Binominal *each* and association with the structure of distributivity

Abstract

Binominal *each*, alongside distributive numerals, has been argued to exhibit an 'association-with-distributivity' effect (Champollion 2015, Kuhn 2015, 2017, see also Henderson 2014). In this paper, I show that analyses along these lines fall short of some empirical generalizations established for binominal *each*, including Counting Quantifier Constraint (Safir and Stowell 1988, Sutton 1993, Szabolcsi 2010) and Extensive Measurement Constraint (Zhang 2013). Instead, I submit that binominal *each* does not associate with distributivity, but with the *internal structure* of distributivity, i.e., the internal, mereological structure of the functional dependency induced by distributive quantification. Concretely, it imposes a monotonicity constraint that the measure function provided by its host should track the part-whole structure of the functional dependency induced by distributivity. Since monotonic measurement typically tracks the part-whole structure of the object being measured (Schwarzschild 2006,Wellwood 2015), this amounts to saying that binominal *each* measures distributivity, with help from the measure function contributed by its host. The proposal is couched in a version of dynamic plural logic that resembles the original Dynamic Plural Logic in van den Berg (1996) but also incorporates more recent innovations such as domain plurality and delayed evaluation, found in its cousin logic Plural Compositional DRT (Brasoveanu 2006, 2008, 2013).

1 Introduction

The need to provide a compositional semantics for free pronouns in natural language, such as the one in (1-a), has inspired a more dynamic way for looking at meaning (Kamp 1981, Heim 1982, 1983, Groenendijk and Stokhof 1991, Dekker 1994, a.o.). For example, in Dynamic Predicate Logic (DPL), Groenendijk and Stokhof (1991) invite us to go beyond the standard semantics of first order logic, which fails to assign an appropriate interpretation for the first order formula in (1-b-ii), intended as a translation of the second clause of (1-a). Their key insight is to treat sentences not as truth conditions, i.e., functions from indices (i.e., possible worlds and/or assignments) to truth values, but as context change potentials, i.e., relations between contexts. In DPL, a context is an assignment function mapping different discourse variables to their respective values.

With help of DPL, both sentences in (1-a) can be treated as a set of pairs of variable assignments, composed via relation composition (see Groenendijk and Stokhof 1991:pp.54). For concreteness, the first sentence is translated as (1-c-i). The existential quantifier resets the value associated with the discourse variable x in an input assignment (i.e., the notation $g[x]h$) and associates it with a man that came in. The new value for x is recorded in all assignments in the output. When the second sentence, translated as (1-c-ii), is interpreted relative to these output assignments, the pronoun *he* can retrieve the value stored in x and add to it a new condition that the value belongs to a set of entities that sat down.

(1) a. A^x man came in. He_x sat down.
b. (i) $\exists x(\textbf{man}\ x \wedge \textbf{came.in}\ x)$
(ii) $\textbf{sat.down}\ x$
c. (i) $\{\langle g,h\rangle \mid g[x]h \wedge h(x) \in \textbf{man} \wedge h(x) \in \textbf{came.in}\}$
(ii) $\{\langle g,h\rangle \mid g = h \wedge h(x) \in \textbf{sat.down}\}$

Making values dynamic provides a way to model pronominal binding, which may happen at a distance and across sentence boundaries. However, Krifka (1996) observes that it is still inadequate to handle so-called 'dependent pronouns', as shown in (2).

(2) $Every^x$ boy saw a^y movie. $They_x$ really enjoyed it_y.

In this sentence, the singular pronoun may refer to a particular movie that every boy saw, or a different movie for each boy. The former reading is straightforwardly predicted by DPL. However, the latter reading requires access to the *functional dependency* between the boys and the movies they saw, a piece of information not directly available in DPL without further extensions.

If a pronoun does nothing fancier than what a discourse variable does, i.e., reading in a context and returning a value, a convenient way to grant it access to dependency is to make dependency part of a context fed to the pronoun[1]. To enable this, van den Berg (1996) devises Dynamic Plural (Predicate) Logic (DPlL) on the basis of DPL to store value dependency. This is done by upgrading the meaning of sentences again, this time from relations between assignments, to relations between sets of assignments. These sets of assignments are capable of keeping track of dependencies in a discourse, as we will see shortly.

To distinguish the more complex context in DPlL from the simpler one in DPL, van den Berg calls his context an '*information state*', or '*info-state*' for short. Figure 1 shows an info-state as a matrix following the convention in the literature. It records the dynamic effect of the first clause in (2) (how this record comes about in a compositional way is given in Section 5). The top row records the discourse variables, x for the boys and y for the movies (and perhaps also other variables introduced before this utterance.) The leftmost column lists all the variable assignments in the info-state. Each cell in the middle represents a *value*, i.e., a value a particular assignment function assigns a variable to. Each column then encodes the collective value a variable receives under a set of assignments, so x stores all the boys and y stores all the movies associated with these boys.

G	x	y	...
g_1	b1	m1	...
g_2	b2	m2	...
g_3	b3	m3	...

Figure 1: An info-state in DPlL

[1]This is not the only option that has been explored in the literature. See Elworthy (1995) and Krifka (1996) for two prominent alternatives.

Because an info-state is a set of assignments, it is capable of recording *dependency* using assignments. For example, in this info-state, we see that the values stored in y changes depending on the values stored in x. A formal notion of dependence can be found in (40) in Section 5. For now, it suffices to have an intuitive sense of it. With dependency being part of a dynamic contribution and hence part of a context, a pronoun like *it* in (2) can now refer to a different movie for a different boy–we just need to make reference to the dependency. It is also worth pointing out that semantic interpretation in DPlL is non-deterministic, so there may be alternative info-states encoding different values and dependencies.

The original DPlL has been enriched and modified in subsequent studies, especially Nouwen (2003), and Brasoveanu (2006, 2008, 2013). However, the main architecture, namely, that sets of assignments (or stacks in Nouwen 2003) can be used to encode dependency, remains. This architecture has been capitalized on to account for many interesting phenomena, some of them having to do with dependency in pronominal anaphora (e.g., Nouwen 2003, Brasoveanu 2008), while others pertaining to non-proniminal dependency-thirsty expressions like dependent indefinites (Brasoveanu and Farkas 2011), distributive numerals (Henderson 2014, Champollion 2015, Kuhn 2017), reciprocals (Dotlačil 2010) and adjectives like *same* and *different* (Dotlačil 2010, Brasoveanu 2011, Kuhn 2017), and bare cardinal partitives (DeVries 2016).

Building on the architecture of DPlL, the present paper makes three contributions. First, it recognizes that distributivity not only contributes values and dependency recordable in an info-state, the very info-state also encodes the **internal, mereological structure** of the distributivity dependency[2]. Second, it demonstrates that binominal *each* makes reference to such a mereological structure of distributivity. Third, it offers a version of DPlL that allows us to take the full advantage of van den Berg's innovation to arrive at an explanatory account of binominal *each*. These contributions are expanded further in the remainder of this section.

It is quite easy to see that in the info-state in Figure 1, there is a dependence relation between x and y. So, we can talk about how the values stored in y change in relation to the values stored in x and vice versa. For example, y has a value $\mathsf{m2}$ when x has a value $\mathsf{b2}$, and x has a value $\mathsf{b3}$ when y has a value $\mathsf{m3}$. However, there is also information about the internal, mereological structure of the distributivity dependency. This is best seen when we turn the info-state in Figure 1 into a semi-lattice, as shown in Figure 2. The elements in this semi-lattice are non-empty subsets of the set of assignments in G, referred to as '*sub-states*' in this paper. The lines connecting the sub-states represent the proper subset relation. Since these sub-states contain discourse information, we can also think of the proper subset relation as an ordering relation on informativity—a bigger sub-state contains more discourse information than a smaller sub-state.

With this mereological structure in place, we gain a new kind of perspective on information stored in info-states. With help of info-states, studies such as van den Berg (1996), Nouwen (2003), Brasoveanu (2008, 2011), Brasoveanu and Farkas (2011), Henderson (2014) and Kuhn (2017) can speak of how the *values* stored in a variable, say y, change depending on the *values* stored in another variable, say x, accounting for a wide range of dependency-thirsty expressions. With mereological structure added as another dimension of info-states, we can now speak of the

[2] Landman 2000 and Champollion 2010, 2017 are two other studies that make explicit the mereological structure of distributive predication, both with help of event semantics. While the present study is couched in dynamic semantics instead of event semantics, it can still be seen as extending the mereological approach to distributivity. More concretely, it provides evidence that there are natural language expressions that piggyback on the mereological structure of distributivity, in addition to the ones identified in Landman (2000) and Champollion (2017).

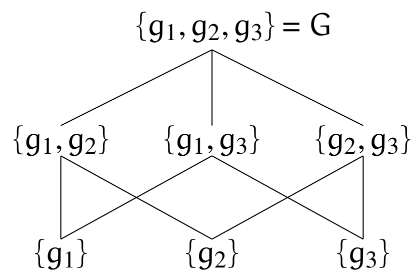


Figure 2: Mereological structure of a distributivity dependency

algebraic properties of the information stored in an info-state. For example, do we get more stuff by going from a smaller sub-state to a bigger sub-state? For another example, if the values associated with a particular variable are transformed into another algebraic structure, such as by means of measurement, how does that structure stand in relation to the mereological structure of the info-state?

It is one thing to recognize that there is a mereological structure to the dependency encodable in an info-state, it is another to show that this property matters for natural language. In this paper, I identify binominal *each* as an expression that capitalizes on the mereological structure of distributivity dependency.[3] An example involving binominal *each* is given in (3).

(3) The boys read one book each.
  a. Distributivity inference: Each boy read a book.
  b. Variation inference: More than one book was read by the boys.

Previous studies have noted that the presence of binominal *each* correlates with two inferences: a distributivity inference and a variation inference. The *distributivity inference* indicates that the predicate containing *each* holds true of every atomic subpart of the plural individual that typically serves as the subject, as shown in the inferences in (3-a); the *variation inference* requires the witness for the cardinal indefinite *one book* to exhibit variation in association with distributivity. Based on these two inferences, recent studies such as Champollion (2015) and Kuhn (2017) classify binominal *each* as a distributive numeral marker, following Henderson (2014)'s seminal work on the latter. Briefly speaking, an indefinite marked by binominal *each*, like distributive numerals, signal distributive quantification and variation of witnesses. Distributivity quantification creates an info-state with dependency information. The variation component piggybacks on the dependency, requiring a particular variable to store a plurality of values in the dependency. This line of research makes binominal *each* a case of 'association with distributivity' (see also Brasoveanu 2011, Bumford and Barker 2013; Bumford 2015).

However, accounts along the lines of 'association-with-distributivity' fall short of explaining a few important generalizations about binominal *each*, including : (i) its widely observed affinity to counting quantifiers beyond numeral expressions (e.g., Safir and Stowell 1988, Sutton 1993),

[3] It is plausible that distributive numerals should also be treated along the same lines as binominal *each* proposed in this study, but I have not collected sufficient cross-linguistic data to verify this. Other distributivity markers that form a paradigm with binominal *each* include adnominal *jeweils* in German and *ge* in Mandarin. See Zimmermann (2002) for a detailed discussion of the selectional properties of the former and Li and Law (2016) for the latter. I reserve a more detailed investigation of these two distributivity markers for another occasion.

and (ii) a less well-known observation that it requires its host to contribute an extensive measure function (Zhang 2013). Instead, I argue that binominal *each* does not just piggyback on the *dependency* in an info-state, they make active use of the *mereological structure of the dependency*. More precisely, they require that the measurement of the values in a particular variable stand in a **monotonic** relation to the mereological structure of the distributivity dependency.[4] Since this monotonicity condition makes reference to distributivity and the internal structure of its dependency, I call it **d-monotonicity** to distinguish it from the closely related, yet different, notion of monotonicity found in the studies of Schwarzschild (2006), Rett (2014) and Wellwood (2015).

To facilitate the implementation of d-monotonicity, I devise a version of dynamic plural logic intermediate between DPlL and PCDRT. Like DPlL and unlike PCDRT, it inhibits dependency arising from random assignment, and checks lexical relations in a collective manner. However, like PCDRT and unlike DPlL, it allows Linkean plural individuals to be in the range of assignment functions. This logic is also extended with tools for measurement beyond cardinality, which are absent in DPlL and PCDRT. The resulting logic is a dynamic plural logic with domain pluralities and various sorts of measure functions. I call this logic DPlLM to show its descendancy from DPlL.

The rest of this paper is organized as follows: Section 2 introduces PCDRT-theoretic analyses of distributive numerals (extended to binominal *each*), demonstrating how the variation inference is modeled as an association-with-distributivity effect. Readers who already have backgrounds in PCDRT, especially those who are familiar with Henderson (2014), Champollion (2015) or Kuhn (2017) can skim or skip this part. Section 3 discusses the challenges faced by the PCDRT-theoretic accounts, showing that it is not enough to just consider the dependency introduced by distributivity. Section 4 sketches an account for binominal *each* that makes crucial use of the structure of a distributivity dependency and measurement to arrive at a monotonicity constraint called 'd-monotonicity'. D-monotonicity answers the challenges raised in Section 3 but is not implemented in a compositional manner. Section 5 lays out DPlLM, with help of which d-monotonicity is implemented compositionally. Section 6 examines how d-monotonicity interacts with negation and downward monotone quantifiers, showing that formulating d-monotonicity in a dynamic semantics like DPlLM makes correct predictions regarding the interactions. Section 7 compares the present proposals with previous studies on binominal *each*. Section 8 concludes.

2 Background: Distributive numerals in the PCDRT tradition

A framework in which recent studies on distributive numerals and binominal *each*, such as Henderson (2014), Champollion (2015) and Kuhn (2017), are couched is Plural Compositional Discourse Representation Theory (PCDRT). This framework, devised by Brasoveanu (2006, 2008, 2013), is a cousin of Dynamic Plural (Predicate) Logic (DPlL, van den Berg 1996, also Nouwen 2003). Like DPlL, PCDRT is a dynamic logic using sets of assignments (i..e., info-states) as contexts to model dependency information in discourse. Henderson (2014) extended PCDRT to account for distributive numerals in Kaqchikel. The framework is later extended to account for binominal *each* in Champollion (2015), with few modifications, and Kuhn (2017), with more substantial differences. The primary concern of the present paper is to defend an alternative formulation of the variation

[4] Monotonicity here is understood as a non-decreasing and non-constant mapping, instead of a strictly increasing mapping, as found in Schwarzschild 2006 and Wellwood 2015.

inference that does not just make use of the distributivity dependency, but also its internal structure, to shed light on Counting Quantifier Constraint and Extensive Measurement Constraint. For this reason, I use Henderson (2014)'s framework for distributive numerals as a representative and will not go into the differences among his study and other studies in the PCDRT tradition.

There are two ways for dependency to enter an info-state in PCDRT, via random assignment or via distributivity (with help of random assignment). In PCDRT, random assignment, as defined in (4), is responsible for variable introduction, just like random assignment in DPL and DPlL.

Definition Random assignment (PCDRT)

(4) a. $G[\![[x]]\!]H := \forall g \in G\ \exists h \in H$ such that $g[x]h$ and $\forall h \in H\ \exists g \in G$ such that $g[x]h$
b. $g[x]h := \forall v \neq x. g(v) = h(v)$

For example, let info-state G as depicted in Figure 3 be the input info-state and let G' as depicted in Figure 4 be the output info-state.

G	...	y	...
g_1	...	a	...
g_2	...	b	...
g_3	...	c	...

Figure 3: An info-state in PCDRT

G'	...	y	x	...
g'_1	...	a	d	...
g'_2	...	b	e	...
g'_3	...	c	f	...

Figure 4: Dependency in PCDRT

We know that $G[x]G'$ holds because each assignment in G has a G'-descendent with which it at most differs on the values assigned to x (i.e., the values assigned to y remain intact), each assignment in G' has a G-ascendent with which it also at most differs on the values assigned to x. Importantly, since PCDRT does not restrict the way random assignment assigns values to a new variable, random assignment is free to introduce dependency (or not) into an info-state. This is a new innovation not found in DPlL, which explicitly inhibits dependency introduced by random assignment (see van den Berg 1996:101–102). In Section 3.1, I discuss the repercussion of this design feature on the analysis of binominal *each* in the PCDRT tradition, ultimately arguing for a dependency-free formulation of random assignment.

In order for distributivity to introduce dependency into an info-state, PCDRT crucially differs from the more well-known DPL in the way it treats distributive quantification. In DPL, distributivity as a kind of universal quantification is internally dynamic but externally static, transmitting (dynamic) binding information only from its restriction to its scope, but not recording the binding information once the quantification is evaluated. By contrast, PCDRT follows DPlL in treating distributivity as fully dynamic. To see this, let us zoom into the distributivity operator δ, defined in (5). Basically, the δ operator splits an input info-state into singleton sets of assignments, then updates each set with ϕ and finally collects all the updated sets as the output info-state. If ϕ has any dynamic effect, it is recorded in H and can be used later.

Definition Distributivity (PCDRT, see Henderson 2014)

(5) $G[\![\delta(\phi)]\!]H = \mathbb{T}$ iff there is a partial function f from assignments to sets of assignments s.t.

a. $G = \textbf{domain}\ f$ and $H = \bigcup \textbf{range}\ f$

b. $\forall g \in G.\ \{g\}[\![\phi]\!](f\ g) = \mathbb{T}$

With this much background on PCDRT, we can now translate a sentence with binominal *each* such as (6) in light of Henderson (2014)'s analysis of distributive numerals. The noun phrase with binominal *each* is analyzed as an existential quantifier with a post-suppositional test marked by the overline, as in (7). The existential quantifier is dynamic, in the sense that it introduces a discourse variable x for storing dog-values. The post-suppositional test, according to Henderson (2014), is a test evaluated 'after all the at-issue content is evaluated.' As we shall see shortly, the post-suppositional test eventually escapes the scope of a distributivity operator when it is evaluated.

(6) The students hugged one dog each.

