Polynomial Event Semantics
Non-Montagovian Proper Treatment of Quantifiers

Oleg Kiselyov

Tohoku University, Japan
oleg@okmij.org

Abstract. We propose a simple extension of event semantics that naturally supports the compositional treatment of quantification. Our analyses require neither quantifier raising or other syntactic movements, nor type-lifting. Denotations are computed strictly compositionally, from lexical entries up, and quantifiers are analyzed in situ. We account for the universal, existential and counting quantification and the related distributive coordination, with the attendant quantifier ambiguity phenomena. The underlying machinery is not of lambda-calculus but of much simpler relational algebra, with straightforward set-theoretic interpretation.
The source of quantifier ambiguity in our approach lies in two possible analyses for the existential (and counting) quantification. Their inherent ambiguity however becomes apparent only in the presence of another, non-existential quantification.

1 Introduction

The recent paper [8] reported an application of so-called transformational semantics to textual inference within the FraCaS bank [4] – actually, only within the generalized quantifier section of FraCaS. It was left to future work to extend the approach to event semantics, so to handle tense and aspect. The present paper clears one theoretical hurdle on the road to such extension.

The major and immediate hurdle is the compositional treatment of quantifiers in event semantics, which is a well-known thorny problem: see, for example, [3, 6]. The latter paper also describes two recent solutions, in the tradition of Montagovian semantics. We present an alternative, non-Montagovian treatment. Besides metatheoretic preferences, we are motivated by the goal of solving entailment problems (at first, in FraCaS) completely automatically. We hence aim not just at presenting an analysis of various quantification phenomena. Our goal is to develop a mechanical procedure, an algorithm, of hopefully low-complexity, to obtain the meaning of a sentence from its surface (or, at least, treebank-annotated)

form *without any human intervention*, without any fuzzing and ad hoc adjustments.

A characteristic of [8] is the use of a first-order theorem prover to decide entailments. The meaning of sentences had to be described by first-order formulas. The careful analysis of the generated formulas shown that they fall within a subset of first-order logic, and could in fact be represented in Description Logic (DL) [9]. DL has roots in databases and relational algebra rather than lambda-calculus, and has straightforward set-theoretic semantics (see §5 for more discussion). Thus the motivation for the present work is analyzing complex quantification phenomena within event semantics taking inspiration from DL.

Our contribution is the compositional, easily mechanizable, non-Montagovian treatment of quantified NP and adverbial phrases. In contrast to syntactic approaches – movements, raising, transformations – ours is purely semantic, based on the construction of a suitable semantic domain. We use no continuations, no monads, no lambda-calculus, and, in fact, no variables. Rather, we rely merely on sets, relations and simple algebra. We arrive at the event semantics with all of its benefits, and account for the universal, existential and counting quantification and the attendant quantifier ambiguities. Quantifiers are analyzed in situ.

In this first paper on this topic we only deal with positive polarity phrases; however, §4.3 briefly discusses handling negative polarity.

The analyses of all the examples have been mechanically verified. The accompanying source code presents the model calculations in full, and includes more examples. It is available at `http://okmij.org/ftp/gengo/poly-event/`.

2 Classical Event Semantics

First, we recall the 'classical' (Davidsonian) event semantics, albeit in a different notation inspired by DL [9].

Our semantic domain is comprised of *individuals* such as john and bM, of *concepts* such as $\mathsf{Student}$, and of *roles* such as subj'. Individuals, identified by names, refer to entities in the domain of discourse: people, things, moments of time – and also events. Event names, such as bM, are meant to be suggestive, see Fig.1 for the key. Concepts denote properties of individuals (that is, refer to sets of entities); concept names are always capitalized. Roles are binary relations – specifically, relations of events to individuals, which may also be events. Role names are in lower-case and end in an apostrophe. Unless stated otherwise, roles are functional

relations. Fig.1 shows the sample domain to be used in running examples.

