Journal of Semantics, 35, 2018, 263–303
doi:10.1093/jos/ffx020
Advance Access Publication Date: 29 January 2018
Article

OXFORD

Dynamic Update Anaphora Logic: A Simple Analysis of Complex Anaphora

Ezra Keshet
Department of Linguistics, University of Michigan, 440 Lorch Hall, 611 Tappan St, Ann Arbor, MI, 48109-1220, USA
E-mail: keshet@alum.mit.edu

First version received 06 October 2016; Second version received 11 December 2017; Accepted 20 December 2017

Abstract

An antecedent relationship may hold between an indefinite and a pronoun across non-quantified sentences ('Jane bought a book. She read it immediately.'), from the restrictor to the nuclear scope of a single quantified sentence ('Every woman who bought a book read it immediately.'; Geach 1962), and even across two quantified sentences ('Every woman bought a book. Most read it immediately.'; Sells 1985). First-generation dynamic semantic systems (Kamp 1981; Heim 1983; Groenendijk & Stokhof 1991) cannot handle anaphora across quantified sentences, but do handle the first two cases above quite elegantly: anaphora within a single quantified sentence (i.e. donkey anaphora) is a simple extension of anaphora across non-quantified sentences. Dynamic Plural Logic (van den Berg 1996) expands empirical coverage to anaphora across two quantified sentences (i.e. quantificational subordination) by generalising all three cases above to this worst case. This paper presents Dynamic Update Anaphora Logic, an alternative extended logic that introduces variables to store full dynamic clause meanings and compound variable terms to construct sets of individuals. Clause meanings may be retrieved later to handle donkey anaphora and quantificational subordination as straightforward extensions to non-quantified cross-sentential anaphora. This new system also handles paycheck pronouns (Karttunen 1969b), an empirical deficit of Dynamic Plural Logic.

An antecedent relationship may hold between an indefinite and a pronoun across non-quantified sentences (1a), from the restrictor to the nuclear scope of a single quantified sentence (1b) and even across two quantified sentences (1c). First-generation dynamic semantic systems (Kamp 1981; Heim 1983; Groenendijk & Stokhof 1991) handle the first two cases above quite elegantly. For instance, anaphora within a single quantified sentence (i.e. donkey anaphora) is a simple extension of anaphora across non-quantified sentences. This is plainly shown in (2); a Dynamic Predicate Logic (DPL) (Groenendijk & Stokhof 1991)

formula for (1b) is almost identical to one for (1a). ('$[x]$' is the dynamic equivalent of existential quantification.)

(1) a. A woman bought a book. She read it immediately.
b. No woman who bought a book read it immediately.
c. Every woman bought a book. Most of them read it immediately.

(2) a. (1a) $\rightsquigarrow$ $[w]; woman(w); [b]; book(b); bought(w, b); read(w, b)$
b. (1b) $\rightsquigarrow \sim ([w]; woman(w); [b]; book(b); bought(w, b); read(w, b))$

These first systems do not capture plurals, though, nor do they handle anaphora across quantified sentences.

Dynamic Update Anaphora Logic (DUAL), newly presented here, extends first-generation dynamic systems to handle plurals and complex anaphora without radically altering their elegant account of simple anaphora. Consider (3), the DUAL formula for (1c), which illustrates two innovations. First, the three uppercase variables W_1, W_2 and W_3 store full dynamic clause meanings as visualised in (4). Second, compound terms like '$W_1.w$' extract all possible values for a (standard) variable inside one such clause meaning, as shown in (5). Finally, the clauses '$EVERY(W_1.w, W_2.w)$' and '$MOST(W_2.w, W_3.w)$' assert the appropriate relationship between their two set arguments.

(3) $(W_1 : [w]; woman(w)); (W_2 : W_1; [b]; book(b); bought(w, b)) ;$
$EVERY(W_1.w, W_2.w); (W_3 : W_2; read(w, b)) ; MOST(W_2.w, W_3.w)$

(4) $\underbrace{[w]; woman(w)}_{W_1}$ 'There's a woman...'
$\underbrace{\underbrace{\qquad\qquad ; [b]; book(b); bought(w, b)}_{W_2}}$ '... and she bought a book ...'
$\underbrace{\qquad\qquad\qquad\qquad\qquad ; read(w, b)}_{W_3}$ '... and she read it.'

(5) a. $W_1.w \approx$ all women
b. $W_2.w \approx$ all women who bought a book
c. $W_3.w \approx$ all women who bought a book and read it immediately.

Notice that DUAL treats the antecedent–pronoun relationships in (1c) essentially the same way as in non-quantified cases. For instance, as indicated in (4), the denotation of W_3 is identical to the formula for (1a) in (2a), once the meanings for W_2 and W_1 are replaced by their denotations. This means that the pronoun *it* effectively takes the phrase *a book* as its antecedent, despite the fact that this antecedent is embedded under a quantifier in a previous sentence.

Besides the cases discussed so far, DUAL also handles discourse plurals, strong and weak donkey anaphora and telescoping, amongst other phenomena. Notably, this new system handles so-called paycheck pronouns (Karttunen 1969b; Jacobson 1997, 2000), an empirical deficit of existing plural logics.

The paper proceeds as follows. Section 1 immediately below introduces dynamic semantics in more detail and presents the formal details of Dynamic Predicate Logic, including a discussion of its empirical deficits. Next, Section 2 introduces the DUAL system and Section 3 discusses the applications of the new system. Section 4 presents and critiques Dynamic Plural Logic, a prior dynamic semantic system to handle plurals and complex

anaphora. Finally, Section 5 gives an overview of how DUAL fits in with prior developments in logic.

1 DYNAMIC SEMANTICS

Under a dynamic view of meaning, part of what speakers do as they engage in discourse is to keep track of who and what is under discussion. According to theories such as Discourse Representation Theory (DRT, Kamp 1981), File Change Semantics (FCS, Heim 1983) and Dynamic Predicate Logic (DPL, Groenendijk & Stokhof 1991), each point in a discourse is associated with an *information state*, which tracks ways to assign individuals to a number of labelled slots or *variables*.

To make this more concrete, consider the following short discourse. The superscript indices on indefinites (following Barwise 1987) intuitively represent the introduction of a new variable and the subscripts on pronouns represent the retrieval of a variable's value.

(6) a. A^{w} woman entered the shop.
b. She_{w} bought a^{b} book.
c. She_{w} read it_{b} immediately.

Imagine a simplified world consisting of three books, Anna Karenina (A), Beloved (B) and Catch-22 (C), and two women, Delores (D) and Emily (E). If we know nothing about how to assign these individuals to our two variables w and b, our information state will look like (7):

(7)
$$\left\{ \begin{bmatrix} w \mapsto \text{A} \\ b \mapsto \text{A} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{A} \\ b \mapsto \text{B} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{A} \\ b \mapsto \text{C} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{A} \\ b \mapsto \text{D} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{A} \\ b \mapsto \text{E} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{B} \\ b \mapsto \text{A} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{B} \\ b \mapsto \text{B} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{B} \\ b \mapsto \text{C} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{B} \\ b \mapsto \text{D} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{B} \\ b \mapsto \text{E} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{C} \\ b \mapsto \text{A} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{C} \\ b \mapsto \text{B} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{C} \\ b \mapsto \text{C} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{C} \\ b \mapsto \text{D} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{C} \\ b \mapsto \text{E} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{A} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{B} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{C} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{D} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{E} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{E} \\ b \mapsto \text{A} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{E} \\ b \mapsto \text{B} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{E} \\ b \mapsto \text{C} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{E} \\ b \mapsto \text{D} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{E} \\ b \mapsto \text{E} \end{bmatrix} \right\}$$

This state contains all 25 possible pairings of individuals. (Each component pairing is known as an *assignment*).

Before the discourse starts, any of these 25 assignments could be correct. Each sentence updates the discourse information state, adding or removing assignments, an action called an *update*. This is illustrated for (6) in (8):

(8) a. $\xrightarrow{(6a)}$
$$\left\{ \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{A} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{B} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{C} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{D} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{E} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{E} \\ b \mapsto \text{A} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{E} \\ b \mapsto \text{B} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{E} \\ b \mapsto \text{C} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{E} \\ b \mapsto \text{D} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{E} \\ b \mapsto \text{E} \end{bmatrix} \right\}$$

b. $\xrightarrow{(6b)} \left\{ \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{A} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{B} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{C} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{E} \\ b \mapsto \text{B} \end{bmatrix} \right\}$

c. $\xrightarrow{(6c)} \left\{ \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{A} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{E} \\ b \mapsto \text{B} \end{bmatrix} \right\}$

Sentence (6a) restricts the information state so that only woman who entered the shop can fill slot *w*, removing any assignments that violate this restriction. The result, assuming both Delores and Emily entered the shop, is depicted in (8a). Next, (6b) further restricts the information state to assignments where *b* is a book that *w* bought. Assuming that Delores bought all three books, but Emily only bought Beloved, the result is shown in (8b). Finally, (6c) removes those assignments where *w* did not read *b* immediately. Assuming Delores read only Anna Karenina immediately and Emily read only Beloved immediately, the resulting information state is shown in (8c).

The dynamic effect of an indefinite, such as *a woman* or *a book*, is to introduce a new index into a discourse. Dynamic theories guarantee that a discourse information state as updated by a^{w} *woman entered*, for instance, will have values for *w* representing all women who entered. This guarantee was trivially satisfied in (8) since we started from a state of complete ignorance.[1] If information is already represented in an information state, though, it is overwritten by an indefinite, as visualised in (9):

(9) $\left\{ \begin{bmatrix} w \mapsto \text{A} \\ b \mapsto \text{A} \end{bmatrix} \right\} \xrightarrow{(6a)} \left\{ \begin{bmatrix} w \mapsto \text{D} \\ b \mapsto \text{A} \end{bmatrix}, \begin{bmatrix} w \mapsto \text{E} \\ b \mapsto \text{A} \end{bmatrix} \right\}$

The input information state above on the left represents knowledge that the value for both *w* and *b* is Anna Karenina. The update in (6a) overwrites the information about *w* in order to guarantee that both women are possible values for *w*. This behaviour means that information states are *maximal* for all variables involved; they track all still-possible values for each variable.[2]

Another feature of information states is revealed as more variables are introduced: information states represent all the *quantificational dependencies* between their variables. For instance, (8b) contains each possible assignment whose value for *w* and value for *b* together make (6b) true (i.e. all cases where the woman *w* bought the book *b*). Such dependencies are only possible because information states are sets of assignments; if

1 Theories differ on what happens when prior information is stored about *w*: FCS, for instance, generates a presupposition failure if previous information exists about *w*, while DPL simply overwrites the existing information. The DPL approach is assumed for the remainder of the paper only because it is simpler to implement overall; a true choice between the two approaches is beyond the scope of this paper.

2 This may seem odd, since the speaker of (6a), in using the singular DP *a woman*, probably had one woman in mind (either Delores or Emily). However, the multiple outputs can be intuitively conceived of here as representing the uncertainty of the listener as to which woman was intended. See van den Berg (1996), Aside 5.1, p. 162.

variable values were represented, for instance, by sets of individuals for each variable, the connections between these values would be erased.

The solid arrows in the diagrams above effectively represent updates as functions over information states. A different way to compactly represent updates is suggested by the dotted lines in (9) that connect input and output assignments. In each update, in fact, input assignments connect to zero or more output assignments, effectively defining a (mathematical) relation over assignments. Somewhat confusingly, the assignments within such relations are often also referred to as states or *machine states*. This relational approach is adopted, for example by Groenendijk & Stokhof (1991), van den Berg (1996) and Brasoveanu (2007), and the remainder of this paper will adopt it as well.

Taking g as an input assignment and h as an output, we can represent the sentence meanings from (6) as in (10). Here, '$g[x]h$' is true iff g and h differ at most in their values for variable x.

(10) a. $W_1 = \left\{ \begin{array}{l} \langle g, h\rangle : g[w]h \text{ and } h(w) \text{ is a woman} \\ \qquad\quad \text{and } h(w) \text{ entered the shop} \end{array} \right\}$

b. $W_2 = \left\{ \begin{array}{l} \langle g, h\rangle : g[b]h \text{ and } h(b) \text{ is a book} \\ \qquad\quad \text{and } h(w) \text{ bought } h(b) \end{array} \right\}$

c. $W_3 = \left\{ \langle g, h\rangle : g{=}h \text{ and } g(w) \text{ read } g(b) \text{ immediately} \right\}$

Such relations can easily be converted into functions over information states: the function over information states S corresponding to a relation R is $\lambda S.\{h : \exists g{\in}S(gRh)\}$. (The expression $\langle a, b\rangle{\in}R$ is abbreviated as aRb as usual with relations.)

The meaning of the three-sentence discourse in (6) is the composition of the three relations in (10), the output of each serving as the input to the next, as shown in (11).

(11) $D = W_1 \circ W_2 \circ W_3 = \left\{\langle g, h\rangle : \exists k, l \left(gW_1k \;\&\; kW_2l \;\&\; lW_3h\right)\right\}$

A notion of truth relative to a single assignment can be defined for such a relational approach to dynamic semantics. This discourse is true for a given input g if there's an output h such that gDh. In other words, the discourse will be true whenever there is a woman who entered the shop, bought a book and read it immediately.

Dynamic systems such as DPL take advantage of updates' quantificational dependencies and maximality to explain quantified sentences and so-called *donkey anaphora* (Geach 1962), phenomena shown in (12) and analysed in (13):

(12) Every^f farmer who owns a^d donkey beats it$_d$.

(13) $\left\{\langle g, g\rangle : \forall h \;\left(gF_1h \Rightarrow \exists k(hF_2k)\right)\right\}$, where

a. $F_1 = \left\{ \langle g, h\rangle : \exists k \left(\begin{array}{l} g[f]k \text{ and } k(f) \text{ is a farmer and } k[d]h \text{ and} \\ h(d) \text{ is a donkey and } h(f) \text{ owns } h(d) \end{array} \right) \right\}$

b. $F_2 = \left\{\langle g, g\rangle : \quad g(f) \text{ beats } g(d)\right\}$

The update shown in (13), which represents the meaning of (12), relates pairs $\langle g, h\rangle$ such that all outputs of update F_1 given input g are inputs for update F_2 that yield a valid output. Notice that this mirrors the process of interpreting a simple discourse: the output of update F_1 is the input of update F_2 akin to the discourse 'A^f farmer owns a^d donkey. He$_f$ beats it$_d$.'

Unlike in (6), where maximality at best represented some sort of listener uncertainty, the maximality of F_1 here is crucial to capture the truth conditions properly. For instance,

if a farmer who owned a donkey was missing from the values of f in the output of F_1, (13) could be true even if this missing farmer did not beat their donkey. And again parallel to simple discourses, the correct quantificational dependencies are maintained, allowing a pronoun like it_d to correctly denote a donkey d bought by the farmer f being quantified over. The fact that every relevant farmer–donkey pair is represented also helps explain the so-called *strong* reading of donkey pronouns: a salient reading of (12) is true only when every farmer who owns a donkey beats every donkey they own. If any farmer–donkey pair were missing from F_1, this reading could not be properly derived.

The remainder of this section first introduces the formal system behind DPL and then discusses some empirical deficits of this system.

1.1 *Formalising DPL*

DPL (Groenendijk & Stokhof 1991) has the same syntax as (static) predicate logic with identity. Following van den Berg (1996) and Brasoveanu (2007), amongst others, the version of DPL presented below introduces minor differences to this syntax to emphasise the parallel with programming languages: ';' is used instead of '$\wedge$' for conjunction and '$[x];\phi$' instead of '$\exists x\phi$' for introducing variables. As usual, the semantic value $[\![\phi]\!]_M$ of a formula ϕ is calculated relative to a model $M = \langle D, I\rangle$, where D is a set of individuals and I is an interpretation function such that $I(P) \subseteq D^n$ for each n-place predicate P. (For instance, $I(book)$ would be the set of books in the model and $I(bought)$ would be the pairs $\langle w, b\rangle$ such that w bought b.) In practise the subscript M is suppressed in most instances.