(7) $\textbf{one dog each}^x := \lambda P.[x] \wedge \textbf{one}\ x \wedge \textbf{dog}\ x \wedge P\ x \wedge \overline{x > 1}$

Both the existential quantifier part and the post-supposition part have a cardinality test, albeit defined differently. The cardinality test associated with the existential quantifier is known as domain-level cardinality and is defined in (8), while the cardinality test associated with the post-supposition is an evaluation-level cardinality and is defined in (9).

Definition Domain-level cardinality (PCDRT)

(8) $G[\![\textbf{one}\ x]\!]H = \mathbb{T}$ iff $G = H$ and $\forall h \in H.\ |\{x' \mid x' \leqslant h\ x \wedge \textbf{atom}\ x'\}| = 1$

Definition Evaluation-level cardinality (PCDRT)

(9) $G[\![x > 1]\!]H = \mathbb{T}$ iff $G = H$ and $|\{h\ x \mid h \in H\}| > 1$

The domain-level cardinality test is evaluated by checking the cardinality of the set of atomic parts of an individual assigned by each h. The evaluation-level cardinality test works by gathering all values of a variable under a set of assignments and checking the cardinality of the resulting set. It is responsible for deriving the variation inference. In simple words, domain-level cardinality is evaluated *locally*, while evaluation-level cardinality is evaluated *globally*. Accordingly, (7) identifies a set of atomic dogs satisfying some property P and the post-supposed evaluation-level cardinality test globally checks that the resulting set of dogs has more than one member.

At first glance, (7) looks almost headed for a contradiction. What sort of dogs count as one and more than one at the same time? This is where the ingenuity of Henderson's account comes in. Once a distributive numeral is embedded inside a bigger structure with a distributivity operator, distributivity sets a barrier between the two cardinality tests, interpreting the domain-level cardinality test inside the scope of distributivity and deferring the evaluation-level cardinality test until after distributivity is fully evaluated.

To see this concretely, consider a fuller representation of (6), in which the distributive numeral takes narrow scope relative to the (covert) distributivity operator, as in (10) (there is a variety of ways to compositionally derive the truth condition. See Henderson (2014) and Kuhn (2017) for two examples.)[5]. In this structure, the post-suppositional test falls inside the scope of distributivity. However, due to the definition of post-suppositions as delayed evaluations, (10) has the same interpretation as (11), in which the post-suppositional test is evaluated outside the scope of distributivity. This way, a contradiction is avoided: of course we can talk about each student hugging a dog and more than one dog being hugged by all the students.

(10) $\mathbf{max}^{y}(\mathbf{student}\ y) \wedge \delta_{y}([x] \wedge \mathbf{one}\ x \wedge \mathbf{dog}\ x \wedge \mathbf{hug}\ x\ y \wedge \overline{x > 1})$

(11) $\mathbf{max}^{y}(\mathbf{student}\ y) \wedge \delta_{y}([x] \wedge \mathbf{one}\ x \wedge \mathbf{dog}\ x \wedge \mathbf{hug}\ x\ y) \wedge x > 1$

In the final representation (11), maximization by $\mathbf{max}^{y}$ delivers the biggest set of students, whose members are then distributivity checked for atomicity and hugging of dogs. The results are then stored in an output info-state, as shown in Figure 5.

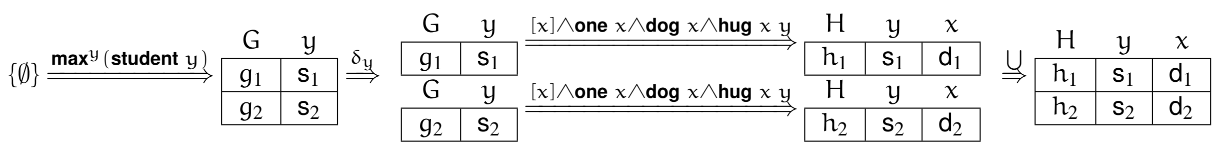


Figure 5: Interpreting distributivity in (6)

The post-supposition tests each output info-state, requiring that it has at least two distinct values stored in x. This can only be satisfied if at least a student hugged a dog distinct from the dog hugged by another student. All those info-states that pass the post-suppositional test are kept and those that fail the test are tossed. A sentence is interpreted 'true' as long as there is at least one info-state in the output. Evaluation-level plurality is an ingenious way to capture the variation inference and has been followed by subsequent studies on binominal *each* such as Champollion (2015) and Kuhn (2017).

It is worth pointing out that although an evaluation-level cardinality test does not seem to make reference to the functional dependency introduced by distributivity, it nonetheless does. This is most obvious from the way an evaluation-level cardinality is computed. It is computed by first taking the set of values assigned to a variable by different assignments and then finding out the cardinality of the set. Since these assignments are drawn after a distributivity update, these assignments together store the functional dependency introduced by distributivity. By stating that the values associated with a variable is evaluation-level plural, it requires the functional dependency to store at least two values for that variable, giving rise to a case of association with distributivity.

In short, the main innovation in the PCDRT-theoretic analyses is that binominal *each* have a two-part contribution. Inside the scope of distributivity, it contributes a normal existential quantifier with a cardinality condition. Outside the scope of distributivity, it contributes an evaluation-level plurality requirement that associates with the functional dependency of distributivity. The

[5] Inside the scope of distributivity, y is required to be singular, i.e., $\mathbf{one}\ y$. It serves two purposes: (i) ensuring that we are distributing down to atoms, instead of some bigger covers, and (ii) ensuring that we discard mixed readings, i.e., each student hugged a dog and some of them also hugged another dog together. The latter is possible because y is only required to store all the relevant students collectively, in whichever way one sees fit. There is no guarantee that it maps each student to an assignment.

different loci of evaluation is made possible by use of post-supposition. The rest of the present paper focuses on the empirical and theoretical challenges faced by the PCDRT approach to binominal *each* and what opportunities these challenges bring. I refer the reader to Brasoveanu (2006, 2008, 2013) for more details about PCDRT and to Henderson (2012, 2014) for the particular version of PCDRT adopted to analyze distributive numerals. As for how post-suppositions are defined in PCDRT, please refer to Brasoveanu (2013) and Henderson (2014). If the reader is interested in how post-suppositions can be translated into a more standard scope taking account, an example is given in Section 5.3, following the spirit of Charlow (to appear). An alternative scope-taking strategy is explored by Kuhn (2017).

3 Empirical challenges

3.1 Association with distributivity

Although the PCDRT approach offers an intuitive way to capture the variation inference of binominal *each*, it does not address a fundamental issue about binominal *each*, namely, a sentence with binominal *each* always expresses distributivity. This is the reason why distributive numeral markers like binominal *each* has long been argued to be a distributivity operator (Choe 1987, Safir and Stowell 1988, Link 1998; Zimmermann 2002; a.o.)

As discussed in Section 2, random assignment in PCDRT is designed not only to contribute a variable, but also to introduce dependency (albeit non-deterministically). This can be seen with the definition of random assignment in (4). What this random assignment does is that when introducing a variable, each assignment function in the set of assignment functions is free to assign a different value to the variable. To see the relevance of this, let's consider the cumulative reading of the sentence in (12) and its translation into PCDRT in (13).

(12) Three students read five books.

(13) $[x] \wedge \textbf{boy}\ x \wedge |x| = 3 \wedge [y] \wedge \textbf{book}\ y \wedge |y| = 5 \wedge \textbf{read}\ y\ x$

A formula of this form may relate many pairs of input and output info-states. A possible one is given in Figure 6, in which different values stored in x correspond to different values stored in y. Hence, there are a plurality of values stored in y in the output info-state.

$\{\emptyset\} \xrightarrow{[x]\wedge\textbf{boy}\ x\wedge|x|=3}$

G	x
g_1	$s_1 \oplus s_2$
g_2	$s_2 \oplus s_3$

$\xrightarrow{[y]\wedge\textbf{book}\ y\wedge\textbf{five}\ y\wedge\textbf{read}\ y\ x}$

H	x	y
g_1	$s_1 \oplus s_2$	$b_1 \oplus b_2 \oplus b_3$
g_2	$s_2 \oplus s_3$	$b_2 \oplus b_4 \oplus b_6$

Figure 6: Dependency in a cumulative reading

As a consequence, if binominal *each* is attached to *three books* in (12), as in (14), the PCDRT approach wrongly predicts that the sentence with the cumulative reading is acceptable.

(14) Three students read five books each.

After all, the post-supposition $y > 1$ contributed by binominal *each* is satisfied in the info-state H shown in Figure 6, as there is more than one value stored in y in the global info-state. There are two values, to be concrete. However, (14) only receives a distributive interpretation, i.e., each of the three students read five books.[6]

Kuhn (2017) notices this problem, and suggests building distributivity into the meaning of binominal *each*. This strategy resolves the empirical issue of over-generation, but instead raises the question why the cumulative version of binominal *each* has never been reported. In other words, given that cumulative readings also have dependency, why don't we see a lexical item that forces the cumulative reading and ensures evaluation-level plurality? In languages with distributive numerals, when a numeral phrase receives a special morphology or marker requiring it to have evaluation-level plurality, it seems that it only ever has a distributive reading, and never a cumulative reading, as pointed out by Kuhn (2017).

In short, if we start with PCDRT, random assignment in this framework can always generate dependency and hence helps satisfy the evaluation-level plurality requirement of binominal *each*, making it difficult to model the close relationship between the distributivity component and the variation component. If we take this relationship seriously and make no revisions to the key insights of Henderson (2014), the minimal change we need to bring about is a framework that inhibits dependency arising from random assignment. In fact, such a framework already exists. van den Berg (1996) discusses in detail the pros and cons for dependency-introducing random assignment and dependency-free random assignment, and opts for the latter. Therefore, in DPlL, random assignment is incapable of introducing dependency into an info-state. Distributivity then becomes the only way to give rise to dependency, and hence explains why the variation component is only compatible with distributivity. The present paper develops a logical framework, detailed in Section 5, that is a natural descent of DPlL in regard to its restrictive introduction of dependency.

3.2 Counting Quantifier Constraint

Evaluation-level plurality offers an intuitive way to capture the variation inference of binominal *each*. However, it does not capture an important observation that has puzzled linguists for years—the so-called Counting Quantifier Constraint. The background of this constraint is that the distribution of binominal *each* is very restricted (Safir and Stowell 1988; Zimmermann 2002; Stowell 2013). Sutton (1993) concludes that only counting quantifiers, i.e., noun phrases with (modified)

[6] Henderson (2014) is aware of this problem, and hence requires variables to have only one value at the evaluation-level (i.e., $\varkappa = 1$) by default. This is an interesting strategy to remove undesirable dependencies introduced into discourse by the powerful random assignment in PCDRT. The random assignment proposed in Section 5.1 of this paper, which is based on van den Berg (1996), can be seen as a more 'automatic' way of getting rid of undesirable dependencies—they are not generated in the first place.

On a related note. Evaluation-level cardinality serves two functions in Henderson (2014): to force dependence (when set to $\varkappa > 1$) and forbid dependence (when set to $\varkappa = 1$). This is closely related to the distributivity operator in van den Berg (1996), with which dependence between variables is allowed and without which it is disallowed. A decision on which strategy is better is a delicate one. However, distributivity is often thought of as a grammatical device that does not necessarily require morphological reflexes while evaluation-level cardinality often calls for morphological support. Therefore, the more morphological reflexes we see of evaluation-level cardinality constraints, the more support we have for using them as the default strategy for marking (in)dependence. On the other hand, if we can show that morphological reflexes for evaluation-level cardinality (like $\varkappa > 1$) can be analyzed as doing something more than just signaling (in)dependence, then we have an argument that the (in)dependence may come from a more basic part of grammar. In this paper, I demonstrate that at least in the analysis of binominal *each*, there is no need to use evaluation-level cardinality, which takes away part of the shine of using this special cardinality to manage the (non-)generation of dependency.

numerals and vague quantity words like *many*, *a few* or *several*, can host binominal *each* (see also Szabolcsi 2010). All other noun phrases are rejected. The contrast is illustrated in (15) and (16).

(15) The boys saw $\left\{\begin{array}{c}\text{two}\\ \text{at least two}\\ \text{more than two}\\ \text{a few}\\ \text{several}\\ \text{many}\end{array}\right\}$ movies each.

(16) *The boys saw $\left\{\begin{array}{c}\emptyset\\ \text{some}\\ \text{a certain}\\ \text{the}\\ \text{those}\\ \text{few}\\ \text{most}\\ \text{all}\end{array}\right\}$ movies each.

Most theories of generalized quantifiers take indefinites to align with counting quantifiers in contributing existential quantifiers (Barwise and Cooper 1981). Although bare noun phrases are sometimes analyzed as kind-terms, they nonetheless also contribute existential quantification after the kind-level meaning is shifted to an object-level meaning in episodic sentences (e.g., Carlson 1977, Chierchia 1998, Dayal 2004). If existential quantification is what standardly introduces discourse variables via random assignment, then the first two cases in (16) should have the same PCDRT-theoretic translation as (17).

(17) $\textbf{max}^{x}(\textbf{boy}\ x) \wedge \delta_{x}([y] \wedge \textbf{movies}\ y \wedge \textbf{saw}\ y\ x) \wedge y > 1$

As long as it is not the case that the boys saw the same movies, the evaluation-level plurality test is satisfied. In other words, these sentences should be well-formed, contrary to fact.

In the next subsection, I show that however useful as a descriptive label, Counting Quantifier Constraint does not reliably help us pick out the class of noun phrases that can host binominal *each*. What matters to binominal *each* is the measure function embedded inside its host.

3.3 Extensive Measurement Constraint and Monotonicity

Counting Quantifier Constraint has been around for a while, but only quite recently did Zhang (2013) notice that it is insufficient even as a description. Concretely, Zhang (2013) observes that the type of measurement also plays a crucial role in constraining what counting quantifiers may host binominal *each*: extensive measurements give rise to good hosts but non-extensive measurements give rise to hostile hosts.

It is widely assumed that numeral expressions such as *two students* and *seven feet* have more structure than what meets the eye. In addition to the number word and the common noun, they also contain measure functions like **cardinality**, **height**, **weight**, **speed**, and **temperature**. According to Lønning (1987), a measure function denotes a mapping between a class of physical objects and a degree scale that preserves a certain empirically given ordering relation, such as "be lighter than" or "be cooler than." Degrees are further mapped to numbers by unit functions like **pound** or **kilogram**. Krifka (1989, 1998) classifies measure functions into two types—extensive and non-extensive measure functions. Crucially, these two types of measure functions differ from each other with respect to the property *additivity*. For example, **weight** is extensive since for any object, its weight is equal to the weight of all its parts added together; whereas **temperature** is non-extensive since the temperature of an object is not always equal to adding up the temperature of its parts.

The examples in (18) and (19) demonstrate Zhang's observation that binominal *each* can only be hosted by a noun phrase with an extensive measure function. To rule out the concern that some of the non-intensive measure functions give rise to a more complex structure, as in the case of **speed**, or a less natural noun phrase, as in the case of **purity**, a minimal pair using the measure phrase *60 degrees* is offered. In (18-d), *60 degrees* is a measurement of the angles, and in (19-a), the same form is a measurement of the temperature of drinks.

(18) a. The boys read two books each. **cardinality**
b. The girls walked three miles each. **distance**
c. The windows are four feet (tall) each. **height**
d. The angles are 60 degrees each. **angle**

(19) a. *The drinks are 60 degrees (Fahrenheit) each. **temperature**
b. *The girls walked at three miles-per-hour each. **speed**
c. *The gold rings are 24 Karat each. **purity**

Analyses based on evaluation-level plurality do not expect the contrast between (18) and (19), since they do not have a component making reference to the property of measure functions. In addition, they wrongly predict that binominal *each* cannot be hosted by measure phrases other than those involving cardinality. Consider (18-c), in which every window has the same height, i.e., four feet. In other words, there is only one height-value that the windows are associated with, which should fail the evaluation-level plurality requirement. The fact that (18-c) is judged true and acceptable is therefore unexpected.

In fact, binominal *each* is not the only natural language item that cares about the distinction between extensive and non-extensive measurement. Schwarzschild (2002, 2006) points out a similar contrast in pseudo-partitives: pseudo-partitives admit extensive measurement, as in (20), but reject non-extensive measurement, as in (21).

(20) a. two pounds of cherries **weight**
b. thirty liters of water **volume**

(21) a. *five degrees Celsius of the water in this bottle **temperature**
b. *five miles an hour of running **speed**

In addition, Wellwood (2015) observes similar contrasts in comparatives. Both sentences in (22) can express comparisons involving extensive measurement, but neither can express a comparison involving non-extensive measurement. For example, in (22-a) the amount of the soup that Al bought is larger than the amount of the soup that Bill bought. The amount may be understood in terms of **volume** or **weight**, but not **temperature**.

(22) a. Al bought as much soup as Bill did. **volume**, **weight**, ***temperature**

b. Al ran as much as Bill did. **time**, **distance**, ***speed**

Schwarzschild (2002, 2006) accounts for the sensitivity of measurement constructions to types of measure function by invoking the notion of **monotonicity**. Wellwood (2015) provides a formal definition of this monotonicity condition on measurement, as shown in (23). This condition requires the part-whole structure of the domain of a measure function be preserved in the domain of degrees.

(23) **Monotonic Measurement** (Wellwood 2015)
A measure function μ is monotonic iff

a. there exists $x, y \in D_{\sqsubseteq_{part}}$, such that $x \neq y$, and
b. for all $x, y \in D_{\sqsubseteq_{part}}$, if $x \sqsubset^{part} y$, then $\mu(x) <^{deg} \mu(y)$

The monotonicity constraint of Schwarzschild and Wellwood says the following: only monotonic measure functions can be used in the measurement constructions like pseudo-partitives or comparatives. Consequently, extensive measure functions, but not non-extensive ones, pass the constraint. Consider a portion of coffee, c, and two of its proper parts, c1 and c2. c necessarily measures a greater degree by **volume** or **weight** than that of the parts c1 and c2, but c, c1 and c2 typically have the same temperature. If they don't, the temperature of the c is falls somewhere between the temperature of c1 and the temperatures of c2, making **temperature** non-monotonic.

