THEORETICAL LINGUISTICS PROGRAMME, BUDAPEST UNIVERSITY (ELTE)

DYNAMIC UPDATE PREDICATE LOGIC

LÁSZLÓ KÁLMÁN AND GÁBOR RÁDAI

RESEARCH INSTITUTE FOR LINGUISTICS, HUNGARIAN ACADEMY SCIENCES

WORKING PAPERS IN THE THEORY OF GRAMMAR, VOL. 2, NO. 6

RECEIVED: DECEMBER 1995

DYNAMIC UPDATE PREDICATE LOGIC

László Kálmán and Gábor Rádai

Theoretical Linguistics Programme, Budapest University (ELTE)
Research Institute for Linguistics, HAS, Room 120
Budapest I., P.O. Box 19. H-1250 Hungary
E-mail: kalman@nytud.hu, radai@nytud.hu

Working Papers in the Theory of Grammar, Vol. 2, No. 6
Supported by the Hungarian National Research Fund (OTKA)

Theoretical Linguistics Programme, Budapest University (ELTE)
Research Institute for Linguistics, Hungarian Academy of Sciences

Budapest I., P.O. Box 19. H-1250 Hungary
Telephone: (36-1) 175 8285; Fax: (36-1) 212 2050

DYNAMIC UPDATE PREDICATE LOGIC

László Kálmán
AND
Gábor Rádai
Theoretical Linguistics Programme, Budapest University
AND
Research Institute for Linguistics, Hungarian Academy of Sciences
kalman@nytud.hu, radai@nytud.hu

0. Introduction

In current (dynamic) theories of semantics, such as Groenendijk and Stokhof's (1991) **Dynamic Predicate Logic** (**DPL**), anaphoric expressions are treated as bound variables. The main drawback of this approach is that it relegates the way in which anaphor/antecedent relations are determined to some external (and notoriously unspecified) module. For example, consider:

(1) *Joe has a cat.* It *is black.*
'$\exists x(\text{cat}(x) \wedge \text{owns}(j, x)) \wedge \text{black}(x)$'

The DPL formula in the above translation is interpreted as if the variable x in black(x) was bound by the existential quantifier in the first conjunct. However, why x rather than any other variable appears in the translation of the second sentence is outside the scope of the theory.

No doubt, anaphoric reference has certain aspects which, in all probability, cannot be put in semantic terms. Those aspects may be governed by genuinely formal (e.g., syntactic) properties of the utterances in which the anaphors appear, and there is not much hope that we can explain them within semantics.[1] But the fact that some properties of anaphoric binding fall outside the scope of semantics does not justify a treatment in which anaphor/antecedent relations are determined by mechanisms entirely independent from semantics, by pure magic, as it were. In particular, an anaphor always requires that there be exactly one salient individual

[1] For example, the gender agreement between the anaphor and its antecedent often has no semantic counterpart, because grammatical gender is a purely formal feature in many languages. The other well-known case when anaphoric relations depend on formal factors is related to the distinction between **anaphoric** vs. **reflexive** pronouns in languages like English:

in the context that can serve as its antecedent, at least among the candidates not excluded by formal factors (such as gender and syntactic structure). Since this is a fact about the **semantics** of anaphors, it should be captured by semantics. On the other hand, anaphors lexically carry some **descriptive content** which constrains the range of their possible antecedents, and those constraints are also semantic.

In what follows, we will develop a dynamic theory of anaphors which could account for both the uniqueness requirement and the descriptive content of anaphors, as an alternative to the bound-variable view. Our treatment is an attempt to formally develop, in a dynamic framework, ideas on anaphoric binding that were proposed by various researchers as early as Cooper (1979) and Evans (1980), and which have been informally proposed in dynamic semantics by Kálmán (1995) and Groenendijk *et al.* (1995).

Given the fact that the view of information states as sets of assignment functions in DPL is motivated by the bound-variable view of anaphors that we are criticizing, we will have to develop a different concept of information states in the first place (**Section 1.1**). If we adopt this new concept, we can store what entities have been introduced in the discourse, without knowing what variables have been used for introducing them. There is no reason why we should keep track of discourse referents by arbitrary names (i.e., variables in the domains of assignment functions) if we do not want to use those names to refer to them later on. Then,

---

(2) *Joe likes* him.
*'Joe likes himself'

(3) *Joe reminded Peter of* him.
*'Joe reminded Peter of himself'

In these sentences, the anaphoric pronoun *him* may not be co-referential with any other noun phrase in the sentence. The fact that co-reference has to be expressed by a reflexive pronoun in these syntactic configurations probably cannot be expressed or explained in semantic terms. Finally, the syntactic and rhetorical structure of an utterance sometimes biasses anaphoric relations without fully determining them:

(4) *Whenever Joe meets Peter,* he *greets* him.

(5) *Joe didn't recognize Peter, because* he *had shaved his beard.*

The preferred reading of (4) is when the grammatical role of each anaphor matches that of its antecedent. In (5), on the other hand, the best way of ensuring the coherence of the discourse is to take *he* to be co-referential with *Peter* rather than *Joe*, because this is the simplest way of establishing the desired rhetorical relation between the two clauses. For details on this mechanism see Polanyi (1988) and Prüst (1992).

in **Section 1.2.1**, we define a first-order language interpreted dynamically in which quantification over the discourse universe can be expressed in an elegant way. This is necessary because the uniqueness of the antecedents of anaphors must be satisfied in the discourse universe rather than the entire model. After examining certain logical properties of the resulting system (**Section 1.2.2**), we 'partialize' that language to account for presuppositions (**Section 1.2.3**). The reason for this is that the requirements that anaphors impose on their antecedent are of a presuppositional character. We also examine the most essential logical properties of the system introduced. Finally, in **Section 2**, we explain the consequences of our treatment for various phenomena related to anaphors, such as donkey sentences and the interaction of anaphors with modal operators.

1. Dynamic Update Predicate Logic

Possible alternatives to the bound-variable approach to anaphors must treat anaphors as **quantificational**. Consider:

(1) *Joe has a cat.* It *is black.*

Under the quantificational view, the anaphoric pronoun *it* in the above piece of discourse must be interpreted as the condensed form of a definite description such as 'the non-human individual', which involves a quantifier ('there is exactly one non-human individual') that is possibly presuppositional. We will call this type of quantification **anaphoric quantification**.

The interesting fact about expressions of anaphoric quantification is that they constitute **unstable** propositions in the sense of Veltman (1981). That is, for instance, 'there is exactly one non-human individual' may be true in the context in which (1) is uttered, but false in a subsequent context, in which more than one non-human individual has been introduced. The only dynamic semantic theory in which such unstable propositions exist is Veltman's (1981, 1990) **Update Semantics**. Update Semantics, however, uses a propositional logic, so it cannot express quantification over individuals. The only possible source of instability in Veltman (1990) is the **possibility** operator. Formulae of the form '$\Diamond\varphi$' (read as 'might φ', and interpreted in an epistemic manner) may be true in an information state that does not exclude the truth of φ, but can be falsified by subsequent information. We will introduce another type of unstable propositions, namely, those involving anaphoric quantification. For example, the anaphoric version of the quantifier 'there is exactly one' quantifies over the domain of entities already introduced in a context. Thus, it gives rise to **non-upward-entailing** quantification, which may become false from true as we enlarge the universe of discourse.

The instability of anaphoric quantification will manifest itself in the properties of the function that assigns a **truth value** to every formula and information state.