The domains used in DPL are shown below:

(14) Domains

a. D : individuals in the model M
b. V : variables $a, \ldots, z$
c. $A(=V\rightarrow D)$: assignment functions / (machine) states g, h, k, l
d. $U(=\wp(A \times A))$: updates u

The set of individuals D is provided by the model. The set of variables V comprises the lowercase Latin letters. The (machine) states of DPL, members of A, are assignment functions from V to D, usually represented by g, h, k, l. (Thus when the term 'state' is used in this section, it refers to assignments rather than information states.) Finally the interpretation function $[\![\ ]\!]$ returns denotations in the set U, relations on A.

The full semantics for DPL follows. Using this semantics, a DPL formula ϕ is true given an input g iff there is an output h such that $g[\![\phi]\!]h$. Note that x and y (and decorated versions of these) are used below for arbitrary variables, P for an arbitrary predicate and ϕ, ψ for arbitrary formulas.

(15) DPL Semantics

$g[\![P(x_1,\ldots,x_n)]\!]h$	iff	$g{=}h$ & $\langle g(x_1),\ldots,g(x_n)\rangle \in I(P)$	[predicate formula]
$g[\![x{=}y]\!]h$	iff	$g{=}h$ & $g(x){=}g(y)$	[identity formula]
$g[\![\phi;\psi]\!]h$	iff	$\exists k(g[\![\phi]\!]k$ & $k[\![\psi]\!]h)$	[conjunction]
		(i.e., $g\ ([\![\phi]\!] \circ [\![\psi]\!])\ h$)	
$g[\![\sim\phi]\!]h$	iff	$g{=}h$ & $\neg\exists k(g[\![\phi]\!]k)$	[negation]
$g[\![[x]]\!]h$	iff	$g[x]h$	[random assignment]

Several examples are shown below to illustrate the interpretation of DPL formulas (the $\rightsquigarrow$ symbol links natural language statements with their logical translations). First consider a simple sentence illustrating random assignment, predicate formulas and conjunction:

(16) A^w woman bought a^b book $\rightsquigarrow [w]; woman(w); [b]; book(b); bought(w, b)$

a. $g[\![[w]]\!]h$ iff $g[w]h$

b. $g[\![woman(w)]\!]$ h iff $g{=}h$ & $g(w){\in}I(woman)$

c. $g[\![[w]; woman(w)]\!]h$

i. iff $\exists k(g[\![[w]]\!]k$ & $k[\![woman(w)]\!]h)$

ii. iff $g[w]h$ & $h(w){\in}I(woman)$

d. $g[\![[b]; book(b); bought(w, b)]\!]h$
iff $g[b]h$ & $h(b){\in}I(book)$ & $\langle h(w), h(b)\rangle{\in}I(bought)$

e. $g[\![[w]; woman(w); [b]; book(b); bought(w, b)]\!]h$
iff $g[w, b]h$ & $h(w){\in}I(woman)\&h(b){\in}I(book)$
$\&\langle h(w), h(b)\rangle{\in}I(bought)$

The subformula shown in (16a) represents the random assignment of the variable w. Any state g in A may act as the input for such a formula; the corresponding output state h, though, must be identical to g up to (and possibly including) its value for the input w. Next, (16b) is a simple predicate formula. Notice that its input and output state must be the same; any such formula, always having the same input and output state, is known as a **test**. Furthermore, this input/output state g for (16b) must set $g(w)$ to be a woman; no additional constraints are placed on g, though, so $g(w)$ may be any woman in the model. The conjunction of these two formulas is shown in (16c). No matter the input state g for this conjunction, the output h must return a woman for $h(w)$. No constraints are set on the input g, so any state is a possible input value, and once again the formula does not specify which woman $h(w)$ should be. Therefore, any state valuing w with a woman is a possible output for the formula.

The subformula shown in (16d) represents the rest of the original formula: there is a random assignment of the variable b and then the value for b is constrained to be a book that w bought. Finally, the interpretation of the full formula is shown in (16e). (The notation '$g[w, b]h$' is meant to convey that g and h may differ only in their values for the two variables w and b.)

Negation is shown in the following example:

(17) No woman entered $\rightsquigarrow \sim([w]; woman(w); entered(w))$

a. $g[\![[w]; woman(w); entered(w)]\!]h$
iff $g[w]h$ & $h(w){\in}I(woman)$ & $h(w){\in}I(entered)$

b. $g[\![\sim([w]; woman(w); entered(w))]\!]h$
iff $g{=}h$ & $\neg\exists h(g[w]h$ & $h(w){\in}I(woman)$ & $h(w){\in}I(entered))$

Notice that negated formulas are tests; their inputs match their outputs. In this case, the test holds of any state g as long as there is no woman who entered in the model. If there were any woman who entered, there would be an output h such that $g[w]h$ and $h(w)$ returned such a woman who entered. In this case, the formula would not hold of any states. (Thus, the set of states it holds of will either be A or $\varnothing$.)

DPL treats proper names as predicates accepting a single individual; therefore, the following formula illustrates an identity formula:

(18) Clark is Superman $\rightsquigarrow [c]; clark(c); [s]; superman(s); c{=}s$

As long as the predicates *clark* and *superman* hold of the same individual in the model, this formula will be true.

Additional logical operators can be defined in terms of these main ones:

(19) Formula Abbreviations

$\phi \vee \psi \;:= \sim(\sim\phi; \sim\psi)$ [disjunction]
$\phi \rightarrow \psi \;:= \sim(\phi; \sim\psi)$ [implication]
$\forall x\phi \;:= [x] \rightarrow \phi$ [universal quantification]

Notice that all these formulas end up being under the scope of negation, even '$\forall x\phi$' (whose entirely unabbreviated form would be '$\sim([x];\sim\phi)$'). Therefore, they are all tests: any random assignments within them will remain strictly local, not altering their output states.

1.2 *Discussion*

DPL handles simple quantified statements, cross sentential anaphora and even donkey anaphora quite well:

(20) a. A^w woman entered. She$_w$ bought a^b book. She$_w$ read it$_b$.
$\rightsquigarrow [w]; woman(w); entered(w); [b]; book(b); bought(w,b); read(w,b)$

b. Every^w woman who bought a^b book read it$_b$.
$\rightsquigarrow \forall w\,((woman(w); [b]; book(b); bought(w,b)) \rightarrow read(w,b))$
Or: $\sim([w]; woman(w); [b]; book(b); bought(w,b); \sim read(w,b))$

For instance, (20a) shows a simple discourse with pronouns referring back to indefinites in previous sentences. In DPL, this works no differently from pronouns in the same sentence, as long as discourse sequencing of sentences is treated as conjunction. (20b) shows a donkey anaphora example. As mentioned before, the reading derived is the strong reading of a donkey pronoun: if any woman bought multiple books, the sentence is true only if she read all of these books. This is perhaps clearest in the second version of the formula for (20b), which shows the negation explicitly; the formula will clearly be false if there is any value for *b* such that a woman bought *b* but did not read *b*. This is indeed a salient interpretation of the sentence, a satisfactory result in this case.

The system for donkey anaphora does suffer from two problems, though: a failure to handle *weak* donkey pronouns and the so-called *Proportion Problem*. The problem with weak donkey pronouns is illustrated by the most salient reading of the following sentence:

(21) Every^w woman who owns a^b car is driving it$_b$ now.

The translation for (21) is basically parallel to that for (20b); therefore, the formula for (21) implies that women who own multiple cars are driving all of them as we speak. Assuming that people normally can only physically drive one car at a time, this interpretation is not favoured. Instead, (21) seems to entail that any woman who owns multiple cars is driving one of them now.

The Proportion Problem arises in a slight variant of DPL, which allows a quantifier such as 'most'. Under the definition in (22), the formula in (23) effectively says that most pairs

$\langle w, b\rangle$ where w is a woman who bought the book b are such that w read b:

(22) $g[\![MOST(\phi, \psi)]\!]h$ iff for most k such that $g[\![\phi]\!]k$, also $k[\![\psi]\!]h$.

(23) Most women who bought a book read it.
$\rightsquigarrow MOST\Big(\left([w]; woman(w); [b]; book(b); bought(w, b)\right), read(w, b)\Big)$

Now, imagine that amongst the 10 women who bought books, there is one voracious reader who bought 20 books and read them all; and further imagine that the remaining nine bought one book each but did not read it. In this case, the DPL formula is true (for any input g): for most of the $\langle w, b\rangle$ pairs, w read b; it just so happens that it was the same woman in each case. However, the natural language sentence in (23) is not true in this scenario; natural language quantifiers like *most* seem to count individuals and not states or pairs.

In addition to these problems with donkey anaphora, DPL undergenerates in another way. Since so many formulas in DPL are tests, updates within the scope of a quantifier like *every* (and the antecedents and dependencies they introduce) are only temporarily maintained. This feature of DPL aims to explain the oddity of discourses like the following:

(24) Every woman bought a^c car. #I'm driving it$_c$ as we speak.

The systems are specifically designed so that antecedents within quantified sentences cannot value later pronouns. However, this restriction is too strict, for example in cases of *quantificational subordination* (Karttunen 1969a; Sells 1985), where dependencies seem to carry over from one quantified sentence to another. Consider the following, for instance, perhaps the report from a sales survey:

(25) a. Every^w woman bought a^b book.
b. Most^w of them$_{w?}$ read it$_b$ immediately.

First, DPL has no way to capture a plural, such as *them* in (25b). Next, notice that the dependency established in (25a) between women w and books b is maintained in (25b): in other words, most women w read the book w bought, not some other book. As indicated above, this dependency cannot be explained in first-generation dynamic systems, such as DPL. Therefore, Kamp & Reyle (1993), van den Berg (1996), Krifka (1996a), Nouwen (2003) and Brasoveanu (2007), amongst others, all propose revisions to classic dynamic systems partly aimed at capturing cases like (25). For instance, van den Berg (1996) redefines updates as relations over information states (or, equivalently, functions over sets of information states). With appropriate additions to the logic, this move suffices to capture quantificational subordination and related phenomena. However, whereas DPL maintained a simple analysis of simple discourses like (6) and minimally extended this analysis to quantified sentences/discourses like (12) and (25), later systems use complex theoretical machinery to correctly maintain dependencies across quantified structures and use this same complex mechanism for the simpler cases. Furthermore, as detailed in Section 4.2, existing dynamic systems have empirical deficits, as well, such as the inability to directly capture paycheck pronouns.

2 DUAL

DUAL aims to capture the insight from DPL that complex anaphora can be handled entirely analogously to simple anaphora while extending empirical coverage to more complex cases.

To see the intuition behind the new solution, consider the following annotated example of quantificational subordination:

(26) a. Each^{w} woman^{W_1} [bought a^{b} book]$^{W_2}_{W_1}$.
b. Most^{w} of them [read it$_{b}$ immediately]$^{W_3}_{W_2}$.

The annotated nodes above correspond roughly to the following simpler discourses:

(27) a. $W_1 \approx$ There's a^{w} woman.
b. $W_2 \approx$ There's a^{w} woman. She$_{w}$ bought a^{b} book.
c. $W_3 \approx$ There's a^{w} woman. She$_{w}$ bought a^{b} book. She$_{w}$ read it$_{b}$ immediately.

Notice that W_1 just introduces the woman w. W_2 includes the statement from W_1 and also adds the new statement that w bought a book b. This relationship between W_2 and W_1 is reflected in the subscript index W_1 at the end of (26a). Finally, W_3 continues the discourse begun in W_2, adding the statement that w read b immediately.

Next, consider the following paraphrases for the complex sentences in (26):

(28) a. Every value for w consistent with discourse W_1 is also consistent with discourse W_2.
b. Most values for w consistent with discourse W_2 are also consistent with discourse W_3.

These paraphrases only refer to simple discourses, easily handled by DPL. They use a meta-level comparison of the values these simple discourses can give to a variable, w in this case. However, the anaphora is mediated entirely via the original DPL method of dynamic conjunction of clauses.

In order to implement this intuition in logic, DUAL employs two new pieces of machinery. First, *update variables*, represented as uppercase Latin letters with numerical indices, store the update values of clauses. This move parallels a standard technique in computer programming languages: the ability to store sequences of instructions in so-called *procedures*. Once a procedure is defined, it may be called later in the same program to execute its stored instructions. In the same way, denotations stored in update variables can be executed later in a discourse to correctly capture the sorts of quantified sentences shown above. DUAL employs a new colon (':') operator in *update assignments*, which set the value of an update variable, as shown in (29):

(29) a. woman^{W_1} $\rightsquigarrow (W_1 : [w]; woman(w))$
b. [bought a book]$^{W_2}_{W_1}$ $\rightsquigarrow \big(W_2 : W_1; [b]; book(b); bought(w, b)\big)$
c. [read it$_{b}$ immediately]$^{W_3}_{W_2}$ $\rightsquigarrow \big(W_3 : W_2; read\text{-}immediately(w, b)\big)$

Update variables may also serve as clauses in the logic, retrieving and applying their stored update value, as shown in (29b) and (29c).

The second new piece of machinery is the *compound variable term*, which collects all output values for an individual variable inside an update. For instance, $W_2.b$ represents all values for b in the output of update W_2. In the example given in (29), W_2 stores the DPL update $[\![[w]; woman(w); [b]; book(b); bought(w, b)]\!]^{DPL}$, and therefore $W_2.b$ is the set of all books bought by a woman. Similarly, $W_3.b$ is all the books that any woman bought and read immediately. (From here on, a superscript *DPL* will be added to the interpretation function when it represents a simple DPL meaning.)

Compound terms will help us define the restrictor and reference sets for quantificational sentences. For instance, $W_1.w$ (the set of women) is the restrictor set for (26a) and $W_2.w$ (the set of women who bought a book) is the reference set. Given the definitions of EVERY and MOST in (31), the translation of (26) can now be given as in (30).

(30) $(W_1 : [w]; woman(w)); (W_2 : W_1; [b]; book(b); bought(w, b)); EVERY(W_1.w, W_2.w); (W_3 : W_2; read\text{-}immediately(w, b)); MOST(W_2.w, W_3.w)$

(31) a. $\text{EVERY}(A, B) := |A \cap B| = |A|$
b. $\text{MOST}(A, B) := |A \cap B| > |A \setminus B|$

Note that the restrictor and reference sets can also be used as discourse plurals.

Previous dynamic semantic systems have expanded the potential values for variables. Most similar to the approach here is the system introduced by Hardt (1993, 1999). For instance, Hardt (1999) allows variables to store both dynamic properties and also functions from such properties to updates. These new meanings are custom-built for verb-phrase ellipsis and paycheck pronouns, respectively, and Hardt does not explore their use much beyond these applications. DUAL, on the other hand, only allows variables to store simple updates; but it integrates these meanings into every quantified sentence, in a way that has ramifications for anaphora throughout the system. Hardt's system, though, especially his paycheck pronouns, are an important precedent to this work.