It is reasonable to assume that constructions with binominal *each* also obey some form of Monotonicity Constraint. This has the following effect: to qualify as a host of binominal *each*, a noun phrase must have a measure function, and the measure function must be an extensive measure function to satisfy Monotonicity Constraint. It is clear that Monotonicity Constraint straightforwardly accounts for the sensitivity of binominal *each* towards extensive and non-extensive measure functions. In addition, it illuminates Counting Quantifier Constraint. Counting Quantifiers are a type of measure phrases, typically involving the extensive measure function **cardinality**. By contrast, bare nouns and indefinites do not contribute any measure function, making them unsuitable hosts. Additionally, Schwarzschild (2006) shows that NPs with Q-adjectives like *many*, *a few*, *a little* and *a lot of* have a syntax similar to measurement phrases and must be associated with a monotonic measure function. In this respect, NPs formed out of them are no different from counting quantifiers.

3.4 Interim summary

It is important to understand which component of the PCDRT-theoretic analyses the three empirical challenges actually challenge. At the core of in the PCDRT-theoretic analyses are the following two pieces:

- A dynamic plural logic allowing information about plurality and dependency to be recorded.
- An evaluation-level plurality constraint that accounts for the variation inference.

The first empirical challenge does not target the whole dynamic plural logic tradition, but only disagrees with how flexibly one should allow dependency to be introduced into a discourse record. It is still useful to have a dynamic plural logic that records plurality and dependency so that we can make reference to this structured discourse record to state some form of the evaluation-level constraint to deliver the variation inference. The second and third empirical challenges target the evaluation-level plurality constraint, suggesting that it is not precise enough to capture why counting quantifiers with extensive measurement should be privileged.

The purpose of the present paper is precisely to offer an analysis that deviates from the PCDRT-theoretic analyses in these two regards. At the heart of the analysis is a monotonicity constraint that makes reference to measurement and distributivity. To ease the reader into this novel monotonicity constraint, I first lay out the key ideas of the constraint without the burden of a compositional semantics in Section 4, allowing the reader to assess the validity of the monotonicity constraint and to imagine their own semantic framework in which such a constraint may be embodied. Then, in Section 5.1, I suggest a concrete dynamic logic to implement the monotonicity constraint in a compositional manner. The dynamic logic is rooted in the tradition of van den Berg (1996) but also has innovations borrowed from the PCDRT framework.

4 Discourse-level monotonicity, informally

I propose that binominal *each* introduces a constraint that ensures the measure function in its host to be monotonic relative to the internal structure of the functional dependency established by distributivity. In slightly more concrete terms, the distributivity expressed in (24) establishes a dependency between the boys and the movies, as visualized in Figure 7.

(24) The boys saw two movies each.

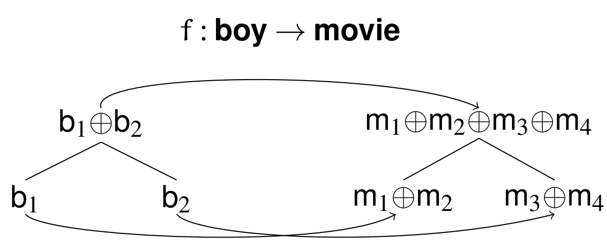


Figure 7: Dependency established via distributivity

Here, boy1 saw movie1 and movie2, while boy2 saw movie3 and movie4. The movies seen between the two boys are movie1, movie2, movie3 and movie4. Let's assume a function f that maps each boy to the movies he saw and also sums of boys to the sums of movies they saw. In other words, f encodes the functional dependency induced by distributivity and is cumulatively closed (just like an info-state encoding functional dependency). With f, we can define a monotonicity condition on a measure function μ relative to distributivity, as in (25). I call this type of monotonicity 'd-monotonicity'.

(25) **Monotonicity relative to distributivity (d-monotonicity, with f)**
A measure function μ is d-monotonic iff there is a function f such that

a. NON-DECREASING MAPPING
$\forall a, a' \in \textbf{domain}\ f.\ a \leqslant a' \rightarrow \mu(f a) \leqslant \mu(f\ a')$, and

b. NON-CONSTANT MAPPING
$\exists b, b' \in \textbf{domain}\ f.\ \mu(f\ b) \neq \mu(f\ b')$

(25-a) requires that the part-whole relation found in the domain of f be mapped **non-decreasingly** to the measurement of the range of f. Since f encodes the functional dependency induced by distributivity, this amounts to making reference to the structure of distributivity. In other words, we are referring to parts of a distributivity dependency that stand in a part-whole relation. Modulo the association with f, (25-a) is a standard definition of non-decreasing monotone functions. It is weaker than the definition of monotonic measure functions found in Wellwood (2015), which picks out strictly increasing functions among the non-decreasing ones. I will return to this difference after demonstrating how the definition in (25) works as a whole. (25-b) requires measurement variability in the range of f. It is comparable, but crucially not the same, as evaluation-level plurality found in the PCDRT-theoretic analyses. I will also return to their differences shortly.

<u>*Extensive Measurement Constraint*</u> Let me start by demonstrating how d-monotonicity captures Extensive Measurement Constraint. Consider (24) with a function f as illustrated in Figure 7. The measure function in this case is **cardinality** (or $\mu_{\textbf{card}}$). It is clear that (24) in this setup satisfies (25). First, suppose we take elements b1, b2 and b1⊕b2, the former two are proper subparts of the last one. f maps b1 to m1⊕m2, b2 to m3⊕m4 and b1⊕b2 to m1⊕m2⊕m3⊕m4. The cardinality function μ maps f b1 to 2, f b2 also to 2, and f b1⊕b2 to 4, as shown in Figure 8. Since the measurement of the range of f does not decrease (in fact, it increases) as we consider increasingly bigger elements in the domain of f, we can conclude that (25-a) is satisfied. In addition, there are at least two elements in the domain of f that get mapped to elements in the range of f that also yield different measurements. For example, b1 and b1⊕b2 are such a pair, so are b2 and b1⊕b2. We can conclude that (25-b) is also satisfied.

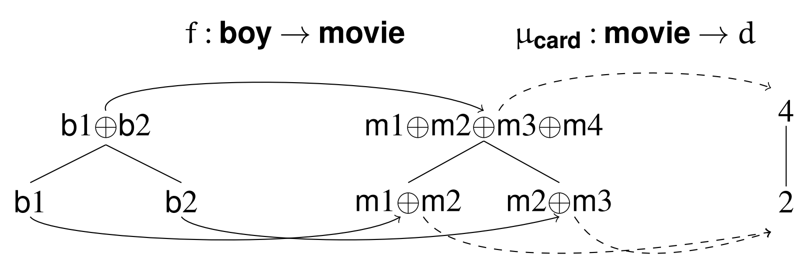


Figure 8: Extensive measurement tracks the internal structure of distributivity

(25-a) alone is a rather weak condition. In fact, as long as the measure function involved in the host of binominal *each* is extensive, it is always satisfied, regardless of how many elements there are in the domain and the range of f. One can verify this by constructing scenarios with only one element in the domain of f and/or only one element in the range. In addition, if $\mu_{\textbf{dim}}$ is non-extensive, as long as the range of f is a singleton, or the range of $\mu_{\textbf{dim}}$ is a singleton, (25-a) is satisfied. Therefore, to have the right strength, (25-a) has to be complemented by (25-b).

What (25-b) requires is that the values stored in the range of f must yield different degrees after being measured by a measure function. This rules out the possibility of all values in the range of f having the same measured degree. For example, (26) has a non-extensive measure function **temperature** (or $\mu_{\textbf{temp}}$) that typically yields a uniform degree for all the values in the range of f, as illustrated in Figure 9. It is predicted to fail non-constant mapping, i.e., (25-b), and hence violate d-monotonicity. Note that it does not violate NON-DECREASING MAPPING in (25-a), as the measurement is indeed a non-decreasing mapping of the domain of f, albeit in a trivial way as there is only one degree in the range of $\mu_{\textbf{temp}}$.

(26) *The boys bought 60-degree coffee each.

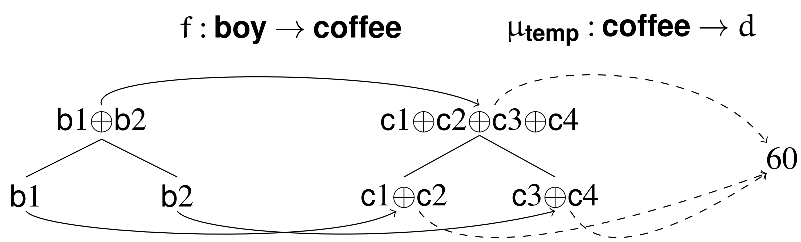


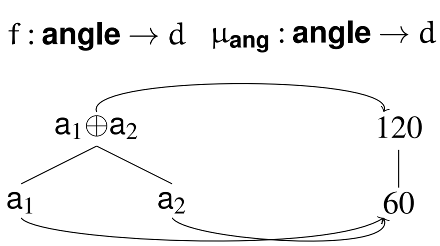
Figure 9: Non-extensive measurement dose not track the internal structure of distributivity

If measuring the range of f indeed yields different degrees, as in the case of cardinality measurement as illustrated in Figure 8, non-constant mapping is satisfied.

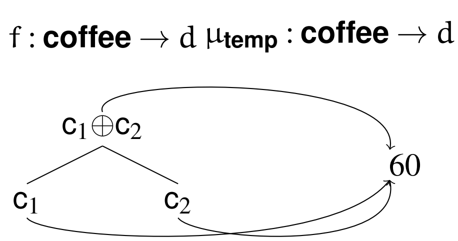
Interestingly, f and $\mu_{\textbf{dim}}$ may happen to be the same function, and the contrast between extensive and intensive measurement still holds, as shown in (27-a) and (27-b).

(27) a. *The coffees are 60 degrees each.
b. The angles are 60 degrees each.

In these two examples, a predicative measure phrase helps map individuals in the distributivity key to the corresponding degrees of measurement as indicated by the measure phrase. Concretely, the coffees (or angles) are distributively checked for their temperature (or degree). So, f encodes a functional dependency between coffees (or angles) and their temperatures (or degrees). $\mu_{\textbf{temp}}$ (or $\mu_{\textbf{ang}}$) is identical to f in being a temperature (or angle) measure function. Both sentences satisfy NON-DECREASING MAPPING as stated in (25-a). However, (27-a) fails NON-CONSTANT MAPPING while (27-b) satisfies it. This is because there is only one temperature, i.e., 60 degrees, in association with f in (27-a), but two degrees, i.e., 60 degrees and 120 degrees, in association with f in (27-b). The contrast is illustrated in Figure 10.



(a) Angle



(b) Temperature

Figure 10: Measure phrase host

At first glance, NON-CONSTANT MAPPING, as conceived here, is quite similar to evaluation-level plurality, as formulated in Henderson (2014), Champollion (2015) and Kuhn (2017). For one thing, they both require a non-constant function. However, they crucially differ in whether they make reference to the internal mereological structure of the domain and the range of f. Evaluation-level plurality does not, while NON-CONSTANT MAPPING does. Moreover, evaluation-level plurality does not have a measurement component while NON-CONSTANT MAPPING does. Despite these differences, the two make very similar predictions if we are only concerned with individuals in the range of f and cardinality measurement. After all, more values in the range of f automatically means a bigger plurality and hence a larger measurement.

To see how the two notions differ, we need to let f map not just individuals to individuals, but also individuals to degrees, i.e., let f store the dependency provided by measure functions. We also need to extend evaluation-level plurality accordingly so that we have designated degree drefs x_d for storing degrees, as illustrated in (28).

Definition Evaluation-level plurality (PCDRT)

(28) $G[\![x_d > 1]\!]H = \mathbb{T}$ iff $G = H$ and $|\{h\, x_d \mid h \in H\}| > 1$

Again, evaluation-level plurality is defined by considering the values, i.e., degrees in this case, stored in x_d globally. If all assignments assign the same degree to the variable, then there is no variation of degrees at the evaluation level. In other words, evaluation-level plurality undesirably neutralizes (27-a) and (27-b). The neutralization happens not because accounts based on evaluation-level plurality do not have measure functions, but because they fail to take into consideration the mereological nature of the functional dependency induced by distributivity. We may say that they succeed in accounting for variability in association with distributivity, but fail to account for variability in association with the structure of distributivity.

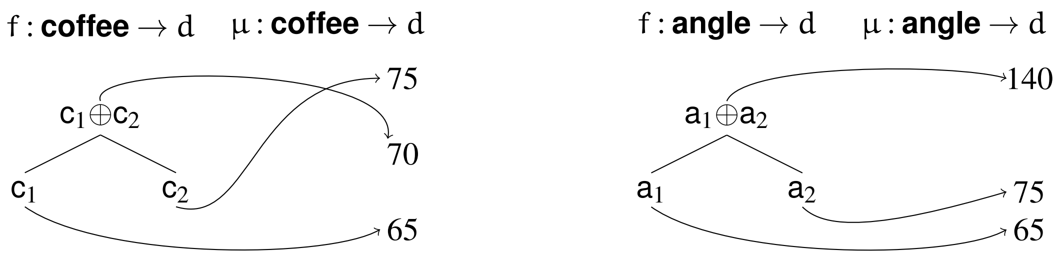
Switching to measurement instead of values preserves the variation requirement. In particular, (25-b) entails that the range of f is **not a singleton**, regardless of the type of measure function involved. Should it be a singleton, there would be no measurement variability in the range of f. This predicts that (29) fails the monotonicity constraint.

(29) *The boys saw two movies each, namely *Inception* and *The Matrix 1*.

One may suspect that (25-b) alone is sufficient to guarantee the variation inference and the privilege of extensive measure functions. It is not. It can be satisfied with a non-extensive measure function as long as the function yields different degrees for different values in the range of f. For example, consider binominal *each* whose host is a measure phrase with a modified numeral, such as (30-a) and (30-b). Figure 11 illustrates how f and $\mu_{\mathbf{dim}}$ works in these two sentences.

(30) a. *The drinks are more than 60 degrees each.
b. The angles are more than 60 degrees each.

(30-a) satisfies NON-CONSTANT MAPPING (as well as evaluation-level plurality), as the range of f has different degrees. However, it is still not well-formed. This is because it violates NON-DECREASING MAPPING: there is a pair of elements in the domain of f that stand in a part-whole relation whose corresponding measurement fails to preserve the the order of the pair, as indicated by the crossing lines in Figure 11a. By contrast, (30-b) satisfies both NON-CONSTANT MAPPING



(a) (30-a) (b) (30-b)

Figure 11: Violation (a) and observation (b) of non-decreasing mapping

Expressions	compatibility with measure units	host binominal *each*
(modified) numerals two, at least/most two, more/less than two e.g., Hackl (2000), Kennedy (2015)	yes e.g., two pounds more than five miles	yes e.g., two books each
quantity expressions a few, a couple, many e.g., Rett (2014), Solt (2015)	yes e.g., a few gallons	yes e.g., many movies each
quantity comparative more, as many (much) as e.g., Wellwood (2015)	yes e.g., as many pounds as	yes as many books each
quantificational determiners no, some, few, most, every, all	no e.g., *most miles	no e.g., *most books each

Table 1: Expressions that can (and cannot) form a host for binominal *each*

(as well as evaluation-level plurality) and NON-DECREASING MAPPING, as indicated in Figure 11b.

Counting quantifier constraint Lastly, we predict that noun phrases without an appropriate measure function component cannot host binominal *each*. A natural question that arises is how we can diagnose the presence of a measure function component. I do not have a satisfactory answer at this point. However, compatibility with unit functions like *pound(s)* and *mile(s)* seems to be a rather reliable test: if a determiner-like expression is compatible with measure units like *pounds* and *miles*, then it can form a noun phrase that can host binominal *each*. Some examples are given in Table 1.[7]

It has been pointed out that noun phrases with the indefinite article *a* are better than those with the determiner *some* in hosting binominal *each*, although not all speakers accept them equally well (Safir and Stowell 1988, Szabolcsi 2010, Milačić et al. 2015), as illustrated in (31).

(31) a. %The boys read a book each.

[7] *Some* sometimes does occur with unit functions, as in *gained some inches* and *lost some pounds*. In these cases, the unit functions are interpreted as standing in for the entities they measure, i.e., *height* and *weight*, respectively.

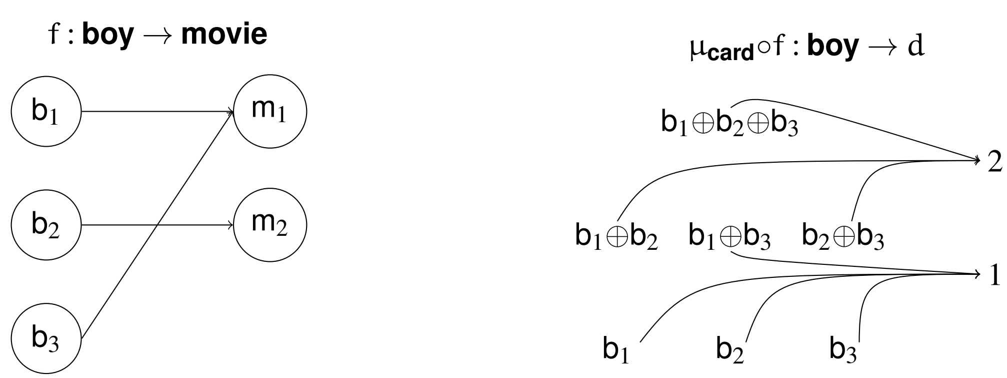
b. *The boys read some book(s) each.

Interestingly, *a* is also compatible with unit functions in ways that *some* is not. Of course, this is not at all surprising given how many linguists think that (synchronically and/or diachronically) *a* is derived from *one* (e.g., Perlmutter 1970, Chierchia 2013, Kayne 2015). Given these considerations, it is conceivable that *a* is ambiguous between a (weak) numeral *one* and an existential determiner, while *some* is only an existential quantificational determiner without a measure function component.

