# Dynamic Unioning Plural Logic

Ezra Keshet

University of Michigan

**Abstract**

Popular plural logics based on van den Berg (1996) require complex machinery: a ***structured inclusion*** relation to properly maintain dependencies (e.g., '$\sqsubseteq$' in Brasoveanu (2013)), and ***maximization*** and ***distributivity*** operators to correctly analyze quantification, among others. Systems following van den Berg (1996) introduce even further machinery (see Nouwen, 2003; Brasoveanu, 2008). Here instead I propose the new Dynamic Unioning Plural Logic (DUPL) a simpler system that replicates the van den Berg system (and addresses some of its empirical issues) with only one new operator and one new term type.

## 1 Defining Dynamic Unioning Plural Logic

Following van den Berg (1996), I assume an interpretation relation $[\![\ ]\!] :: \langle \mathbf{G}, \mathbf{G}\rangle \to \mathbf{t}$ over pairs of states $\langle G, H\rangle$, each consisting of sets of assignments. $[\![\ ]\!]$ is interpreted relative to a model $\langle D, I\rangle$ comprising a domain of individuals $D :: \mathbf{e}$ and a predicate interpretation function $I :: \mathbf{p_n} \to \{\mathbf{e}\}^{\mathbf{n}}$. The domain contains only singular individuals, but the predicates expect plural individuals (sets of individuals) as their arguments, where singleton sets represent singular arguments. Specifically, $I$ maps each predicate of arity $n$ to an $n$-tuple of plural individuals, those intuitively making the predicate true. The assignments $g :: \mathbf{v} \to \mathbf{e}|\star$ in each state $G :: \mathbf{G}$ map variables to $D \cup \{\star\}$, individuals in the model plus the dummy value $\star$, representing missing values in the assignment. Finally, we will use two varieties of terms (type $\tau$) in predicate literals: plain variables like "$x$" and augmented variables preceded by a plus-sign, like "$+x$." See Figure 1 for a summary of these types and their conventional meta-language notations.

| Type | Meta-language notation | Description |
|---|---|---|
| $\mathbf{t}$ | $\mathbf{T}$, $\mathbf{F}$ | Truth values |
| $\mathbf{e}$ | $A_1 \ldots A_n, \ldots, Z_1 \ldots Z_n$ | Individuals in the model |
| $\star$ | $\star$ | Dummy value |
| $\mathbf{v}$ | $\nu$, $a \ldots z$, $a' \ldots z'$, $\ldots$ | Variables |
| $\tau$ | $\tau$ | Terms: "$\nu$" or "$+\nu$" |
| $\mathbf{g} = \mathbf{v} \to \mathbf{e}\|\star$ | $g, h$ | Assignments |
| $\mathbf{G} = \{\mathbf{g}\}$ | $G, H, K, L$ | Sets of assignments |
| $\mathbf{f}$ | $\phi, \psi$ | Formulas |
| $\mathbf{p_n}$ | $P$ | Predicates of arity $n$ |

Figure 1: Domains in DUPL

Some useful abbreviations are shown in Figure 2. Namely, "$g[x]h$" indicates that assignments $g$ and $h$ differ at most in their value for variable $x$. Relatedly, "$G[x]H$ indicates that $H$ is a superset of $G$ where each assignment in $H$ is related to an assignment in $G$ via $[x]$. Next, "$G_{x,+y}$" (for example) represents that subset of state $G$ where assignments all have defined values for the variables $x$ and $y$ in the subscript (ignoring the plus signs of terms). "$G(x)$" represents a plural individual drawn from all the values in component assignments $g \in G$. This is the only source of plurality, as assignments map variables to singular individuals. Sequences of terms "$(\tau_1, \ldots, \tau_n)$" may be abbreviated as "$\overrightarrow{\tau_n}$." Finally, $\circledast$ is the "all-star" assignment, mapping any variable to $\star$.

