Tilburg University

Dynamics

Muskens, R.A.; van Benthem, J.; Visser, A.

Published in:
Handbook of logic and language

Publication date:
1997

Link to publication in Tilburg University Research Portal

Citation for published version (APA):
Muskens, R. A., van Benthem, J., & Visser, A. (1997). Dynamics. In J. van Benthem, & A. ter Meulen (Eds.), *Handbook of logic and language* (pp. 587-648). Elsevier Science Publishers.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 09. Dec. 2023

CHAPTER 10

Dynamics

Reinhard Muskens

Department of Linguistics, Tilburg University, PO Box 90153, 5000 LE Tilburg, The Netherlands
E-mail: r.a.muskens@kub.nl

Johan van Benthem

University of Amsterdam, ILLC, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands
E-mail: johan@fwi.uva.nl, johan@csli.stanford.edu

Albert Visser

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

Commentator: D. McCarty

Contents

HANDBOOK OF LOGIC AND LANGUAGE
Edited by J. van Benthem and A. ter Meulen
© 1997 Elsevier Science B.V. All rights reserved

0. Introduction

Intriguing parallels can be observed between the execution of computer programs and the interpretation of ordinary discourse. Various elements of discourse, such as assertions, suppositions and questions, may well be compared with statements or sequences of statements in an imperative program. Let us concentrate on assertions for the moment. Stalnaker (1979) sums up some of their more or less obvious characteristics in the following way.

> Let me begin with some truisms about assertions. First, assertions have content; an act of assertion is, among other things, the expression of a proposition – something that represents the world as being a certain way. Second, assertions are made in a context – a situation that includes a speaker with certain beliefs and intentions, and some people with their own beliefs and intentions to whom the assertion is addressed. Third, sometimes the content of the assertion is dependent on the context in which it is made, for example, on who is speaking or when the assertion takes place. Fourth, acts of assertion affect, and are intended to affect, the context, in particular the attitudes of the participants in the situation; how the assertion affects the context will depend on its content.

If we are prepared to think about assertions as if they were some special kind of programs, much of this behaviour falls into place. That assertions are made in a context may then be likened to the fact that execution of a program always starts in a given initial state; that the content of an assertion may depend on the context parallels the situation that the effect of a program will usually depend on this input state (for example, the effect of $x := y + 7$ will crucially depend on the value of y before execution); and that a program or part of a program will change and is intended to change the current program state is no less a truism as the contention that an act of assertion changes the context. After the change has taken place, the new state or the new context can serve as an input for the next part of the program or the next assertion.

The metaphor helps to explain some other features of discourse as well. For instance, it makes it easier to see why the meaning of a series of assertions is sensitive to order, why saying "John left. Mary started to cry." is different from saying "Mary started to cry. John left.". Clearly, the result of executing two programs will in general also depend on the order in which we run them. If we think about sequences of sentences as ordinary conjunctions on the other hand, this non-commutativity remains a puzzle. The picture also helps us see how it can be that some assertions are inappropriate in certain contexts, why we cannot say "Harry is guilty too" with a certain intonation just after it has been established that nobody else is guilty. This is like dividing by x just after x has been set to 0.

Discourse and programming then, seem to share some important structural properties, to the extent that one can serve as a useful metaphor for the other. We need not restrict application of the metaphor to that part of discourse that is expressed by overt linguistic means. Not only are assertions, suppositions and questions made in a context, other, non-verbal, contributions to conversation, such as gestures and gazes, are too. These non-verbal acts of communication likewise have a potential to change the current context state. A speaker may for instance introduce discourse referents into the conversation with the help of a gesture or a gaze, or may use such means (or more overt linguistic ones such as changes in tense and aspect or rise in pitch) to announce the introduction of a new 'discourse segment purpose' (Grosz and Sidner, 1986; Polanyi, 1985). Appropriateness conditions for gestures or gazes do not seem to differ in principle from those for linguistic acts: a case of pointing where there is nothing to be pointed at may be likened to saying "The king is bald" where there is no king, or the use of a variable that has not been declared.

But if even gestures and gazes share the structural properties that we have seen are common to computer programs and linguistic acts, then we may wonder whether the properties involved are not simply those that *all* actions (or at least all rule-based actions) have in common, and indeed we feel that this is the right level of abstraction to think about these matters. An action – whether it be a communicative act, the execution of an assignment statement, a move in chess, or simply the movement of an arm – is performed in a given situation, typically changes that situation, and is dependent upon that situation for the change that it brings about. The effect of castling is dependent on the previous configuration on the board and your friend's stepping forward may result in his stepping on your toe in some situations but not in others. The order in which we perform our actions will typically effect the result, as we are all aware, and in many situations an action may be inappropriate – you cannot move your rook if this exposes your king.

The similarity between linguistic acts and moves in a game was stressed by the philosopher Ludwig Wittgenstein (Wittgenstein, 1953), but the first paper with immediate relevance to theoretical linguistics that explicitly took such similarities as its point of departure was the influential Lewis (1979). In this article, which refers to Wittgenstein in its title, Lewis compares conversation with baseball and says that 'with any stage in a well-run conversation, there are many things analogous to the components of a baseball score'. The latter is defined as a septuple of numbers: the number of home team runs, the number of runs that the visiting team has, the half (1 or 2), and so on. And in a similar way Lewis lets conversational score consist of several components: a component that keeps track of the presuppositions at any moment of conversation, a component that ranks the objects in the domain of discourse according to salience, the point of reference, the possible worlds that are accessible at any given point, and many others. Just as the rules of baseball tell us how the actions of the players alter the baseball score, the rules of conversation specify the kinematics of context change. If you mention a particular cat during conversation, for example, the rules bring it about that that cat will become salient and that a subsequent use of the definite description "the cat" will most likely refer to it. And if you say "John went to Amsterdam", the point of reference will move to Amsterdam as well, so that if you continue by saying "Mary came the following day", it will be understood that Mary came to Amsterdam and not to any other place.

Clearly, Lewis' picture of a conversational scoreboard that gets updated through linguistic acts of the participants in a conversation has much in common with our previous computational picture. In fact, we can imagine the conversational scoreboard to be a list of variables that the agents may operate on by means of programs according to certain rules. But a caveat is in order, for although there are important structural similarities between games and programs on the one hand and discourse on the other, there are of course also many features that are particular to conversation and our metaphor is not intended to make us blind to these. An example is the phenomenon of *accommodation* that Lewis describes. If at some point during a conversation a contribution is made that, in order to be appropriate, requires some item of conversational score to have a certain value, that item will automatically assume that value. For instance, if you say "Harry is guilty too" in a situation where the presupposition component of conversational score does not entail that someone else is guilty (or that Harry has some salient property besides being guilty), that very presupposition will immediately come into existence. This accommodation does not seem to have a parallel in games or computing: trying to divide by x after this variable has been set to 0 will not reset x to another value and, to take an example used by Lewis, a batter's walking to first base after only three balls will not make it the case that there were four balls after all.

Such examples, however, need not change the basic picture. That conversation and other cognitive activities have many special properties besides the ones that they have in virtue of being examples of rule-governed activities in general need not surprise us. Accommodation can be thought of as such a special property and we may model it as one of the particular effects that the programs that model communicative acts have; one of the effects that they have in virtue of being a special kind of program rather than just any program. It is the logic of the general properties that we are after in this paper.

The paper is divided into two sections. In Section 1, without any attempt at giving a complete rubrication, we shall give an overview of some important dynamic theories in linguistics and artificial intelligence which have emerged in the last two decades and we shall see how these fit into the general perspective on communication sketched in this introduction. In Section 2 we shall offer some more general logical considerations on dynamic phenomena, discussing various ways to model their logic and discussing how the logic that emerges is related to its classical static predecessors.

1. Some specific dynamic systems

1.1. The kinematics of context change: Stalnaker, Karttunen, Heim and Veltman

Certain things can only be said if other things are taken for granted. For example, if you say (1a) you signal that you take the truth of (1b) for granted, and a similar relation obtains between (2a) and (2b) and between (3a) and (3b). The (b)-sentences are presuppositions of the (a)-sentences and in a situation where the falsity of any of the (b)-sentences is established, the corresponding (a)-sentence cannot be uttered felicitously (for an overview of theories of presupposition cf. Soames (1989), Beaver (1996), this Handbook).

(1a) The king of France likes bagels.
(1b) France has a king.
(2a) All of Jack's children are fools.
(2b) Jack has children.
(3a) John has stopped seeing your wife.
(3b) John was seeing your wife.

Stalnaker (1974) gives a rough definition of the notion of presupposition which runs as follows: *a speaker presupposes that P at a given moment in a conversation just in case he is disposed to act, in his linguistic behavior, as if he takes the truth of P for granted,* *and as if he assumes that his audience recognizes that he is doing so.* Note that this defines the notion of presupposition not only relative to a speaker and the assumptions that he makes regarding his audience, but also relative to a moment in conversation. This leaves open the possibility that the set of propositions that can be assumed to be taken for granted changes during discourse and indeed this is what normally happens. When you say: 'John was seeing your wife', you may from that moment on assume that your audience recognizes that you take it for granted that he did. Consequently, in order to be able to say (4) you need not assume in advance that your audience recognizes anything at all about your views on his wife's past fidelity; the necessary precondition for a felicitous uttering of the second conjunct will be in force from the moment on that the first conjunct has been uttered, regardless of the assumptions that were made beforehand.

(4) John was seeing your wife but he has stopped doing so.
(5) If France has a king, then the king of France likes bagels.
(6) Either Jack has no children or all of his children are fools.
(7) The king of France does not like bagels.

In (5) and in (6) something similar happens. If a speaker utters a conditional, his audience can be assumed to take the truth of the antecedent for granted during the evaluation of the consequent and hence a speaker need not presuppose that France has a king in order to utter (5) in a felicitous way. Similarly, when evaluating the second part of a disjunction, a hearer will conventionally take the falsity of the first part for granted and so (6) can be uttered by someone who does not presuppose that Jack has children. The presuppositions that a speaker must make in order to make a felicitous contribution to discourse with a negated sentence on the other hand do not seem to differ from those of the sentence itself and so (7) simply requires (1b) to be presupposed.

Such regularities suggest the possibility of calculating which presuppositions are in force at any given moment during the evaluation of a sentence and indeed rules for calculating these are given in (Karttunen, 1974). Let us call the set of sentences C that are being presupposed at the start of the evaluation of a given sentence S the *initial context* of S. Then we can assign local contexts $\mathrm{LC}(S')$ to all subclauses S' of S by letting $\mathrm{LC}(S) = C$ and, proceeding in a top-down fashion, by assigning local contexts to the proper subclauses of S with the help of the following rules.

(i) $\mathrm{LC}(\text{not } S) = C \Rightarrow \mathrm{LC}(S) = C$,
(ii) $\mathrm{LC}(\text{if } S \text{ then } S') = C \Rightarrow \mathrm{LC}(S) = C \ \& \ \mathrm{LC}(S') = C \cup \{S\}$,
(iii) $\mathrm{LC}(S \text{ and } S') = C \Rightarrow \mathrm{LC}(S) = C \ \& \ \mathrm{LC}(S') = C \cup \{S\}$,
(iv) $\mathrm{LC}(S \text{ or } S') = C \Rightarrow \mathrm{LC}(S) = C \ \& \ \mathrm{LC}(S') = C \cup \{\text{not } S\}$.

The local context of a clause consists of the presuppositions that are in force at the time the clause is uttered. The rules allow us to compute e.g. the local context of the first occurrence of S' in 'if (S and S') then (S'' or S')' as $C \cup \{S\}$, where C is the initial context, and the local context of the second occurrence of this sentence can be computed to be $C \cup \{S \text{ and } S', \ \text{not } S''\}$.

A speaker who presupposes an initial set of sentences C is now predicted to be able to utter a sentence S felicitously just in case the local context of each subclause of S entails all presuppositions that are triggered at the level of that subclause. If this is the case we say that C *admits* or *satisfies the presuppositions of* S. Since, e.g., C need not entail that Jack has children in order to admit (6) it is predicted that a speaker need not presuppose that he has in order to be able to make a suitable contribution to discourse with the help of this sentence.

Rules (i)–(iv) only allow us to compute the admittance conditions of sentences that are built from atomic clauses with the usual propositional connectives, but Karttunen also extends the theory to sentences constructed with complementizable verbs. The latter are divided into three: (a) verbs of saying such as *say, mention, warn, announce* and the like, which are called *plugs*; (b) verbs such as *believe, fear, think, doubt* and *want*, which are *filters*; and (c) verbs such as *know, regret, understand* and *force*, which are *holes*. Three extra rules are needed for assigning local contexts to the subclauses of sentences containing these constructions.

(v) $\mathrm{LC}(NP\ V_{\mathrm{plug}} S) = C \Rightarrow \mathrm{LC}(S) = \{\bot\}$,
(vi) $\mathrm{LC}(NP\ V_{\mathrm{filter}} S) = C \Rightarrow \mathrm{LC}(S) = \{S' \mid NP\ \mathit{believes}\ S' \in C\}$,
(vii) $\mathrm{LC}(NP\ V_{\mathrm{hole}} S) = C \Rightarrow \mathrm{LC}(S) = C$.

For example, in (8) the local context for 'the king of France announced that John had stopped seeing his wife' is simply the initial context C, and so a speaker who is to utter (8) should presuppose that there is a king. But it need not be presupposed that John was seeing Bill's wife since the local context for the complement of *announce* is simply the falsum $\bot$, from which the required presupposition follows of course. With respect to (9) it is predicted that the initial context must entail that Sue believes there to be a king of France and that she believes that Jack has children for the utterance to be felicitous.

(8) Joe forced the king of France to announce that John had stopped seeing Bill's wife.

(9) Sue doubts that the king of France regrets that all of Jack's children are fools.

Karttunen's rules for the admittance conditions of a sentence are completely independent from the rules that determine its truth conditions (a feature of the theory criticized in (Gazdar, 1979)), but Heim (1983a) shows that there is an intimate connection. Many authors (e.g., Stalnaker, 1979) had already observed that a sequence of sentences $S_1, \dots, S_n$ suggests a dynamic view of shrinking sets of possibilities $[S_1]$, $[S_1] \cap [S_2]$, ..., $[S_1] \cap \dots \cap [S_n]$, where each $[S_i]$ denotes the possibilities that are compatible with sentence S_i. The idea is illustrated by the game of *Master Mind*, where some initial space of possibilities for a hidden sequence of colored pegs is reduced by successive answers to one's guesses, encodable in conjunctions of propositions like "either the green peg is in its correct position or the blue one is". Complete information corresponds to the case where just one possibility is left. Identifying the possibilities that are still open at any point with the local context C, we may let the *context change potential* $\|S\|$ of a sentence S be defined as the function that assigns $C \cap [S]$ to any C. Processing $S_1, \dots, S_n$ will then reduce an initial context C to $\|S_1\| \circ \dots \circ \|S_n\|(C)$, where $\circ$ denotes composition of functions.

This last set-up defines the context change potential of a sentence in terms of its truth conditions, but Heim takes the more radical approach of defining truth conditions in terms of context change potentials. The context change potential of a complex expression in her theory is a function of the context change potentials of its parts. In particular, she dynamicizes the interpretation of the propositional connectives by giving the following clauses for negation and implication.[1]

$$\|\text{not } S\|(C) = C - \|S\|(C),$$
$$\|\text{if } S \text{ then } S'\|(C) = C - (\|S\|(C) - \|S'\|(\|S\|(C)).$$

The functions $\|S\|$ considered here may be undefined on contexts C where the presuppositions of S fail to hold and it is to be understood that if an argument of a function is undefined, the value of that function also is. For example, $\|\text{if } S \text{ then } S'\|(C)$ is defined if and only if both $\|S\|(C)$ and $\|S'\|(\|S\|(C))$ are. This means that C acts as a local context of S, while $\|S\|(C)$ is the local context of S'. The local context for S in $\|\text{not } S\|(C)$ simply is C. Essentially then, Karttunen's local contexts for a sentence can be derived from the definition of its context change potential, but the definition also determines the sentence's truth conditions, as we may define S to be *true* in a point i iff $\|S\|(\{i\}) = \{i\}$ and *false* in i iff $\|S\|(\{i\}) = \emptyset$.[2] For sentences not containing any presupposition this is just the standard notion, but a sentence S may be neither true nor false in i if $\|S\|(\{i\})$ is undefined.

Heim's idea suggests adding a two-place presupposition connective / to the syntax of propositional logic, where φ/ψ is to mean that ψ holds but that φ is presupposed.[3] We shall interpret the resulting system dynamically, letting contexts be sets of ordinary valuations V, and defining context change potentials as follows.

(i) $\|p\|(C) = C \cap \{V \mid V(p) = 1\}$ if p is atomic,
(ii) $\|\neg\varphi\|(C) = C - \|\varphi\|(C)$,
(iii) $\|\varphi \wedge \psi\|(C) = \|\psi\|(\|\varphi\|(C))$,
(iv) $\|\varphi/\psi\|(C) = \|\psi\|(C)$ if $\|\varphi\|(C) = C$,
$\quad = \text{undefined otherwise.}$

The demand that $\|\varphi\|(C) = C$ is a way to express admittance of φ by the context C (compare the notion of *acceptance* in (Veltman, 1991)). Again, it is to be understood that if an argument of a function is undefined, the value of that function also is. Implication and disjunction can be defined as usual, i.e. $\varphi \to \psi$ is to abbreviate $\neg(\varphi \wedge \neg\psi)$ and $\varphi \vee \psi$ is short for $\neg\varphi \to \psi$. The reader is invited to verify that the resulting logic gives us exactly the same admittance conditions as we had in (the propositional part of) Karttunen's theory. In particular, we may formalize sentences (4), (5) and (6) as $p \wedge (p/q)$, $p \to (p/q)$ and $\neg p \vee (p/q)$ respectively and see that these are admitted by any context.

[1] Heim writes $C + S$ where we prefer $\|S\|(C)$.

[2] Heim (1983a) and Heim (1982, p. 330) let a context (or a file) be true iff it is non-empty. A sentence S is then stipulated to be true with respect to a given C if $\|S\|(C)$ is true, and false with respect to C if C is true and $\|S\|(C)$ is false. The case where both C and $\|S\|(C)$ are false is not covered. Heim notices this and in (Heim, 1982) makes an effort to defend the definition. The present definition is more limited than Heim's original one, since it essentially instantiates C as $\{i\}$. But truth in i is always defined in our definition and the definition serves its purpose of showing that classical truth conditions can be derived from context change potentials.

[3] See Beaver (1992) for a unary presupposition connective ∂ which is interdefinable with /.

This then is a version of propositional logic which supports presuppositions and is truly dynamic, as its fundamental semantic notion is that of context change potential rather than truth. The reader be warned though that an alternative static definition gives exactly the same results. To see this, define the *positive extension* $[\varphi]^+$ and the *negative extension* $[\varphi]^-$ of each sentence φ as follows.

(i′) $[p]^+ = \{V \mid V(p) = 1\}$, $[p]^- = \{V \mid V(p) = 0\}$.

(ii′) $[\neg\varphi]^+ = [\varphi]^-$, $[\neg\varphi]^- = [\varphi]^+$.

(iii′) $[\varphi \wedge \psi]^+ = [\varphi]^+ \cap [\psi]^+$, $[\varphi \wedge \psi]^- = [\varphi]^- \cup ([\varphi]^+ \cap [\psi]^-)$.

(iv′) $[\varphi/\psi]^+ = [\varphi]^+ \cap [\psi]^+$, $[\varphi/\psi]^- = [\varphi]^+ \cap [\psi]^-$.

The connectives $\neg$ and $\wedge$ are essentially treated as in (Peters, 1975) here (see also Karttunen and Peters, 1979), while/is the so-called *transplication* of Blamey (1986). An induction on the complexity of φ will show for any C (a) that $\|\varphi\|(C)$ is defined iff $C \subseteq [\varphi]^+ \cup [\varphi]^-$ and (b) that $\|\varphi\|(C) = C \cap [\varphi]^+$ if $\|\varphi\|(C)$ is defined. This means that Heim's logic is not essentially dynamic after all, even if its dynamic formulation is certainly natural.

Essentially dynamic operators do exist, however. Let us call a total unary function F on some power set *continuous* if it commutes with arbitrary unions of its arguments, i.e. if for any indexed set $\{C_i \mid i \in I\}$ it holds that $\cup\{F(C_i) \mid i \in I\} = F(\cup\{C_i \mid i \in I\})$. Call F *introspective* if $F(C) \subseteq C$ for any C. Van Benthem (1986) shows that these two properties give a necessary and sufficient criterion for an operator to be static: F is continuous and introspective if and only if there is some P such that $F(C) = C \cap P$ for all C (see also Groenendijk, Stokhof and Veltman, 1996). This means that an essentially dynamic operator must either not be continuous or not be introspective. A key example of a non-continuous operator is Veltman's (1991) epistemic *might* in a theory called *Update Semantics*. A minimal version of Veltman's system can be obtained by taking propositional modal logic and interpreting it by adding the following clause to (i)–(iii) above.

$$\begin{aligned} \| \diamond\varphi\|(C) = \emptyset \quad & \text{if } \|\varphi\|(C) = \emptyset, \\ C \quad & \text{otherwise.} \end{aligned}$$

The operator helps explain the difference between the acceptability of discourses such as (10) and (11).

(10) Maybe it is raining. ... It is not raining.
(11) It is not raining. ... # Maybe it is raining.

A naive translation into modal logic would make this into the commutative pair $\diamond r \wedge \neg r, \neg r \wedge \diamond r$. But dynamically, there is a difference. In (10) the initial state can still be consistently updated with the information that it is raining. Only after the second sentence is processed this possibility is cut off. In (11), however, the information that it is not raining has been added at the start, after which the test for possibility of raining will fail. This modality is no longer a continuous function, and it does not reduce to classical propositions in an obvious way. Nevertheless, there are still strong connections with classical systems. Van Benthem (1988) provides a translation into monadic predicate logic computing the update transitions, and Van Eijck and De Vries (1995) improve this to a translation into the modal logic S5, where $\diamond$ behaves like a modality after all. This means that these systems are still highly decidable.

In addition to mere elimination of possibilities the update framework also supports other forms of movement through its phase space. A phrase like *unless* φ, for instance, may call for enlargement of the current state by reinstating those earlier situations where φ held. Other plausible revision operators which are not introspective in the sense given above are not hard to come by.

Clearly the picture of updating information that is sketched here, with contexts or information states being flatly equated with sets of valuations, gives an extremely simplified model of what goes on in actual natural language understanding and it is worthwhile to look for subtler definitions of the notion of information state and for operations on information states subtler than just taking away possibilities or adding them. Assertions, for example, may not only change our views as to which things are possible, they may also upgrade our preferences between possibilities, i.e. change our views as to which possibilities are more likely than others. The latter phenomenon may be represented in terms of preference relations between models, as it is currently done in Artificial Intelligence (Shoham, 1988) in a tradition that derives from Lewis's possible worlds semantics for conditional logic (cf. Lewis, 1973; Veltman, 1985). For instance, processing a conditional default rule *if A, then B* need not mean that any exceptions (i.e. *A* & *not B* worlds) are forcibly removed, but rather that the latter are downgraded in some sense. This idea has been proposed in (Spohn, 1988; Boutilier, 1993; Boutilier and Goldszmidt, 1993) – and most extensively, for natural language, in (Veltman, 1991). In the latter system, static operators may model adverbs like *presumably* or *normally*, whereas a default conditional leads to a change in expectation patterns. To simplify matters, in what follows, φ, ψ are classical formulas. States C now consist of a set of worlds plus a preference order $\leqslant$ over them, forming a so-called *expectation pattern.* Maximally preferred worlds in such patterns are called *normal.* Incoming propositions may either change the former 'factual' component, or the latter (or both). For instance, given C and φ we may define the upgrade C_φ as that expectation pattern which has the same factual component as C, but whose preference relation consists of $\leqslant$ with all pairs $\langle w, v\rangle$ taken out in which we have $v \models \varphi$ without $w \models \varphi$.

$$\begin{aligned} \|\textit{normally } \varphi\|(C) &= C_\varphi \text{ if } \varphi \text{ is consistent with some normal world,} \\ &\quad \emptyset \quad \text{otherwise.} \\ \|\textit{presumably } \varphi\|(C) &= C \quad \text{if } \varphi \text{ holds in all maximally preferred situations in } C, \\ &\quad \emptyset \quad \text{otherwise.} \end{aligned}$$

A much more complicated explication takes care of the binary operator *if* φ, *then* ψ. Cf. Veltman (1991) for details, basic theory and applications of the resulting system. In particular, this paper provides a systematic comparison of the predictions of this system against intuitions about natural default reasoning. A more abstract perspective on update semantics is provided in (Van Benthem, Van Eijck and Frolova, 1993), which also includes connections with dynamized versions of conditional logic.

