AN UPDATE SEMANTICS FOR
DYNAMIC PREDICATE LOGIC

Paul Dekker

ITLI Prepublication Series
for Logic, Semantics and Philosophy of Language  LP-92-04

University of Amsterdam
AN UPDATE SEMANTICS FOR DYNAMIC PREDICATE LOGIC

Paul Dekker
Department of Philosophy
University of Amsterdam
An update semantics for dynamic predicate logic *

Paul Dekker
ILLC/Department of Philosphy
University of Amsterdam
dekker@illc.uva.nl

Abstract
In this paper I propose an update semantics for Groenendijk and Stokhof's dynamic predicate logic. In the proposed system, sentences are interpreted as (partial) update functions on a domain of information states which are sets of partial variable assignments. I show how adverbs of quantification (symmetric and asymmetric) and generalized quantifiers can be introduced in a perspicuous and uniform way.

1 Introduction
Groenendijk and Stokhof, in [1990], characterize the meaning of a sentence in a static semantics as the set of indices at which the sentence is true. Such a set of indices, also called an information state, defines the information content of the sentence. In a dynamic semantics, it is not the information content, but the information change potential of a sentence that is regarded as constituting its meaning. In a dynamic semantics the meaning of a sentence is a function on the domain of information states.

Groenendijk and Stokhof compare two examples of such a dynamic semantics, namely dynamic predicate logic (DPL, Groenendijk and Stokhof [1991]) and update semantics (US, Veltman [1990]). The two systems formalize different aspects of the dynamics of discourse and in their present formulations they have conflicting logical properties. This paper sets out to reconcile the two systems by formulating an update semantics for DPL.

I will proceed as follows. In the next section I sketch DPL and US and argue for a genuine update formulation of DPL. In section 3 such an update style DPL, EDPL, is presented and discussed, and in section 4 it is extended with quantifiers.

2 Two theories of dynamic semantics
2.1 Dynamic predicate logic
DPL gives a dynamic interpretation of the language of first order predicate logic that accounts, among other things, for intersentential anaphoric relationships like we find

* I would like to thank David Beaver, Jeroen Groenendijk, Herman Hendriks, Martin Stokhof and Frank Veltman for pleasant and instructive discussion. I have had great benefit from their comments. Of course, remaining errors are mine.
them in *A cowgirl meets a boy. She slaps him*. Like in discourse representation theory and file change semantics, natural language noun phrases in DPL are associated with variables, or discourse markers, and information states determine what values they can have given the conditions imposed on them in the course of a discourse.

Information states in DPL, then, contain information about the values of variables. They are modeled as sets of variable assignments that determine what are the possible values of the variables at a certain stage in the processing of a discourse. So, if \( D \) is a domain of individuals, and \( V \) the set of variables, then \( D^V \) is the set of variable assignments and \( S = \mathcal{P}(D^V) \), the set of subsets of all variable assignments, is the set of information states.

The language of DPL is that of predicate logic, but for ease of exposition I disregard individual constants and identity. The semantics is defined with respect to a model \( M = (D, F) \) consisting of a non-empty set of individuals \( D \) and an interpretation function \( F \) that assigns sets of \( n \)-tuples of objects to \( n \)-ary relation expressions. (I omit reference to \( M \) whenever this does not lead to confusion.) The interpretation of formulas is a function on the domain of information states:

**Definition 2.1 (Semantics of DPL)**

- \( s[R x_1 \ldots x_n] = \{i \in s \mid (i(x_1), \ldots, i(x_n)) \in F(R)\} \)
- \( s[\neg \phi] = s - \{i(\phi)\} \)
- \( s[\exists x \phi] = s[x][\phi] \)
- \( s[\phi \land \psi] = s[\phi][\psi] \)

where

- \( \{i(\phi)\} = \{i \mid \{i\}(\phi) \neq \emptyset\} \)
- \( s[x] = \{j \mid \exists i \in s \exists d \in D: i[x/d] = j\} \)

The interpretation of an atomic formula in a state \( s \) retains all those variable assignments in \( s \) with respect to which the atomic formula is true in a classical sense. The negation of \( \phi \) subtracts those \( i \) in \( s \) which constitute a context \( \{i\} \) with respect to which the interpretation of \( \phi \) does not produce the absurd state \( \emptyset \). Conjunction is just function composition. The characteristic clause concerns the interpretation of the existential quantifier. If a formula \( \exists x \phi \) is interpreted in a state \( s \), the variable \( x \) is reinstated, and next \( \phi \) is interpreted.

**DPL licenses the following equivalences:**

**Fact 2.1 (Weak donkey equivalences)**

- \( (\exists x (\phi \land \psi)) \Leftrightarrow \exists x (\phi \land \psi) \)
- \( ((\phi \land \psi) \land \chi) \Leftrightarrow (\phi \land (\psi \land \chi)) \)

It is typical of DPL that the first equivalence also holds if the variable \( x \) is free in \( \psi \). These equivalences therefore allow DPL to deal with the following textbook example (which explains one half of the equivalences' label):

1. A farmer owns a donkey. He beats it.
   
   \( (\exists x (Fx \land \exists y(Dy \land Oxy)) \land Bxy) \Leftrightarrow \exists x (Fx \land \exists y(Dy \land (Oxy \land Bxy))) \)

This sequence of sentences turns out to be equivalent with the sentence *A farmer owns a donkey that he beats.*
The next fact follows from the weak donkey equivalence, on the classical definition of \( \rightarrow \) and \( \forall x \):

**Fact 2.2 (Strong donkey equivalences)**
- \( (\exists x \phi \rightarrow \psi) = \neg(\exists x(\phi \land \neg \psi)) \leftrightarrow \forall x(\phi \rightarrow \psi) \)
- \( ((\phi \land \psi) \rightarrow \chi) = \neg((\phi \land \psi) \land \neg \chi) \leftrightarrow \neg(\phi \land (\psi \land \neg \chi)) \equiv (\phi \rightarrow (\psi \rightarrow \chi)) \)

This enables DPL to deal with the museum piece donkey sentences:

(2) If a farmer owns a donkey he beats it.
\[ (\exists x(Fx \land \exists y(Dy \land Ox y)) \rightarrow Bxy) \leftrightarrow \forall x(Fx \rightarrow \forall y((Dy \land Ox y) \rightarrow Bxy)) \]

(3) Every farmer who owns a donkey beats it.
\[ \forall x((Fx \land \exists y(Dy \land Ox y)) \rightarrow Bxy) \leftrightarrow \forall x(Fx \rightarrow \forall y((Dy \land Ox y) \rightarrow Bxy)) \]

These sentences are assigned their so called ‘strong’ readings (hence the label of the equivalences). Both sentences state that every farmer beats every donkey he owns. It is the merit of DPL that it gives a compositional treatment of these examples.