| Notation | :: | Type | Expansion |
|---|---|---|---|
| $g[x]h$ | :: | $\mathbf{t}$ | $g\backslash\{\langle x, g(x)\rangle\}{=}h\backslash\{\langle x, h(x)\rangle\}$ |
| $G[x]H$ | :: | $\mathbf{t}$ | $G \subseteq H \subseteq \{h : \exists g{\in}G\,(g[x]h)\}$ |
| $G_{(+)x,(+)y,\ldots}$ | :: | $\mathbf{G}$ | $\{g{\in}G : g(x){\neq}\star \;\&\; g(y){\neq}\star \;\&\; \ldots\}$ |
| $G(x)$ | :: | $\{\mathbf{e}\}$ | $\{g(x) : g{\in}G_x\}$ |
| $\overrightarrow{\tau_n}$ | :: | $\tau^{\mathbf{n}}$ | $(\tau_1, \ldots, \tau_n)$ |
| $\circledast$ | :: | $\mathbf{g}$ | $\lambda x.\star$ |

Figure 2: Abbreviations

The full definition of DUPL, summarized in Figure 3, defines the following kinds of formulas:

- Random assignment "$[x]$" relates an input state $G$ to any superset $H$ of $G$ whose assignments differ from assignments in $G$ only in their values for variable $x$. Thus, random assignment introduces states with new plural values for $x$ and new dependencies between these values and existing values in $G$. This (along with the definitions of conjunction ";" and the union operator "$\cup_x(\ldots)$" below) ensures that states always get larger as formulas progress.
- Predicate literals "$P\,\overrightarrow{\tau_n}$" are tests over the values of their argument terms $\tau_1 \ldots \tau_n$. For plain variable terms $x$, this is the set of values for $x$ just in the substate where all the argument variables are defined (i.e., $\neq\star$). For instance, $P(x, y)$ evaluated in a state $G$ will only examine $G_{x,y}$. This allows testing a predicate literal to ignore values defined outside the current context. When global testing is required, augmented terms $+x$ are used, since their interpretation comprises all values for $x$ throughout the current state.
- Dynamic conjunction is defined, as usual, via relation composition, and negation is a test ensuring that $[\![\;]\!]$ does not relate the input $G$ to any output.
- The ***union*** operator "$\cup_x\phi$" is the only new operator in DUPL. It relates an input $G$ to the union of certain outputs for $G$ in $[\![[x];\phi]\!]$, namely those outputs $K$ that only differ from $G$ in assignments where $x$ is defined. The expression $K\backslash K_x$ returns the set of assignments

$$\begin{aligned}
&G[\![\;[x]\;]\!]H \text{ iff } G[x]H\\
&G[\![\,P\,\overrightarrow{\tau_n}\,]\!]H \text{ iff } G{=}H \;\&\; \left\langle G^{\overrightarrow{\tau_n}}(\tau_1), \ldots, G^{\overrightarrow{\tau_n}}(\tau_n)\right\rangle{\in}I(P)\\
&G[\![\;\phi;\psi\;]\!]H \text{ iff } G([\![\phi]\!]{\circ}[\![\psi]\!])H\\
&G[\![\;\neg\phi\;]\!]H \text{ iff } G{=}H \;\&\; G{\notin}dom([\![\phi]\!])\\
&G[\![\,\cup_x\phi\,]\!]H \text{ iff } H{=}\bigcup\{K : G[\![[x];\phi]\!]K \;\&\; K\backslash K_x{=}G\} \;\&\; H{\neq}\varnothing
\end{aligned}$$

Figure 3: The DUPL System

$$\begin{aligned}
G^{\overrightarrow{\tau_n}}(x) &= G_{\overrightarrow{\tau_n}}(x)\\
G^{\overrightarrow{\tau_n}}(+x) &= G(x)
\end{aligned}$$

Figure 4: Terms

in $K$ where $x{=}\star$ (i.e., $x$ is undefined). This set must be identical to the input $G$. The overall output state $H$ will always be a superset of $G$, since each $K$ will also be a superset of $G$. This definition also effectively requires $x$ to be undefined before $\cup_x$; otherwise, each $K \backslash K_x$ cannot be equal to $G$ (if $G$ had values for $x$ they would be removed by "$\backslash K_x$").