Individuals	students, classes, days of week, events
Concepts	Student : {bill , john , seth} Cut : events bM through sF Class : {peMo , peWd , peFr}
Roles	subj′, ob1′ as in the table below

event	subj	obj	event	subj	obj
bM	bill	peMo	jM	john	peMo
bW	bill	peWd	sW	seth	peWd
bF	bill	peFr	sF	seth	peFr

Fig. 1. Sample domain of student cutting classes

Just as in [8], our input are sentences annotated in the Penn Historical Corpora system (extensively used in [2]), such as

(1) `(IP-MAT (NP-SBJ (NPR Bill)) (VBD cut) (NP-OB1 (NPR PEMo)))`

(2)
```
(IP-MAT (NP-SBJ (D A) (N student))
        (VBD cut) (NP-OB1 (Q every) (N class)))
```

Erasing the annotations gives the original plain-text sentence: "Bill cut PEMo" for (1) (where 'PEMo' is the abbreviation for 'physical education class on Monday') and "A student cut every class" for (2).

The meaning of (1) can be expressed by the intersection of three concepts:

$$\mathsf{subj'/bill} \sqcap \mathsf{Cut} \sqcap \mathsf{ob1'/peMo} \tag{3}$$

Here $\mathsf{subj'}/x$ stands for the concept denoting the set of events that are related by the role $\mathsf{subj'}$ to the individual x: $\{y \mid \mathsf{subj'}\,(y, x)\}$. That is, $\mathsf{subj'/bill}$ means the events in which Bill is the subject. The notation extends to concepts: $\mathsf{subj'/C}$ denotes the set of events whose subjects are characterized by C. Therefore, $\mathsf{subj'/bill}$ can also be written as $\mathsf{subj'}\,/\,\{\mathsf{bill}\}$ where $\{\mathsf{bill}\}$ is the singular concept, whose sole instance is the named individual. The concept Cut refers to events whose action is cutting classes. The third term of (3) is similar to the first one; it is the concept denoting events whose object is the PEMo class. The whole sentence is characterized by the intersection of the three concepts.

Formally, we call a concept (or the concept formula) like (3) a denotation of the corresponding sentence, (1) in our case. A non-empty set of events that have the property specified by the concept is a *model* for the concept. In our sample domain (world), the model of (3) is the singleton $\{\mathsf{bM}\}$. If a sentence denotation does not have a model, then the sentence is 'false': incompatible with the record of events in the world in question.

One may think of a model as the evidence why the sentence is 'true'. The sentence (1) is true in the sample world because of the event bM that has transpired there. This point of view turns out illuminating when contemplating models of sentences with quantifiers, see §3.

One must have noticed how closely the denotation formula (3) corresponds to the structure of its sentence, (1). The correspondence will be formally defined in §4. The approach easily extends to adverbs (e.g., "deliberately" – whose denotation, Deliberately, denotes events whose action is done deliberately), temporal relations, etc. It does stumble, however, on quantification.

3 Poly-concepts

Our goal is to analyze sentences with quantifiers, such as

(4) Bill cut every class

or, annotated

(5) `(IP-MAT (NP-SBJ (NPR Bill)) (VBD cut) (NP-OB1 (Q every) (N class)))`

just as straightforwardly as we did (1). That is not easy, however. To account for quantification, this section refines concepts to poly-concepts, whose models are sets with some structure.

The trouble with quantification begins when trying to formulate the concept that would describe `(NP-OB1 (Q every) (N class))`. It cannot be $\mathsf{ob1}'/\mathsf{Class}$, because that concept admits a mere singleton $\{\mathsf{bM}\}$ model: an event whose object is *a* class. On the other hand, an event whose object is all classes, besides physically implausible, would give too narrow interpretation of (4): After all, the sentence does not assert that Bill cut all the classes in 'one shot': the class-cutting may have been spread over time.

Let us step back and consider what should be the evidence for (4) in our sample world. It should be the events of Bill cutting the Physical Education class on Monday, Wednesday and Friday (which are all classes

in our world). These events taken *together* is the evidence for (4). We call such a set of events a *group*, and write in angular brackets, for example: $\langle \mathsf{bM}\,,\,\mathsf{bW}\,,\,\mathsf{bF}\rangle$. The events in a group have no particular order, temporal or causal connection – but they are all regarded as a part of a single collective of events. Thus our intuition is that sentences with quantifiers are statements about groups of events.

However, groups alone are not enough to give denotations to all quantifier phrases (QNP). Consider

(6) Bill cut two classes.