A property of DPL that distinguishes it from classical, static, theories of interpretation, is that it has a non-eliminative semantics:

**Fact 2.3 (Non-eliminativity)**
- \( s[\phi] \not\subseteq s \)

Interpretation in DPL does not merely involve eliminating possibilities, but it may also involve introducing new possibilities, viz., by existential quantification. A characteristic property of DPL updates is furthermore that they are all distributive:

**Fact 2.4 (Distributivity)**
- \( s[\phi] = \bigcup_{i \in s} \{i\}[\phi] \)

Distributivity means that in the update of a state \( s \) with a formula \( \phi \) only properties of the individual elements of \( s \) count.

Truth and entailment are defined as follows:

**Definition 2.2 (Truth and entailment in DPL)**
- \( \phi \) is true in \( s \), \( s \models \phi \), iff \( s \subseteq [\phi] \)
- \( \phi_1, \ldots, \phi_n \models \psi \) iff \( \forall M, s : s[\phi_1]_M \ldots [\phi_n]_M \models _M \psi \)

A formula \( \phi \) is true in a state \( s \) iff all possible variable assignments \( i \in s \) constitute a state \( \{i\} \) with respect to which \( \phi \) can be succesfully computed (i.e., with respect to which the interpretation of \( \phi \) does not produce the absurd state \( \emptyset \)). A conclusion \( \psi \) follows from a sequence of premisses \( \phi_1, \ldots, \phi_n \), if the update of any information state \( s \) with \( \phi_1, \ldots, \psi_n \), respectively, produces a state in which \( \psi \) is true. The DPL notion of entailment is a dynamic one. For instance, \( \exists xFx \) entails \( Fx \). This fact corresponds to the following line of elementary reasoning:

(4) If a man comes from Rhodes, he likes pineapple-juice. A man I met yesterday comes from Rhodes. So, he likes pineapple-juice.
\[ (\exists x(Mx \land Rx) \rightarrow Lx), \exists x(Mx \land Rx) \models Lx \]
2.2 Update semantics

I now turn to Veltman’s first example of an update semantics in [1990]. It is sketched in its most rudimentary form as a (dynamic) propositional logic with an additional sentential operator $\Box$.

The kind of information US deals with is information about the world. Information states are modeled as subsets of the set of possible worlds $W$. For someone in information state $s$, each world in $s$ might correspond to the real world.

Interpretation in US involves update of information about the world. An atomic sentence $p$ in US updates an information state $s$ by eliminating the worlds in $s$ which are inconsistent with $p$, and negation and conjunction are interpreted as set subtraction and composition, respectively. The interesting bit comes in with the operator $\Box$. In an information state $s$, $\Box \phi$ tests whether $\phi$ is still consistent with $s$. Like its natural language counterpart might, it reflects upon the present information state and expresses that that state can be consistently updated with $\phi$.

The semantics of US is defined with respect to a model $M = (W, V)$ consisting of a set of worlds $W$ and an interpretation function $V$ that assigns sets of worlds to proposition letters. (Again, reference to $M$ is omitted when that does not lead to confusion.) The interpretation of a formula is defined as follows:

**Definition 2.3 (Update semantics)**

- $s[p] = \{i \in s \mid i \in V(p)\}$
- $s[\neg \phi] = s - s[\phi]$
- $s[\Box \phi] = \{i \in s \mid s[\phi] \neq \emptyset\}$
- $s[\phi \land \psi] = s[\phi][\psi]$

As is evident from the semantics of US, the result of interpreting a formula in a state $s$ is always a subset of $s$. Interpretation can only eliminate possibilities:

**Fact 2.5 (Eliminativity)**

- $s[\phi] \subseteq s$

This fact implies that interpretation guarantees update of information.

The $\Box$-operator reflects upon the specific stages in the process of information growth. The interpretation of $\Box \phi$ in a state $s$ returns $s$ if $\phi$ is acceptable in $s$, and the absurd state $\emptyset$ if $\phi$ is not acceptable in $s$. As a consequence, conjunction in US is non-commutative:

**Fact 2.6 (Non-commutativity)**

- $\phi \land \psi \neq \psi \land \phi$

An example of a non-commutative conjunction is $\Box \phi \land \neg \phi$. This conjunction, with this order of conjuncts, is consistent. On the other hand, the commutation $\neg \phi \land \Box \phi$ of this conjunction is inconsistent. The following pair of examples exemplifies this pattern (granted that we know that John is not Mary):

5) Somebody is knocking at the door. .... It might be John. .... It's Mary.
6) *Somebody is knocking at the door. .... It's Mary. .... It might be John.
If somebody hears someone knocking at the door, he may of course be curious who it is and not exclude the possibility that it is, say, John. Still, in that situation it is perfectly possible for him to find out that it is Mary who is knocking, not John. On the other hand, once he has found out that Mary is knocking on the door, it is excluded that it is John, and then it is quite absurd to say that, as far as he knows, it might be John who is knocking at that door.

Truth, or acceptance, and entailment are defined as follows:

**Definition 2.4 (Truth and entailment in update semantics)**

- $\phi$ is true in $s$, $s \models \phi$, iff $s \subseteq s[\phi]$
- $\phi_1, \ldots, \phi_n \models_3 \psi \text{ iff } \forall M, s : s[\phi_1]_M \cdots [\phi_n]_M \models_3 M \psi$

A formula $\phi$ is true in $s$ if after updating $s$ with $\phi$ we still envisage the possibilities we envisaged in state $s$, i.e., if $s[\phi]$ doesn’t contain more information than $s$; $\phi_1, \ldots, \phi_n$ entail $\psi$ if always, if you update your information with $\phi_1 \ldots \phi_n$, in that order, you arrive at a state of information to which update with $\psi$ doesn’t add anything more.

### 2.3 Update semantics and DPL

DPL and US are genuinely dynamic systems as can be seen from the fact in none of the two conjunction is commutative. The example $\Diamond \phi \land \neg \phi$ is a counterexample to commutativity in US, and in DPL a counterexample is $\exists x Fx \land Gx$. (Think of a little sequence like A man walks in the park. He whistles. This means something different than the example He whistles. A man walks in the park.)