This condition on the states $K$ essentially maintains the input state as "read-only": the output state $H$ may add assignments, but never remove assignments from $G$ or alter members of $G$ without also introducing a value for $x$. In practice, this feature serves to limit the operation of random assignment. For instance, inside a union clause $\cup_x(\ldots[y];\ldots)$, a clause "$[y]$" will add values for $y$ only to those assignments that also have a value for $x$.

Finally, we will call a formula $\phi$ true in DUPL iff $\{\circledast\} \in \mathit{dom}([\![\phi]\!])$, where $\circledast$, again, is the "all-star" assignment that maps every variable to $\star$.

## 2 Using DUPL

### 2.1 Preliminaries

Without "$\cup_x$," and restricting ourselves to singleton values, DUPL roughly replicates Dynamic Predicate Logic (DPL, Groenendijk and Stokhof, 1991), since each state contains a single contentful assignment, as shown in (1):[1]

(1) A woman entered. She sat. $\rightsquigarrow [w]; 1(w); \mathit{woman}(w); \mathit{entered}(w); \mathit{sat}(w)$
$\forall h : \{\circledast\}[\![(1)]\!]\{h\}$ iff $h(w)$ is a woman who entered and sat, and $\forall \nu{\neq}w\,(h(\nu){=}\star)$.

The clause "$[w]$" introduces all states whose assignments differ from $\circledast$ at most in their values for $w$. For instance, $\{\circledast\}$, $\{[w{\to}A_1]\}$, $\{\circledast, [w{\to}A_1]\}$, $\{[w{\to}W_7]\}$, $\{[w{\to}A_1], [w{\to}W_7]\}$, and $\{\circledast, [w{\to}A_1], [w{\to}W_7]\}$ are all possible output states. The predicate "1" is meant to be true of only singleton values, and therefore "$1(w)$" eliminates those states $G$ with non-singleton values for $G(w)$. The remainder of the formula eliminates those states $G$ in which $G(w)$ is not a woman who entered and sat. This still may leave several states, each of whose singular value for $G(w)$ satisfies these requirements.

Such variation among states in DUPL serves roughly the same purpose as in DPL: it models indeterminacy about the value of variables (sometimes explained as listener uncertainty about speaker reference). In order to introduce plurality, though, we will need variation within a DUPL state, among assignments, as shown in (2):

(2) Three women entered. They sat (together). $\rightsquigarrow [w]; 3(w); \mathit{women}(w); \mathit{entered}(w); \mathit{sat}(w)$
$\forall H : \{\circledast\}[\![(2)]\!]H$ iff $H(w)$ is 3 women who entered and sat, and $\forall \nu{\neq}w\,(H(\nu){=}\varnothing)$.

Here, the predicate literal $3(w)$ allows only states $G$ where $|G(w)| = 3$. Assuming the predicates *women*, *entered*, and *sat* all apply to (non-singleton) sets of individuals, the output states for $\{\circledast\}$ generated by (2) will all contain three women who entered and sat together in the model. Depending on the definitions for the predicates, these sets may even overlap: for instance, if five women entered and sat together, the outputs of this formula will contain all size-three subsets of this group of five.

[1] The output states $H$ may also contain $\circledast$, but this will not affect the value for $H(w)$ since $\circledast(w) = \star$.

## 2.2 Introducing Unions

The "$\cup_x$" operator converts such state-external variation into state-internal variation, achieving a sort of maximization. For instance, a formula including "$\cup_w(\ldots)$" can represent a maximal version of plural "some," as shown in (3). The "$\cup_w$" flattens several states exhibiting indeterminacy (i.e., which women are $w$?) into one state exhibiting plurality (i.e., $G(w)$ represents all women who entered).