For example, we want to say that 'there is exactly one non-human individual' is true in an information state if and only if the existence of exactly one non-human individual can be taken for granted in the given information state. Thus, a formula may be true in an information state without being true in the actual world, even if the information state is true to the world. On the other hand, if 'there is exactly one non-human individual' is true in a truthful information state and, moreover, in all its **extensions** as well (i.e., in the more informative information states that may arise from it), then this proposition must indeed be true in the world. Obviously, the meaning of an anaphor like *it* does not entail that the world contains just one non-human individual.

The system that we will use in the following is designed to be the simplest one that can express the above concepts (including Veltman's (1990) concept of 'updating'). It is called **Dynamic Update Predicate Logic** (**DUPLO** for short), to express its close kinship to both Update Semantics and DPL. In the rest of this section we first explain the concept of an information state in DUPLO. Then we describe the semantics of DUPLO formulae, which are first-order formulae with a possibility operator '$\Diamond$' similar to Veltman's (1990) and a presupposition operator '$\triangle$' similar to Beaver's (1992).

1.1. Information States in DUPLO

As can be expected from the above, information states in DUPLO do not store possible values of variables (assignments are parameters of the semantic-value functions as in classical, static logic), but they do keep track of what entities have been mentioned and what has been learnt about them. An information state in DUPLO consists of **possibilities** each of which carries information about what the world may be like. But possibilities are not complete descriptions of the world, they are partial in the sense that they express that certain facts are known to be true about some entities the existence of which is suggested by previous discourse, but not **every** fact is known about them.[2]

The way in which we will represent possibilities, i.e., partial information about the world, is the following. Each possibility contains a number of alternatives, some of them more complete than the others, possibly with complete possible worlds at one end of the scale. These alternatives will be called **model fragments**. A model fragment is exactly like a (first-order) model, except that its universe may be empty. It is like a subset of the universe of the model, plus an interpretation function that is restricted to that universe.

[2] In what follows, we will assume that the interpretation functions are first-order, i.e., they just assign a set of n-tuples to each n-ary predicate constant. Extending information states to higher-order interpretation functions would raise no problem at all.

(6) **Definition: Model fragments**
A model fragment f is an ordered pair $\langle \mathcal{U}_f, \mathcal{I}_f \rangle$ such that $\mathcal{U}_f$, the **universe** of f, is a set of individuals (taken from a countable set $\mathcal{U}$ containing all individuals), and the **interpretation function** of f, $\mathcal{I}_f$, which assigns a set of n-tuples of individuals in $\mathcal{P}(\mathcal{U}_f^n)$ to every n-ary predicate constant. We will refer to the entire set of model fragments satisfying these constraints as $\mathcal{F}$.

We also need the concept of the **informativity** of model fragments:

(7) **Definition: Informativity of model fragments**
A model fragment $f_1 = \langle \mathcal{U}_{f_1}, \mathcal{I}_{f_1} \rangle$ is **at least as informative as** the model fragment $f_2 = \langle \mathcal{U}_{f_2}, \mathcal{I}_{f_2} \rangle$ (written: $f_1 \sqsubseteq f_2$) iff
(i) $\mathcal{U}_{f_2} \subseteq \mathcal{U}_{f_1}$, and
(ii) for every predicate constant P, $\mathcal{I}_{f_2}(P) \subseteq \mathcal{I}_{f_1}(P)$.

Every possibility p in an information state π is a set of model fragments. We will talk about the **core** of p, which is the set of its least informative elements:

(8) **Definition: Core of a possibility**
If $p \subseteq \mathcal{F}$ is a possibility, then the **core** of p (written: $\mathcal{C}(p)$) is

$$\mathcal{C}(p) = \{f \in p \colon \bigwedge_{f' \in p} (f \sqsubseteq f' \Rightarrow f' = f)\}.$$

The universes of the core elements of a possibility represent potential discourse universes. When updating an information state in such a way that a new individual is mentioned in the discourse, the cardinality of every core element of every possibility will increase by one. We will refer to this process informally as 'extending the discourse universe'.

We will also need the set of model fragments in a possibility p that are at least as informative as a certain model fragment $f \in p$:

(9) **Definition: The cloak of a model fragment**
If p is a possibility in an information state, and $f \in p$ is a model fragment, then **the cloak of** f **in** p (written $\downarrow_p f$) is

$$\downarrow_p f =_{\text{def}} \{f' \in p \colon f' \sqsubseteq f\}.$$

Finally, we will use the concept of the set of the **least informative extensions** of a possibility p satisfying Φ:

$$\text{MAX}_{p' \subseteq p}(\Phi) =_{\text{def}} \{p' \subseteq p \colon \Phi \,\&\, \bigwedge_{p'' \subseteq p} ((\Phi[p'/p''] \,\&\, p' \subseteq p'') \Rightarrow p'' = p)$$

where $\Phi[p'/p'']$ is the same as Φ, with the free occurrences of p' substituted for by p''.

The possibilities in an information state need not be compatible with each other; the model fragments within a possibility need not, either. That is, they need not be fragments of the same model, so to say. The most informative elements of a possibility can be complete models, namely, those that could be models of the real world given the information represented by the information state. This need not originate from the discourse itself; certain possible worlds can be excluded already at the outset, in the beginning of a conversation, so that the single possibility that we start the conversation with need not contain absolutely all possible models. For example, it may be the case that 'all birds fly' is true in all most informative model fragments in the initial possibility, although no bird whatsoever is present in the universes of the core elements. We will not dwell upon the question what type of information may be present in this form, nor how it gets there. We simply will allow any set of possibilities to act as an information state:

(10) **Definition: Information states**
The set Π of **information states** is $\mathcal{P}(\mathcal{P}(\mathcal{F}))$.

1.2. The Semantics of DUPLO

The language of DUPLO is a first-order language with equality, a modal operator '$\Diamond$', and a presupposition operator '$\triangle$'.[3] We will proceed in two steps for the sake of expository convenience. We will first define a version of DUPLO with total functions (without the $\triangle$ operator), then we will introduce the $\triangle$ operator and, at the same time, partial semantic-value functions.

1.2.1. The Total Version of DUPLO

We define two important concepts in the following. The first concept is the **truth** of a formula in an information state (under an assignment). The truth function is a three-valued function, because we want to make a distinction between 'known to be true' (truth value 1), 'known to be false' (truth value 0) and 'not known to be either true or false' (truth value $\frac{1}{2}$). The second central concept is the result of **updating** an information state with (the information content of) a formula (under an assignment). The truth function and the update function are defined using simultaneous recursion.

[3] For the sake of simplicity, we will not introduce functors into the language. In particular, we exclude individual constants. We decided to proceed in this way to avoid complications that have no relevance for the topic of this paper. We believe the system to be proposed could easily be enriched with functors.

(11) **Definition: DUPLO's language (total version)**
The total version of DUPLO's language is an ordered quadruple

$$\mathcal{L}' =_{\text{def}} \langle \text{LC}, \text{Var}, \text{Con}, \text{Form} \rangle.$$

The set LC of **logical constants**, the set Var of **individual variables** and the set Con of **non-logical constants** are pairwise disjunct.

$$\text{LC} =_{\text{def}} \{\exists, \neg, \Diamond, \rightarrow, `, (,)\};$$
$$\text{Var} =_{\text{def}} \{x_i\}_{i<\omega};$$
$$\text{Con} =_{\text{def}} \bigcup_{1 \leq n < \omega} \text{Con}^{(n)};$$
$$\text{Con}^{(n)} =_{\text{def}} \{P_i^{(n)}\}_{i<\omega}.$$

The set Form of **formulae** is defined as the smallest set satisfying the clauses of the definitions of atomic formulae and of the truth-value function below.

