found in the literature):

(15) If a client turns up, you treat him politely. You offer him a cup of coffee and ask him to wait

(16) Every player chooses a pawn. He puts it on square one.

(17) It is not true that John doesn't own a car. It is red, and it is parked in front of his house

(18) Either there is no bathroom here, or it is in a funny place. In any case, it is not on the first floor

Different conclusions may be drawn from these observations, some of which are compatible with the way in which the logical constants are interpreted in *DRT* and *DPL*. However, one might also take these examples to show that, at least in certain contexts, the universal quantifier, implication, disjunction, and negation are both internally and externally dynamic. Without wanting to commit ourselves to the latter position, we want to explore its consequences a little here.

As for the first of the examples, it can then be observed that the second sentence is interpreted as an additional conjunct of the consequent of the implication in the first sentence, as the following paraphrase of (15) shows:

(19) If a client turns up, you treat him politely, you offer him a cup of coffee, and ask him to wait

So, what we need is an interpretation of implication which will make (20) equivalent to (21):

(20) $[\exists x[\texttt{client}(x) \wedge \texttt{turn_up}(x)] \rightarrow \texttt{treat_politely}(\texttt{y}, x)] \wedge \texttt{offer_coffee}(\texttt{y}, x) \wedge \texttt{ask_to_wait}(\texttt{y}, x)$

(21) $\exists x[\texttt{client}(x) \wedge \texttt{turn_up}(x)] \rightarrow [\texttt{treat_politely}(\texttt{y}, x) \wedge \texttt{offer_coffee}(\texttt{y}, x) \wedge \texttt{ask_to_wait}(\texttt{y}, x)]$

More generally, an externally dynamic interpretation of implication will make $[\phi \rightarrow \psi] \wedge \chi$ equivalent with $\phi \rightarrow [\psi \wedge \chi]$.

As for the second example, similar observations can be made. It can be paraphrased as (22), which indicates that an externally dynamic treatment of universal quantification will make (23) equivalent with (24):

(22) Every player chooses a pawn, and (he) puts it on square one

(23) $\forall x[\texttt{player}(x) \rightarrow \exists y[\texttt{pawn}(y) \wedge \texttt{choose}(x, y)]] \wedge \texttt{put_on_square_one}(x, y)$

(24) $\forall x[\texttt{player}(x) \rightarrow \exists y[\texttt{pawn}(y) \wedge \texttt{choose}(x, y) \wedge \texttt{put_on_square_one}(x, y)]]$

So, on this approach, $\forall x \phi \wedge \psi$ turns out to be equivalent with $\forall x[\phi \wedge \psi]$. And if this is combined with a dynamic interpretation of implication, $\forall x[\phi \rightarrow \psi] \wedge \chi$ will be equivalent with $\forall x[\phi \rightarrow [\psi \wedge \chi]]$.

In a similar fashion, the third example may be taken to indicate that a dynamic version of negation is needed for which the law of double negation holds.

The last example indicates that disjunction, too, can sometimes be interpreted dynamically. This interpretation should make $[\phi \vee \psi] \wedge \chi$ equivalent with $\phi \vee [\psi \wedge \chi]$. Notice that the dynamic interpretation of disjunction that is at stake here, differs from the one discussed in section 4.3. The latter, as we have seen, is essentially internally static, and only externally dynamic, whereas the present notion is both internally and externally dynamic. Also, their external dynamic behaviour is different: $[\phi \cup \psi] \wedge \chi$ is equivalent with $[\phi \wedge \chi] \cup [\psi \wedge \chi]$.

These observations characterize one way of dealing with such as examples as (15)–(18). We end our discussion of them with three remarks. First of all, saying what the desired effect of the dynamic interpretations of the logical constants involved are, is, of course, not the same as actually giving the interpretations themselves. And secondly, the availability of suitable dynamic interpretations would leave unanswered the question why it is that the logical constants involved act dynamically in certain contexts, but not in others. Finally, in view of the latter fact, one would not want to postulate two independent interpretations. Rather, the static interpretation should be available from the dynamic one by a general operation of closure.