1.2. Change of assignments: Heim, Kamp, Groenendijk and Stokhof

A person who is reading a text must keep track of the items that are being introduced, since these items may be referred to again at a later point. The first sentence of text (12), for example, requires its reader to set up *discourse referents* (the term and the idea are from (Karttunen, 1976)) for the indefinite noun phrases *a woman* and *a cat*. The anaphoric pronoun *it* in the second sentence can then be interpreted as picking up the discourse referent that was introduced for *a cat* and the pronoun *her* may pick up the referent for *a woman*. Thus, while you are reading, not only the set of sentences that you can be assumed to take for granted changes, but your set of discourse referents grows as well. This latter growth gives us another example of contextual change.

(12) A woman catches a cat. It scratches her.

There are many semantic theories that use this kind of change to explain the possibilities and impossibilities of anaphoric linking in natural language. Here we shall briefly discuss three important ones, *File Change Semantics* (FCS, Heim, 1982, 1983b), *Discourse Representation Theory* (DRT, Kamp, 1981; Kamp and Reyle, 1993; Van Eijck and Kamp, 1996; this Handbook), and *Dynamic Predicate Logic* (DPL, Groenendijk and Stokhof, 1991). The first two of these theories were formulated independently in the beginning of the eighties, address roughly the same questions and make roughly the same predictions (see also Seuren, 1975, 1985), the third was formulated at a later time and differs mainly from the first and second from a methodological point of view.

1.2.1. File change semantics

The basic metaphor underlying Heim's theory is a comparison between the reader of a text and a clerk who has to keep track of all that has been said by means of a file of cards. Each card in the file stands for a discourse referent and the information that is written on the cards tells us what we have learned about this discourse referent thus far. Reading text (12), for example, the clerk would first have to make a card for the indefinite noun phrase *a woman*.

x_1
x_1 is a woman

His next step would be to set up a card for *a cat*. His file now looks as follows.

x_1
x_1 is a woman

x_2
x_2 is a cat

The information that the woman catches the cat is now written upon both cards,

x_1
x_1 is a woman x_1 catches x_2

x_2
x_2 is a cat x_2 is caught by x_1

and finally the second sentence is interpreted. *It* is interpreted as x_2 and *her* is identified with x_1. This leads to the following file.

x_1
x_1 is a woman x_1 catches x_2 x_1 is scratched by x_2

x_2
x_2 is a cat x_2 is caught by x_1 x_2 scratches x_1

In this way our clerk proceeds, setting up a new card for each indefinite noun phrase that he encounters and identifying each definite noun phrase with a card that was already there. A file is said to be *true* if there is some way of assigning objects to the discourse referents occurring in it such that all the statements on the cards come out true, i.e. a file is true (in a given model) if there is some finite assignment satisfying all the open sentences in it, it is false if there is no such assignment. In fact, for the purposes at hand we can identify a file F with a pair $\langle Dom(F),\ Sat(F)\rangle$, where $Dom(F)$, the *domain* of F, is the set of all discourse referents (i.e. variables) occurring in F and $Sat(F)$, the *satisfaction set* of F, is the set of assignments with domain $Dom(F)$ which satisfy F. The meaning of a text is now identified with its *file change potential*, the way in which it alters the current file. Formally, it is a partial function from files to files.

Texts are connected to their file change potentials via a two-tier procedure in Heim's system. First, at the level of syntax, the text is associated with its so-called *logical form*. Logical forms are then interpreted compositionally by means of file change potentials. We shall look at each of these steps in a little detail.

The logical form of a sentence, which may be compared to the analysis tree that it gets in Montague Grammar, or to its logical form (LF) in contemporary generative grammar (cf. Higginbothan, this Handbook), is obtained from the syntactic structure of that sentence via three rules. The first, *NP Indexing*, assigns each NP a referential index. For ease of exposition we shall assume here that this index appears on the determiner of the noun phrase. If we apply NP Indexing to (14) (which for our purposes we may take to be the surface structure of (13)), for instance, (15) is a possible outcome. The second rule, *NP Prefixing*, adjoins every non-pronominal NP to S and leaves a coindexed empty NP behind. A possible result of this transformation when applied to (15) is (16), but another possibility (which will result in the wide scope reading for *a cat*) is (17). The last rule, *Quantifier Construal*, attaches each quantifier as a leftmost immediate constituent of S. Determiners such as *every, most* and *no* count as quantifiers in Heim's system, but the determiners *a* and *the* do not. The result of applying the transformation to (16) is (18) and applying it to (17) gives (19).

(13) Every woman catches a cat,

(14) $[_S[_{NP}\text{every woman}][_{VP}\text{catches}[_{NP}\text{a cat}]]]$,

(15) $[_S[_{NP}\text{every}_1\text{ woman}][_{VP}\text{catches}[_{NP}\text{a}_2\text{ cat}]]]$,

(16) $[_S[_{NP}\text{every}_1\text{ woman}][_S[_{NP}\text{a}_2\text{ cat}][_S e_1\text{ catches } e_2]]]$,

(17) $[_S[_{NP}\text{a}_2\text{ cat}][_S[_{NP}\text{every}_1\text{ woman}][_S e_1\text{ catches } e_2]]]$,

(18) $[_S\text{every}[_{NP-1}\text{woman}][_S[_{NP}\text{a}_2\text{ cat}][_S e_1\text{ catches } e_2]]]$,

(19) $[_{S}[_{NP}a_2 \text{ cat}][_{S}\text{every}[_{NP-1}\text{woman}][_{S}e_1 \text{ catches } e_2]]]$.

The logical form of a text consisting of sentences $S_1, \ldots, S_n$ (in that order) will simply be $[_T\xi_1 \cdots \xi_n]$, where each of the ξ_i is the logical form of the corresponding S_i. For example, (20) will be the logical form of text (12).

(20) $[_T[_S[_{NP}a_1 \text{ woman}][_S[_{NP}a_2 \text{ cat}][_S e_1 \text{ catches } e_2]]][_S it_2 \text{ scratches her}_1]]$.

Logical forms such as (18), (19) and (20) can now be interpreted compositionally; each will be associated with a partial function from files to files. The smallest building blocks that the interpretation process will recognize are atoms such as $[_{NP}a_1 \text{ woman}]$, $[_{NP-1}\text{woman}]$, $[_S e_1 \text{ catches } e_2]$ and $[_S it_2 \text{ cratches her}_1]$, all of the form $[x_{i_1} R x_{i_2} \cdots x_{i_n}]$, with definite and indefinite determiners, pronouns, empty NP's and the trace – identified with variables x. We shall assume that indefinite determiners and the trace – carry a feature [−def] and that the other variables are [+def]. The following condition gives us the domain of the file change potential $\|[x_{i_1} R x_{i_2} \cdots x_{i_n}]\|$.

(i^a) $\|[x_{i_1} R x_{i_2} \cdots x_{i_n}]\|(F)$ is defined iff for each x_{i_k} $(1 \leqslant k \leqslant n)$:
(Novelty) if x_{i_k} is [–def] then $x_{i_k} \notin Dom(F)$ and
(Familiarity) if x_{i_k} is [+def] then $x_{i_k} \in Dom(F)$.

This requirement, which Heim calls the *Novelty/Familiarity Condition*, corresponds to the file clerk's instruction to make a new card whenever he encounters an indefinite noun phrase but to update an old card whenever he encounters a definite NP.

In order to define what $\|[x_{i_1} R x_{i_2} \cdots x_{i_n}]\|(F)$ is in case the Novelty/Familiarity requirement is met, we suppose that a first-order model $M = \langle D, I\rangle$ that interprets the predicates of our language is given and stipulate the following.

(i^b) If $\|[x_{i_1} R x_{i_2} \cdots x_{i_n}]\|(F)$ is defined then
$Dom(\|[x_{i_1} R x_{i_2} \cdots x_{i_n}]\|(F)) = Dom(F) \cup \{x_{i_1}, \ldots, x_{i_n}\}$
$Sat(\|[x_{i_1} R x_{i_2} \cdots x_{i_n}]\|(F)) = \{a \mid dom(a) = Dom(F) \cup \{x_{i_1}, \ldots, x_{i_n}\}$ &
$\exists b \subseteq a$: $b \in Sat(F)$ & $\langle a(x_{i_1}), \ldots, a(x_{i_n})\rangle \in I(R)\}$.

For example, if we apply $\|[_{NP}a_1 \text{ woman}]\|$ to the empty file $\langle\emptyset, \{\emptyset\}\rangle$, i.e. the file with empty domain and satisfaction set $\{\emptyset\}$, we obtain the file with domain $\{x_1\}$ and satisfaction set (21). If we apply $\|[_{NP}a_2 \text{ cat}]\|$ to the latter we get (22) as our new satisfaction set and $\{x_1, x_2\}$ as the new domain. Applying $\|[_S e_1 \text{ catches } e_2]\|$ to this file sets the satisfaction set to (23) and leaves the domain as it is. A last application of $\|[_S it_2 \text{ scratches her}_1]\|$ changes the satisfaction set to (24). Of course this set is non-empty if and only if (25) is true.

(21) $\{\{\langle x_1, d'\rangle\} \mid d' \in I(\text{woman})\}$,
(22) $\{\{\langle x_1, d'\rangle, \langle x_2, d''\rangle\} \mid d' \in I(\text{woman})\ \&\ d'' \in I(\text{cat})\}$,
(23) $\{\{\langle x_1, d'\rangle, \langle x_2, d''\rangle\} \mid d' \in I(\text{woman})\ \&\ d'' \in I(\text{cat})\ \&\ \langle d', d''\rangle \in I(\text{catches})\}$,
(24) $\{\{\langle x_1, d'\rangle, \langle x_2, d''\rangle\} \mid d' \in I(\text{woman})\ \&\ d'' \in I(\text{cat})\ \&\ \langle d', d''\rangle \in I(\text{catches})\ \&$
$\langle d'', d'\rangle \in I(\text{scratches})\}$,

(25) $\exists x_1 x_2(\text{woman } x_1 \wedge \text{cat } x_2 \wedge \text{catches } x_1 x_2 \wedge \text{scratches } x_2 x_1)$.

Thus by successively applying the atoms of (20) in a left-to-right fashion we have obtained its satisfaction set and thereby its truth conditions. Indeed, the general rule for obtaining the file change potential of two or more juxtaposed elements from the file change potentials of those elements is simply functional composition.

(ii) $\|[\xi_1 \cdots \xi_n]\|(F) = \|\xi_1\| \circ \cdots \circ \|\xi_n\|(F)$.

Note that the interpretation process of (20) would have broken down if $[_{\text{NP}}\text{a}_2 \text{ cat}]$ would have been replaced by $[_{\text{NP}}\text{a}_1 \text{ cat}]$ (a violation of the Novelty condition) or if, say, it_2 would have been replaced by it_6, which would violate Familiarity. Thus some ways to index NPs lead to uninterpretability.

With the help of rules (i) and (ii) we can only interpret purely existential texts; universals are treated somewhat differently. While an indefinite makes the domain of the current file grow, application of a universal sentence leaves it as it is. On the other hand, in general it will cause the satisfaction set to decrease. The following definition gives us the file change potential of a universal sentence.

(iii) $Dom(\|[\text{every } \xi\theta]\|(F)) = Dom(F)$,
$Sat(\|[\text{every } \xi\theta]\|(F)) = \{a \in Sat(F) \mid \forall b \supseteq a\colon\ b \in Sat(\|\xi\|(F)) \rightarrow$
$\exists c \supseteq b\colon\ c \in Sat(\|\xi\| \circ \|\theta\|(F))\}$.

Here it is understood that $\|[\text{every } \xi\theta]\|(F)$ is undefined iff $\|\xi\| \circ \|\theta\|(F)$ is. Applying this rule we can find truth conditions for logical forms (18) and (19): as the reader may verify, the value of $\|(18)\|$ applied to the empty file will have a non-empty satisfaction set if and only if (26) is true, and similarly $Sat(\|(19)\|(\langle\emptyset, \{\emptyset\}\rangle))$ will be non-empty iff (27) holds. A crucial difference between these two readings is their impact on the domain of any given file. While $Dom(\|(18)\|(F))$ will simply be $Dom(F)$ for any F, $Dom(\|(19)\|(F))$ will be $Dom(F) \cup \{x_2\}$, which makes it possible to pick up the discourse referent connected with *a cat* at a later stage in the conversation. And indeed (28) does not violate the Novelty/Familiarity constraint, provided that its first sentence is analyzed along the lines of (19), not along the lines of (18).

(26) $\forall x_1(\text{woman } x_1 \rightarrow \exists x_2(\text{cat } x_2 \wedge \text{catches } x_1 x_2))$.
(27) $\exists x_2(\text{cat } x_2 \wedge \forall x_1(\text{woman } x_1 \rightarrow \text{catches } x_1 x_2))$.
(28) Every_1 woman caught a_2 cat. The_2 cat scratched every_3 woman.

Thus rule (iii) predicts that a definite element can only be anaphorically related to an indefinite occurring within the scope of the quantifier *every* if the definite itself also occurs within that scope. If the first sentence of (28) is analyzed as (18), the universal quantifier blocks a coreferential interpretation of *a cat* and *the cat*, but in (29) we see that an anaphoric link between *a donkey* and *it* is possible since both elements are within the scope of *every* and, as the reader may verify, the file change potential of (30) is defined and leads to the truth conditions of (31).[4]

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

[4] Here $\|\text{who}\|$ may be interpreted as the identity function.

(30) $[_S\text{every}[_{NP}[_{NP-1}\text{ farmer}][_{S'}\text{who}[_S[_{NP}\text{a}_2\text{ donkey}][_S e_1\text{ owns } e_2]]]]\ [_S e_1\text{ beats it}_2]]$

(31) $\forall x_1 x_2((\text{farmer } x_1 \wedge \text{donkey } x_2 \wedge \text{owns } x_1 x_2) \to \text{beats } x_1 x_2)$.

(29) of course is one of Geach's famous "donkey" sentences and its treatment may serve to illustrate another important feature of Heim's system. Since rule (iii) involves a universal quantification over all extensions of the finite assignment a satisfying $\|\xi\|(F)$ and since indefinites in ξ will increase the domain of F, those indefinites will all be interpreted universally, not existentially. For a similar reason indefinites occurring in θ will get an existential interpretation. This explains the chameleontic behaviour of indefinites: if they are not within the scope of any operator they are interpreted existentially, within the "restrictor" ξ of a universal quantifier or the antecedent of an implication they behave universally, but occurring within the "nuclear scope" θ of a universal quantifier or within the consequent of an implication they are existentials again.

1.2.2. Discourse representation theory

The basic ideas of Heim's FCS and Kamp's Discourse Representation Theory (DRT) are very much the same. While in Heim's theory the reader or hearer of a text represents the information that he has obtained by means of a file, DRT lets him keep track of that information with the help of a *Discourse Representation Structure* (a *DRS* or *box* for short) and, just as a file is defined to be true iff some assignment satisfies all the open sentences in it, a box is also defined to be true iff it is satisfied by some assignment. Simple DRSs are much like files, be it that all information is written upon one card only. Thus the DRS corresponding to the first sentence of (12) is (32) and that corresponding to both sentences is (33). The variables written at the top of these boxes are called *discourse referents*, the open sentences underneath are called *conditions*.

(32)

x_1 x_2
woman x_1 *cat* x_2 x_1 *catches* x_2

(33)

x_1 x_2
woman x_1 *cat* x_2 x_1 *catches* x_2 x_2 *scratches* x_1

Boxes such as these are built from the discourses that they represent with the help of a *construction algorithm*. Box (32), for instance, can be obtained from the tree representing the surface structure of the first sentence in (12) by (a) putting this tree in an otherwise empty box and then (b) applying certain rules called *construction principles* until none of these principles is applicable any longer. Box (33) can then be obtained by extending (32) with a tree for the second sentence of the text and applying the construction principles again. A sentence can thus be interpreted as an instruction to update the current box, just as in FCS it can be interpreted as an instruction to change the current file.

Unlike Heim's files however, boxes can also directly represent universal information. (34), for instance, is a box that results from applying the construction algorithm to a tree for the surface structure of (13). It contains only one condition, an implication whose antecedent and consequent are themselves boxes, and it expresses that any way to satisfy the condition in the antecedent box can be extended to a way to satisfy the conditions in the consequent.

(34)
$$\boxed{\boxed{\begin{array}{c} x_1 \\ \hline \textit{woman } x_1 \end{array}} \Rightarrow \boxed{\begin{array}{c} x_2 \\ \hline \textit{cat } x_2 \\ x_1 \textit{ catches } x_2 \end{array}}}$$

It would take us too far to spell out the construction principles that lead to boxes such as these in any detail here (see Kamp and Reyle (1993) for these), but it should be mentioned, firstly, that processing an indefinite noun phrase leads to the creation of a new discourse referent, and, secondly, that anaphoric pronouns must be linked to already existing discourse referents. However, not all existing discourse referents are *accessible* to a pronoun that is being processed at some level of embedding in the DRS. For example, no pronoun may be linked to a discourse referent that exists at some deeper level of embedding, a pronoun in the antecedent of an implication cannot be linked to a discourse referent in the consequent, and so on. With the help of such *accessibility conditions* DRT makes predictions about the possibilities and impossibilities of anaphoric linking that correspond to the predictions that are made by FCS by means of the Novelty/Familiarity condition.

While Discourse Representation Structures are being thought of as psychologically real, in the sense that a language user really creates representations analogous to them while interpreting a text, they also form the language of a logic that can be interpreted on first-order models in a more or less standard way. It is handy to linearize the syntax of this language. The following rules in Backus–Naur Form define the basic constructs, conditions (γ) and boxes (K), for the core part of DRT.

$$\gamma ::= Px \mid x_1 R x_2 \mid x_1 = x_2 \mid \neg K \mid K_1 \vee K_2 \mid K_1 \Rightarrow K_2,$$
$$K ::= [x_1 \cdots x_n \mid \gamma_1, \ldots, \gamma_m].$$

We can write (33) now more concisely as $[x_1 x_2 \mid \textit{woman } x_1,\ \textit{cat } x_2,\ x_1 \textit{ catches } x_2,\ x_2 \textit{ scratches } x_1]$ and (34) as $[\,|[x_1 \mid \textit{woman } x_1] \Rightarrow [x_2 \mid \textit{cat } x_2,\ x_1 \textit{ catches } x_2]]$. These, by the way, are examples of *closed* boxes, boxes containing no free discourse referents;[5] all boxes that result from the construction algorithm are closed.

The dynamic character of DRT does not only reside in the fact that the theory interprets sentences as instructions to change the current discourse representation, it also manifests itself in the formal evaluation of these discourse representations themselves.

[5] For the definition of a *free* discourse referent see Kamp and Reyle (1993).

For a discourse representation structure in its turn can very well be interpreted as an instruction to change the current context, contexts being formalized with the help of finite assignments here. Formally, we shall define the value $\|K\|^M$ of a box K on a first order model $M = \langle D, I\rangle$ (superscripts M will be suppressed) to be a binary relation between finite assignments, the idea being that if $\langle a, b\rangle \in \|K\|$, carrying out the instruction K with a as input may nondeterministically give us b as output.[6] The semantic value $\|\gamma\|$ of a condition γ will simply be a set of finite assignments for the given model. Clauses (i)–(iii) give a compositional definition of the intended meanings;[7] in the last clause we write $a[x_1 \cdots x_n]b$ for '$a \subseteq b$ and $dom(b) = dom(a) \cup \{x_1, \dots, x_n\}$'.

(i) $\|Px\| = \{a \mid x \in dom(a) \;\&\; a(x) \in I(P)\}$,
$\|x_1 R x_2\| = \{a \mid x_1, x_2 \in dom(a) \;\&\; \langle a(x_1), a(x_2)\rangle \in I(R)\}$,
$\|x_1 = x_2\| = \{a \mid x_1, x_2 \in dom(a) \;\&\; a(x_1) = a(x_2)\}$.

(ii) $\|\neg K\| = \{a \mid \neg\exists b\; \langle a, b\rangle \in \|K\|\}$,
$\|K_1 \vee K_2\| = \{a \mid \exists b(\langle a, b\rangle \in \|K_1\| \vee \langle a, b\rangle \in \|K_2\|)\}$,
$\|K_1 \Rightarrow K_2\| = \{a \mid \forall b(\langle a, b\rangle \in \|K_1\| \to \exists c\; \langle b, c\rangle \in \|K_2\|)\}$.

(iii) $\|[x_1 \cdots x_n \mid \gamma_1, \dots, \gamma_m]\| = \{\langle a, b\rangle \mid a[x_1 \cdots x_n]b \;\&\; b \in \|\gamma_1\| \cap \cdots \cap \|\gamma_m\|\}$.

A box K is defined to be *true in* a model M *under* an assignment a iff the domain of a consists of exactly those discourse referents that are free in K and there is an assignment b such that $\langle a, b\rangle \in \|K\|$. The reader may verify that the closed box (33) is true in any model iff (25) is, and that the truth conditions of (34) correspond to those of (26).

The semantic definition given here differs somewhat from the set-up in (Kamp and Reyle, 1993), but is in fact equivalent, as it is easy to show that a closed box is true in our set-up if and only if it is true in Kamp and Reyle's. A slightly different semantics for DRT is given in (Groenendijk and Stokhof, 1991). The Groenendijk and Stokhof semantics is obtained by letting a, b and c range over *total* assignments in the above definition and letting $a[x_1 \cdots x_n]b$ stand for '$a(y) = b(y)$ for all $y \notin \{x_1, \dots, x_n\}$'. In later sections we will refer back to this definition as to the *total* semantics for DRT.

We have seen that DRT does not only predict certain possibilities of anaphoric linking, but, like Heim's FCS, also assigns truth conditions to the discourses that it considers. Both theories, moreover, to a certain extent fit within the framework of semantics that was laid out by Richard Montague in his 'Universal Grammar' (Montague, 1970). Both first replace the constructs of ordinary language by a 'disambiguated language', which is the language of logical forms in Heim's theory and the language of conditions and boxes in Kamp's case. The relation that connects ordinary language and unambiguous language (Montague's R) is given by a set of transformations in Heim's theory and a construction algorithm in Kamp's DRT. In both cases the 'disambiguated language' can be interpreted in a fully compositional way with the help of first-order models and assignments for these models.

---

[6] The first author to describe the dynamic potential of a discourse as a relation between finite variable assignments was Barwise in (Barwise, 1987), a paper which was presented at CSLI in the spring of 1984 and at the Lund meeting on generalized quantifiers in May 1985.

[7] The definition is formally equivalent to the one given in (Kamp and Reyle, 1993) but its form is inspired by the discussion in (Groenendijk and Stokhof, 1991). See especially Definition 26 of that paper.

1.2.3. Dynamic predicate logic

In an attempt to make the Kamp/Heim theory of discourse anaphora look even more like a conventional Montagovian theory, Jeroen Groenendijk and Martin Stokhof have published an alternative formulation called *Dynamic Predicate Logic* (DPL, Groenendijk and Stokhof, 1991), which offers a dynamic interpretation of the formulae of ordinary predicate logic and gives an interesting alternative to the Kamp/Heim approach.

The usual Tarski truth definition for predicate logic provides us with a three-place satisfaction relation $\models$ between models, formulae and assignments and we can identify the meaning of a formula in a model with the set of assignments that satisfy it in that model. But here too, the definition can be generalized so that the meaning of a formula is rendered as a binary relation between (total) assignments. The DPL definition runs as follows (we write $a[x]b$ for '$a(y) = b(y)$ for all $y \neq x$').