(3) Some women entered. They sat together. $\rightsquigarrow \cup_w(\mathit{women}(w); \mathit{entered}(w))\,; \mathit{sat}(w)$
$\{\circledast\}[\![(3)]\!]H$ for the $H$ s.t. $\circledast{\in}H$ and $H(w)$ comprises all women who entered, $H(w)$ all sat together, and $\forall\nu{\neq}w\,(H(\nu){=}\varnothing)$.

Note that the formula "$[w]; \mathit{women}(w); \mathit{entered}(w)$" alone always relates its input $G$ to outputs containing $G$. Therefore, the definition of $\cup_w$ requires that the input $\{\circledast\}$ be a subset of the output state in (3), since each for each $K$ being unioned inside $\cup_w$, $K$ will always be a superset of its input. Since each $K$ must therefore contain $\circledast$, the union of all such $K$ will also include $\circledast$ and the output of the union clause will be a superset of its input. In fact, the outputs of a union clause $\cup_x$ will always be supersets of their inputs. This is reflected in Fact 1:

**Fact 1.** *For any $G$ and $H$ such that $G[\![\cup_x(\ldots)]\!]H$, $G \subseteq H$.*

This fact prevents assignments in the input from being lost in the output of "$\cup_x(\ldots)$".

Building on (3), we can introduce a formula for a maximal reading of the indefinite "three women," as shown in (4). The clause "$3(w)$" appears outside the union, and therefore it simply acts as a test on states. Specifically, it will only be true in models where three women total entered, since $w$ has already been maximized to all the women in the model who entered.

(4) Three women entered. They sat together. $\rightsquigarrow \cup_w(\mathit{women}(w); \mathit{entered}(w)); 3(w); \mathit{sat}(w)$

## 2.3 Collective Quantifiers

Generalized quantifiers use the union operator once for their restrictor and once for their nuclear scope. This process allows later reference to a restrictor set and a nuclear scope / reference set. For instance, in outputs $G$ of the formula in (5), $G(s)$ will comprise all the students, and $G(s')$ all the students who gathered on the quad. As shown in (6), such plural values may be referenced by future pronouns.

Fact 1 ensures that the embedded union clause of (5) does not erase any of the values for $s$, guaranteeing that $G(s)$ will still hold all the students by the end of the formula. The subclause "$s'{=}s$" requires that $K(s') \subseteq K(s)$ in each of its output states $K$. Recall that $[s']$ can introduce any size plural value for $s'$ in its output state. Two-variable literals[2] like $s'{=}s$ only check their truth in the substate where both variables are defined, though. Thus, for the output $H$ of (5), the values for $s$ where $s'$ is defined, namely $H_{s,s'}(s)$, must equal the values for $s'$, namely $H_{s,s'}(s')$. This only requires $H(s')$ to be a subset of $H(s)$, though, since there can be other values for $s$, in assignments where $s'$ is undefined.

Finally, $\mathit{MOST}$ represents an appropriate predicate over sets. Notice that the literal $\mathit{MOST}(+s, +s')$ is the first we have seen to use the augmented terms $+x$. This type of term is necessary here, because we want to compare all values for $s$, not just those where $s'$ is also defined. An illustration of states used to calculate this sentence is shown in Figure 5, assuming that $S_1 \ldots S_4$ are the students in the model and $S_1 \ldots S_3$ gathered on the quad. The values for

[2]For simplicity, I will assume that each model provides an appropriate equality predicate.

$s$ and $s'$ do not need to line up in any given assignment, as long as they match up correctly overall.