Additionally, Kibble (1994) and Geurts (1995, 1999) allow variables to store propositions, in part to explain modal subordination. In both proposals, assignments may be associated with particular worlds; this facilitates binding across two modal statements, when a certain set of worlds is used in both statements. And van Rooij (2005) achieves a similar effect using an accessibility relation over worlds (which are similarly associated with assignments). Again, these are important precedents to the proposal presented in this paper.[3]

Previous systems have employed operators similar to the interpretation of compound variable terms, as well. In particular, the Summation operator Σ of DRT (Kamp & Reyle 1993, p. 310) collects all values for a single variable within a supplied formula. One major difference is that DUAL only defines this operation over an update variable, not an arbitrary formula, as in DRT. This distinction allows us to prevent spurious plurals in the system. For instance, the translation procedure sketched below only allows generalised quantifiers (including definite determiners) to introduce update variables, (correctly) restricting the anaphoric potential. This is a stipulation, but one greatly simplified by restricting summation to update variables. (See Section 3.1 and Appendix A.1 for discussion of this point.) Finally, there is no parallel in DRT to the actual compound variable terms in DUAL.

The rest of this section presents a formal analysis of the semantics of DUAL.

2.1 *Formal system*

The semantics of DUAL operates over the same domains as DPL, repeated as (a)–(d) below, plus two new domains, W for update variables and B for update assignment functions:

3 Thank you to an anonymous reviewer for pointing out this work.

(32) Domains

a. D : individuals in the model M
b. V : variables $a, \ldots, z$
c. $A(=V \rightarrow D)$: individual assignments g, h, k, l
d. $U(=\wp(A \times A))$: updates u
e. W : update variables $A_1, A_2, \ldots, Z_1, Z_2, \ldots$
f. $B(=W \rightarrow U)$: update assignments G, H, K, L

The models $M=\langle D, I\rangle$ for DUAL are slightly different, too, to allow plural terms: D is still a set of individuals, but I is such that $I(P) \subseteq \left[\wp(D)\backslash\{\varnothing\}\right]^n$ for each n-place predicate P. In other words, predicates now take arguments that are sets of one or more individuals, instead of single individuals, in order to accommodate plurals. (Only predicates over individuals are allowed, none over updates.)

The formal semantics of DUAL will be presented as a pair of mutually recursive functions: an *individual interpretation function* $[\![\phi]\!]^G$ returns a standard DPL update, given a formula ϕ and update assignment G; and an *update interpretation function* written '$G{+}\phi$' returns a new update assignment given an input update assignment G and a formula ϕ. In addition, since we have two kinds of terms in DUAL, a *term interpretation function* $[\![\tau]\!]_{g,G}$ is defined to return the denotation of a term τ relative to the assignment functions g and G. These interpretation functions are summarised as follows (along with $[\![\ ]\!]^{DPL}$, which is not a part of DUAL):

(33)

Function	Parameters	Domain	Range
$[\![\phi]\!]^{DPL}$	(none)	DPL formulas ϕ	U
$[\![\tau]\!]_{g,G}$	$g{\in}A, G{\in}B$	DUAL terms τ	$\wp(D)$
$[\![\phi]\!]^G$	$G{\in}B$	DUAL formulas ϕ	U
$G{+}\phi$	$G{\in}B$	DUAL formulas ϕ	B

2.1.1 *Individual interpretation function.* The individual interpretation function is defined almost identically to the DPL interpretation function (cf. (15) in Section 1.1). Note that X_n and Y_m represent arbitrary update variables in the definitions below and decorated versions of τ represent terms.

(34) Individual Interpretation Function

$g[\![P(\tau_1, \ldots, \tau_n)]\!]^G h$	iff	$g{=}h$ & $\langle [\![\tau_1]\!]_{g,G}, \ldots, [\![\tau_n]\!]_{g,G}\rangle \in I(P)$	[predicate formula]
$g[\![\tau_1{=}\tau_2]\!]^G h$	iff	$g{=}h$ & $[\![\tau_1]\!]_{g,G}{=}[\![\tau_2]\!]_{g,G}$	[identity formula]
$g[\![\phi; \psi]\!]^G h$	iff	$g\left([\![\phi]\!]^G \circ [\![\psi]\!]^{G+\phi}\right)h$	[conjunction]
$g[\![[x]]\!]^G h$	iff	$g[x]h$	[individual assignment]
$g[\![{\sim}\phi]\!]^G h$	iff	$g{=}h$ & $\neg\exists k\left(g[\![\phi]\!]^G k\right)$	[negation]
$g[\![X_n]\!]^G h$	iff	$g(G(X_n))h$	[update formula]
$g[\![(X_n : \phi)]\!]^G h$	iff	$g{=}h$	[update assignment]

One obvious difference from DPL is that our new individual interpretation function is parameterized to an update assignment G. This parameter is passed to the new term interpretation function $[\![\ ]\!]_{g,G}$, used above in the definitions of predicate and identity formulas. Otherwise, predicate formulas, identity formulas, individual assignments and

negation are the same as in DPL. Dynamic conjunction is quite similar to DPL, the only difference being that ϕ and ψ are interpreted relative to update assignments: ϕ relative to G and ψ relative to $G{+}\phi$, G as modified by the first clause ϕ using the new update interpretation function '+'.

The main differences to DPL, then, come in the last two clauses, entirely new to DUAL. An update formula simply applies the update stored in the given update variable X_n. (Recall that the parameter G maps such variables to their values.) Finally, an update assignment has no effect on the individual assignment g, as reflected in the trivial last line of (34).

2.1.2 *Term interpretation function.* The version of DPL presented above had only one type of term: single (individual) variables. Therefore, the definition of DPL simply interpreted terms/variables by plugging them into the local assignment g. DUAL instead has two types of terms: simple and complex. Therefore, DUAL terms must be interpreted via the term interpretation function shown in (35):

(35) Term Interpretation Function

$$[\![x]\!]_{g,G} = \{g(x)\} \quad \text{[simple terms]}$$

$$[\![X_n.\tau]\!]_{g,G} = \bigcup \{[\![\tau]\!]_{h,G} : g(G(X_n))h\} \quad \text{[compound terms]}$$

The term interpretation function is relative to both an individual assignment g and an update assignment G. Simple terms are just single individual variables x, and they denote the singleton set containing the value $g(x)$. (For consistency, all terms denote sets of individuals, even singular terms.) Complex terms consist of an update variable X_n and a (simple or complex) term τ. The interpretation of a compound term requires generating all outputs h of $G(X_n)$, the update stored in X_n, given the input g. The compound term then denotes the union of the possible denotations of τ relative to such an h. Notice finally that this recursive definition allows stacked complex terms like '$A_1.B_1.b$'.

By way of example, for any g and $G = \left[W_1 \mapsto [\![[w]; woman(w)]\!]^{DPL}\right]$:

(36)
$$\begin{aligned}
[\![W_1.w]\!]_{g,G} &= \bigcup \{[\![w]\!]_{h,G} : g\big(G(W_1)\big)h\} \\
&= \bigcup \left\{[\![w]\!]_{h,G} : g\Big([\![[w]; woman(w)]\!]^{DPL}\Big)h\right\} \\
&= \bigcup \left\{\{h(w)\} : g\Big([\![[w]; woman(w)]\!]^{DPL}\Big)h\right\} \\
&\qquad \text{[since } h(w) \text{ doesn't depend on } g\text{]} \\
&= \{h(w) : h(w) \in I(woman)\} \\
&= I(woman)
\end{aligned}$$

Last, as mentioned above, unlike DRT, which allows a similar operation to range over arbitrary formulas, DUAL only allows compound terms with update variables before the dot. That is, DUAL does not allow clauses like '([w];$woman(w)$).w', only like '$W_1.w$'. This will allow us to distinguish precisely which items may serve as antecedents to later plural terms. In particular, while full generalised quantifiers like *most women* generate discourse plurals, simple indefinites like *a woman* do not. Compare, for example (37a) to (37b):

(37) a. I spoke with a woman at the reception. #They were gathered in the corner.
b. I spoke with most women at the reception. They were gathered in the corner.

(37a) sounds odd even if there were multiple women at the reception.

2.1.3 *Update interpretation function.* Finally, let's turn to the new update assignment semantics. The function '+' is designed to sweep from left to right of a formula, modifying an update assignment function every time it reaches an update assignment clause.

(38) Update Interpretation Function

a. $G+(X_n : \phi) = \big(G+\phi\big)\Big[X_n/[\![\phi]\!]^G\Big]$
b. $G+(\phi;\psi) = \big(G+\phi\big)+\psi$
c. $G+(\sim\phi) = G+\phi$
d. $G+\phi$ = G for any other well-formed ϕ

The function is rather simple; for most formulas, it simply returns its input G, as shown in (38d). Negation is ignored, as shown in (38c), so that any update variables defined under negation will scope out.[4] Dynamic conjunction is again composition, as shown in (38b). The only tricky clause is (38a), which defines an update assignment. The main effect of a formula '$(X_n : \phi)$' is to store $[\![\phi]\!]^G$, the update value for ϕ relative to G, in the update variable X_n. This effect is achieved using the assignment modification expression '$\Big[X_n/[\![\phi]\!]^G\Big]$'. This operation, defined in (39), returns an output assignment identical to its input except that the output assignment is guaranteed to return the given value for the given variable:

(39) $G[X_n/u] := \lambda Y_m. \begin{cases} u & \text{if } Y_m{=}X_n \\ G(Y_m) & \text{otherwise} \end{cases}$

The reason the right-hand side of (38a) has the expression '$\big(G+\phi\big)$' before this modification, rather than simply 'G', is to allow embedded update assignments. For instance,

$$G+\big(A_1 : [a]; p(a); (B_1 : [b]; q(b))\big)$$

should return an update assignment whose value for A_1 is $[\![[a]; p(a)]\!]^{DPL}$ and whose value for B_1 is $[\![[b]; q(b)]\!]^{DPL}$. Note that the clause '$\big(B_1 : [b]; q(b)\big)$' has no effect on the value for A_1; in particular, using the update formula 'A_1' later will not redefine B_1.

For a more complete example of this interpretation function in action, let's examine how the formula for (26a), repeated in (40), affects an update assignment. For simplicity, assume that the domain of the assignment function includes only W_1 and W_2. The initial values for these variables are represented as question marks, since both will be overwritten and any real value would be completely arbitrary.

(40) $(W_1 : [w]; woman(w)); \big(W_2 : W_1; [b]; book\big(b\big); bought\big(w, b\big)\big)$;

4 Under this definition, the system does not treat negation as a test for update assignments, only for individual assignments. So, although plain individual variables cannot scope out of negation, update variables can. This correlates with how discourse plurals can scope out of a position below negation, as shown in (i):

(i) Not many students^{S_1} are here$^{S_2}_{S_1}$

a. They$_{S_1.s}$ [=students in general] tend to sleep in on Mondays.
b. They$_{S_2.s}$ [=the students here] are enthusiastic, though!

Notice that S_1 (the restrictor) and S_2 (the nuclear scope) can be felicitously referenced in a subsequent sentence even when their antecedents scope below negation.

$$EVERY(W_1.w, W_2.w)$$

The first clause in (40) is interpreted via (38a) as shown in (41a):

$$(41)\quad \begin{bmatrix} W_1 \mapsto ? \\ W_2 \mapsto ? \end{bmatrix} + (W_1 : [w]; woman(w))$$

$$\text{a.}\quad = \left(\begin{bmatrix} W_1 \mapsto ? \\ W_2 \mapsto ? \end{bmatrix} + ([w]; woman(w))\right) \begin{bmatrix} W_1/[\![[w]; woman(w)]\!]^{\begin{bmatrix} W_1 \mapsto ? \\ W_2 \mapsto ? \end{bmatrix}} \\ \end{bmatrix}$$

$$\text{b.}\quad = \left(\begin{bmatrix} W_1 \mapsto ? \\ W_2 \mapsto ? \end{bmatrix}\right) \begin{bmatrix} W_1/[\![[w]; woman(w)]\!]^{\begin{bmatrix} W_1 \mapsto ? \\ W_2 \mapsto ? \end{bmatrix}} \\ \end{bmatrix}$$

$$\text{c.}\quad = \left(\begin{bmatrix} W_1 \mapsto ? \\ W_2 \mapsto ? \end{bmatrix}\right) \left[W_1/[\![[w]; woman(w)]\!]^{DPL}\right]$$

$$\text{d.}\quad = \begin{bmatrix} W_1 \mapsto [\![[w]; woman(w)]\!]^{DPL} \\ W_2 \mapsto ? \end{bmatrix}$$

(41a) simplifies to (41b), because, as defined in (38d), $G + ([w]; woman(w)) = G$ for any G. (41b) simplifies to (41c) because (as defined in (34)) $[\![[w]; woman(w)]\!]^G$ does not depend on the value for G, since the formula being interpreted does not contain any update variables. Finally, the result of the assignment modification is shown in (41d).

Next, as defined in (38b), dynamic conjunction is interpreted via composition. Therefore, the second clause of (40), '$(W_2 : W_1; [b]; book(b); bought(w, b))$' is interpreted using the assignment output by the first, as shown in (42):

$$(42)\quad \begin{bmatrix} W_1 \mapsto [\![[w]; woman(w)]\!]^{DPL} \\ W_2 \mapsto ? \end{bmatrix} + (W_2 : W_1; [b]; book(b); bought(w, b))$$

$$\text{a.}\quad = \left(\begin{bmatrix} W_1 \mapsto [\![[w]; woman(w)]\!]^{DPL} \\ W_2 \mapsto ? \end{bmatrix} + (W_1; [b]; book(b); bought(w, b))\right) \begin{bmatrix} W_2/[\![W_1; [b]; book(b); bought(w, b)]\!]^{\begin{bmatrix} W_1 \mapsto [\![[w]; woman(w)]\!]^{DPL} \\ W_2 \mapsto ? \end{bmatrix}} \\ \end{bmatrix}$$

$$\text{b.}\quad = \left(\begin{bmatrix} W_1 \mapsto [\![[w]; woman(w)]\!]^{DPL} \\ W_2 \mapsto ? \end{bmatrix}\right) \begin{bmatrix} W_2/[\![W_1; [b]; book(b); bought(w, b)]\!]^{\begin{bmatrix} W_1 \mapsto [\![[w]; woman(w)]\!]^{DPL} \\ W_2 \mapsto ? \end{bmatrix}} \\ \end{bmatrix}$$

$$\text{c.}\quad = \left(\begin{bmatrix} W_1 \mapsto [\![[w]; woman(w)]\!]^{DPL} \\ W_2 \mapsto ? \end{bmatrix}\right) \left[W_2/[\![[w]; woman(w); [b]; book(b); bought(w, b)]\!]^{DPL}\right]$$

$$\text{d.}\quad = \begin{bmatrix} W_1 \mapsto [\![[w]; woman(w)]\!]^{DPL} \\ W_2 \mapsto [\![[w]; woman(w); [b]; book(b); bought(w, b)]\!]^{DPL} \end{bmatrix}$$

Just like before, we can simplify (42a) to (42b), since the formula '$(W_1; [b]; book(b); bought(w, b))$' contains no update assignments. Next, (42b) simplifies to (42c), since in our input assignment (call it G), $G(W_1) = [\![[w]; woman(w)]\!]^{DPL}$. The update variable clause of the individual interpretation function ensures $[\![W_1]\!]^G = [\![[w]; woman(w)]\!]^G = [\![[w]; woman(w)]\!]^{DPL}$, allowing the simplification shown. The end result is to store $[\![[w]; woman(w); [b]; book(b); bought(w, b)]\!]^{DPL}$ in the update variable W_2.

The final clause of (40), '$EVERY(W_1.w, W_2.w)$' is vacuous as far as the update interpretation function is concerned, just as the first two clauses are vacuous for the individual interpretation function. This last clause, though, carries the meaning of the quantifier, ensuring that $W_1.w$ (the set of women in the model) is a subset of $W_2.w$ (the women who bought a book in the model).