(i) $\|R(x_1, \ldots, x_n)\| = \{\langle a, a\rangle \mid \langle a(x_1), \ldots, a(x_n)\rangle \in I(R)\}$,
$\|x_1 = x_2\| = \{\langle a, a\rangle \mid a(x_1) = a(x_2)\}$.

(ii) $\|\neg\varphi\| = \{\langle a, a\rangle \mid \neg\exists b\ \langle a, b\rangle \in \|\varphi\|\}$,
$\|\varphi \vee \psi\| = \{\langle a, a\rangle \mid \exists b(\langle a, b\rangle \in \|\varphi\| \vee \langle a, b\rangle \in \|\psi\|)\}$,
$\|\varphi \to \psi\| = \{\langle a, a\rangle \mid \forall b(\langle a, b\rangle \in \|\varphi\| \to \exists c\ \langle b, c\rangle \in \|\psi\|)\}$,
$\|\varphi \wedge \psi\| = \{\langle a, c\rangle \mid \exists b(\langle a, b\rangle \in \|\varphi\|\ \&\ \langle b, c\rangle \in \|\psi\|)\}$.

(iii) $\|\exists x\varphi\| = \{\langle a, c\rangle \mid \exists b(a[x]b\ \&\ \langle b, c\rangle \in \|\varphi\|)\}$,
$\|\forall x\varphi\| = \{\langle a, a\rangle \mid \forall b(a[x]b \to \exists c\ \langle b, c\rangle \in \|\varphi\|)\}$.

A formula φ is defined to be *true under* an assignment a if $\langle a, b\rangle \in \|\varphi\|$ for some assignment b. Note that $\|\neg\varphi\|$ is given as the set of those $\langle a, a\rangle$ such that φ is not true under a, $\|\varphi \vee \psi\|$ as those $\langle a, a\rangle$ such that either φ or ψ is true under a. But the clause for implication is close to the corresponding DRT clause and conjunction is treated as relational composition. The value of $\exists x\varphi$ is in fact given as the relational composition of $\{\langle a, b\rangle \mid a[x]b\}$ (random assignment to x) and the value of φ; and $\forall x\varphi$ is treated as $\neg\exists x\neg\varphi$. Operators that have a semantics of the form $\{\langle a, a\rangle \mid \cdots\}$ are called *tests*.

By the associativity of relational composition we immediately see that $\exists x\varphi \wedge \psi$ is equivalent to $\exists x(\varphi \wedge \psi)$ in this set-up, *even if x is free in ψ*, and this enables Groenendijk and Stokhof to propose the following straightforward translation of text (12).

(35) $\exists x_1 x_2(\text{woman } x_1 \wedge \text{cat } x_2 \wedge \text{catches } x_1 x_2) \wedge \text{scratches } x_2 x_1$.

The first conjunct of this formula clearly corresponds to the first sentence of the text that is formalized, the second conjunct to the second sentence. But unlike in ordinary predicate logic, (35) is equivalent with (26), and since it is provable that truth conditions in DPL and ordinary logic correspond for closed sentences, the text gets the right truth conditions. In a similar way, since $\exists x\varphi \to \psi$ is equivalent with $\forall x(\varphi \to \psi)$, as the reader may verify, (29) can be rendered as (36), which is equivalent with (37) and hence with (31).

(36) $\forall x_1((\text{farmer } x_1 \wedge \exists x_2(\text{donkey } x_2 \wedge \text{owns } x_1 x_2)) \to \text{beats } x_1 x_2)$.

(37) $\forall x_1(\exists x_2(\text{farmer } x_1 \wedge \text{donkey } x_2 \wedge \text{owns } x_1 x_2) \to \text{beats } x_1 x_2)$.

Thus it is possible to give rather straightforward translations of texts into predicate logical formulae in DPL, while at the same time accounting for the possibility of anaphora between a pronoun and an indefinite in a preceding sentence, or between a pronoun in the consequence of an implication and an indefinite in the antecedent. Anaphoric linking is predicted to be impossible if any test intervenes. This conforms to the predictions that are made by Kamp and Heim's theories.

Extensions of DPL to dynamic theories of generalized quantifiers have been proposed in (Chierchia, 1988; Van Eijck and De Vries, 1992; Kanazawa, 1993b; Van der Does, 1992), and extensions to full type theories have been achieved in the Dynamic Montague Grammar of Groenendijk and Stokhof (1990), and the Compositional DRT of Muskens (1991, 1994, 1995a, 1995b) (see also Section 2.3.3). Extensions such as these raise the issue of systematic strategies of dynamization for existing systems of static semantics, which would somehow operate uniformly, while transforming the traditional semantic theory in systematic ways. For instance, in dynamic accounts of generalized quantifiers, a key role has been played by the fate of the Conservativity and Monotonicity principles that play such a prominent role in the standard theory (cf. Keenan and Westerståhl (1996), this Handbook).

Several variations have been investigated for the basic DPL framework. For instance, Van den Berg (1995) proposes a three-valued partial version, in which new operators appear (cf. also Beaver, 1992; Krahmer, 1995). This system allows for a distinction between 'false' transitions, such as staying in a state where an atomic test has failed, and merely 'inappropriate' ones, such as moving to a different state when testing. A more radical partialization, using analogies with partial functions in Recursion Theory, has been proposed in (Fernando, 1992). This will allow for a natural distinction between re-assignment to an old variable and pristine assignment to a new variable. Versions with still richer accounts of data structures, and thereby of the dynamic function of predicate-logical syntax, may be found in (Visser, 1994; Vermeulen, 1994).

1.2.4. Integrating dynamic predicate logic and update semantics

Natural language involves different dynamic mechanisms. For instance, DRT and DPL highlight changing anaphoric bindings, whereas Veltman's Update Semantics (US), described in Section 1.1 focuses on information flow and epistemic statements about its stages. Obviously, a combination of the two is desirable. There have been some technical obstacles to this endeavor, however, in that the two systems have different flavors of implementation. DPL involves an algebra of binary relations over assignments, and US rather a family of functions operating on sets of valuations. Various proposals have been made for a mathematical unification of the two, but the most sophisticated attempt is surely (Groenendijk, Stokhof and Veltman, 1996). The latter paper takes its empirical point of departure in the linguistic evidence which normally drives modal predicate logic. Here is a typical example. Consider the pair of sentences

(38) A man who might be wearing a blue sweater is walking in the park.
(39) A man is walking in the park. He might be wearing a blue sweater.

The relative clause in the first discourse expresses a property of the man introduced in the main clause: what we learn is that *he* might be wearing a blue sweater. But intuitively, Groenendijk, Stokhof and Veltman argue, this is not the function of the second sentence in the second discourse. The latter rather serves to express the possibility that some discourse individual introduced in the antecedent sentence might be wearing a blue sweater. A combined dynamic semantics will have to account for this. Since these two discourses are equivalent in standard DPL, some essential departure is needed from the latter system, in which antecedent existentials need no longer scope over free variables in succedents. The combined semantics is a more sophisticated follow-up to that of (Van Eijck and Cepparello, 1993), employing so-called 'referent systems' from (Vermeulen, 1994). In particular, the new information states consist of three components, namely: (1) an assignment of variables to 'pegs' (discourse individuals; as in (Landman, 1986)), (2) an assignment of pegs to individuals in some standard domain, (3) a set of possible worlds over that domain (encoding the current range of descriptive uncertainty). Updating will now combine several processes: such as elimination of possibilities and enrichment of assignments. One noticeable feature of this approach is its treatment of the existential quantifier. In DPL, $\exists x$ is essentially a single instruction for performing a random assignment. Thus, in the current setting, it would denote an enrichment for a given state so as to include every possible assignment of objects to (the peg associated with) the variable x. A compound formula $\exists x\varphi$ will then denote the composition of this move with the ordinary update for φ. But this account will yield unintuitive results on a modal statement like $\exists x \diamond Px$: the resulting state may still contain assignments to x denoting objects which cannot have the property P. Therefore, the new proposal is to make $\exists x\varphi$ a syncategorematic operation after all, whose update instruction is as follows: "Take the union of all actions $x := d\,;\varphi$ for all objects d in the domain". This will make an update for $\exists x \diamond Px$ end up with x assigned only to those objects which have P in some available possible world. In this richer setting, one can also review the vast semantic evidence surrounding the usual puzzles of modality and identity in the philosophical literature, and propose a dynamic cut on their solution. (Groenendijk, Stokhof and Veltman (1996) contains further innovations in its discussion of consistency and discourse coherence, which we must forego here.) Whatever technical theory exists for this paradigm is contained in this single reference (but cf. Cepparello, 1995).

1.3. Change of attentional state: Grosz and Sidner

Discourse Representation Theory models the way in which anaphoric elements can pick up accessible discourse referents, it tells us which referents are accessible at any given point of discourse, but it tells us little about the question which referent must be chosen if more than one of them is accessible. There are of course obvious linguistic clues that restrict the range of suitable antecedents for any given anaphoric element, such as the constraint that antecedent and anaphoric element must agree in gender and number, but it is also believed that the structure of discourse itself puts important further constraints on the use of referring expressions.

Thus theories of discourse structure, such as the ones discussed in (Polanyi, 1985; Scha and Polanyi, 1988; Grosz and Sidner, 1986), are a natural complement to the theories discussed in Section 1.2. Since these discourse theories are also good examples of dynamic modeling of natural language phenomena in linguistics, we shall have a closer look at one of them here. Of the theories mentioned, we shall choose Grosz and Sidner's, being the one that is most explicitly dynamic.

(a) E: First you have to remove the flywheel.
(b) A: How do I remove the flywheel?
(c) E: First, loosen the two allen head setscrews holding it to the shaft, then pull it off.
(d) A: OK.
(e) I can only find one screw. Where's the other one?
(f) E: On the hub of the flywheel.
(g) A: That's the one I found. Where's the other one?
(h) E: About ninety degrees around the hub from the first one.
(i) A: I don't understand. I can only find one. Oh wait, yes I think I was on the wrong wheel.
(j) E: Show me what you are doing.
(k) A: I was on the wrong wheel and I can find them both now. DS2
(l) The tool I have is awkward. Is there another tool that I could use instead?
(m) E: Show me the tool you are using.
(n) A: OK.
(o) E: Are you sure you are using the right size key?
(p) A: I'll try some others.
(q) I found an angle I can get at it. DS3
(r) The two screws are loose, but I'm having trouble getting the wheel off.
(s) E: Use the wheelpuller. Do you know how to use it?
(t) A: No.
(u) E: Do you know what it looks like?
(v) A: Yes.
(w) E: Show it to me please.
(x) A: OK.
(y) E: Good. Loosen the screw in the center and place the jaws around the hub of the wheel, then tighten the screw onto the center of the shaft. The wheel should slide off. DS5 DS4 DS1

Fig. 1. A segment of a task oriented dialogue.

Grosz and Sidner distinguish three parts of discourse structure. The first of these, called *linguistic structure,* consists of a segmentation of any given discourse in various *discourse segments.* Experimental data suggest that a segmentation of this kind is present in discourses. Speakers, when asked to segment any given discourse, seem to do so more or less along the same lines. Moreover, the boundaries that are drawn between segments correspond to speech rate differences and differences in pause lengths when the text is read out aloud. There are also certain clue words that signal a discourse boundary. For example the expressions 'in the first place', 'in the second place' and 'anyway' are such clues. Changes in tense and aspect also indicate discourse boundaries.

In Figure 1 a segment of a dialogue between an expert (E) and an apprentice (A) is given and factored into further discourse segments. Each segment comes with a *discourse segment purpose* (DSP). The expert wants the apprentice to remove a flywheel and this,

DSP1:	E intends A to intend to remove the flywheel
DSP2:	A intends E to intend to tell him the location of the other setscrew
DSP3:	A intends E to intend to show him another tool
DSP4:	A intends E to intend to tell him how to get off the wheel
DSP5:	E intends A to know how to use the wheelpuller

Fig. 2. Discourse Segment Purposes connected to task oriented dialogue.

DSP1 dominates DSP2	DSP2 satisfaction-precedes DSP3
DSP1 dominates DSP3	DSP2 satisfaction-precedes DSP4
DSP1 dominates DSP4	DSP3 satisfaction-precedes DSP4
DSP4 dominates DSP5	

Fig. 3. Intentional structure for the task oriented dialogue.

or rather DSP1 in Figure 2, is the purpose of the discourse segment as a whole. The apprentice adopts the intention to remove the fly wheel, but in order to do this must perform certain subactions such as loosening screws and pulling off the wheel. In order to loosen the screws, he must first locate them, and, as it turns out that he can only find one, DSP2 is generated. This intention is connected to a discourse segment (DS2) that consists of utterances (e) to (k).

In the same manner two other discourse segment purposes that are connected to subtasks of the apprentice's task of removing the wheel come up, DSP3 and DSP4, and both intentions give rise to the creation of discourse segments (DS3 and DS4). The last, moreover, invokes DSP5 as a response from the expert, an intention related to DS5.

One discourse segment purpose may *dominate* another in the sense that satisfying the second segment's purpose provides part of the satisfaction of the first segment's purpose. For example, DSP4 in our example dominates DSP5. It may also occur that the satisfaction of one discourse segment purpose must precede another, it is then said to *satisfaction-precede* it. For example, since DSP2 and DSP3 both contribute to loosening the setscrews, DSP4 contributes to pulling off the wheel and, since world-knowledge tells us that the screws must be loosened before the wheel can be pulled off, it can be inferred that DSP2 and DSP3 satisfaction-precede DSP4. The relations of dominance and satisfaction-precedence constitute the second part of discourse structure which is identified by Grosz and Sidner, the *intentional state.* The intentional state connected with the discourse segment in Figure 1 consists of the seven statements given in Figure 3.

The third and last part of discourse structure, *attentional state,* is the part that is most truly dynamic. It consists of a stack of *focus spaces* containing the objects (discourse referents), properties, relations and discourse purposes that are salient at any given moment. Each focus space is connected to a discourse segment and contains its purpose. The closer a focus space is to the top of the stack, the more salient the objects in it are. Anaphoric expressions pick up the referent on the stack that is most salient, so if more than one focus space on the stack would contain, say, a pink elephant, then the definite description *the pink elephant* would refer to the elephant represented in the space that is nearer to the top of the stack.

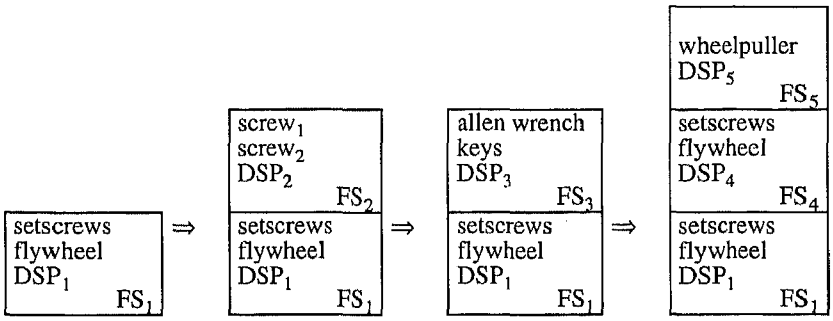


Fig. 4. Focus stack transitions leading up to utterance (y).

Change is brought about by pushing and popping the stack. Entering a discourse segment causes its focus space to be pushed onto the stack and leaving a segment causes its space to be popped. In Figure 4 a series of stacks leading up to the utterance in (y) is given. Note that the theory predicts that in DS5 no reference to the allen wrench is possible: its discourse referent was contained in FS3, which is popped from the stack at the time that DS5 is processed. Note also that the noun phrase *the screw in the center* refers to a screw on the wheelpuller, not to one of the two setscrews. Since the wheelpuller is in the focus space on top of the stack at the moment this noun phrase is uttered, its central screw is chosen as a referent instead of one of the setscrews that are in a lower focus space.

Two similarities strike us when we consider the Grosz and Sidner model of discourse. First there is a strong resemblance between the structure that the model assigns to ordinary discourse and the structure of programs in an imperative language such as PASCAL. The nested discourse segments of Figure 1 remind us of the nested loops and subloops that we find in a typical program. We can also compare the nested structure with the structure of procedures calling subroutines, which may in their turn also call subroutines etc. In this case the stack of focus spaces which constitutes attentional state finds its equivalent in the computer stack.

A second similarity that is to be noted is that between the structure of discourse and the structure of *proofs* in a natural deduction system. The discourse segments in figure 1 here compare to those fragments of a proof that start with the adoption of an assumption and end when that assumption is discharged. The *purpose* of such a segment may perhaps be compared with the conclusion it is intended to establish and there is a clear notion of *satisfaction-precedence* since one such segment may need the conclusion of another. That there is also a natural connection to the concept of a *stack* will be shown in the next section where we shall discuss the semantics of proofs.

1.4. Change of assumptions: Intuitionistic propositional logic in Zeinstra's style

Douglas Hofstadter, in his delightful (Hofstadter, 1980), gives an exposition of natural deduction systems using the idea of *fantasies*. Making an assumption is 'pushing into fantasy', discharging one is 'popping out of fantasy' in his terminology. Hofstadter's system has explicit push and pop operators, '[' and ']' respectively, and a simple derivation looks as follows.

[	push into fantasy,
p	assumption,
$\neg\neg p$	double negation rule,
]	pop out of fantasy.

The next step in this derivation would be an application of detachment (the 'fantasy rule' in Hofstadter's words) to obtain $p \rightarrow \neg\neg p$. It is usual of course to distinguish between the latter (object level) sentence and the (metalevel) derivation given above, which we shall write in linear form as $([p, \neg\neg p])$. For some purposes, however, one might want to have a system in which the distinction between metalevel entailment and object level implication is not made. Consider the following pair of texts.

(A) Suppose $x > 0$. Then $x + y > 0$.
(B) If $x > 0$, then $x + y > 0$.

The assertive production of (A) can be described as follows. First an assumption is introduced. Then a conclusion is drawn from it (possibly in combination with information derived from preceding text). Finally there is the hidden act of cancelling the assumption. The assertion of (B), on the other hand, on the classical account does not involve introducing, cancelling, etc. It is simply an utterance with assertive force of a sentence. What, then, are we to do with the strong intuition that (A) and (B) are 'assertively equivalent'?

The intuition that (A) and (B) should be treated on a par motivated (Zeinstra, 1990) to give a semantics for a simple propositional system which bases itself upon Hofstadter, has explicit push and pop operators, but retains the equivalence. The assertive utterance of a sentence is viewed – quite in the spirit of the more general dynamic program – as consisting of a sequence of all kinds of acts, and an utterance of *if* is taken as being just a variant of an utterance of *suppose*. Before we give an exposition of Zeinstra's logic, let us rehearse the Kripke semantics for the $\{\bot, \wedge, \rightarrow\}$ fragment of intuitionistic propositional logic (IPL$[\bot, \wedge, \rightarrow]$), as Zeinstra's system can be viewed as an extension of the latter. A *model* K for this logic is a triple $\langle W, \leqslant, V\rangle$ such that – in the present set-up – W, the set of worlds, contains the *absurd* world T; the relation $\leqslant$ is a reflexive and transitive ordering on W, such that $w \leqslant T$ for all $w \in W$; and V is a function sending propositional letters to subsets of W such that (a) $w \in V(p)$ implies $w' \in V(p)$ if $w \leqslant w'$ and (b) $T \in V(p)$ for each propositional letter p. The relation $w \models_K \varphi$ (φ is *true on* a model $K = \langle W, \leqslant, V\rangle$ *in* a world $w \in W$) is defined inductively as follows (we suppress subscripts K).

(i) $w \models p$ iff $w \in V(p)$, for propositional letters p,
(ii) $w \models \bot$ iff $w = T$,
(iii) $w \models \varphi \wedge \psi$ iff $w \models \varphi$ and $w \models \psi$,
(iv) $w \models \varphi \rightarrow \psi$ iff $\forall w' \geqslant w$: $w' \models \varphi \Rightarrow w' \models \psi$.

The language of Zeinstra's logic is given by the following Backus–Naur Form.

$$\varphi ::= p \mid \bot \mid] \mid [\mid (\varphi_1, \varphi_2) \mid \varphi_1 ; \varphi_2 .$$

Here p stands for arbitrary propositional letters, $\bot$ is the falsum,] and [are the pop and push operators we have met before, (φ, ψ) is to be read as φ, *hence* ψ, and the semicolon is our sign for conjunction. We prefer the latter over the more conventional $\wedge$ since its semantics will be relational composition as in Groenendijk and Stokhof's system, not intersection or meet as in standard logic. We usually write $\varphi\psi$ for $\varphi ; \psi$. Since the negation $\neg\varphi$ of a formula φ can be considered to be an abbreviation of $([\varphi, \bot])$ the toy derivation in Hofstadter's system given above can now indeed be represented as $([p, \neg\neg p])$ or $([p, ([([p, \bot]), \bot])])$. The latter are examples of formulae in which the push and pop brackets are well-balanced, but in general no such condition need be imposed.

Kripke's semantics for IPL provides us with good candidates for the explication of Hofstadter's fantasies: fantasies are worlds. Since fantasies can be nested, we need stacks (sequences) of worlds for our semantics. For stacks $\sigma = \langle w_1, \ldots, w_n \rangle$ we demand that $w_i \leqslant w_{i+1}$, for all $i < n$, i.e. worlds that are higher in a stack, are also higher in the underlying model. We write $\mathrm{Last}(\langle w_1, \ldots, w_n \rangle)$ to refer to w_n and we write $\sigma \leqslant_1 \tau$ if $\sigma = \langle w_1, \ldots, w_n \rangle$ and $\tau = \langle w_1, \ldots, w_n, w \rangle$, i.e. if τ is a possible result of pushing the stack σ. The *meaning* $\|\varphi\|$ of a formula φ in Zeinstra's language is a binary relation between stacks of worlds in a Kripke model K, defined with the help of the following clauses.

(i) $\sigma\|p\|\tau$ iff $\sigma = \tau$ and $\mathrm{Last}(\sigma) \in V(p)$, for propositional p,
(ii) $\sigma\|\bot\|\tau$ iff $\sigma = \tau$ and $\mathrm{Last}(\sigma) = T$,
(iii) $\sigma\|\,[\,\|\tau$ iff $\sigma \leqslant_1 \tau$,
(iv) $\sigma\|\,]\,\|\tau$ iff $\tau \leqslant_1 \sigma$,
(v) $\sigma\|(\varphi, \psi)\|\tau$ iff $\exists\rho(\sigma\|\varphi\|\rho \;\&\; \rho\|\psi\|\tau)$ and $\forall\rho(\sigma\|\varphi\|\rho \Rightarrow \exists\nu\rho\|\psi\|\nu)$,
(vi) $\sigma\|\varphi;\psi\|\tau$ iff $\exists\rho(\sigma\|\varphi\|\rho \;\&\; \rho\|\psi\|\tau)$.

Truth is defined just as it was done in Discourse Representation Theory or in Dynamic Predicate Logic: in terms of the domain of the given relation. Formally, we write $K, \sigma \models \varphi$ if $\sigma\|\varphi\|\tau$ for some stack τ.

As an example of how this semantics works consider the formula $([p, q])$. We have:

$$\begin{aligned}
\sigma\|([p,q])\|\tau \quad &\text{iff} \quad \exists\rho(\sigma\|[p\|\rho \;\&\; \rho\|q]\|\tau) \text{ and } \forall\rho(\sigma\|[p\|\rho \Rightarrow \exists\nu\rho\|q]\|\nu),\\
&\text{iff} \quad \sigma = \tau \text{ and } \forall\rho(\sigma\|[p\|\rho \Rightarrow \exists\nu\rho\|q]\|\nu),\\
&\text{iff} \quad \sigma = \tau \text{ and } \forall\rho \geqslant_1 \sigma(\rho \models p \Rightarrow \rho \models q),\\
&\text{iff} \quad \sigma = \tau \text{ and } \forall w \geqslant \mathrm{Last}(\sigma)(w \models p \Rightarrow w \models q),\\
&\text{iff} \quad \sigma = \tau \text{ and } \mathrm{Last}(\sigma) \models p \to q.
\end{aligned}$$

The first equivalence is an instantiation of clause (v), the second follows since the required ρ in $\exists\rho(\sigma\|[p\|\rho \;\&\; \rho\|q]\|\tau)$ can simply be σ extended with T, and the last two equivalences are simple consequences of the definitions. It may amuse the reader to try her hand at $([p[q, r]s])$.