We will use the notation $|\cdot|_g$ for the first-order semantic-value function for the **atomic formulae** of DUPLO. This function yields a classical truth value (1 or 0) for every formula and model fragment.

(12) **Definition: Atomic formulae and their first-order truth values**
If g is an assignment function and f is a model fragment, then the **first-order semantic-value function** $|\cdot|_g^f$ is defined in a standard way, as follows:

(i) If $x_1, \ldots, x_n \in \text{Var}$ and $P \in \text{Con}^{(n)}$, then '$P(x_1, \ldots, x_n)$' $\in$ Form is an **atomic formula** of DUPLO.

$$|P(x_1, \ldots, x_n)|_g^f =_{\text{def}} \begin{cases} 1 & \text{if } \langle g(x_1), \ldots, g(x_n) \rangle \in \mathcal{I}_f(P); \\ 0 & \text{otherwise.} \end{cases}$$

(ii) If $x_1, x_2 \in \text{Var}$, then '$(x_1 = x_2)$' $\in$ Form is an **atomic formula** of DUPLO.

$$|(x_1 = x_2)|_g^f =_{\text{def}} \begin{cases} 1 & \text{if } g(x_1) = g(x_2); \\ 0 & \text{otherwise.} \end{cases}$$

The truth value of a formula φ in an information state π (under an assignment g) will be written as $[\varphi]'_g(\pi)$ (the partial version of this function will be $[\cdot]_g$). The result of updating an information state π with a formula φ (under an assignment g) is written as $[\![\varphi]\!]'_g(\pi)$ (the partial version being $[\![\varphi]\!]_g(\pi)$). We will now define both the total truth function and the total update function. The definition of the truth function also contains the definition of non-atomic formulae of DUPLO. We will rely on the concept of **modified assignments**, defined in the usual way:

(13) **Definition: Modified assignment**

If g is an assignment function in ${}^{\mathrm{Var}}\mathcal{U}$, $x \in \mathrm{Var}$ and $u \in \mathcal{U}$, then $g[x\colon u]$, the **assignment modified** in x for u, is defined as follows:

$$g[x\colon u](y) =_{\mathrm{def}} \begin{cases} u & \text{if } y = x; \\ g(y) & \text{otherwise} \end{cases}$$

for every $y \in \mathrm{Var}$.

(14) **Definition: Total truth function**

The **truth value** of a formula φ in an information state π under the assignment g in the total version of DUPLO, written '$[\varphi]'_g(\pi)$', is defined as follows:

i. If φ is an atomic formula, then

$$[\varphi]'_g(\pi) =_{\mathrm{def}} \begin{cases} 1 & \text{if } \bigwedge_{p\in\pi} \bigwedge_{f\in p} |\varphi|^f_g = 1; \\ 0 & \text{if } \bigwedge_{p\in\pi} \bigwedge_{f\in p} |\varphi|^f_g = 0; \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

That is, for an atomic formula to be true (false) in an information state, it has to be true (false) in every model fragment of every possibility. The truth and falsity of atomic formulae are stable because, as we will see later on, updating an information state can at most **eliminate** model fragments (see (20)).

ii. If $x \in \mathrm{Var}$ and $\varphi \in \mathrm{Form}$, then '$\exists x\varphi$' $\in$ Form.

$$[\exists x\varphi]'_g(\pi) =_{\mathrm{def}} \begin{cases} 1 & \text{if } \bigwedge_{p\in\pi} \bigwedge_{f\in\mathcal{C}(p)} \bigvee_{u\in\mathcal{U}_f} [\varphi]'_{g[x:u]}(\{\downarrow_p f\}) = 1; \\ 0 & \text{if } \bigwedge_{p\in\pi} \bigwedge_{f\in\mathcal{C}(p)} \bigwedge_{u\in\mathcal{U}_f} [\varphi]'_{g[x:u]}(\{\downarrow_p f\}) = 0; \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

For an existential formula of the form $\exists x\varphi$ to be true in an information state, each core element of each possibility must contain at least one individual for which the body (φ) is true in the cloak of the core element. On the other hand, we can only take it for granted that it is false if no core element of any possibility contains such individuals. There is uncertainty in the remaining cases. An existential formula is **T-unstable** (it may become false from true) just in case its body is; it is **F-unstable** (it may become true from false) under normal circumstances, because extending the discourse universe may introduce entities for which φ holds.

iii. If $\varphi \in \mathrm{Form}$, then '$\neg\varphi$' $\in$ Form.

$$[\neg\varphi]'_g(\pi) =_{\mathrm{def}} 1 - [\varphi]'_g(\pi).$$

This is trivial. Note that $\neg\varphi$ is T-unstable just in case φ is F-unstable, and it is F-unstable just in case φ is T-unstable.

iv. If $\varphi, \psi \in$ Form, then '$(\varphi \wedge \psi)$' $\in$ Form.

$$[(\varphi \wedge \psi)]'_g(\pi) =_{\text{def}} \begin{cases} 1 & \text{if } \bigvee_{\pi' \in \Pi}([\varphi]'_g(\pi') = 1 \;\&\; [\![\psi]\!]'_g(\pi') = \pi); \\ 0 & \text{if } [\varphi]'_g(\pi) = 0 \vee [\psi]'_g(\pi) = 0; \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

As can be seen, the truth and the falsity of conjunction are defined in an asymmetric way. The truth of a conjunction is **dynamic**: If the first conjunct introduces some entity, then the second conjunct may pick it up. On the other hand, if the first conjunct is T-unstable, then it need not be true in π provided the second conjunct falsifies it. Take, e.g., $\Diamond\varphi \wedge \neg\varphi$: its truth in π only requires for $\neg\varphi$ to hold in π. So the intuition behind this definition is that a formula must be true in an information state that could have been produced by updating another information state with it. On the other hand, our impression is that defining falsity in a dynamic way is both technically impossible and unnecessary.

v. If $\varphi, \psi \in$ Form, then '$(\varphi \rightarrow \psi)$' $\in$ Form.

$$[(\varphi \rightarrow \psi)]'_g(\pi) =_{\text{def}} [\psi]'_g([\![\varphi]\!]'_g(\pi)).$$

The question arises why we need a separate connective for material implication instead of using the standard definition '$\neg(\varphi \wedge \neg\psi)$'. As we have seen, '$[\neg(\varphi \wedge \neg\psi)]'_g(\pi) = 1$' expresses that either φ is false or ψ is true in π. '$[\varphi \rightarrow \psi]_g(\pi) = 1$', on the other hand, means that neither $\neg\varphi$ nor ψ need to be true in π, but updating π with φ yields an information state in which ψ is true. This cannot be expressed using the other connectives, but we may need it for translating natural-language sentences which express a conditional relation between two propositions the truth values of which need not be known, as we will see later on.

vi. If $\varphi \in$ Form, then '$\Diamond\varphi$' $\in$ Form.

$$[\Diamond\varphi]'_g(\pi) =_{\text{def}} \begin{cases} 1 & \text{if } \bigwedge_{p \in \pi}[\![\varphi]\!]'_g(\{p\}) \neq \emptyset; \\ 0 & \text{if } \bigwedge_{p \in \pi}[\![\varphi]\!]'_g(\{p\}) = \emptyset; \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

This is almost like the usual interpretation of $\Diamond\varphi$ in Update Semantics, except that we stipulate that it must be possible to update every possibility **separately** ('distributively') with φ in order for $\Diamond\varphi$ to be true.