The first and the last issue are discussed at length in Groenendijk & Stokhof [1990]. There it is shown that using the richer framework of *DMG*, the required dynamic interpretations can indeed be obtained, and in such a fashion that the static interpretations are the closures of the dynamic ones. As for the second point, this seems to be more an empirical than a formal question, to which *DMG* as such does not provide an answer.

From this, we conclude that the kind of dynamic approach to natural language meaning that is advocated in this paper, is not restricted to the particular form it has taken here, i.e., that of the *DPL*-system, but is sufficiently rich to allow for alternative analyses and extensions (see, e.g., Chierchia [1988], Dekker [1990], van den Berg [1990]).

5.2 Meaning and compositionality

The primary motivation for the *DPL*-undertaking was that we were interested in the development of a compositional and non-representational theory of meaning for discourses. Compositionality is the corner-stone of all semantic theories in the logical tradition. As a consequence, it has also been of prime importance in those approaches to natural language semantics which use tools developed in the logical paradigm. However, compositionality has been challenged as a property of natural language semantics. Especially when dealing with the meaning of discourses people have felt, and sometimes argued, that a compositional approach fails.

In the context of natural language semantics, we interpret compositionality as primarily a methodological principle, which gets empirical, computational, or philosophical import only when additional, and independently motivated constraints are put on the syntactic or the semantic part of the grammar that one uses. In other words, it being a methodological starting point it is always possible to satisfy compositionality by simply adjusting the syntactic and/or semantic tools one uses, unless that is, the latter are constrained on independent grounds. In view of this interpretation of compositionality, our interest in the possibility of a compositional semantics of discourse is also primarily of a methodological nature. Faced with non-compositional theories that give an account of interesting phenomena in the semantics of natural language discourses, we wanted to investigate the properties of a theory that is compositional and accounts for the same facts. We knew in advance that such a theory should exist, what we wanted to know is what it would look like: it might have been that being compositional was the only thing that speaks in favour of such a theory, in which case there would have been good reasons to abandon it.

As we already remarked in the introduction, beside these methodological considerations, there may also be practical reasons to be interested in trying to keep to compositionality. One such reason can be found in computational requirements on the semantics of discourses, or texts. For example, in a translation program one would like to be able to interpret a text in an on-line manner,

i.e., incrementally, processing and interpreting each basic unit as it comes along, in the context created by the interpretation of the text sofar. Although certainly not the only way to meet this requirement, compositionality is a most intuitive way to do so. As such, on-line interpretation does not preclude that in the interpretation of a unit of text, other things than the interpretation of the text sofar play a role. But it does require that at any point in the processing of a text we are able to say what the interpretation thus far is. In other words, it does rule out approaches (such as *DRT*) in which the interpretation of a text is a two-stage process, in which we first build a representation, which only afterwards, i.e., at the end of the text, or a certain segment of it, mediates interpretation of the text as such. So, from the viewpoint of a computational semantics, there is ample reason to try and keep to compositionality.

Yet another reason is provided by certain philosophical considerations. These concern the fact that non-compositional semantic theories usually postulate a level of semantic representation, or 'logical form', in between syntactic form and meaning proper, which is supposed to be a necessary ingredient of a descriptively and explanatorily adequate theory. Consider the following two sequences of sentences (the examples are due to Partee, they are cited from Heim [1982]):

(25) I dropped ten marbles and found all of them, except for one. It is probably under the sofa.

(26) I dropped ten marbles and found only nine of them. It is probably under the sofa.

There is a marked contrast between these two sequences of sentences. The first one is all right, and the pronoun *it* refers to the missing marble. The second sequence, however, is out. Even though it may be perfectly clear to us that the speaker is trying to refer to the missing marble with the pronoun *it*, evidently, this is not the way to do this. Like most authors, we start from the assumption that co-reference and anaphora are, by and large semantic phenomena. ('By and large' in view of the fact that sometimes certain syntactic features are involved in pronoun resolution as well. A case in point is syntactic gender in languages like German and Dutch.) Therefore, we may take the following for granted: the contrast between (23) and (24) marks a difference between the respective opening sentences, and this difference is one of meaning, in the broad, intuitive sense of the word. But what does this difference consist in? For notice that the first sentences of (23) and (24) do characterize the same situation. There is no difference in their truth conditions, therefore it seems that they are semantically equivalent. Indeed, they are equivalent in any standard semantic system that explicates meaning solely in terms of truth (or more generally, denotation) conditions. And we speculate that it is for this reason that many semanticists have taken the view that the difference in question is one of (logical) form, of (semantic) representation, rather than one of content.