(32) a. a mile, a pound, an inch
b. *some mile(s), *some pound(s), *some inch(es)

NON-DECREASING *and* NON-CONSTANT *vs. strictly increasing* The decision on a weaker form of monotonicity, one in terms of a non-decreasing mapping, instead of a strong form requiring a strictly increasing mapping, as suggested in Wellwood (2015), is empirically motivated. Consider (33). It is judged true in a scenario like Figure 12a, in which both boy1 and boy3 saw movie1, while boy2 saw movie2. Since the range of the mapping function serves as the domain of the measure function, we can compose the two functions to form a composite function, $\mu_{\mathbf{dim}} \circ \mathrm{f}$ as illustrated in Figure 12b: the domain of the function is the values associated with the distributivity key, i.e., the boys in this case, and its range is the measured degree of the values introduced by the host, i.e., the cardinality of the movies.

(33) The boys watched one motive each.



(a) Dependency between boys and movies

(b) Dependency established by $\mu_{\mathbf{card}} \circ \mathrm{f}$

Figure 12: Non-decreasing mapping

In this situation, the cardinality of $\mathrm{f}\ \mathsf{b}_1$ is the same as that of $\mathrm{f}\ \mathsf{b}_1 \oplus \mathsf{b}_3$. Similarly, the respective cardinality of $\mathrm{f}\ \mathsf{b}_1 \oplus \mathsf{b}_2$ and $\mathrm{f}\ \mathsf{b}_2 \oplus \mathsf{b}_3$ is the same as that of $\mathrm{f}\ \mathsf{b}_1 \oplus \mathsf{b}_2 \oplus \mathsf{b}_3$. In other words, the composite function is non-injective.

If the monotonicity constraint is formulated to require a strictly increasing mapping, like (34), the situation in Figure 12 is predicted to be incompatible with (33), precisely because of the non-injective nature of the composite function illustrated in Figure 12b.

(34) **Strictly increasing d-monotonicity** (rejected)
$\forall a, a' \in \mathbf{domain}\ \mathrm{f}.\ a < a' \rightarrow \mu(\mathrm{f}\ a) < \mu(\mathrm{f}\ a')$

However, formulating the monotonicity constraint as a non-decreasing and non-constant mapping, as in (25), does not run into this problem. The composite function in Figure 12b is non-decreasing and non-constant. Therefore, it is predicted that (33) is acceptable in the scenario depicted in Figure 12a.

5 Formalizing d-monotonicity

Now that d-monotonicity has been established, we are ready to supplement it with a more compositional semantics. In fact, it is not difficult to imagine what kind of framework we need to implement d-monotonicity compositionally. The framework should satisfy the following criteria:

- Criterion 1: It should allow us to talk about measure functions of various sorts.
- Criterion 2: It should allow us to introduce distributivity dependency and its structure and refer back to it. In other words, it should make concrete how f is assembled.
- Criterion 3: Since measurement kicks in after f is established, we need a way to split up the contribution of a host marked by binominal *each*, evaluating one part (i.e., the basic semantics of the host) inside the scope of distributivity and the other part (i.e., d-monotonicity) outside the scope of distributivity. The former provides the necessary ingredient to build distributivity and hence the function f. The latter can access f after it is assembled.

Criterion 1 is very easy to satisfy. Any framework that can be enriched to include pluralities and measure functions can be used to model monotonicity. Therefore, a decisive choice depends on the remaining two criteria.

A well-known framework satisfying Criterion 2 is Dynamic Plural Logic of van den Berg (1996) and its close cousin Plural Compositional DRT, devised in Brasoveanu (2006, 2008, 2013).[8] Both approaches have been used to model phenomena pertaining to distributivity. PCDRT has been used more often in the literature of distributive numerals, by Henderson (2014), Champollion (2015) and Kuhn (2017). In this paper, I develop a hybrid approach that is intermediate between DPlL and PCDRT, and enriched to include various sorts of measurement. The developed logic lets assignment functions range over not only atomic individuals, as in van den Berg (1996), but also plural individuals, as suggested in Brasoveanu (2008). However, it sides with van den Berg (1996) in inhibiting dependency introduced by random assignment. Rather, the privilege to introduce dependency into discourse is reserved for a distributivity operator. This is crucially different from the PCDRT tradition, which allow any random assignment to introduce dependency into discourse. I argue in Section 7.1 that this choice explains why d-monotonicity is only seen with distributive predication.[9]

[8]There are other frameworks that track distributivity dependency. For example, Landman (2000) and Champollion (2017) develop accounts for distributivity based on event semantics, in which the dependency is retrievable from events. Huang (1996) develops a semantics for distributivity based on skolem functions, in which the distributivity dependency can be retrieved by using skolem functions. The merit of DPlL is that it not only tracks the dependency in context, but the built-in anaphoric device (i.e., discourse variables) allows us to access the dependency relatively easily.

[9]Besides van den Berg (1996), many studies have observed that distributive quantification has a much easier time introducing dependency than non-distributive quantification, such as cumulative and collective quantification. Some examples are Nouwen (2003), Solomon (2011), and Bumford (2015).

Criterion 3 essentially asks for a split-scope mechanism. Several alternatives have been explored in the literature. An option is by means of a post-supposition (Henderson 2014, Champollion 2015), as discussed in Section 2. Kuhn (2017) points out that post-suppositions, without further assumptions, predict the lack of locality in the licensing of distributive numerals. The prediction is not borne out, as distributive numeral licensing is subject to locality (Kuhn 2017, Charlow, to appear). The follow example, modeled after a similar unacceptable Hungarian sentence reported in Charlow (to appear, ex.104), shows this point.

(35) *The linguists would be happy if two theories each are refuted.

This sentence cannot mean 'for every linguist, he or she would be happy if two theories are refuted.' This is likely due to the fact that the distributive numeral 'two theories each' is embedded inside an island. To model the island sensitivity of distributive numerals, Kuhn (2017) suggests a scope-taking analysis, in which a distributive numeral like *two theories each* has to undergo quantifier-raising (QR) to take wide scope. A drawback of Kuhn's QR analysis is that it fails to account for the grammaticality of distributive numerals with a bound pronoun inside them.

(36) Minden rendezö benevezte két-két filmjét.
every director entered two-two film-POSS.-3SG-ACC
'Every^{x} director entered two fils of his$_{x}$ (in the competition).'

In this Hungarian example, the noun phrase restriction of the distributive numeral has a (possessor) pronominal bound by the quantifier that licenses the distributive numeral. If the distributive numeral has to take wide scope over its licensor to be licensed, then the pronoun is left unbound.

Based on considerations of island sensitivity and pronominal binding, Charlow (to appear) suggests a slightly more involved scope-taking mechanism, one that uses higher order meaning. While deferring a more detailed discussion until Section 5.3 as to what higher-order meaning a binominal *each* phrase has, it suffices to note that Charlow's higher-order meaning approach has a very similar empirical coverage as the post-supposition approach, with the exception of island sensitivity, which favors the former. In this study, I adopt the higher-order meaning approach for it has better empirical coverage, although the choice is largely immaterial to the main claim that binominal *each* makes reference the mereological structure of distributivity.

It should be clear by now what kind of framework is needed to account for the novel properties of distributive numerals observed in this paper. In the next sections, the essential components of such a framework is provided. I begin by discussing the general framework in Section 5.1, followed by translating d-monotonicity into this framework, and lastly in Section 5.3 d-monotonicity is implemented in a compositional manner.

5.1 Formal background: DPlLM

The background for the account is van den Berg's Dynamic Plural Logic (DPlL, van den Berg 1996), extended to include measure functions beyond cardinality.[10] I call this logic **Dynamic Plural Logic with Measurement (DPlLM)**. van den Berg's approach is also adopted and extended in

[10]The logic in van den Berg (1996) is three-valued, i.e., evaluation can result in truth, falsity or undefinedness. I will adopt a simpler version here, by only stating the true transitions, i.e., the pairs of assignments that are in the denotation of a sentence.

Nouwen (2003) and DeVries (2016). This approach is modified in nontrivially ways in Brasoveanu (2006), to give rise to Plural Compositional DRT (PCDRT). PCDRT is then adopted to account for distributive numerals in Henderson (2014), Champollion (2015) and Kuhn (2017). The current account represents the only account targeting distributive numerals in van den Berg (1996)'s tradition. In section 5.1.1, I argue that DPlL, for it ties distributivity and dependency, allows us to understand binominal *each* and its d-monotonicity requirement in a much more transparent way. The remainder of this section introduces DPlLM.

5.1.1 General architecture

In DPlLM, interpreting a formula yields a relation between information states, just as in DPlL and PCDRT. An information state is a set of assignment functions. This differs minimally from the more well known Dynamic Predicate Logic (DPL), which interprets a formula as a relation between assignments, rather than sets of assignments. So, here comes our first formal definition, of information states.

Definition: Information states (info-states)

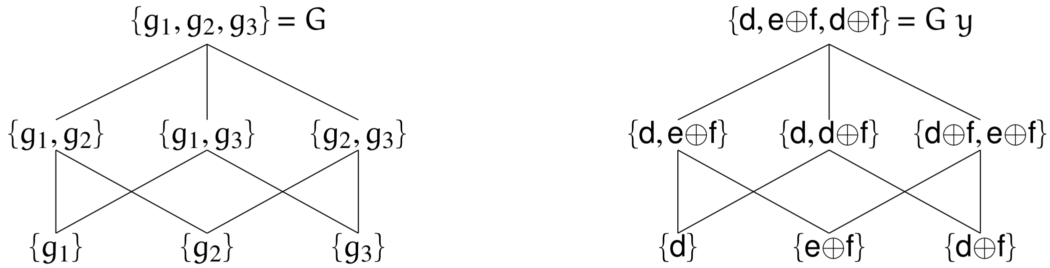
(37) An info-state is a set of assignment functions.

Following a common practice in the literature, an information state is represented as a matrix. The first row has the discourse variables ($x, y, ...$), also known as discourse referents (or d-refs for short), introduced into the info-state (G) so far. The first column lists the variable assignments in it (g_1, g_2, ...). All the other individual cells store a value obtained by applying a variable assignment to a discourse variable. Each column (headed by a discourse variable) stores the collective values assigned to the variable by all the assignments together. Each row (headed by assignments) stores the values a single assignment assigns to all the variables. By considering different assignments, we get potentially different values for the variables, enabling us to talk about dependency among variables.

G	...	x	y	...
g_1	...	a	d	...
g_2	...	b	e⊕f	...
g_3	...	c	d⊕f	...
...	...	...	...	...

Figure 13: An information state G in DPlLM

Although not explicitly mentioned in previous studies using info-states, an info-state also stores the mereological structure of the assignments, as shown in the assignment semi-lattice in Figure 14a. On the basis of the assignment mereology, we can derive the mereology of the values stored in the variables in an info-state. For example, the corresponding values assigned to the variable y by sub-states with various numbers of assignments are organized into a semi-lattice in Figure 14b. It is precisely the ability of info-states to store the mereological structure of the assignments that I capitalize on to implement d-monotonicity, which makes crucial reference to the internal structure of the distributivity dependency encoded in the assignments.



(a) Mereological structure of G (b) Mereological structure of G y

Figure 14: Mereology of an info-state

We turn to assignment functions next. In DPlL, an assignment function only has atomic individuals in its range. This is so because the goal of van den Berg (1996) is to reduce all domain-level pluralities (i.e., the pluralities found in the model other than the pluralities of assignments) into discourse-level, or evaluation-level, pluralities. However, since DPlLM needs to deal with mass nouns and measurement, I modify this assumption and let assignment functions range over any entity (atomic or plural) in a domain.

Definition: Assignment function

(38) An assignment function g takes a variable and when defined, returns a (*possibly plural*) individual.

This is essentially the same modification that Brasoveanu (2008) has made to DPlL in PCDRT. The motivating idea is that if DPlL sets out to model plurality, like the plurality in many static frameworks, then depriving itself of domain plurality (plurality in the range of assignment functions) results in a framework that fails to deal with any domain that lacks well-defined atomicity, like the domains of mass nouns, atelic events, and spatial and temporal intervals (see, for example Bach 1986).

Of course, introducing domain plurality into DPlL comes with a price tag. The system thus developed has two types of pluralities. Discourse plurality, also known as evaluation plurality, comes from considering a plurality of assignment functions. Domain plurality, also known as referential plurality, comes from considering a single assignment function that assigns a plurality to a variable. When we talk about cardinality and measurement, we need to be careful and keep these two types of plurality apart. Fortunately, thanks to the collectivity evaluation of lexical relations, to be defined in (46), we only need to make reference to evaluation-level cardinality in this framework.

Assignment functions take (discourse) variables as their arguments. Variables are d-refs introduced into an info-state by specific expressions. The strategy for introducing a variable is by using random assignment, as defined in (39).

Definition: Random assignment (DPlLM)

(39) $G[\![[x]]\!]H = \mathbb{T}$ iff $G[x]H$, where

a. $G[x]H := \exists D.\ H = \{g^{x \to d} \mid g \in G \ \&\ d \in D\}$

b. D is a set that may include atomic or plural entities.

This version of random assignment differs from both the DPlL version and the PCDRT version. Unlike the PCDRT version but like the DPlL version, it does not introduce dependency between the introduced variable and any existing variable. Unlike the DPlL version but like the PCDRT version, a single assignment is free to assign any value, atomic or not, to a variable. The result is a random assignment that may introduce plurality but is still dependency free.[11]

An example should help us see how this random assignment works. Assume a domain with atomic and plural entities c, d and $c\oplus d$. Introducing a variable x with the formula $[x]$ on an info-state G leads to seven outputs info-states. Since we're bringing domain plurality into the range of assignment functions, it is necessary for us to represent pluralities as well as singletons. I have chosen to represent them as atomic individuals and sum-individuals, following the tradition of Link (1983).[12]

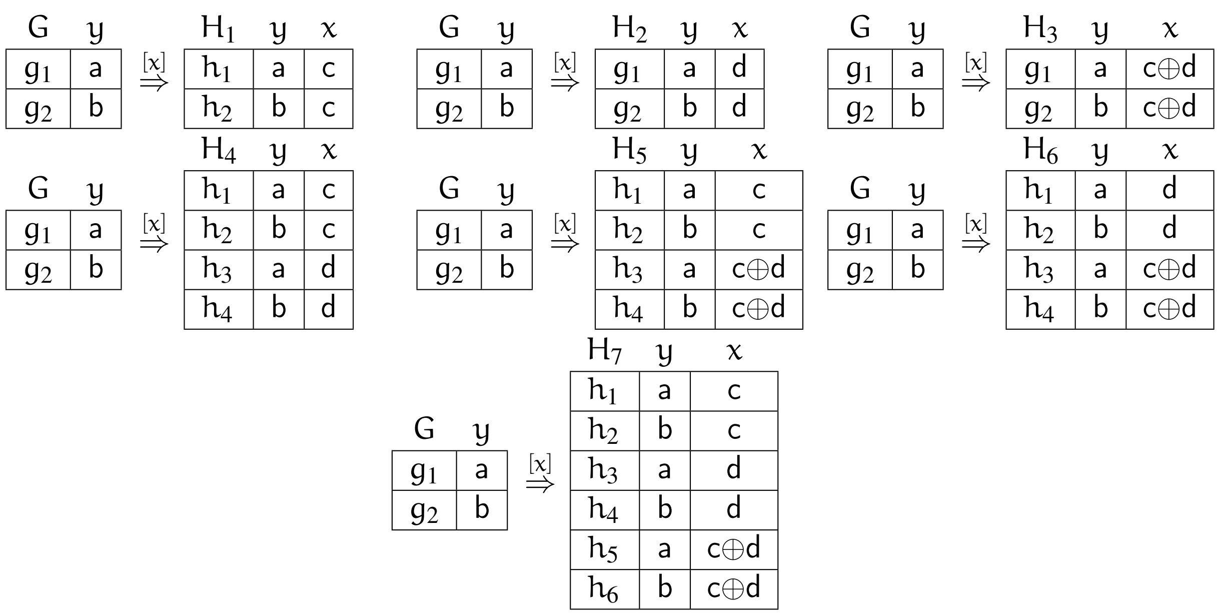


Figure 15: Dependency-free plural-assigning random assignment (DPlLM)

Note that in none of these output info-states does the variable y depend on the variable x or vice versa. To better see this, we need the formal notion of dependence:

Definition: Dependence (Nouwen 2003, see also Kuhn 2017)

[11] It is generally accepted that whether assignment functions may range over pluralities and whether new variable introduction is dependency free are two independent design choices of a dynamic plural logic. The particular version of random assignment devised here offers a concrete demonstration that these two aspects are indeed independent.

[12] An alternative is to represent pluralities as sets following the tradition of Scha (1981), Schwarzschild (1996) and van den Berg (1996). My decision is primary based on readability–the plurality resulting from sets of assignments are represented as sets already, and having referential pluralities modeled as sets inside these sets is not very aesthetically appealing.

(40) y is dependent on x in an information state G iff
$\exists a, b \in G\, x.G|_{x=a}\, y \neq G|_{x=b}\, y$

The notation $G|_{x=a}\, y$ is a shorthand for projecting the values stored in the variable y in the info-state G by only considering those sub-states in which x is assigned the value a. For dependence between two variables x and y to hold, a variable (say y) should not always be assigned the same value for different values assigned to x. In other words, the values stored in y is not constant relative to the values in x.

Since we will make heavy use of value projection, it is handy to have the following ways of projecting values in an info-state and in its sub-states.