The update function is defined in such a way that updating π with φ yields an information state that is minimally more informative than π and makes φ true.

(15) **Total update function**
If π is an information state, φ is a formula and g is an assignment function, then the total version of the result of **updating** π with φ under g, written '$[\![\varphi]\!]'_g(\pi)$', is defined as

$$[\![\varphi]\!]'_g(\pi) =_{\text{def}} \bigcup_{p\in\pi} \text{MAX}_{p'\subseteq p}([\varphi]'_g(\{p'\}) = 1).$$

It is easy to see that φ is indeed true in $[\![\varphi]\!]'_g(\pi)$. The reason is that, since every clause in the truth function is defined possibility by possibility,

$$[\varphi]'_g(\pi_1) = [\varphi]'_g(\pi_2) \Rightarrow [\varphi]'_g(\pi_1 \cup \pi_2) = [\varphi]'_g(\pi)$$

for any π_1, π_2 and φ. On the other hand, for the same reason, $[\![\varphi]\!]'_g(\pi)$ is indeed the least informative extension of π which has this property.

Let us now review each type of formulae and elaborate on their update effects.

(16) **Facts about the total update function**
i. If φ is an atomic formula, then

$$[\![\varphi]\!]'_g(\pi) = \bigcup_{p\in\pi} \{\{f \in p\colon |\varphi|^f_g = 1\}\}.$$

If φ is an **atomic formula**, then the least informative extension of π in which φ is true can be produced by leaving out every model fragment from every possibility in which φ is false.

ii.
$$[\![\exists x\varphi]\!]'_g(\pi) = \bigcup_{p\in\pi} \{\{f \in p\colon \bigvee_{f'\in p} (f \sqsubseteq f' \;\&\; \bigvee_{u\in\mathcal{U}_{f'}} [\varphi]'_{g[x:u]}(\{\downarrow_p f\}) = 1).$$

By the above argument, this is indeed the least informative extension of π which makes $\exists x\varphi$ true. When updating an information state with $\exists x\varphi$, we guarantee that every core element of every possibility will contain at least one individual that satisfies φ, but not more than absolutely necessary. If the input information state did not contain individuals satisfying φ, then every possible value of x will be present in some fragment of every possibility in the output state. As a result, no new possibilities arise, but many new core elements do which only differ in what individual plays the role of 'x', so to say.

iii.

$$[\![\neg\varphi]\!]'_g(\pi) = \bigcup_{p\in\pi} \mathrm{MAX}_{p'\subseteq p}([\varphi]'_g(\{p'\}) = 0).$$

This is just the instantiation of the definition in (15) for negative formulae.

iv.

$$[\![\varphi \wedge \psi]\!]'_g(\pi) = [\![\psi]\!]'_g([\![\varphi]\!]'_g(\pi)).$$

As usual in dynamic semantics, conjunction corresponds to function composition. It is easy to prove that this is indeed the least informative extension of π in which '$\varphi \wedge \psi$' is true.

v.

$$[\![\varphi \rightarrow \psi]\!]'_g(\pi) = \bigcup_{p\in\pi} \mathrm{MAX}_{p'\subseteq p}([\psi]'_g([\![\varphi]\!]'_g(\{p'\})) = 1).$$

Just like disjunction (the negation of a conjunction), conditionals are also able to multiply possibilities when we update an information state with them.

vi.

$$[\![\Diamond\varphi]\!]'_g(\pi) = \{p \in \pi \colon [\![\varphi]\!]'_g(\{p\}) \neq \emptyset\}.$$

Unlike in Update Semantics, modal formulae **do** have an updating effect. As we will see shortly, this is just a technical difference.

The concept of **entailment** in DUPLO could be defined in the usual way:

$$\varphi_1, \ldots, \varphi_n \models \psi \Leftrightarrow [\psi]'_g([\![\varphi_1 \wedge \ldots \wedge \varphi_n]\!]'_g(\pi)) = 1$$

for all π and g. This would probably do the job, but we can provide a stronger definition, and one that is closer to the classical concept of entailment:

(17) **Definition: Entailment in DUPLO**
A sequence of formulae $\varphi_1, \ldots, \varphi_n$ **entails** a formula ψ ($\varphi_1, \ldots, \varphi_n \models \psi$) iff, for all information states π and assignments g,

$$[\psi]'_g(\pi) \geq [\varphi_1 \wedge \ldots \wedge \varphi_n]'_g(\pi).$$

Note that, in DUPLO, entailment is trivially **reflexive**. In Update Semantics, a formula like $\Diamond\varphi \wedge \neg\varphi$ does not entail itself. In that theory, $\Diamond\varphi \wedge \neg\varphi$ would entail itself if and only if, for any π, the result updating π with $\Diamond\varphi \wedge \neg\varphi$ supported $\Diamond\varphi \wedge \neg\varphi$ (i.e., in our terminology, if $\Diamond\varphi \wedge \neg\varphi$ was true in it). But $\Diamond\varphi \wedge \neg\varphi$ has the same update effect as $\neg\varphi$, and updating π with $\neg\varphi$ yields an information state that does not support $\Diamond\varphi$. In DUPLO, however, we have the following fact:

(18) **Fact: Entailment is reflexive**
$\varphi \models \varphi$ for every φ.

Moreover, entailment is also trivially **transitive** in DUPLO. In first-order versions of Update Semantics, a formula like $\exists x(P(x))$ entails $\exists y(P(y))$, and the latter entails $P(y)$, but $\exists x(P(x))$ does not entail $P(y)$. DUPLO does not have this rather inconvenient feature.

(19) **Fact: Entailment is transitive**
$\varphi \models \psi \mathbin{\&} \psi \models \chi \Rightarrow \varphi \models \chi$.

1.2.2. Some Properties of DUPLO

In what follows, we examine certain general properties of DUPLO. In particular, we are interested in those features that distinguish it from other dynamic theories. However, we defer the discussion of differences related to anaphors to **Section 2**.

The first property that is usually relevant in the assessment of a dynamic semantic theory is **eliminativity**:

(20) **Fact: Eliminativity**
$\bigwedge_{p \in [\![\varphi]\!]'_g(\pi)} \bigvee_{p' \in \pi} p \subseteq p'$.

That is, the update function always yields an information state that is at least as informative as its input. Some version of eliminativity is usually satisfied by dynamic theories, unless they set out to account for cases of **belief revision**.

We have mentioned earlier that the truth and falsity of certain DUPLO formulae are not stable:

(21) **Fact: Instability**
$[\varphi]'_g(\pi) = 1 \mathbin{\&} [\![\psi]\!]'_g(\pi) = \pi' \not\Rightarrow [\varphi]'_g(\pi') = 1$;
$[\varphi]'_g(\pi) = 0 \mathbin{\&} [\![\psi]\!]'_g(\pi) = \pi' \not\Rightarrow [\varphi]'_g(\pi') = 0$.

