Computing Scope Dominance with Upward Monotone Quantifiers

Alon Altman
Technion

Yoad Winter
Technion/UiL OTS

Abstract

This paper describes an algorithm that characterizes logical relations between different interpretations of scopally ambiguous sentences. The proposed method uses general properties of natural language determiners in order to generate a model which is indicative of such entailment relations. The computation of this model involves information about cardinalities of noun denotations and containment relations between them, which often affect entailment relations with quantifiers. After proving the correctness of the proposed method, the paper briefly describes a demo implementation that illustrates its main results.

1 Introduction

Since its introduction into linguistic research in the early 1980s, the theory of Generalized Quantifiers has been pivotal to the formal semantics of natural language. This theory has allowed the expression of sound generalizations about quantification in natural language which are not expressible in first order logic. At the same time, the higher level of expressibility in GQ theory causes undecidability problems for inference in computational semantics. Such problems, which are perhaps theoretically unavoidable given the higher-order semantics of natural language, can be at least partly tackled by studying a limited variety of inference patterns that are of special interest for computational needs. One example for such patterns is what we henceforth refer to as *scope dominance* — a situation in which one reading of a scopally ambiguous sentence entails another reading. A familiar case of such relations is in simple transitive sentences where the subject and object both denote existential or universal quantifiers, as in the following example.

(1) A priest visited every church.

Here the object narrow scope reading (the $\exists\forall$ order of the quantifiers) entails the object wide scope reading (the $\forall\exists$ order). We describe this situation by saying that an existential quantifier is scopally *dominant* over a universal quantifier (but not the other way around). This is of course a simple fact of first order logic, but the entailment pattern it exemplifies is quite general, and appears in any case where an *upward monotone* noun phrase replaces either the subject or the object in (1) (but not both of them). This includes NPs like *more than half of the churches* or *infinitely many priests*, whose semantics is not first order definable.

The computation of scope dominance is not only a significant challenge for the usability of generalized quantifier theory for limited cases of inference in natural language. A procedure that fully computes scope dominance relations in a substantial part of natural language would reduce, at least for some purposes, the extent of the ambiguity that we need to represent in scopally ambiguous sentences. For instance, in sentence (1) only the object wide scope reading — the weaker of the two readings — is relevant if we want to make sure that any reading of the ambiguous sentence is entailed by some given statement. Conversely, to test if an ambiguous use of sentence (1) is definitely refuted by a given statement, or, alternatively, if the sentence *entails* a given statement, we can use only the narrow scope reading of (1).

A partial characterization of scope dominance relations with upward monotone determiners over finite domains appears in Westerståhl (1986), which has been recently extended in Altman et al. (2002) for all upward monotone generalized quantifiers over countable domains. In this paper we show that this characterization of scope dominance can be applied "globally" across models to identify valid inferences that involve scope dominance. This involves three non-trivial phenomena:

- *Noun phrase coordination*: For simple noun phrases of the form *Determiner-Nominal*, the semantic properties of the noun phrase that affect scope dominance can be computed directly from lexical syntactic features of the determiner. However, computing the semantic properties of NP coordinations involves more intricate procedures because of possible relations between the denotations of the NP conjuncts.

- *Cardinality information*: In everyday use, common nouns often come with information about the cardinality of their set denotation. Such cardinality information affects the semantic properties of quantified NPs. For instance, knowing that there are exactly three people in the room, we can deduce that the noun phrase *more than two people in the room* denotes a universal quantifier in the given situation, equivalent to *every person in the room*. Such universal quantifiers take part in scope dominance relations than do not appear without the additional cardinality information.

- *Containment relations*: Many nominals, common nouns as well as proper names, are connected to each other in "super-concept" or *containment* relations. Such semantic relations between nominals, as for example between the nouns *student* and *person*, impose further cardinality restrictions on their possible denotations. For instance, in the situation just described, with three people in the room, the noun phrase *more than two students in the room* denotes either an empty set (if there are less than three students in the room) or a universal quantifier (in case the three people in the room are all students). This of course affects the scope dominance properties of the noun phrase.

The method that is developed in this paper takes these factors into account by computing a model that is *indicative* of scope dominance in simple transitive sentences. The generation of this model relies on a novel observation that natural language determiners are *downward consistent* with respect to certain semantic properties. A determiner D is called *downward consistent* with respect to a property $\mathcal{F}$ of generalized quantifiers if for all sets $A \subseteq B$: if the generalized quantifier $D(B)$ has property $\mathcal{F}$, the quantifier $D(A)$ has property $\mathcal{F}$ as well. It is hypothesized that all upward monotone determiners in natural language are downward consistent with respect to the properties of generalized quantifiers that affect scope dominance. Consequently, for any $A \subseteq B$: if $D(B)$ is scopally dominant over a quantifier Q (or Q is dominant over $D(B)$), then $D(A)$ is scopally dominant over Q as well (Q is dominant over $D(A)$, respectively). Together with the familiar *conservativity*, *extension* and *isomorphism invariance* properties of natural language determiners, it follows that there is a minimal cardinality for A such that $D(A)$ exhibits no more scope dominance relations than $D(B)$, for any countable B. This in turn guarantees that under certain limitations, simple transitive sentences with NP coordinations can be effectively assigned a model that is indicative of scope dominance in the sentence: the quantifiers in this model exhibit scope dominance if and only if there is an entailment between the two readings of the sentence.

Section 2 introduces some background on generalized quantifiers, and the problem of scope dominance in particular. Section 3 introduces the proposed algorithm for computing scope dominance with upward monotone quantifiers. Section 4 makes some remarks about the implementation of this algorithm in a working demo, which is available at `http://www.yeda.cs.techion.ac.il/~alon_a/`. Section 5 concludes with a discussion of some remaining problems and prospects for further research on scope dominance in natural language.

2 Scope Dominance in Generalized Quantifier Theory

This section overviews the notions and facts from Generalized Quantifier Theory (GQT) that are essential for the proposed method of computing scope dominance. After reviewing some familiar basics of GQT, we mention without proof the main results in Altman et al.'s (2002) characterization of scope dominance. This allows us to define the notion of *downward consistency* and introduce the specification of natural language determiners that will be used by the algorithm of Section 3.