As Groenendijk and Stokhof observe, the properties of (non-)eliminativity and (non-)distributivity serve to distinguish the two systems from classical theories of interpretation. It is fairly easily shown that a dynamic semantics in which all sentences are interpreted as eliminative and distributive updates is not really dynamic after all (Cf., Groenendijk and Stokhof [1990, p. 57] and van Bentham [1991, p. 137]).

However, as we have seen above, DPL is non-eliminative, since after interpreting an existentially quantified formula $\exists x \phi$ in a state $s$, the information $s$ has about the value of $x$ is lost. Furthermore, US is non-distributive, since the might-operator expresses global properties of an information state which do not need to hold of all singleton subsets of that state.

It is important to notice that it is distinct properties that distinguish DPL and US from static theories, non-eliminativity and non-distributivity respectively. These two different ways in which DPL and US depart from the static paradigm are reflected by a difference in the respective notions of truth. For $\psi$ to be true in $s$ in DPL, each singleton subset $s$ must allow update with $\psi$. This notion of truth is that of a distributive system. On the other hand, the US notion of truth is that of an eliminative system. For $\psi$ to be true in $s$, US requires that the update of $s$ with $\phi$ does not eliminate possibilities in $s$.

From these remarks it may already appear that the two different notions of truth and entailment should not be substituted for one another. For instance, if we adopt the US notion of truth in DPL, then the DPL-valid entailment $\exists x Px \models \exists y Py$ would no longer be valid. The reason is that, on the US notion of truth, $\exists y Py$ is
true in s iff $P_y$ is true in s, and, clearly, $\exists x P x \not\models P y \Rightarrow$ any of the two notions of entailment. On the other hand, if we adopt the (distributive) DPL notion of truth in US, then the US-valid entailment $\Diamond \phi \models \Diamond \phi$ would no longer go through. The reason is that, on the DPL notion of truth, $\Diamond \phi$ is true in s iff $\phi$ is true in s, and $\Diamond \phi \not\models \phi$ on any of the two notions of entailment.

We see that DPL and US are two really different systems of dynamic interpretation with conflicting characteristic properties. This is not to say that the two are incompatible though. As Groenendijk and Stokhof suggest, the two systems can be combined within a system that preserves the characteristic features of both and that, to some extent, gives a separate treatment of the two different kinds of (update of) information that the two systems deal with.

However, in this paper I want to show that it is worthwhile to remove one of the differences between the two systems, by adapting the logic of one of the two (DPL) to the format of that of the other (US). Doing thus, a combination of the two theories may have more the character of an integration, since it allows us to employ a singular notion of truth instead of the product of two, so to speak.

Two aspects of information growth

Characteristic feature of DPL is that it has a non-eliminative semantics. DPL’s non-eliminativity originates from existential quantifiers which effectuate a reinstatiation of variables, and this formalizes the phenomenon that indefinite noun phrases of natural language set up discourse referents for future anaphoric coreference.

However, the reinstatiation of a variable $x$ in a state $s$ at the same time implies the loss of information that $s$ may have about $x$, and therefore, in DPL, the introduction of one discourse referent goes hand in hand with the exit of another. So, in fact, DPL fails an account of the fact the indefinites may introduce ‘novel’ objects, without interfering with (information about) other discourse referents introduced before.

It seems that, apart from information about the values of variables, another aspect of information about variables needs to be taken into account, viz., information about the domain of variables. When processing successive sentences in a discourse, the information that is passed on determines, first, which variables are under discussion, and, second, what the possible values of the variables are. Consequently, update of information need not solely consist of getting better informed about the values of the variables at issue, it may also involve the addition of variables to the universe of discourse. (In fact, in discourse representation theory and file change semantics, these two aspects have been separately encoded.) Clearly, with such a notion of information, the introduction of discourse referents can be conceived of as genuine updates.

In what follows, I will give a reformulation of DPL that gives formal expression to the insight that the introduction of discourse referents involves genuine updates. Like DPL, the system of interpretation that I propose (EDPL) gives a compositional interpretation of the language of predicate logic in terms of updates of information states. However, the information states which are used are more fine-grained than
those in DPL, since they consist of partial variable assignments. Information states in EDPL, thus, determine a domain of variables whose values are at issue, viz., the joint domain of the assignments in such states. Besides that, information states contain information about the values of the variables in their domain.

Since information states in EDPL model two aspects of information about variables, we can also distinguish two basic kinds of update of information. Update of information consists either of restricting the set of partial variable assignments by elimination, or of extending the domain of partial variable assignments, or, of course, of a mixture of both. The general notion of information update that then results rules a state t a possible update of a state s iff all assignments in t are an extension of an assignment in s. Clearly, under such a notion of update, t is an update of s iff t contains the information that s has about the values of the variables in the domain of s, but t may contain more information about more variables.

It goes without saying that for indefinite noun phrase to extend the domain of a state s they have to be associated with variables which are not in the domain of s. In DRT, for this reason, the so-called discourse representation algorithm is used to ensure that indefinites always associate with novel variables and in FCS felicity-conditions serve the same purpose. Likewise, also in EDPL it is required, on pain of undefinedness, that noun phrases introduce new variables and, consequently, do not mess up established information about variables in use.

In EDPL this is simply achieved by ruling the extension of the domain of a state s with a variable x undefined, if x is already in the domain of s. Consequently, EDPL involves partiality in two different respects. In the first place, EDPL’s semantics is stated as an update semantics for information states which are sets of partial variable assignments. In the second place, the interpretation function of EDPL itself is a partial function. Since the interpretation of a formula $\exists x \phi$ requires an extension of the domain of an information state with the variable $x$, it is undefined for a state that is already defined for $x$. Similarly, if a formula contains a free variable $x$, then the interpretation of the formula is undefined for a state that is undefined for $x$.

It must be stressed here that the envisaged partiality of EDPL has a technical motivation only. EDPL’s partial interpretation is not, at least not in the first place, intended to figure in an account of undefinedness phenomena in natural language, such as, for instance, presupposition failure (but, cf., Beaver [1992]). Partial interpretation in EDPL, like DPL’s non-eliminativity, merely expels nasty side effects of ‘infelicity’ translation.

3 Update semantics for DPL

In this section, I will first introduce the required notions of information and of information update in EDPL, and some operations on information states (section 3.1). In section 3.2 I give the semantics of EDPL, discuss some characteristic facts and show some examples. Section 3.3 treats entailment in EDPL, undefinedness in EDPL, and the relation with DPL.
3.1 Information states

If $D$ is our domain of individuals and $V$ the set of variables, then $S^X$, the set of information states about the values of $X \subseteq V$, and $S$, the set of all information states, are defined as follows:

**Definition 3.1 (Information states)**
- $S^X = \mathcal{P}(D^X)$
- $S = \bigcup_{X \subseteq V} S^X$

Given any domain of variables, the notions of minimal and maximal information states are as in DPL and US. For any domain of variables $X$, the minimal information state about the values of $X$ is $D^X$ (the state in which all valuations of $X$ are possible). A maximal information state about the values of $X$ is $\{i\}$ for any $i \in D^X$ (the state in which we know all about the values of variables in $X$). Furthermore, for any domain $X$, the absurd information state is $\emptyset$ (the information state that excludes every possibility).