(5) Most students gathered on the quad.
$\rightsquigarrow \cup_s(students(s);\ \ \cup_{s'}(s'{=}s; gathered(s')))\ ;\ MOST(+s, +s')$

(6) They waved signs. $\rightsquigarrow$ $waved\text{-}signs(s')$

| $s$ | $s'$ |
|---|---|
| $\star$ | $\star$ |
| $S_1$ | $\star$ |
| $S_2$ | $\star$ |
| $S_3$ | $\star$ |
| $S_4$ | $\star$ |
| $S_1$ | $S_2$ |
| $S_2$ | $S_1$ |
| $S_3$ | $S_3$ |

| $s$ | $s'$ |
|---|---|
| $\star$ | $\star$ |
| $S_1$ | $\star$ |
| $S_2$ | $\star$ |
| $S_3$ | $\star$ |
| $S_4$ | $\star$ |
| $S_1$ | $S_1$ |
| $S_2$ | $S_2$ |
| $S_2$ | $S_3$ |

Figure 5: Two possible outputs of (5), ignoring unused variables

One more feature of the nested union operators in (5) is that there are no assignments $g$ in any output state such that $g(s){=}\star$ but $g(s'){\neq}\star$. Such an assignment might be part of an output of the inner union clause, since $\cup_{s'}(\ldots)$ does not remove outputs with defined values for $s'$. However, the outer union clause will not let such assignments pass, since $\cup_s(\ldots)$ will not allow assignments lacking a value for $s$ to add a value for $s'$. In general, the following fact can be deduced:

**Fact 2.** *For any $G$ such that $G_y{=}\varnothing$ and $H$ such that $G[\![\cup_x(\ldots[y];\ldots)]\!]H$, $H_y \backslash H_x = \varnothing$.*

This is a relativized form of opacity: variables $y$ introduced inside unions $\cup_x(\ldots)$ will not be accessible outside of $G_x$ for output states $G$.

## 2.4 Distributive Quantifiers

Distributivity can be achieved in DUPL by ensuring that states within a union clause contain at most one value for a particular variable, i.e., $\cup_x(1(x);\ldots)$. This is illustrated in (7). Internal to the distributive "$\cup_{w'}(1(w');\ldots)$", representing the nuclear scope of (7), "$w'{=}w; entered(w')$" might produce outputs $K_1$ and $K_2$ in Figure 6, assuming only $W_1$ and $W_2$ smiled in the model. Note that $K_1(w')$ and $K_2(w')$ both contain a single woman who smiled.[3] Crucially, though, $K_1(w)$ and $K_2(w)$ both still contain all women in the domain. Therefore, the union of these sets creates a state $H$ such that $H(w)$ is all women (here $W_1 \ldots W_3$) while $H(w')$ is only those who smiled.

(7) Most women smiled.
$\rightsquigarrow \cup_w(1(w); woman(w);\ \ \cup_{w'}(1(w'); w'{=}w; smiled(w')))\ ;\ MOST(+w, +w')$

Donkey anaphora follows easily from this set-up, as shown in (8). For example, Figure 7 shows a component nuclear scope state representing a woman $W_1$ who bought two books $B_1$ and

[3] Technically, distribution would not be necessary on the nuclear scope if the restrictor is already distributed, as in this case. Distribution is shown for consistency with other cases and the final version of the system presented below.

$B_2$, but only read $B_1$, capturing (weak) donkey anaphora. Notice that the other women who bought books are also included in this state (along with their books), but the interpretation of predicate literals like "$\mathit{read}(w', b)$" only considers the substate where all their argument variables are defined. In this state, that substate would be the single assignment represented by the last row of Figure 7.

For strong donkey anaphora, we can simply replace the nuclear scope with the formula "$\cup_{w'}(1(w'); w'{=}w;\ \cup_{b'}(b'{=}b)\,;\ \mathit{read}(w', b'))$" which stores in the variable $b'$ all books read by the woman $w'$ and then tests whether she read all these books.[4] The reason $b'$ will only store values for the current woman $w'$ is that the outer union $\cup_{w'}(1(w'); \ldots)$ will filter out states that alter assignments where $w'{=}\star$. Since there is only one value for $w'$ in each state, the only books stored in $b'$ in each state that passes this filter will be those bought by this particular woman $w'$. The same process could be repeated separately for donkey pronouns other than books, and therefore mixed cases of weak and strong donkey anaphora can also be handled (van der Does, 1992; Brasoveanu, 2008).