2.1 Generalized quantifiers

In GQT,[1] noun phrases such as *some student, every teacher* and *at least half of the children* each denote a subset of $\wp(E)$, where E is an arbitrary non-empty domain, and $\wp(E)$ is its powerset. Such sets are called *generalized quantifiers* (GQs) *over* E. Determiner expressions such as *some, every* and *at least half* (*of*) denote *determiner functions*. A determiner function D_E over E is a function from $\wp(E)$ to $\wp(\wp(E))$, or — equivalently modulo isomorphism — a two place relation over $\wp(E)$. If we standardly assume that nominal expressions such as *student*, *tall student*, or *student in the room* denote subsets of the domain E, then NPs such as *every* (*tall*) *student* (*in the room*) denote GQs over E, as required. We often sloppily use the notation 'D' instead of 'D_E' for a determiner function, when the domain E is understood from the context. Some examples of determiners with their denotations are given in Table 1. In this table the determiners above the line are treated as lexical items in the fragment that will be introduced in Section 3. Here and henceforth we use the notation $\mathbf{blik}'_E$ for the denotation of the word or phrase *blik* under the domain E. Note that the determiners *at least half* and *each of the five* are assumed to denote partial functions, which impose a *cardinality presupposition* on their first argument — finiteness in the first case, cardinality five in the second case. Such cardinality requirements will play a crucial role in the computation of scope dominance.

In this paper, we limit ourselves to *upward monotone* quantifiers. Standardly, a quantifier Q over E is called upward monotone if it is closed under supersets: for all $A \subseteq B \subseteq E$, $A \in Q$ entails $B \in Q$. A determiner D over a domain E is called *right upward monotone* if for all $A \in E$, the quantifier $D(A)$ is upward monotone if it is defined. We henceforth refer to (right) upward monotone quantifiers and determiners simply as *monotone*. Table 1 gives examples of both monotone and non-monotone determiners. Note that the definition of monotone determiners ignores cases where they are undefined. Henceforth in this paper, we keep to this convention in the characterization of determiners and ignore sets that are outside their domain. For sake of briefness, this point will not be explicitly mentioned in the definitions.

[1] See Barwise and Cooper (1981), Van Benthem (1984), and Keenan and Stavi (1986) for three classic works on the application of GQT to linguistics. See Keenan and Westerståhl (1996) for a more up-to-date overview of the field.

Determiner	Denotation — for all $A \subseteq E$:	Mon.
every, each, all	$\mathbf{every}'_E(A) \stackrel{def}{=} \{B \subseteq E : A \subseteq B\}$	Yes
some	$\mathbf{some}'_E(A) \stackrel{def}{=} \{B \subseteq E : A \cap B \neq \emptyset\}$	Yes
at least *m*	$\mathbf{at\_least\_}m'_E(A) \stackrel{def}{=} \{B \subseteq E : \|A \cap B\| \geq m\}$	Yes
more than *m*	$\mathbf{more\_than\_}m'_E(A) \stackrel{def}{=} \{B \subseteq E : \|A \cap B\| > m\}$	Yes
at least half	$\mathbf{at\_least\_half}'_E(A) \stackrel{def}{=} \begin{cases} \{B \subseteq E : \|A \cap B\| \geq \|A \setminus B\|\} & A < \aleph_0 \\ \text{undefined} & \text{otherwise} \end{cases}$	Yes
more than half	$\mathbf{more\_than\_half}'_E(A) \stackrel{def}{=} \begin{cases} \{B \subseteq E : \|A \cap B\| > \|A \setminus B\|\} & A < \aleph_0 \\ \text{undefined} & \text{otherwise} \end{cases}$	Yes
all but at most *m*	$\mathbf{all\_but\_at\_most\_}m'_E(A) \stackrel{def}{=} \{B \subseteq E : \|A \setminus B\| \leq m\}$	Yes
infinitely many	$\mathbf{inf\_many}'_E(A) \stackrel{def}{=} \{B \subseteq E : \|A \cap B\| \geq \aleph_0\}$	Yes
all but finitely many	$\mathbf{all\_but\_fin\_many}'_E(A) \stackrel{def}{=} \{B \subseteq E : \|A \setminus B\| < \aleph_0\}$	Yes
each of the *m*	$\mathbf{each\_of\_the\_}m'_E(A) \stackrel{def}{=} \begin{cases} \{B \subseteq E : A \subseteq B\} & \|A\| = m \\ \text{undefined} & \text{otherwise} \end{cases}$	Yes
some and every	$\mathbf{some\_and\_every}'_E(A) \stackrel{def}{=} \{B \subseteq E : A \subseteq B \wedge \|A\| \geq 1\}$	Yes
some or every	$\mathbf{some\_or\_every}'_E(A) \stackrel{def}{=} \{B \subseteq E : A \cap B \neq \emptyset \vee A = \emptyset\}$	Yes
no	$\mathbf{no}'_E(A) \stackrel{def}{=} \{B \subseteq E : A \cap B = \emptyset\}$	No
exactly *m*	$\mathbf{exactly\_}m'_E(A) \stackrel{def}{=} \{B \subseteq E : \|A \cap B\| = m\}$	No

Table 1: Denotations of some determiner expressions

A quantifier $Q \subseteq \wp(E)$ is called *trivial* (TRIV) if $Q = \emptyset$ (Q is TRIV_0) or $Q = \wp(E)$ (Q is TRIV_1). For example, the noun phrases *at least zero students* and *less than zero students* denote trivial quantifiers ($\wp(E)$ and $\emptyset$ respectively) in any model.

A monotone quantifier $Q \subseteq \wp(E)$ is a *filter* (FLT) if it is closed under finite intersections. A filter $Q \subseteq \wp(E)$ is *principal* if it is closed under arbitrary (finite or infinite) intersections. For example, the noun phrase *every student* denotes a principal filter in any model. The noun phrase *all but finitely many students* denotes a non-principal filter whenever the denotation of the noun *students* is infinite. A filter $Q \subseteq \wp(E)$ is called an *ultrafilter* if for all $A \subseteq E$, either $A \in Q$ or $E \setminus A \in Q$. A quantifier Q is therefore a *principal ultrafilter* (i.e. a principal filter that is also an ultrafilter) iff $Q = I_x \stackrel{def}{=} \{A \subseteq E : x \in A\}$ for some $x \in E$. Proper names such as *John*, *Mary* etc. denote principal ultrafilters in any model. As far as we know, non-principal ultrafilters do not exist in natural language. In fact, their (non-constructive) existence proof relies on Zorn's lemma (see Comfort and Negrepontis (1974)).

Quantifiers $Q \subseteq \wp(E)$ s.t. $Q = \{B \subseteq E : B \cap A \neq \emptyset\}$ for some $A \subseteq E$ are referred to as EXIST quantifiers.[2] Quantifiers $Q \subseteq \wp(E)$ s.t. $Q = \{A \subseteq E : B \subseteq A\}$ for some $B \subseteq E$ are referred to as *universal* (UNIV). The latter are simply the principal filters over E. By definition, NPs headed by the determiners *some* and *every* denote in any model EXIST and UNIV quantifiers respectively.