2.1.4 *DUAL states and truth.* Machine states in DUAL can be conceived of as pairs containing an individual assignment and an update assignment. Thus, a full DUAL update $[\![\phi]\!]^{DUAL}$ for formula ϕ can be defined as follows:

(43) $\langle g, G\rangle [\![\phi]\!]^{DUAL} \langle h, H\rangle$ iff $g[\![\phi]\!]^G h$ and $H{=}G{+}\phi$

This notion is of limited use, though, since the output update assignment H is calculated deterministically.

In fact, truth relative to a DUAL state $\langle g, G\rangle$ can be defined without $[\![\ ]\!]^{DUAL}$, as in (44). This input state might represent an ongoing conversation, or extralinguistic antecedent information. Since this paper is most interested in discourse-internal anaphora, though, another definition of truth is given in (45), which assumes all antecedents are within the discourse itself.

(44) Relative Truth
ϕ is true relative to state $\langle g, G\rangle$ iff $\exists h \left(g[\![\phi]\!]^G h\right)$.

(45) Truth
ϕ is true iff ϕ is true relative to all states.

2.2 *Binding in DUAL*

Groenendijk & Stokhof (1991, Section 3.3) give a detailed account of when variables in DPL are free and when they are bound. A semi-formal description of binding in the version of DPL discussed in Section 1.1 is given in (46):

(46) Binding in DPL

a. An occurrence O of a variable x is bound by an occurrence B of the assignment $[x]$ iff
   i. B precedes O,
   ii. any negation scoping over B also scopes over O and
   iii. no other binder for O occurs between B and O.

b. An occurrence of x is bound simpliciter iff it is bound by some occurrence of $[x]$.

c. An occurrence of x is free iff it is not bound.

Basically, an occurrence of an assignment $[x]$ binds any occurrence of x to its right until another occurrence of $[x]$. The only operator that blocks this binding is negation: if an assignment occurs inside negation, it cannot bind any variables outside that negation (although it can bind those within the same negation).

Binding in DUAL is complicated by its use of update variables, which can essentially bind variables themselves. In addition, since the value of free individual variables within update variable values are set by the local context, known as *dynamic binding* in computer science, the simplest way to determine all bindings involves effectively searching through the whole formula. This section presents a method for such a search, in order to help the reader understand the intricacies of the system.

In DUAL, it will be useful to define a notion of the *proper use* of a term, extending Groenendijk and Stokhof's notion of binding to cover both simple terms (individual variables) and compound terms. In addition to negation, the only new operators that block binding of an individual variable in DUAL are update assignments (the ':' operator) and complex terms (the '.' operator). Nothing blocks binding of an update variable. Our first step is to define binding in DUAL, starting with update variables:

(47) Update variable binding
An occurrence O of X_n is bound by an occurrence B of $(X_n : \phi)$ iff
   a. B precedes O and
   b. no other binder for X_n occurs between B and O.

An update assignment $(X_n : \phi)$ binds any occurrence of the update variable X_n to its right until another occurrence of $(X_n : \phi)$ (although translations of natural language are assumed to have only one binder per update variable). No operators block such binding; no matter how far embedded, the update assignment binds any corresponding update variable to its right. As discussed in footnote 4 above, this even holds for negation.

Next, potential binders for an individual variable x are defined; such a binder may be an occurrence of the individual assignment $[x]$ or an occurrence of an update variable that intuitively 'contains' such an assignment:

(48) Potential binder
B is a potential binder for an occurrence of x iff
   a. B is an occurrence of $[x]$ or
   b. B is an occurrence of X_n bound by an occurrence of $(X_n : \phi)$ such that
      i. ϕ contains a potential binder B' for x such that ...
      ii. ... no negation, update assignment or complex term in ϕ scopes over B'.

Finally, binding for individual variables may be defined parallel to DPL:

(49) Individual variable binding
An occurrence O of x is bound by a potential binder B of x iff
   a. B precedes O,
   b. all negations, update assignments and complex terms scoping over B also scope over O and
   c. no other potential binder for x occurs between B and O.

All binding of an individual variable x begins with an occurrence of the individual assignment $[x]$. Just as in DPL, such an occurrence binds any individual variable to its right until another binder for x occurs (barring intervening negation, update assignments and complex terms). In addition, though, if $[x]$ occurs in the definition of an update variable, occurrences of that update variable can bind x. And since the definition is recursive, another update variable whose definition contains that first update variable can also bind x and so

on. Basically, an update variable X_n may bind x as long as we can trace back a chain through update variable definitions from X_n to an occurrence of $[x]$.

Finally, proper terms and formulas can be defined:

(50) Proper Individual Variables
An occurrence of x is proper iff it is bound.

(51) Proper Update Variables
An occurrence O of X_n is proper iff O is bound by an occurrence of $(X_n : \phi)$ and formula ϕ is proper in the position of O.

(52) Proper Complex Terms
An occurrence of an $X_n.\tau$ is proper iff

a. X_n is proper and
b. τ is a proper complex term or a proper individual variable.

(53) Proper Formulas
An occurrence of a formula ϕ is proper iff any improper terms in ϕ are within the scope of a negation, update assignment or compound term also in ϕ.

It will be assumed from here on that translations of natural language discourses into DUAL must be proper formulas in order to be acceptable.

These definitions allow a variable to be improper when it is first used, say in an update assignment, but proper when used outside this assignment. For instance, consider (54):

(54) $(A_1 : P(x)); [x]; A_1; \ldots$

Here, when x is first introduced, it is unbound and hence improper. Therefore, A_1 effectively contains a free variable. However, by the time that A_1 is used at the top level of the formula, a binder for x has been introduced. Therefore, according to (51), A_1 is used properly, since the free x is effectively bound in this position.

3 APPLICATIONS

The previous section presented the logic of DUAL and discussed its binding properties. This section will discuss how DUAL can successfully translate natural language examples, concentrating mainly on various kinds of anaphora.

3.1 *Translation procedure*

A fully compositional translation algorithm from natural language discourse to DUAL is beyond the scope of this paper. This subsection, though, will sketch out a rough translation procedure to be used in the rest of the paper. As already shown above, indices, representing both individual and update variables, are used to decorate natural language nodes. For instance, consider (55), repeated from (26a) above:

(55) $\text{Each}^{w}\ \text{woman}^{W_1}\ [\text{bought a}^{b}\ \text{book}]^{W_2}_{W_1}$.

The determiners *each* and *a* each carry a superscript individual variable introduced in the formula for the sentence. The restrictor *woman* carries a superscript update variable and the nuclear scope *bought a book* is indexed with both a superscript and a subscript update

variable. The subscript matches the superscript on the restrictor, indicating a relationship between the two.[5]

Phrases with superscript update variables translate to update assignment clauses, as shown in (56), repeated from (29) above:

(56) a. woman^{W_1} $\rightsquigarrow (W_1 : [w]; woman(w))$
b. $[\text{bought a}^{b}\ \text{book}]^{W_2}_{W_1}$ $\rightsquigarrow \big(W_2 : W_1; [b]; book(b); bought(w, b)\big)$

The individual assignment '$[w]$' in (56a) is the translation of the superscript w on the determiner *each*. All generalised quantifiers randomly assign an individual variable inside their restrictors like this. As in DPL, the superscript b on the indefinite determiner *a* also translates to a random assignment, namely '$[b]$'. The update variable clause 'W_1' inside (56b) is the translation of the subscript W_1 on the nuclear scope *bought a book*.

The direct translation of the quantifier *each* applies the set relation *EVERY* defined above to the complex terms combining the update variables W_1 and W_2 (introduced by the quantifier's two arguments) and the individual variable w (indexed on the quantifier). By convention, the letter used for the update variables is the uppercase version of the letter used for the individual variable. The complete translation is the conjunction of these three clauses:

(57) $(W_1 : [w]; woman(w)) ; \big(W_2 : W_1; [b]; book(b); bought(w, b)\big) ;$
$EVERY(W_1.w, W_2.w)$

Crucially, the translation of the quantifier comes last, after the update variables have been defined.

A standard Logical Form, including quantifier raising which leaves traces indexed with variables (just like pronouns), will be assumed. For instance, consider the inverse-scope reading of (55), where there is a single book that every woman bought. This reading would require the LF in (58) and yield the formula in (59), where *a book* has raised to the top of the sentence, leaving the trace t_b:

(58) $\Big[\text{A}^{b}\ \text{book}\Big]\Big[\text{every}^{w}\ \text{woman}^{W_1}\ [\text{bought}\ t_b\ ]^{W_2}_{W_1}\Big].$

(59) $[b]; book(b); (W_1 : [w]; woman(w)); \big(W_2 : W_1; bought(w, b)\big) ;$
$EVERY(W_1.w, W_2.w)$

The formula in (59) defines $W_2.w$ not as the women who bought one or more books, but rather as the women who bought the particular item b as defined in the local context. Since the local context defines b as a book, the last clause of the formula asserts that every woman bought one and the same book.

Notice that since *a* is not a generalised quantifier, it does not store update variables for its 'restrictor' *book* or its 'nuclear scope' comprising the remainder of the sentence; this is reflected in the lack of superscripts on these phrases. For comparison, a similar case with two generalised quantifiers is shown below for the inverse-scope reading of 'Every woman bought most books':

(60) $\Big[\text{Most}^{b}\ \text{books}^{B_1}\Big]\Big[\text{every}^{w}\ \text{woman}^{W_1}[\text{bought}\ t_b]^{W_2}_{W_1}\Big]^{B_2}_{B_1}.$

5 For instance, perhaps a DP like $\Big[_{\text{DP}}\ \text{each}^{w}\ \text{woman}^{W_1}\Big]$ raises, leaving a trace indexed W_1.

(61) $\big(B_1 : [b]; book(b)\big)$;
$\Big(B_2 : B_1; (W_1 : [w]; woman(w)); (W_2 : W_1; bought(w, b)); EVERY(W_1.w, W_2.w)\Big)$;
$MOST\big(B_1.b, B_2.b\big)$

Notice that (61) is similar to the formula from (59), but split into two parts: '$[b]; book(b)$' is now the definition for the update variable B_1 and the remainder of (59) is the definition for B_2. Finally, the clause '$MOST\big(B_1.b, B_2.b\big)$' represents the top-level quantification over books in this (inverse-scope) reading.

One straightforward way to define a compositional translation procedure for quantifiers in general might be to allow the use of λ terms over variables (and functions from variables to DUAL updates etc.) à la Muskens (1996). Such a system (not assumed below) would probably require placing all three indices on the quantifier itself. This quantifier's lexical entry would have to look something like the rough sketch shown in (63). (A similar definition for the indefinite article is shown for comparison.)

(62) $[\![\mathbf{a}^x]\!] = \lambda\phi.\lambda\psi.\big[[x]; \phi(x); \psi(x)\big]$

(63) $[\![\mathbf{each/every}^{x, X_n, Y_m}]\!] = \lambda\phi.\lambda\psi.\Big[(X_n : [x]; \phi(x)); (Y_m : X_n; \psi(x)); EVERY(X_n.x, Y_m.x)\Big]$

The meaning in (63) returns a formula/DUAL update after taking two arguments, ϕ and ψ, each a function taking a variable as input and returning a formula.

One final note about the implicit translation procedure assumed here concerns indices on pronouns. Such indices may be any term in DUAL, including complex terms. Translations of predications involving pronouns use these terms. Some care has been taken, therefore, to only introduce variables—individual and update variables alike—that can form the meaning of a later pronoun. For example, consider this (incorrect) alternative translation for (55):

(64) $\big(W_1 : [w]; woman(w)\big); \big(W_2 : [w]; [b]; book(b); bought(w, b)\big)$;
$EVERY\big(W_1.w, W_1.W_2.w\big)$

This formula has basically the same truth conditions as (57). The difference is that the nuclear scope clause '$\big(W_2 : [b]; book(b); bought(w, b)\big)$' is missing the update variable clause 'W_1' found in (57). This does not affect the meaning of the formula, though, since the second argument of the predicate *EV ERY* in (64) is '$W_1.W_2.w$' instead of '$W_2.w$'. The addition of 'W_1' in this complex term has the same effect, as far as this formula is concerned.

Where the two versions of the formula do differ, though, is in their potential for future discourse plurals. The version in (64) suggests that a future pronoun indexed '$W_2.w$' should denote the set of all book-buyers, both women and men. This is simply not a potential reading, for example for *they* in (65):

(65) Each woman bought a book. They prefer books to magazines.

Here, *they* can easily denote all women or all women who bought a book, but not all people who bought a book. (See Appendix A.1 for more discussion of how empirical facts about anaphoric potential drive proper translation into DUAL.)

3.2 *Quantificational subordination*

As mentioned above, DUAL can handle quantificational subordination, for instance as shown in (66):

(66) $\text{Each}^{w}\ \text{woman}^{W_1}\ \Big[\text{bought a}^{b}\ \text{book}\Big]_{W_1}^{W_2}$.
$\text{Most}^{w}\ [\text{of them}_{W_2.w}]_{W_2}^{W_3}\ [\text{read it}_b\ \text{immediately}]_{W_3}^{W_4}$.

Let us discuss a few wrinkles in the translation of this discourse into DUAL. First, notice that the restrictor for the quantifier *most* is itself anaphoric, bearing a subscript W_2. This represents the subordination relation between *most* and the previous quantifier *each*. Next, let's assume the partitive restrictor translates to '$(W_3 : [w];\ W_2;\ w{\in}W_2.w)$'. However, this means that W_3 is identical to W_2, since the assignment '$[w]$' is redundant to the clause 'W_2' and the clause '$w{\in}W_2.w$' is vacuous in this position. Therefore, for perspicuity, we will simplify the notation of a subordinate quantifier and omit the redundant restrictor clause from our translations of such sentences.[6] The resulting indexing and translation is shown in (67):

(67) $\text{Each}^{w}\ \text{woman}^{W_1}\ \Big[\text{bought a}^{b}\ \text{book}\Big]_{W_1}^{W_2}$.
$\text{Most}^{w}\ [\text{of them}]_{W_2}\ [\text{read it}_b\ \text{immediately}]_{W_2}^{W_3}$. $\rightsquigarrow$
$(W_1 : [w]; \mathit{woman}(w)); (W_2 : W_1; [b]; \mathit{book}(b); \mathit{bought}(w, b));$
$\mathit{EVERY}(W_1.w, W_2.w); (W_3 : W_2; \mathit{read\text{-}immediately}(w, b));$
$\mathit{MOST}(W_2.w, W_3.w)$

This translation yields the correct truth conditions discussed in the introduction to Section 2 above.

Cooper (1979) points out that quantificational subordination pronouns may be multiply embedded. For instance, altering an example of his slightly, consider the following sentences and their LF structures:

(68) a. Every^{p} professor gives each^{s} student of theirs_p a^{r} mid-semester report.
b. Most^{p} of the professors think that all^{s} of the students read it_r.

(69) a. $\text{Every}^{p}\ \text{professor}^{P_1}\left[\begin{array}{l}\text{each}^{s}\ [\text{student of theirs}_p]^{S_1} \\ \quad [t_p\ \text{gives}\ t_s\ \text{a}^{r}\ \text{mid-semester report}]_{S_1}^{S_2}\end{array}\right]_{P_1}^{P_2}$.

b. $\text{Most}^{p}\ [\text{of the professors}]_{P_2}\left[\begin{array}{l}\text{all}^{s}\ [\text{of the students}]_{S_2} \\ \quad [t_p\ \text{think}\ t_s\ \text{read it}_r]_{S_2}^{S_3}\end{array}\right]_{P_2}^{P_3}$.