(which we take to mean that Bill cut at least two classes). We may cite the group $\langle \mathsf{bM}\,,\,\mathsf{bW}\rangle$ as the evidence for (6) – or the group $\langle \mathsf{bW}\,,\,\mathsf{bF}\rangle$, or $\langle \mathsf{bM}\,,\,\mathsf{bF}\rangle$. We call a set of groups each of which could be the evidence a *factor*, notated as follows (for our example):

$$\lceil \langle \mathsf{bM}\,,\,\mathsf{bW}\rangle\ \langle \mathsf{bW}\,,\,\mathsf{bF}\rangle\ \langle \mathsf{bM}\,,\,\mathsf{bF}\rangle \rceil$$

The singleton set containing this factor is a model for (6). We may distribute groups to factors in a different way:

$$\{\lceil \langle \mathsf{bM}\,,\,\mathsf{bW}\rangle \rceil, \lceil \langle \mathsf{bW}\,,\,\mathsf{bF}\rangle \rceil, \lceil \langle \mathsf{bM}\,,\,\mathsf{bF}\rangle \rceil\}$$

This is also a model for (6), now with three alternative factors. A model is hence a set of alternative factors; once we pick one alternative ('external choice') we get a factor that contains one or more groups ('internal choice') each of which may be used as the evidence. One can see a close connection to 'alternative semantics', as we briefly discuss in §5. Later we will see that these 'external' and 'internal' choices are closely connected to the quantifier scope.

We thus generalize concepts to poly-concepts. Whereas a concept describes a property of individuals/individual events, a poly-concept describes a property of groups of individuals, with alternatives. For example, whereas $\mathsf{Student}$ denotes students, the poly-concept 'two students' describes groups of two students and can be used to give the meaning to 'Two students cut a class': there is a group of two students each of which cut a class (in fact, there is more than one such group to choose from). It should be clear that we take group 'loosely': we do not insists, for example, that the two students cut the class together. (Tight groups are better represented as particular individuals).

Poly-concepts are built from ordinary concepts with the $\mathcal{P}$ operation, and from the existing poly-concepts using union, intersection, the group formation ('multiplication'), and the flattening $\mathcal{N}$, as formally shown in Fig.2. The empty poly-concept, just like the empty concept, is denoted by $\bot$.

Concept	c	
Poly-concept	x, y	$::= \quad \bot \mid \mathcal{P}c \mid \mathcal{N}x \mid x \sqcup y \mid x \sqcap y \mid x \otimes y$

Fig. 2. Poly-concepts: Syntax

Set-theoretically, a poly-concept is a set of factors; a factor is a set of groups, and a group is a set of entities. The meta-variables used to refer to groups, factors and poly-concepts, and the corresponding notation are collected below:

Individuals	i			
Concept	c	set of individuals	$\{i_1, \ldots, i_n\}$	for some $n \geq 0$
Group	g	set of individuals	$\langle i_1, \ldots, i_m \rangle$	for some $m \geq 1$
Factor	d	set of groups	$\lceil g_1 \ \ldots \ g_n \rceil$	for some $m \geq 0$
Poly-concept	x, y	set of factors	$\{d_1, \ldots, d_n\}$	for some $n \geq 0$

Groups are always non-empty. Although empty factors can come up during calculations, they are not included in a poly-concept. A poly-concept hence is a set of non-empty factors. The empty poly-concept $\bot$ is the empty set of factors. All groups within one factor have the same cardinality. We write $|d|$ for the cardinality of groups in the factor d.

$$
\begin{array}{ll}
\mathcal{P}c & = \{\lceil\langle i\rangle\rceil \mid i \in c\} \\
\mathcal{N}x & = \bigcup_{d \in x} d \\
x \sqcup y & = x \cup y \\
x \otimes y & = \{d_1 \otimes d_2 \mid d_1 \in x, d_2 \in y\} \\
d_1 \otimes d_2 & = \{g_1 \cup g_2 \mid g_1 \in d_1, g_2 \in d_2, g_1 \cap g_2 = \varnothing\} \\
x^n & = x \otimes x \ldots \otimes x \qquad n\text{-times multiplication} \\
x \sqcap y & = \{d_1 \sqcap d_2 \mid d_1 \in x, d_2 \in y\} \\
d_1 \sqcap d_2 & = d_1^{|d_2|} \ \cap \ d_2^{|d_1|}
\end{array}
$$

Fig. 3. Poly-concepts: Set-theoretic semantics. Empty factors are always suppressed when forming poly-concepts.