The *dual* of a quantifier $Q \subseteq \wp(E)$ is the quantifier $Q^d \stackrel{def}{=} \{A \subseteq E : E \setminus A \notin Q\}$. For example: any EXIST quantifier is the dual of a UNIV quantifier (and vice versa), the two trivial quantifiers are duals of each other, and every ultrafilter is self-dual. Further, in any model

[2]Note the difference from the standard (and more general) notion of *existential* (or *intersective*) quantifiers: those quantifiers $Q \subseteq \wp(E)$ for which there is $A \subseteq E$ s.t. for all $B, C \subseteq E$ that satisfy $|B \cap A| = |C \cap A|$: $B \in Q \Leftrightarrow C \in Q$.

the dual of the quantifier that is denoted by the noun phrases *infinitely many students* is the quantifier denoted by the noun phrase *all but finitely many students*. Note also that $\left(Q^d\right)^d = Q$ for any quantifier Q.

Following Barwise and Cooper (1981), we say that a quantifier $Q \subseteq \wp(E)$ *lives on* a set $A \subseteq E$ (or alternatively: A is a *live on set* of Q) if for all $B \subseteq E$: $B \in Q \Leftrightarrow A \cap B \in Q$. For example, the noun phrase *some student* denotes a quantifier that lives on the set of students. A determiner function D_E is *conservative (*CONS*)* if for all $A \subseteq E$: $D_E(A)$ lives on A. The well-known *conservativity* generalization of Barwise and Cooper (1981) states that all natural language determiners are conservative.

The denotation of determiner expressions varies with the choice of the domain E. Their meaning across different domains is therefore described using *global determiners* — functionals that map a domain E to a (local) determiner function D_E. We say that a global determiner D (and the determiner expression it corresponds to) has a property $\mathcal{F}$ if D_E has property $\mathcal{F}$ for any choice of E. One useful property of global determiners is *isomorphism invariance*. Standardly, we say that a global determiner D is *isomorphism invariant* (ISOM) if for all bijections $\pi : E \to E'$, for all $A, B \subseteq E$: $\{\pi(y) : y \in B\} \in D_{E'}(\{\pi(x) : x \in A\}) \Leftrightarrow B \in D_E(A)$. An ISOM global determiner is thus sensitive only to cardinalities of sets and not to the identity of elements in the domain. All the determiners in Table 1 are ISOM.

Another useful property of global determiners is *extension*. We say that a global determiner D satisfies *extension* (EXT) if for all $A, B \subseteq E \subseteq E'$: $B \in D_E(A) \Leftrightarrow B \in D_{E'}(A)$. Intuitively, the global determiners that satisfy EXT are those determiners that do not change their behavior on subsets of a domain E, when E is extended with new elements.

It is standard in GQT to concentrate on conservative global determiners in natural language that satisfy ISOM and EXT, and in this paper we follow this practice. While it is doubtful whether there are any non-conservative determiners in English,[3] there are some determiner functions in natural language that do not satisfy ISOM and EXT, most notably genitive pre-nominals like *my* or *John's*. However, the treatment of scope dominance with these expressions is fairly straightforward, as the quantifiers they lead to are either principal ultrafilters or principal filters, which will be treated below. It is therefore instructive for us to officially adopt the following standard assumption in the rest of this paper:

Assumption 1 *Natural language determiners are conservative and satisfy* ISOM *and* EXT.

2.2 Scope dominance

Reconsider the following simple example from the introduction.

(2) A priest visited every church.

This sentence is ambiguous with regard to the relative scope of the subject and object quantifiers. Assume that the nouns *priest* and *church* denote sets $P, C \subseteq E$ respectively, and that the transitive verb *visited* denotes a binary relation $V \subseteq E \times E$. The two readings of the sentence are the following:

1. $\exists p \in P, \forall c \in C : \langle p, c\rangle \in V$
2. $\forall c \in C, \exists p \in P : \langle p, c\rangle \in V$

[3]The expression *only*, as in *only students smiled*, is the typical counterexample to this claim. But it is often pointed out that this is just one of many indications that *only* requires a different semantic treatment than determiners in natural language.

The first reading, in which the object quantifier takes scope below the existential quantifier denoted by the subject, is standardly called the *object narrow scope* (ONS) reading of the sentence. The second reading is called the *object wide scope* (OWS) reading.

In general, let Q_1 and Q_2 be the generalized quantifier denotations of the subject and object respectively, and let R be the denotation of the transitive verb V in a sentence of the form SVO. The ONS and OWS readings of the sentence are the following, respectively.

(3)
$$\begin{array}{lcl} Q_1Q_2R & \overset{def}{\Leftrightarrow} & \{x \in E | R_x \in Q_2\} \in Q_1; \\ Q_2Q_1R^{-1} & \overset{def}{\Leftrightarrow} & \{y \in E | R^y \in Q_1\} \in Q_2, \end{array}$$

We standardly use the notations $R_x \overset{def}{=} \{y \in E : R(x,y)\}$ and $R^y \overset{def}{=} \{x \in E : R(x,y)\}$. Scope dominance between quantifiers is now defined as follows.

Definition 1 (scope dominance) *A quantifier $Q_1 \subseteq \wp(E)$ is* scopally dominant *over a quantifier $Q_2 \subseteq \wp(E)$ if for all binary relations $R \subseteq E \times E$: $Q_1Q_2R \Rightarrow Q_2Q_1R^{-1}$.*

Any EXIST quantifier (e.g. the denotation of *a priest* in (2)) is scopally dominant over any UNIV quantifier (e.g. the denotation of *every church*). When two quantifiers Q_1 and Q_2 are dominant over each other, we say that they are *scopally independent*.

Previous studies of generalized quantifiers have characterized some commutativity properties of quantifiers in constructions with multiple quantification. Notably, Westerståhl (1996) characterizes the class of *self-commuting* quantifiers — those quantifiers Q that satisfy the following equivalence.

(4) For all $R \subseteq E \times E : QQR \Leftrightarrow QQR^{-1}$.

Thus, Q is self-commuting iff Q stands in the independence (equivalently, scope dominance) relation to itself. Westerståhl shows that a quantifier Q is self-commuting iff it is existential, universal, a symmetric difference of principal ultrafilters, or a negation of such a symmetric difference. Another commutativity problem was studied by Zimmermann (1993). Zimmermann characterizes the class of *scopeless* quantifiers — those quantifiers Q that satisfy the following equivalence.