The equivalence given above shows a connection between the formula $([p, q])$ in Zeinstra's language and the implication $p \to q$ in IPL and indeed there is a more systematic connection between the two logics. Let $(\cdot)^\circ$ be the translation of IPL$[\bot, \wedge, \rightarrow]$ into Zeinstra's language such that $(p)^\circ = p$ for all propositional p, $(\bot)^\circ = \bot$, $(\varphi \wedge \psi)^\circ := \varphi^\circ; \psi^\circ$, and $(\varphi \rightarrow \psi)^\circ = ([\varphi^\circ, \psi^\circ])$. Then $K, \langle w \rangle \models \varphi^\circ$ iff $w \models_K \varphi$, for all formulae in IPL$[\bot, \wedge, \rightarrow]$, as the reader may care to verify. But a converse holds as well since Zeinstra has shown that for all formulae φ in her language such that the pop and push operators] and [are well-balanced in φ there is an IPL$[\bot, \wedge, \rightarrow]$ formula φ' such that $K, \langle w \rangle \models \varphi$ iff $w \models_K \varphi'$ for any K and w.

In essence then, the logic contains a fragment of well-balanced formulae which is equivalent to IPL$[\bot, \wedge, \rightarrow]$ and in which there is no longer a distinction between implication and entailment. But the logic is a true extension of that fragment, as it also gives a semantics for formulae that are not well-balanced. The latter correspond to almost arbitrary segments of proofs in which assumptions may be made without discharging them and where even pops may occur without the corresponding pushes.

1.5. *Change of beliefs: Gärdenfors' theory of belief revision*

Let us return to the Stalnaker–Karttunen theory of presuppositions temporarily and ask ourselves what will happen when a speaker utters a sentence A that carries a presupposition B which the hearer in fact does not take for granted. In many cases no problem will arise at all, because the very utterance of A will tell the hearer that B is presupposed by the speaker and the hearer may tacitly add B to his stock of beliefs or, in any case, he may pretend to do so. This process, which is called *accommodation* in (Lewis, 1979), allows a presupposition to spring into existence if it was not there when the sentence requiring it was uttered. But what if the required presupposition cannot be accommodated because it is not consistent with the hearer's existing set of beliefs? Karttunen (1973) remarks that this problem is reminiscent of a problem that arises in connection with conditionals. An influential theory about the evaluation of the latter, first proposed by Ramsey (1929), and later formalized in (Stalnaker, 1968) and (Lewis, 1973), wants you to hypothetically add the antecedent of a conditional to your stock of beliefs. If it turns out that the consequent of the conditional follows from this new set of beliefs, you may conclude that the conditional itself is true. Again the problem arises how consistency can be maintained. Disbelieving the antecedent of a counterfactual should not necessarily lead to acceptance of the counterfactual itself, simply because adding the antecedent to your stock of beliefs would lead to inconsistency. This means that some beliefs must be given up (hypothetically) before the (hypothetical) addition can take place. But not all ways to discard beliefs are equally rational; for instance, you do not want to end up with a proper subset of some set of beliefs that is consistent with the antecedent.

Of course the question how beliefs can be given up and how opinions can be revised rationally in the light of new evidence is a general one. The problem is central to an interesting research line that was initiated by Peter Gärdenfors and that is exemplified by papers such as (Makinson, 1985; Gärdenfors, 1988; Gärdenfors and Makinson, 1988; Rott, 1992). Suppose we have a set of beliefs K, which we may for present purposes take to be a deductively closed theory of predicate logic, and a new insight φ (a predicate logical sentence) and suppose we revise K in the light of φ, obtaining a new theory $K^*\varphi$.

What are the properties that $K^*\varphi$ should conform to? Gärdenfors gives eight postulates. Writing $K+\varphi$ for $\{\psi \mid K, \varphi \vdash \psi\}$ (the *expansion* of K by φ), he demands the following.

(*1) $K^*\varphi$ is deductively closed,
(*2) $\varphi \in K^*\varphi$,
(*3) $K^*\varphi \subseteq K + \varphi$,
(*4) If $K + \varphi$ is consistent then $K + \varphi \subseteq K^*\varphi$,
(*5) $K^*\varphi$ is consistent if $\{\varphi\}$ is consistent,
(*6) If φ is equivalent with ψ then $K^*\varphi = K^*\psi$,
(*7) $K^*\varphi \wedge \psi \subseteq (K^*\varphi) + \psi$,
(*8) If $(K^*\varphi) + \psi$ is consistent then $(K^*\varphi) + \psi \subseteq K^*\varphi \wedge \psi$.

We can think of the first of these postulates as being merely a matter of technical convenience: it allows us to formulate principles about $K^*\varphi$ instead of principles about its deductive closure. Postulates (*2)–(*6) seem reasonable in view of the intended meaning of $K^*\varphi$: (*2) states that after revising K in the light of φ we should come to believe φ, (*3) and (*4) that revising in the light of φ is just adding φ to one's set of beliefs, if this can be done consistently, (*5) is the requirement that consistency should be maintained if at all possible and (*6) demands that $K^*\varphi$ depends on the content rather than on the form of φ. Principles (*7) and (*8) are supplementary postulates about iterated revisions, the idea being that $K^*\varphi \wedge \psi$ ought to be the same as the expansion of $K^*\varphi$ by ψ, as long as ψ does not contradict the beliefs in $K^*\varphi$.

Gärdenfors also considers the process of giving up a belief, i.e. subtracting some belief φ from a set of beliefs K. The result $K \dot{-} \varphi$, the *contraction* of K with respect to φ, should conform to the following axioms.

($\dot{-}$1) $K \dot{-} \varphi$ is deductively closed,
($\dot{-}$2) $K \dot{-} \varphi \subseteq K$,
($\dot{-}$3) If $\varphi \notin K$ then $K \dot{-} \varphi = K$,
($\dot{-}$4) If $\varphi \in K \dot{-} \varphi$ then $\vdash \varphi$,
($\dot{-}$5) $K \subseteq (K \dot{-} \varphi) + \varphi$,
($\dot{-}$6) If φ is equivalent with ψ then $K \dot{-} \varphi = K \dot{-} \psi$,
($\dot{-}$7) $(K \dot{-} \varphi) \cap (K \dot{-} \psi) \subseteq K \dot{-} (\varphi \wedge \psi)$,
($\dot{-}$8) If $\varphi \notin K \dot{-} (\varphi \wedge \psi)$ then $K \dot{-} (\varphi \wedge \psi) \subseteq K \dot{-} \varphi$.

Again, motivations for the *basic* postulates ($\dot{-}$1)–($\dot{-}$6) follow readily from the intended meaning of $\dot{-}$. For a motivation of the (*supplementary*) postulates ($\dot{-}$ 7) and ($\dot{-}$ 8) see (Gärdenfors, 1988).

The operations $*$ and $\dot{-}$ are not unrelated, as revising in the light of φ can in fact be thought to consist of two operations, namely first contracting with respect to the negation of φ and then adding φ itself. Conversely, we may define the contraction with respect to φ as the set of those of our original beliefs that would still hold after a revision in the light of the negation of φ.

(Def L) $K^*\varphi := (K \dot{-} \neg\varphi) + \varphi$ (Levi Identity).
(Def H) $K \dot{-} \varphi := K \cap K^*\neg\varphi$ (Harper Identity).

Write L($\dot{-}$) for the revision function obtained from $\dot{-}$ by the Levi identity and H(*) for the contraction function obtained from * by the Harper identity. The following theorem (see Gärdenfors, 1988) connects revisions and contractions and states the duality of L and H.

THEOREM 1.1.

(i) *If* * *satisfies* (*1)–(*8) *then* H(*) *satisfies* ($\dot{-}$1)–($\dot{-}$8),
(ii) *If* $\dot{-}$ *satisfies* ($\dot{-}$1)–($\dot{-}$8) *then* L($\dot{-}$) *satisfies* (*1)–(*8),
(iii) *If* * *satisfies* (*1)–(*6) *then* L(H(*)) = *,
(iv) *If* $\dot{-}$ *satisfies* ($\dot{-}$1)–($\dot{-}$6) *then* H(L($\dot{-}$)) = $\dot{-}$.

In fact this theorem can be generalized to some degree since the number 8 can be replaced uniformly by 6 or 7 in each of the first two clauses. This is satisfactory as in both sets of postulates the first six seem to give some very general properties of the concept under investigation, while the last two more in particular pertain to conjunctions.

It is one thing to give a set of postulates for a concept and another to give structures which satisfy them. One need not go as far as Russell, who said that the method of postulation has 'the advantages of theft over honest toil' (the quote is from Makinson, 1985), to feel that an abstract set of postulates should be complemented with more explicit constructions if at all possible. But there are many ways to obtain constructs satisfying the Gärdenfors postulates and we shall consider three of them. The first construction – from Alchourrón, Gärdenfors and Makinson (1985) – takes $K\dot{-}\varphi$ to be the intersection of some maximal subsets of K that fail to imply φ. More precisely, let $K\bot\varphi$ (*K less φ*) be the set of all such maximal subsets, i.e. the set $\{X \subseteq K \mid X \nvdash \varphi \ \& \ \forall Y(X \subseteq Y \subseteq K \ \& \ Y \nvdash \varphi \Rightarrow X = Y)\}$, and let γ be a function such that $\gamma(K\bot\varphi) \neq \emptyset$, $\gamma(K\bot\varphi) \subseteq K\bot\varphi$ if $K\bot\varphi \neq \emptyset$ and $\gamma(K\bot\varphi) = \{K\}$ otherwise. Then the *partial meet contraction* $K\dot{-}\varphi$ can be defined as $\bigcap\gamma(K\bot\varphi)$. The following representation theorem holds.

THEOREM 1.2. *The operation of partial meet contraction satisfies* ($\dot{-}$1)–($\dot{-}$6). *Conversely, any operation that satisfies* ($\dot{-}$1)–($\dot{-}$6) *is itself a partial meet contraction operation.*

The theorem can be extended to a representation theorem for ($\dot{-}$1)–($\dot{-}$8) by placing extra conditions on γ. Of course, the Levi identity also allows us to obtain an operation of *partial meet revision* from the operation of partial meet contraction. This operation then satisfies (*1)–(*6), or (*1)–(*8) if extra conditions are added.

Another way to construct a contraction function makes use of the notion of *epistemic entrenchment*. Giving up some beliefs will have more drastic consequences as giving up others and consequently some beliefs have preferential status over others. Write $\varphi \leqslant \psi$ (*ψ is at least as epistemologically entrenched as φ*) if φ and ψ are both logical truths (and hence cannot be given up), or if φ is not believed at all, or if a need to give up one of φ or ψ will lead to discarding φ (or both). It seems reasonable to demand the following.

(EE1) If $\varphi \leqslant \psi$ and $\psi \leqslant \chi$, then $\varphi \leqslant \chi$,
(EE2) If $\varphi \vdash \psi$ then $\varphi \leqslant \psi$,

(EE3) $\varphi \leqslant \varphi \wedge \psi$ or $\psi \leqslant \varphi \wedge \psi$,
(EE4) If K is consistent then $\varphi \notin K$ iff $\varphi \leqslant \psi$ for all ψ,
(EE5) If $\varphi \leqslant \psi$ for all φ then $\vdash \psi$.

Transitivity of $\leqslant$ (EE1) must be required if $\leqslant$ is to be an ordering relation. If φ entails ψ, then ψ cannot be given up without giving up φ, whence (EE2). Since a choice between giving up φ or $\varphi \wedge \psi$ is in fact a choice between giving up φ or ψ, (EE3) in fact states that $\varphi \leqslant \psi$ or $\psi \leqslant \varphi$, a natural requirement. (EE4) identifies the sentences that are not believed with those that are least entrenched and the last requirement says that only logically valid sentences are maximal in $\leqslant$, i.e. that anything can be given up, logical truths excepted.

Given a contraction relation we can define a relation of epistemic entrenchment with the help of (C) below. Conversely, supposing that an entrenchment relation $\leqslant$ is given, then (E) defines a contraction relation in terms of it. ('$\varphi < \psi$' is defined as '$\varphi \leqslant \psi$ and not $\psi \leqslant \varphi$'.)

(C) $\varphi \leqslant \psi$ iff $\varphi \notin K \dot{-} (\varphi \wedge \psi)$ or $\vdash \varphi \wedge \psi$.
(E) $K \dot{-} \varphi = K \cap \{\psi \mid \varphi < \varphi \vee \psi\}$ if $\nvdash \varphi$,
$= K$ otherwise.

Write $C(\leqslant)$ for the contraction function obtained from $\leqslant$ by (C) and $E(\dot{-})$ for the relation of epistemic entrenchment obtained from $\dot{-}$ by def (E). The following representation theorem is proved in (Gärdenfors and Makinson, 1988).

THEOREM 1.3.
(i) *If* $\leqslant$ *satisfies* (EE1)–(EE5) *then* $C(\leqslant)$ *satisfies* $(\dot{-}1)$–$(\dot{-}8)$.
(ii) *If* $\dot{-}$ *satisfies* $(\dot{-}1)$–$(\dot{-}8)$ *then* $E(\dot{-})$ *satisfies* (EE1)–(EE5).
(iii) *If* $\leqslant$ *satisfies* (EE1)–(EE5) *then* $E(C(\leqslant)) = \leqslant$.
(iv) *If* $\dot{-}$ *satisfies* $(\dot{-}1)$–$(\dot{-}8)$ *then* $C(E(\dot{-})) = \dot{-}$.

A third way to construct operations satisfying the Gärdenfors postulates that we want to mention is the oldest of them all and in fact precedes the formulation of the postulates themselves. Gärdenfors (1988) notes that the *probability functions* that we find in the Bayesian tradition provide us with the necessary material to construct such operations. For example, the conditional probability functions axiomatized in (Popper, 1959) immediately give us revision functions satisfying (*1)–(*8) above and again a representation theorem can be proved. For more details and a careful discussion see (Gärdenfors, 1988).

2. Logical observations

2.1. *General dynamic logic*

Dynamic semantics provides a fresh look at most aspects of logical theory. In this section we shall use the paradigm of Dynamic Logic (Pratt, 1976; Harel, 1984; Goldblatt, 1987; Harel and Kozen, 1994), broadly conceived, and twisted to suit our purposes wherever this is needed, for bringing out some of these. To appreciate what follows, there is a useful analogy with Generalized Quantifier Theory (cf. Keenan and Westerståhl, 1996; this Handbook): Dynamic Logic provides a broad logical space for dynamic operators and inference and this logical space may be contrasted fruitfully with the empirical space of what we find realized in natural language and human cognition. But the most fruitful analogy is the earlier one of the Introduction. Dynamic semantics has many counterparts in computer science, for obvious reasons. There are striking similarities between variable binding mechanisms in programming languages and what is currently being proposed for natural language. Similar observations may be made about Artificial Intelligence, witness the parallels in the study of default reasoning between Veltman (1991), Boutilier (1993), Boutilier and Goldszmidt (1993), and Van Benthem, Van Eijck and Frolova (1993). For our current purposes, we wish to emphasize the richer process theory available in the computational literature. We hope that, eventually, natural language semantics will come up with a similar refined view of its dynamic structures.

2.1.1. Dynamic logic

The expressions of Propositional Dynamic Logic (PDL) are divided in two categories: the category of *formulae*, which form the static part of the language, and the category of *programs*, the truly dynamic part. But formulae can be constructed from programs and vice versa, so that there is an active interplay between the two parts. The following Backus–Naur Form defines formulae (φ) and programs (π) from basic propositional letters (p) and atomic programs (α).

$$\varphi ::= p \mid \bot \mid \varphi_1 \rightarrow \varphi_2 \mid [\pi]\varphi,$$
$$\pi ::= \alpha \mid \varphi? \mid \pi_1;\pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^*.$$

The intuitive meaning of $[\pi]\varphi$ is the statement that φ will be true after any successful execution of π. A *test* program $\varphi?$ tests whether φ is true, continues if it is, but fails if it is not. The *sequence* $\pi_1;\pi_2$ is an instruction to do π_1 and then π_2. The *choice* program $\pi_1 \cup \pi_2$ can be executed by either carrying out π_1 or by doing π_2 and the *iteration* π^* is an instruction to do π any number ($\geqslant 0$) of times.

The last two constructs introduce nondeterminism into the language. An execution of $p;p;q$ will count as an execution of $(p \cup q)^*$, but an execution of q alone, or of any finite sequence of p's and q's, will do as well. Programs are regular expressions and an execution of any sequence in the denotation of such an expression will count as an execution of the program itself.

The semantics of PDL is obtained by considering poly-modal Kripke models (also known as *labeled transition systems*) $\langle S, \{R_\alpha \mid \alpha \in \mathrm{AT}\}, V\rangle$, consisting of a set of abstract *program states* S, a set of binary relations R_α over S, indexed by the set of atomic programs AT, and a *valuation* function V which assigns a subset of S to each propositional letter in the language. In general, the meaning of a formula is identified with the set of all states where the formula is true, the meaning of a program with the set of pairs $\langle a, b\rangle$ such that the program, if started in state a, may end up in state b. Writing $R \circ R'$ for the relational composition of R and R' and $(R)^*$ for the reflexive transitive closure of R, we can define the meaning $\|\varphi\|^M$ of a formula φ and the meaning $\|\pi\|^M$ of a program π with respect to a given model $M = \langle S, \{R_\alpha \mid \alpha \in \mathrm{AT}\}, V\rangle$ as follows.

(i) $\|p\| = V(p)$,

(ii) $\|\bot\| = \emptyset$,
(iii) $\|\varphi_1 \to \varphi_2\| = (S - \|\varphi_1\|) \cup \|\varphi_2\|$,
(iv) $\|[\pi]\varphi\| = \{a \mid \forall b(\langle a, b\rangle \in \|\pi\| \to b \in \|\varphi\|)\}$,
(v) $\|\alpha\| = R_\alpha$,
(vi) $\|\varphi?\| = \{\langle a, a\rangle \mid a \in \|\varphi\|\}$,
(vii) $\|\pi_1; \pi_2\| = \|\pi_1\| \circ \|\pi_2\|$,
(viii) $\|\pi_1 \cup \pi_2\| = \|\pi_1\| \cup \|\pi_2\|$,
(ix) $\|\pi^*\| = (\|\pi\|)^*$.

We see that $[\pi]\varphi$ is in fact interpreted as a modal statement ('in all π-successors φ') with the modal accessibility relation given by the denotation of π and we may define a dual modality by letting $\langle\pi\rangle\varphi$ be an abbreviation of $\neg[\pi]\neg\varphi$. This new statement will then have the meaning that it is possible that φ will hold after execution of π. Abbreviations will also give us a host of constructs that are familiar from the usual imperative programming languages. For example, **while** φ **do** π **od** can be viewed as an abbreviation of $(\varphi?; \pi)^*; \neg\varphi?$; a little reflection will show that the latter has the intended input/output behaviour. Correctness statements (in Hoare's sense) about such programs can be formalized too; for example $\{\varphi\}\pi\{\psi\}$, the assertion that in any state where φ holds any successful execution of π will lead to a state where ψ holds, can be taken to be an abbreviation of $\varphi \to [\pi]\psi$.

A formula φ is said to be *universally valid* if $\|\varphi\| = S$ for each model $\langle S, \{R_\alpha \mid \alpha \in \text{AT}\}, V\rangle$. Segerberg (1982) shows that this notion is axiomatizable by means of the following seven axiom schemes and two rules of inference.

(A1)	all instances of tautologies of the propositional calculus	
(A2)	$[\pi](\varphi \to \psi) \to ([\pi]\varphi \to [\pi]\psi)$	(Distribution)
(A3)	$[\varphi?]\psi \leftrightarrow (\varphi \to \psi)$	(Test axiom)
(A4)	$[\pi_1; \pi_2]\psi \leftrightarrow [\pi_1][\pi_2]\psi$	(Sequence axiom)
(A5)	$[\pi_1 \cup \pi_2]\psi \leftrightarrow ([\pi_1]\psi \wedge [\pi_2]\psi)$	(Choice axiom)
(A6)	$[\pi^*]\psi \leftrightarrow (\psi \wedge [\pi][\pi^*]\psi)$	(Iteration axiom)
(A7)	$(\varphi \wedge [\pi^*](\varphi \to [\pi]\varphi)) \to [\pi^*]\varphi$	(Induction axiom)
(MP)	from φ and $\varphi \to \psi$ to infer ψ	(Modus Ponens)
(N)	from φ to infer $[\pi]\varphi$	(Necessitation)

As a simple illustration we give a derivation of one of Hoare's rules of *Composition*, the rule that $\{\varphi\}\pi_1; \pi_2\{\chi\}$ can be inferred from $\{\varphi\}\pi_1\{\psi\}$ and $\{\psi\}\pi_2\{\chi\}$.

1. $\varphi \to [\pi_1]\psi$,
2. $\psi \to [\pi_2]\chi$,
3. $[\pi_1](\psi \to [\pi_2]\chi)$, necessitation, 2,
4. $[\pi_1]\psi \to [\pi_1][\pi_2]\chi$, distribution, 3,
5. $\varphi \to [\pi_1][\pi_2]\chi$, propositional logic, 1, 4,
6. $\varphi \to [\pi_1; \pi_2]\chi$, sequence axiom, 5.

We invite the reader to show that $\{\varphi\}$**while** ψ **do** π **od**$\{\varphi \wedge \neg\psi\}$ can be derived from $\{\varphi \wedge \psi\}\pi\{\varphi\}$.

The system of *Quantificational Dynamic Logic* (QDL) can be obtained from PDL by specifying the structure of atomic formulae and atomic programs. In particular, the atomic formulae of standard predicate logic will be atomic formulae of the new logic and *assignment statements* of the forms $x :=?$ (random assignment) and $x := t$ are its atomic programs. The following Backus–Naur Form gives a precise syntax.

$$\varphi ::= R(t_1, \ldots, t_n) \mid t_1 = t_2 \mid \bot \mid \varphi_1 \to \varphi_2 \mid [\pi]\varphi,$$
$$\pi ::= x := ? \mid x := t \mid \varphi? \mid \pi_1 ; \pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^*.$$

The idea here is that $x := ?$ sets x to an arbitrary new value and that $x := t$ sets x to the current value of t. The semantics of this logic is given relative to ordinary first-order models $M = \langle D, I \rangle$ with the set of states S now being played by the set of all M-assignments, i.e. the set of all (total) functions from the variables in the language to D. Letting $\|t\|^a$ (the value of a term t under an assignment a) be defined as usual, we can define $\|\varphi\|^M$ and $\|\pi\|^M$ by taking the clauses for the PDL semantics given above, but replacing those for atomic formulae and programs by the following. (Here $a[x]b$ is to mean that $a(y) = b(y)$ if $x \neq y$.)

$$\|R(t_1, \ldots, t_n)\| = \{a \mid \langle \|t_1\|^a, \ldots, \|t_n\|^a \rangle \in I(R)\},$$
$$\|t_1 = t_2\| = \{a \mid \|t_1\|^a = \|t_2\|^a\},$$
$$\|x := ?\| = \{\langle a, b \rangle \mid a[x]b\},$$
$$\|x := t\| = \{\langle a, b \rangle \mid a[x]b \ \&\ b(x) = \|t\|^a\}.$$

We say that ψ *follows from* φ, $\varphi \models_{\mathrm{QDL}} \psi$, iff $\|\varphi\|^M \subseteq \|\psi\|^M$ for every model M. The logic thus obtained is a truly *quantificational* logic since $\forall x\varphi$ can taken to be an abbreviation of $[x := ?]\varphi$ and $\exists x\varphi$ of $\langle x := ?\rangle\varphi$. Note also that $[x := t]\varphi$ and $\langle x := t\rangle\varphi$ are both equivalent with the result of substitution of t for x in φ. However, the logic really extends first-order logic. Consider $[x := ?]\langle y := 0; (y := Sy)^*\rangle x = y$ in the language of Peano Arithmetic. Together with the usual first-order Peano axioms this sentence will characterize the natural numbers, a feat which first-order logic cannot perform.