A special set of information states is $S^\emptyset$, the possible states of information about the values of no variables. There are only two such states: the set containing the empty assignment, and the empty set. (In fact this is just the domain of truth values on their set-theoretic definition.) So, with respect to the empty domain the minimal and the maximal information state coincide. This reflects the fact that one can have no substantial information about no variables.

The fact that a state $t$ contains more information than a state $s$, or, in other words, that the state $t$ is an update of $s$ is defined as follows:

**Definition 3.2 (Update)**
- $t$ is an update of $s$, $t \leq s$, iff $\forall j \in t \exists i \in s: i \leq j$

If $t$ is an update of $s$, then every assignments in $t$ is an extension of some assignment in $s$. In that case $t$ contains at least the information that $s$ contains about the variables in the domain of $s$. Moreover, $t$ may contain information about variables which $s$ is silent about.

The update relation $\leq$ induces a partial order of information states:

**Fact 3.1** $\leq$ is a partial order
- reflexive ($s \leq s$)
- transitive (if $s \leq s'$ and $s' \leq s''$ then $s \leq s''$)
- antisymmetric (if $s \leq s'$ and $s' \leq s$ then $s = s'$)

As we will see below, interpretation in EDPL is a process of information growth. For any formula $\phi$ we will find that $s[\phi] \leq s$.

I now define some more helpful notions:

**Definition 3.3**
- $i$ is a partial element of $s$, $i \in s$, iff $\exists j \in s: i \leq j$

If $i$ is a partial element of $s$, I will also say that $i$ survives in $s$. 
Definition 3.4
For all states $s \in S^X$, $t \in S^Y$, if $X \subseteq Y$:
- $s \cap t = \{i \in s \mid i \in t\}$
- $s - t = \{i \in s \mid i \not\in t\}$
- $s \subseteq t$ iff $\forall i \in s: i \in t$

Operation $\cap$ is an operation labeled `state restriction`. The restriction of state $s$ by $t$ preserves the elements of $s$ which survive in $t$, i.e., the assignments in $s$ which have an extension in $t$. So, $s \cap t$ contains the information that $s$ contains supplemented with the information that $t$ contains about the variables in the domain of $s$. The operation $-$ is labelled `state subtraction`. Subtracting $t$ from $s$ we preserve the assignments in $s$ which do not survive in $t$, i.e., we eliminate the assignments in $s$ which have an extension in $t$. So, $s - t$ contains the information that $s$ has about the variables in the domain in $s$ supplemented with the information excluded by $t$. Finally, $s \subseteq t$, `s is a substate of t', iff all possibilities in $s$ survive in $t$.

The substate relation is used in the definition of $EDPL$-entailment. For $s$ to be a substate of $t$, $s$ must contain as much information about the values of the variables in the domain of $s$ as $t$. In other words, $s$ is a subset of the restriction of $s$ by $t$ ($s \subseteq t$ iff $s \subseteq s \cap t$ iff $s = s \cap t$). Notice that, if $s$ is a substate of $t$, the latter may contain information about the values of variables which are not in the domain of $s$.

Like the update relation, the substate relation is a partial order:

Fact 3.2 $\subseteq$ is a partial order
- reflexive ($s \subseteq s$)
- transitive ($s \subseteq s'$ and $s' \subseteq s''$ then $s \subseteq s''$)
- antisymmetric ($s \subseteq s'$ and $s' \subseteq s$ then $s = s'$)

We also have some kind of contraposition:

Fact 3.3
- $s - t \subseteq z$ iff $s - z \subseteq t$

So, if we substitute the absurd state for $z$, we find that $s - t \subseteq \emptyset$ iff $s \subseteq t$.

The last operation on states discussed here is the extension of the domain of a state. Let $i \leq_X j$ iff $i \leq j$ and the domain of $j$ is the join of that of $i$ with $X$. Then the extension of the domain of a state with a variable $x$ is defined as follows:

Definition 3.5 (Domain extension)
- $\forall s \in S^X: s[x] = \{j \mid \exists i \in s: i \leq \{x\}, j\}$ if $x \not\in X$; undefined otherwise

Extending the domain of $s$ with $x$ gives a state $s[x]$ that only differs from $s$ in that it is defined for $x$. The values of variables for which $s$ is defined the new state $s[x]$ contains precisely the same information as $s$, and about $x$ it is completely impartial: for each $i$ in $s$, and for each $d$ in $D$, there is an extension $j$ of $i$ in $s[x]$ that assigns $d$ to $x$. What is added is only the information `that $x$ has a value'.
3.2 Semantics of EDPL

We are now ready to turn to the semantics of EDPL. Like a DPL model, an EDPL model is a pair \( M = (D, F) \) consisting of a non-empty set of individuals \( D \) and an interpretation function \( F \) that assigns sets of \( n \)-tuples of objects to \( n \)-ary relation expressions. (Again, I omit reference to \( M \) whenever this does not lead to confusion.) EDPL interpretation is defined as a partial update function on information states:

**Definition 3.6 (Semantics of EDPL)**

- \( s[Rx_1 \ldots x_n] = \{ i \in s \mid (i(x_1), \ldots, i(x_n)) \in F(R) \} \) if defined
- \( s[\neg \phi] = s - s[\phi] \)
- \( s[\exists x \phi] = s[x][\phi] \)
- \( s[\phi \land \psi] = s[\phi][\psi] \)

The clauses in this definition closely correspond to those in the definitions of the semantics of \( DPL \) and \( US \). The interpretation of an atomic formula is the same as in DPL. Negation is like in \( US \), apart from the fact that we use state-, in stead of set-, subtraction. Like in \( DPL \), the existential quantifier \( \exists x \) introduces arbitrary values for \( x \), the difference being that domain extension is used, instead of reinstatation. Finally, like in \( DPL \) and \( US \), conjunction is interpreted as function composition.