For various reasons, we think that one should not adopt this point of view

too hastily. For, it means that one has to postulate an intermediate level of representation in between natural language and its interpretation. True, most semantic frameworks interpret natural language via translation into a logical language, but the general methodological strategy here has always been to make sure that the translation procedure is compositional, and hence, in view of the compositional nature of the interpretation of the logical language, in principle dispensable. The logical translation serves practical purposes only, in principle it can be discarded. But notice that the level of representation that is assumed if one views the difference between (23) and (24) as one of form, is not of this (optional) nature. The two sentences involved will be mapped onto different logical forms, or semantic representations, which in their turn will receive an equivalent interpretation. Accounting for the difference between (23) and (24) in this way, makes the existence of this level of representation imperative, rather than useful. It would be a necessary go-between natural language and its meaning. So it seems that, perhaps without being aware of it, many have put a constraint on the semantics: meaning *is* truth (denotation) conditions. Then, indeed, compositionality becomes a contentfull, rather than a methodological principle, and one which is falsified: the facts force the existence of a level of semantic representation on us.

There are several reasons why we think that the move to a semantic theory which assumes such an independent level of semantic representation, distinct both from syntactic structure and from meaning proper, should be looked upon with reserve. First of all, there is the familiar, almost commonplace reason of theoretical parsimony. Levels of representation, too, should not be multiplied beyond necessity, and although this is perhaps not too exciting a comment to make, we feel that from a methodological point of view it is still a sound one. Of course, its relevance in the present context does presuppose that we are not really forced to introduce such a level of semantic representation, that we can do without it. Such a claim can not be substantiated in general, but it can be shown to be correct in particular cases. And the development of the *DPL*-system shows that, in the case at hand, the principle of compositionality has not only negative implications, but also points positively towards a satisfactory treatment of the issues involved. For the phenomena in question, no level of representation is needed, for compositionality clearly guides towards a notion of meaning which allows us to do without.

Be that as it may, our appeal to this methodological principle will be waved by those who claim that there is empirical evidence for the existence of a level of semantic representation. In fact, quite often when such a level of representation is postulated, this is accompanied by the claim that it is somehow psychologically 'real'. We must be careful in our evaluation here, for one might be making a weaker and a stronger claim. The weaker one is that in producing and understanding language, people somehow represent meanings, extract them from linguistic structures, manipulate them, 'put them into words', and so on. This claim is in fact subsidiary to the view of the mind as a calculating

machine. Notice, however, that this weaker claim is not necessarily at odds with our parsimonious starting point. For, as such there may very well be a separate level of semantic representation, without it being a necessary ingredient of a descriptively and explanatorily adequate semantic theory. The stronger claim adds exactly this to the weaker one: it claims, not just the cognitive reality of representation of meaning, but the existence of a level of representation which carries information that goes beyond that what is represented there, viz., meaning. In effect, this view splits the intuitive notion of meaning in two: those aspects which are covered by the technical notion of meaning (or interpretation) that the theory provides (or borrows from other frameworks), and those which are accounted for by properties of the particular kind of representation of the former.

Thus, a mentalist we call someone who claims that a level of representation is *necessary*, not someone who merely claims that it exists. Should we include among the mentalists the latter kind of person too, we would be forced to consider the Wittgenstein of the *Tractatus* as a mentalist, for he claimed that there exists a level of thoughts and thought-elements which is isomorphic to language, and hence to the world. However, he did consider this level completely irrelevant for an account of the nature of meaning and the way in which it is established. Thus Wittgenstein apparently accepted the *existence* of a level of 'semantic representation', but considered its existence of no interest for semantics proper. In connection with this, it may be worthwhile to point out the close correspondence between the isomorphic 'picturing' relation between language and the world of the *Tractatus*, and the modern-day, algebraic explication of compositionality, as it can be found, for example, in Janssen [1986]. Of course, the later Wittgenstein would have discarded even any talk of a 'cognitive' substratum of our linguistic behaviour, which may help to remind us that even the weaker claim is not as philosophically neutral as some apparently think it is.