(5) For all $Q_1 \subseteq \wp(E)$, for all $R \subseteq E \times E : QQ_1R \Leftrightarrow Q_1QR^{-1}$.

Thus, Q is scopeless if it stands in the independence relation to any quantifier Q_1. Zimmermann shows that the scopeless quantifiers over E are precisely the ultrafilters over E. In particular, all the principal ultrafilters — the denotations of proper names such as *Mary* and *John* — are scopeless.

The more general notion of scope dominance was first studied in Westerståhl (1986). Westerståhl points out the following fact about scope dominance relations with monotone quantifiers over *finite domains*:[4]

Fact 1 *Let Q_1 and Q_2 be monotone quantifiers over a finite domain E. Q_1 is dominant over Q_2 iff these quantifiers fall under at least one of the following cases:*

(i) Q_1 *is* EXIST.

(ii) Q_2 *is* UNIV.

[4]Actually, Westerståhl characterizes scope dominance for determiners, and Fact 1 is a simpler statement of his result for generalized quantifiers. Westerståhl uses the term *order* to refer to what we here call *scope dominance*.

(iii) $Q_1 = \wp(E)$ *and* $Q_2 \neq \emptyset$.

(iv) $Q_2 = \emptyset$ *and* $Q_1 \neq \wp(E)$.

The "if" direction of the proof is easy, and holds independently of the cardinality of E. For the construction that proves the "only if" direction, which will be useful for the algorithm in the next section, assume that Q_1 is dominant over Q_2. First it is easy to see that if $Q_1 = \wp(E)$ then $Q_2 \neq \emptyset$ and that if $Q_2 = \emptyset$ then $Q_1 \neq \wp(E)$. Assume for contradiction that no one of the conditions *(i)–(iv)* holds. By finiteness of E, there is a minimal set $A \in Q_1$ such that $|A| \geq 2$ (otherwise by monotonicity, $Q_1 = \wp(E)$ or $Q_1 = \{A \subseteq E | \exists x \in A : \{x\} \in Q_1\}$ which is EXIST). Similarly, by finiteness of E, there are $B_1, B_2 \in Q_2$ such that $B_1 \cap B_2 \notin Q_2$ (otherwise by monotonicity, $Q_2 = \emptyset$ or $Q_2 = \{A \subseteq E | \cap Q_2 \subseteq A\}$ which is UNIV). Given the sets A, B_1, and B_2, and an arbitrary $a \in A$, it is easy to verify that the relation $R = (\{a\} \times B_1) \cup ((A \setminus \{a\}) \times B_2)$ contradicts our assumption that Q_1 is dominant over Q_2.

One familiar example of scope dominance was already given in sentence (2). According to Westerståhl's observation, the situation is similar over finite domains in cases where (exactly) one of the quantifiers is replaced by another monotone quantifier. The sentences in (6) below illustrate some cases like that, where the ONS reading entails the OWS reading over finite domains.

(6) a. At least half/at least two/all but at most five of the students saw every teacher.

b. Some student saw at least half/at least two/all but at most five of the teachers.

Westerståhl's result shows that over finite domains, EXIST quantifiers (for Q_1) and UNIV quantifiers (for Q_2) are the only non-trivial monotone quantifiers that exhibit scope dominance relations with other monotone quantifiers. Thus, the sentences in (2) and (6) are representative of the cases where upward monotone quantifiers lead to an entailment from the ONS reading to the OWS reading over finite domains. It is easy to see that the scope dominance relations that are characterized by Fact 1 also hold in infinite domains. However, over infinite domains there are also cases of scope dominance that are not covered by Westerståhl's characterization. Consider the following example (Altman et al. (2002), following Westerståhl), where E is assumed to be countable.

(7) Infinitely many dots are contained in at least one of the three circles.

$$Q_1 = \{A \subseteq E : |D \cap A| = \aleph_0\}$$
$$Q_2 = \{A \subseteq E : C \cap A \neq \emptyset\}, \text{ where } |C| = 3$$

It is easy to verify that Q_1 is dominant over Q_2, but Q_1 and Q_2 are upward monotone and the conditions of Fact 1 do not hold. Altman et al. (2002) extend Westerståhl's result to cover also such dominance relations over countable domains. They use the following quantifier properties, in addition to the ones that were defined in Subsection 2.1.

A quantifier Q is said to satisfy the *union property* (U) if for all $A_1, A_2 \subseteq E$: if $A_1 \cup A_2 \in Q$ then $A_1 \in Q$ or $A_2 \in Q$. For example, any EXIST quantifier satisfies (U) while a UNIV quantifier $\mathbf{every}'(X)$ satisfies (U) if and only if X is either a singleton or the empty set. The set of all infinite subsets of E satisfies (U) as well.

Further, we say that a quantifier Q satisfies the *Descending Chain Condition* (DCC) if for every descending sequence $A_1 \supseteq A_2 \supseteq \cdots \supseteq A_n \supseteq \cdots$ in Q, the intersection $\cap A_i$ is in Q as well. For example, any UNIV quantifier satisfies (DCC). An EXIST quantifier $\mathbf{some}'(X)$

satisfies (DCC) if and only if X is finite. Another quantifier that satisfies (DCC) is the following, where the domain $E = \mathbb{N}$ is the set of natural numbers:

$$\{A \subseteq \mathbb{N} : \forall n \in \mathbb{N}[2n \in A \vee 2n+1 \in A]\}.$$

If every set in a quantifier Q contains a finite subset that is also in Q, we say that Q satisfies (FIN). The following fact uses (FIN) in characterizing dual properties to (U) and (DCC) over the class of monotone quantifiers.

Fact 2 *For any monotone generalized quantifier Q over a domain E:*

(i) Q satisfies (U) *iff Q^d is a filter.*

(ii) If E is countable: Q satisfies (DCC) *iff Q^d satisfies* (FIN).

Using these two pairs of dual properties, Altman et al. prove the following theorem.

Theorem 3 *Let Q_1 and Q_2 be monotone quantifiers over a countable domain E. Then the following conditions (I) and (II) are equivalent:*

I. Q_1 is dominant over Q_2.

II. (i) Q_1^d or Q_2 (or both) are filters;
(ii) Q_1^d or Q_2 (or both) satisfy (DCC)*;*
(iii) Q_1^d or Q_2 (or both) are not empty.