The price that must be paid is non-axiomatizability of the system, of course. However, there is a simple proof system which is complete relative to structures containing a copy of the natural numbers (see Harel, 1984). Note that the iteration operator * is the sole culprit for non-axiomatizability: the Segerberg axioms (A3)–(A5) plus the equivalences between $[x :=?]\varphi$ and $\forall x\varphi$ and $[x := t]\varphi$ and $[t/x]\varphi$ provide an easy method to find a predicate logical equivalent for any formula $[\pi]\varphi$ not containing the star (see also the "weakest precondition" calculi in Section 2.3.4).

The interest of QDL for natural language semantics derives partly from the fact that the DRT and DPL systems that were considered in Section 1.2 can easily be shown to be fragments of the star free part of this logic. For example, we can translate DRT into QDL in the following way.

$$(\varphi)^\dagger = \varphi \quad \text{if } \varphi \text{ is atomic},$$
$$(\neg K)^\dagger = [K^\dagger]\bot,$$
$$(K_1 \vee K_2)^\dagger = \langle K_2^\dagger\rangle\top \vee \langle K_2^\dagger\rangle\top,$$

$$(K_1 \Rightarrow K_2)^\dagger = [K_1^\dagger]\langle K_2^\dagger\rangle\top,$$

$$([x_1,\ldots,x_n \mid \varphi_1,\ldots,\varphi_m])^\dagger = x_1 := ?\,;\ldots;x_n := ?\,;\varphi_1^\dagger?\,;\ldots;\varphi_m^\dagger?.$$

If we let DRT be interpreted by means of its total semantics (see Section 1.2.2), we have that $\|\delta\|^{\mathrm{DRT}} = \|\delta^\dagger\|^{\mathrm{QDL}}$ for any condition or DRS δ. If both DRT and QDL are provided with a semantics based on partial assignments an embedding is possible as well – see (Fernando, 1992). The reader will have no difficulty in defining a translation function from DPL to QDL either (see also Groenendijk and Stokhof, 1991).

2.1.2. Dynamization of classical systems

Systems of dynamic semantics may often be derived from static predecessors. For this purpose one has to identify parameters of change in classical systems, and then design dynamic logics exploiting these. For instance, consider Tarski's basic truth definition for a formula φ in a model $M = \langle D, I\rangle$ under some variable assignment a. Its atomic clause involves a static test whether some fact obtains. But intuitively, the clause for an existential quantifier $\exists x$ involves shifting an assignment value for x until some verifying object has been found. A system like DPL makes the latter process explicit, by assigning to each formula a binary relation consisting of those transitions between assignments which result in its successful verification. Entirely analogously, other components of the truth definition admit of such shifts too. For instance, shifting interpretation functions I are involved in questions (cf. Groenendijk and Stokhof, 1984) and ambiguity (Van Deemter, 1991), and shifting of individual domains D occurs with ranges for generalized quantifiers across sentences (Westerståhl, 1984).

In addition to these 'Tarskian Variations' for extensional logics (Van Benthem, 1991b), there are also 'Kripkean Variations' for intensional logics. Consider, e.g., the best-known classical information-oriented model structures, namely Kripke models for intuitionistic logic. Here, worlds stand for information states, ordered by a relation of growth $\subseteq$, which are traversed by a cognitive agent. Intuitively, intuitionistic formulas refer to transitions in this information pattern (cf. Troelstra and Van Dalen, 1988). For example, to see that $\neg\varphi$ holds, one has to inspect all possible extensions of the current state for absence of φ. Van Benthem (1991a) makes this dynamics into an explicit part of the logic, by creating a system of cognitive transitions, such as updates taking us to some minimal extension where a certain proposition has become true. While intuitionistic negation, which is expressible as $\lambda P\lambda x.\forall y(x \subseteq y \to \neg Py)$, takes us from sets of worlds to sets of worlds, Van Benthem is also interested in functions which take us from sets of worlds to binary relations between worlds, such as for example:

$\lambda P.\lambda xy.x \subseteq y \wedge Py$	(*loose updating*)
$\lambda P.\lambda xy.x \subseteq y \wedge Py \wedge \neg\exists z(x \subseteq z \subset y \wedge Pz)$	(*strict updating*)
$\lambda P.\lambda xy.y \subseteq x \wedge \neg Py$	(*loose downdating*)
$\lambda P.\lambda xy.y \subseteq x \wedge \neg Py \wedge \neg\exists z(y \subset z \subseteq x \wedge \neg Pz)$	(*strict downdating*)

Standard intuitionistic logic is a forward-looking system, but the full dynamic logic will include backward-looking downdates and revisions. The resulting Dynamic Modal Logic covers all cognitive tasks covered in the Gärdenfors theory of Section 1.5, and admits much more elaborate statements about them. The system has been studied extensively in (De Rijke, 1993), which has results on its expressive power and axiomatization and proves its undecidability. (Van Benthem (1993) presents a decidable reformulation.) Extensions of the formalism may be defined using operators from Temporal Logic. For instance, appropriate pre- and postconditions for strict updating and downdating will involve the well-known temporal operators *since* and *until*.

Other static systems which have been turned into dynamic ones include the theory of generalized quantifiers. There are many forms of change here: in bindings, ranges of quantification, drawing samples from domains, and model construction. (Cf. Van den Berg, 1995; Van Eijck and De Vries, 1992; Kanazawa, 1993b; Keenan and Westerståhl, 1996; Van Eijck and Kamp, 1996; Hintikka and Sandu, 1996.)

2.1.3. Dynamic constants as operators in relational algebra

Our general perspective employs the usual mathematical notion of a *state space* (i.e. poly-modal Kripke model) $\langle S, \{R_\alpha \mid \alpha \in \mathrm{AT}\}, V\rangle$. Over the atomic actions R_α, there is a procedural repertoire of operations creating compound actions. Examples of such procedural operations are sequential composition, choice, iteration as found in computer programs. Less standard examples include the DPL test negation:

$$\neg R = \{\langle x, x\rangle \mid \neg\exists y\langle x, y\rangle \in R\}$$

or the directed functions of categorial grammar (cf. Moortgat, 1996, this Handbook):

$$A\backslash B = \{\langle x, y\rangle \mid \forall z(\langle z, x\rangle \in A \to \langle z, y\rangle \in B)\},$$
$$B/A = \{\langle x, y\rangle \mid \forall z(\langle y, z\rangle \in A \to \langle x, z\rangle \in B)\}.$$

What we see here is a move from a standard Boolean Algebra of propositions to a Relational Algebra of procedures. The standard repertoire in relational algebras is:

Boolean operations: $-$ (complement) $\cap$ (intersection) $\cup$ (union)
Ordering operations: $\circ$ (composition) $^{\cup}$(converse)

with a distinguished diagonal Δ for the identity relation. These operations are definable in a standard predicate logic with variables over states:

$-R$	$\lambda xy.\neg Rxy,$
$R \cap S$	$\lambda xy.Rxy \wedge Sxy,$
$R \cup S$	$\lambda xy.Rxy \vee Sxy,$
$R \circ S$	$\lambda xy.\exists z(Rxz \wedge Szy),$
$R^{\cup}$	$\lambda xy.Ryx.$

This formalism can define many other procedural operators. In particular,

$$\begin{array}{ll} \neg R & \Delta \cap -(R \circ R^{\cup}), \\ A \backslash B & -(A^{\cup} \circ -B), \\ B/A & -(-B \circ A^{\cup}). \end{array}$$

The literature on Relational Algebra contains many relevant results concerning axiomatization of valid identities between such relational expressions, as well as expressive power of various choices of operators (see Németi, 1991). One natural measure of fine-structure here is the number of state variables needed in their definitions. This tells us the largest configuration of states involved in determining the action of the operator. The resulting Finite Variable Hierarchy of semantic complexity relates Relational Algebra with Modal Logic (cf. Andréka, Van Benthem and Németi, 1994). Its mathematical properties seem significant for dynamic logical operators in general: (1) the above vocabulary of Relational Algebra suffices for defining all relational operators with a 3-variable first-order definition (these include most common cases), (2) each n-variable level has a finite functionally complete set of operators, (3) there is no finite functionally complete set of algebraic operators for the whole hierarchy at once. The latter result shows how the logical space of dynamic propositional operators is much richer than that of classical Boolean Algebra.

2.1.4. Process equivalences and invariance

In order to understand a certain kind of process, one has to set up a criterion of identity among its different representations. One important notion to this effect is *bisimulation*, prominent in the computational literature, which tends to be richer in this respect than traditional logical semantics (cf. Milner, 1980; Hennessy and Milner, 1985). A bisimulation is a binary relation C between states in two 'labeled transition systems' (i.e. our dynamic transition models) $\langle S, \{R_\alpha \mid \alpha \in \mathrm{AT}\}, V\rangle$ and $\langle S', \{R'_\alpha \mid \alpha \in \mathrm{AT}\}, V'\rangle$ which connects only states with the same atomic valuation, and which satisfies the following back-and-forth clauses:

if xCx', $xR_\alpha y$, *then there exists some* y' *with* yCy', $x'R'_\alpha y'$,
if xCx', $x'R'_\alpha y'$, *then there exists some* y *with* yCy', $xR_\alpha y$.

This allows mutual tracing of the process in the two transition models, including its choice points. There are many other notions of process simulation: a coarser one is the 'trace equivalence' discussed in (Van Benthem and Bergstra, 1993), and a finer one is the'generated graph equivalence' discussed in the same paper.

There is a close connection between process equivalences and the design of a dynamic language. In particular, bisimulation is the key semantic invariance for a modal language describing labeled transition systems, which has the usual Boolean operators as well as indexed modalities $\langle a\rangle$ for each atomic action $a \in A$. Whenever C is a bisimulation between two models M, M' with sCs', we have

$$s \in \|\varphi\|^{M} \quad \textit{iff} \quad s' \in \|\varphi\|^{M'}, \quad \textit{for all modal formulas } \varphi.$$

This observation can be reversed:

> A first-order formula over labeled transition systems is invariant for bisimulation iff it is definable by means of a modal formula.

In propositional dynamic logic, this invariance persists for formulas, but there is also a new aspect. The above back-and-forth clauses in bisimulation are inherited by all program relations $\|\pi\|$, not just the atomic ones. More specifically, all regular program operations O are *safe for bisimulation*, in the sense that, whenever C is a bisimulation between two models with transition relations $R_1, \ldots, R_n$, it must also be a bisimulation for the transition relation $O(R_1, \ldots, R_n)$. This observation, too, can be reversed (Van Benthem, 1993):

> A first-order relational operation $O(R_1, \ldots, R_n)$ is safe for bisimulation iff it can be defined using atomic relations $R_\alpha xy$ and atomic tests $\alpha?$, using only the three relational operations of $\circ$ (composition), $\cup$ (union) and $\neg$ (DPL negation).

Thus, bisimulation seems very close to the mark for dynamic semantic operators with a modal flavor. Different outcomes will be obtained with coarser or finer notions of process equivalence. It would be of interest to see which level of invariance is plausible for the procedures involved in processing natural language.

2.1.5. Typology of dynamic procedures

Another source of more specific dynamic structure is the search for denotational constraints, suggested by semantic analysis of key linguistic items (cf. again the theory of generalized quantifiers). For instance, relational operators may obey various natural Boolean constraints (cf. Van Benthem, 1986; Keenan and Faltz, 1985), often of a computational character. One well-known example is *continuity* of an operator in one of its arguments:

$$O\Big(\ldots, \bigcup_{i\in I} R_i, \ldots\Big) = \bigcup_{i\in I} O(\ldots, R_i, \ldots).$$

Continuous operations compute their values locally, on single transitions (note that $R = \cup\{\{\langle x,y\rangle\} \mid Rxy\}$). Boolean intersection and union are continuous in both arguments, and so are relational composition and converse. A non-example is Boolean complement. This restriction has some bite. Van Benthem (1991a) proves that, for each fixed arity, there are only finitely many continuous permutation-invariant relational operators. (Belnap (1977) proposes a weaker notion of *Scott continuity* admitting more candidates.) Another source of constraints in dynamic semantics is the typology of cognitive actions themselves. For instance, updates are often taken to be idempotent: repeating them is unnecessary ($\forall xy(Rxy \to Ryy)$). Veltman (1991) wants them to be functions. Such basic choices will influence the choice of a procedural repertoire. For instance, if all admissible actions are to be idempotent, then composition is not a safe combination, while choice or iteration are. Likewise, special atomic repertoires may be of interest. For instance, the basic DPL actions R of propositional test and random assignment both satisfy the identity $R \circ R = R$, and both are symmetric relations. Other interesting denotational constraints of this kind occur in (Zeinstra, 1990) (cf. Section 1.4).

2.1.6. *Styles of inference*

We now turn from matters of dynamic vocabulary and expressive power to the issue of dynamic inference. The standard Tarskian explication of valid inference expresses transmission of truth: "in every situation where all premises are true, so is the conclusion". But what is the sense of this when propositions are procedures changing information states? There are plausible options here, and no single candidate has won universal favor so far. Here is a characteristic general feature. If premises and conclusions are instructions for achieving cognitive effects, then their presentation must be crucial, including sequential order, multiplicity of occurrences, and relevance of each move. This brings us into conflict with the basic structural rules of standard logic that allow us to disregard such aspects in classical reasoning (cf. Moortgat, 1996, this Handbook). Here are some dynamic styles of inference. The first employs fixed points for propositions (where their update procedure effects no state change) as approximations to classical truth, the second focuses on transitions to achieve an effect, and the third is a compromise between the two (Veltman, 1991; Van Benthem, 1991a).

test-test consequence
In all models, each state which is a fixed point for all premises is also a fixed point for the conclusion:

$$\varphi_1, \ldots, \varphi_n \models_{\text{test-test}} \psi \quad \text{iff} \quad \Delta \cap \|\varphi_1\|^M \cap \cdots \cap \|\varphi_n\|^M \subseteq \|\psi\|^M, \text{ for all models } M.$$

update-update consequence
in all models, each transition for the sequential composition of the premises is a transition for the conclusion:

$$\varphi_1, \ldots, \varphi_n \models_{\text{update-update}} \psi \quad \text{iff} \quad \|\varphi_1\|^M \circ \cdots \circ \|\varphi_n\|^M \subseteq \|\psi\|^M, \text{ for all models } M.$$

update-test consequence
in all models, each state reached after successful processing of the premises is a fixed point for the conclusion:

$$\varphi_1, \ldots, \varphi_n \models_{\text{update-test}} \psi \quad \text{iff} \quad \text{range}(\|\varphi_1\|^M \circ \cdots \circ \|\varphi_n\|^M) \subseteq \text{fix}(\|\psi\|^M), \text{ for all models } M.$$

Thus a variety of dynamic styles of inference emerges, reflecting different intuitions and possibly different applications. These show a certain coherence. For instance, Beaver (1992) analyzes presupposition as a test-update consequence stating that the premises can be processed only from states where the conclusion has a fixed point. Groenendijk and Stokhof (1991) require that the conclusion be processable after the premises have been processed successfully.

DPL consequence
in all models, in each state that is reached after successful processing of the premises, processing of the conclusion is possible:

$$\varphi_1, \ldots, \varphi_n \models_{\text{DPL}} \psi \quad \text{iff} \quad \text{range}(\|\varphi_1\|^M \circ \cdots \circ \|\varphi_n\|^M) \subseteq \text{dom}(\|\psi\|^M), \text{ for all } M.$$

Here, the existential quantification for the conclusion takes care of free variables that are to be captured from the premises. (This "for all – there exists" format may also be observed with implications in DRT.) Van Eijck and de Vries (1995) require a converse, proposing that the domain of the composed premises be contained in the *domain* of the conclusion.

One way of defining a style of inference is through its general properties, expressed in structural rules. For instance, test-test consequence behaves like standard inference:

$$\varphi \Rightarrow \varphi, \qquad \textit{Reflexivity},$$

$$\frac{X \Rightarrow \varphi \quad Y, \varphi, Z \Rightarrow \psi}{Y, X, Z \Rightarrow \psi}, \qquad \textit{Cut Rule},$$

$$\frac{X, \varphi_1, \varphi_2, Y \Rightarrow \psi}{X, \varphi_2, \varphi_1, Y \Rightarrow \psi}, \qquad \textit{Permutation},$$

$$\frac{X, \varphi, Y, \varphi, Z \Rightarrow \psi}{X, \varphi, Y, Z \Rightarrow \psi}, \qquad \textit{Right Contraction},$$

$$\frac{X, \varphi, Y, \varphi, Z \Rightarrow \psi}{X, Y, \varphi, Z \Rightarrow \psi}, \qquad \textit{Left Contraction},$$

$$\frac{X, Y \Rightarrow \psi}{X, \varphi, Y \Rightarrow \psi}, \qquad \textit{Monotonicity}.$$

By contrast, update-update satisfies only Reflexivity and Cut. There are some exact representation results (Van Benthem, 1991a): (1) {*Monotonicity, Contraction, Reflexivity, Cut*} completely determine test-test consequence, (2) {*Reflexivity, Cut*} completely determine update-update inference. But this is not an all-or-nothing matter. Inferential styles may in fact modify standard structural rules, reflecting a more delicate handling of premises. Update-test consequence has none of the above structural properties, but it is completely characterized by

$$\frac{X \Rightarrow \psi}{\varphi, X \Rightarrow \psi}, \qquad \textit{Left Monotonicity},$$

$$\frac{X \Rightarrow \varphi \quad X, \varphi, Z \Rightarrow \psi}{X, Z \Rightarrow \psi}, \qquad \textit{Left Cut}.$$

The DPL style of inference is also non-classical, in that various structural rules from classical logic fail. For instance, it is

non-monotonic: $\exists x Ax \models_{\text{DPL}} Ax$, but not $\exists x Ax, \neg Ax \models_{\text{DPL}} Ax$,
non-contractive: $\exists x Ax, \neg Ax, \exists x Ax \models_{\text{DPL}} Ax$, but not $\exists x Ax, \neg Ax \models_{\text{DPL}} Ax$,
non-transitive: $\exists x Ax, \neg Ax \models_{\text{DPL}} \exists x Ax \models_{\text{DPL}} Ax$, but not $\exists x Ax, \neg Ax \models_{\text{DPL}} Ax$.

The only valid structural rule of inference is *Left Monotonicity*. It is not completely clear, however, that this is the last word. In practice, applications of DPL to natural language will use only very special 'decorations' of grammatical structures with individual variables. For instance, it seems reasonable to require that every quantifier have a unique bound variable associated with it. But then, the DPL *fragment* with this property may be shown to satisfy unrestricted monotonicity, allowing insertion of premises in arbitrary positions (Van Benthem, unpublished). Other well-behaved fragments may be relevant for natural language analysis, too.

Often, one inferential style can be simulated inside another, by adding suitable logical operators. Here is an illustration. Test-test consequence may be reduced to update-update consequence using a relational fixed point operator Φ sending relations R to their diagonal $\lambda xy.Rxy \wedge y = x$:

$$\varphi_1, \dots, \varphi_n \models_{\text{test-test}} \psi \quad \text{iff} \quad \Phi(\varphi_1), \dots, \Phi(\varphi_n) \models_{\text{update-update}} \Phi(\psi).$$

There is no similar faithful converse embedding. (This would imply Monotonicity for update-update consequence.) Another interplay between structural rules and logical constants arises as follows. Operators may license additional structural behaviour, not for all propositions, but for special kinds only (cf. Girard, 1987). For instance, in dynamic styles of inference, let O be some operator that is to admit of arbitrary monotonic insertion:

$$\frac{X, Y \Rightarrow \psi}{X, O(\varphi), Y \Rightarrow \psi}.$$

This can only be the case if $O(\varphi)$ is a test contained in the diagonal relation. It would be of interest to see how the linguistic formulation of actual arguments provides cues for adopting and switching between inferential styles.

Completeness theorems for dynamic styles of inference in various fragments of propositional dynamic logic may be found in (Kanazawa, 1993a; Blackburn and Venema, 1993). These results exemplify one direction of thinking in logic: from semantic notions of inference to their complete axiomatic description. Another line in the literature starts from given axiomatic properties of dynamic operators, and then determines corresponding complete semantics via representation theorems (cf. Alchourrón, Gärdenfors and Makinson (1985) and the ensuing tradition). Eventually, both logical treatments of dynamic inference may be too conservative. Perhaps, the very notion of formal proof needs re-thinking in a dynamic setting (a first attempt at defining 'proofs as texts' may be found in (Vermeulen, 1994)). Natural reasoning seems to involve the interplay of a greater variety of mechanisms at the same time (inferring, updating, querying, etcetera).

2.2. *Categories for dynamic semantics*

Dynamic logic is by no means the only mathematical paradigm for implementing the fundamental ideas of dynamic semantics. As a counterpoint to the preceding sections, we outline an alternative logical framework based on category theory, sometimes called the 'Utrecht approach'. Its basic tenet is this: the business of dynamic semantics is modeling interpretation processes. Thus, it is not sufficient to compositionally specify correct meanings: one should also specify these in a way that reflects temporal processes of interpretation. Category Theory provides the tools to do this.

Category theory is a branch of mathematics that is widely applied in both mathematics and computer science. (Some good textbooks are McLane, 1971; Manes and Arbib, 1975; Barr and Wells, 1989.) The uses of Category Theory in linguistics are less widespread, but multiplying. The reader is referred to (Reyes and Macnamara, 1994) for another application in linguistics.

2.2.1. *The program of monoidal updating*

The Utrecht approach develops radical versions of file-change semantics/DRT (see Visser and Vermeulen, 1995). Consider a simple sample sentence: *John cuts the bread with a sharp knife.* This will be analyzed as follows:

$$((\textit{subject John}_j)\ \textit{cuts}\ (\textit{object the}_u\ \textit{bread})\ (\textit{with}\ a_v\ \textit{sharp knife})).$$

Here, virtually all grammatical structure will be interpreted as semantic actions such as pushing a new file to a data stack or popping the last file from the stack. In an alternative notation:

> *push push subject John*$_j$ *pop cuts push object the*$_u$ *bread pop push with* a_v *sharp knife pop pop.*

In other words, all grammatical structure gets a dynamic potential similar to the existential quantifier in DPL/DRT or to the dynamic *suppose* operator in Zeinstra's logic. As a consequence, the usual components of a sentence, such as (*object the*$_u$ *bread*), are not necessarily the only possible inputs in a compositional interpretation. In fact, the meaning of any contiguous linguistic chunk of text can be specified. Thus, the source algebra of the interpretation is the language of arbitrary strings over an alphabet including such characters as *subject, with,* a_v, *pop,* whose basic operation is concatenation. This syntactic operation is matched at the semantic level with a dynamic operation, say *merge* or *composition*. This merge will be associative, thus reflecting the associativity of concatenation at the syntactic level. This has as a consequence, that the ambiguity of dividing up a sentence into chunks does not result in the assignment of different meanings. Components in the traditional sense, i.e. chunks with matching opening and closing brackets, correspond to *local files* that are introduced, used for some time and then discarded. (The words *subject, object,* and *with* contain machinery to arrange that the information of the discarded files is stored in the correct files associated with *cuts* at the sentence level.) So far, this semantics has been developed for narrative with existential quantifiers only. Even so, it exemplifies some broad programmatic features for a full-fledged dynamic semantics in the above sense.

In this approach, genuine grammaticality is decided at the semantic level, since the syntactic specification language does not have any interesting grammar at all. The fact that tasks that are traditionally assigned to grammar are now shifted to the semantic level, reflects a move that is typical in dynamic semantics: redivision of labor between syntax and semantics.

Since the semantic objects form a monoid (the basic operation is associative and there is a unit element), the semantics satisfies the *break-in principle*: any contiguous chunk of text can be assigned a meaning. As a result, one can process meanings *incrementally*. This seems a linguistically realistic, and hence desirable feature.

2.2.2. *Meanings and contexts*

Meanings in this approach are databases, just as in DRT. The main difference with ordinary DRT is that much more 'dynamic potential' is present in contexts. Contexts contain both global information connected to the anaphoric machinery ('variables') and local syntactic information (e.g., a file that stores local information about the subject of a sentence). Contexts regulate the way in which information is stored in case new information is added to a database.