Definition: (Sub-state) value projection

(41)
a. $G\, x := \{g\, x : g \in G\}$
b. $G|_{x=a}\, y := \{g\, y : g \in G \;\&\; g\, x = a\}$
c. $G|_{x \in X}\, y := \{g\, y : g \in G \;\&\; g\, x \in X\}$

Since random assignment in DPlLM typically gives us many outputs, some contain straightly less information that the others, we may want to focus on the most informative output. A maximization operator, as defined in (42), serves this purpose. It is a very important ingredient in defining dynamic generalized quantifiers, both in DPlL and PCDRT. The dynamic GQs used in this paper are simpler and do not make use of maximization. However, maximization is still needed for plural definites and universal quantifiers, to model their maximality requirements.

Definition: **Maximization**

(42) $G[\![\mathbf{max}^{x}(\phi)]\!]H = \mathbb{T}$ iff
a. $G[\![[x] \wedge \phi]\!]H$
b. $\neg\exists H'.H(x) < H'(x) \& G[\![[x] \wedge \phi]\!]H'$

(43) $H(x) < H'(x) = \oplus H(x) < \oplus H'(x)$

To implement d-monotonicity, it is necessary for info-states to store functional dependencies. Since random assignment in DPlLM does not introduce dependency into an info-state, another operator needs to fill in the gap. Following DPlL, the distributivity operator δ_x serves this purpose.

Definition: Distributivity

(44) $G[\![\delta_x(\phi)]\!]H = \mathbb{T}$ iff
a. $G\, x = H\, x$
b. $\forall a \in G\, x.G|_{x=a}\, [\![\phi]\!]\, H|_{x=a} = \mathbb{T}$

The distributivity operator splits up the input info-state into substates based on the values stored in the subscripted variable (x in this case). It then checks that the formula in its scope, i.e., ϕ, holds for each sub-state. Hence, for each sub-state, a distributivity update generates a set of output sub-states. These sets of sub-states are then pointwisely put back together to form the output info-state. If ϕ carries with it a random assignment $[x]$, the new variable gets passed to the output. This way, DPlLM opens up a door for introducing dependency into info-states.

To take a concrete example, interpreting the formula with a random assignment in the scope of a distributive operator $\delta_y([x])$ relative to the info-state G in Figure 15 first splits up the input info-state along the y dimension, resulting in two atomic sub-states, as shown in Figure 16. Then intermediate sub-states are created by updating x to each of the two atomic sub-states and assigning random values to x. Note that within each leg of the distributive update, there is no dependency between x and y. However, after pointwisely collecting the intermediate sub-states to form the set of output info-state, some of the output info-states actually have dependency between x and y, for example H_2 and H_3 in this case.

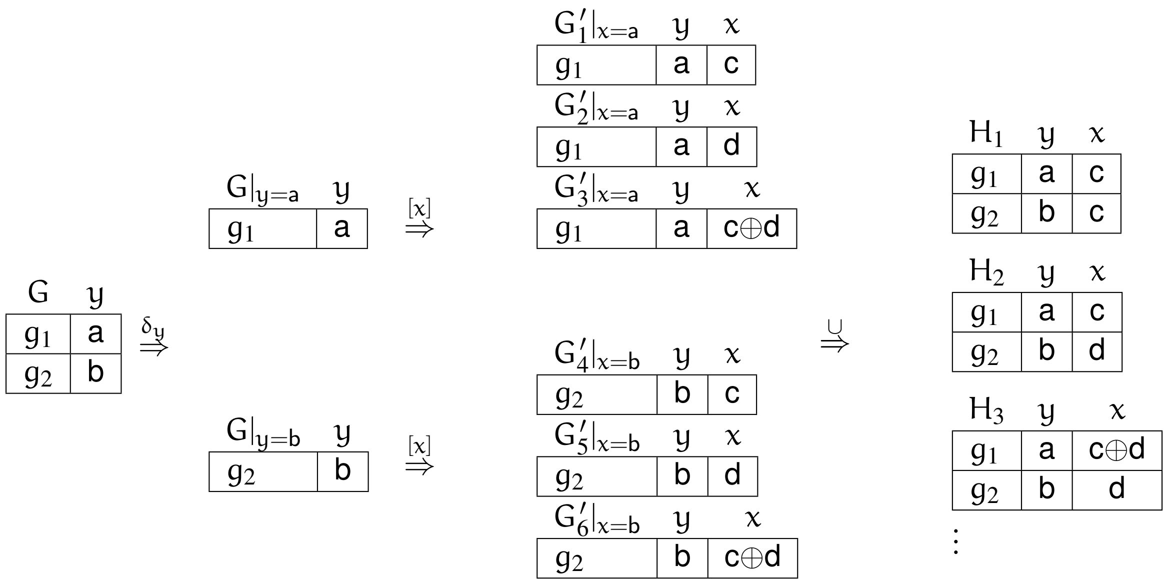


Figure 16: Introducing dependency with help of a distributivity operator

5.1.2 Lexical relations, cardinality and measurement

In DPlLM, all lexical predicates are cumulatively closed by default, following the assumptions in Landman (2000), Kratzer (2007) and Champollion (2017). For this reason, I do not mark a predicate with "*" to indicate cumulative closure.

Definition: Cumulativity of lexical predicates

(45) a. $\mathfrak{I}(\textbf{book}) := \{c, d, c \oplus d, ...\}$
b. $\mathfrak{I}(\textbf{read}) := \{\langle a, c\rangle, \langle b, d\rangle, \langle a \oplus b, c \oplus d\rangle, ...\}$

Lexical relations are satisfied *collectively*, siding with the original DPlL defined in van den Berg (1996). However, since we now have both discourse plurality and domain plurality, we need to modify the way DPlL checks lexical relations to adapt to this change. Let me elaborate on this point.

Recall from the previous paragraph that in our model a lexical predicate like **book** denotes a set of atomic individuals and their sums. A discourse plurality, i.e., the plurality of values stored in a discourse variable like x, has the same type as our lexical predicates, i.e., a set of possibly plural individuals. So, to check whether x satisfies a lexical predicate P, we need to do more than saying that x is in P. There are in principle two ways to move forward. We can either check that each member in our discourse plurality is a member in the lexical relation, amounting to checking lexical relations *distributivity*, a strategy adopted in PCDRT (see also Brasoveanu 2006, 2008).

However, we can also maintain the spirit of DPlL and check lexical relations collectively. To do so, there are also two options. The first option is to 'upgrade' lexical relations to sets of sets of possibly plural individuals. The second option is to 'downgrade' discourse plurality when checking lexical relations. I opt for the latter option and use a summation operator to bring a set of individuals to an individual when checking for lexical relations.

Definition: Lexical relations

(46) $G[\![R(x_1,...,x_n)]\!]H = \mathbb{T}$ iff $G = H$ and $\left\langle \bigoplus H(x_1),...,\bigoplus H(x_n) \right\rangle \in \mathfrak{I}(R)$

The summation operator over discourse pluralities is a useful tool. Let us flesh out its definition below:

Definition: Discourse-level summation

(47) $\bigoplus G(x) = \bigoplus\{g(x) : g \in G\}$

Since there are two types of plurality in this framework, to check cardinality and measurement of the values stored in a variable, we need access to a level that reduces discourse plurality to domain plurality. So, if the variable x stores the values c and $c \oplus d$. When asked how much weight is associated with x, sometimes we don't want to count c twice. It turns out that with help of discourse-level summation, to do so is quite straightforward. We only need to define a cardinality test on the atomic parts of the individual obtained from discourse-level summation (see also Henderson 2014:53 for a similar way to find the atomic parts of a discourse plurality):

Definition: Cardinality test

(48) $G[\![\mu_{\textbf{card}}\ x = n]\!]H = \mathbb{T}$ iff $G = H$ and $|\{x' : x' \leqslant \bigoplus H(x) \wedge \textbf{atom}(x')\}| = n$

Measurement other than cardinality can be defined in a similar way, only this time we do not need to access the atomic parts of our referential plurality. We just need to apply a measure function (with a parameterized dimension, such as **height** or **volume**) to the domain plurality and obtain a degree (on a particular scale, omitted for simplicity).

Definition: Measurement test

(49) $G[\![\mu_{\textbf{dim}}\ x = d]\!]H = \mathbb{T}$ iff $G = H$ and $\mu(\bigoplus H\ x) = d$

Lastly, when there is a standardized measurement unit available, such as *pound* and *gallon*, it is assumed to perform a unit test, ensuring that a measurement unit maps a degree to a number.

Definition: Unit test

(50) $G[\![\mathbf{u}\ d = n]\!]H = \mathbb{T}$ iff $G = H$ and $\mathbf{u}\ d = n$

With these essential enrichments and revisions to DPlL, we have obtained DPlLM, a version of DPlL that is friendly to domain plurality, as PCDRT is, and with useful tools for measurement. Other than the aspects discussed in this section, the core logic largely remains faithful to DPlL. In the next subsection, I define d-monotonicity in terms of DPlLM.

5.2 Proposal: d-monotonicity in DPlLM

Recall that in Section 3, I have sketched the main proposal of this paper: binominal *each* introduces a constraint known as d-monotonicity, checking the monotonic property of measure functions relative to the internal structure of the dependency established via distributivity.

(51) Monotonicity relative to distributivity (d-monotonicity, with f)
A measure function μ is d-monotonic iff there is a function f such that

a. NON-DECREASING MAPPING
$\forall a, a' \in \mathbf{domain}\ f.\ a \leqslant a' \rightarrow \mu(f a) \leqslant \mu(f\ a')$, and

b. NON-CONSTANT MAPPING
$\exists b, b' \in \mathbf{domain}\ f.\ \mu(f\ b) \neq \mu(f\ b')$

The checking of d-monotonicity is facilitated by a function f that maps values stored in the distributivity key to corresponding values stored in the host. The natural correlate of this f in DPlLM is sets of assignment functions, i.e., info-states. To see this, recall that info-states encode not just values assigned to variables and dependencies among different variables, but also internal structures of these dependencies. In more concrete terms, with help of info-states, not only can we retrieve values associated with the distributivity key and the host of binominal *each*, given that distributivity is externally dynamic in this logic, we can also make precise reference to the corresponding values in the host for all the atomic values and their combinations (i.e., pluralities) in the distributivity key. Having access to this structured dependency allows us to conduct measurement on it to check d-monotonicity. Translating d-monotonicity as a dynamic proposition into DPlLM, we obtain (52).

Definition Monotonicity relative to distributivity (d-monotonicity, in DPlLM)

(52) $G[\![\mathbf{dm}_{x,y}\ \mu_{\mathbf{dim}}]\!]H = \mathbb{T}$ iff

a. $H = G$

b. $\forall A, A' \subseteq G\ x.\ A \subseteq A' \rightarrow \mu_{\mathbf{dim}}\left(\bigoplus G|_{x \in A}\ y\right) \leqslant \mu_{\mathbf{dim}}\left(\bigoplus G|_{x \in A'}\ y\right)$

c. $\exists B, B' \subseteq G\ x.\ \mu_{\mathbf{dim}}\left(\bigoplus G|_{x \in B}\ y\right) \neq \mu_{\mathbf{dim}}\left(\bigoplus G|_{x \in B'}\ y\right)$

To begin with, d-monotonicity bears two anaphoric indices. The first one corresponds to the variable introduced by interpreting the distributivity key and the second one corresponds to the variable introduced by interpreting an indefinite host. There is a longstanding tradition in granting binominal *each* an anaphoric component, started in the early work of Burzio (1986) and Safir and Stowell (1988) and later adopted in Dotlačil (2012) and Kuhn (2017). I provide independent justifications for using anaphoric indices in Section 6, where I discuss how negation and downward monotone quantifiers interrupt dynamic binding in binominal *each* constructions.

To check for d-monotonicity of a measure function in DPlLM, we need to access the values stored in the variable the measure function applies to. (52) says that the measure function μ is monotonic on the dependency between x and y iff

- (52-a): Checking for d-monotonicity does not change the info-state in any way (i.e., it's a test).
- (52-b): Measuring y's values in an info-state storing less x's values does not yield a bigger number (or degree) than measuring y's values in an info-state storing more x's values.
- (52-c): In the input info-state, there are at least two sub-parts storing different x''s values that also yield different measurement of y's values.

In addition, I propose that the monotonicity condition in (52) is introduced as an 'output context constraint' in the sense of Farkas (2002) and Lauer (2009, 2012). In particular, (52) is treated as a constraint that is checked after the at-issue content has been established. If the at-issue content cannot pass the test, then the truth condition denoted by the sentence is not defined. As a result, the sentence is *undefined*, rather than *false*. This is to model the fact that sentences with binominal *each* that fail d-monotonicity (for various reasons) are judged unacceptable and not false, as illustrated below:

(53) a. *The drinks are 60 degrees (Fahrenheit) each.
b. *The boys read some books each.
c. *The boys read one book each, namely *Emma*.

The constraint is formulated in (54). The connective $\overline{\wedge}$ indicates that the constraint ψ applies after evaluating the at-issue content ϕ.

Definition Output context constraint

(54) $G[\![\phi \,\overline{\wedge}\, \psi]\!]H = G[\![\phi]\!]H$ if $H[\![\psi]\!]H = \mathbb{T}$; otherwise, undefined.

This definition says: the at-issue content given by ϕ has a truth value only if the output context of ϕ admits ψ. A constraint behaves in a similar way to a presupposition in being a definedness condition, but it differs from a presupposition as the definedness condition is imposed on the *output* context, instead of the *input* context. This way of understanding d-monotonicity makes novel and valid predictions about how it interacts with negation and downward monotone quantifiers. These predictions are discussed in Section 6.1.

5.3 Composition

Like Nouwen (2003) and Brasoveanu (2008), I assume that DPlLM is a typed logic. It includes basic types and derived types as in (55): e for entities, t for truth values, v for events, s for assignments, d for degrees, n for numbers, and a derived type $\tau \to \tau$ for functions.

(55) $\tau : e \mid t \mid v \mid s \mid d \mid n \mid \tau \to \tau$

To keep type description reader-friendly, the following type abbreviations are used:

Name	Type	Abbr.	Variables	Examples
Info-state	$s \to t$	–	G, H	x | y john | sue mary | peter
indivi-dref	$s \to e$	e	u, v	x, y
event-dref	$s \to v$	v	ϵ, ϵ'	e, e'
proposition	$(s \to t) \to ((s \to t) \to t)$	t	ϕ, ψ	John left.
predicate	$(s \to e) \to ((s \to t) \to ((s \to t) \to t))$	$\mathsf{e} \to \mathsf{t}$	P, P'	pretty, book
quantifier	$(\mathsf{e} \to \mathsf{t}) \to \mathsf{t}$	Q	$\mathcal{Q}$	every boy
cardinality functions	$\mathsf{e} \to n$	m	m, m'	$\mu_{\textsf{card}}$
measure functions	$\mathsf{e} \to d$	m	m, m'	$\mu_{\textsf{weight}}$

Table 2: Type abbreviations

I propose that a noun phrase hosting a binominal *each* is a measure phrase. Depending on whether the measure phrase occurs in an argument position, as in (56-a) and (56-b), or a predicate position, as in (56-c), it has slightly different types.

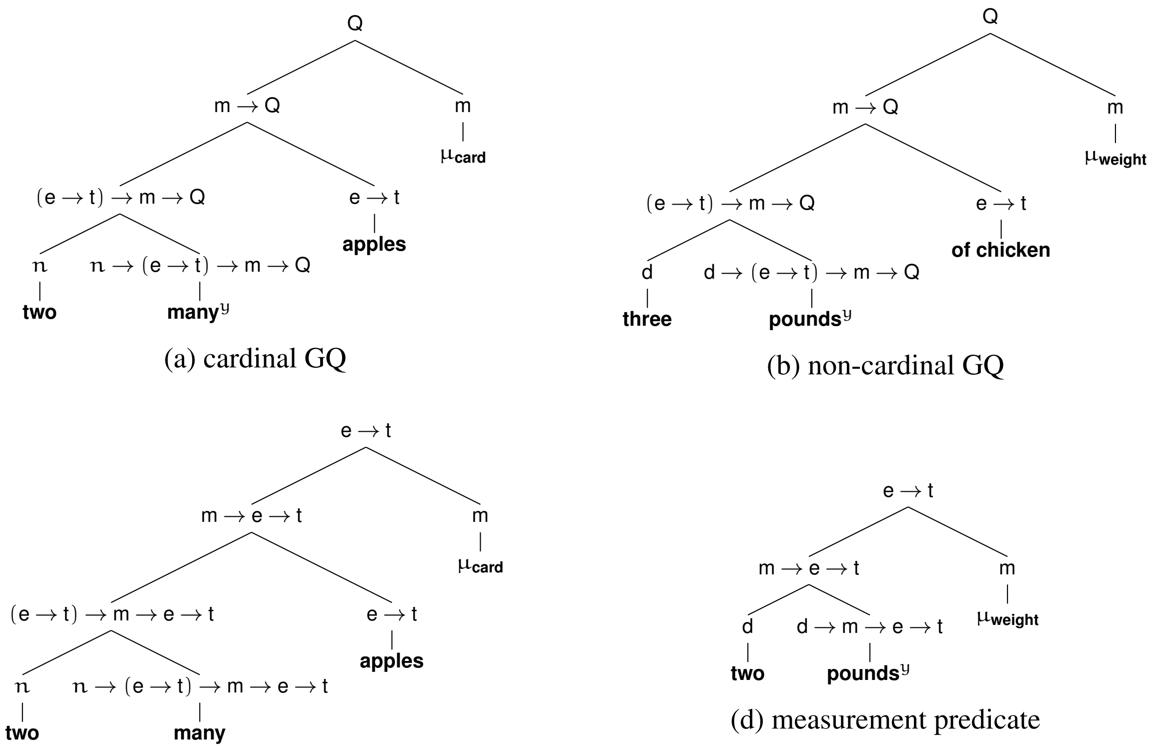
(56) a. John bought two apples.
b. John bought three pounds of chicken.
c. John is six feet (tall).

In an argument position, a measure phrase is a dynamic generalized quantifier (GQ), of type $(\mathsf{e} \to \mathsf{t}) \to \mathsf{t}$. In a predicate position, a measure phrase is simply a predicate, of type $\mathsf{e} \to \mathsf{t}$. However, unlike ordinary dynamic GQs and predicates, measure phrases have two additional components: a measure function and a measure head. The internal structures of different measure phrases are given in Figure 17.