To return to the main point, we think that the stronger claim is unwarranted, and that it certainly cannot be justified simply by an appeal to the linguistic facts of the matter. As for the weaker claim, the view on the mind and its operations that it stems from, when taken literally is, of course, not philosophically neutral. Those who really subscribe to it face the burden of showing that there are such things as 'mental' representations and the like, a task which is not without philosophical pitfalls. Notoriously, these issues are as interesting as they are undecidable. Our own opinion, for whatever it is worth, is that the calculating mind is a metaphor rather than a model. It is a powerful metaphor, no doubt, on which many branches of 'cognitive' science are based, and sometimes it can be helpful, even insightful. But it remains a way of speaking, rather than a true description of the way we are. However, whatever stand one would like to take here, it does not affect the point we want to make, which is that it is better to try to keep ones semantic theory, like every theory, as ontologically parsimonious and as philosophically neutral as possible. The stronger claim goes against this, and hence has to be rejected, unless, somehow,

proven.

As for the weaker claim, subscribing to it or not makes no real difference, but one has to be careful not to let it interfere with the way one sets up ones semantic framework. The best way to go about, then, is to carry on semantics as really a discipline of its own, not to consider it a priori a branch of cognitive science, and to enter into the discussion of the reality of mental representations in a 'modular' frame of mind.

It may be the case, though, that for some the acceptance of a level of logical representation springs forth from a positive philosophical conviction, viz., a belief in the deficiencies of natural language as a means to convey meaning. Now such there may be (or not) when we consider very specialized kinds of theoretical discourse, such as mathematics, or philosophy, or particle physics. And again, natural language may be deficient (or not) when we consider a special task that we want to be performed in a certain way, such as running a theorem prover based on natural deduction on natural language sentences, or such a thing. In such cases, clearly there is room for extension and revision, for regimentation and confinement. But that is not what is at stake here. Here, it turns on the question whether natural language structures themselves, as we encounter them in spoken and written language, then and there are in need of further clarification in order to convey what they are meant to convey. In this matter, semantics, we feel, should start from the premiss that natural language is all right. If anything is a perfect means to express natural language meaning, natural language is. It can very well take care of itself and is in no need of (psycho)logical reconstruction and improvement in this respect. To be sure, that means taking a philosophical stand, too, but one that is neutral with respect to the question whether there is such a thing as an indispensable level of logical representation in semantics. As we said above, if such there is, this has to be shown, not taken for granted.

Our ideological point of view concerning the status of mental representations is in line with the methodological interpretation of the principle of compositionality. As was already remarked above, this interpretation not only forces us to reject certain approaches to the problems we started out with, it also positively suggests us a proper solution. Compositionality dictates that the meanings of (23) and (24) should be functions of the meanings of their parts. We take it to be an obvious fact that the immediate components of the sequences of sentences (23) and (24) are the two sentences of which they consist. Because of the difference in acceptability we cannot but conclude that the first sentence of (23) and the first sentence of (24) differ in meaning. Accepting the fact that their truth conditions are the same, this leads to the inevitable conclusion that truth conditions do not exhaust meaning. ('Do not exhaust meaning...', because we do want to stick to the idea that truth conditions are an essential ingredient of meaning.) What compositionality strongly suggests, then, is that we look for an essentially richer notion of meaning of which the truth conditional one is a special case. Our claim is that the kind of dynamic semantics that *DPL* is an

instance of, naturally suggests itself as a first step on the right track.

Acknowledgements

The first version of this paper was presented in June, 1987 at the ASL/LSA-meeting in Stanford, and on several other occasions. A pre-final version was prepared for the First European Summerschool on Natural Language Processing, Logic and Knowledge Representation, held in Groningen in July 1989.

At these and other meetings, and in correspondence, many friends and colleagues have prompted even more questions and comments, which have helped and stimulated us. We thank them, and the two anonynous referees of *Linguistics and Philosophy*.