The most significant difference with \( DPL \) and \( US \) is that in EDPL the interpretation of a formula \( \phi \) can be undefined for certain states. If an atomic formula contains a variable for which a state \( s \) is undefined, then the interpretation of the formula is undefined for that state \( s \); similarly, the interpretation of \( \exists x \phi \) is undefined for a state \( s \) which is already defined for \( x \). Furthermore, undefinedness persists in the following way. If (the interpretation of) \( \phi \) is undefined for \( s \), then \( \neg \phi \) and \( \phi \land \psi \) are undefined for \( s \). Furthermore, if \( \phi \) is undefined for \( s[x] \), then \( \exists x \phi \) is undefined for \( s \), and if \( \psi \) is undefined for \( s[\phi] \), then \( \phi \land \psi \) is undefined for \( s \). In section 3.3, I will give a more detailed account of the undefinedness phenomena in EDPL. Before that, I will assume definedness whenever this is unlikely to give rise to confusion.

**Some characteristic facts**  Like \( DPL \), EDPL has a distributive semantics. However, unlike \( DPL \), EDPL is not really non-eliminative, although it is not really eliminative either. Instead of eliminativity, what is relevant here is that EDPL interpretation always yields updates:

**Fact 3.4 (Update and distributivity)**

- \( s[\phi] \leq s \) if defined
- \( s[\phi] = \bigcup_{i \in s} \{ i \} [\phi] \) iff \( \forall i \in s : \{ i \} [\phi] \) is defined; undefined otherwise

EDPL updates are not purely eliminative, since a state \( s[\phi] \) may contain extensions of assignments in \( s \). Distributivity holds modulo definedness.

\( DPL \)'s characteristic donkey equivalences are retained:

**Fact 3.5 (Donkey equivalences)**

- \( (\exists x \phi \land \psi) \Leftrightarrow (\exists x (\phi \land \psi)) \)
- \( ((\phi \land \psi) \land \chi) \Leftrightarrow (\phi \land (\psi \land \chi)) \)
(∃xφ → ψ) ⇔ (∀x(φ → ψ))
(φ ∧ ψ) → χ) ⇔ (φ → (ψ → χ))

So, like DPL, EDPL accounts for the fact that indefinite noun phrases (existential quantifiers) in one sentence may bind pronouns (free variables) in another. Furthermore, an indefinite in the antecedent of an implication gets universal force.

Negation EDPL uses a notion of negation that looks more like that of US than that of DPL. If we interpret ¬φ in a state s, φ is interpreted in s and the result is subtracted from s. Still, this notion of negation is an adequate reproduction of DPL negation, since EDPL interpretation, besides being distributive, has the update property. Recall that s[¬φ] in DPL is the set of assignments i ∈ s such that {i}[φ] = ∅. In EDPL, i ∈ s[¬φ] iff i ∈ s and no j ≥ i is in s[φ]. Now, distributivity and update, this means that i ∈ s and {i}[φ] = ∅, like in DPL.

Let us look at one example:

(7) No man sees her
¬∃y(My ∧ Syx)

The interpretation of this sentence in a state s is s[¬∃y(My ∧ Syx)], which is s − s[y][My][Syx]. The state s[y][My][Syx] is the following set of assignments:

(8) {j | ∃k ∈ s: k ≤ (y) j & j(y) ∈ F(M) & (j(y), j(x)) ∈ F(S)}

So, the state s − s[y][My][Syx] is the following set of assignments:

(9) {i ∈ s | ¬∃j: i ≤ (y) j & j(y) ∈ F(M) & (j(y), j(x)) ∈ F(S)}

In other words, the result of interpreting ¬∃y(My ∧ Syx) in s is the set of assignments in s that assign x an individual for which no other individual can be found that is a man and sees her.

Since the interpretation of a formula ¬φ in a state s returns those i in s which do not survive in s[φ], no formula ¬φ brings about extension of the domain of discourse. As a consequence, the law of double negation doesn’t hold in EDPL. The double negation of a formula involves state restriction:

Fact 3.6
s[¬¬φ] = s − (s − s[φ]) = s ∩ s[φ]

The double negation of a formula φ gives us the restriction of s by the update of s with φ. So, ¬¬φ imposes the same restrictions as φ imposes on the assignments in a state s, but it cancels possible extensions of the domain of s brought about by φ. Like Groenendijk and Stokhof, I will call the double negation of φ the static closure of φ, and write ↓φ for ¬¬φ.

As usual, the universal quantifier and implication are defined in terms of the existential quantifier, negation and conjunction. Their interpretation is the following:

Fact 3.7
s[∀xφ] = s[¬∃x¬φ] = s − (s[x] − s[φ])
s[φ → ψ] = s[¬(φ ∧ ¬ψ)] = s − (s[φ] − s[ψ])
• \( s[\forall x(\phi \rightarrow \psi)] = s[\neg \exists x \neg (\phi \land \neg \psi)] = s - (s[x][\phi] - s[x][\phi][\psi]) \)

The interpretation of the formula \( \forall x(\phi \rightarrow \psi) \) in a state \( s \) preserves all those assignments \( i \in s \) such that every extension of \( i \) in \( s[x][\phi] \) survives in \( s[x][\phi][\psi] \). So, the interpretation of \( \text{Every farmer who owns a donkey beats it} \) in a state \( s \), \( s[\forall x((Fx \land \exists y(Dy \land Oxy)) \rightarrow Bxy)] \), is the set of assignments \( i \) in \( s \) such that every extension of \( i \) in \( s[x][Fx][y][Dy][Oxy][Bxy] \) has an extension in \( s[x][Fx][y][Dy][Oxy][Bxy] \). Put more simply, this requires of an \( i \) in \( s \) that under every valuation of \( x \) and \( y \) such that \( x \) is a farmer and \( y \) a donkey that \( x \) owns, we find that \( x \) beats \( y \).

Digression It has been argued, for instance by Schubert and Pelletier [1988], that the strong readings of donkey sentences are misguided, or, at least, not the only reading these sentences have. Schubert and Pelletier’s favourite example is the following sentence, which I label the ‘dime implication’:

(10) If I have a dime in my pocket, I’ll put it in the parking meter.

On its most natural reading this sentence says that if I have a dime in my pocket, then I will throw one in the meter. However, if we interpret the dime implication, like the donkey implication, as one of strong implication, then the sentence would imply that I throw all the dimes I have in my pocket in the meter. As concerns the present example, this strong reading seems quite odd.

It is very well possible to define a notion of weak implication that assigns conditional sentences the weak truth conditions that Schubert and Pelletier argue for:

**Definition 3.7 (Weak implication)**

• \( s[\phi \iff \psi] = (s - s[\phi]) \cup (s \cap s[\phi][\psi]) \)

However, in section 4, we will see that we do not need such an independent notion of weak implication, since it in fact fits in a more general scheme of (universal) adverbial quantification. That concludes the digression.