Argumental measure phrases, analyzed as GQs, are shown in Figure 17a and Figure 17b. If the measure phrase is a cardinal GQ, the measure head is a silent determiner akin to the silent *many* in Hackl (2000). The measure head takes a number, a property and a measure function and returns a GQ. This measure head is defined in (57-a). If the measure phrase is a non-cardinal GQ, the measure head is assumed to be provided by a measure unit like *pound(s)*, which takes a number, a property, and a measure function and returns a GQ, as defined in (57-b).

(57) a. $\textbf{many}^{y} := \lambda n \lambda P \lambda m \lambda P'.[y] \wedge P\, y \wedge P'\, y \wedge m\, y = n$
b. $\textbf{pound}^{y} := \lambda n \lambda P \lambda m \lambda P'.[y] \wedge P\, y \wedge P'\, y \wedge \textbf{lbs}(m\, y) = n$

The cardinality measure head **many** selects (with help of agreement or some other means) a cardinality measure function $\mu_{\textsf{card}}$ (type $\mathsf{e} \to n$), while a non-cardinality measure head like **pound** selects an non-cardinality measure function, like $\mu_{\textsf{weight}}$ (type $\mathsf{e} \to d$). A measure function is assumed to be syntactically present and further away from a measure head, unlike that in Hackl (2000), which builds the measure function into the meaning of a measure head.



(a) cardinal GQ

(b) non-cardinal GQ

(c) cardinal predicate

(d) measurement predicate

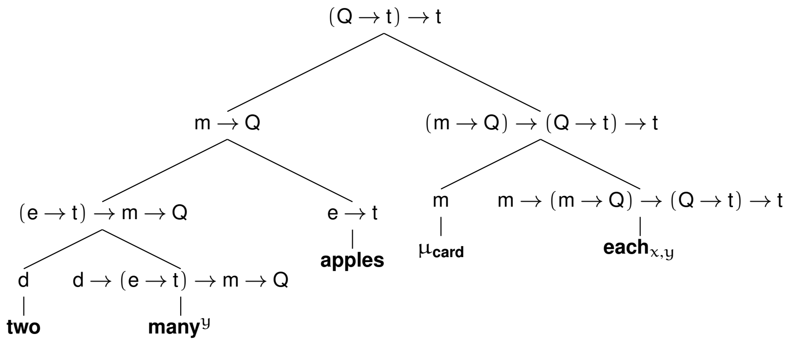
Figure 17: Argumental and predicative measure phrases

If a measure phrase is a predicative and has a nominal predicate (as in *this is two pounds of chicken*), then the measure head takes the same ingredients, but return a predicate rather than a GQ. The corresponding definitions of the predicative measure heads are given in (58-a) and (59-a). Lastly, sometimes a measure phrase may not contain a common head at all, as in *this is two pounds*. I assume that a measure head may optionally not take a nominal predicate as one of its arguments, giving rise to a measure phrase. Sample definitions of the measure heads are given in (58-b) and (59-b).

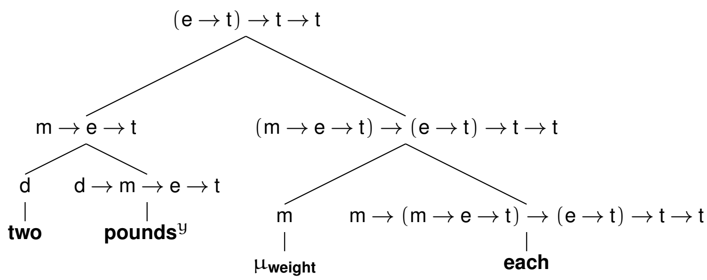
(58) a. $\textbf{many}^{NP} := \lambda n \lambda P \lambda m \lambda u . P\ u \wedge m\ u = n$
b. $\textbf{many}^{MP} := \lambda n \lambda m \lambda u . m\ u = n$

(59) a. $\textbf{pound}^{NP} := \lambda n \lambda P \lambda m \lambda u . P\ u \wedge \textbf{lbs}(m\ y) = n$
b. $\textbf{pound}^{MP} := \lambda n \lambda m \lambda u . \textbf{lbs}(m\ y) = n$

With the assumptions about the internal structure of a measure phrase fleshed out, we are now ready to add binominal *each* to the structure. I assume that binominal *each* attaches to a measure function and turns the whole measure phrase into a higher-order meaning. Concretely, in a cardinal GQ, binominal *each* maps the GQ into a higher-order GQ by turning the measure function from an argument status (it is sought by a $m \to Q$ function) to a function status (it now seeks a $m \to Q$ function), as shown in Figure 18a. Similarly, in a measure phrase predicate, *each* attaches to the measure function and turns the whole measure phrase predicate into a higher order predicate, as shown in Figure 18b.



(a) a higher order dynamic GQ



(b) a higher order measure phrase predicate

Figure 18: Binominal *each* gives rise to a higher-order meaning

Since binominal *each* can be hosted by both argumental and predicative measure phrases, and predicative measure phrases with or without a common noun component, we need to allow it to be type-polymorphic. I offer the schema for defining binominal *each* in (60-a), where f may range over any type α. In addition, when a measure phrase does not introduce any discourse variables, as in the case of a predicative measure phrase, *each* only needs to bear one anaphoric index, i.e., the anaphoric index for the variable storing the individuals measured by the measure function $\mu_{\textbf{dim}}$. This is shown in (60-b).

(60) a. $\textbf{each}_{x,y} := \lambda m \lambda f \lambda c.c(f\ m) \,\overline{\wedge}\, \textbf{dm}_{x,y}\ m$ $\quad m \to (m \to \alpha) \to ((\alpha \to t) \to t)$

b. $\textbf{each}_{x} := \lambda m \lambda f \lambda c.c(f\ m) \,\overline{\wedge}\, \textbf{dm}_{x}\ m$ $\quad m \to (m \to \alpha) \to ((\alpha \to t) \to t)$

As already can be seen in (60-a) and (60-b), after turning a GQ (or predicate) into a higher-order GQ (or a higher-order predicate), binominal *each* is capable of introducing a d-monotonicity constraint in a place different from where the original GQ (or predicate) takes scope. For example, in (60-a), the 'lower-order' GQ $f\ m$ takes scope inside c, but the d-monotonicity constraint is introduced outside c.

To see a concrete example, after composing with all the ingredients inside an argumental cardinal measure phrase, a host with binominal *each* essentially denotes a higher-order dynamic GQ, as shown in (61).

(61) $\textbf{two many}^{y}\ \textbf{movies}\ \mu_{\textbf{card}}\ \textbf{each}_{x,y} =$

$$\lambda c.c\Big(\lambda P.[y] \wedge \textbf{movie}\ y \wedge \mu_{\textbf{card}}\ y = 2 \wedge P\ y\Big) \,\overline{\wedge}\, \textbf{dm}_{x,y}(\mu_{\textbf{card}})$$

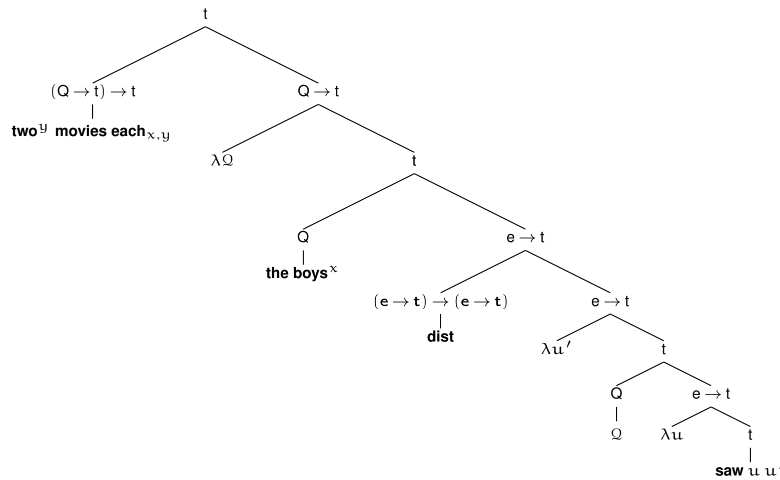


Figure 19: Scope taking of a higher order dynamic GQ

This higher-order dynamic GQ looks for a function from GQ to truth values, puts the GQ (i.e., *two movies*) back in the scope of this function and introduces d-monotonicity outside the scope of this function. Figure 19 shows how scope-taking of the higher-order dynamic GQ works. This is essentially a 'split scope' mechanism that allows *two movies* to scope both inside and outside of distributivity. Scoping it inside distributivity gives us the correct narrow scope reading of *two movies* and scoping it outside of distributivity allows the d-monotonicity to 'associate' with the internal structure of distributivity dependency.

Assuming the lexical entries in Table 3 for the definite NP *the boys*, the verb *saw* and the covert distributivity operator, we obtain the final meaning of the LF, as shown in (62).

Expression	Denotation	Type
$\textbf{the boys}^x$	$\lambda P.\textbf{max}^x(\textbf{boys}\ x) \wedge P(x)$	Q
saw	$\lambda u \lambda u'.\text{SAW}\ u\ u'$	$e \to e \to t$
dist	$\lambda P \lambda u.\delta_u(\textbf{atom}\ u \wedge P\ u)$	$(e \to t) \to (e \to t)$

Table 3: Definite NPs, verbs and the distributive operator

(62) $$\underbrace{\textbf{two}^y\ \textbf{movies each}_{x,y} \left(\underbrace{\lambda \mathcal{Q}.\ \textbf{the boys}^x\ \ \textbf{dist}\left(\lambda u'.\mathcal{Q}\left(\lambda u.\ \textbf{saw}\ u\ u'\right)\right)}_{\beta}\right)}_{\alpha}$$

a. $$\beta := \lambda \mathcal{Q}.\ \textbf{max}^x\left(\textbf{boys}\ x\right) \wedge \delta_x\left(\textbf{atom}\ x\ \wedge\ \mathcal{Q}\left(\lambda u.\ \textbf{saw}\ u\ x\right)\right)$$

b. $$\alpha := \textbf{max}^x\left(\textbf{boys}\ x\right) \wedge \delta_x\left(\begin{array}{l}\textbf{atom}\ x\ \wedge\ [y]\ \wedge\ \textbf{movie}\ y\ \wedge \\ \mu_{\textbf{card}}\ y = 2\ \wedge\ \textbf{saw}\ y\ x\end{array}\right) \bar{\wedge}\ \textbf{dm}_{x,y}(\mu_{\textbf{card}})$$

As shown in (62), the split scope mechanism allows *two movies* to scope inside the distributivity operator but d-monotonicity to scope outside the distributivity operator. The 'association-with-distributivity' effect is clearly seen in the d-monotonicity test in (62-b). The test bears an index x, which is the same index borne by the distributivity operator, i.e., the variable that stores values based on which an info-state is split up into sub-states to check for distributivity.

To test for d-monotonicity, we first assemble the distributivity update. Assuming a scenario in which three boys each saw two different movies, the output of the distributivity update can be visualized in Figure 20.

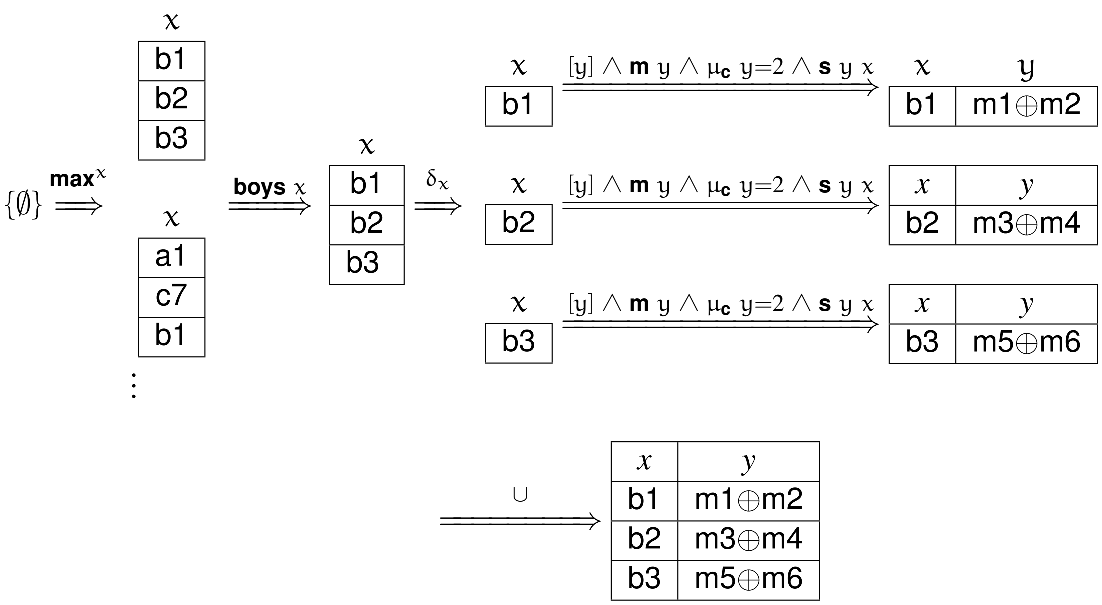


Figure 20: Distributivity update

The d-monotonicity constraint, spelled out in (63), is evaluated against the output of the distributivity update. It first requires that the info-state be split up into sub-states each storing one or more values in the variable x. With three values in x, 7 such sub-states can be found (excluding the empty sub-state, which stores no value in x). Then, it compares these sub-states, requiring that if a sub-state whose x-value is a proper subset of the x-value of another sub-state, then measuring y's cardinality in the former sub-state does not yield a bigger number than measuring y in the latter sub-state.

(63) $G[\![\mathbf{dm}_{x,y}(\mu_{\mathbf{card}})]\!]H = \mathbb{T}$ iff

a. $H = G$ and

b. $\forall A, A' \subseteq G\,x.\ A \subseteq A' \to \mu_{\mathbf{card}}\left(\bigoplus G|_{x\in A}\,y\right) \leqslant \mu_{\mathbf{card}}\left(\bigoplus G|_{x\in A'}\,y\right)$

c. $\exists B, B' \subseteq G\,x.\ \mu_{\mathbf{card}}\left(\bigoplus G|_{x\in B}\,y\right) \neq \mu_{\mathbf{card}}\left(\bigoplus G|_{x\in B'}\,y\right)$

For concreteness, let's consider two info-states, shown in Figure 21, that verify d-monotonicity. In info-state G, three boys each watched a different set of two movies. The cardinality of y (i.e., the movies) in each x sub-state is provided under the matrix. Since the cardinality of y never decreases in a bigger sub-state containing more x-values, non-decreasing mapping is satisfied.

In addition, the cardinality of y is not constant in all the x sub-states, non-constant mapping is satisfied. As a result, d-monotonicity is satisfied by Info-State G. Another info-state that also verifies d-monotonicity is Info-State G', which has two boys seeing two identical movies but a third boy seeing two different movies. Again, this info-state satisfies both non-decreasing mapping and non-constant mapping, hence also d-monotonicity.

G	x	y
g_1	b1	$m1 \oplus m2$
g_2	b2	$m3 \oplus m4$
g_3	b3	$m5 \oplus m6$

Info-state G

$$\mu_{\textbf{card}}(\bigoplus A|_{x\in\{b1\}}\, y) = 2$$
$$\mu_{\textbf{card}}(\bigoplus G|_{x\in\{b2\}}\, y) = 2$$
$$\mu_{\textbf{card}}(\bigoplus G|_{x\in\{b3\}}\, y) = 2$$
$$\mu_{\textbf{card}}(\bigoplus G|_{x\in\{b1,\, b2\}}\, y) = 4$$
$$\mu_{\textbf{card}}(\bigoplus G|_{x\in\{b1,\, b3\}}\, y) = 4$$
$$\mu_{\textbf{card}}(\bigoplus G|_{x\in\{b2,\, b3\}}\, y) = 4$$
$$\mu_{\textbf{card}}(\bigoplus G|_{x\in\{b1,\, b2,\, b3\}}\, y) = 6$$

G'	x	y
g'_1	b1	$m1 \oplus m2$
g'_2	b2	$m1 \oplus m2$
g'_3	b3	$m3 \oplus m4$

Info-state G'

$$\mu_{\textbf{card}}(\bigoplus G'|_{x\in\{b1\}}\, y) = 2$$
$$\mu_{\textbf{card}}(\bigoplus G'|_{x\in\{b2\}}\, y) = 2$$
$$\mu_{\textbf{card}}(\bigoplus G'|_{x\in\{b3\}}\, y) = 2$$
$$\mu_{\textbf{card}}(\bigoplus G'|_{x\in\{b1,\, b2\}}\, y) = 2$$
$$\mu_{\textbf{card}}(\bigoplus G'|_{x\in\{b1,\, b3\}}\, y) = 4$$
$$\mu_{\textbf{card}}(\bigoplus G'|_{x\in\{b2,\, b3\}}\, y) = 4$$
$$\mu_{\textbf{card}}(\bigoplus G'|_{x\in\{b1,\, b2,\, b3\}}\, y) = 4$$

Figure 21: Info-states that verify *the boys saw two movies each*

Of course, not all distributivity updates satisfy d-monotonicity. If the values stored in y does not vary across the distributivity dependency, as in Info-State G''. in Figure 22, d-monotonicity is violated. Recall that since d-monotonicity is modeled as a constraint, the predicted judgment for the corresponding sentence containing a binominal *each* is infelicitous, or unacceptable, rather than false. This is how d-monotonicity captures the variation inference triggered by binominal *each*.