Since the summer of 1988, our ITLI-colleague Roel de Vrijer has joined in with our work on *DPL*. More in particular, he has dedicated himself to the development of a complete and sound proof theory for *DPL*. The results will be reported in a separate, joined paper. Some of the results of this cooperative work have already penetrated the present paper. More in particular, this holds for the sections on scope and binding, and on entailment. And at many other points as well, we have greatly benefitted from Roel's acute comments and criticisms.

The paper partly originates from a research project carried out by the first author from September 1986 to September 1988, commissioned by Philips Research Laboratories, Eindhoven, The Netherlands. At the final stage of this research, both authors were engaged on the *Dyana*-project (EBRA-3715) commissioned by the European Community.

References

Barwise, J., 1987, 'Noun phrases,generalized quantifiers and anaphora', in: Gärdenfors, P. (ed.) *Generalized quantifiers*, Dordrecht: Reidel, 1–29

Berg, M. van den Berg, 1990, 'A dynamic predicate logic for plurals', in: Stokhof, M. & Torenvliet, L. (eds), *Proceedings of the Seventh Amnsterdam Colloquium*, Amsterdam: ITLI

Chierchia, G., 1988, 'Dynamic generalized quantifiers and donkey anaphora', in: Krifka, M. (ed.), *Genericity in natural language*, Tübingen, SNS

Dekker P., 1990, 'Dynamic interpretation, flexibility, and monotonicity', in: Stokhof, M. & Torenvliet, L. (eds), *Proceedings of the Seventh Amnsterdam Colloquium*, Amsterdam: ITLI

Groenendijk, J. & Stokhof, M., 1988, 'Context and information in dynamic semantics', in: Elsendoorn, B. & Bouma, H. (eds), *Working models of human perception*, New York: Academic Press, 457–488

Groenendijk, J. & Stokhof, M., 1990, 'Dynamic Montague grammar', in: Kálmán, L. et. al. (eds), *Proceedings of the Second Symposion on Logic and Language*, Budapest

Harel, D., 1984, 'Dynamic logic', in: Gabbay, D. & Guenthner, F. (eds), *Handbook of philosophical logic, vol. II*, Dordrecht: Reidel

Heim, I., 1982, *The semantics of definite and indefinite noun phrases*, dissertation, University of Massachusetts, Amherst

Heim, I., 1983, 'File change semantics and the familiarity theory of definiteness', in: Baüerle, R., Schwarze, Ch. & Stechow, A. von (eds), *Meaning, use and interpretation of language*, Berlin: De Gruyter,

Janssen, T., 1986, *Foundations and applications of Montague grammar*, Amsterdam: CWI

Kadmon, N. 1987, *On unique and non-unique reference and asymmetric quantification*, dissertation, University of Massachusetts, Amherst

Kamp, H., 1981, 'A theory of truth and semantic representation', in: Groenendijk, J., Janssen, T. & Stokhof, M. (eds), *Formal methods in the study of language*, Amsterdam: Mathematical Centre, 277–322 [reprinted in: Groenendijk, J., Janssen, T. & Stokhof, M. (eds), 1984, *Truth, interpretation and information*, Dordrecht: Foris, 1–41]

Kamp, H., 1983, 'Situations in discourse without time or questions', manuscript

Lewis, D., 1975, 'Adverbs of quantification', in: Keenan, E. (ed), *Formal semantics of natural language*, Cambridge: Cambridge University Press, 3–15

Roberts, C., 1987, *Modal subordination, anaphora, and distributivity*, dissertation, University of Massachusetts, Amherst

Roberts, C., 1989, 'Modal subordination and pronominal anaphora in discourse', *Linguistics and Philosophy*, 12, 683–723

Rooth, M., 1987, 'Noun phrase interpretation in Montague grammar, file change semantics, and situation semantics', in: Gärdenfors, P. (ed.) *Generalized quantifiers*, Dordrecht: Reidel, 237–269

Schubert, L. & Pelletier, F.J., 1989, 'Generally speaking, or, using discourse representation theory to interpret generics', in: Chierchia, G., Partee, B.H. & Turner, R., *Properties, types and meaning, Vol. II*, Dordrecht: Kluwer, 193–268

Seuren, P., 1986, *Discourse semantics*, Oxford: Blackwell

Zeevat, H., 1989, 'A compositional approach to discourse representation theory', *Linguistics and Philosophy*, 12, 95–131