### 3.3 Truth, undefinedness and the relation with DPL

Truth and entailment are defined as follows:

**Definition 3.8 (Truth and entailment in EDPL)**

• \( \phi \) is true in \( s \), \( s \models \phi \), iff \( s \subseteq s[\phi] \)
  \( \phi \) is false in \( s \), \( s \notmodels \phi \), iff \( s[\phi] = \emptyset \)
• \( \phi_1, \ldots, \phi_n \models \psi \) iff \( \forall M, s \in S^X: s[\phi_1]_M \ldots [\phi_n]_M \models_M \psi \)

A formula \( \phi \) is true in a state \( s \) if the update of \( s \) with \( \phi \) does not add any information about the variables in the domain of \( s \). If a formula is true in a state \( s \), its negation is false in \( s \), and vice versa.

The notion of entailment, which has the dynamics of DPL entailment, is defined in terms of the US-style notion of truth. A sequence of premisses entails a conclusion if the update of any state of information about a certain domain of variables with the premisses always produces a state of information in which the conclusion is true.
Notice that a valid conclusion need not be true in the update of all states with the premisses. The conclusion of an entailment is only required to be true in the update with the premisses of all states with a certain domain of variables. The reason for this is that we should not completely exclude the possibility of undefinedness. For instance, for states with a domain \( Y \) such that \( x \notin Y \), the interpretation of \( x = x \) is undefined. Nevertheless, on the present definition of entailment \( \Vdash_X (x = x) \) is valid, that is, for any domain \( X \) such that \( x \in X \). Similarly, for all states \( s \in S^Y \), if \( x \notin Y \), then \( \exists y F y \) is not guaranteed to be defined for \( s[\exists x F x] \). Still, the entailment \( \exists x F x \vdash_X \exists y F y \) is valid, for a domain \( X \) such that \( x, y \notin X \).

Notice that a valid inference with respect to a domain of variables \( X \) may be *undefined* with respect to states in different domains, but can never be *falsified* in states with different domains. If \( \phi_1, \ldots, \phi_n \) entail \( \psi \) with respect to a domain \( X \), and if for some state \( s \notin S^X \) the update with \( \phi_1, \ldots, \phi_n, \psi \) is defined, then \( \psi \) is true in the state \( s[\phi_1] \ldots [\phi_n] \):

**Fact 3.8**

- If \( \phi \vdash_X \psi \), then for all \( s \), if \( s[\psi] \) is defined, then \( s[\phi] \vdash \psi \)

For this reason I will drop the subscript \( X \) whenever irrelevant.

The deduction theorem holds in EDPL:

**Fact 3.9**

- \( \Gamma, \phi \vdash \psi \) iff \( \Gamma \vdash \phi \rightarrow \psi \)

Contrary to the substate relation, EDPL-entailment is not transitive. Whereas \( \exists x F x \) entails \( \exists y F y \) and \( \exists y F y \) entails \( F y \), \( \exists x F x \) does not entail \( F y \).

After having discussed undefinedness in EDPL, I will give a restricted version of transitivity which holds of EDPL entailment.

**Undefinedness** Now let us look more specifically at the presuppositions of EDPL formulas and their projection behaviour in compound formulas. We have seen that presuppositions are triggered by free variable occurrences and existential quantifiers. A formula containing a free variable \( x \) presupposes that a state contains information about the value of \( x \), and an existential quantifier \( \exists x \) presupposes that a state is undefined for \( x \). Furthermore, and this is characteristic of presuppositions, the presuppositions of the negation of a formula \( \phi \) are the presuppositions of \( \phi \).

In the processing of conjunctions falsity persists and undefinedness persists. If \( s[\phi] \) returns the absurd information state (or is undefined), then \( s[\phi \land \psi] \) also returns the absurd state (eq. is undefined). This gives rise to a Karttunen/Heim-style presupposition projection behaviour (Karttunen [1974], Heim [1983]). If \( \pi(\phi) \) expresses the presuppositions of \( \phi \), then, in Karttunen and Heim’s systems, \( \pi(\neg \phi) = \pi(\phi) \) and \( \pi(\phi \land \psi) = \pi(\phi \rightarrow \psi) = \pi(\phi) \land (\phi \rightarrow \pi(\psi)) \).

The presuppositions of EDPL formulas depend on the domain of states with respect to which they are interpreted. Therefore, the presuppositions of a formula must be calculated by means of an indexed function \( \pi_X \) that for any formula \( \phi \) gives a formula \( \pi_X(\phi) \) which states the presuppositions that \( \phi \) carries concerning states of
information about a domain of variables $X$. This function is defined as follows (for ease of exposition I only consider atomic formulas with only one variable $x$):

**Definition 3.9**
- $\pi_X(Fx) = (x = x)$ if $x \in X$
- $\pi_X(x \neq x)$ if $x \not\in X$
- $\pi_X(\exists x \phi) = (x \neq x)$ if $x \in X$
- $\forall x \pi_X(\bigcup_{x \in X})(\phi)$ if $x \not\in X$
- $\pi_X(\neg \phi) = \pi_X(\phi)$
- $\pi_X(\phi \land \psi) = \pi_X(\phi) \land (\phi \rightarrow \pi_X(\psi))$ if $\phi$ is a test

A formula $\phi$ is a test if $\phi$ is an atomic formula $Rx_1 \ldots x_n$ or a negation $\neg \psi$. The presuppositions of conjunctions with other first conjuncts are computed using the equivalences $(\exists x \phi \land \psi) = \exists x(\phi \land \psi)$ and $(\phi \land \psi) \land \chi = (\phi \land (\psi \land \chi))$.

**Fact 3.10**
- $\forall s \in S^X: s[\pi_X(\phi)][\phi]$ is defined

**Fact 3.11**
- $\forall s \in S^X: s[\phi]$ is defined iff $s \models \pi_X(\phi)$

So, for a state $s$ with domain $X$, $\phi$ is defined for $s$ if and only if the formula that expresses $\phi$'s presuppositions is true in $s$.

Using the presupposition function $\pi$, we can now establish a restricted version of transitivity in $EDPL$:

**Fact 3.12 (Restricted transitivity)**
- If $\phi \models_X \psi$ and $\psi \models_X \chi$, then $\pi_X(\phi \land \chi) \models_X \chi$

This fact says that if $\phi$ entails $\psi$ with respect to $X$, and $\psi$ entails $\chi$, then $\phi$ entails $\chi$ with respect to $X$, whenever the presuppositions of $\phi \land \chi$ with respect to $X$ are met. This restricted version of transitivity in effect excludes cases where the mediating formula $\psi$ introduces variables which are free in $\chi$.