That is, the truth and falsity of a formula φ are not preserved under arbitrary updates. For example, $\Diamond\forall x P(x)$ may be true in π (namely, if every possibility in π contains sub-possibilities in which $\forall x P(x)$ is true). Updating π with $\neg\forall x P(x)$ may yield a non-empty information state π'. Obviously, $\Diamond\forall x P(x)$ is not true in the resulting information state. On the other hand, because of the fact mentioned in (14iii), $\neg\Diamond\varphi$ is F-unstable.

Since $\exists x\varphi$ is F-unstable, **universal quantification** defined in the usual way will come out as T-unstable in DUPLO:

(22) **Fact: Truth of a universal formula**

1. $[\forall x\varphi]'_g(\pi) = 1$
2. $[\neg\exists x\neg\varphi]'_g(\pi) = 1$ [**1**, by def.]
3. $[\exists x\neg\varphi]'_g(\pi) = 0$ [**2**, (14iii)]
4. $\bigwedge_{p\in\pi}\bigwedge_{f\in\mathcal{C}(p)}\bigwedge_{u\in\mathcal{U}_f}[\neg\varphi]'_{g[x:u]}(\{\downarrow_p f\}) = 0$ [**3**, (14ii)]
5. $\bigwedge_{p\in\pi}\bigwedge_{f\in\mathcal{C}(p)}\bigwedge_{u\in\mathcal{U}_f}[\varphi]'_{g[x:u]}(\{\downarrow_p f\}) = 1$ [**4**, (14iii)]

As can be seen, the truth of $\forall x\varphi$ involves universal quantification over individuals already introduced. So introducing new referents may alter its truth value.

On the other hand, the F-instability of $\forall x\varphi$ depends on the properties of φ:

(23) **Fact: Falsity of a universal formula**

1. $[\forall x\varphi]'_g(\pi) = 0$
2. $[\neg\exists x\neg\varphi]'_g(\pi) = 0$ [**1**, by def.]
3. $[\exists x\neg\varphi]'_g(\pi) = 1$ [**2**, (14iii)]
4. $\bigwedge_{p\in\pi}\bigwedge_{f\in\mathcal{C}(p)}\bigvee_{u\in\mathcal{U}_f}[\neg\varphi]'_{g[x:u]}(\{\downarrow_p f\}) = 1$ [**3**, (14ii)]
5. $\bigwedge_{p\in\pi}\bigwedge_{f\in\mathcal{C}(p)}\bigvee_{u\in\mathcal{U}_f}[\varphi]'_{g[x:u]}(\{\downarrow_p f\}) = 0$ [**4**, (14iii)]

If $\forall xP(x)$ is false in an information state, then it will remain false forever, because introducing new individuals may not change the truth of P for the referents introduced earlier. However, if $\forall x\exists!yR(x,y)$ was false in an information state for $\exists x\neg\exists yR(x,y)$ is true in it, then introducing a new individual can make the universal formula true.

In contradistinction to DPL and similar theories, universal formulae can also have **updating effects** in DUPLO.

(24) **Fact: Update of universal formulae**

1. $[\![\forall x\varphi]\!]'_g(\pi)$
2. $[\![\neg\exists x\neg\varphi]\!]'_g(\pi)$ [**1**, by def.]
3. $\bigcup_{p\in\pi}\mathrm{MAX}_{p'\subseteq p}([\exists x\neg\varphi]'_g(\{p'\}) = 0)$ [**2**, (16iii)]
4. $\bigcup_{p\in\pi}\mathrm{MAX}_{p'\subseteq p}(\bigwedge_{f\in\mathcal{C}(p')}\bigwedge_{u\in\mathcal{U}_f}[\neg\varphi]'_{g[x:u]}(\{\downarrow_{p'} f\}) = 0)$ [**3**, (14ii)]
5. $\bigcup_{p\in\pi}\mathrm{MAX}_{p'\subseteq p}(\bigwedge_{f\in\mathcal{C}(p')}\bigwedge_{u\in\mathcal{U}_f}[\varphi]'_{g[x:u]}(\{\downarrow_{p'} f\}) = 1)$ [**4**, (14iii)]

This means that universal formulae may introduce new individuals into the discourse universe. In particular, the updating effect of a sentence like

(25) *Every farmer has a donkey*

consists in minimally trimming the input information state so that every farmer in the output state has a donkey.

(26) **Fact: Update of** $\forall x(F(x) \rightarrow \exists y D(x,y))$

1. $[\![\forall x(F(x) \rightarrow \exists y D(x,y))]\!]'_g(\pi)$

2. $\bigcup_{p\in\pi} \text{MAX}_{p'\subseteq p}(\bigwedge_{f\in\mathcal{C}(p')} \bigwedge_{u\in\mathcal{U}_f}$
$[F(x) \rightarrow \exists y D(x,y)]'_{g[x:u]}(\{\downarrow_{p'} f\}) = 1)$ [**1**, (24)]

3. $\bigcup_{p\in\pi} \text{MAX}_{p'\subseteq p}(\bigwedge_{f\in\mathcal{C}(p')} \bigwedge_{u\in\mathcal{U}_f}$
$[\exists y D(x,y)]'_{g[x:u]}([\![F(x)]\!]_{g[x:u]}(\{\downarrow_{p'} f\})) = 1)$ [**4**, (14v)]

That is, the resulting information state will contain as many donkeys as there are individuals in the information state that are known to be farmers. Furthermore, if an individual already present in the discourse universe turns out to be a farmer later on, his donkey will be introduced automatically. (The restriction to individuals already introduced stems from the definition of $\forall$.) On the other hand, if we defined material implication in terms of negation and conjunction, we would get a different result.

(27) **Fact: Update of** $\forall x \neg(F(x) \wedge \neg\exists y D(x,y))$

1. $[\![\forall x \neg(F(x) \wedge \neg\exists y D(x,y))]\!]'_g(\pi)$

2. $\bigcup_{p\in\pi} \text{MAX}_{p'\subseteq p}(\bigwedge_{f\in\mathcal{C}(p')} \bigwedge_{u\in\mathcal{U}_f}$
$[\neg(F(x) \wedge \neg\exists y D(x,y))]'_{g[x:u]}(\{\downarrow_{p'} f\}) = 1)$ [**1**, (24)]

3. $\bigcup_{p\in\pi} \text{MAX}_{p'\subseteq p}(\bigwedge_{f\in\mathcal{C}(p')} \bigwedge_{u\in\mathcal{U}_f}$
$[F(x)]'_{g[x:u]}(\{\downarrow_{p'} f\}) = 0 \vee$
$[\exists y D(x,y)]'_{g[x:u]}(\{\downarrow_{p'} f\}) = 1)$ [**2**, (14iii), (14iv)]

This interpretation has not much to do with a conditional. Rather, it says 'everyone (in the discourse universe) either is definitely not a farmer or positively has a donkey'. This is the argument that we mentioned when we defined a separate connective for material implication.

Turning back to the translation in (26), the minimal modification of the information state that we calculated there involves introducing a donkey for every farmer in the discourse universe if they all own donkeys. (Otherwise, the empty information state is returned.) Updating with (25) cannot introduce new farmers, however. If all farmers in the input information state have donkeys, then introducing new ones would be a **non-minimal** modification; if some do not, then the empty information would be returned, anyway.

The fact that most dynamic semantic theories treat sentences like (25) as 'externally static' (i.e., not giving rise to new discourse referents) stems from the observation that such a sentence cannot be continued with *it* (as referring to a donkey). We believe this is a *non sequitur*: a sentence like (25) is pragmatically inadequate when talking about just one farmer (with just one donkey), so no single donkey is available in its output information state. Several donkeys must be there, which makes *it* inappropriate, but *they* is fine (except that it is ambiguous between 'the farmers' and 'the donkeys').