Words like *with* and *object* stand for *argument places*. Their meanings are little machines that look for the place where information connected with the word (*with a knife*) is to be stored in the database that is being built. ("The knife is the *Instrument* of the cutting" – compare (Davidson, 1967; Parsons, 1990).) An anaphor like he_v links files introduced in the sentence (thematic roles such as *Agent* and *Instrument*) with files globally present in the discourse. In this way the chunk (*subj* he_v) ensures that the file locally known as the subject is connected to the file globally known as v. Thus, he_v gets the correct role in the semantics: it is a locus where local and global information are fused.

2.2.3. *Diachronic information orderings as categories*

Let us look at some chunks of our earlier example.

((*subject* $John_j$) *cuts* (*object*

and

the_u *bread*) (*with* a_v *sharp knife*)).

The meanings associated with these chunks are databases containing files/discourse objects. These databases have a layered structure that reflects some aspects of the local syntactic structure – e.g., the discourse objects are stored on the levels of a stack that represents part of the bracket structure. This structure on discourse objects occurs in the context part of the databases. Our problem now becomes *to describe what happens if two dynamic databases are 'clicked together'*. We do not only want to describe what the new object looks like, but also want to describe the *flow of files*: where do the files of the original databases re-occur in the new one? Apart from philosophical reasons to insist on describing the flow of files there is a pragmatic one: the description of the new meaning object and the verification that it has the desired properties quickly becomes too complicated if we do not have a principled way of describing the flow. This is where categories make their appearance: the flow of files is described by a *diachronic information ordering* and this ordering turns out to be a category.

One should distinguish (at least) two ways of ordering linguistic information. First, there is a *synchronic* ordering. For example, consider two slips of paper. One states *Jan is wearing something new*, the other *Jan is wearing a new hat*. Evidently, the first slip is less informative than the second. Whatever information state someone is in, being offered the second slip will make her at least as informed as being offered the first. So we compare the effects of pieces of information offered at the same time to the same person in different possible situations. The second ordering is the one we are after presently: the *diachronic* ordering, which looks at information as it occurs in time. Consider *Genever is a wonderful beverage. Not only the Dutch are fond of it.* The information content of these two statements forms an indissoluble whole, by virtue of their consecutive presentation. A mathematical analysis of the diachronic ordering $\leqslant$ leads to the core of the Utrecht approach. For a start, assume that $\leqslant$ is a pre-order, i.e. a transitive and reflexive binary relation. (There is no strong evidence for antisymmetry, and hence partial order.) But, there is further relevant dynamic structure. Consider this example:

(40) Genever is a wonderful beverage, I like it. Cognac is not too bad either. I like it too.

Here, the meaning of *I like it* is embedded in the meaning of the whole text twice. But not in the same way: the first *it* will be linked to *Genever*, the second one to *Cognac*. This suggests that the diachronic ordering should rather be a *labeled pre-ordering*, which adds information about the kind of embedding involved.

The preceding observation suggests a move to 'labeled transition systems' similar to those encountered in section Section 2.1 above. Such transition systems can be described in many ways. We describe them here as logical generalizations of partial pre-orders. We have structures $\langle O, L, R\rangle$, where O is a domain of *objects*, L a set of *labels*, and R a ternary relation between objects, labels and objects. A triple $\langle x, \lambda, y\rangle$ in R is called an *arrow*. We shall write $x \leqslant_\lambda y$ for: $\langle x, \lambda, y\rangle \in R$. Here are the analogues of the pre-order principles. Reflexivity says that everything can be embedded into itself *in a trivial way*. This requires a special label *id* such that, for every x, y in $O, x \leqslant_{\mathrm{id}} y$ iff $x = y$. Next, transitivity says we can compose ways of embedding in suitable circumstances. Suppose we have $x \leqslant_\lambda y$ and $y \leqslant_\mu z$. Then $\lambda \circ \mu$ is defined and we have: $x \leqslant_{\lambda\circ\mu} z$. We demand that $id \circ \lambda = \lambda \circ id = \lambda$ and $\lambda \circ (\mu \circ \nu) = (\lambda \circ \mu) \circ \nu$. (Here an equation $\gamma = \delta$ states that γ is defined iff δ is, and that γ and δ are equal where defined.) Finally, for the sake of parsimony, we demand that every label is used at least once in some arrow. (There are obvious analogies here with Dynamic Logic and the Arrow Logic of Section 2.4.8.) Now, with the label *id* we can associate a function from objects to arrows. Moreover the partial operation $\circ$ on labels induces one on arrows. The resulting structure of objects and arrows is a *category* in the sense of Category Theory. (In fact our labeled pre-orderings have slightly more structure than a category.) Thus dynamic semantics can now avail itself of useful notions from an established mathematical discipline. (For instance, an arrow $x \leqslant_\lambda y$ is an *isomorphism* if there is an arrow $y \leqslant_\mu x$ such that $\lambda \circ \mu = \mu \circ \lambda = id$.)

The diachronic ordering may be viewed as a special kind of category, suitable for dynamic meanings. We already had a monoidal merge $\bullet$ on objects. We relax the notion of monoid by allowing that $(x \bullet y) \bullet z$ is not strictly identical to $x \bullet (y \bullet z)$, but that there is a standard isomorphism $\alpha(x, y, z)$ from $(x \bullet y) \bullet z$ to $x \bullet (y \bullet z)$. (This ensures category-theoretic *coherence*: see (McLane, 1971, pp. 161–176).) To make updating yield information growth along our ordering, we also assume standard embeddings of x and y into $x \bullet y$, say, via $\text{in}_1(x, y) : x \to x \bullet y$ and $\text{in}_2(x, y) : y \to x \bullet y$. For example, then, x may be embedded in $(x \bullet y) \bullet z$ as follows. First x is embedded in $x \bullet y$ by $\text{in}_1(x, y)$, and $(x \bullet y)$ in its turn is embedded in $(x \bullet y) \bullet z$ by $\text{in}_1(x \bullet y, z)$. Now $(x \bullet y) \bullet z$ is identified with $x \bullet (y \bullet z)$ by $\alpha(x, y, z)$. Alternatively, x is embedded in $x \bullet (y \bullet z)$ by $\text{in}_1(x, y \bullet z)$. Putting all this together, one obtains equalities like the following.

$$\text{in}_1(x, y) \circ \text{in}_1(x \bullet y, z) \circ \alpha(x, y, z) = \text{in}_1(x, y \bullet z).$$
$$\text{in}_2(x, y) \circ \text{in}_1(x \bullet y, z) \circ \alpha(x, y, z) = \text{in}_1(y, z) \circ \text{in}_2(x, y \bullet z).$$
$$\text{in}_2(x \bullet y, z) \circ \alpha(x, y, z) = \text{in}_2(y, z) \circ \text{in}_2(x, y \bullet z).$$

The resulting mathematical structures are called *m-categories*. m-categories are the natural medium for thinking about dynamic updating and dynamic contexts. Starting from simple m-categories that describe contexts and contents, we can now assemble meanings by the well-known categorical *Grothendieck construction* (see Barr and Wells, 1989; Visser and Vermeulen, 1995).

2.3. Dynamics related to statics

2.3.1. Translations

It is often useful to define functions from the expressions of one logic to those of another. If such a function preserves logical consequence it is called a *translation* and in the following section we shall define translations from PDL and QDL to classical logic. Our method will be to take the truth conditions of the source logics and transcribe them in the object language of the target logic. This is in fact an old procedure, as the so-called *standard translation* from modal logic into predicate logic may witness. To obtain this translation, associate a unary predicate symbol P with each propositional letter p of the modal language and let R be some binary relation symbol. Then define the translation ST, sending sentences from propositional modal logic to formulae of predicate logic having at most the fixed variable i free, as follows.

$$\begin{aligned} \text{ST}(p) &= Pi, \\ \text{ST}(\bot) &= \bot, \\ \text{ST}(\varphi \to \psi) &= \text{ST}(\varphi) \to \text{ST}(\psi), \\ \text{ST}(\Diamond\varphi) &= \exists j(Rij \wedge [j/i]\text{ST}(\varphi)). \end{aligned}$$

Whether this function really preserves entailment depends on the modal system under investigation, of course. For the minimal modal logic K the translation will do as it stands, but for stronger logics we need to put additional constraints on the relation denoted by R. For many modal systems S a lemma of the following form will hold.

EMBEDDING LEMMA. $\varphi \models_S \psi$ *iff* $\mathrm{AX}, \mathrm{ST}(\varphi) \models \mathrm{ST}(\psi)$.

Here AX is some set of axioms putting extra requirements on R. For example, we can take AX to be the requirement that R be reflexive and transitive, while instantiating S as the system S4. In general, the correctness of a translation may require working with special classes of models.

There are various reasons why it is handy to have translations around whenever they are available. One reason is that it is often possible to derive information about the source logic of a translation from properties of the target logic that are already known. For example, the standard translation immediately tells us that the modal logics that can be treated in this way are recursively axiomatizable and will have the Löwenheim–Skolem property. Other translations often give us decidability of a system. Some information may not be obtainable in this easy way, of course. For example, although the above translation shows that there *are* recursive axiomatizations of the modal logics under consideration, it does not tell us what these axiomatizations look like. Moreover, some semantic characteristics of the original logic may be lost in translation. *Traduttore traditore,* not only in real life, but in logic as well.

Reasons for studying translation functions also include some of a more applied character. One is that a translation into classical logic will make it possible to use a general purpose classical theorem prover for the source logic. Another reason is that for applied purposes we often need to have many logics working in tandem. In linguistics, for example, we need logics that can deal with modalities, with temporal expressions, with verbs of perception, with propositional attitudes, with defaults, with dynamics, and with many other things. Trying to set up a logic that can simultaneously deal with all these things by adding up the characteristics of modal logic, temporal logic, default logic, dynamic logic, etc. will almost certainly result in disaster. Translating all these special logics into one common general purpose target logic may be a viable strategy, however.

2.3.2. *From dynamic logic to classical logic*

In this section we shall give translations of Dynamic Logic into classical logic. It will not be possible to let elementary predicate logic be our target language, because of the infinitary nature of the iteration operator. However, if we allow infinite disjunctions, and thus obtain the logic known as $L_{\omega_1\omega}$, translations are possible. The following function τ sends PDL constructs to classical formulae. The idea is that each PDL formula is translated as a formula which may have one variable i free and that a PDL program goes to a formula which may contain an additional free variable j. The variables i and j are fixed in advance, say as the first and second variables in some given ordering. Think of i as being the input state, of j as the output state. Each propositional letter p is associated with a unary predicate symbol P and each atomic program α with a binary relation symbol R_α. Let π^0 stand for $\top?$ (the **skip** command) and π^{n+1} for $\pi^n; \pi$.

$$\tau(p) = Pi,$$

$$\tau(\bot) = \bot,$$
$$\tau(\varphi \to \psi) = \tau(\varphi) \to \tau(\psi),$$
$$\tau([\pi]\varphi) = \forall k(\tau(\pi) \to [k/i]\tau(\varphi)), \text{ where } k \text{ is new},$$
$$\tau(\alpha) = R_\alpha ij,$$
$$\tau(\varphi?) = i = j \wedge \tau(\varphi),$$
$$\tau(\pi_1 ; \pi_2) = \exists k([k/j]\tau(\pi_1) \wedge [k/i]\tau(\pi_2)), \text{ where } k \text{ is new},$$
$$\tau(\pi_1 \cup \pi_2) = \tau(\pi_1) \vee \tau(\pi_2),$$
$$\tau(\pi^*) = \bigvee_n \tau(\pi^n).$$

This translation, which obviously follows the semantics for PDL given in Section 2.1.1 above, can be extended to a translation of QDL into $L_{\omega_1\omega}$ (cf. Harel, 1984). PDL may also be translated into second-order logic: with clauses as before, except that now

$$\tau(\pi^*) = \forall X\big((Xi \wedge \forall kh((Xk \wedge [k/i, h/j]\tau(\pi)) \to Xh) \to Xj\big),$$

where k and h are fresh variables and X varies over sets of states. The formula says that i and j are in the reflexive transitive closure of the denotation of π, which is true iff j is in all sets containing i which are closed under π successors.

We shall extend the last translation to a translation of QDL into three-sorted second order logic plus some axioms. There will be three types of objects: states, entities and registers. We use the following notation: u (with or without superscripts or subscripts) will be a constant that denotes a register; v will be a variable over registers; ρ will vary over terms of type register. The constant $\mathcal{V}$ will denote a two-place function from registers and states to entities; $\mathcal{V}(\rho, i)$ can be thought of as the value of register ρ in state i. We define $i[\rho_1 \cdots \rho_n]j$ to be short for $\forall v((\rho_1 \neq v \wedge \cdots \wedge \rho_n \neq v) \to \mathcal{V}(v,i) = \mathcal{V}(v,j))$ (i and j differ at most in $\rho_1, \ldots, \rho_n$). We require the following: for each state, each register and each entity, there must be a second state that is just like the first one, except that the given entity is a value of the given register. Moreover, we demand that different constants denote different registers.

AX1 $\forall i \forall v \forall x \exists j (i[v]j \wedge \mathcal{V}(v,j) = x)$,
AX2 $u \neq u'$ for each two syntactically different constants u and u'.

The translation is now obtained in the following way. We assume the set of QDL variables and the set of register constants to have a fixed ordering each. We let $\tau(x_n) = \mathcal{V}(u_n, i)$; $\tau(c) = c$, for each constant c and $\tau(f(t_1, \ldots, t_n)) = f(\tau(t_1), \ldots, \tau(t_n))$. Moreover, we let

$$\tau(R(t_1, \ldots, t_n)) = R(\tau(t_1), \ldots, \tau(t_n)),$$
$$\tau(t_1 = t_2) = \tau(t_1) = \tau(t_2),$$
$$\tau(x_n := ?) = i[u_n]j,$$
$$\tau(x_n := t) = i[u_n]j \wedge \mathcal{V}(u,j) = [j/i]\tau(t).$$

The remaining constructs of QDL are translated as before. It is not difficult to prove the following lemma.

EMBEDDING LEMMA. *Let $\models_2$ be the semantical consequence relation of three sorted second order logic, then*

$$\varphi \models_{\text{QDL}} \psi \quad \textit{iff}. \quad \text{AX1}, \text{AX2}, \tau(\varphi) \models_2 \tau(\psi).$$

Since we have already observed (in Section 2.1.1) that both DRT and DPL can be embedded in the star free part of QDL, this immediately gives us embeddings from DRT and DPL into (three-sorted) predicate logic; for each DRS K we have a predicate logical formula with at most the state variables i and j free which shows the same input/output behaviour as K. In the next section we shall see an application of this.

2.3.3. An application: Compositional DRT

Several researchers (e.g., Groenendijk and Stokhof, 1990; Asher, 1993; Bos, Mastenbroek, McGlashan, Millies and Pinkal, 1994) have stressed the desirability to combine the dynamic character of DRT and DPL with the possibility to interpret expressions compositionally as it is done in Montague Grammar (see also Van Eijck and Kamp, 1996, this Handbook). To this end one must have a logic that combines the constructs of DRT with lambda abstraction, but until recently no simple semantically interpreted system supporting full lambda conversion was forthcoming. Using the ideas from the previous section it is easy to define such a logic, however. We shall follow Muskens (1991, 1994, 1995a, 1995b) in giving an interpretation of DRT in the first-order part of classical type logic.

To get the required embedding, let $\mathcal{V}$ be a constant of type $\pi(se)$ (where π is the type of registers) and identify discourse referents with constants of type π. The original DRT constructs can now be obtained by means of the following abbreviations; conditions will be terms of type st DRSs terms of type $s(st)$.

Pu	*abbreviates*	$\lambda i.P(\mathcal{V}(u)(i))$,
$u_1 R u_2$	*abbreviates*	$\lambda i.R(\mathcal{V}(u_1)(i))(\mathcal{V}(u_2)(i))$,
u_1 **is** u_2	*abbreviates*	$\lambda i.(\mathcal{V}(u_1)(i)) = (\mathcal{V}(u_2)(i))$,
not K	*abbreviates*	$\lambda i \neg \exists j K(i)(j)$,
K_1 **or** K_2	*abbreviates*	$\lambda i \exists j (K_1(i)(j) \vee K_2(i)(j))$,
$K_1 \Rightarrow K_2$	*abbreviates*	$\lambda i \forall j (K_1(i)(j) \rightarrow \exists k K_2(j)(k))$,
$[u_1 \cdots u_n \mid \gamma_1, \ldots, \gamma_m]$	*abbreviates*	$\lambda i \lambda j. i[u_1, \ldots, u_n]j \wedge \gamma_1(j) \wedge \cdots \wedge \gamma_m(j)$,
$K_1 ; K_2$	*abbreviates*	$\lambda i \lambda j \exists k (K_1(i)(k) \wedge K_2(k)(j))$.

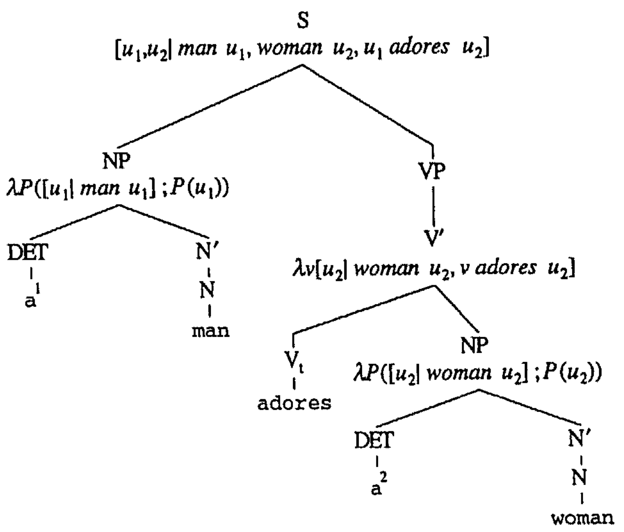
To allow for the possibility of compositional interpretation we have added the PDL sequencing operator (DPL conjunction) to the constructs under consideration. The following simple lemma is useful.

MERGING LEMMA. *If $u'_1, \ldots, u'_k$ do not occur in any of $\varphi_1, \ldots, \varphi_m$ then* $\models_{\text{AX}} [u_1 \cdots u_n \mid \varphi_1, \ldots, \varphi_m] ; [u'_1 \cdots u'_k \mid \gamma_1, \ldots, \gamma_r] = [u_1 \cdots u_n u'_1 \cdots u'_k \mid \varphi_1, \ldots, \varphi_m, \gamma_1, \ldots, \gamma_r]$.

We sketch the treatment of a small fragment of ordinary language in this system. It will be assumed that all determiners, proper names and anaphoric pronouns are indexed on the level of syntax. Here are translations for a limited set of basic expressions (variables P are of type $\pi(s(st))$, variables p and q of type $s(st)$ and variable Q is of type $(\pi(s(st)))(s(st)))$.

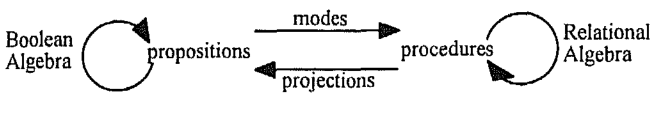
$\mathtt{a}^{\mathtt{n}}$	*translates as*	$\lambda P'\lambda P([u_n \mid] ; P'(u_n) ; P(u_n))$,
$\mathtt{no}^{\mathtt{n}}$	*translates as*	$\lambda P'\lambda P[\mid \mathbf{not}([u_n \mid] ; P'(u_n) ; P(u_n))]$,
$\mathtt{every}^{\mathtt{n}}$	*translates as*	$\lambda P'\lambda P[\mid ([u_n \mid] ; P'(u_n)) \Rightarrow P(u_n)]$,
$\mathtt{he}_{\mathtt{n}}$	*translates as*	$\lambda P(P(u_n))$,
`who`	*translates as*	$\lambda P'\lambda P\lambda v(P(v) ; P'(v))$,
`man`	*translates as*	$\lambda v[\mid \mathit{man}\ v]$,
`woman`	*translates as*	$\lambda v[\mid \mathit{woman}\ v]$,
`stink`	*translates as*	$\lambda v[\mid \mathit{stinks}\ v]$,
`adore`	*translates as*	$\lambda Q\lambda v(Q(\lambda v'[\mid v\ \mathit{adores}\ v']))$,
`if`	*translates as*	$\lambda pq[\mid p \Rightarrow q]$.

Note that the translation of (say) no^3 applied to the translation of `man` can be reduced to $\lambda P[\mid \mathbf{not}([u_3 \mid \mathit{man}\ u_3] ; P(u_3))]$ with the help of lambda-conversion and the merging lemma. In a similar way the sentence $\mathtt{a}^1$ `man adores` $\mathtt{a}^2$ `woman` can be translated as suggested in the tree below.



The method provides us with an alternative for the construction algorithm in standard DRT and with a fusion of insights from the Montague tradition with those of DRT. For more applications see also (Van Eijck and Kamp, 1996, this Handbook).

2.3.4. Two-level architecture and static tracing of dynamic procedures
The two-level approach of PDL suggests the following two-level architecture. Declarative propositions and dynamic procedures both have reasonable motivations. Presumably, actual inference is a mixture of more dynamic sequential short-term processes and more static long-term ones, not necessarily over the same representations. Thus, both systems must interact:



In such a picture, logical connections between the two levels become essential. There will be *modes* taking standard propositions to correlated procedures, such as 'updating' to make a proposition true, or 'testing' whether the proposition holds already. In the opposite direction, there are *projections* assigning to each procedure a standard proposition recording some essential feature of its action. Examples are the fixed point operator Φ giving the states where the procedure is already satisfied, or set-theoretic domain, giving the states where it can be performed at all. These new operators of 'logical management' may be analyzed technically much as those in Section 2.1, e.g., through type-theoretic analysis (cf. Turner, 1996, this Handbook). For instance, fixed-point is the only permutation-invariant projection that is a Boolean homomorphism (Van Benthem, 1991a). This style of analysis has been extended to eliminative update logic in (Van Benthem and Cepparello, 1994).

The above setting can also be analyzed using concepts from computer science. In particular, one can trace a dynamic process by means of propositions describing successive images of sets of states under its action. Define *strongest postconditions* and *weakest preconditions* as follows.

$$\mathrm{SP}(A,R) = R[A] \quad (= \{b \mid \exists a \in A\colon \langle a,b\rangle \in R\}),$$
$$\mathrm{WP}(R,A) = R^{-1}[A] \quad (= \{b \mid \exists a \in A\colon \langle b,a\rangle \in R\}).$$

The set $\mathrm{WP}(R,A)$ is also known as the *Peirce product* of R and A (cf. Brink, Britz and Schmidt, 1992). Note that $\|\langle\pi\rangle\varphi\| = \mathrm{WP}(\|\pi\|,\|\varphi\|)$. These notions may be used to render dynamic validity. For example, for update-update consequence, we have

$$\varphi_1,\ldots,\varphi_n \models_{\text{update-update}} \psi \quad \text{if and only if}$$
$$\mathrm{SP}(A,\|\varphi_1\|\circ\cdots\circ\|\varphi_n\|) \subseteq \mathrm{SP}(A,\|\psi\|) \quad \text{for arbitrary sets } A.$$

Moreover, there is an inductive calculus for computing weakest preconditions and strongest postconditions, with clauses such as:

$$\mathrm{SP}(A, R\circ S) = \mathrm{SP}(\mathrm{SP}(A,R),S),$$
$$\mathrm{WP}(R\circ S, A) = \mathrm{WP}(R,\mathrm{WP}(S,A)),$$

$$\begin{aligned}
\mathrm{SP}(A, R \cup S) &= \mathrm{SP}(A, R) \vee \mathrm{SP}(A, S), \\
\mathrm{WP}(R \cup S, A) &= \mathrm{WP}(R, A) \vee \mathrm{WP}(S, A), \\
\mathrm{SP}(A, R^{\cup}) &= \mathrm{WP}(R, A), \\
\mathrm{WP}(R^{\cup}, A) &= \mathrm{SP}(A, R).
\end{aligned}$$

As an application we give a weakest preconditions calculus which computes the truth conditions of any DRS or condition, given the total semantics for DRT discussed in Section 1.2.2. A simple induction will prove that $\mathrm{TR}(\varphi)$ is a predicate logical formula which is true under the same assignments as the condition φ is and that $\mathrm{WP}(K, \chi)$ is true under a iff there is some b such that $\langle a, b\rangle \in \|K\|$ and χ is true under b. In particular, $\mathrm{WP}(K, \top)$ will give the truth conditions of K.

$$\begin{aligned}
&\mathrm{TR}(\varphi) = \varphi \quad \text{if } \varphi \text{ is atomic}, \\
&\mathrm{TR}(\neg K) = \neg \mathrm{WP}(K, \top), \\
&\mathrm{TR}(K_1 \vee K_2) = \mathrm{WP}(K_1, \top) \vee \mathrm{WP}(K_2, \top), \\
&\mathrm{TR}(K_1 \Rightarrow K_2) = \neg \mathrm{WP}(K_1, \neg \mathrm{WP}(K_2, \top)), \\
&\mathrm{WP}([x_1, \dots, x_n \mid \varphi_1, \dots, \varphi_m], \chi) = \exists x_1 \cdots x_n (\mathrm{TR}(\varphi_1) \wedge \cdots \wedge \mathrm{TR}(\varphi_m) \wedge \chi).
\end{aligned}$$

A similar calculus can be given for DPL:

$$\begin{aligned}
&\mathrm{WP}(\neg\varphi, \chi) = \neg \mathrm{WP}(\varphi, \top) \wedge \chi, \\
&\mathrm{WP}(\varphi \to \psi, \chi) = \neg \mathrm{WP}(\varphi, \neg \mathrm{WP}(\psi, \top)) \wedge \chi, \\
&\mathrm{WP}(\varphi \wedge \psi, \chi) = \mathrm{WP}(\varphi, \mathrm{WP}(\psi, \chi)), \\
&\mathrm{WP}(\exists x \varphi, \chi) = \exists x \mathrm{WP}(\varphi, \chi), \text{ etc.}
\end{aligned}$$

And again $\mathrm{WP}(\varphi, \top)$ gives φ's truth conditions. Van Eijck and De Vries (1992) extend a calculus such as this one with clauses for generalized quantifiers and a description operator (see also Van Eijck and De Vries, 1995; Van Eijck and Kamp, 1996, this Handbook, where the format of the Segerberg axioms of Section 2.1.1 is used).