The LF structures translate to the following formulas:[7]

(70) a. $(P_1 : [p]; \mathit{professor}(p)); \left(\begin{array}{l} P_1; (S_1 : [s]; \mathit{student\text{-}of}(s, p)); \\ P_2 : (S_2 : S_1; [r]; \mathit{report}(r); \mathit{give}(p, s, r)); \\ \quad \mathit{EVERY}(S_1.s, S_2.s)\end{array}\right);$
$\mathit{EVERY}(P_1.p, P_2.p)$

6 It has been noted since at least Jackendoff (1968, 1971) and Perlmutter (1968) that subordinated restrictors are often not pronounced at all; for example, 'Most <> read it immediately.' (This is perhaps related to their low semantic content.) See Gagnon (2013, Section 3.2) for arguments that such missing restrictors are partitive, like the overt ones shown here.

7 This translation again simplifies the anaphoric partitive restrictors *of the professors* and *of the students*.

b. $\Big(P_3 : P_2; (S_3 : S_2; \mathit{think\text{-}read}(p, s, r)); \mathit{EVERY}(S_2.s, S_3.s)\Big); \mathit{MOST}(P_2.p, P_3.p)$

The pronoun *it* (indexed r) in (68) depends on both the professors and the students: r is the report that professor p gave to student s. The updates stored in the first sentence must be reintroduced in the second sentence to give the proper values for p, s, r and the relationships between them. The quantifiers *most* and *all* in the second sentence are therefore anaphoric/subordinate to the quantifiers in the first sentence. Since the clause $\mathit{think\text{-}read}(p, s, r)$ in (70) is in the scope of both the quantifiers *every* and *most*, any individual variable established in either of the updates S_2 and P_2 will be available for use in this clause.

Note that the value of the update variable S_1, namely $[\![[s]; \mathit{student\text{-}of}(s, p)]\!]^{DPL}$, contains a free occurrence of the individual variable p. This means that S_2 (which contains S_1) and S_3 (which contains S_2) will not be proper expressions unless they are preceded by a binder for p. In these formulas, they are bound: in (70a), S_2 (in the clause '$\mathit{EVERY}(S_1.s, S_2.s)$') is bound by P_1 and in (70b), both S_2 and S_3 are bound by P_2. This ability for an update variable to 'contain' a free individual variable is crucial for the analysis of paycheck pronouns presented below.

3.3 *Telescoping*

A phenomenon quite related to quantificational subordination is known as *telescoping*, discussed by Roberts (1987), who credits Barbara Partee with naming the phenomenon. She analyses the following:

(71) [≈ Roberts' (29), credited to Barbara Partee]
   a. Each degree candidate walked to the stage.
   b. She took her diploma from the Dean and returned to her seat.

(72) [Roberts' (30), from Sells (1985)]
   a. Every chess set comes with a spare pawn.
   b. It is taped to the top of the box.

Here, an apparently matrix-level pronoun seems to be bound by a quantifier in a previous sentence. Roberts' solution to this problem is to introduce a new universal quantification into the translations for the second sentences in these cases.[8] She assumes that this quantifier is accommodated directly into the semantics, with a full formula for its restrictor.

We can translate Roberts' solution directly into DUAL by assuming accommodation of a covert distributive operator[9] $\delta^x_{X_n}$, akin to a universal quantifier over a variable x whose restrictor is anaphoric to an update variable X_n. A translation procedure for this operator is indicated in (73):

(73) $\Big[\delta^x_{X_n} \phi^{Y_m}_{X_n}\Big] \rightsquigarrow (Y_m : X_n; \phi); \mathit{EVERY}(X_n.x, Y_m.x)$

We will remain officially agnostic as to whether this operator appears in the syntax or merely in the logic/semantics. However, for perspicuity, we will show the distributive operator in

8 Since Roberts (1987) deals mostly with modal subordination, the universal quantifier is assumed to be a necessity modal. The analysis here uses a universal quantifier over individuals.

9 I appreciate the suggestion by an anonymous reviewer that this be a distributive operator rather than a covert version of a standard quantifier.

the LF syntax for the appropriate sentences. This will make it easier to read the scope of the quantifier relative to other quantifiers in the sentence.

In particular, we can capture the telescoping cases above by assuming a covert distributive operator in the second sentence whose restrictor is anaphoric to the nuclear scope update variable of the first sentence:

(74) a. Each^{c} candidate^{C_1} approached$^{C_2}_{C_1}$.
$\rightsquigarrow (C_1 : [c]; \mathit{candidate}(c)); (C_2 : C_1; \mathit{approached}(c)); \mathit{EVERY}(C_1.c, C_2.c)$

b. $\delta^{c}_{C_2}$ [she$_c$ took her$_c$ diploma]$^{C_3}_{C_2}$.
$\rightsquigarrow (C_3 : C_2; \mathit{took\text{-}diploma}(c)); \mathit{EVERY}(C_2.c, C_3.c)$

(75) a. Every^{c} [chess set]^{C_1} [comes with a^{p} spare pawn]$^{C_2}_{C_1}$.
$\rightsquigarrow (C_1 : [c]; \mathit{chess\text{-}set}(c)); (C_2 : C_1; [p]; \mathit{spare\text{-}pawn}(p); \mathit{comes\text{-}with}(c, p));$
$\mathit{EVERY}(C_1.c, C_2.c)$

b. $\delta^{c}_{C_2}$ [it$_p$ is taped to its$_c$ top]$^{C_3}_{C_2}$.
$\rightsquigarrow (C_3 : C_2; \mathit{taped\text{-}to\text{-}top}(p, c)); \mathit{EVERY}(C_2.c, C_3.c)$

Whether the dependent pronoun refers back to the item quantified over (c in both cases) or a second item, such as the spare pawn p in (75), this analysis captures the most salient readings of these sentences. In fact, the analysis reduces telescoping to a case of quantificational subordination, simply one where the subordinate quantifier is an unpronounced distributive operator. Accommodation of this operator can be viewed as a repair strategy: without the anaphoric restrictor introduced by this operator, the pronouns *she*$_c$ in (74a) and *it*$_p$ and *its*$_c$ in (75) would not be properly bound by the co-indexed items in the previous sentence.

Of course, telescoping is a rather marked construction, and Roberts herself notes many restrictions on its use. Authors since Roberts (Poesio & Zucchi 1992; Wang *et al.* 2003; Keshet 2008) have also made very careful studies of the conditions under which telescoping and related phenomena are available. For the purpose of this paper, we can translate these conditions into restrictions on the accommodation of our covert operator. (For instance, it is most likely used when other antecedents for apparently free pronouns are unavailable or not salient enough in some relevant sense.) In this way, the approach here is no different from any other analysis of telescoping, all of which must assume separately a semantics for telescoping and some proper licencing conditions for the construction.

Finally, there are quite similar cases where a potentially free variable occurs within the semantic value of an update variable (instead of as the index of a pronoun directly). We can solve these cases using the same covert operator. For instance, consider the following:

(76) a. At every conference, most attendees donate a book.
b. They are kept on a shelf in the library.

Under one salient reading of (76b), there is a different shelf set aside for each conference, where all the books donated during that conference are kept. Consider the following annotated version of this sequence, with its DUAL translation:

(77) a. Every^{c} conference^{C_1} $\Big[$most^{a} attendees^{A_1} $\Big[$donate a^{b} book at t_c $\Big]^{A_2}_{A_1}\Big]^{C_2}_{C_1}$.
$\delta^{c}_{C_2}$ [they$_{A_2.b}$ are on a^{s} shelf]$^{C_3}_{C_2}$. $\rightsquigarrow$

b. $\big(C_1 : [c]; \mathit{conf}(c)\big);$
$\left(C_2 : C_1; \big(A_1 : [a]; \mathit{attendee}(a)\big); \left(\begin{array}{r} A_2 : A_1; [b]; \mathit{book}(b); \\ \mathit{donate\text{-}at}(a, b, c) \end{array}\right); \mathit{MOST}(A_1.a, A_2.a)\right);$
$\mathit{EVERY}(C_1.c, C_2.c,);$
$\big(C_3 : C_2; [s]; \mathit{shelf}(s); \mathit{on}(s, A_2.b)\big); \mathit{EVERY}(C_2.c, C_3.c)$

The value of A_2 in (77b) is the DPL update denotation $[\![[a]; \mathit{attendee}(a); [b];\ \mathit{book}(b);\ \mathit{donate\text{-}at}(a, b, c)]\!]^{DPL}$. Notice how the variable c here is free. The expression $A_2.b$ is therefore not proper without a binder for c. To derive the correct reading for (76b), then, this term must be embedded under an operator that correctly binds c, such as C_2 in the penultimate clause of (77b).

Incidentally, one additional reading of (76b) is where all the books from all the conferences are kept on the same shelf. We can capture this reading, without any covert quantifier, via the recursive definition of compound terms. Since we are speaking collectively of all the books donated at all conferences, we can simply refer to them as '$C_2.A_2.b$', the collected set of values for the term '$A_2.b$' in the update C_2. Since '$A_2.b$' in turn is the set of values for 'b' in the update A_2, we get the correct result. This collective reading bolsters by contrast the idea that the covert operator δ is in fact a distributive operator.

3.4 *Paycheck pronouns*

As we have just seen, DUAL allows update variables to effectively contain unbound variables. This feature is crucial for capturing so-called *paycheck* or *neontological* pronouns (see Karttunen 1969b; Jacobson 1977; Cooper 1979; Elbourne 2005), pronouns referring to individuals never mentioned before. This is one case where DUAL is empirically superior to existing complete plural logics, which would require additional machinery to handle such cases.

Since most examples of paycheck pronouns involve definite descriptions, some (quite simplified) translations for singular definite descriptions are given here, where the predicate *singular* is only true of singleton sets:[10]

(78) a. $\left[\text{the}^{x}\phi^{X_1}\right]\ \psi \rightsquigarrow (X_1 : [x]; \phi(x)); \mathit{singular}(X_1.x);\ \psi(X_1.x)$
b. $\left[\text{her}^{x}_{x'}\phi^{X_1}\right]\ \psi \rightsquigarrow \left(X_1 : [x]; \phi(x); \mathit{of}(x, x')\right); \mathit{singular}(X_1.x);\ \psi(X_1.x)$

(Take ϕ and ψ to be functions over terms/variables and '$\mathit{of}(x, x')$' to be a generic relation of possession.) The discourse in (79) is thus given the following translation using these definitions:

(79) Most^{s} students^{S_1} $\Big[$submitted the^{p} $\big[$paper they$_s$ wrote$\big]^{P_1}$ on time$\Big]^{S_2}_{S_1}$.
Some$_s$ [of them]$_{S_1}$ $\Big[$submitted it$_{P_1.p}$ over a week late, though$\Big]^{S_3}_{S_1}$.

(80) $\Big(\mathrm{S}_1 : [s]; \mathit{student}(s)\Big); \left(\begin{array}{l} S_2 : S_1; (P_1 : [p]; \mathit{paper}(p); \mathit{wrote}(s, p)); \\ \quad \mathit{submit\text{-}on\text{-}time}(s, P_1.p) \end{array}\right);$
$\mathit{MOST}(S_1.s, S_2.s);$
$(S_3 : S_1; \mathit{submit\text{-}late}(s, P_1.p)); \mathit{SOME}(S_1.s, S_3.s)$

$P_1.p$ stores the paper that s wrote, and not just for those students s previously quantified over, but crucially for any contextually local value of s. For instance, in the first sentence of (79), $P_1.p$ retrieves the papers that students submitted on time; but this same update can be used in the second sentence to retrieve the papers a second set of students wrote and submitted late. (Note that the quantifier *some* in (79) is anaphoric to S_1, the set of all students, rather than S_2.)

10 These translations are for expository purposes only; they are clearly inadequate. For instance, the *singular*() clause should be a presupposition.

Such cases are called paycheck pronouns after classic examples due to Karttunen (1969a). These examples can be captured in DUAL as well:

(81) The^{w} woman who deposited her$^{p}_{w}$ paycheck^{P_1} in the bank was wiser than the^{w} woman who cashed it$_{P_1.p}$. (Jacobson 2000)

Here $P_1.p$ will be the paycheck of woman w, for the local value of w. In the first half of the sentence, this will be the wise woman who deposited her cheque; in the second half, it will be the spendthrift who cashed it.

As previously mentioned, Hardt (1993, 1999) introduces new machinery to handle paycheck pronouns. In contrast, DUAL can handle this anaphora without any additional machinery. Paycheck pronouns simply use (i) the complex terms necessary for quantifier meanings and discourse plurals and (ii) the fact that DUAL does not restrict free variables inside update meanings.

3.5 *Donkey anaphora, weak and strong*

Recall that DUAL's analysis of anaphora, like DPL's, maintains the basic insight that seemingly complex anaphora can be handled analogously to simple anaphora. This feature helps DUAL capture donkey anaphora, such as Geach's (1962) original example in (82a), given a possible translation in (82b):

(82) a. Any^{m} [man who owns a^{d} donkey]^{M_1} [beats it$_{d}$]$^{M_2}_{M_1}$.

b. $(M_1 : [m]; man(m); [d]; donkey(d); owns(m,d))$;
$(M_2 : M_1; beats(m,d))$; $EVERY(M_1.m, M_2.m)$

The occurrence of d inside the nuclear scope clause '$(M_2 : M_1; beats(m,d))$' is bound by the occurrence of M_1 also inside this clause since M_1 stores an update including the random assignment '$[d]$'. Next, $M_1.m$ here denotes the set containing every man who owns a donkey (no matter how many) and $M_2.m$ denotes the set containing every man who owns a donkey that he beats (no matter how many other donkeys he owns and does not beat). This formula therefore represents the weak reading of the donkey pronoun *it*. However, Geach's example is usually taken to mean that the men in question beat all of their donkeys, if they own more than one. In other words, DUAL seems to suffer from the opposite problem from DPL: it derives the weak reading for donkey pronouns but not the strong reading.

Schubert & Pelletier (1989) point out cases where the weak reading is in fact preferred. For instance, examples like (83) do not seem to imply that the drivers put all their quarters in the parking meter:

(83) Every driver who had a quarter put it in the parking meter.

We could take this distinction to be purely pragmatic. If there were a way to pragmatically strengthen the weak reading in appropriate contexts to derive the strong reading, DUAL would require no additional semantic machinery.

However, if a semantic solution to strong donkey anaphora is desired, DUAL will require one additional bit of machinery. (Although see Appendix A for two alternative solutions to this problem, which do not require any additional machinery; empirical problems and other complications prevent the adoption of either proposal.) Somehow, for each man m owning multiple donkeys, the formula must identify all of m's donkeys. The new expression in (84) constrains a value within an update variable to match the surrounding context:

(84) Restricted Update Variable

$g[\![X_n{}^{x}]\!]^{G}h$ iff $g[\![X_n]\!]^{G}h$ & $g(x)=h(x)$

So, for instance, after the formula in (82b), the term $M_1^m.d$ will return all the donkeys owned by the man m in the local context. Intuitively, this restriction has the same effect as removing the clause '[m]' from the value stored in M_1, making it no longer a binder for m.