Using Fact 2, it can be easily shown that when Q_1 and Q_2 are monotone quantifiers over a countable domain, Statement (II) above is equivalent to Statement (II') below.

(II') Q_1 and Q_2 fall under at least one of the following cases:

(i') Q_1 is EXIST.
(ii') Q_2 is UNIV.
(iii') Q_1 satisfies (U), $Q_2 \neq \emptyset$ and Q_2 satisfies (DCC).
(iv') Q_2 is a filter, $Q_1 \neq \wp(E)$ and Q_1 satisfies (FIN).
(v') $Q_1 = \wp(E)$ and $Q_2 \neq \emptyset$.
(vi') $Q_2 = \emptyset$ and $Q_1 \neq \wp(E)$.

Altman et al.'s result thus shows that in addition to EXIST and UNIV quantifiers over finite domains, the other non-trivial quantifiers that show scope dominance over countable domains are as classified in Clauses (iii') and (iv'). In sentence (7) above we have already seen an example for Clause (II')(iii'): Q_1 satisfies (U), and $Q_2 = \mathbf{some}'(C)$ for a finite C ($|C| = 3$), hence it satisfies (DCC). The following example illustrates quantifiers Q_1 and Q_2 that fall under the dual case of Clause (II')(iv').

(8) Each of the three circles contains all but finitely many dots.

$$Q_1 = \{A \subseteq E : C \subseteq A\}, \text{ where } |C| = 3$$
$$Q_2 = \{A \subseteq E : |D \setminus A| < \aleph_0\}$$

By using the more general results of Altman et al. (2002), we will be able to take into account in the system more cases that go beyond first order logic, and to deal also with the characterization of scope dominance with monotone determiners whose definition involves infinite domains, such as *infinitely many* or *all but finitely many*.

2.3 Specifying determiner functions

Our usage of Theorem 3 in the computation of scope dominance requires to specify the meanings of lexical determiner expressions. Van Benthem (1984) points out that for any globally conservative determiner D that satisfies ISOM and EXT there is a relation f_{det} s.t. for all $A, B \subseteq E$: $B \in D_E(A)$ iff $f_{det}(|A \cap B|, |A \setminus B|)$ holds. For such monotone Ds over *finite* domains, we can furthermore use the following fact (cf. Väänänen and Westerståhl (2001)).

Fact 4 *For any monotone conservative global determiner D that satisfies* ISOM *and* EXT, *there is a function $g_{det} : \mathbb{N} \to \mathbb{N}$ that satisfies for all $A, B \subseteq E$ s.t. A is finite:*

$$B \in D(A) \Leftrightarrow |A \cap B| \geq g_{det}(|A|).$$

For countably infinite arguments of D, we observe the following fact.

Fact 5 *For any monotone conservative global determiner D that satisfies* ISOM *and* EXT, *there is $n \in \mathbb{N} \cup \{\aleph_0\}$ s.t. one of the following holds:*

(i) For all $A, B \subseteq E$ s.t. $|A| = \aleph_0$: $B \in D(A) \Leftrightarrow |A \cap B| \geq n$;

(ii) For all $A, B \subseteq E$ s.t. $|A| = \aleph_0$: $B \in D(A) \Leftrightarrow |B \setminus A| < n$.

Proof Let f_{det} be the "Van Benthem relation" over $\mathbb{N} \cup \{\aleph_0\}$ that specifies the behavior of D over countable sets. By monotonicity of D, we have for all $a, b \in \mathbb{N} \cup \{\aleph_0\}$ s.t. $a < b$:

(*) $f_{det}(a, \aleph_0) \Rightarrow f_{det}(b, \aleph_0)$,

(**) $f_{det}(\aleph_0, b) \Rightarrow f_{det}(\aleph_0, a)$.

Let $A, B \subseteq E$ be sets s.t. $|A| = \aleph_0$.

Assume that $f_{det}(\aleph_0, \aleph_0)$ holds. Let $n \in \mathbb{N} \cup \{\aleph_0\}$ be the minimal value s.t. $f_{det}(n, \aleph_0)$ holds. By (*), $f_{det}(a, \aleph_0)$ holds for all $a \geq n$. By (**), $f_{det}(\aleph_0, b)$ holds for all $b \in \mathbb{N} \cup \{\aleph_0\}$. Thus, $B \in D_E(A) \Leftrightarrow f_{det}(|A \cap B|, |A \setminus B|) \Leftrightarrow |A \cap B| \geq n$.

Now assume $f_{det}(\aleph_0, \aleph_0)$ does not hold. Let $n \in \mathbb{N} \cup \{\aleph_0\}$ be the minimal value s.t. $f_{det}(\aleph_0, n)$ does not hold. By (**), $f_{det}(\aleph_0, a)$ does not hold for any $a \geq n$. By (*), $f_{det}(b, \aleph_0)$ does not hold for any $b \in \mathbb{N} \cup \{\aleph_0\}$. Thus, $B \in D_E(A) \Leftrightarrow f_{det}(|A \cap B|, |A \setminus B|) \Leftrightarrow |A \setminus B| < n$. □

This fact means that with countably infinite arguments, monotone determiners with the assumed properties either put a "minimum cardinality" requirement on the intersection of their A and B arguments, or a "maximum cardinality" requirement on the set $B \setminus A$. In the first case the determiner is equal over countably infinite domains to the one of the determiners *at least n* (for $n \in \mathbb{N}$) and *infinitely many*. In the second case, the determiner behaves over countably infinite domains like the determiner *all but less than n* (for $n \in \mathbb{N}$) or *all but finitely many*. In the sequel we use the property $inf_{det} \in \{$'at_least_n', 'infinitely_many', 'all_but_lt_n', 'all_but_fin_many'$\}$ to denote the behavior of a determiner over countably infinite domains.