Another important concept of dynamic theories of semantics is **distributivity**. A formula is said distributive if and only if updating an information state with it consists in updating each possibility separately. A language interpreted dynamically is said distributive if all of its formulae are distributive. As shown by Groenendijk and Stokhof (1990), a logic that is both distributive and eliminative is not really dynamic. But their result does not apply to our theory, because DUPLO updates do not 'distribute' down to the model fragments (except for atomic formulae).

(28) **Fact: Distributivity in DUPLO**
$[\![\varphi]\!]'_g(\pi) = \bigcup_{p\in\pi}[\![\varphi]\!]'_g(\{p\})$.

That is, the process of updating an information state is distributive in the sense that we get the same result if we update the singleton information states containing each possibility one by one, then take the union of the results.

As we mentioned earlier, the distributivity of the possibility operator is harmless in DUPLO. In particular, the argument in Groenendijk *et al.* (1994) why possibility should be non-distributive does not apply to our treatment:

(29) *If someone is hiding in the closet, then he might have done it.*

The interpretation that both Groenendijk *et al.* (1994) and DUPLO predict for this sentence and the DUPLO definitions yield is 'there must be at least one individual among those who might be hiding in the closet who might have done it'. As for the DUPLO treatment of (29), updating an information state with its *if*-clause yields one in which **every** possibility will contain everyone who may be hiding in the closet, so looking at possibilities one by one to assess the truth of the *then*-clause yields the correct result.

On the other hand, DUPLO's interpretation of the **necessity** operator is quite different from that of Update Semantics. In Update Semantics, a formula like $\Box\varphi$ (i.e., $\neg\Diamond\neg\varphi$) may have no updating effect (it yields either the input information state, if φ is true in π, or else the empty information state). In DUPLO, on the other hand, updating π with $\Box\varphi$ yields an extension of π in which φ is true in a **stable** way:

(30) **Fact: Update effect of $\Box\varphi$**

1.	$[\![\Box\varphi]\!]'_g(\pi)$	
2.	$[\![\neg\Diamond\neg\varphi]\!]'_g(\pi)$	[**1**, by def.]
3.	$\bigcup_{p\in\pi}\text{MAX}_{p'\subseteq p}([\Diamond\neg\varphi]'_g(\{p'\}) = 0)$	[**2**, (16iii)]
4.	$\bigcup_{p\in\pi}\text{MAX}_{p'\subseteq p}([\![\neg\varphi]\!]'_g(\{p'\}) = \emptyset)$	[**3**, (14vi)]
5.	$\bigcup_{p\in\pi}\text{MAX}_{p'\subseteq p}(\bigwedge_{p''\subseteq p'}[\varphi]'_g(\{p''\}) = 1)$	[**4**, (15)]

The asymmetry between possibility and necessity is not unrealistic. Consider the following pieces of discourse:

(31) *A man is walking in the park. Maybe he wears a blue T-shirt.*
(32) *A man is walking in the park. It is known that he wears a blue T-shirt.*

Assuming that all our model fragments are fragments of a model in which two men exist altogether, exactly one of which wears a blue T-shirt, the piece of discourse in (31) says nothing about who is walking in the park (if we start from a minimally informative information state).[4] It could be either man, so both possibilities will be present in the output information state. The piece of discourse in (32), on the other hand, seems to exclude the possibility that the man walking in the park is not wearing a blue T-shirt. Under Update Semantics' definition of negation, the piece of discourse in (32) should be a contradiction. Since we start from a minimally informative information state, nothing is known about the man after processing a first sentence, and the second sentence would claim that something **is** known about him. Under DUPLO's view, however, if communicating '$\Box\varphi$' is to communicate 'it is known that φ is true in the world', so it makes sense to consider it synonymous with φ (including its dynamic effects) except when φ is T-unstable.

The stabilizing effect of the necessity operator '$\Box$' can be put to use in the translation of natural-language sentences. For example, in most cases, overt negation is stable in natural languages, although the negation that we need in anaphoric quantification and universal quantification in general is not. Under the above treatment, we can assume that a natural language sentence of the form *It is not the case that* φ is always translated as '$\Box\neg\varphi$', which yields the desired effect. In this way, we do not have to introduce a separate, 'stabilizing' negation (together with a 'stabilizing' counterpart of universal quantification, and so on). Similarly, a natural-language sentence of the form *There is exactly one x such that* φ can be translated as '$\Box\exists!x\varphi$', because it is normally understood in a stable way, unlike anaphoric quantification. By the same token, if *Every x is such that* φ is not understood as quantifying over the discourse universe, but is intended as an 'eternal truth', its translation must be prefixed with '$\Box$'.

Note that **double negation** can be eliminated in DUPLO. As a consequence, double negation does not make a formula **externally static**. For example, we predict the following piece of discourse to be acceptable under normal circumstances:

(33) *It is not the case that Joe does not have a car. I saw* it *parked next to the entrance.*

---

[4] Here we assume that our information about men in the model and who wears a blue T-shirt does not originate from previous discourse information.

It is not entirely clear whether this is a desirable prediction. People's judgments diverge on this piece of discourse, so it probably cannot serve as a key example in the assessment of theories of anaphors.

1.2.3. Partial DUPLO

For the sake of simplicity, we will not be very precise in the definition of the partial version of DUPLO. The language $\mathcal{L}$ of partial DUPLO is the same as $\mathcal{L}'$ except that the set of logical constants contains an additional symbol, the **presupposition operator** '$\triangle$', and if $\varphi \in$ Form, then '$\triangle\varphi$' $\in$ Form. These formulae are the ultimate sources of presupposition, i.e., undefinedness of semantic values. We use the symbol '$*$' for an undefined value rather than a special value. The partial truth function will be written as $[\cdot]_g$, and the partial update function is $[\![\cdot]\!]_g$. The truth values of $\triangle\varphi$ are as follows:

(14) vii.

$$[\triangle\varphi]_g(\pi) =_{\text{def}} \begin{cases} * & \text{if } [\varphi]_g(\pi) \neq 1; \\ 1 & \text{otherwise.} \end{cases}$$

As for the other clauses of the truth function, we will just specify when its value is undefined. When it does assign a truth value, the value is always calculated in the same way as that of the total version, except that we have to use the partial versions of the truth and the update functions in the calculations. The definition below attempts to express standard assumptions about presupposition and presupposition projection, which we will not dwell upon here, as they are tangential to our central concern.