Quantificational subordination is the same process as donkey anaphora, just across sentences, as shown in (9), where $\mathit{SOME}$ is also a set relation. Discourse plurals are also simple in DUPL, as shown, e.g., by $w$ and $w'$ in $\mathit{MOST}(w, w')$; $b$ in the same context would be all woman-bought and -read books. For instance a following clause "$\mathit{on\text{-}table}(b)$" could assert that the books the women bought and read are on the table.

(8) Most women who bought a book read it.
$$\rightsquigarrow \cup_w(1(w); \mathit{woman}(w); [b]; \mathit{book}(b); \mathit{bought}(w, b);\ \ \cup_{w'}(1(w'); w'{=}w; \mathit{read}(w', b)))\,; \mathit{MOST}(+w, +w')$$

(9) Some of them loved it.
$$\rightsquigarrow \cup_{w^2}\big(1(w^2); w^2{=}w';\ \ \cup_{w^3}\big(1(w^3); w^3{=}w^2; \mathit{loved}(w^3, b)\big)\big)\,;\ \mathit{SOME}(+w^2, +w^3)$$

| $K_1$ | | $K_2$ | |
|---|---|---|---|
| $w$ | $w'$ | $w$ | $w'$ |
| $\star$ | $\star$ | $\star$ | $\star$ |
| $W_1$ | $\star$ | $W_1$ | $\star$ |
| $W_2$ | $\star$ | $W_2$ | $\star$ |
| $W_3$ | $\star$ | $W_3$ | $\star$ |
| $W_1$ | $W_1$ | $W_2$ | $W_2$ |

| $H$ | |
|---|---|
| $w$ | $w'$ |
| $\star$ | $\star$ |
| $W_1$ | $\star$ |
| $W_2$ | $\star$ |
| $W_3$ | $\star$ |
| $W_1$ | $W_1$ |
| $W_2$ | $W_2$ |

Figure 6: Possible states in (7)

| $w$ | $b$ | $w'$ |
|---|---|---|
| $\star$ | $\star$ | $\star$ |
| $W_1$ | $B_1$ | $\star$ |
| $W_1$ | $B_2$ | $\star$ |
| $W_2$ | $B_3$ | $\star$ |
| $W_3$ | $B_4$ | $\star$ |
| $W_1$ | $B_1$ | $W_1$ |

Figure 7: State in the calculation of (8)

## 3 Discussion

### 3.1 Multiple Maximal Outputs

Referential indeterminacy may extend to seemingly quantified variables, not just those introduced by indefinites, when collective predicates are involved. For instance, (10) may be true

[4]This formula assumes *read* can apply collectively to a set of books; an additional distribution would be required otherwise.

where there are multiple separate groups of students who sat together, each comprising one-third of the students total. This reading is not captured by a straight-forward generalized quantifier translation like the first translation in (10):

(10) One third of the students sat together.
$\rightsquigarrow_1 \cup_s(\mathit{students}(s);\ \cup_{s'}(s'{=}s; \mathit{sat\text{-}together}(s')))\,;\ \frac{1}{3}(+s, +s')$
$\rightsquigarrow_2 \cup_s(\mathit{students}(s))\,;\ [s']; s'{=}s;\ \frac{1}{3}(+s, +s');\ \mathit{sat\text{-}together}(s')$

The "multiple thirds" reading seems to be captured better by an indefinite-like definition for "one-third", as shown in the second translation in (10). In general, then, it seems as though certain determiners are ambiguous between a maximal reading and an indefinite reading. This distinction is not captured in the current system, although the indefinite reading seems to be easier with the more numerically precise determiners. For instance, *they* in (11) cannot refer to a set of most students that sat together that is not the maximal one:

(11) Most students sat together. They discussed politics.

## 3.2 Reference to the Restrictor Set inside the Nuclear Scope

Nouwen (2003) points out cases where within a distributive nuclear scope, a plural pronoun seems to refer to a value established in the restrictor, as shown in (12). Failure to capture such cases is one empirical shortcoming of the original van den Berg (1996) system. DUPL can handle this case without further modification, though, since even in a distributive context, the entire previous state is available for reference. (See, for instance, states $K_1$ and $K_2$ in Figure 6 above.)