From Facts 4 and 5 it follows that the values g_{det} and inf_{det} completely specify the behavior over countable domains of any monotone conservative global determiner that satisfies ISOM and EXT. In the algorithm that we propose in the next section these two values are used for computing scope dominance in simple transitive sentences. The algorithm generates a model $\mathcal{M}$ which is *indicative of scope dominance* in the sentence. In an indicative model $\mathcal{M}$, a scope dominance relation exists between the quantifier denotations of the subject and the object if and only if such a relation is exhibited under *any* model. Thus, for a model $\mathcal{M}$ to be indicative of scope dominance in a sentence s, we have to guarantee that in terms of the properties that affect scope dominance, the quantifiers that $\mathcal{M}$ assigns to the subject and object in s exhibit the "most general" behavior of the global determiners in s. For example, consider a model in which the subject NP, headed by a determiner d, denotes an EXIST quantifier $D(A)$. This model is not indicative of scope dominance if the global determiner D that d denotes, when applied to a different set A', leads to a non-EXIST quantifier $D(A')$. To ensure that the set argument of each determiner leads to a "most general" behavior of the determiner, we keep for each lexical determiner a value that is called *lowest general cardinality* (*lgc*). This is the minimal cardinality in $\mathbb{N} \cup \{\aleph_0\}$ of set arguments for which the determiner displays its "most general" behavior in terms of scope dominance. Consider for example the determiner expression *at least three*. The generalized quantifier $\mathbf{at\_least\_}\mathit{3}'(A)$ is neither EXIST nor UNIV for any A s.t. $|A| \geq 4$. For smaller cardinalities of A, the quantifier is either UNIV (when $|A| = 3$) or empty, hence EXIST (when $|A| \leq 2$). Thus, the value of *lgc* for *at least three* is 4. Similarly for $|A| \leq 2$, the quantifier $\mathbf{at\_least\_half}'(A)$ is UNIV (when $|A| = 0$ or $|A| = 1$) or EXIST (when $|A| = 1$ or $|A| = 2$). But for $|A| > 2$ this quantifier is neither UNIV nor EXIST. Hence the *lgc* value of *at least half* is 3.

Formally, the *lgc* value of a determiner is defined as follows.

Definition 2 (lowest general cardinality) *Let D be a global* ISOM *determiner. The* lowest general cardinality *(lgc) of D is the minimal value in $\mathbb{N} \cup \{\aleph_0\}$ s.t. the class of generalized quantifiers $D_E(A)$ with $A \subseteq E$ and $|A| \geq lgc$ is contained in (exactly) one of the following four classes of GQs:* TRIV, $\mathrm{EXIST} \setminus \mathrm{TRIV}$, $\mathrm{UNIV} \setminus \mathrm{TRIV}$ *or* $\overline{\mathrm{EXIST} \cup \mathrm{UNIV}}$.

It is easy to verify that *lgc* is well-defined for any ISOM determiner (though not necessarily for other determiners). Table 2 shows the values of g_{det}, inf_{det}, and *lgc* for the (monotone) determiner expressions that are treated as lexical in the fragment of the next section. Note that although in theory the *lgc* value of a determiner is determined by the g_{det} and inf_{det} values, its specification in the lexicon is required in order for it to be easily computable.

Determiner	$g_{det}(n)$	inf_{det}	*lgc*
every, each, all	n	all_but_lt_1	2
some	1	at_least_1	2
at least m	m	at_least_m	$\begin{cases} 0 & m = 0 \\ m+1 & \text{otherwise} \end{cases}$
more than m	$m+1$	at_least_$m+1$	$m+2$
at least half	$\lceil n/2 \rceil$	*undefined*	3
more than half	$\lceil (n+1)/2 \rceil$	*undefined*	3
all but at most m	$\max(n-m, 0)$	all_but_lt_$m+1$	$m+2$
infinitely many	$n+1$	infinitely_many	$\aleph_0$
all but finitely many	0	all_but_fin_many	$\aleph_0$

Table 2: Values of g_{det}, inf_{det}, and *lgc* for some determiner expressions

A priori, we could think that the *lgc* value of a determiner underspecifies its "most general" behavior. This is because *lgc* is defined as a minimal cardinality for this behavior, and in smaller cardinalities of its set argument, the determiner can still behave in an arbitrary manner. However, we do not anticipate such cases to appear in natural languages. Consider for instance the following global determiner D_0:

$$D_0(A) \stackrel{def}{=} \begin{cases} \textbf{at\_least\_}\mathit{3}'(A) & |A| \leq 5 \\ \wp(E) & |A| > 5 \end{cases}$$

This determiner behaves like the determiner *at least three* for any set argument of cardinality ≤ 5, but it returns the trivial powerset quantifier for any argument of a larger cardinality. While this kind of determiners is not ruled out by the standard constraints in Assumption 1, it is not expected in any natural language (cf. Keenan and Westerståhl (1996)) — a determiner that returns a TRIV_1 quantifier only when its argument *exceeds* a certain cardinality is quite "unnatural". We say that any natural language determiner is expected to be *downward consistent* with respect to triviality.[5] Formally, we define downward consistency as follows.

Definition 3 (downward consistency) *A global determiner D is* downward consistent *with respect to a property $\mathcal{F}$ of GQs if for all $A \subseteq B \subseteq E$, if $D_E(B)$ satisfies $\mathcal{F}$ then $D_E(A)$ satisfies $\mathcal{F}$ as well.*

This definition raises the following question: what are the properties $\mathcal{F}$ of GQs with respect to which natural language determiners are downward consistent? We will not try to address this question in its generality, but only suggest the following hypothesis about downward consistency, which will be useful in the rest of this paper.

Assumption 2 *Lexical determiners in natural language are downward consistent with respect to the following properties:* (U), (FLT), (FIN), (DCC), (TRIV_0) *and* (TRIV_1).

For example, consider the determiners D_1 and D_2 below. These determiners, which mix non-trivial EXIST and UNIV quantifiers, are also ruled out by Assumption 2, as is D_0 above.

$$D_1(A) \stackrel{def}{=} \begin{cases} \mathbf{some}'(A) & |A| \leq 5 \\ \mathbf{every}'(A) & |A| > 5 \end{cases} \qquad D_2(A) \stackrel{def}{=} \begin{cases} \mathbf{every}'(A) & |A| \leq 5 \\ \mathbf{some}'(A) & |A| > 5 \end{cases}$$

Here, the determiner D_1 is not downward consistent for the filter property, whereas D_2 is not downward consistent for (U).

Note that Assumption 2 does not rule out the possibility of *non-lexical* determiners that are not downward consistent with respect to one of the above properties. For instance, the disjunctive determiner *every or infinitely many* is not downward consistent with respect to the (U) property. In this paper we do not treat such boolean compounds of determiners, and therefore Assumption 2 holds of all the determiners in the fragment of the following section.

3 Computing scope dominance in a simple fragment

It is easy to use Theorem 3 for computing scope dominance relations between simple NPs — noun phrases of the structure *Determiner-Nominal*. This can be done directly by using the lexical information on determiners that was specified in Table 2. However, in order to deal with more substantial parts of natural language, certain non-trivial problems have to be tackled with. In this section we introduce a method for computing scope dominance in a simple fragment of natural language that includes NP coordinations, cardinality presuppositions about nouns, and containment relations between them. After defining the problem and introducing the main ideas of the algorithm, we describe in detail its main parts and give their correctness claims, which are proven in Appendix A. The section is concluded by a discussion of some of the limitations of the algorithm.