G''	x	y
g_1	b1	$m1 \oplus m2$
g_2	b2	$m1 \oplus m2$
g_3	b3	$m1 \oplus m2$

Info-State G''

$$\mu_{\textbf{card}}(\bigoplus G''|_{x\in\{b1\}}\, y) = 2$$
$$\mu_{\textbf{card}}(\bigoplus G''|_{x\in\{b2\}}\, y) = 2$$
$$\mu_{\textbf{card}}(\bigoplus G''|_{x\in\{b3\}}\, y) = 2$$
$$\mu_{\textbf{card}}(\bigoplus G''|_{x\in\{b1,\, b2\}}\, y) = 2$$
$$\mu_{\textbf{card}}(\bigoplus G''|_{x\in\{b1,\, b3\}}\, y) = 2$$
$$\mu_{\textbf{card}}(\bigoplus G''|_{x\in\{b2,\, b3\}}\, y) = 2$$
$$\mu_{\textbf{card}}(\bigoplus G''|_{x\in\{b1,\, b2,\, b3\}}\, y) = 2$$

Figure 22: An info-state that fails to verify *the boys saw two movies each*

H	x
g_1	w1
g_2	w2
g_3	w3

Info-State H

$$\mu_{\textsf{angle}}(\bigoplus H|_{x\in\{a1\}}\, x) = 60$$
$$\mu_{\textsf{angle}}(\bigoplus H|_{x\in\{a2\}}\, y) = 60$$
$$\mu_{\textsf{angle}}(\bigoplus H|_{x\in\{a3\}}\, y) = 60$$
$$\mu_{\textsf{angle}}(\bigoplus H|_{x\in\{a1,\, a2\}}\, y) = 120$$
$$\mu_{\textsf{angle}}(\bigoplus H|_{x\in\{a1,\, a3\}}\, y) = 120$$
$$\mu_{\textsf{angle}}(\bigoplus H|_{x\in\{a2,\, a3\}}\, y) = 120$$
$$\mu_{\textsf{angle}}(\bigoplus H|_{x\in\{a1,\, a2,\, a3\}}\, y) = 180$$

H′	x
g_1	c1
g_2	c2
g_3	c3

Info-State H′

$$\mu_{\textsf{temp}}(\bigoplus H'|_{x\in\{c1\}}\, x) = 60$$
$$\mu_{\textsf{temp}}(\bigoplus H'|_{x\in\{c2\}}\, x) = 60$$
$$\mu_{\textsf{temp}}(\bigoplus H'|_{x\in\{c3\}}\, x) = 60$$
$$\mu_{\textsf{temp}}(\bigoplus H'|_{x\in\{c1,\, c2\}}\, x) = 60$$
$$\mu_{\textsf{temp}}(\bigoplus H'|_{x\in\{c1,\, c3\}}\, x) = 60$$
$$\mu_{\textsf{temp}}(\bigoplus H'|_{x\in\{c2,\, c3\}}\, x) = 60$$
$$\mu_{\textsf{temp}}(\bigoplus H'|_{x\in\{c1,\, c2,\, c3\}}\, x) = 60$$

Figure 23: Info-states illustrating Extensive Measurement Constraint

When the measure phrase is a predicate, as in (64-a) and (64-b), the measure phrase does not introduce a discourse variable. D-monotonicity is checked by just using a single discourse variable, i.e., the variable storing the values for the distributivity key (the relevant angles for (64-a) and the relevant coffees for (64-b)).

(64) a. The angles are 60 degrees each.
b. *The coffees are 60 degrees each.

The corresponding d-monotonicity constraints have a similar form, as shown in (65), differing only respect to whether the values stored in x are angles or coffees, and whether the measure function measures angle degree or temperature.

(65) $G[\![\textbf{dm}_x(\mu_{\textsf{angle/temp}})]\!]H = \mathbb{T}$ iff

a. $H = G$ and

b. $\forall A, A' \subseteq G\,x.\ A \subseteq A' \rightarrow \mu_{\textsf{angle/temp}}\left(\bigoplus G|_{x\in A}\, x\right) \leqslant \mu_{\textsf{angle/temp}}\left(\bigoplus G|_{x\in A'}\, x\right)$

c. $\exists B, B' \subseteq G\,x.\ \mu_{\textsf{angle/temp}}\left(\bigoplus G|_{x\in B}\, x\right) \neq \mu_{\textsf{angle/temp}}\left(\bigoplus G|_{x\in B'}\, x\right)$

As shown in the info-states in Figure 23, it is possible to satisfy d-monotonicity if the measure function is extensive, as in the case of $\mu_{\textsf{angle}}$ (Info-State H), but not if the measure function is non-extensive, as in the case of $\mu_{\textsf{temp}}$ (Info-State H′).

5.4 Interim summary

I have demonstrated how to translate d-monotonicity as an output constraint in DPlLM, a dynamic plural logic enriched with domain pluralities and measure functions but otherwise faithful to van den Berg (1996) (with the exception of negation, see Section 6.1). The use of plural logic enables us to model distributivity-induced dependency as a discourse plurality, and marrying plural logic with a dynamic logic allows us to record this dependency and its internal structure. The anaphoric component on binominal *each* retrieves this dependency, and the d-monotonicity constraint makes crucial use of the internal structure of this dependency.

In the next section, I take up some predictions of the current proposal, with a goal to show that the use of dynamic semantics makes correct predictions about some rarely discussed properties of binominal *each*, in particular, its interactions with negation and downward monotone quantifiers.

6 Extensions

6.1 Negation

It is well known that negation in dynamic semantics has interesting properties. In DPL, negation is a static closure that confines any dynamic effect in its scope (Groenendijk and Stokhof 1991). Translating its definition into DPlLM gives rise to (66).[13]

(66) **Static negation**
$\mathsf{G}[\![\neg\phi]\!]\mathsf{H} = \mathbb{T}$ iff $\mathsf{G} = \mathsf{H}$ & $\neg\exists\mathsf{K} : \mathsf{G}[\![\phi]\!]\mathsf{K}$

Importantly, if an indefinite occurs in the scope of negation, its dynamic effect is not accessible outside the scope of negation. For this reason, cross-sentential anaphora is predicted to be impossible, as shown in (67).

(67) a. John does not own a car. #It's red.
b. Nobody talked to a man. #He left.

Given the well-documented behavior of negation in dynamic semantics, if binominal *each* indeed makes reference to (the structure of) a distributivity dependency via dynamic binding, as proposed in this study and in recent studies such as Kuhn (2017), one wonders if it interacts with negation as predicted by dynamic semantics and the treatment of negation. This turns out to be a slightly more involved question, given the split scope behavior of a noun phrase hosting binominal *each* and the fact that d-monotonicity is defined as an output constraint. However, after unpacking the complexities, I demonstrate that binominal *each* indeed interacts with negation in a predictable way.

In a simplest sentence with binominal *each* and negation like (68), there are four scopal positions for negation, as indicated in Figure 24.

(68) The students didn't see one movie each.

[13] This is a different negation from the 'dynamic negation' defined in van den Berg (1996). In particular, van den Berg wants to give negation a fully dynamic treatment, which would allow dynamic effects to flow out of the scope of negation. The cost for such a dynamic negation is that a new theory for the interruption effect of negation is necessary. With such a new theory, it is possible to model the same range of data discussed in this section. In short, the empirical consequences discussed here does not pertain to the decision on whether static negation is used or dynamic negation is used in conjunction with mechanisms for dynamic binding interruption.

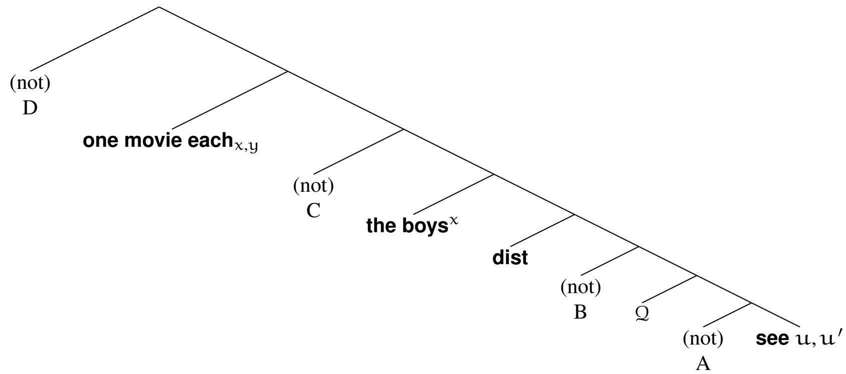


Figure 24: Four scopal possibilities

According to the definition of static negation, there is no well-formed interpretation if negation takes scope anywhere between the higher-order dynamic GQ and the GQ trace. In other words, B and C are not possible scope positions for negation. In position B, the dynamic effect stemming from the GQ trace (more precisely, the reconstructed GQ to the trace position) is blocked outside the scope of negation, as shown in (69); in position C, the dynamic effect stemming from both the reconstructed GQ trace and the distributivity key is blocked, as shown in (70).

(69) B. $\textbf{max}^x\left(\textbf{boy}\ x\right) \wedge \delta_x\left(\textbf{atom}\ x \wedge \neg\left(\begin{array}{l}[y] \wedge \textbf{movie}\ y \wedge \\ \mu_{\textbf{card}}\ y = 1 \wedge \textbf{saw}\ y\ x\end{array}\right)\right) \bar{\wedge}\, \textbf{dm}_{x,y}(\mu_{\textbf{card}})$

‘For each boy, it is not true that he saw any movie. The measurement of movies is d-monotonic to the boys.’

(70) C. $\neg\left(\textbf{max}^x\left(\textbf{boy}\ x\right) \wedge \delta_x\left(\textbf{atom}\ x \wedge \begin{array}{l}[y] \wedge \textbf{movie}\ y \wedge \\ \mu_{\textbf{card}}\ y = 1 \wedge \textbf{saw}\ y\ x\end{array}\right)\right) \bar{\wedge}\, \textbf{dm}_{x,y}(\mu_{\textbf{card}})$

‘Not every boy saw a movie. The measurement of movies is d-monotonic to the boys.’

How do we know that (69) and (70) are indeed out? Ideally, we should detect plain unacceptability. However, due to the availability of scopal option D, which results in a weaker reading, we only observe the lack of d-monotonicity instead of plain unacceptability in these cases. This is because a scenario that verifies (69) or (70) (without d-monotonicity) also verifies the reading generated from having negation in position D. Moreover, d-monotonicity can be negated or sometimes be ignored when negation is in position D. Therefore, our best evidence that (69) and (70) are indeed out comes from the fact that in situations where no boy read any book (B), or not every boy read a book (C), there is no pressure for d-monotonicity to hold. In other words, we simply judge the sentence to be true in these situations regardless of the status of d-monotonicity. This is suggestive that d-monotonicity cannot be evaluated in these scope configurations.

Returning to the remaining scopal options A and D. Interpreting negation in positions A and D gives rise to well-formed interpretations. We begin with A, the simpler case. If negation takes the narrowest scope, it does not interfere with the dynamic effects associated with the distributivity key or the dynamic GQ trace, as illustrated in (71–i). As a result, d-monotonicity can be successfully tested. This can be seen in the paraphrased in plain English in (71–ii), as well as in the sample follow-up utterance in (71–iii).

(71) A. (i) $\textbf{max}^x\big(\textbf{boy}\ x\big) \wedge \delta_x \left(\begin{array}{l} \textbf{atom}\ x \wedge [y] \wedge \textbf{movie}\ y \wedge \\ \mu_{\textbf{card}}\ y = 1 \wedge \textbf{saw}\ y\ x \end{array} \right) \bar{\wedge}\, \textbf{dm}_{x,y}(\mu_{\textbf{card}})$

(ii) For each boy, there is a movie that he failed to see. By the way, the measurement of the movies is d-monotonic relative to the boys.

(iii) Mary didn't see *Avatar*, John didn't see *Matrix*, and Susan didn't see *Kungfu*.

If negation is to take widest scope over both the asserted distributivity and the output constraint d-monotonicity, the resulting interpretation may be false or undefined, depending the particular truth values of the assertion and the constraint.

(72) D. (i) $\neg\left(\textbf{max}^x\big(\textbf{boy}\ x\big) \wedge \delta_x \left(\begin{array}{l} \textbf{atom}\ x \wedge [y] \wedge \textbf{movie}\ y \wedge \\ \mu_{\textbf{card}}\ y = 1 \wedge \textbf{saw}\ y\ x \end{array} \right) \bar{\wedge}\, \textbf{dm}_{x,y}(\mu_{\textbf{card}})\right)$

(ii) It's not true (that every boy saw two movies and by the way, the measurement of movies is d-monotonic to the boys).

Using ϕ to represent the asserted content, i.e., the distributivity update, ψ to represent the output constraint, i.e., d-monotonicity, $\bar{\wedge}$ to represent the outcome of the ϕ as constrained by ψ, and $\neg$ to represent the predicted outcome of negation over the complex meaning, Table 4 summarizes the possible interpretation of $\neg$.

ϕ	ψ	$\bar{\wedge}$	$\neg$
1	1	1	0
1	0	#	#
0	#	#	#

Table 4: Negation over complex meaning

Let us begin with the first two rows. When both the distributivity evaluation and the d-monotonicity constraint evaluation are true, negation is evaluated to 'false', in accordance with native speakers' intuition that the sentence in (73) is simply a false statement.

(73) *In a scenario in which every boy watched a different movie:*
The boys didn't watch one movie each.

When distributivity is evaluated to be true but d-monotonicity is evaluated to be false, negation is evaluated to 'undefined'. The cases in (74) and (75) support this prediction.

(74) *In a scenario in which all the boys watched one and the same movie:*
#The boys didn't watch one movie each.

(75) *In a scenario in which all the cocktails are exactly 60 degrees Fahrenheit.*
#The cocktails aren't 60 degrees (Fahrenheit) each.

When distributivity is evaluated to false. The output is an empty set, so the d-monotonicity constraint cannot be tested. As a result, $\bar{\wedge}$ is evaluated to undefined. Applying negation to an undefined output is also predicted to be 'undefined'. This is where native speakers' intuition differs from the prediction. Intuitively, if distributivity is false, speakers will simply evaluate the negated sentence to 'true', rather than undefined.

This discrepancy between what the semantics predicts and what native speakers perceive is actually a more general phenomenon in dynamic semantics. Consider (76) first. Imagine a situation in which no man came in. The evaluation of *a man came in* is false in this situation. The result is the absurd state, i.e., having no info-state in the output. Consequently, *he*$_x$ in the subsequent clause has no way to refer to *a*^{x} *man* as its antecedent, leading to an undefined evaluation. When the outer negation is evaluated, it should then be evaluated to **undefined**. However, this is not what native speakers do. In fact, they readily judge the negated sentence to be **true**, instead of undefined. It seems as if negation has the ability to ignore dynamic binding failure that happens in its scope.

(76) It's not the case that a^x man came in and he$_x$ sat down.

However, negation cannot ignore just any dynamic binding failure. For example, it cannot ignore dynamic binding failure that fails a gender presupposition, as evidence by the fact that (77) is evaluated to 'undefined'.

(77) #It's not the case that a^x man came in and she$_x$ sat down.

Returning to the last row of Table 4. Native speakers evaluate the negated sentence with binominal *each* to be 'true', rather than the predicted 'undefined', precisely for the same reason. That is, negation has the ability to ignore dynamic binding failure in its scope. Similarly, negation only has the ability to ignore dynamic binding failure in this case, but not other anomaly. For example, if the measurement involved in the host becomes non-extensive, undefinedness resurfaces, as demonstrated in (78).

(78) a. *The boys didn't walked 3 miles per hour each.
b. *The drinks are not 60 degrees each.

6.2 Downward monotone quantifiers

In addition to interacting with negation, binominal *each* is incompatible with downward monotone quantifiers, as illustrated in (79-a) (see also Safir and Stowell 1988). That upward monotone and non-monotone quantifiers are fine is shown in (79-b) and (79-c).

(79) a. *{Few/no/none/at most 10} (of the) boys saw two movies each.
b. Some/many/most (of the) boys saw two movies each.
c. Exactly 10/between 10 and 20 boys saw two movies each.

This is reminiscent of the fact, shown in (80-a), that indefinites in the scope of downward monotone quantifiers also have a hard time supporting a pronoun outside the scope of the quantifiers, in contrast to indefinites in the scope of upward monotone quantifiers (80-b) and non-monotone quantifiers (80-c). So, *they* in (80-a) cannot refer to the collective movies that few boys saw, while the same pronoun in (80-b) and (80-c) can refer to the collective movies that many (or exactly 10) boys saw.

(80) a. Few boys saw two^y movies. #They$_y$'re very boring.
b. Many boys saw two^y movies. They$_y$'re very boring.

c. Exactly 10 boys saw two^{y} movies. They$_{y}$'re very boring.

The parallelism between the two paradigms is intriguing for two reasons. First, there are visible anaphors in (80) but not in (79). The fact that even without visible pronouns, a sentence with binominal *each* interacts with downward monotone quantifiers in a remarkably similar way suggests that binominal *each* has an anaphoric component very similar to overt anaphors like pronouns. Second, the fact that in both paradigms only downward monotone quantifiers cause trouble is revealing. There must be something in downward monotone quantifiers that interact with dynamic binding to cause the markedness.

van den Berg (1996) offers an interesting proposal that can help us tease apart downward monotone quantifiers and other quantifiers. In particular, the suggestion is to analyze downward monotone quantifiers as the negated version of their upward monotone duals. For example, *few* is analyzed as the negation of *many*. Without using the full power of dynamic generalized quantifiers, (81-a) is a way to translate *few* and (81-b) is a way to translate *many* in DPlLM.

(81) a. $\textbf{few}^{x}\,(\phi,\psi) := \neg(\textbf{many}^{x}(\phi,\psi))$

b. $\textbf{many}^{x}(\phi,\psi) := [x] \wedge \phi \wedge \psi \wedge \mu_{\textbf{card}}\, x > \theta$, where θ is a contextually provided threshold.

The crucial bit is the use of negation in the translation of *few* as *not many*. If this analysis is on the right track, then we can reap the benefit of our static negation in the previous subsection. After all, we have already seen that d-monotonicity cannot be tested if the distributivity update is within the scope of negation, as dynamic binding cannot reach into the scope of negation.