One possible way of representing the strong reading of (82a) would thus be

$$\begin{array}{l}(M_1 : [m]; man(m); [d]; donkey(d); owns(m,d)) ; \\ (M_2 : M_1; beats(m, M_1^m.d)) ; EVERY(M_1.m, M_2.m)\end{array}$$

Another way, though, would be to assume, as in telescoping, that there is a covert distributive operator in the translation of a strong donkey sentence:

(85) Any^m [man who owns a^d donkey]^{M_1} $\left[\delta^d_{M_1^m}[\text{beats it}_d]^{D_1}_{M_1^m}\right]^{M_2}_{M_1}$

$\rightsquigarrow (M_1 : [m]; man(m); [d]; donkey(d); owns(m,d));$
$(M_2 : M_1; (D_1 : M_1^m; beats(m,d)) ; EVERY(M_1^m.d, D_1.d)) ;$
$EVERY(M_1.m, M_2.m)$

There are now two quantifications in the sentence: the matrix quantification over m introduces the update variables M_1 and M_2; the embedded quantification over d via the cover distributive operator introduces D_1. As shown above, this embedded distributive operator has a restrictor that is anaphoric to M_1^m, a restricted version of M_1. Now, by convention, restrictor and nuclear scope update variables are named using the uppercase version of the individual variable being quantified over, for example M_1 for m. In previous cases of anaphoric restrictors, the anaphora has always been to another update variable using the same letter; for example M_3 might be anaphoric to M_2 or M_1. This is the first example of a switch from letter to letter. A switch is required, though, since although the embedded quantifier is anaphoric to M_1^m, the individual variable it quantifies over is d, not m. (See Appendix A for more discussion of this point.) Though it is more complex, this second method will be adopted, in order to explain certain sentences involving both strong and weak donkey pronouns (discussed in the next subsection).

Just as in telescoping, there are complex licensing conditions for weak versus strong donkey pronouns (see Kanazawa 1994; Barker 1996; Krifka 1996b; Yoon 1996). One suggestive similarity, though, is that the interpretation of donkey pronouns is sensitive to the local pragmatic context (Gawron *et al.* 1991; Champollion, 2016; Champollion *et al.* 2017). For instance, the pronoun in (86) (due to Gawron *et al.* 1991) can be interpreted as (a) weak or (b) strong depending on whether (a) the speaker simply wants a sample to examine or (b) the speaker wants to eradicate the species entirely.

(86) Anyone who catches a medfly should bring it to me.

In this situation, the weak reading (the default reading in DUAL) works fine for the scientific sample scenario. However, in a total eradication scenario, the weak reading is odd. The strong reading (facilitated by the insertion of the covert distributive operator) could again be considered a repair strategy here, ameliorating the pragmatic conflict between the weak reading and the scenario involved. Example (87) (due to Chierchia 1995) is a similar case:

(87) No man who has an umbrella leaves it home on a day like this.

If we are pragmatically interested in who will stay dry, we must employ the strong reading here, perhaps via a repair mechanism employing a distributive operator.

3.6 *Mixed weak and strong donkey readings*

As van der Does (1992) points out, there can be mixed weak and strong donkey readings within the same sentence. For instance, consider (88):

(88) Every parent who has a minivan and a daughter in the game picked her up in it.

Imagine that some parents have two (or more!) minivans and also two or more daughters in the game. The most salient reading of (88) vis à vis the daughters, is that each parent would pick up all of their daughters. However, this most salient reading does not seem to require that such parents use all of their minivans in doing so. So, it seems that the reading for *her* can be strong while the reading for *it* is weak. Furthermore, Brasoveanu (2008, p. 185) points out that in such a case, the parent could have used a different minivan to pick up different daughters; for instance, if one daughter left at half-time in one minivan, but that minivan would not start when it came time to pick up another daughter, and so another one was used.

DUAL can handle such cases as well, without any added machinery beyond the restricted update variables introduced above:

(89) Every^{p} [parent who has a^{m} minivan and a^{d} daughter in the game]^{P_1}
[$\delta^{d}_{P^{p}_{1}}$ [picked her$_{d}$ up in it$_{m}$]$^{D_1}_{P^{p}_{1}}$]$^{P_2}_{P_1}$.

The crucial portion of the formula for this case is the value for the compound term $D_1.d$ (the set of daughters picked up by a given parent p), which is used in the calculation for the distributive operator. D_1 is anaphoric to the restricted variable P^{p}_{1}. Recall that restricted variables effectively no longer bind their restricted individual variable (p in this case). This is equivalent to removing the clause '[p]' from the value stored in P_1, and therefore the value for $D_1.d$ is (for any g, G):

$$[\![D_1.d]\!]_{g,G} = \left\{ h(d) : g \left(\left[\!\!\left[\begin{array}{l} [d]; \mathit{parent}(p); [m]; \mathit{minivan\text{-}of}(m,p); \\ \mathit{daughter\text{-}of}(d,p); \mathit{picked\text{-}up\text{-}in}(p,d,m) \end{array} \right]\!\!\right]^{DPL} \right) h \right\}$$

As shown, the term represents the set of all daughters that the (contextually local) parent p picked up in some minivan or other. Crucially, the value includes random assignment of the variable m, allowing the minivan to vary from daughter to daughter.

3.7 *Plural indefinites*

The indefinite article a^{x} has been translated so far simply using individual assignment '[x]'. Some refinements to this picture include the introduction of *singular* and *plural* predicates, as well as numbers:[11]

(90) a. $\mathit{singular}(\alpha)$ $:= |\alpha|{=}1$
b. $\mathit{plural}(\alpha)$ $:= |\alpha|{\neq}1$
c. $1(\alpha)$ $:= \mathit{singular}(\alpha)$
d. $2(\alpha)$ $:= |\alpha|{=}2$
e. $\boldsymbol{n}(\alpha)$ $:= |\alpha|{=}\boldsymbol{n}$

11 This definition of plural accords with van der Does (1992) rather than van den Berg (1996), who makes $\mathit{plural}(x)$ tautological. I think van den Berg is correct, but I am presenting a simpler system here since this is not the main focus of the paper.

In the system presented above, though, only *singular* would actually produce a valid translation of an indefinite. For instance, consider:

(91) a. A^{w}/one^{w} woman entered. $\rightsquigarrow [w]; singular(w); woman(w); entered(w)$
b. $Some^{w}$ women entered. $\rightsquigarrow [w]; plural(w); woman(w); entered(w)$
c. $Four^{w}$ women entered. $\rightsquigarrow [w]; 4(w); woman(w); entered(w)$

As defined above, individual variables only ever denote singleton sets, so while *singular* will always be true of an individual variable, *plural* and numbers $\boldsymbol{n}$ will always be false. Only compound terms in DUAL can ever be plural. Thus, for plural indefinites we must move to translations involving update variables.

In order to accommodate such cases, we can define static generalised quantifier meanings for plural indefinites as in (92), and update our translations as in (93):

(92) a. $PLURAL(A, B) := plural(A \cap B)$
b. $ONE(A, B) := 1(A \cap B)$
c. $TWO(A, B) := 2(A \cap B)$
d. $\boldsymbol{N}(A, B) := \boldsymbol{n}(A \cap B)$

(93) a. $Some^{w}\ women^{W_1}\ entered^{W_2}_{W_1}. \rightsquigarrow$
$\big(W_1 : [w]; woman(w)\big); \big(W_2 : W_1; entered(w)\big); PLURAL(W_1.w, W_2.w)$
b. $Four^{w}\ women^{W_1}\ entered^{W_2}_{W_1}. \rightsquigarrow$
$\big(W_1 : [w]; woman(w)\big); \big(W_2 : W_1; entered(w)\big); FOUR(W_1.w, W_2.w)$

These translations will help capture cases due to Gareth Evans, discussed in the next subsection.

3.8 *Sheep anaphora*

Although donkey anaphora is often discussed as a kind of *E-type* anaphora (Geach 1962; Evans 1977, 1980), there is a second kind of E-type anaphora, which I will call *sheep anaphora* after Evans' (1980) example below. (For an excellent introduction to this topic, see also Nouwen (Submitted).) Sheep anaphora involves reference to the maximal set of values for a variable introduced earlier:

(94) a. John owns some sheep, and Harry vaccinates them in the spring.
b. Few senators admire Kennedy, and they are rather junior.

The pronouns *them* and *they* above most readily refer to all John's sheep and all senators who admire Kennedy. DUAL can capture this phenomenon using compound terms. The following is a possible translation for (94b) (assuming a collective reading of *rather junior*):

$$\big(S_1 : [s]; senator(s)\big); \big(S_2 : S_1; [k]; kennedy(k); admire(s, k)\big); FEW(S_1.s, S_2.s); rather\text{-}junior(S_2.s)$$

The compound term $S_2.s$ will denote all the senators who admire Kennedy, as required to derive the correct reading. A similar effect is achieved in (94a) with the generalised

quantifier plural *some*:[12]

(95) a. some^{s} sheep^{S_1} [John^{j} owns t_s]$^{S_2}_{S_1} \rightsquigarrow$
$(S_1 : [s]; sheep(s)); (S_2 : S_1; [j]; john(j); owns(j, s)); PLURAL(S_1.s, S_2.s)$

b. Harry^{h} vaccinates them$_{S_2.s} \rightsquigarrow [h]; harry(h); vaccinates(h, S_2.s)$

In this case, $S_2.s$ denotes all of John's sheep, giving (95b) the correct meaning.

Nouwen (2003) points out that pronouns such as these can even occur inside of quantificational contexts. For instance, in (96), *them* can refer to the entire set of soccer-playing daughters:

(96) Most^{f} [area fathers]^{F_1} [have a^{d} daughter in the youth soccer league]$^{F_2}_{F_1}$.
Each$_f$ [one]$_{F_2}$ [promised he$_f$ would gather them$_{F_2.d}$ for a pre-game pep talk at least once this season]$^{F_3}_{F_2}$.

Nouwen shows how systems due to van den Berg (1996) and Krifka (1996a) cannot handle the pronoun *they* in (95), a point we return to in section 4.2.

3.9 *Proper names and definite descriptions*

There has been some debate in dynamic semantics about the treatment of proper names; the issue is that pronouns referring back to proper names seem to be available no matter how embedded the proper name is (see e.g. Muskens 1996). If we treat these antecedent–pronoun pairs the same way as an indefinite–pronoun pair in DRT or DPL, we incorrectly predict, for instance, that *she$_m$* in (97) denotes a free variable, unconnected to *Mary* in the previous sentence (since *Mary* is embedded under the quantifier *every*):

(97) Everyone [likes Mary^{m}]. She$_m$ is quite charismatic.

This problem has a solution in DUAL, though, since we can easily index the pronoun with a compound term, as shown in (98):

(98) Everyone^{L_1} [likes Mary^{m}]$^{L_2}_{L_1}$. She$_{L_2.m}$ is quite charismatic.

The pronoun *she$_{L_2.m}$* takes singular morphology because m does not vary within L_2 and therefore $L_2.m$ represents a singleton set. This contrasts with cases where the embedded item is an indefinite, such as (94a) above; when an indefinite varies within a context, later compound-term references will be plural.

Similarly, definite descriptions whose value does not vary within the value of an update variable will also induce singular morphology in later pronouns:

(99) Every^{g} $\left[\text{girl in the}^{c}\ \text{class}^{C_1}\right]^{G_1}$ $\left[\text{likes her}^{t}_{g}\ \text{teacher}^{T_1}\right]^{G_2}_{G_1}$.
He$_{G_2.T_1.t}$ makes it$_{C_1.c}$ fun.

12 It is more difficult for DUAL to capture any non-maximal reading of a sentence like (94a). Plurals in DUAL are only available via update variables and update variables are always maximal. One strategy would be to assume a hidden contextual restriction on the first sentence, akin to *John owns some particular sheep that I have in mind.* (This strategy would recognize the marginal status of such readings.) Alternatively, this issue might be an additional reason to add genuine plural individuals to DUAL, as hinted at in footnote 17.

The update variable C_1 will store a single class c, and therefore *it*, indexed $C_1.c$, displays singular morphology. And as long as all the girls have the same teacher, *he* can have singular morphology, too. This latter pronoun requires the recursive index '$G_2.T_1.t$' since the update stored in T_1 contains a free variable, g. The value for t does not depend on g in this case, though, so a singleton set is derived.

3.10 *For future work: intensional contexts*

Fully exploring the interaction between DUAL and intensional/modal contexts is beyond the scope of this paper. However, this subsection briefly sketches how such interaction might work.

First, as mentioned above, Kibble (1994) and Geurts (1995, 1999) extend dynamic semantics to store propositions (sets of possible worlds) in order to capture modal subordination. More recently, Brasoveanu (2007) details a similar method for modal subordination to be treated in a plural dynamic system as an extension of an account for quantificational subordination. DUAL can capture modal subordination in a similar fashion if we allow the individual variables in DUAL to refer to worlds as well as individuals. Basically, each predicate must now take one extra argument (say, its first argument) representing the world in which it is evaluated. Next, update variables such as W_1 can be introduced by quantifiers over possible worlds, such as modals.

For instance, consider a version of Roberts' 1987 classic example, 'A bear might come in; it would eat you first' (where w is a world variable, R is an accessibility relation and @ is the actual world):

(100) a. $\text{might}^{w}\, R^{W_1}\, [\text{a}^{b}\ \text{bear come in}]^{W_2}_{W_1} \rightsquigarrow$
$(W_1 : [w]; @Rw); (W_2 : W_1; [b]; \mathit{bear}(w, b); \mathit{come\text{-}in}(w, b))$;
$\mathit{MIGHT}(W_1.w, W_2.w)$

b. $\text{would}^{w}\ \mathit{pro}_{W_2}\ [\text{it}_{b}\ \text{eat you first}]^{W_3}_{W_2} \rightsquigarrow$
$(W_3 : W_2; \mathit{eat\text{-}you\text{-}first}(w, b))$; $\mathit{WOULD}(W_2.w, W_3.w)$

After the sentence in (100a), W_2 will represent an update involving all the worlds w accessible from @ where a bear enters in w. This forms the restrictor for the sentence in (100b), and therefore the variable b in (100b) will represent the bear that enters in world w.

With an intensional definition of negation such as (101), DUAL could also help explain modal subordination to negation, as illustrated in (102), after Sells (1985):

(101) $\text{not}^{w}\, \phi^{W_1} \rightsquigarrow (W_1 : [w]; \phi(w))$; $\sim(@ \in W_1.w)$

(102) John doesn't own a car. It would be too big for his driveway.

a. $\text{not}^{w}\ [\text{John}^{j}\ \text{own a}^{c}\ \text{car}]^{W_1} \rightsquigarrow$
$(W_1 : [w]; [j]; \mathit{john}(j); [c]; \mathit{car}(w, c); \mathit{owns}(w, j, c)); \sim(@ \in W_1.w)$

b. $\text{would}^{w}\ \mathit{pro}_{W_1}\ [\text{it}_{c}\ \text{be too big}]^{W_2}_{W_1} \rightsquigarrow$
$(W_2 : W_1; W_1; \mathit{too\text{-}big}(w, c)); \mathit{WOULD}(W_1.w, W_2.w)$

After (102a), W_1 is an update involving worlds w where John owns a car. Although (102a) asserts that the real world @ is not amongst these worlds, they are still available for later anaphoric reference, as in (102b), which states that the car c in such worlds would be too big.

4 DYNAMIC PLURAL LOGIC

Dynamic Plural Logic, proposed in van den Berg (1996), is another logic expressly designed to handle plurals and quantificational subordination. This section gives a brief overview of the system, and discusses some of its empirical and theoretical issues.

4.1 *The system*

Formula denotations in Dynamic Plural Logic, as in DPL, are relations on machine states, but these states represent sets of assignments rather than single assignments as in DPL. In other words, a state S in Dynamic Plural Logic is a member of $\wp(A)\times\wp(A)$, where A is the domain of (individual) assignments.

For instance, if Delores and Emily each bought one or more books, the output state illustrated in (104) might hold after the utterance of (103). Importantly, the ⋆'s below represent undefined or missing values, so certain variables can effectively lack a value under certain assignments.

(103) Two$^{w,w'}$ women (each) bought a^{b} book.

(104) $S =$

	w'	w	b
g_1	Delores	Delores	Anna Karenina
g_2	Delores	Delores	Beloved
g_3	Delores	Delores	Catch-22
g_4	Emily	Emily	Beloved
g_5	Sarah	⋆	⋆
g_6	Ivy	⋆	⋆
...			