[5] This would be incorrect for the (TRIV_0) property in a Russellian treatment of definite articles as determiners. The Russellian determiner denotation of *the* leads to an empty quantifier for any argument that is not a singleton. Hence this determiner function is not downward consistent for (TRIV_0). The presuppositional analysis of definites that we use in this paper avoids such counterexamples.

3.1 The fragment and definition of the problem

The grammar and lexicon in Figure 1 specify a fragment which involves simple upward monotone NPs (possibly involving cardinality presuppositions) and their *and*/*or* coordinations. In addition, the lexicon specifies which pairs of common nouns in it stand in a containment relation to one another.

1.	S	$\rightarrow$	NP V NP
2.	NP	$\rightarrow$	DetP N / NP and NP / NP or NP
3.	DetP	$\rightarrow$	Det'
4.	DetP	$\rightarrow$	Det' of the NUM
5.	Det'	$\rightarrow$	D / F NUM

V:	*saw, visited, graded ...*
N:	*student*(*s*), *teacher*(*s*), *person*(*s*), ...
NUM:	*one, two, three, ...*
D:	*every, all, some, at least half, more than half, infinitely many, all but finitely many*
F:	*at least, more than, all but at most*

Containment relation: $\langle student, person\rangle$, $\langle teacher, person\rangle$, ...

Figure 1: simple grammar and lexicon

Information about cardinality of denotations of nominal expressions in a given sentence may come from the context or from presuppositions that are induced by NPs in the same sentence. Cardinality presuppositions are induced by the constructions like the following:

1. Definite (numeral) NPs such as *the* (*ten*) *students*;
2. Partitive NPs such as *all*/*some*/*at least three of the ten students*;
3. Proper names.
4. Proportional NPs such as *at least half of the students*, which we assume contribute a finiteness presupposition about the denotation of *students*.

Of these four types of NPs, only 2 and 4 are represented in the fragment above. This simplification is innocuous, however, because a definite numeral like *the* (*ten*) *students* can be simply represented for the purposes of scope dominance as equivalent to the partitive *each of the ten students*. Similarly, a proper name like *Mary* can be represented as equivalent to the partitive numeral *at least one of the one mary*, where *mary* is an artificial common noun.

Cardinality information is crucial for our purposes because it may create more scope dominance relations than what is predicted from the semantics of the determiners in the sentence. To see that, consider the following example. Suppose that in some academic unit, there are two folders for storing reviews of research proposals: one for internal reviews and the other for external reviews. In a certain occasion, proposals for review were given to five internal referees. Consider now the following discourse.

(9) A: Could you check for me what's going on with the reviews we are expecting?

B: Well, let me check first the internal referees: *more than four of them have already reviewed each proposal*.

B's reaction is completely acceptable, assuming she is unaware that there were only five referees, all of which are internal. However, once also taking this background knowledge into account, we can conclude that the two readings of B's utterance are equivalent. This is because the noun phrase *more than four of them* is interpreted as equivalent to *every referee*. Note that without the information about the number of referees, the two readings of B's utterance would not be equivalent. This is an example of how downward consistency and the *lgc* value of a determiner affect its participation in scope dominance relations: since the *lgc* of *more than four* is 6, with the information that there are only five referees the noun phrase *more than four referees* does not exhibit its most general behavior with respect to scope dominance, as we saw is the case in B's sentence above.

A similar example is illustrated more concisely by the following sentence.

(10) At least two of the three persons in this room admire more than two students in this room.

In this sentence there is a presupposition that the total number of people in the room is three. Consequently, given the containment between *students* and *persons*, the noun phrase *more than two students in this room* is either empty (in case there are not more than two students in the room) or universal (in case the three people in the room are all students). According to Theorem 3, in both cases the subject quantifier is scopally dominant over the object quantifier under any model that respects the given cardinality presuppositions and containment relations between nouns. Again, without any cardinality presuppositions or containment relations about N_1 and N_2, there is no scope dominance of NPs of the form *at least two of* N_1 over NPs of the form *more than two* N_2.

Now we can formally define the requirements from a system for computing scope dominance in sentences of the fragment in Figure 1. For simplicity, we will not model here possible contextual information about cardinality of noun denotations, which does not add any significant complications to the problems exhibited by cardinality presuppositions. The implementation that is described in the next section, however, takes into account this information as well.

Let *NOM* be the set of all common nouns in the lexicon, and let *cont* $\subseteq$ *NOM* $\times$ *NOM* be the lexical *containment relation* between them. When $\langle \mathsf{n}_1, \mathsf{n}_2 \rangle \in$ *cont* holds, this is interpreted as meaning that the denotation of n_1 is contained in the denotation of n_2 in every model. Let $\mathcal{M}$ be an extensional model: a pair $\langle E, I \rangle$, where E is a non-empty set of entities, and I is an interpretation function of non-logical constants.[6] Let s be a sentence in the language $L(G)$ that is generated by the grammar G in Figure 1. Consider the following conditions on a model $\mathcal{M}$ for the language $L(G)$.

(11) a. For all $\langle \mathsf{n}_1, \mathsf{n}_2 \rangle \in$ *cont*: $[\![\mathsf{n}_1]\!]_{\mathcal{M}} \subseteq [\![\mathsf{n}_2]\!]_{\mathcal{M}}$;

b. For each parse tree T_{s} in G of s, for each subtree of T_{s} of the form $[_{\mathrm{NP}}[_{\mathrm{DetP}}[_{\mathrm{Det'}}...]\mathrm{of\ the}\ [_{\mathrm{NUM}}\mathsf{num}][_{\mathrm{N}}\mathsf{n}]]]$ (generated by Rule 4 in G) the cardinality $|\,[\![\mathsf{n}]\!]_{\mathcal{M}}\,|$ is the natural number expressed by the numeral expression num.

c. For each parse tree T_{s} in G of s, for each subtree of T_{s} of the form $[_{\mathrm{NP}}[_{\mathrm{DetP}}\mathsf{d}][_{\mathrm{N}}\mathsf{n}]]]$ (generated by Rule 3 in G), the cardinality $|\,[\![\mathsf{n}]\!]_{\mathcal{M}}\,|$ is finite if d is *more than half* or *at least half*.

[6] We skip here the routine technical details about the definition of I and its extension for denotations in $\mathcal{M}$.