If we use plain $x$ variable terms, the only reading available is one where the fathers each give a pep talk only to their own daughters, not all the daughters. This is because plain terms in predicate literals are always evaluated at the substate where all their arguments are defined; so, "$\mathit{gave\text{-}peptalk}(f', d)$" for instance would exclude any value for $d$ other than $f'$'s daughter.

Thus, here is another case where we will use augmented variable terms "$+x$," which are evaluated in the full state, rather than some substate thereof. Using these terms, "$+d$" in the literal "$\mathit{gave\text{-}peptalk}(f', +d)$" in (12) will refer to all the daughters, rather than only the local value of $d$ where $f'$ is defined:

(12) Each father with a daughter at the meet gave them (all) a pep talk.
$\rightsquigarrow \cup_f(\cup_f(1(f); \mathit{father}(f);\ [d]; \mathit{daughter}(d, f); \mathit{at\text{-}meet}(d))\,;$
$\cup_{f'}(1(f'); f'{=}f;\ \mathit{gave\text{-}peptalk}(f', +d)));\ \mathit{ALL}(+f, +f')$

Notice that the translation here assumes a second, embedded union term $\cup_f(1(f); \ldots)$ distributing over the restrictor. This is so that the full collected / unioned set of fathers and daughters from the restrictor is available in the calculation of the nuclear scope. The outer, surrounding union $\cup_f(\ldots)$ merely serves to restrict where the nuclear scope variable $f'$ is defined, as discussed when Fact 2 was introduced. This outer union will have no other effect, though, since the inner union over $f$ prevents its input state from having any defined value for $f$. This is therefore required to be a case where random assignment of $f$ in the outer union does nothing.

## 4 Conclusion and Comparisons

A few small features in the definition of DUPL conspire to allow the behaviors described above. First, the progression of discourse states in DUPL is monotonically increasing, in the sense that every output state (where there is one) is a superset of its input. This means that previous discourse states are always available for reference in later clauses of a formula. For instance, even when distributing over the nuclear scope of a quantifier, while the nuclear scope variable, say $x'$, might only have one value, all previous values for the restrictor variable $x$ will always be available. This allows reference to the restrictor set inside the nuclear scope. This monotonicity is due to random assignment always outputting a superset of its input (and no other clause type interfering with this feature). Although predicate literals still must refer sometimes to a substate of the current state, this is achieved via plain $x$ variable terms, whose values are calculated within such a substate.

Second, the union clauses in DUPL provide the maximization necessary for quantification, but limit changes as follows: the union clauses introduce a new reference variable $x$ and filter out any states that make changes outside the "scope" of $x$ (where the scope is the substate where $x$ is defined). This constraint prevents spurious values for a subordinate variable from projecting beyond the union clause. Instead, any variable $y$ introduced inside the scope of $x$ will only be available in future contexts where $x$ is defined, as in quantificational subordination.

Further operators, such as complex inclusion relations and distributive operators are not necessary, given this initial set-up.

### 4.1 Comparison to previous systems

Random assignment in van den Berg (1996) introduces a new plural value for a variable $x$, but does not introduce any dependencies between the new variable and other variables. His definition also explicitly maintains the dependencies between other variables in the input state. Nouwen (2003) uses a very similar definition for his plural logic. Brasoveanu (2007), following earlier work of van den Berg (van den Berg, 1994), introduces both a value and dependencies for a new variable $x$, but still explicitly maintains the dependencies between other variables.

Random assignment in DUPL is most similar to Brasoveanu's version, introducing any value or dependencies for a new variable. The major difference, as mentioned above, is that DUPL ensures that the output of random assignment is a superset of its input. This guarantees that we preserve the input as read-only, a fact which was crucial for correctly capturing generalized quantifiers and reference to the restrictor set inside the nuclear scope.