The three variables w, w', b established in (103) make discourse plurals available for the interpretation of future (plural) pronouns, as shown in (105), potential follow-up sentences. The relevant plural value of each variable is the set of values in that variable's column of the information state: for example, {Delores, Emily} for w in S above.

(105) a. They$_{w'}$ buy more books than men do. [restrictor set anaphora]
b. They$_{w}$ used their$_{w}$ VISA cards. [reference set anaphora]
c. They$_{b}$ are on that table over there.

(105a) is an example of restrictor set anaphora, roughly paraphraseable in this case as 'women buy more books than men', since the primed variable w' on the quantifier represents the restrictor set for van den Berg. The unprimed variable w in (105b) represents the reference set, Delores and Emily in (104). Finally, b in (105c) represents a different sort of discourse plural that arises when a noun phrase is embedded in the scope of quantification. Here, b represents the set of all books bought by a woman.

A translation of (103) into Dynamic Plural Logic is given in (106), to be explained below:

(106) $M_{w'}\left([w'];\mathit{women}(w')\right);$
$M_{w}\left([w];w\sqsubseteq w';\delta_{w}\left([b];\mathit{book}(b);\mathit{bought}(w,b)\right)\right);$
$\mathit{TWO}(w',w)$

In order to make sure that discourse plurals receive the correct interpretation in every output state, Dynamic Plural Logic needs to explicitly enforce maximality. For instance,

after the formula (106) above, we only want states S where the three plurals illustrated in (105) are maximal: $S(w')$ should be all women, $S(w)$ should be all women who bought a book, and $S(b)$ should be all books bought by a woman.[13] To this end, van den Berg employs the maximisation operator M_x, used twice in (106), which ensures the information state includes every relevant value for the variable x being maximised. Notice that unlike DPL and DUAL, where maximality simply falls out of the standard definitions, Dynamic Plural Logic requires an additional operator, with a corresponding additional clause in the definition of ⟦ ⟧, just to ensure maximality during quantification.

The precise relationship between the variables w and w' is established by the *dependency preserving subset* relation '$\sqsubseteq$'.[14] This operator ensures, first, that $S(w) \subseteq S(w')$, and, second, that all dependencies between w' and other variables are 'copied over' to w. This is very important when a dependency is established in the restrictor; such dependencies must carry over to the nuclear scope in order to handle donkey sentences such as *Every$^{w,w'}$ woman who bought a^{b} book read it$_{b}$ immediately*. Once again, unlike DPL and DUAL, where dependency preservation simply falls out of the standard definitions, Dynamic Plural Logic requires an additional operator to ensure dependencies are preserved correctly during quantification.

Finally, in order to introduce dependencies, Dynamic Plural Logic requires one last special operator, the distributive operator δ_x. This operator ensures that the output information state has rows representing all relevant dependencies on x, for each value that x takes in the information state. It is used in the maximisation expression defining the variable w: '$M_w\,([w]; w \sqsubseteq w'; \delta_w\,([b]; book(b)\,;\, bought(w, b)))$'. Here, it ensures that each woman w is matched up in rows of the information state with all and only the books b that w herself bought. The distributive operator also helps Dynamic Plural Logic handle cases of quantificational subordination, as shown in (107):

(107) Each$_w$ read it$_b$ (soon thereafter) $\rightsquigarrow \delta_w\,(read(w, b))$.

4.2 *Issues with Dynamic Plural Logic*

Some empirical problems with Dynamic Plural Logic have been discovered since its proposal. Nouwen (2003) points out that the logic predicts that the discourse plural associated with a variable x should not be available inside the scope of the distributive operator δ_x, contrary to observations. For instance, recall the sentence in (108), first discussed in Section 3.8, and its two potential follow-up sentences:

(108) Most$^{f,f'}$ area fathers have a^{d} daughter in the youth soccer league.

a. Each^{f} one promised he$_f$ would take her$_d$ to this Saturday's game.
b. Each^{f} one promised he$_f$ would gather them$_{d???}$ for a pre-game pep talk at least once this season.

The first sentence in (108) establishes the set of local fathers f', the subset f of f' who have a daughter in the soccer league and the set d of these daughters. In addition, it sets up dependencies between the soccer fathers and their daughters, available within a distribution over either variable (f or d). For instance, (108a) could be captured (ignoring

13 Some of this maximisation is arguably due to scalar implicature in this case, but let us ignore this complication.

14 Brasoveanu's (2007) notation '$\sqsubseteq$' avoids confusion with ordinary subsets '$\subseteq$'.

many complications) as '$\delta_f\left(\mathit{promised\text{-}to\text{-}take}(f,d)\right)$' where the f inside the distribution refers to each father in turn and the d to that father's daughter. In fact, as distribution is defined above, this is the only reading for the variables inside this distribution.

Consider (108b), though; here, once again, we must distribute over the fathers f in order to get the correct reading for he_f. And yet, the pronoun *them*, although clearly within the scope of this distribution, seems to pick up the discourse plural referring to all the soccer-playing daughters. This is not expressible in Dynamic Plural Logic, though, since once we distribute, we lose the ability to retrieve the discourse-level value for the variable d. This is by design, remember, to prevent incorrect readings in cases such as (107): for instance, a reading where Emily reads books that only Delores bought. DUAL gets around this problem via its new compound terms, allowing the discourse-plural reading of a pronoun to appear inside a distributive context.[15]

Brasoveanu (2007, 2008, 2010) points out a number of issues with Dynamic Plural Logic, but let us concentrate here on one concrete empirical deficit he notes, a failure to capture the full range of mixed weak and strong donkey anaphora readings, discussed above in Section 3.6. Brasoveanu's observation is somewhat subtle, but in the end it does show a case of undergeneration for van den Berg's original Dynamic Plural Logic. Consider the example in (109), due to Brasoveanu. Even assuming that the books induce a strong donkey reading and the credit cards a weak one, Brasoveanu points out two potential readings, involving an interaction between these two pronouns:

(109) Every^p person who has a^c credit card and buys a^b book on amazon.com uses it$_c$ to pay for it$_b$.

a. *Single credit card reading*: each shopper uses the same credit card to buy all the books they get from Amazon.
b. *Multiple credit card reading*: each shopper may use a different credit card for different books. For instance, they may use a credit card from work for work-related books and a personal credit card for personal books.

He notes that the multiple credit card reading is indeed available for this sentence. However, as he explains, van den Berg's system predicts that only the single credit card reading should be available.

To see why this is, consider the following (simplified) translation for the restrictor of (109) into Dynamic Plural Logic, adapted from Brasoveanu (2007) (the labelled braces are simply for ease of reference below):

$$(110)\quad \delta_{p'}\left(\begin{array}{l}\overbrace{\mathit{person}(p')}^{a};\overbrace{M_c\left([c];\mathit{singular}(c);\mathit{credit\text{-}card}(c);\mathit{has}(p',c)\right)}^{b};\\ \underbrace{M_b\left([b];\mathit{book}(b);\mathit{buys}(p',b)\right)}_{c}\end{array}\right)$$

(110a) establishes that each p' is a person, (110b) sets up the weak reading for the variable c representing the credit cards and (110c) sets up the strong reading for the variable b

15 An anonymous reviewer suggests another way to solve this problem, while mostly maintaining Dynamic Plural Logic. If pronouns themselves are allowed to take scope, as suggested for example in Barker & Shan (2014), the problematic reading could be derived by scoping this plural outside the distributive operator, while maintaining a trace (or equivalent) in its surface position.

representing the books. Notice that (110b) but not (110c) includes the predicate *singular*, which requires that its argument be a singleton set. This is the crucial difference that allows van den Berg's Dynamic Plural Logic to capture weak donkey pronouns—they must be singular in each state. In other words, each output state must be such that all assignments map c to the same credit card (per person p', since we are distributing over p'). The various output states may map c to a different, single credit card, but not more than one (per person) per state.

This situation may seem at odds with an intuitive notion of maximisation, and yet van den Berg's M operator allows it. As van den Berg defines it, M_c only requires of an output state T (amongst other things) that there be no alternative output state T' whose values for c form a proper superset of the values for c in T, that is, no T' such that $T(c) \subsetneq T'(c)$. Therefore, this operator allows two output states whose values for c do not overlap, such as two singleton set values for c. Singleton sets can be local maxima, which do not preclude other singleton sets also being local maxima. As soon as (within one state) someone uses different credit cards, though, the *singular*(c) clause will fail.

One additional drawback to Dynamic Plural Logic is that it does not handle paycheck pronouns, first discussed in Section 3.4 and exemplified by 111. Recall that the pronoun *it* refers to items never mentioned before: the bonuses given to the minority of irresponsible employees mentioned in (111b). This pronoun apparently takes as its antecedent the phrase *the end-of-year bonus they got* in (111a), despite the fact that this antecedent is only used to refer to the bonuses given to the majority of responsible employees.

(111) a. Most^e employees wisely deposited the$_b$ end-of-year bonus they$_e$ got.
b. Some, however, cashed it$_b$ and went on a binge.

As Nouwen (Submitted) notes, this feature makes paycheck pronouns quite difficult for a standard (plural) dynamic theory to capture. For instance, it is a desired feature of Dynamic Plural Logic that it only admits output states for (111a) that store in b those bonuses given to employees quantified over in (111a) and no more. This way, a later discourse plural indexed b (such as in *They$_b$ all grew in value over the next year*) will refer to all and only the bonuses of those employees who deposited their bonus. However, this beneficial feature bars the pronoun in (111b) from referring to bonuses not originally part of the value for b. Without major additions, Dynamic Plural Logic cannot handle these cases (and neither can systems based on this logic).

One final issue with Dynamic Plural Logic is more conceptual than empirical. As noted in Section 1, the simple DPL (Dynamic *Predicate* Logic) system has the features of maximality, dependency storage and distributivity built-in, as a natural consequence of its core definitions. These features afford DPL its simple, elegant form of quantification, at least for the cases it captures. As noted several times above, such simplicity is lost in Dynamic Plural Logic, which requires new and relatively complex operators to handle maximality and distributivity. These new operators, and the core logic, must also employ quite careful definitions to avoid introducing spurious dependencies or erasing existing ones. In addition, recall that quantification in Groenendijk & Stokhof's DPL is a straightforward extension of the system DPL uses for cross-sentential anaphora, both simply using the output of one update denotation as the input of the next; in Dynamic Plural Logic, this simple extension analysis is lost. DUAL recaptures these conceptual advantages, while extending DPL's empirical coverage to capture those cases handled by Dynamic Plural Logic and beyond, including paycheck pronouns.

5 CONCLUSION

Brasoveanu (2013) traces a step-by-step progression in logics for natural language. To start, he discusses how First Order Logic (FOL) quantifiers introduce variables, but strictly limit their scope. An existential quantification '$\exists x\phi$' in FOL can be represented as a set of assignments g each related to another set of assignments h:

(112) $[\![\exists x\phi]\!] = \{g : \{h : g[x]h \;\&\; h \in [\![\phi]\!]\} \neq \varnothing\}$

The assignments g in (112) are those where there is at least one h such that $g[x]h$ and h makes ϕ true. Notice that in FOL, the set of assignments h is lost after the quantification is over; since the denotations are sets of single assignments, only the 'input' assignments g are saved.

As discussed above, DPL formulas denote sets of pairs of assignments, an input state and an output state (what we have called updates). DPL can therefore store the output assignments of an existential quantification as well as the input assignments:

(113) $[\![[x];\phi]\!] = \{\langle g, h\rangle : \{k : g[x]k \;\&\; \langle k, h\rangle \in [\![\phi]\!]\} \neq \varnothing\}$

This move from single assignments to assignment pairs allows DPL to extend the scope of variables essentially indefinitely. A variable introduced in DPL can scope anywhere to right (except, of course, as limited by negation and additional quantification over the same variable).

Whilst individual variables have extended scope in DPL as compared to FOL, the logic still limits the scope of dependencies introduced under generalised quantifiers in natural language. As seen above, access to such quantificational dependencies is necessary for analyses of quantificational subordination, amongst other phenomena. Brasoveanu shows how van den Berg's (1996) Dynamic Plural Logic can handle subordination, and he argues that Dynamic Plural Logic is the natural successor to DPL, just as DPL was the successor to FOL. Like DPL, Dynamic Plural Logic denotations feature an input state and an output state; but these states are sets of assignments instead of single assignments. In this way, not only variables themselves, but also dependencies amongst variables can scope beyond their local contexts. However, as we have seen above, the maintenance of these dependencies (not to mention maximality for single variables) requires quite a bit of extra machinery. In addition, paycheck pronouns cannot be implemented in standard Dynamic Plural Logic.

This paper has proposed DUAL instead as a successor to DPL. DUAL introduces a different generalisation of DPL denotations; states in DUAL comprise pairs of assignments: one over individuals and one over updates. This generalisation can handle the same empirical phenomena as Dynamic Plural Logic (and more), in a way that better captures the parallels between simple and complex anaphora. Therefore, no complex extra machinery is required to maintain dependency and maximality across sentences. Just as DPL stores variables introduced only temporarily in FOL, allowing later access to these variables, DUAL stores updates introduced only temporarily in DPL, allowing later access to these updates.

A ALTERNATIVE STRONG DONKEY SOLUTIONS

This appendix presents two alternative solutions to allow strong donkey readings in DUAL. Both proposals below require minimal additional machinery, but both potentially run into fatal empirical problems.

A.1 *Indefinites, Singular and Plural*

Section 3.7 above introduced generalised quantifier translations for plural indefinites. These were put to use in Section 3.8 to capture E-type readings of plural pronouns whose antecedents were in previous sentences. Now, if singular indefinites translated to generalised quantifiers, as well, this would provide a potential solution to the strong donkey pronoun problem raised in Section 3.5 above. For instance, consider the following new translation of Geach's classic sentence:

(114)
$$\begin{array}{l}\text{Any}^{m}\Big[\text{man who}\Big[\text{a}^{d}\ \text{donkey}^{D_1}\Big]\,[t_m\ \text{owns}\ t_d]^{D_2}_{D_1}\Big]^{M_1} \rightsquigarrow \\ \quad\Big[\delta^{d}_{D_2}[t_m\ \text{beats it}_d]^{D_3}_{D_2}\Big]^{M_2}_{M_1}. \\ \left(\begin{array}{l}M_1 : [m]; man(m); \big(D_1[d]; donkey(d)\big); \\ \quad \big(D_2 : D_1; owns(m,d)\big); AT\text{-}LEAST\text{-}ONE(D_1.d, D_2.d)\end{array}\right); \\ \big(M_2 : M_1; \big(D_3 : D_2; beats(m,d)\big); EVERY(D_2.d, D_3.d)\big); \\ EVERY(M_1.m, M_2.m)\end{array}$$

As shown, D_2 stores $[\![[d]; donkey(d); owns(m,d)]\!]^{DPL}$, with variable m free. Therefore, $D_2.d$ will denote the set of all donkeys owned by the local value of m. This is precisely what is needed in order to correctly capture the strong reading of (114), using a covert quantifier whose restrictor is entirely anaphoric to D_2.