2.4. *General perspectives*

In this final section, we summarize our main logical themes, and point out some further issues and lines of formal investigation in dynamic semantics.

2.4.1. *Models for dynamics*

Our main logical paradigm has been *Dynamic Logic*, broadly conceived (Harel, 1984), viewing procedures as sets of transitions over spaces of (information) states. Dynamic operators then resemble those found in the relation-algebraic literature. Alternative universal-algebraic approaches are *Process Algebra* (Baeten and Weyland, 1990) or *Arrow Logic* (Venema, 1994). More sensitive notions of computation might involve 'failure paths' (Segerberg, 1991) or 'trace models' (Vermeulen, 1994). These may suggest richer languages. With processes as sets of state transition sequences ('traces'), the proper formalism is a 'branching time logic' combining evaluation at states with that on traces ('epistemic histories'). But further mathematical paradigms were available. Gärdenfors' original theory of belief change (Section 1.5) uses *Category Theory*, with dynamic procedures as morphisms that can be combined via categorial limit constructions. Also, Arrow Logic has categorial models (Van Benthem, 1994). And we have seen some Utrecht-style examples of concrete category-theoretic analysis for anaphora. Clearly, this alternative route deserves exploration.

Dynamic semantic paradigms have proof-theoretic alternatives – with Curry–Howard–deBruyn isomorphisms assigning algorithmic procedures to derivations for assertions. (Cf. this Handbook, the chapters by Moortgat, Buszkowski, and Turner.) *Proof Theory* has been proposed as a general paradigm of linguistic meaning in (Kneale and Kneale, 1962; Dummett, 1976), as well as (Van Benthem, 1991a) (categorial logic and typed lambda calculus), (Ranta, 1991) (Martin-Löf style type theories), (Gabbay and Kempson, 1992) ('labeled deductive systems'). We also briefly considered *Game Theory* as yet another alternative (Hintikka and Sandu, 1996, this Handbook), which provides logical games for evaluating statements, comparing model structures, or carrying on debates, with suitably assigned roles for players and winning conventions (cf. the survey Van Benthem, 1988). Winning strategies in evaluation or debating games provide analyses for truth and consequence in the work of Lorenzen (1959), Hintikka (1973). For model-theoretic 'Ehrenfeucht Games', cf. (Doets, 1993).

The paradigms of programs, proofs and games are not mutually exclusive. All involve movement through a space of deductive stages, information states, or game configurations. This requires a repertoire of atomic moves over states, that can be combined into complex procedures through 'logical constructions'. Thus, proofs involve basic combinatorics for trees: 'combination', 'arguing by cases' and 'hypothesizing', creating a dynamic block structure. Programs involve the usual constructions for instructions or plans such as 'sequential composition', 'indeterministic choice' or 'iteration', possibly guided by 'control assertions'. Finally, game operations reflect roles of different players, such as conjunctions or disjunctions indicating their rights of choice and duties of response, as well as the notion of 'role change' (signaled by negation). Finally, all three paradigms involve an explicit interplay between actions changing states, and standard declarative statements about the states traversed by actions.

2.4.2. *Higher levels of aggregation*

Language use is guided by *global strategies*, such as 'preferring the more specific interpretation' (Kameyama (1992) has computational linguistic architectures reflecting this). Global strategies have been most prominent in the game-theoretical literature. As a result, one also needs *global structures*, viz. texts or theories, and the meta-rules that govern our activities at these higher levels. Early work on logical structure of scientific theories in the Philosophy of Science is suggestive here (cf. Suppe, 1977), as well as the analysis of global structures of definition, proof and refutation in (Lakatos, 1976), or recent computational work on structured data bases (cf. Ryan, 1992). But these have not yet been integrated with mainstream logic. Another global challenge is the fact that cognition is usually a social process with more than one participant. The role of *multiple agents* has been taken seriously in the game-theoretic approach, but hardly in the other two (but cf. Halpern and Moses, 1985). Many-person versions of dynamic theories are needed, replacing programs by protocols for a group of distributed agents, and proofs by more interactive formats of reasoning. (Jaspars (1994) is an exploration.)

2.4.3. Resources

Our 'dynamic turn' crucially involves cognitive *resources*. There are no unlimited supplies of information or 'deductive energy', and logical analysis should bring out which mechanisms are adopted, and which cost in resources is incurred. This requires management of *occurrences* of assertions or instructions in proofs, programs and games. Stating a formula twice in a proof means two calls to its evidence, repeating the same instruction in a program calls for two executions, and repeating it in the course of a game will signal a new obligation as to its defense or attack. (Unlimited energy or standing commitment must be encoded explicitly via a logical 'repetition operator': (Girard, 1987; Van Benthem, 1993).) Thus, many recent logics work with occurrences, at a finer level of detail than the usual classical or intuitionistic calculi. (Moortgat (1996) and Buszkowski (1996, this Handbook), provide detailed linguistic motivation for this shift in emphasis.) Another form of fine-structure is *dependence*. Standard logics assume that all individuals under discussion can be freely introduced into discourse. But in general, some objects may depend on others (cf. Fine, 1985; Meyer Viol, 1995; Hintikka and Sandu, 1996, this Handbook), either 'in nature' or procedurally, in the course of dynamic interpretation. This further degree of freedom has interesting consequences. For example, on the usual proof-theoretic account, non-standard generalized quantifiers like *most* or *many* are difficult to analyze (Sundholm, 1986). But Van Lambalgen (1991) gives a Gentzen calculus with 'dependence management' for variables in quantifier rules to provide complete logics for non-standard quantifiers, where the classical ones become the limiting case with 'unlimited access'. Alechina (1995) is a more systematic study of various current dependence semantics with a dynamic flavor.

2.4.4. States and atomic actions

In this chapter, we have tried to identify some general strands in a process theory for natural language, at a suitable level of abstraction. In particular, no single notion of cognitive state can serve all of natural language. For instance, the DRT/DPL treatment of anaphora uses (partial or total) Tarskian variable assignments. Dynamic accounts of learning or updating have used probability functions over propositions, sets of worlds, states in Kripke models, or data bases. More complex syntactic discourse states occur in the computational literature. Nevertheless, useful general distinctions have emerged, such as that between *constructive* and *eliminative* views of information processing (cf. Landman, 1986), where epistemic states become 'richer' under updating in the former case, but 'simpler', by dropping alternatives, in the latter. (The two viewpoints may be combined in a dynamic epistemic logic; cf. (Jaspars, 1994).) Another general feature is 'dynamization'. Many update calculi may be viewed as 'dynamizations' of ordinary modal logics (cf. Van Benthem, 1991a), and standard extensional or intensional semantics may dynamicized through their natural parameters of variation (Cepparello, 1995). A final interesting issue is *combination* of different notions of state, with the resulting marriage of the corresponding logics, as in the merges of DPL and Update Semantics mentioned in Section 1.2.4.

Atomic actions in linguistics include testing of propositions, as well as the updating, contracting and revision found in the computational literature. Other speech acts have only been touched upon, such as questions (cf. Groenendijk and Stokhof, 1984). There is little uniformity in basic actions for different notions of state (compare assignment change versus updating), unless we either (i) move to a higher level of abstraction, where pops or pushes are general computational moves (Visser, 1994), or (ii) analyze atomic actions into a combination of 'modes' plus resultative static propositions, such as *test*(φ), *achieve*(φ), *query*(φ), where modes may be uniform across many different situations (Van Benthem, 1991a).

2.4.5. Dynamic operators and invariance

Which dynamic operations construct complex programs, plans, actions? One can approach this question at the level of linguistic items (programs, scripts) or their denotations (executions). There is great diversity here, witness our earlier survey. The proper definition for various dynamic operators is still under debate, witness the extensive discussion of an appropriate dynamic negation for natural language in (Dekker, 1993). Moreover, different dynamic paradigms may cut the cake in different ways. For example negation is less at home in the proof-theoretic perspective, unless one treats refutation on a par with proof (cf. Wansing, 1992). Likewise, negation as complement of programs is a marginal operation ("avoidance") – but negation as role switching is a crucial element in games. Another difference occurs with quantifiers, which are no longer on a par with propositional connectives in some dynamic semantics. They rather signal atomic moves establishing some binding or drawing some object, plus (in some cases) some further assertion following these. Thus, the syntax of the usual formal languages may even be misleading, in that it does not allow us to regard, say, a prefix $\exists x$ as an independent instruction by itself (say, 'pick an object ($\exists$), and assign it a temporary name (x)'). A more sensitive account of quantificational activity, involving changing data structures and bindings, was found in Section 2.2.

What we have outlined in this chapter is a general framework for the whole logical space of possibilities. Dynamic logic is all about control operators that combine procedures. Much dynamic semantics investigates what standard logical constants mean when viewed as procedural combinators. But dynamics allows for finer distinctions than statics, so that there may not be any clear sense to this. Standard conjunction really collapses several notions: sequential composition, but also various forms of parallel composition. Likewise, standard negation may be either some test as in DPL, or merely an invitation to make any move refraining from some forbidden action ("you can do anything, but don't step on my blue suede shoes"). Some natural operators in dynamic logic even lack classical counterparts altogether, such as 'conversion' or 'iteration' of procedures. One general perspective of *semantic invariance* relates the static and dynamic notions (Sher, 1991; Van Benthem, 1989). Truly logical operators do not depend on specific individuals in their arguments. This is also true for procedural operators. What makes, say, a complement $-R$ a logical negation is that it works uniformly on all ordered pairs (or arrows) in R, unlike an adjective like "clever" which depends on the content of its relational arguments. The mathematical generalization is *invariance under permutations* π of the underlying universe of individuals (here, information states). Dynamic procedures denote binary relations between states, and hence procedural operators satisfy the commutation schema:

$$\pi[O(R, S, \ldots)] = O(\pi[R], \pi[S], \ldots).$$

For a general type-theoretic formulation of this notion, cf. (Van Benthem, 1991a).

Permutation invariance leaves infinitely many potential dynamic logical constants. There are several ways of tightening up. One insists on suitable forms of *linguistic definability*. For instance, many dynamic operators have first-order definitions with variables over states and binary relation letters for procedures. We shall encounter this view-point in the next section. Another strengthening increases the demands on invariance, by requiring commutation for much looser forms of process equivalence than isomorphism over the same base domain. A typical example was the 'safety for bisimulation' discussed earlier.

2.4.6. Dynamic styles of inference

We have identified several dynamic styles of inference. These may still vary according to one's dynamic paradigm. The proof-theoretic perspective justifies an inference by composing it as the result of a number of basic moves. For instance, the basic inference $A \vee B, \neg A/B$ is a combination of argument by cases and one basic negation step:

$$\frac{A \vee B \quad \dfrac{A \quad \neg A}{B} \quad B}{B}.$$

In the programming perspective, this same inference would rather be viewed as an procedural update instruction:

> Updating any information state by $A \vee B$ and then by $\neg A$ (given some suitable procedural meaning for these operations) leads to a new information state which may be tested to validate B.

In the context of games, the story is different again. For instance, the 'agonistic' Lorenzen style would express the relevant validity as follows:

> There exists a winning strategy for defending the claim B in a dialogue game against any opponent who has already granted the two concessions $A \vee B, \neg A$.

One locus of difference here lies in the *structural rules* governing inference. Important examples are the admissibility, without loss of previous conclusions, of shuffling premises by Permutation, or of adding new premises by Monotonicity (Section 2.1.6 provided detailed formulations). For instance, Permutation is reasonable on both proof-theoretic and game-theoretic views, whereas it seems unreasonable on the programming view, since the sequential order of instructions is usually crucial to their total intended effect. Likewise, Monotonicity is plausible in games (the more concessions from one's opponent the better), but less so on the other two accounts. Still, if premise ordering in a game encodes priority of commitments incurred, then Permutation loses its appeal in the latter model too.

But also, analyzing cognitive activity via different interacting mechanisms raises issues of *logical architecture*. What systematic methods are available for switching between components (proof-theoretic, algorithmic, game-theoretic – and within these, between different facilities), and how do we transport information from one to the other? In other words, what are natural constructions of heterogeneous logical calculi? Some relevant material on these issues exists in the logical literature (cf. Gabbay, 1996), but no general theory exists.

2.4.7. Connections with computer science

Process Algebra views the denotations of procedures, not as binary relations, but rather as labeled transition models themselves (identified modulo bisimulation, or some other appropriate semantic equivalence). Some key references in this extensive field are (Milner, 1980; Bergstra and Klop, 1984; Baeten and Weyland, 1990). The result is a family of equational calculi for operations on, rather than inside, labeled transition systems. These provide abstract algebraic axiomatizations for various program constructions, including a much richer repertoire than what has been considered in dynamic semantics. (Examples are various parallel merges, as well as operators for 'hiding' structure, or for performing recursion.) For connections between Process Algebra and Dynamic Logic, see (Hennessy and Milner, 1985; Van Benthem and Bergstra, 1993; Van Benthem, Van Eijck and Stebletsova, 1993; Van Benthem, 1994). An eventual process theory for natural language may well have to be of this level of semantic sophistication.

2.4.8. Lowering complexity: Arrow logic and modal state semantics

One immediate concern in dynamic semantics is computational complexity. Many systems in Section 1 are supposed to mirror mechanisms in human cognition, and presumably, these procedures are geared towards speed and efficiency. Nevertheless, little is known about the complexity of various procedural logics – and what little is known, often makes their behaviour more complex than that of standard static systems (cf. Harel, 1984). For example, static propositional logic is decidable, relational algebra is not. Some recent logical proposals exist for coming to terms with such apparent paradoxes. We mention two of these.

Relational Algebra is not the only candidate for analyzing dynamic procedures. Intuitively, the latter seem to consist of transitions or *arrows* as objects in their own right. This alternative view is brought out in Arrow Logic, a modal logic over *arrow frames* $\langle W, C, R, I\rangle$ with a set W of arrows, a ternary relation C of *composition*, a binary relation R of *reversal* and a set I of *identical* arrows. Formulas φ will describe sets of arrows $\|\varphi\|$, i.e. transition relations in the new sense. Some key clauses in the basic truth definition are as follows.

$$\|\varphi \cap \psi\| = \|\varphi\| \cap \|\psi\|,$$

$$\|\varphi \circ \psi\| = \{a \mid \exists bc(\langle a,b,c\rangle \in C \ \&\ b \in \|\varphi\| \ \&\ c \in \|\psi\|)\},$$
$$\|\varphi^{\cup}\| = \{a \mid \exists b(\langle a,b\rangle \in R \ \&\ b \in \|\varphi\|)\},$$
$$\|\Delta\| = I.$$

Arrow Logic is a minimal theory of composition of actions, which may be studied by well-known techniques from Modal Logic (cf. Van Benthem, 1991a; Venema, 1991, 1994). Standard principles of Relational Algebra then express constraints on arrow patterns, which can be determined via *frame correspondences* (Van Benthem, 1985; De Rijke, 1993). For instance, the algebraic law $(\varphi \cup \psi)^{\cup} = (\varphi^{\cup} \cup \psi^{\cup})$ is a universally valid principle of modal distribution on arrow frames, but $(\varphi \cap \psi)^{\cup} = (\varphi^{\cup} \cap \psi^{\cup})$ expresses the genuine constraint that the conversion relation be a partial function f, whose idempotence would be expressed by the modal axiom $\varphi^{\cup\cup} = \varphi$. As an illustration, basic categorial laws of natural language (cf. Moortgat, 1996, this Handbook) now acquire dynamic content.

$$A \bullet (A\backslash B) \Rightarrow B \quad \textit{expresses that } \forall abc(\langle a,b,c\rangle \in C \to \langle c, f(b), a\rangle \in C).$$
$$(B/A) \bullet A \Rightarrow B \quad \textit{expresses that } \forall abc(\langle a,b,c\rangle \in C \to \langle b, a, f(c)\rangle \in C).$$

In particular, one can now study dynamic counterparts of the Lambek Calculus (cf. Kurtonina (1995) for a full development).

More radically, one can take this same deconstructionist line with respect to first-order predicate logic, the lingua franca of modern semantics – which suffers from undecidability. What makes first-order predicate logic tick at an abstract computational level? As we saw, the basic Tarski truth definition makes choices that are inessential to a compositional semantics for first-order quantification. In particular, concrete assignments and the concrete relation $a[x]b$ between assignments are not needed to make the semantic recursion work. The abstract core pattern that is needed replaces assignments by abstract states and the relations $[x]$ by arbitrary binary relations R_x between states. Models will then be poly-modal Kripke models $\langle S, \{R_x\}_{x \in \mathrm{VAR}}, V\rangle$, where S is the set of states and the valuation function V assigns a subset of S to each atomic sentence $R(x_1, \dots, x_n)$. The standard truth definition now generalizes to the following modal set-up.

(i′) $\|R(x_1, \dots, x_n)\| = V(R(x_1, \dots, x_n))$.

(ii′) $\|\neg\varphi\| = S - \|\varphi\|$,
$\|\varphi \vee \psi\| = \|\varphi\| \cup \|\psi\|$.

(iii′) $\|\exists x\varphi\| = \{a \in S \mid \exists b(aR_x b \ \&\ b \in \|\varphi\|)\}$.

This semantics treats existential quantifiers $\exists x$ as labeled modalities $\langle x\rangle$. Its universal validities constitute the well-known *minimal modal logic*, whose principles are (a) all classical propositional laws, (b) the axiom of Modal Distribution: $\exists x(\varphi \vee \psi) \leftrightarrow (\exists x\varphi \vee \exists x\psi)$, and (c) the rule of Modal Necessitation: *if* $\vdash \varphi$, *then* $\vdash \neg\exists x\neg\varphi$. A completeness theorem may be proved using the standard Henkin construction. This poly-modal logic can be analyzed in a standard fashion (Andréka, Van Benthem and Németi (1994) is a modern treatment), yielding the usual meta-properties such as the Craig Interpolation

Theorem, and the Łos–Tarski Preservation Theorem for submodels. In particular, the logic can be shown to be *decidable* via any of the usual modal techniques (such as filtration). This means that the particular set-theoretic implementation of the set S and the relations R_x that we find in the usual Tarski semantics can be diagnosed as the source of undecidability of elementary logic.

The modal perspective on classical logic uncovers a whole *fine-structure* of predicate-logical validity. The minimal predicate logic consists of those laws which are 'very much valid'. But we can analyze what other standard laws say too by the technique of modal *frame correspondence*. Here are some illustrations.

$(\varphi \wedge \exists x\varphi) \leftrightarrow \varphi$	*expresses that R_x is reflexive.*
$\exists x(\varphi \wedge \exists x\psi) \leftrightarrow (\exists x\varphi \wedge \exists x\psi)$	*expresses that R_x is transitive and euclidean,*
$\exists x\exists y\varphi \leftrightarrow \exists y\exists x\varphi$	*expresses that $R_x \circ R_y = R_y \circ R_x$,*
$\exists x\forall y\varphi \rightarrow \forall y\exists x\varphi$	*expresses that whenever aR_xbR_yc, there is a d such that aR_ydR_xc.*

The first two constraints make the R_x into equivalence relations, as with the modal logic S5. They do not impose existence of any particular states in frames. The third axiom, by contrast, is existential in nature; it says that sequences of state changes may be traversed in any order. Abstract state models need not have enough intermediate states to follow all alternative routes. The fourth example says that another well-known quantifier shift expresses a Church–Rosser property of computational processes. Thus, the valid laws of predicate logic turn have quite different dynamic content when analyzed in the light of this broader semantics.

We have found a minimal decidable system of predicate logic in addition to the standard undecidable one. Intermediate systems arise by varying requirements on states and updates R_x. Thus a whole landscape of intermediate predicate logics is opened up to us. Here, we seek expressive logics that share important properties with predicate logic (Interpolation, Effective Axiomatizability) and that even *improve* on this, preferably by being decidable. An attractive option, already known from Cylindric Algebra (cf. Henkin, Monk and Tarski, 1985; Németi, 1991) is CRS, the logic consisting of all predicate-logical validities in the state frames satisfying all *universal frame conditions* true in standard assignment models. These are the general logical properties of assignments that do not make existential demands on their supply. (The latter would be more 'mathematical' or 'set-theoretic'.) CRS is known to be decidable, though non-finitely axiomatizable. Moreover, its frame definition needs only universal *Horn* clauses, from which Craig Interpolation follows (Van Benthem, 1994). Another way of describing CRS has independent appeal. Consider state frames where S is a family of ordinary assignments (but not necessarily the full function space D^{VAR}), and the R_x are the standard relations $[x]$. Such frames admit 'assignment gaps', i.e. essentially they need not satisfy axiom AX1 of Section 2.3.2 above. This can be used to model dependencies between variables: changes in value for one variable x may induce, or be correlated with changes in value for another variable y (cf. our earlier discussion of resources). This phenomenon cannot be modeled in standard Tarskian semantics, the latter being a degenerate case where all interesting dependencies between variables have been suppressed. From CRS one can move upward in the hierarchy of logics by considering only families of assignments that satisfy natural closure conditions. Such further structure supports the introduction of further operators into the language (e.g., permutation or substitution operators). For the resulting logics, cf. (Marx, 1994; Mikulas, 1995).

2.4.9. Philosophical repercussions

We conclude with some sweeping thoughts. Dynamic paradigms suggest general cognitive claims. The programming model supports Church's Thesis which claims that any form of effective (cognitive) computation can be programmed on a Turing Machine, or some equivalent device from Recursion Theory. In its broader sense, the Turing Test is a well-known dramatized version. But similar claims can be made concerning proofs or games (in the setting of a suitably general Proof Theory or Game Theory), and that even in two ways. Church's Thesis may be interpreted as the *extensional* statement that the input-output behaviour of every effective function can be adequately programmed on some abstract machine. But it also has a stronger *intensional* version, stating that any algorithm can be reflected faithfully in some specific universal programming repertoire (cf. Moschovakis, 1991). This intensional question returns for proof-theoretic and game-theoretic approaches. What are their natural repertoires of logical constructions that should suffice for faithful modeling of any rational form of inference or cognitive play? (Compare the proof-theoretic functional completeness results in (Sundholm, 1986); or the hierarchies of programming operators in (Van Benthem, 1992).) There could also be 'Small Church Theses' at lower levels of computational complexity, closer to actual linguistic processing (cf. various equivalence results in (Kanovich, 1993)). Of course, one will have to analyze more carefully to which extent the computational metaphor is realistic for natural language (Fernando (1992) proposes recursion-theoretic models for this purpose). In this respect, another desideratum emerges. Our paradigms mostly provide *kinematics*: an extensional analysis of transitions made, whereas one eventually wants genuine *dynamics*: an account of the underlying processes, which explains observed transition behaviour. So far, much of logical semantics has had an extensional engineering flavor, following Lewis's (1972) dictum: *In order to say what a meaning* is, *we may first ask what a meaning* does, *and then find something that does that.*

References

Alchourrón, C., Gärdenfors, P. and Makinson, D. (1985), *On the logic of theory change: Partial meet functions for contraction and revision*, J. Symb. Logic 50, 510–530.