Predicate literals in van den Berg's system (and Nouwen's) always take the entire state-wide value for their argument variables. Brasoveanu introduces both a distributive and a state-wide version of predicate literal interpretation, building on the "abstraction" operation of Kamp and Reyle (1993).

DUPL uses a different definition, restricting variable values in a predicate literal to those in the substate where all the predicate's argument variables are defined. In addition, "$+x$" style variable terms allow reference to the complete state. This small change allows reference to local contexts via plain terms and global contexts via augmented terms.

Existing plural logics all help themselves to further operators defined as relations over states, rather than as abbreviations for more basic operations. DUPL is a reaction against this, an attempt to derive as much as possible from a smaller basic set of operations.

The other systems all include a distributive operator. Distribution in van den Berg's system is over a variable $x$, and explicitly applies a given formula to all substates where $x$ has only one value, before unioning / stitching these values back together. Nouwen updates this definition only to allow reference to the restrictor set. Brasoveanu distributes down to the individual assignment level, rather than relativizing the distribution to the value of a particular variable.

Existing systems also include a maximization operator, somewhat akin to $\cup_x(\ldots)$ above. For van den Berg and Brasoveanu, maximizing a formula with respect to a variable $x$ outputs only those states whose values for $x$ is not smaller than any other output's value for $x$. Nouwen's version of maximization ($\varsigma$) is closer to DUPL, since his is a combination distributive and maximization operator.

DUPL, instead, takes advantage of the fact that the union operation inherently stitches together states. Then, distribution can be defined without a new operator, simply by requiring substates within a union clause to contain at most one value for a given variable.

The systems due to van den Berg and Brasoveanu define a special inclusion operator "$\subseteq$" or "$\sqsubseteq$" to ensure the correct relationship between the restrictor set variablex (e.g., $x$) and the nuclear scope set variable (e.g., $x'$). (Nouwen has a different view on projection of the restrictor set.) Clauses like "$w'{=}w$" in DUPL replicate van den Berg's inclusion operator, ensuring that if $w'$ has a non-$\star$ value, this value will match $w$. Since random assignment always carries along the full input state, even if $w'$ does not comprise all values in $w$, all values in $w$ will nevertheless be available for later use. No special inclusion operator is required.

There are other differences. Brasoveanu allows individual assignments to return plural values. Nouwen's system eschews variables and assignments in preference to indexed stacks of values. These choices are largely orthogonal to the points in this paper, though. In addition, the other systems introduce further operators that cannot be defined as abbreviations of multiple simpler operations (for instance, Brasoveanu (2007) has a relativized "atom" operator to capture certain scope effects). Although DUPL aims to avoid the introduction of further such operators, I leave to future work the examination of whether DUPL already captures the empirical cases that drove the creation of each such operator.

## References

Martin H. van den Berg. A direct definition of generalized dynamic quantifiers. In MJB Stokhof and P Dekker, editors, *Proceedings 9th Amsterdam Colloquium*, pages 121–140. ILLC, 1994.

Martin H. van den Berg. *Some aspects of the internal structure of discourse. The dynamics of nominal anaphora*. PhD thesis, University of Amsterdam, 1996.

Adrian Brasoveanu. *Structured nominal and modal reference*. PhD thesis, Rutgers, The State University of New Jersey, 2007.

Adrian Brasoveanu. Donkey pluralities: plural information states versus non-atomic individuals. *Linguistics and philosophy*, 31(2):129–209, 2008.

Adrian Brasoveanu. The grammar of quantification and the fine structure of interpretation contexts. *Synthese*, 190(15):3001–3051, 2013.

Jacob Maarten van der Does. *Applied quantifier logics*. PhD thesis, University of Amsterdam, 1992.

Jeroen Groenendijk and Martin Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14(1):39–100, 1991.

Hans Kamp and Uwe Reyle. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, volume 42. Kluwer Academic Dordrecht,, The Netherlands, 1993. ISBN 0792310276.

Rick Nouwen. *Plural pronominal anaphora in context.* PhD thesis, University of Utrecht, 2003.