Let us briefly look at the counterexample to transitivity given above. In $EDPL$ we have that $\exists x \neg \phi \models_X \exists y \neg \chi$, where $x, y \not\in X$, and $\exists y \neg \chi \models Fy$. Restricted transitivity gives $\pi_X(\exists x \neg \phi \land \neg \chi), \exists x \neg \chi \models_X Fy$.

Computing the presuppositions of the two formulas with respect to $X$, we obtain $\pi_X(\exists x \neg \phi \land \neg \chi) = \forall x((x = x) \land (F \neg \chi \rightarrow \forall y(y \neq y)))$. With respect to a state $s \in S^X$, this equals $\forall x \neg \phi$. So, restricted transitivity tells us that $\forall x \neg \phi, \exists x \neg \chi \models_X Fy$, which, clearly, is correct.

**DPL and EDPL** I now turn to the relation between $EDPL$ and $DPL$. The relation between the two systems is partially characterized by the following fact ($s^V$ is the set of total extensions of $s$, i.e., $s^V = \{g \in D^V \mid \exists i \in s: i \leq g\}$):

**Fact 3.13**
- $s \models \phi$ in $EDPL$ iff $s^V \models \phi$ in $DPL$ if defined
In order to complete the picture of the relation between EDPL and DPL, we only need to combine the facts 3.13 and 3.11. Fact 3.13 relates truth in EDPL to truth in DPL on the provision of definedness. Fact 3.11 relates definedness in EDPL to truth in EDPL. The facts 3.13 and 3.11, thus, jointly entail the following fact:

**Fact 3.14**

- $\forall s \in S^X: s \models \phi$ in EDPL iff $s^V \models (\pi_X(\phi) \land \phi)$ in DPL

4 Quantifiers

In this section I introduce quantifiers in EDPL and show that we can give a perspicuous and uniform interpretation of adnominal and adverbial quantifiers, symmetric as well as asymmetric. I start with adverbs of quantification.

4.1 Adverbs of quantification (unselective)

Lewis [1975] argues that in many cases adverbs of quantification (like always, sometimes, usually) unselectively quantify over the values of 'free variables' (or indefinite noun phrases) in their restrictive clause. The examples Lewis discusses are of the following form:

(11) Sometimes/usually/always if a man owns a donkey, he beats it.

Lewis points out that the quantifying adverbs unselectively quantify over the values of the free variables, eq., indefinites, in the restrictive clause, and this phenomenon has been, quite elegantly, formalized in the frameworks of FCS, DRT and DPL. In DPL, for instance, the formula Always($\phi$)($\psi$) tests, given an initial assignment $g$, whether all assignments that verify $\phi$ with respect to $g$ are assignments with respect to which $\psi$ is true.

EDPL, too, allows a straightforward interpretation of unselectively quantifying adverbs. For any adverb of quantification $A$, with its usual set-theoretic interpretation $A'$, the interpretation is defined as follows:

**Definition 4.1 (Adverbs of quantification (symmetric))**

- $s[A(\phi)(\psi)] = \{i \in s \mid A'(\{j \in s[\phi] \mid i \leq j\})(\{j \in s[\phi][\psi]\})\}$

So, if we interpret If a farmer owns a donkey he always beats it in $s$, we get $s[Always(\exists x(Fx \land \exists y(Dy \land Oxxy))(Bzyy)]$. This is the set of assignments $i$ in $s$ such that on every extension of $i$ to $x$ and $y$, if the value of $x$ is a farmer who owns a donkey which is the value of $y$, then the value of $x$ beats the value of $y$. In other words, this formula tests whether all pairs of a farmer and a donkey he owns are pairs of which the first element beats the second element. A second example is If a man gives her a present, she usually thanks him for it, Usually(3y(My \land 3z(Pz \land Gyzx))(Txyz)). Interpreted in a state $s$, this example gives all those $i$ in $s$ in return that assign $x$ an individual that renders thanks in most cases in which a man gives her a present.

There are some interesting correspondences between the sentential connectives of EDPL and adverbs of quantification ($\phi$ abbreviates $\neg\neg\phi$):
Fact 4.1

- \textit{Sometimes}(\phi)(\psi) \iff \downarrow(\phi \land \psi)
- \textit{Always}(\phi)(\psi) \iff (\phi \rightarrow \psi)
- \textit{Never}(\phi)(\psi) \iff \neg(\phi \land \psi)

So conjunction (disregarding its external dynamics) and implication fit in the more general scheme of adverbial quantification.

4.2 Adverbs of quantification (asymmetric)

Adverbs of quantification not always unselectively quantify over the values of all variables introduced in their restriction. Several authors (Bäuerle and Egli [1985], Root [1986], Rooth [1987] and Kadmon [1987], see also Heim [1990] and Chierchia [1992]) have discussed examples in which adverbial quantifiers seem to involve quantification over the values of a proper subset of the introduced variables. Following Rooth and Kadmon, I call this kind of quantification \textit{asymmetric}. We find it in the following sentences:

(12) If a farmer owns a donkey, he is usually rich.
(13) If a \textit{drummer} lives in an apartment complex, it is usually half empty.
(14) If a drummer lives in an \textit{apartment complex}, it is usually half empty.

On its most natural reading, the adverb \textit{usually} in the first example quantifies over farmers who own a donkey and not over farmer-donkey pairs. The sentence says that most farmers who own a donkey are rich. (If the adverb is taken to quantify unselectively, we get the different reading that for most pairs consisting of a farmer and a donkey he owns, it holds that the farmer is rich.) The second and third example are different for a similar reason. In the second example, with focal stress on \textit{drummer}, we (may) find quantification over apartment complexes in which a drummer houses. The example then states that most apartment complexes where a drummer lives are usually half-empty. In the third example, where we find focal stress on \textit{apartment complex}, the adverb may be taken to quantify over drummers. On this reading, the sentence says that most drummers that live in an apartment complex live in an half empty apartment complex.

In EDPL, asymmetric adverbs naturally fit into the general scheme of adverbial quantification. Since EDPL has the update property, we can (unselectively) quantify over the assignments that satisfy the restriction \(\phi\) of an adverb by considering all extensions of \(i\) in \(s[\phi]\), for any assignment \(i\) in an input state \(s\). Now, in case of asymmetric quantification, we only need to take into account extensions of \(i\) that survive in \(s[\phi]\), and test whether these extensions also survive in further update with the nuclear scope of the adverb.