This analysis is appealing since it uses only independently motivated machinery to solve the problem. However, it immediately runs into serious issues. First, the (static) quantifier meaning of *a* must be 'at least one' instead of 'exactly one' in order to allow the sentence to quantify over multiple-donkey owners. Next, for any m, $D_2.d$ will always be the complete set of donkeys that m owns. This is a positive feature of DUAL, and it correctly captures the 'sheep anaphora' facts introduced in Section 3.8. It means, though, that the generalised quantifier meaning of *a donkey* can never yield the weak reading of the sentence, even if the covert universal is removed:

(115)
$$\begin{array}{l}\left(\begin{array}{l}M_1 : [m]; man(m); \big(D_1 : [d]; donkey(d)\big); \\ \quad \big(D_2 : D_1; owns(m,d)\big); AT\text{-}LEAST\text{-}ONE(D_1.d, D_2.d)\end{array}\right); \\ \big(M_2 : M_1; beats(m, D_2.d)\big); \\ EVERY(M_1.m, M_2.m)\end{array}$$

(115) still asserts that each m beats all of their donkeys. The variable d would not be proper in place of $D_2.d$ since '$[d]$' only appears inside an update assignment.[16,17]

Therefore, even if we allow a generalised quantifier translation for singular indefinites, we still need to keep the standard translation as random assignment, in order to derive weak donkey readings. We also need this standard translation for matrix-level indefinites;

16 Of course, we could add the clause 'D_2' inside the nuclear scope to bind d, but there is no motivation for such a move.

17 It is unclear whether the same problem arises for plural donkey anaphora. For instance, does *Every farmer who owns two donkeys beats them* have a weak reading? What would it mean, if it did? Depending on the answer to this question, DUAL may ultimately be forced to allow individual variables (and not just compound terms) to refer to plurals. I leave such a decision to future research.

generalised quantifier readings of matrix singular indefinites don't always yield the correct reading:

(116) A^{w} woman^{W_1} $\text{entered}^{W_2}_{W_1}$. $\text{She}_{W_2.w}$ was with four other women.

Since five women entered in (116), the term $W_2.w$ will be a set of five women, not suitable for a singular pronoun like *she*. And yet, (116) sounds fine and seems to mean that the speaker has one of the five women in mind to discuss further. Moreover, consider (117):

(117) A^{w} woman^{W_1} $\text{entered}^{W_2}_{W_1}$. $\text{They}_{W_2.w}$ were a group of five.

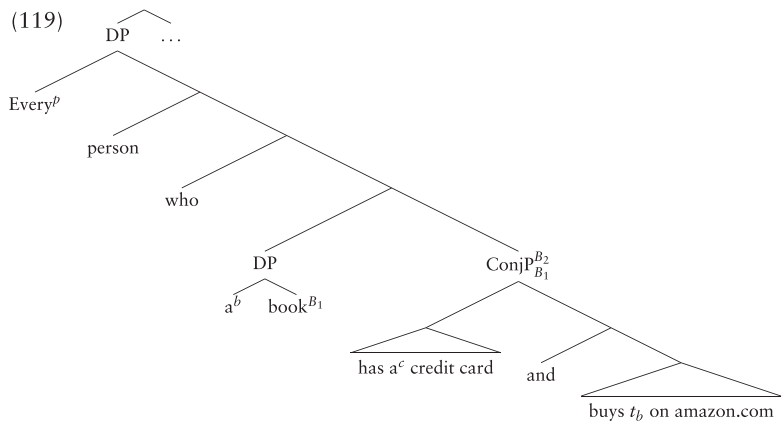
This is predicted to sound good even though it is odd to use a plural pronoun in this situation to mean 'the (multiple) women who entered.'

Finally, the generalised quantifier analysis of strong donkey pronouns cannot handle all cases of mixed weak and strong donkey anaphora. For instance, recall Brasoveanu's sentence in (109), repeated in (118):

(118) Every^{p} person who has a^{c} credit card and buys a^{b} book on amazon.com uses it_{c} to pay for it_{b}.

The crucial reading to consider is the one that is compatible with multiple-book-buyers who used different credit cards to pay for different books. In order to capture such a reading, which is strong relative to the books, we need the generalised quantifier translation of *a book*. Next, in order to allow different credit cards for different books, this quantifier must scope above *a credit card*. This scoping would give the nuclear scope update variable B_2 the denotation $[\![[b]; book(b); [c]; credit\text{-}card(c); buys(p, b)]\!]^{DPL}$, which includes the random assignment of c. However, as pointed out by Brasoveanu (2007), this runs afoul of the coordinate structure constraint (Ross 1967) against raising a quantifier out of one clause of a conjunction:

(119)



So, generalised quantifier translations for singular indefinites don't even solve all the cases of strong donkey anaphora.

A.2 *Delayed Individual Assignments*

Another potential solution comes even closer to solving the strong donkey problem for DUAL, again without any new machinery. The idea is to remove the explicit individual

assignments from restrictor update variables, as shown in (120):[18]

(120) Any $\left[\text{man who owns a}^{d}\text{ donkey}\right]^{M_1}\left[\delta^{D_1}_{M_1}\ [t_m \text{ beats it}_d]^{D_2}_{D_1}\right]^{M_2}_{M_1}$. $\rightsquigarrow$
$\left(M_1 : man(m); [d]; donkey(d); owns(m,d)\right)$;
$\left(M_2 : M_1; (D_1 : M_1); (D_2 : D_1; beats(m,d)); EVERY(D_1.d, D_2.d)\right)$;
$EVERY(M_1.m, M_2.m)$

(Notice that we also no longer annotate the quantifiers with superscript individual variables like m.)

To make up for the missing random assignments, two changes to DUAL are required. First, recall that by convention, quantifications over an individual variable like x introduce uppercase versions of x for their update variables: X_1, X_2, etc. This convention must be codified in the new version of DUAL: each update variable X_n is associated with a *designated individual variable* x, the lowercase version of X. Second, our definition for compound terms $X_n.\tau$ must introduce the random assignment of the designated variable x:

(121) $[\![X_n.\tau]\!]_{g,G} = \bigcup\left\{[\![\tau]\!]_{h,G} : g\left([x]\circ G(X_n)\right)h\right\}$

(The only difference is the addition of '$[x]\circ$'.)

Let us step through how this system works for strong donkey pronouns. First consider the clause in (120) that establishes the value for the update variable M_2. Here, the embedded covert distributive operator defines its restrictor variable D_1 as the single clause 'M_1'. However, crucially, the distinguished variable for D_1 is d and not m. Thus, while $M_1.d$ would represent all donkeys owned by any man (since the distinguished variable m is always randomly assigned in such a compound term), $D_1.d$ instead represents all donkeys owned by the man represented by the current local value for m (since m is not randomly assigned when it is not the distinguished variable). The two values can be compared using the expressions in (122):

(122) a. $[\![M_1.d]\!]_{g,G} =$
$\left\{h(d) : g[\![[m]; man(m); [d]; donkey(d); owns(m,d)]\!]^{DPL}h\right\}$
b. $[\![D_1.d]\!]_{g,G} =$
$\left\{h(d) : g[\![[d]; man(m); [d]; donkey(d); owns(m,d)]\!]^{DPL}h\right\}$

(The doubled '$[d]$' makes no difference since the term d does not appear between its two occurrences.) Similarly, after the nuclear scope variable D_2 is defined, the compound term $D_2.d$ represents all the donkeys that the current local value of m owns and beats, rather than all donkeys that any man owns and beats. The final quantification clause in (120), $EVERY(D_1.d, D_2.d)$, ensures that the collection of donkeys that man m owns is a subset of the collection of donkeys that man m owns and beats. In this fashion, $M_2.m$ will represent all men that own one or more donkeys and beat all of those donkeys, deriving the strong reading.

One place where this analysis might stumble, though, is in conditional variants of donkey sentences. Although the details of such cases are beyond the scope of this paper, the basic

18 Note that the distributive operator is indexed here with two update variables rather than one individual and one update variable.

issue arises in a structure like (123). Any update variable here (such as M_1) that contains 'a^m man' will be unsuitable as the anaphoric antecedent for the covert universal quantifier for the strong donkey pronoun. This is because 'a^m man' includes random assignment of m, which must be avoided to get the correct strong reading.

(123) If $[a^m \text{ man owns } a^d \text{ donkey}]^{M_1}$, $\delta_{M_1}^{D_1}$ $[he_m \text{ beats } it_d]_{D_1}^{D_2}$.

This is not a problem for the analysis presented in the main body of the paper, since we can use the expression 'M_1^m' to effectively remove this random assignment.

One potential solution to this problem would be to quantify over situations instead of individuals but work out a way to guarantee a one-to-one correspondence between situations and the men m. (Such a correspondence would perhaps be needed anyways to solve the proportion problem.) In this case, even if m is randomly assigned, the correct reading could be derived since there would be only one m associated with the local situation. I leave the details of this proposal to future work.

REFERENCES

Barker, C. (1996), 'Presuppositions for proportional quantifiers'. *Natural Language Semantics* **4**: 237–59.

Barker, C. & C.-c. Shan (2014), 'Continuations and natural language'. Vol. **53**, *Oxford Studies in Theoretical Linguistics*. Oxford University Press. Oxford.

Barwise, J. (1987), 'Noun phrases, generalized quantifiers and anaphora'. In P. Gärdenfors (ed.), *Generalized Quantifiers. Studies in Linguistics and Philosophy (formerly Synthese Language Library)*,, vol. **31**. Springer. Dordrecht.

Brasoveanu, A. (2007), *Structured Nominal and Modal Reference*. Ph.D. thesis, Rutgers, The State University of New Jersey, New Brunswick, NJ.

Brasoveanu, A. (2008), 'Donkey pluralities: plural information states versus non-atomic individuals'. *Linguistics and Philosophy* **31**: 129–209.

Brasoveanu, A. (2010), 'Structured anaphora to quantifier domains'. *Information and Computation* **208**: 450–73.

Brasoveanu, A. (2013), 'The grammar of quantification and the fine structure of interpretation contexts'. *Synthese* **190**: 3001–51.

Champollion, L. (2016), 'Homogeneity in donkey sentences'. *Semantics and Linguistic Theory* **26**: 684–704.

Champollion, L., Bumford, D., & R. Henderson (2017), 'Homogeneity in donkey sentences'.

Chierchia, G. (1995), *Dynamics of Meaning: Anaphora, Presupposition, and the Theory of Grammar*, The University Of Chicago Press. Chicago.

Cooper, R. (1979), 'The interpretation of pronouns'. *Syntax and Semantics* **10**: 61–92.

Elbourne, P. (2005), *Situations and Individuals*. The MIT Press. Cambridge.

Evans, G. (1977), 'Pronouns, quantifiers, and relative clauses (i)'. *Canadian Journal of Philosophy* **7**: 467–536.

Evans, G. (1980), 'Pronouns'. *Linguistic Inquiry* **11**: 337–62.

Gagnon, M. R. (2013), *Anaphors and the Missing Link*. Ph.D. thesis, UMD, College Park, MD.

Gawron, J. M., Nerbonne, J., & S. Peters (1991), 'The absorption principle and e-type anaphora'. In J. Barwise, J. M. Gawron, P. Tutiya, and S. Tutiya (eds.), *Situation Theory and its Applications*, vol. **2**. Center for the Study of Language (CSLI). pp. 335–62.

Geach, P. (1962), *Reference and Generality: An Examination of Some Medieval and Modern Theories*. Cornell University Press. Ithaca, NY.

Geurts, B. (1995), *Presupposing*. Ph.D. thesis, University of Stuttgart, Stuttgart.

Geurts, B. (1999), *Presuppositions and Pronouns*, vol. **3** of CRiSPI, Elsevier. Amsterdam.

Groenendijk, J. & M. Stokhof (1991), 'Dynamic predicate logic'. *Linguistics and Philosophy* **14**: 39–100.
Hardt, D. (1993), *Verb Phrase Ellipsis: Form, Meaning, and Processing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.
Hardt, D. (1999), 'Dynamic interpretation of verb phrase ellipsis'. *Linguistics and Philosophy* **22**: 187–221.
Heim, I. (1983), *Meaning, Use and Interpretation of Language*, De Gruyter, Berlin, chapter File change semantics and the familiarity theory of definiteness, pp. 223–48.
Jackendoff, R. S. (1968), 'Quantifiers in english'. *Foundations of Language* pp. 422–42.
Jackendoff, R. S. (1971), 'Gapping and related rules'. *Linguistic Inquiry* **2**: 21–35.
Jacobson, P. (2000), 'Paycheck pronouns, bach-peters sentences, and variable-free semantics'. *Natural Language Semantics* **8**: 77–155.
Jacobson, P. I. (1977), *The Syntax of Crossing Coreference Sentences*. Ph.D. thesis, University of California, Berkeley.
Kamp, H. (1981), 'A theory of truth and semantic representation'. *Formal Semantics* 189–222.
Kamp, H. & U. Reyle (1993), 'From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory'. vol. **42**, Kluwer Academic Dordrecht. The Netherlands.
Kanazawa, M. (1994), 'Weak vs. strong readings of donkey sentences and monotonicity inference in a dynamic setting'. *Linguistics and Philosophy* **17**: 109–158.
Karttunen, L. (1969a), 'Discourse referents'. *Proceedings of the 1969 Conference on Computational linguistics*, *Association for Computational Linguistics*, pp. 1–38.
Karttunen, L. (1969b), 'Pronouns and variables'. *Fifth Regional Meeting of the Chicago Linguistic Society*, pp. 108–15.
Keshet, E. (2008), 'Telescoping and scope economy'. *Proceedings of the Twenty-Sixth West Coast Conference on Formal Linguistics*, vol. **26**, Linguistics Dept., Stanford University.
Kibble, R. (1994), 'Dynamics of epistemic modality and anaphora'. *International Workshop on Computational Semantics*, ITK, Tilburg, pp. 121–130.
Krifka, M. (1996a), 'Parametrized sum individuals for plural anaphora'. *Linguistics and Philosophy* **19**: 555–98.
Krifka, M. (1996b), 'Pragmatic strengthening in plural predications and donkey sentences'. *Proceedings of SALT*, vol. **6**: pp. 136–53.
Muskens, R. (1996), 'Combining Montague semantics and discourse representation'. *Linguistics and Philosophy* **19**: 143–86.
Nouwen, R. (2003), *Plural Pronominal Anaphora in Context*. Ph.D. thesis, University of Utrecht, Utrecht.
Nouwen, R. Submitted, E-type pronouns: congressmen, sheep and paychecks, The Semantics Companion., Wiley. http://ricknouwen.org/rwfn/wp-content/uploads/2014/01/descriptive.pdf.
Perlmutter, D. M. (1968), *On the Article in English*, ERIC/PEGS Clearinghouse for Linguistics, Center for Applied Linguistics.
Poesio, M. & A. Zucchi (1992), 'On Telescoping'. *Proceedings of SALT* **2**: 347–66.
Roberts, C. (1987), *Modal Subordination, Anaphora, and Distributivity*. Ph.D. thesis, UMass Amherst.
Ross, J. R. (1967), *Constraints on Variables in Syntax*. Ph.D. thesis, MIT.
Schubert, L. K. & F. J. Pelletier (1989), 'Generically speaking, or, using discourse representation theory to interpret generics'. *Properties, Types and Meaning*, Springer, pp. 193–268.
Sells, P. (1985), *Restrictive and Non-Restrictive Modification*, Vol. **28**: Center for the Study of Language and Information, Stanford University.
van den Berg, M. H. (1996), *Some Aspects of the Internal Structure of Discourse. The Dynamics of Nominal Anaphora*. Ph.D. thesis, University of Amsterdam, Amsterdam.

van der Does, J. M. (1992), *Applied Quantifier Logics*. Ph.D. thesis, University of Amsterdam, Amsterdam.

van Rooij, R. (2005), 'A modal analysis of presupposition and modal subordination'. *Journal of Semantics* **22**: 281–305.

Wang, L., McCready, E. & N. Asher (2003), 'Information Dependency in Quantificational Subordination'. in K. von Heusinger and K. Turner (eds), *Where Semantics Meets Pragmatics: the Michigan Papers*, Elsevier.

Yoon, Y. (1996), 'Total and partial predicates and the weak and strong interpretations'. *Natural Language Semantics* **4**: 217–36.