(34) **Definition: Partial truth function**

i. The truth value of an atomic formula is always defined. Atomic formulae have no presuppositions.

ii.

$$[\exists x\varphi]_g(\pi) = * \Leftrightarrow_{\text{def}} \bigvee_{p\in\pi} \bigwedge_{f\in\mathcal{C}(p)} \bigwedge_{u\in\mathcal{U}_f} [\varphi]_{g[x:u]}(\{p\}) = *.$$

iii.

$$[\neg\varphi]_g(\pi) = * \Leftrightarrow_{\text{def}} [\varphi]_g(\pi) = *.$$

iv.

$$[\varphi \wedge \psi]_g(\pi) = * \Leftrightarrow_{\text{def}} \bigwedge_{\pi'\in\Pi} ([\![\psi]\!]_g(\pi') = \pi \Rightarrow [\varphi]_g(\pi') = *).$$

v.

$$[\varphi \rightarrow \psi]_g(\pi) = * \Leftrightarrow_{\text{def}} [\![\varphi]\!]_g(\pi) = * \vee ([\![\varphi]\!]_g(\pi) \neq * \,\&\, [\psi]_g([\![\varphi]\!]_g(\pi)) = *).$$

vi.

$$[\Diamond\varphi]_g(\pi) = * \Leftrightarrow_{\text{def}} [\varphi]_g(\pi) = *.$$

The partial update function is undefined for a formula φ in an information state π if the least informative extension of π in which φ is true cannot be calculated because of presupposition failure. In most cases, this means that the truth value of φ must be defined in π in order for $[\![\varphi]\!]_g(\pi)$ to be defined. There are only two exceptions from this rule, namely, conjunction and material implication, because their truth in the extensions of π depends on π indirectly: an intermediate information state is involved. But, in the case of material implication, the undefinedness of the update function coincides with that of the truth function as in the case of most other types of formula, because determining its truth involves updating the information state with the antecedent. Therefore, we have the following definition:

(35) **Definition: Partial update function**
If π is an information state, φ is a formula and g is an assignment function, then the result of **updating** π with φ under g, written '$[\![\varphi]\!]_g(\pi)$', is defined as follows.

(i) If φ is of the form $\psi \wedge \xi$, then

$$[\![\varphi]\!]_g(\pi) = * \Leftrightarrow_{\text{def}} [\![\psi]\!]_g(\pi) = * \vee ([\![\psi]\!]_g(\pi) \neq * \,\&\, [\![\xi]\!]_g([\![\psi]\!]_g(\pi)) = *);$$

(ii) Otherwise,

$$[\![\varphi]\!]_g(\pi) = * \Leftrightarrow_{\text{def}} [\varphi]_g(\pi) = *;$$

(iii) If $[\![\varphi]\!]_g(\pi) \neq *$, then

$$[\![\varphi]\!]_g(\pi) =_{\text{def}} \bigcup_{p \in \pi} \text{MAX}_{p' \subseteq p}([\varphi]_g(\{p'\}) = 1).$$

All the facts about the total version of DUPLO are valid in the partial version, except that the proviso 'when defined' is necessary in many cases (such as the definition of entailment). This does not affect the important logical properties of DUPLO, though.

2. Anaphors in DUPLO

Obviously, we have introduced both unstable quantification and presuppositions in order to deal with anaphors (although they may be put to use for different purposes as well). In terms of what we have anticipated in the earlier sections, the translation of a sentence like (1) will be as follows:

(1′) *Joe has a cat. It is black.*
'$\exists x(\text{cat}(x) \wedge \text{owns}(j, x)) \wedge$
$\triangle(\exists! y(\text{nonhuman}(y))) \wedge \forall z(\text{nonhuman}(z) \rightarrow \text{black}(z))$'

It is easy to see that the translation of the second sentence can be produced in a compositional manner. In general, if the descriptive content of an anaphor is $\mathcal{A}$, then its translation is

(36) *Translation of anaphors*
$\lambda P(\triangle(\exists! x(\mathcal{A}(x))) \wedge \forall y(\mathcal{A}(y) \rightarrow P(y)))$.[5]

Accordingly, we will use 'he′', 'it′' etc. as the translations of anaphors with the descriptive contents 'male', 'non-human' etc. below.

This treatment accounts for one of the most obvious characteristics of anaphors, namely that, in most constructs, the antecedent can be present in the **initial** information state even if the anaphor is deeply embedded:

(37) a. *Mary does not know* him.
b. *Mary would know* it.
c. *If she was there, she would know it.*

In (37a), the anaphor is within a negative predicate, in (37b), within a modal predicate, and in (37c), in the consequent of a (counterfactual) conditional. Yet in all three cases the antecedent is required to be present in the initial information state. The presuppositional treatment of anaphors ensures that the anaphoric quantifier will have 'wide scope' in the translations of the sentences in (37) without relying on wildly non-compositional devices.

---

[5] As a matter of course, this translation does not apply to so-called 'lazy pronouns', which do not stand for their antecedents but an analogous entity, as in

The man who gave his paycheck to his wife was wiser than the one who gave it *to his mistress.*

On the other hand, the antecedent does not have to be present in the initial information state, so we cannot say that the anaphoric quantifier always has a 'wide scope':

(38) *If she had a car, she would lend* it *to me.*

In this sentence, the antecedent of the anaphoric pronoun *it* is within the conditional; it is difficult to imagine a compositional mechanism that would assign 'wide scope' to the anaphoric quantifier in (37c), but 'narrow scope' to that in (38). The presuppositional treatment of anaphoric quantification, on the other hand, deals with both 'wide-scope' and 'narrow-scope' cases. In terms of the definedness conditions of the update function, the presuppositions of formulae must be satisfied in the initial information state as a rule. Formulae of the forms '$\varphi \wedge \psi$' and '$\varphi \rightarrow \psi$' are exceptional: the presuppositions of ψ must be satisfied by $[\![\varphi]\!]_g(\pi)$. That is, the presuppositions of the second member of a conjunction or a conditional are to be satisfied in an information state that their first member yields. This is essentially the same behaviour of 'presupposition projection' that has been proposed by Karttunen (1973, 1974) and Karttunen and Peters (1979), except for some marked cases, which are to be dealt with in special ways (cf. Kálmán (1994)).

Just the same as the choice of the variables corresponding to anaphors is done by magic in DPL, we also need magic for those conditions in the translations of anaphors that result from factors external to semantics. For example, consider the following translation:

(2) *Joe likes* him.
'$\triangle(\exists! x(\text{male}(x) \wedge x \neq j)) \wedge \forall y((\text{male}(x) \wedge x \neq j) \rightarrow \text{likes}(j, x))$'

Here it is not clear how the condition '$x \neq j$' gets into the translation of *him*. It is allegedly a condition that can be produced by syntactic information (since *Joe* c-commands *him*, they cannot be co-referential). If we did not include this condition on the antecedent, the sentence could only mean 'Joe likes himself'.

Obviously, the properties of dynamic logics such as DPL which are related to the bound-variable view of anaphors do not hold for DUPLO. In particular, the so-called **donkey equivalences** of DPL,

$$\exists x(\varphi) \wedge \psi \equiv \exists x(\varphi \wedge \psi);$$
$$(\exists x(\varphi)) \rightarrow \psi \equiv \forall x(\varphi \rightarrow \psi)$$

do not hold in DUPLO. ('$\varphi \equiv \psi$' means that both $[\cdot]_g$ and $[\![\cdot]\!]_g$ yield identical values for φ and ψ.) We have argued against the first of these in the previous sections. We will show now that the second equivalence is also not necessary for dealing with the relevant facts.

The use of anaphors in conditional sentences is an age-long problem. Consider:

(39) *If a farmer owns a donkey,* he *beats* it.

As can be seen, anaphors in the consequent of a conditional may refer to entities introduced in the antecedent. Moreover, these anaphors are in the singular (at least in English), which suggests that the conditional is a case-by-case statement about the entities that the antecedent introduces. According to a proposal in Heim (1990), this phenomenon may be due to the fact that conditional sentences involve a (usually implicit) quantification over 'cases' or 'situations', which are selected by the antecedent. For example, (39) should be interpreted as 'in each situation in which a farmer owns a donkey, that farmer beats that donkey'.