Alechina, N. (1995), *Modal quantifiers*, Dissertation, Institute for Logic, Language and Computation, University of Amsterdam.

Andréka, H., Van Benthem, J. and Németi, I. (1994), *Back and forth between modal logic and classical logic*, Mathematical Institute, Hungarian Academy of Sciences, Budapest/Institute for Logic, Language and Computation, University of Amsterdam, Report ILLC-ML-95-04. (To appear in Bulletin of the Interest Group for Pure and Applied Logic, London and Saarbrücken, 1995.)

Asher, N. (1993), *Reference to Abstract Objects in Discourse*, Kluwer, Dordrecht.

Baeten, J. and Weyland, P. (1990), *Process Algebra*, Cambridge Univ. Press, Cambridge, MA.

Barr, M. and Wells, C. (1989), *Category Theory for Computing Science*, Prentice-Hall, New York.

Barwise, J. (1987), *Noun phrases, generalized quantifiers and anaphora*, Generalized Quantifiers. Logical and Linguistic Approaches, P. Gärdenfors, ed., Reidel, Dordrecht, 1–29.

Beaver, D. (1992), *The kinematics of presupposition*, Proceedings of the Eighth Amsterdam Colloquium, P. Dekker and M. Stokhof, eds, Institute for Logic, Language and Computation, University of Amsterdam, 17–36.

Beaver, D. (1996), *Presupposition*, This Handbook, Chapter 17.

Belnap, N. (1977), *A useful four-valued logic*, Modern Uses of Multiple-Valued Logics, J.M. Dunn and G. Epstein, eds, Reidel, Dordrecht, 8–37.

Bergstra, J. and Klop, J.-W. (1984), *Process algebra for synchronous communication*, Inform. and Control **60**, 109–137.

Blackburn, P. and Venema, Y. (1993), *Dynamic squares*, Logic Preprint 92, Department of Philosophy, University of Utrecht. (J. Philos. Logic, to appear.)

Blamey, S. (1986), *Partial logic*, Handbook of Philosophical Logic vol. III, D. Gabbay and F. Günthner, eds, 1–70.

Bos, J., Mastenbroek, E., McGlashan, S., Millies, S. and Pinkal, M. (1994), *A compositional DRS-based formalism for NLP applications*, Proceedings International Workshop on Computational Semantics, H. Bunt, R. Muskens and G. Rentier, eds, Institute for Language Technology and Artificial Intelligence, Tilburg, 21–31.

Boutilier, C. (1993), *Revision sequences and nested conditionals*, Proceedings of the 13th IJCAI, R. Bajcsy, ed., Morgan Kaufmann, Washington, DC, 519–525.

Boutilier, C. and Goldszmidt, M. (1993), *Revision by conditional beliefs*, Proceedings of the 11th National Conference on Artificial Intelligence (AAAI), Morgan Kaufmann, Washington, DC, 649–654.

Brink, C., Britz, K. and Schmidt, R. (1992), *Peirce algebras*, Report MPI-I-92-229, MPI, Saarbrücken.

Buszkowski, W. (1996), *Mathematical linguistics and proof theory*, This Handbook, Chapter 12.

Cepparello, G. (1995), *Dynamics: Logical design and philosophical repercussions*, Dissertation, Scuola Normale Superiore, Pisa.

Chierchia, G. (1988), *Dynamic Generalized Quantifiers and Donkey Anaphora*, Genericity in Natural Language, M. Krifka, ed., Tübingen, SNS, 53–84.

Davidson, D. (1967), *The Logical Form of Action Sentences*, Reprinted: D. Davidson, 1980, Essays on Actions and Events, Clarendon Press, Oxford.

Dekker, P. (1993), *Transsentential meditations*, ILLC Dissertation Series 1993-1, Institute for Logic, Language and Computation, University of Amsterdam.

Doets, K. (1993), *Model theory*, Lecture Notes for the Fifth European Summer School in Logic, Language and Information, University of Lisbon.

Dummett, M. (1976), *What is a theory of meaning?*, Truth and Meaning, G. Evans and J. McDowell, eds, Oxford Univ. Press, Oxford, 67–137.

Fernando, T. (1992), *Transition systems and dynamic semantics*, Logics in AI, LNCS 633, Springer, Berlin.

Fine, K. (1985), *Reasoning With Arbitrary Objects*, Blackwell, Oxford.

Gabbay, D. (1994), *Labeled Deductive Systems*, Oxford Univ. Press, Oxford. (To appear.)

Gabbay, D. and Kempson, R. (1992), *Natural language content: A proof-theoretic perspective*, Proceedings of the Eighth Amsterdam Colloquium, P. Dekker and M. Stokhof, eds, Institute for Logic, Language and Computation, University of Amsterdam, 173–195.

Gärdenfors, P. (1988), *Knowledge in Flux. Modelling the Dynamics of Epistemic States*, MIT Press, Cambridge, MA.

Gärdenfors, P. and Makinson, D. (1988), *Revisions of knowledge systems using epistemic entrenchment*, Theoretical Aspects of Reasoning about Knowledge, M. Vardi, ed., Morgan Kaufmann, Los Altos, CA, 83–95.

Gazdar, G. (1979), *Pragmatics*, Academic Press, New York.

Girard, J.-Y. (1987), *Linear logic*, Theor. Comput. Sci. **50**, 1–102.

Goldblatt, R. (1987), *Logics of Time and Computation*, CSLI Lecture Notes, Chicago Univ. Press, Chicago.

Groenendijk, J. and Stokhof, M. (1984), *Studies in the semantics of questions and the pragmatics of answers*, Doctoral Dissertation, University of Amsterdam.

Groenendijk, J. and Stokhof, M. (1990), *Dynamic Montague grammar*, Papers from the Second Symposium on Logic and Language, L. Kálmán and L. Pólos, eds, Akadémiai Kiadó, Budapest, 3–48.

Groenendijk, J. and Stokhof, M. (1991), *Dynamic predicate logic*, Ling. and Philos. **14**, 39–100.

Groenendijk, J., Stokhof, M. and Veltman, F. (1996), *Coreference and modality*, The Handbook of Contemporary Semantic Theory, S. Lappin, ed., Blackwell, Oxford, 179–214.

Grosz, B. and Sidner, C. (1986), *Attention, intention, and the structure of discourse*, Comput. Ling. **12**, 175–204.

Halpern, J. and Moses, Y. (1985), *Towards a theory of knowledge and ignorance*, Logics and Models of Concurrent Systems, K. Apt, ed., Springer, Berlin, 459–476.

Harel, D. (1984), *Dynamic logic*, Handbook of Philosophical Logic vol. II, D. Gabbay and F. Günthner, eds, Reidel, Dordrecht, 497–604.

Harel, D. and Kozen, D. (1994), *Dynamic logic*, Department of Computer Science, Technion, Haifa/Department of Computer Science, Cornell University.

Heim, I. (1982), *The semantics of definite and indefinite noun phrases*, Dissertation, Univ. of Massachusetts, Amherst, published in 1989 by Garland, New York.

Heim, I. (1983a), *On the projection problem for presuppositions*, Proceedings of the West Coast Conference on Formal Linguistics vol. II, Stanford Linguistic Association, Stanford, CA, 114–125, Reprinted in: S. Davies (ed.) (1991), *Pragmatics*, OUP, Oxford, 397–405.

Heim, I. (1983b), *File change semantics and the familiarity theory of definiteness*, Meaning, Use and Interpretation of Language, R. Bäuerle, C. Schwarze and von A. Stechow, eds, De Gruyter, Berlin.

Henkin, L., Monk, D. and Tarski, A. (1985), *Cylindric Algebra*, Part II, North-Holland, Amsterdam.

Hennessy, M. and Milner, R. (1985), *Algebraic laws for nondeterminism and concurrency*, J. Assoc. Comput. Mach. **32**, 137–161.

Hintikka, J. (1973), *Logic, Language Games and Information*, Clarendon Press, Oxford.

Hintikka, J. and Sandu, G. (1996), *Game-theoretical semantics*, This Handbook, Chapter 6.

Hofstadter, D. (1980), *Gödel, Escher, Bach: An Eternal Golden Braid*, Vintage Books, New York.

Jaspars, J. (1994), *Calculi for constructive communication*, ILLC Dissertation Series 1994-1, Institute for Logic, Language and Computation, University of Amsterdam/Institute for Language Technology and Artificial Intelligence, Tilburg University.

Kameyama, M. (1992), *The linguistic information in dynamic discourse*, Research Report CSLI-92-174, Center for the Study of Language and Information, Stanford University.

Kamp, H. (1981), *A theory of truth and semantic representation*, Truth, Interpretation and Information, J. Groenendijk et al., eds, Foris, Dordrecht, 1–41.

Kamp, H. and Reyle, U. (1993), *From Discourse to Logic*, Kluwer, Dordrecht.

Kanazawa, M. (1993a), *Completeness and decidability of the mixed style of inference with composition*, Proceedings of the Ninth Amsterdam Colloquium, P. Dekker and M. Stokhof, eds, Institute for Logic, Language and Computation, University of Amsterdam, 377–390.

Kanazawa, M. (1993b), *Dynamic generalized quantifiers and monotonicity*, Report LP-93-02, Institute for Logic, Language and Computation, University of Amsterdam.

Kanovich, M. (1993), *The expressive power of modalized purely implicational calculi*, Report CSLI-93-184, Center for the Study of Language and Information, Stanford University.

Karttunen, L. (1973), *Presuppositions of compound sentences*, Ling. Inq. **4**, 167–193.

Karttunen, L. (1974), *Presupposition and linguistic context*, Theor. Ling. **1**, 181–194.

Karttunen, L. (1976), *Discourse referents*, Syntax and Semantics 7: Notes from the Linguistic Underground, J. McCawley, ed., Academic Press, New York, 363–385.

Karttunen, L. and Peters, S. (1979), *Conventional implicature*, Syntax and Semantics 11: Presupposition, C.-K. Oh and D. Dinneen, eds, Academic Press, New York, 1–56.

Keenan, E. and Westerståhl, D. (1996), *Quantifiers*, This Handbook, Chapter 15.

Keenan, E. and Faltz, L. (1985), *Boolean Semantics for Natural Language*, Reidel, Dordrecht.

Kneale, W. and M. Kneale (1962), *The Development of Logic*, Clarendon Press, Oxford.

Krahmer, E. (1995), *Discourse and presupposition: From the man in the street to the king of France*, Doctoral Dissertation, Tilburg University.

Kurtonina, N. (1995), *Frames and labels. A logical investigation of categorial structure*, Dissertation, Onderzoeksinstituut voor Taal en Spraak, Universiteit Utrecht.

Lakatos, I. (1976), *Proofs and Refutations*, Cambridge Univ. Press, Cambridge.

Landman, F. (1986), *Towards a Theory of Information. The Status of Partial Objects in Semantics*, Foris, Dordrecht.

Lewis, D. (1972), *General semantics*, Semantics of Natural Language, D. Davidson and G. Harman, eds, Reidel, Dordrecht, 169–218.

Lewis, D. (1973), *Counterfactuals*, Blackwell, Oxford.

Lewis, D. (1979), *Score keeping in a language game*, J. Philos. Logic **8**, 339–359.

Lorenzen, P. (1959), *Ein dialogisches Konstruktivitätskriterium*, Lecture reprinted: Lorenzen, P. and Lorenz, K. (1978), *Dialogische Logik*, Wissenschaftliche Buchgesellschaft, Darmstadt.

Makinson, D. (1985), *How to give it up: A survey of some formal aspects of the logic of theory change*, Synthese **62**, 347–363.

Manes, E. and Arbib, M. (1975), *Arrows, Structures and Functors, the Categorical Imperative*, Academic Press, New York.

Marx, M. (1994), *Arrow logic and relativized algebras of relations*, Dissertation, CCSOM, Faculty of Social Sciences/Institute for Logic, Language and Computation, University of Amsterdam.

McLane, S. (1971), *Categories for the Working Mathematician*, Springer, Berlin.

Meyer Viol, W. (1995), *Instantial logic*, Dissertation, Onderzoeksinstituut voor Taal en Spraak, Universiteit Utrecht.

Milner, R. (1980), *A Calculus of Communicating Systems*, Springer, Berlin.

Montague, R. (1970), *Universal grammar*, Reprinted: Montague, R. (1974), *Formal Philosophy*, Yale Univ. Press, New Haven, CT, 222–246.

Moortgat, M. (1996), *Categorial grammar*, This Handbook, Chapter 2.

Moschovakis, Y. (1991), *Sense and reference as algorithm and value*, Department of Mathematics, University of California, Los Angeles.

Muskens, R. (1991), *Anaphora and the logic of change*, JELIA '90, European Workshop on Logics in AI, J. van Eijck, ed., Springer Lecture Notes, Springer, Berlin, 414–430.

Muskens, R. (1994), *Categorial grammar and discourse representation theory*, Proceedings of COLING 94, Kyoto, 508–514.

Muskens, R. (1995a), *Tense and the logic of change*, Lexical Knowledge in the Organization of Language, U. Egli, P.E. Pause, C. Schwarze, A. von Stechow and G. Wienold, eds, Benjamin, Amsterdam, 147–183.

Muskens, R. (1995b), *Combining Montague semantics and discourse representation*, Ling. and Philos. **19**, 143–186.

Németi, I. (1991), *Algebraizations of Quantifier Logics: An Introductory Overview*, Mathematical Institute, Hungarian Academy of Sciences, Budapest.

Parsons, T. (1990), *Events in the Semantics of English*, MIT Press, Cambridge, MA.

Peters, S. (1975), *A truth-conditional formulation of Karttunen's account of presuppositions*, Texas Linguistic Forum, University of Texas, Austin, TX, 137–149.

Polanyi, L. (1985), *A theory of discourse structure and discourse coherence*, Papers from the General Session of the Chicago Linguistic Society, CLS vol. 21, 306–322.

Popper, K. (1959), *The Logic of Scientific Discovery*, Hutchinson, London.

Pratt, V. (1976), *Semantical considerations on Floyd–Hoare logic*, Proc. 17th IEEE Symp. on Foundations of Computer Science, 109–121.

Ramsey, F.P. (1929), *General propositions and causality*, Reprinted: F.P. Ramsey, *Foundations: Essays in Philosophy, Logic, Mathematics and Economics*, Routledge and Kegan Paul, London, 1978.

Ranta, A. (1991), *Intuitionistic categorial grammar*, Ling. and Philos. **14**, 203–239.

Reyes, G.E. and Macnamara, J. (1994), *The Logical Foundations of Cognition*, Oxford Univ. Press, New York/Oxford.

De Rijke, M. (1993), *Extending modal logic*, Dissertation Series 1993–4, Institute for Logic, Language and Computation, University of Amsterdam.

Rott, H. (1992), *Preferential belief change using generalized epistemic entrenchment*, J. Logic, Lang., Inform. **1**, 45–78.

Ryan, M. (1992), *Ordered presentations of theories: Default reasoning and belief revision*, PhD Thesis, Department of Computing, Imperial College, University of London.

Scha, R. and Polanyi, L. (1988), *An augmented context free grammar for discourse*, Proceedings of the 12th International Conference on Computational Linguistics, Budapest.

Scott, D.S. (1982), *Domains for denotational semantics*, Proceedings 9th International Colloquium on Automata, Languages and Programming, M. Nelsen and E.T. Schmidt, eds, Lecture Notes Comput. Sci. vol. 140, Springer, Berlin, 577–613.

Segerberg, K. (1982), *A completeness theorem in the modal logic of programs*, Universal Algebra and Applications, T. Traczyk, ed., Banach Centre Publications 9, PWN – Polish Scientific, Warsaw, 31–46.

Segerberg, K. (1991), *Logics of action*, Abstracts 9th International Congress on Logic, Methodology and Philosophy of Science, Uppsala.

Seuren, P. (1975), *Tussen Taal en Denken*, Scheltema, Holkema en Vermeulen, Amsterdam.

Seuren, P. (1985), *Discourse Semantics*, Blackwell, Oxford.

Sher, G. (1991), *The Bounds of Logic. A Generalised Viewpoint*, Bradford Books/MIT Press, Cambridge, MA.

Shoham (1988), *Reasoning about Change. Time and causation from the standpoint of artificial intelligence*, Yale Univ. Press, New Haven, CT.

Soames, S. (1989), *Presupposition*, Handbook of Philosophical Logic vol. IV, D. Gabbay and F. Günthner, eds.

Spohn, W. (1988), *Ordinal conditional functions: A dynamic theory of epistemic states*, Causation in Decision, Belief Change and Statistics II, W.L. Harper et al., eds, Kluwer, Dordrecht, 105–134.

Stalnaker, R. (1968), *A theory of conditionals*, Studies in Logical Theory, N. Rescher, ed., Basil Blackwell, Oxford, 98–112.

Stalnaker, R. (1974), *Pragmatic presuppositions*, Semantics and Philosophy, M. Munitz and P. Unger, eds, New York Univ. Press, New York, 197–213.

Stalnaker, R. (1979), *Assertion*, Syntax and Semantics 9: Pragmatics, P. Cole, ed., Academic Press, New York, 315–332.

Sundholm, G. (1986), *Proof theory and meaning*, Handbook of Philosophical Logic vol. III, D. Gabbay and F. Guenthner, eds, Reidel, Dordrecht, 471–506.

Suppe, F. (1977), *The Structure of Scientific Theories*, Univ. of Illinois Press, Urbana, IL.

Troelstra, A. and Van Dalen, D. (1988), *Constructivism in Mathematics*, two volumes, North-Holland, Amsterdam.

Turner, R. (1996), *Types*, This Handbook, Chapter 9.

Van Benthem, J. (1986), *Essays in Logical Semantics*, Studies in Linguistics and Philosophy vol. 29, Reidel, Dordrecht.

Van Benthem, J. (1988), *Games in logic: A survey*, Representation and Reasoning, J. Hoepelman, ed., Niemeyer Verlag, Tübingen, 3–15.

Van Benthem, J. (1989), *Semantic parallels in natural language and computation*, Logic Colloquium. Granada 1987, H.-D. Ebbinghaus et al., eds, North-Holland, Amsterdam, 331–375.

Van Benthem, J. (1991), *Language in Action. Categories, Lambdas and Dynamic Logic*, North-Holland, Amsterdam.

Van Benthem, J. (1991a), *General dynamics*, Theor. Ling. **17**, 159–201.

Van Benthem, J. (1993), *Logic and the flow of information*, Proceedings 9th International Congress of Logic, Methodology and Philosophy of Science. Uppsala 1991, D. Prawitz, B. Skyrms and D. Westerståhl, eds, Elseviers, Amsterdam, 693–724.

Van Benthem, J. (1993a), *Modeling the kinematics of meaning*, Proceedings Aristotelean Society 1993, 105–122.

Van Benthem, J. (1993b), *Programming operations that are safe for bisimulation*, Report 93-179, Center for the Study of Language and Information, Stanford University. (To appear in *Logic Colloquium. Clermont-Ferrand 1994*, Elsevier, Amsterdam.)

Van Benthem, J. (1994a), *Dynamic Arrow logic*, Dynamic Logic and Information Flow, J. van Eijck and Visser, eds, MIT Press, Cambridge, MA.

Van Benthem, J. (1994b), *Modal foundations for predicate logic*, Research Report CSLI-94-191, Center for the Study of Language and Information, Stanford University.

Van Benthem, J. and Cepparello, G. (1994), *Tarskian variations. Dynamic parameters in classical semantics*, Technical Report CS-R9419, CWI, Amsterdam.

Van Benthem, J. and Bergstra, J. (1993), *Logic of transition systems*, Report CT-93-03, Institute for Logic, Language and Computation, University of Amsterdam. (To appear in J. Logic, Lang., Inform.)

Van Benthem, J., Van Eijck, J. and Frolova, A. (1993), *Changing preferences*, Technical Report CS-R9310, CWI, Amsterdam.

Van Benthem, J., Van Eijck, J. and Stebletsova, V. (1993), *Modal logic, transition systems and processes*, Logic Comput. **4**(5), 811–855.

Van den Berg, M. (1995), *Plural dynamic generalized quantifiers*, Dissertation, Institute for Logic, Language and Computation, University of Amsterdam.

Van Deemter, K. (1991), *On the composition of meaning*, Dissertation, Institute for Logic, Language and Information, University of Amsterdam.

Van der Does, J. (1992), *Applied quantifier logics*, Dissertation, Institute for Logic, Language and Computation, University of Amsterdam.

Van Eijck, J. and G. Cepparello (1994), *Dynamic modal predicate logic*, Dynamics, Polarity and Quantification, M. Kanazawa and C. Piñon, eds, CSLI, Stanford, 251–276.

Van Eijck, J. and Visser, A. (eds) (1994), *Dynamic Logic and Information Flow*, MIT Press, Cambridge, MA.

Van Eijck, J. and De Vries, F.-J. (1992), *Dynamic interpretation and Hoare deduction*, J. Logic, Lang., Inform. **1**, 1–44.

Van Eijck, J. and De Vries, F.-J. (1995), *Reasoning about update logic*, J. Philos. Logic **24**, 19–45.

Van Eijck, J. and Kamp, H. (1996), *Representing discourse in context*, This Handbook, Chapter 3.

Van Lambalgen, M. (1991), *Natural deduction for generalized quantifiers*, Generalized Quantifiers: Theory and Applications, J. van der Does and J. van Eijck, eds, Dutch PhD Network for Logic, Language and Information, Amsterdam, 143–154. (To appear with Cambridge Univ. Press.)

Veltman, F. (1985), *Logics for conditionals*, Dissertation, University of Amsterdam.

Veltman, F. (1991), *Defaults in update semantics*, Report LP-91-02, Institute for Logic, Language and Computation, University of Amsterdam. (To appear in the J. Philos. Logic.)

Venema, Y. (1991), *Many-dimensional modal logic*, Dissertation, Institute for Logic, Language and Computation, University of Amsterdam.

Venema, Y. (1994), *A crash course in Arrow logic*, Knowledge Representation and Reasoning under Uncertainty, Logic at Work, M. Masuch and L. Polos, eds, Lecture Notes in Artificial Intelligence vol. 808, Springer, Berlin.

Vermeulen, K. (1994), *Exploring the dynamic environment*, Dissertation, Onderzoeksinstituut voor Taal en Spraak, University of Utrecht.

Visser, A. (1994), *Actions under presuppositions*, Logic and Information Flow, J. van Eijck and A. Visser, eds, MIT Press, Cambridge, MA.

Visser, A. and Vermeulen, K. (1995), *Dynamic bracketing and discourse representation*, Logic Group Preprint Series 131, Department of Philosophy, University of Utrecht. (To appear in Notre Dame J. Formal Logic, special issue *Logical Aspects of Complex Structures*.)

Wansing, H. (1992), *The logic of information structures*, Dissertation, Department of Philosophy, Free University, Berlin.

Westerståhl, D. (1984), *Determiners and context sets*, Generalized Quantifiers in Natural Language, J. van Benthem and A. ter Meulen, eds, Foris, Dordrecht, 45–71.

Wittgenstein, L. (1953), *Philosophische Untersuchungen*, edited by G. Anscombe and R. Rhees with an English translation by G. Anscombe, Blackwell, Oxford.

Zeinstra, L. (1990), *Reasoning as discourse*, Master's Thesis, Department of Philosophy, University of Utrecht.