For concreteness, let us consider the example in (82-a) and its translation into DPlLM in (82-b). Negation is assumed to have a fixed scope (i.e., position C in Figure 24), as it is part of the lexical meaning of *few*. Binominal *each* along with its host takes wide scope over negation. As a result, d-monotonicity is outside the scope of negation, leading to dynamic binding failure and hence unacceptability.

(82) a. *Few boys saw one movie each.

b. $\neg\left([x] \wedge \left(\textbf{boy}\ x\right) \wedge \delta_{x}\left(\textbf{atom}\ x \wedge \begin{array}{l}[y] \wedge \textbf{movie}\ y \wedge \\ \mu_{\textbf{card}}\ y = 1 \wedge \textbf{saw}\ y\ x\end{array}\right) \wedge \mu_{\textbf{card}}\ x > \theta\right)$

$\bar{\wedge}\, \textbf{dm}_{x,y}(\mu_{\textbf{card}})$

To summarize, the fact that binominal *each* cannot be licensed by a downward monotone quantifier provides additional support for using dynamic semantics to model d-monotonicity.

7 Comparisons with and connections to previous studies

7.1 More on studies in the PCDRT tradition

There are three major differences between the proposal developed here and extant studies in the PCDRT tradition. The first one concerns how the variation inference is modeled. All studies in the PCDRT tradition follow the insight of Henderson (2014) and treat the variation inference as a plurality constraint on the global info-state. However, the present study models the variation inference as a special monotonicity constraint evaluated in association with the internal structure of a distributivity dependency. I have shown that these two strategies make distinct predictions regarding the types of noun phrases that may host binominal *each*.

The second difference lies in the dynamic logic in which the constraint giving rise to the variation inference is couched. Studies in the PCDRT tradition make use of PCDRT, a dynamic plural logic with domain pluralities, dependency-introducing random assignment, and distributive evaluation of lexical relations. The present study is couched in DPlLM, a dynamic plural logic with domain pluralities and a collective evaluation of lexical relations, but crucially no dependency-introducing random assignment. The two logics make distinct predictions regarding whether variation inference could in principle occur without distributivity. The PCDRT-theoretic studies say 'yes' and our DPlLM-theoretic analysis says 'no'.

I must add that d-monotonicity, the constraint designed to replace evaluation-level plurality, can be reformulated with minor changes to adapt to a PCDRT-style dynamic plural logic just as easily. Ultimately, whether a PCDRT-style logic or a DPlLM-style logic should be chosen to couch d-monotonicity should be based on empirical considerations. In particular, if the dependency introduced by random assignment turns out to be very useful, as argued in Brasoveanu (2006, 2008), then a PCDRT-style logic should be favored. However, there is at least some initial evidence, reported in Champollion et al. (forthcoming), that the original empirical motivations for the dependency-introducing random assignment considered in Brasoveanu (2006, 2008) may be accounted for without the machinery used in PCDRT.

The third difference lies in what kind of meaning status is given to the constraint leading to the variation inference. In Henderson (2014) and Champollion (2015), it is analyzed as a delayed at-issue test. In Kuhn (2017), it is modeled as a presupposition. In this study, it is modeled as a delayed output constraint. Modeling it as an at-issue test fails to account for the fact that failure of the variation component leads to ungrammaticality instead of falsity. Modeling it as a presupposition suggests, rather oddly, that the interlocutors should be aware of the variation component even before the distributivity component is asserted.

Despite these differences, the present paper sides with PCDRT-theoretic studies on many fronts, including (i) making the variation inference a core component of the semantics of distributive numerals, (ii) using a dynamic plural logic to model the contribution of distributivity and hence making the dependency induced by distributivity an accessible component for use by the variation component, and (iii) using a delayed evaluation strategy to delay the variation component until distributivity is evaluated. As such, the present study can be seen as a descendent of previous PCDRT-theoretic studies.

7.2 Studies in static semantics

There is a vast literature on binominal *each* couched in static semantics. It is beyond the scope of the present paper to offer a comprehensive review of previous studies on this topic. However, it is worth pointing out the major developments that have paved way for the ideas used in the present paper.

An early study on binominal *each* is Link (1987). He set the stage for treating binominal *each* as a distributivity operator, which is adopted in many subsequent studies, including Zimmermann (2002), Dotlačil (2012), Champollion (2010, 2017). However, these studies places their primary focus on the distributivity component and do not really recognize the variation component. As such, they differ quite drastically from the present paper, which takes the variation component as its primary concern.

There are a few studies that take up the variation component. For example, Safir and Stowell (1988) recognize a strong form of the variation component, and conceive binominal *each* as a one-to-one distribution function, establishing a one-to-one correspondence between elements in the distributivity key to elements in the distributivity share. This strong form of variation is later criticized by Moltmann (1991) and Zimmermann (2002). Cable (2014) extends the semantics established for distributive numerals in Tlingit to binominal *each*, arguing that *each* is both a distributivity marker and bears a variation inference. Despite recognizing the variation component, these studies do not account for Counting Quantifier Constraint or Extensive Measurement Constraint.

Despite these differences, studies in the static tradition have offered great insights to the study of binominal *each* in the present paper. For one thing, it has been a longstanding puzzle how binominal *each* access the distributivity key. The received wisdom is that there are null pronouns in the NP that hosts binominal *each* that help connect it with the distributivity key, as suggested in Safir and Stowell (1988). This idea is further refined in Zimmermann (2002), with the pronoun treated as an anaphoric index directly borne by binominal *each*. The strategy is then imported into a dynamic framework by Dotlačil (2012) and adopted in Kuhn (2017) and the present paper.

Many studies also share the intuition that *each* is a marker of quantificational dependence or subordination. Choe (1987) and Milačić et al. (2015) are notable examples. This intuition is also relevant in the present study, albeit in a slightly different manner. In previous studies, the core contribution of binominal *each* is to signal quantificational subordination. However, in the present study, the core contribution is a variation component formalized in terms of d-monotonicity. Quantificational subordination is a *necessary but not sufficient* condition for using binominal *each*.

As a final note, I would like to relate the present study to the idea of 'structure-preserving binding', developed in Jackendoff (1996) to deal with a a host of phenomena ranging from telicity to quantification. Jackendoff suggests to broaden the notion of binding from a relation between two identical variables to a relation between two variables that are linked in some way. Most importantly, he argues that it is fruitful to study the links in terms of structure-preserving maps. He implements structure-perving binding in his own framework of Conceptual Semantics, which differs from the framework used in this paper quite substantially. However, the core of the idea of structure-preserving binding resonates with the notion of d-monotonicity developed here.

8 Conclusion

In this paper, I have borrowed the insight from studies such as van den Berg (1996) and Champollion (2017)) that distributivity makes available a dependency with a nontrivial internal structure. Following many recent studies, this dependency is modeled with help of a dynamic plural logic. The particular version developed in this paper is a direct descent of van den Berg (1996)'s DPlL. In addition, I have argued that binominal *each* piggybacks on this dependency, and introduces a monotonicity constraint requiring that the measurement of the values associated with its host tracks the part-whole relation of the dependency. I have demonstrated how the measurement and monotonicity-based constraint (called d-monotonicity) better accords with two important generalizations on binominal *each*: Counting Quantifier Constraint and Extensive Measurement Constraint. Lastly, I have also shown that a dynamic treatment of binominal *each* makes correct predictions about its interactions with negation and downward monotone quantifiers, justifying the use of dynamic semantics in this area of research.

References

Bach, E. (1986). The algebra of events. *Linguistics and Philosophy 9*(1), 5–16.

Barwise, J. and R. Cooper (1981). Generalized quantifiers and natural language. *Linguistics and Philosophy 4*, 159–219.

Brasoveanu, A. (2006). *Structured nominal reference and modal reference*. Ph. D. thesis, Rutgers University.

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

Brasoveanu, A. (2011). Sentence-internal *different* as quantifier-internal anaphora. *Linguistics and Philosophy 34*, 93–168.

Brasoveanu, A. (2013). Modified numerals as post-suppositions. *Journal of Semantics 30*, 155–209.

Brasoveanu, A. and D. Farkas (2011). How indefinites choose their scope. *Linguistics and Philosophy 34*, 1–55.

Bumford, D. (2015). Incremental quantification and the dynamics of pair-list phenomena. *Semantics & Pragmatics 8*, 1–70.

Bumford, D. and C. Barker (2013). Association with distributivity and the problem of multiple antecedents for singular *different*. *Linguistics and Philosophy 36*, 355–369.

Burzio, L. (1986). *Italian Syntax*. Reidel Publishers.

Cable, S. (2014). Distributive numerals and distance distributivity in Tlingit. *Language 90*, 562–606.

Carlson, G. N. (1977). A unified analysis of the English bare plural. *Linguistics and Philosophy 1*, 413–457.

Champollion, L. (2010). *Parts of a whole: Distributivity as a bridge between aspect and measurement*. Ph. D. thesis, University of Pennsylvania, Philadelphia, Pennsylvania.

Champollion, L. (2015). Every boy bought two sausages each: Distributivity and dependent numerals. In U. Steindl, T. Borer, H. Fang, A. G. Pardo, P. Guekguezian, B. Hsu, C. O'Hara, and I. C. Ouyang (Eds.), *Proceedings of the 32nd West Coast Conference on Formal Linguistics*, Somerville, MA, pp. 103–110. Cascadilla Proceedings Project.

Champollion, L. (2017). *Parts of a Whole: Distributivity as a Bridge between Aspect and Measurement*. Oxford: Oxford University Press.

Champollion, L., D. Bumford, and R. Henderson (Forthcoming). Donkeys under discussion. *Semantics & Pragmatics*.

Charlow, S. (to appear). Post-suppositions and semantic theory. *Journal of Semantics*.

Chierchia, G. (1998). Reference to kinds across languages. *Natural Language Semantics 6*(4), 339–405.

Chierchia, G. (2013). *Logic in grammar: Polarity, free choice, and intervention*. Oxford University Press.

Choe, J.-W. (1987). *Anti-quantifiers and a theory of distributivity*. Phd, University of Massachusetts-Amherst.

Dayal, V. (2004). Number marking and (in)definiteness in kind terms. *Linguistics and Philosophy 27*(4), 393–450.

Dekker, P. (1994). Predicate logic with anaphora. In M. Harvey and L. Santelmann (Eds.), *Proceedings of Semantics and Linguistic Theory 9*, Ithaca, NY, pp. 79–95. Cornell University.

DeVries, K. (2016). *Independence Friendly Dynamic Semantics: Integrting exceptional scope, Anaphora and their interactions*. Ph. D. thesis, University of California at Santa Cruz.

Dotlačil, J. (2010). *Anaphora and distributivity: a study of same, different, reciprocals and others*. Phd. dissertation, University of Utrecht.

Dotlačil, J. (2012). Binominal *each* as an anaphoric determiner: a compositional analysis. In *Proceedings Sinn und Bedeutung 16*.

Elworthy, D. A. H. (1995). A theory of anaphoric information. *Linguistics and Philosophy 18*(3), 297–332.

Farkas, D. F. (2002). Varieties of indefinites. In B. Jackson (Ed.), *Proceedings of SALT XII*, Cornell University, Ithaca, NY, pp. 59–83. CLC Publications.

Groenendijk, J. and M. Stokhof (1991). Dynamic predicate logic. *Linguistics and Philosophy 14*, 38–100.

Hackl, M. (2000). *Comparative Quantifiers*. Ph. D. thesis, Massachusetts Institute of Technology.

Heim, I. (1982). *The Semantics of Definite and Indefinite Noun Phrases*. Ph. D. thesis, University of Massachusetts, Amherst.

Heim, I. (1983). File Change Semantics and the the familiarity theory of definiteness. In R. Bäuerle, C. Schwarze, and A. v. Stechow (Eds.), *Meaning, Use and Interpretation of Language*, pp. 164–189. Berlin: De Gruyter.

Henderson, R. (2012). *Ways of pluralizing events*. Ph. D. thesis, University of California at Santa Cruz.

Henderson, R. (2014). Dependent indefinites and their post-suppositions. *Semantics & Pragmatics 7*, 1–58.

Huang, S.-Z. (1996). *Quantification and Predication in Mandarin Chinese*. Phd. dissertation, University of Pennsylvania.

Jackendoff, R. (1996). The proper treatment of measuring out, telicity, and perhaps even quantification in english. *Natural Language and Linguistic Theory 14*(2), 305–354.

Kamp, H. (1981). A theory of truth and semantic representation. In J. Groenendijk (Ed.), *Formal Methods in the Study of Language*. Amsterdam: Mathematical Center.

Kayne, R. (2015, June). English *one* and *ones* as complex determiners. New York University.

Kennedy, C. (2015). A "de-Fregean" semantics (and neo-gricean pragmatics) for modified and unmodified numerals. *Semantics & Pragmaticss 8*, 1–44.

Kratzer, A. (2007). On the plurality of verbs. In J. Dölling, T. Heyde-Zybatow, and M. Shafer (Eds.), *Event structures in linguistic form and interpretation*, pp. 269–300. Berlin: Mouton de Gruyter.

Krifka, M. (1989). Nominal reference, temporal constitution and quantification in event semantics. In R. Bartsh, J. Van Benthem, and P. van Emde Boas (Eds.), *Semantics and Contextual Expression*, Dordrecht, Netherlands, pp. 75–115. Foris.

Krifka, M. (1996). Parametrized sum individuals for plural anaphora. *Linguistics and Philosophy 19*(6), 555–598.

Krifka, M. (1998). The origins of telicity. In S. Rothstein (Ed.), *Events and Grammar*, pp. 197–236. Dordrecht: Kluwer Academic Publishers.

Kuhn, J. (2015). *Crosscategorial singular and plural references in sign language*. Ph. D. thesis, New York University.

Kuhn, J. (2017). Dependent indefinites: the view from sign language. *Journal of Semantics 34*, 407–446.

Landman, F. (2000). *Events and plurality: The Jerusalen Lectures*, Volume 76 of *Studies in Linguistics and Philosophy*. Dordrecht: Kluwer.

Lauer, S. (2009). Free relatives with *-ever*: Meaning and use. Ms. Stanford University.

Lauer, S. (2012). Some news on *Irgendein* and *algún*. Ms. Stanford University.

Li, H. and J. H.-K. Law (2016). At least some distributive operators aren't distributive operators. In *Proceedings of North East Linguistic Society 46*.

Link, G. (1983). The logical analysis of plurals and mass terms: A lattice-theoretical approach. In R. Baeuerle, C. Schwarze, and A. v. Stechow (Eds.), *Meaning, Use and Interpretation of Language*. DeGruyter.

Link, G. (1987). Generalized quantifiers and plurals. In P. Gärdenfors (Ed.), *Generalized Quantifiers. Linguistic and Logical Approaches*, Volume 31 of *Studies in Linguistics and Philosophy*, pp. 151–180. D. Reidel Publishing Company.

Link, G. (1998). *Algebraic Semantics in Language and Philosophy*. Stanford, California: CSLI Publications.

Lønning, J. T. (1987). Mass terms and quanti cation. *Linguistics and Philosophy 10*(1), 1–25.

Milačić, D., R. Singh, and I. Toivonen (2015). On the morphosyntactic representation of dependent quantification: Distance distributivity, dependent indefinites, and Skolemization. In E. Csipak and H. Zeijlstra (Eds.), *Proceedings Sinn und Bedeutung 19*, Göttingen, Germany., pp. 411–425.

Moltmann, F. (1991). On the syntax and semantics of binary distributive quantifiers. In *Proceedings of the North East Linguistic Society (NELS) 22*, pp. 279–292.

Nouwen, R. (2003). *Plural pronominal anaphora in context: Dynamic aspects of quantification*. Ph. D. thesis, Utrecht Univeristy, Utrecht, The Netherlands.

Perlmutter, D. M. (1970). On the article in English. In M. Bierwisch and K. E. Heidolph (Eds.), *Progress in Linguistics*, pp. 233–248. The Hague: Mouton.

Rett, J. (2014). The polysemy of measurement. *Lingua 143*, 242–266.

Safir, K. and T. Stowell (1988). Binominal *each*. In J. Blevins and J. Carter (Eds.), *Proceedings of North East Linguistic Society 8*, Toronto, pp. 426–450. GLSA.

Scha, R. (1981). Distributive, collective and cumulative quantification. In M. Stokhof and T. Janssen (Eds.), *Formal methods in the study of language*, pp. 483–512. Amsterdam: Mathematisch.

Schwarzschild, R. (1996). *Pluralities*. Dordrecht ; Boston: Kluwer Academic.

Schwarzschild, R. (2002). The grammar of measurement. In B. Jackson (Ed.), *Proceedings of SALT XII*, Cornell University, Ithaca, NY, pp. 225–245. CLC Publications.

Schwarzschild, R. (2006). The role of dimensions in the syntax of noun phrases. *Syntax 9*(1), 67–110.

Solomon, M. (2011). True distributivity and the functional interpretation of indefinites. Unpublished manuscript.

Solt, S. (2015). Q-adjectives and the semantics of quantity. *Journal of Semantics 32*, 221–273.

Stowell, T. (2013). Binominal *Each*: A dp that may not be. In D. Gil, S. Harlow, and G. Tsoulas (Eds.), *Strategies of quantification*, Oxford, pp. 260–313. Oxford University Press.

Sutton, M. (1993). Binominal *each*. Master's thesis, UCLA.

Szabolcsi, A. (2010). *Quantification*. Cambridge University Press.

van den Berg, M. (1996). *Some aspects of the internal structure of discourse*. Ph. D. thesis, University of Amsterdam, Amsterdam.

Wellwood, A. (2015). On the semantics of comparison across categories. *Linguistics and Philosophy 38*, 67–101.

Zhang, L. (2013). A ratio analysis of binominal each. In *The 3rd Mid-Atlantic Colloquium of Studies in Meaning (MACSIM 3)*.

Zimmermann, M. (2002). *Boys buying two sausages each: On the syntax and semantics of distance-distributivity*. Phd, University of Amsterdam.