The (set-theoretical) meaning of the poly-concept operations is defined in Fig.3. $\mathcal{P}c$ lifts a concept c to a poly-concept by turning each element of c into its own group, which are collected in the single factor. Empty factors are always suppressed when forming poly-concepts; therefore, $\mathcal{P}\bot$ is $\bot$, the empty set of factors. As another example, $\mathcal{P}$ Student

is the poly-concept $\{\lceil\langle \mathsf{bill}\rangle\ \langle \mathsf{john}\rangle\ \langle \mathsf{seth}\rangle\rceil\}$. Flattening (or narrowing) $\mathcal{N}x$ joins all factors of x into one. The poly-concept union $x \sqcup y$ is the mere set-union of x and y regarded as sets of factors.

The poly-concept multiplication $x \otimes y$ and intersection $x \sqcap y$ are interpreted as the multiplication (resp. intersection) of each factor of x with each factor of y, dropping the empty factors. Factor multiplication $d_1 \otimes d_2$ is almost as straightforward: the pairwise union of d_1's and d_2's groups – provided the groups are disjoint. Thus the result of the multiplication has bigger groups; in fact

$$|d_1 \otimes d_2| = |d_1| + |d_2| \tag{7}$$

The disjointness condition is subtle. The table below shows the result of exponentiating $\mathcal{P}\mathsf{Student}$ (i.e., multiplying with itself several times), in our sample domain:

$(\mathcal{P}\ \mathsf{Student})^1$	$\{\lceil\langle \mathsf{bill}\rangle\ \langle \mathsf{john}\rangle\ \langle \mathsf{seth}\rangle\rceil\}$
$(\mathcal{P}\ \mathsf{Student})^2$	$\{\lceil\langle \mathsf{bill}\ ,\ \mathsf{john}\rangle\ \langle \mathsf{john}\ ,\ \mathsf{seth}\rangle\ \langle \mathsf{bill}\ ,\ \mathsf{seth}\rangle\rceil\}$
$(\mathcal{P}\ \mathsf{Student})^3$	$\{\lceil\langle \mathsf{bill}\ ,\ \mathsf{john}\ ,\ \mathsf{seth}\rangle\rceil\}$
$(\mathcal{P}\ \mathsf{Student})^4$	$\bot$

In general, if c is a concept with n entities $\{i_1, \ldots, i_n\}$, then

$$\underbrace{\mathcal{P}c \otimes \mathcal{P}c \otimes \cdots \otimes \mathcal{P}c}_{m} = \begin{cases} \{\lceil\langle i_1,\ldots,i_n\rangle\rceil\} & \text{if } m = n \\ \bot & \text{if } m > n \end{cases}$$

The factor intersection $d_1 \sqcap d_2$ is even more subtle: $d_1^{|d_2|} \cap d_2^{|d_1|}$, that is, intersecting exponentiated factors. From (7) we see that the factors to intersect have the same cardinality:

$$|d_1^{|d_2|}| = |d_2^{|d_1|}| = |d_1|\,|d_2|$$

The reason the factor intersection is so complex will become clear in the next section.

4 Compositional Semantics: from a Sentence to a Poly-concept

We now describe how the poly-concept that represents the meaning of a sentence is built up, compositionally, from the concepts of lexical entries up to the tree root. Roughly, (non-functional) lexical entries contribute concepts: common nouns and adjectives are properties of individuals; verbs and adverbs are properties of the transpired events. The

concepts are lifted to poly-concepts with the $\mathcal{P}$ operation. Adjoining nodes intersects the corresponding poly-concepts.