So, assume that an asymmetric adverb of quantification comes with a set of selection indices \(X\) that selects the variables whose values the adverb quantifies over. Its interpretation then is defined as follows:

\textbf{Definition 4.2 (Adverbs of quantification (asymmetric))}

\begin{itemize}
    \item \(s[A_X(\phi)(\psi)] = \{i \in s \mid A'(\{j \in s[\phi] \mid i \leq_X j\})(\{j \in s[\phi]\}(\psi))\}\)
\end{itemize}
Let us briefly consider two mutually related examples.

(15) If a man gives her a present, she usually thanks him for it

\[ \text{Usually}_{(y)}(\exists y(My \land \exists z(Pz \land Gyxz))(Txyz)) \]

Interpreted in a state \( s \), this example returns all those \( i \) in \( s \) that assign \( x \) an individual that renders thanks to most men that gives her a present, irrespective of the number of presents given.

(16) If a man gives her a present, she usually thanks him for it

\[ \text{Usually}_{(z)}(\exists y(My \land \exists z(Pz \land Gyxz))(Txyz)) \]

When interpreted in a state \( s \), this example returns all those \( i \) in \( s \) that assign \( x \) an individual that renders thanks for most presents given by a man, irrespective of the number of men that give it.

We find the following equivalences:

**Fact 4.2**

- \( \text{Sometimes}_X(\phi)(\psi) \Leftrightarrow \text{Sometimes}(\phi)(\psi) \)
- \( \text{Never}_X(\phi)(\psi) \Leftrightarrow \text{Never}(\phi)(\psi) \)
- \( \text{Always}_X(\phi)(\psi) \Leftrightarrow (\phi \leftarrow \psi) \)

(for \( X \) a subset of the variables introduced by \( \phi \))

So, for the adverbs *sometimes* and *never* it makes no difference whether or not they select variables to quantify over. This is as it should be, since there seems to be no evidence whatsoever that there are distinct asymmetric readings of these adverbs.

On the other hand, for the adverbs *usually* and *always*, it does make a difference whether or not they select variables for asymmetric quantification, and which variables they select. Furthermore, we see that the weak implication \((\leftarrow)\) addressed in the digression of section 3.2, now appears to be a borderline case of asymmetric adverbial quantification, i.e., universal quantification over the values of an empty set of selection indices.

That the weak implication is one of the many forms of asymmetric quantification, may be further substantiated by slightly varying the exemplary dime implication:

(17) If I have a dime in my pocket, I throw it in the parking meter.

\[ (\exists y(Dy \land Pyy) \leftarrow Ty) \Leftrightarrow \text{Always}_y(\exists y(Dy \land Pyy))(Ty) \]

(18) If a man has a dime in his pocket, he throws it in the parking meter.

\[ \text{Always}_{(x)}(\exists x(Mx \land \exists y(Dy \land Pyz))(Txz)) \]

On its most natural reading, the first example was argued to state that if I have a dime in my pocket, then I throw one in the meter, and this reading is captured by interpreting the conditional sentence as one of weak implication. However, the second example, which is a minor variation of the first, is most likely interpreted as stating that every man who has a dime, throws one in the meter. Neither the weak, nor the strong, reading of the implication gives us this. The preferred reading of this example is strong, i.e., universal, with respect to the parkers which have a dime in their pocket, and weak with respect to the dimes they throw in the meter. And, in fact, if the sentence is understood as asymmetrically quantifying over the parkers,
as indicated by its translation above, this reading results. Notice, that the two dime sentence, thus, can be interpreted in a uniform way.

4.3 Adnominal quantifiers

Internally dynamic generalized quantifiers, with an interpretation proposed by several authors (Chierchia [1992] to start with, see also van den Berg [1991] and van Eijck and de Vries [1991]) are easily introduced in a way that neatly fits in with the interpretation of quantifying adverbs. Let \( D \) be an arbitrary binary quantifier which has \( D' \) as its usual set-theoretic interpretation, then:

**Definition 4.3 (Binary quantifiers)**

- \( s[[Dx(\phi)(\psi)]] = \{ i \in s \mid D'(\{ j \in s[\phi] \mid i \leq \langle x \rangle j \})(\{ j \in s[\phi][\psi] \}) \} \)

We find the following correspondences with the unary quantifiers in EDPL:

**Fact 4.3**

- An \( x(\phi)(\psi) \) \( \iff \exists x(\phi \land \psi) \)
- No \( x(\phi)(\psi) \) \( \iff \neg \exists x(\phi \land \psi) \)
- Every \( x(\phi)(\psi) \) \( \iff \forall x(\phi \rightarrow \psi) \)

We see that the binary determiners \( an(n) \) and no have the same truth-conditional content as their usual first order paraphrases. (The only difference is that the binary quantifiers defined above are externally static.) We also see that EDPL licenses a weak and a strong reading of the quantifier every, both of which are intuitively motivated. If we treat it as a binary generalized quantifier, the weak reading results. This reading is appropriate for the sentence Every man who has a dime puts in the parking meter. On the other hand, if we translate every with the unary first order quantifier as \( \forall x(\phi \rightarrow \psi) \), then the strong reading results, and this is the proper reading of the (strong) donkey sentence.

It may be a bit tantalizing that we find two non-equivalent ways to translate natural language every in the language of EDPL. I must say I don’t know of any knock down arguments for taking every to be ambiguous or for choosing one interpretation in favour of the other. On the other hand, it is especially the determiner every which has been associated with both weak and strong readings in the literature. Now it is not so much a merit of EDPL that it accounts for both kinds of readings in a principled way, but it is a relative merit that it does do so without entailing ambiguity of all the other quantifiers.

A last observation concerns the relation between determiners and quantifying adverbs. If adverb \( A \) and determiner \( D \) are set-theoretically the same, then:

**Fact 4.4**

- \( A_{\langle x \rangle}(\exists x \phi)(\psi) \) \( \iff D x(\phi)(\psi) \)

So, here we see that the sentence If a man owns a donkey, he is usually rich on its asymmetric construal in which usually selects donkey owning men for quantification, is equivalent to the sentence Most men that own a donkey are rich.
References


Benthem, J. van, 1991, Language in action, Amsterdam: North-Holland


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


Heim, I., 1990, 'E-type pronouns and donkey anaphora', Linguistics and Philosophy 13, pp. 137–178

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


Karttunen, L., 1974, 'Presupposition and linguistic context', Theoretical Linguistics 1, pp. 181–194


Root, R., 1986, The semantics of anaphora in discourse, dissertation, University of Texas, Austin

19