The machinery of DUPLO as introduced above offers a natural way of capturing the concept of 'cases' or 'situations'. Each 'case' or 'situation' can be seen as a possibility in the information state. Updating an information state with the antecedent of a conditional sentence yields just those 'minimal situations' that Heim (1990) mentions.[6]

As for the anaphors in the consequent of (39), the possibilities that updating with the antecedent yields are the 'biggest' (i.e., least informative) among those in which 'a farmer owns a donkey'. Therefore, their discourse universes (i.e., the universes of their core elements) will be minimally larger than those in the input information state, i.e., they will contain just one farmer and one donkey.[7] That is why the translations that we proposed for *he* and *it* are appropriate in this case:

(39′) *If a farmer owns a donkey, he beats it.*
$\exists x(F(x) \wedge \exists y D(x,y)) \rightarrow \text{beats}(\text{he}',\text{it}')$

(40) **Fact: Truth of (39′)**

1. $[\exists x(F(x) \wedge \exists y D(x,y)) \rightarrow \text{beats}(\text{he}',\text{it}')]_g(\pi) = 1$

2. $[\text{beats}(\text{he}',\text{it}')]_g([\![\exists x(F(x) \wedge \exists y D(x,y))]\!]_g(\pi)) = 1$ [**1**, (14v)]

3. $[\text{beats}(\text{he}',\text{it}')]_g(\bigcup_{p\in\pi} \text{MAX}_{p'\subseteq p}(\bigwedge_{f\in\mathcal{C}(p')} \bigvee_{u,u'\in\mathcal{U}_f}$
$[F(x) \wedge D(x,y)]_{g[x:u][y:u']}(\{\downarrow_{p'} f\}) = 1)) = 1$ [**2**, (15)]

---

[6] As a matter of fact, it yields the minimal extensions of the input state satisfying the antecedent. We are aware of the fact that not every conditional sentence is to be interpreted in this way. For example, the antecedent of a counterfactual selects minimal situations that are not even compatible with the input information state. We believe that even normal conditional sentences could be treated in this way. That is, in that case, the antecedent could yield 'sub-situations' of the input situations, i.e., minimal modifications possibly different from its extensions.

[7] Unless the input information state already contains farmers and donkeys; see below.

Since the possibilities selected by the antecedent will contain all possible farmer/donkey pairs, we get the interpretation that DPL yields if we translate sentences like (39) as above. The translation of (39) is true if and only if, in every possibility that arises from the input state by introducing a farmer and a donkey he owns, the farmer will beat the donkey. Whether all conditional sentences in natural language are to be translated in this way is arguable. In fact, the uniform 'double-universal' readings that DPL's 'donkey equivalences' predict have been largely questioned in the literature. Maybe the 'double-universal' reading that the above translation yields is peculiar for such atemporal sentences, and other universal sentences have different translations.

There is an interesting problem that this analysis raises. If the initial information state in which (39) is uttered already contains more than one farmer and/or donkey, then the biggest sub-possibilites that satisfy 'a farmer owns a donkey' also contain several of them. In that case, we predict *he* and/or *it* in the antecedent to be infelicitous. This prediction may sound odd, but — at least in principle — the anaphoric pronouns *he* and/or *it* cannot in fact be used felicitously if there are several possible antecedents in the context. The peculiarity of this prediction in the case of (39) may be due to rhetorical reasons. But this problem would not even arise if we adopted the strategy described in footnote 6. This possibility opens directions for further research.

References

Beaver, D. 1992. *The Kinematics of Presupposition.* ITLI Prepublication Series LP-92-05, University of Amsterdam.

Cooper, R. 1979. 'The interpretation of pronouns'. In F. Heny and H. Schnelle, eds., *Syntax and Semantics 10.* Academic Press, New York.

Dekker, P. 1992. *An Update Semantics for Dynamic Predicate Logic.* ITLI Prepublication Series LP-92-04, University of Amsterdam.

Evans, G. 1980. 'Pronouns'. *Linguistic Inquiry* **11**.

Groenendijk, J. and M. Stokhof, 1990. 'Two theories of dynamic semantics'. In J. van Eijck, ed., *Logics in AI, European Workshop JELIA '90.* Springer, Berlin, pp. 55–64.

Groenendijk, J. and M. Stokhof. 1991. 'Dynamic Predicate Logic'. *Linguistics and Philosophy* **14**, 39–100.

Groenendijk, J., M. Stokhof and F. Veltman. 1994. *This Might Be It.* ITLI Prepublication Series LP-94-13, University of Amsterdam.

Groenendijk, J., M. Stokhof and F. Veltman. 1995. 'Coreference and contextually restricted quantification'. Ms., September 11, University of Amsterdam.

Heim, I. 1990. 'E-type pronouns and donkey anaphora'. *Linguistics and Philosophy* **13**, 137–177.

Kálmán, L. 1994. *Conditionals, Quantification and Bipartite Meanings.* Working

Papers in the Theory of Grammar, Vol. 1, No. 3, Theoretical Linguistics Programme, Budapest University (ELTE) and Research Institute for Linguistics, Hungarian Academy of Sciences.

Kálmán, L. 1995. 'Donkeys without donkey equivalences'. Paper given at the Colloquium on Designing Logics, CCSOM, University of Amsterdam, September 8.

Karttunen, L. 1973. 'Presuppositions of compound sentences'. *Linguistic Inquiry* **4**, 169–193.

Karttunen, L. 1974. 'Presuppositions and linguistic context'. *Theoretical Linguistics* **1**, 181–194.

Karttunen, L. and S. Peters. 1979. 'Conventional implicature'. In C. Oh and D. Dinneen, eds., *Syntax and Semantics. Vol. 11: Presupposition.* Academic Press, New York, pp. 1–56.

Polanyi, L. 1988. 'A formal model of the structure of discourse'. *Journal of Pragmatics* **12**, 601–638.

Prüst, H. 1992. *On Discourse Structuring, VP Anaphora and Gapping.* Ph.D. diss., University of Amsterdam.

Veltman, F. 1981. 'Data semantics'. In J. Groenendijk, T. Janssen and M. Stokhof, eds., *Formal Methods in the Study of Language.* Vol. II. Mathematical Centre, Amsterdam, pp. 541–567.

Veltman, F. 1990. 'Defaults in Update Semantics'. In H. Kamp, ed., *Conditionals, Defaults and Belief Revision.* DYANA report No. R2.5A, Centre for Cognitive Science, University of Edinburgh, pp. 28–63.

PREVIOUS TITLES IN THIS SERIES:

1/1: M. BRODY, *Phrase Structure and Dependence.*

1/2: K. É. KISS, *Generic and Existential Bare Plurals and the Classification of Predicates.*

1/3: L. KÁLMÁN, *Quantification, Conditionals and Bipartite Meanings.*

1/4: Z. BÁNRÉTI, *Modularity in Sentence Parsing: Grammaticality Judgments in Broca's Aphasics.*

2/1: A. SZABOLCSI, *Strategies for Scope Taking.*

2/2: G. RÁDAI and L. KÁLMÁN, *Compositional Interpretation of Computer Command Languages.*

2/3: L. KÁLMÁN, *Strong Compositionality.*

2/4: M. BRODY, *Towards Perfect Syntax.*

2/5: A. ZSÁMBOKI, *Contrastive Co-Ordinations with Focussed Clauses.*