To be more precise, consider the following (simplified) grammar for the treebank annotated sentences, which we take as our input. We disregard tense and aspect and gloss plurality, which are to be dealt with in the future work.

clause	::=	$\text{node}_1 \ldots \text{node}_n$
node	::=	NP-SBJ np | NP-OB1 np | VB verb | ADV adverb | pp
np	::=	det $\text{nom}_1 \ldots \text{nom}_n$ | proper-noun
nom	::=	common-noun | pp | adj
pp	::=	PP preposition np

The poly-concept describing a clause is the intersection of poly-concepts for each node. The poly-concept for (VB verb) is the concept for the verb (the set of events where the verb action took place), extended to the poly-concept with $\mathcal{P}$. Adverbs are similar. Nominals nom are described by poly-concepts, or simple concepts (for common-noun and adjectives) subsequently lifted. The poly-concept for a sequence of nominals is the intersection of poly-concepts for the members of the sequence.

What is left to define are poly-concepts for nominal phrases. NPs always appear in some role, such as NP-SUBJ, NP-OB1, or the preposition-role. Therefore, we define poly-concepts not for NPs per se but for an NP in a role. The definitions are uniform in the treatment of roles; we take subj' for concreteness:

(8)	Proper noun	$\mathcal{P}(\mathsf{subj}' \,/\, \{\mathsf{properNoun}\})$
(9)	"at least" k nom	$\bigcup_{s\subset\mathsf{CN},\lvert s\rvert=k} \prod_{i\in s} \mathcal{P}(\mathsf{subj}' \,/\, \{\mathsf{i}\})$
(10)	"at least" k nom	$\mathcal{N}x$ where x is from (9)
(11)	"an" nom	$\bigcup_{i\in\mathsf{CN}} \mathcal{P}(\mathsf{subj}' \,/\, \{\mathsf{i}\})$
(12)	"an" nom	$\mathcal{P}(\mathsf{subj}'/\mathsf{CN})$
(13)	"every" nom	$\prod_{i\in\mathsf{CN}} \mathcal{P}(\mathsf{subj}' \,/\, \{\mathsf{i}\})$

Here, {ProperNoun} is the concept representing the proper noun in question, and CN is the concept for the nominal nom. There are two alternative definitions for existential and counting quantifiers: they are inherently ambiguous in our approach. (The number quantification "exactly n" and "at most n" also carry negative polarity, describing events that should not take place. Negative polarity is not considered in the present paper.)

For the starting example (1) from §2, reproduced below

(14) `(IP-MAT (NP-SBJ (NPR Bill)) (VBD cut) (NP-OB1 (NPR PEMo)))`

we obtain, following the just given definitions, the poly-concept denotation

$$\mathcal{P}(\mathsf{subj}' \,/\, \{\mathsf{bill}\}) \sqcap (\mathcal{P}\ \mathsf{Cut} \sqcap \mathcal{P}(\mathsf{ob1}' \,/\, \{\mathsf{peMo}\})) \tag{15}$$

which is the $\mathcal{P}$-lifted concept denotation (3) described in §2.

We are now in a position to compute the denotation of quantified phrases, in particular, (5) (repeated below), used as the motivation in §3:

(16) `(IP-MAT (NP-SBJ(NPR Bill)) (VBD cut) (NP-OB1 (Q every)(N class)))`

The poly-concept denotation is:

$$\mathcal{P}(\mathsf{subj}' \,/\, \{\mathsf{bill}\}) \sqcap (\mathcal{P}\ \mathsf{Cut} \sqcap \prod\nolimits_{i \in \mathsf{Class}} \mathcal{P}(\mathsf{ob1}' \,/\, \{\mathsf{i}\})) \tag{17}$$

To evaluate this denotation in our sample world, we compute, following Fig.3:

$\mathcal{P}(\mathsf{subj}' \,/\, \{\mathsf{bill}\})$:

(18) $\{\lceil \langle \mathsf{bM} \rangle\ \langle \mathsf{bW} \rangle\ \langle \mathsf{bF} \rangle\ \rceil\}$

$\mathcal{P}(\mathsf{ob1}' \,/\, \{\mathsf{peMo}\}) \otimes \mathcal{P}(\mathsf{ob1}' \,/\, \{\mathsf{peWd}\}) \otimes \mathcal{P}(\mathsf{ob1}' \,/\, \{\mathsf{peFr}\})$:

(19) $\{\lceil \langle \mathsf{bM}, \mathsf{bW}, \mathsf{bF} \rangle\ \langle \mathsf{bM}, \mathsf{bW}, \mathsf{sF} \rangle\ \langle \mathsf{bM}, \mathsf{sW}, \mathsf{bF} \rangle\ \langle \mathsf{bM}, \mathsf{sW}, \mathsf{sF} \rangle$
$\langle \mathsf{jM}, \mathsf{bW}, \mathsf{bF} \rangle\ \langle \mathsf{jM}, \mathsf{bW}, \mathsf{sF} \rangle\ \langle \mathsf{jM}, \mathsf{sW}, \mathsf{bF} \rangle\ \langle \mathsf{jM}, \mathsf{sW}, \mathsf{sF} \rangle \rceil\}$

Taking the poly-concept intersection, we obtain the model for the entire denotation (17):

$$\{\lceil \langle \mathsf{bM}, \mathsf{bW}, \mathsf{bF} \rangle \rceil\}$$

The model shows the events that justify the truth of "Bill cut every class" in our sample world. A similar calculation shows that "Every student cut every class" does not have a model in our world.

4.1 Quantifier Ambiguity

Since the existential and counting quantifiers can be analyzed in two different ways, ambiguity arises. Indeed, consider (20) (which is (2) repeated):

(20)
```
(IP-MAT (NP-SBJ (D A) (N student))
  (VBD cut) (NP-OB1 (Q every) (N class)))
```

for which we can derive either (21) or (22), depending on whether we use (11) or (12):

(21) $\bigcup_{i \in \mathsf{Student}} \mathcal{P}(\mathsf{subj}' \,/\, \{\mathsf{i}\}) \sqcap (\mathcal{P}\ \mathsf{Cut} \sqcap \prod_{i \in \mathsf{Class}} \mathcal{P}(\mathsf{ob1}' \,/\, \{\mathsf{i}\}))$

(22) $\mathcal{P}(\mathsf{subj}'/\mathsf{Student}) \sqcap (\mathcal{P}\ \mathsf{Cut} \sqcap \prod_{i \in \mathsf{Class}} \mathcal{P}(\mathsf{ob1}' \,/\, \{\mathsf{i}\}))$

In our sample world,

$$\begin{array}{l} \bigcup_{i \in \mathsf{Student}} \mathcal{P}(\mathsf{subj}' \,/\, \{\mathsf{i}\}) \text{ (“a student” according to (11)) :} \\ \quad \{\lceil\langle \mathsf{bM}\rangle\ \langle \mathsf{bW}\rangle\ \langle \mathsf{bF}\rangle\rceil, \lceil\langle \mathsf{jM}\rangle\rceil, \lceil\langle \mathsf{sW}\rangle\ \langle \mathsf{sF}\rangle\rceil\} \\ \mathcal{P}(\mathsf{subj}'/\mathsf{Student}) \text{ (“a student” according to (12)) :} \\ \quad \{\lceil\langle \mathsf{bM}\rangle\ \langle \mathsf{bW}\rangle\ \langle \mathsf{bF}\rangle\ \langle \mathsf{jM}\rangle\ \langle \mathsf{sW}\rangle\ \langle \mathsf{sF}\rangle\rceil\} \end{array}$$

Keeping in mind (19) as the denotation for “every class” we obtain for the whole (20)

$$\begin{array}{ll} \text{Model of (21) :} & \{\lceil\langle \mathsf{bM}\ ,\ \mathsf{bW}\ ,\ \mathsf{bF}\rangle\rceil\} \\ \text{Model of (22) :} & \text{the same as (19)} \end{array}$$

The former model demonstrates the linear reading of “a student cut every class”, with the existential taking the wide scope: the sentence is true in our world because there exists one particular student (namely, Bill) who skipped every class. In contrast, the denotation (21) corresponds to the narrow-scope reading of the existential. The model has many choices for the evidence: all three classes are cut, but by generally different students.

Sentences with several existential quantifiers also have several interpretations, for example, (23):

(23)
```
(IP-MAT (NP-SBJ (D a) (N Student))
  (VBD cut) (NP-OB1 (D a) (N class)))
```

Each of the two indefinite determiners can be analyzed either as (11) or (12), giving four possible poly-concepts for (23). They are all distinct, and have a model in our world:

$$\begin{array}{c} \{\lceil\langle \mathsf{bM}\rangle\ \langle \mathsf{bW}\rangle\ \langle \mathsf{bF}\rangle\ \langle \mathsf{jM}\rangle\ \langle \mathsf{sW}\rangle\ \langle \mathsf{sF}\rangle\rceil\} \\ \{\lceil\langle \mathsf{bM}\rangle\ \langle \mathsf{bW}\rangle\ \langle \mathsf{bF}\rangle\rceil, \lceil\langle \mathsf{jM}\rangle\rceil, \lceil\langle \mathsf{sW}\rangle\ \langle \mathsf{sF}\rangle\rceil\} \\ \{\lceil\langle \mathsf{bM}\rangle\ \langle \mathsf{jM}\rangle\rceil, \lceil\langle \mathsf{bW}\rangle\ \langle \mathsf{sW}\rangle\rceil, \lceil\langle \mathsf{bF}\rangle\ \langle \mathsf{sF}\rangle\rceil\} \\ \{\lceil\langle \mathsf{bM}\rangle\rceil, \lceil\langle \mathsf{bW}\rangle\rceil, \lceil\langle \mathsf{bF}\rangle\rceil, \lceil\langle \mathsf{jM}\rangle\rceil, \lceil\langle \mathsf{sW}\rangle\rceil, \lceil\langle \mathsf{sF}\rangle\rceil\} \end{array}$$

It is easy to see that the four denotation are equivalent: if one has a model, so are the others. Therefore, (23) is not really ambiguous.

Quantified adverbial modifiers like "everyday" and quantified adverbial phrases are analyzed similarly to NP-SBJ and NP-OB1 phrases. Like the latter, adverbials also describe the set of events, which occur within some time moments or places.

4.2 Counting Quantification and Ambiguity

Like existential, counting quantification can also be analyzed in two different ways, giving rise to ambiguity. Indeed, consider "Two students cut every class". Similarly to the calculations above, we obtain two denotations; one of them has the model

$$\{\lceil\langle \mathsf{bM}\,,\mathsf{bW}\,,\mathsf{bF}\,,\mathsf{jM}\,,\mathsf{sW}\,,\mathsf{sF}\rangle\rceil\}$$

and the other does not, demonstrating the two readings of the sentence neither of which entails the other.

4.3 Negative-Polarity Phrases

Our approach is easily extensible to negative-polarity quantifiers such as 'no', adverbs such as 'never', and also quantifiers such as 'at most' and 'exactly'. So far, we have been computing a poly-concept that describes events that justify the sentence in question (provide the model for the sentence) – a group of events which, if occur, would make the sentence true. To deal with negative polarity, we also should compute *false conditions* – events which, if occur, will falsify the sentence. The false conditions are computed just as compositionally as truth conditions.

5 Related Work

One inspiration for this work comes from Description Logics (DL), which are subsets of C2 (first-order logic with two variables and counting quantifiers) developed for the task of knowledge representation. DL can be traced to databases and relational algebra. DL exploits the fact that the two variables in C2 formulas can be kept implicit and do not have to be named, which eliminates the whole class of problems inherent in lambda-calculus, regarding alpha-conversion, substitution and binding. The variable-free nature of DL and its roots in knowledge representation offer a different, arguably, more linguistically intuitive perspective than Montagovian semantics. Also important is that DL are developed to be

decidable, and easily. The decidability/complexity of various DL are thoroughly investigated, resulting in practical decision procedures and highly optimized implementations. We refer to DL Primer [9] and the tutorial [1] for good introduction.

DL have certainly used before for computational linguistics/NLP – for example, [7], but not for theoretical linguistics, to my knowledge. The NLP applications are either at hoc or "best-effort" (or both) – neither of which is a problem for NLP since compositional treatment and building a semantic theory are not the goals there. I have not seen using DL as an alternative to Montagovian semantics, specifically, PTQ.

Our work, especially the earlier [8], have much in common with the work of Tian, Miyao et al. on Dependency-based Compositional Semantics (DCS) [5, 11]. The similarity with [8] is using relation algebra semantics and representing the properties of generalized quantifiers as axioms. We share the observation that our semantic representations are essentially DL. Unlike [5], we had no problems with quantifiers like 'few', downward monotone on the first argument. The characteristic feature of the present paper is the explicit use of event semantics.

Our main difference from [5, 11] is methodological: we are interested in theoretical semantics rather than NLP. Therefore, we have no use for approximately paraphrasing sentences, word sense sumilarity and other NLP techniques. The methodological difference leads to many technical differences. First, whereas Tian et al. semantics is coarse, ours is 'hyperfine': true sentences have distinct denotations. Therefore, the model of our denotations can be used as the evidence for the truth of the sentences. Another distinction is our use of event semantics, and the aim to resolve problems of quantification in event semantics.

Our idea of alternative factors and alternative evidence is closely related to the alternative semantics [10]. For example, our $\mathcal{N}x$ operator also occurs in the alternative semantics.

Unlike Champollion [3] we do not try to combine Montagovian treatment of quantifiers with event semantics; we investigate the alternative to the Montagovian treatment instead.

6 Conclusions

We have outlined yet another proper treatment of quantification – this time, with no lifting, lambda calculus or even variables. Nevertheless, we are able to analyze quantifier scope (for positive polarity phrases, at the

moment), quantifier ambiguity. Our semantics has straightforward set-theoretic interpetation: the models or denotations are triple-nested sets.

The future work is to fully develop the treatment of negation, only briefly hinted at in the present paper. Another item is the treatment of tense and aspect. It is intriguing to explore connections with collective readings of quantifiers.

Acknowledgments I am grateful to anonymous reviewers for very helpful comments and the pointers to related work. I thank Yu Izumi for an inspiring conversation.

Bibliography

[1] Baader, F.: Description logics. In: Reasoning Web: Semantic Technologies for Information Systems, 5th International Summer School 2009, Lecture Notes in Computer Science, vol. 5689, pp. 1–39. Springer–Verlag (2009)

[2] Butler, A.: The treebank semantics parsed corpus. `http://www.compling.jp/tspc/` (2017)

[3] Champollion, L.: The interaction of compositional semantics and event semantics. Linguistics and Philosophy **38**(1), 31–66 (2015)

[4] Cooper, R., Crouch, D., van Eijck, J., Fox, C., van Genabith, J., Jan, J., Kamp, H., Milward, D., Pinkal, M., Poesio, M., Pulman, S., Briscoe, T., Maier, H., Konrad, K.: Using the framework. Deliverable D16, FraCaS Project (1996)

[5] Dong, Y., Tian, R., Miyao, Y.: Encoding generalized quantifiers in dependency-based compositional semantics. In: Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation, PACLIC 28, Cape Panwa Hotel, Phuket, Thailand, December 12-14, 2014. pp. 585–594 (2014), `http://aclweb.org/anthology/Y/Y14/Y14-1067.pdf`

[6] de Groote, P., Winter, Y.: A type-logical account of quantification in event semantics. In: JSAI-isAI Workshops. Lecture Notes in Computer Science, vol. 9067, pp. 53–65. Springer (2014), `\url{https://hal.inria.fr/hal-01102261}`

[7] Gyawali, B., Shimorina, A., Gardent, C., Cruz-Lara, S., Mahfoudh, M.: Mapping natural language to description logic. In: The Semantic Web. pp. 273–288. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-58068-5_17

[8] Kiselyov, O.: Transformational semantics on a tree bank. In: New Frontiers in Artificial Intelligence. pp. 241–252. Springer International Publishing (2018). https://doi.org/10.1007/978-3-319-93794-6_17

[9] Krötzsch, M., Simancik, F., Horrocks, I.: A description logic primer. CoRR **abs/1201.4089** (2013)

[10] Rooth, M.: Alternative semantics. Oxford University Press (02 2016). https://doi.org/10.1093/oxfordhb/9780199642670.013.19

[11] Tian, R., Miyao, Y., Matsuzaki, T.: Logical inference on dependency-based compositional semantics. In: ACL (1). pp. 79–89. The As-

sociation for Computer Linguistics (2014), http://aclweb.org/anthology/P/P14/