The conditions in (11) guarantee that the model satisfies the containment relations of the lexicon, the background information about cardinality of nouns, and the cardinality presuppositions about nouns in the sentence.[7] Let T be a parse tree of a sentence $NP_1\ V\ NP_2$ that is generated by the fragment in Figure 1. Let T_1 and T_2 be the subtrees of T for the subject NP_1 and the object NP_2 respectively. We say that T *exhibits scope dominance* of the subject over the object if for every model $\mathcal{M}$ that satisfies the conditions in (11), the quantifier $[\![T_1]\!]_{\mathcal{M}}$ is scopally dominant over the quantifier $[\![T_2]\!]_{\mathcal{M}}$. Symmetrically for scope dominance of the object over the subject.

The algorithm we describe in this section is developed according to the following input-output specification.

(12) **Input**:

- Grammar G of Figure 1.
- Lexicon L for this grammar, plus a containment relation *cont* over its common nouns (*NOM*).
- The information on determiners in Table 2.
- An input string s.

(13) **Output**:

The algorithm first verifies that s is in the language of G and L, and that there is no contradiction the between the containment and cardinality information about nouns, as specified by the relation *cont* and the cardinality presuppositions of s.

If this is the case, then for each parse tree T of s in G, for each direction of scope dominance (subject over object or vice versa), the algorithm outputs the following:

- "Yes", "No" or "Undecided" if T exhibits scope dominance (in the respective direction), does not exhibit scope dominance, or the algorithm fails to determine that, respectively.
- If the output is "No" and there is a finite model that demonstrates that, the algorithm outputs such a model.

3.2 Overview: on the generation of an indicative model

The algorithm that is described below generates a model $\mathcal{M}$ which is indicative of scope dominance relations in the given sentence. Such a model $\mathcal{M}$, if found, guarantees that computing scope dominance between the GQ denotations it assigns to the NPs in the sentence is an indication of scope dominance under *any* model.

Finding an indicative model for sentences in the fragment with only simple (i.e. non-coordinate) NPs is straightforward once observing the following fact, which is a direct result of Theorem 3 and our assumptions about the downward consistency of determiners.

Fact 6 *Let D_1 and D_2 be global determiners that satisfy Assumptions 1 and 2. Assume that over a countable domain E_1 with $A_1, B_1 \subseteq E_1$, the following hold:*

(i) either $|A_1| = lgc(D_1) < \aleph_0$ and $|B_1| = lgc(D_2) < \aleph_0$, or
$|A_1| = \aleph_0$ and $|B_1| = \aleph_0$;

[7] Note that in this grammar, different parse trees of a sentence always agree on its cardinality presuppositions.

(ii) $D_1(A_1)$ *is scopally dominant over* $D_2(B_1)$*;*

(iii) $D_1(\emptyset)$ *is scopally dominant over* $D_2(\emptyset)$*.*

Then $D_1(A)$ *is scopally dominant over* $D_2(B)$ *for all* $A, B \subseteq E$*, where* E *is countable.*

This fact means that in transitive sentences with only simple NPs, denoted by $D_1(A)$ and $D_2(B)$, in order to decide about scope dominance it is sufficient to consider two models:

(14) a. A model $\mathcal{M}_1$ where A and B's cardinalities are the lgc values of the respective determiners (if these *lgc* values are both finite), or where A and B are of cardinality $\aleph_0$ (if one of the *lgc* values is infinite).

b. A model $\mathcal{M}_2$ where $A = B = \emptyset$.

In such a pair of models we can compute the relevant properties of the determiners using the g_{det} and inf_{det} values and apply Theorem 3. Since the model $\mathcal{M}_2$ is needed only for checking the trivial borderline cases of Theorem 3, it is appropriate to say that a model like $\mathcal{M}_1$, if it exists, is *indicative* of scope dominance.

Note however that cardinality presuppositions on nouns, possibly combined with containment relations between them, can put restrictions on nominals that rule out such models as $\mathcal{M}_1$ and $\mathcal{M}_2$ for the interpretation of the sentence. This is illustrated by example (10) above, and a similar point holds for example (9), when contextual information about cardinality of noun denotations is taken into account. In sentence (10), for example, it is impossible to assign the nominal *students in this room* a set denotation of cardinality four, as the *lgc* value of the determiner *at least two* would require. This is because of the cardinality presupposition about the nominal *persons in this room*. Similarly, of course, it is impossible to assume that the cardinality of the denotation of *students in this room* is infinite. Thus, a model such as $\mathcal{M}_1$ would violate the cardinality presupposition of sentence (10). A model such as $\mathcal{M}_2$ would also be illegitimate for similar reasons.

To solve this problem, the algorithm we propose searches for two models with the following properties, which generalize the properties in (14).

(15) a. A model $\mathcal{M}_1$ that satisfies:

- If $lgc(D_1)$ and $lgc(D_2)$ are both finite then A's (B's) cardinality is greater or equal to the minimum between $lgc(A)$ ($lgc(B)$, respectively), and any possible cardinality for A (B) that satisfies the presupposition and containment requirements of the sentence.
- If $lgc(D_1)$ or $lgc(D_2)$ is infinite then A and B are of the maximal cardinality possible for them.

b. A model $\mathcal{M}_2$ where A's and B's cardinalities are the minimal cardinalities that satisfy the presupposition and containment requirements of the sentence.

By putting these requirements on the indicative models, the correctness proof of the algorithm in Section 3.5 will show that such a pair of models as $\mathcal{M}_1$ and $\mathcal{M}_2$ is sufficient for deciding about scope dominance in simple sentences with quantifiers $D_1(A)$ and $D_2(B)$, when taking into account also the cardinality restrictions on A and B. This proof will follow quite directly from the definition of *lgc* and our assumptions about downward consistency.

Moving on to NP coordinations, the following two simple facts are crucial for the computation of their semantic properties that affect scope dominance according to Theorem 3.

Fact 7 *Let Q_1 and Q_2 be two GQs. Then, the following hold:*

1. *If both Q_1 and Q_2 satisfy* $(\mathrm{DCC})/(\mathrm{FIN})/(\mathrm{FLT})$ *then $Q_1 \cap Q_2$ satisfies* $(\mathrm{DCC})/(\mathrm{FIN})/(\mathrm{FLT})$ *respectively.*

2. *If both Q_1 and Q_2 satisfy* $(\mathrm{DCC})/(\mathrm{FIN})/(\mathrm{U})$ *then $Q_1 \cup Q_2$ satisfies* $(\mathrm{DCC})/(\mathrm{FIN})/(\mathrm{U})$ *respectively.*

Fact 8 *Let Q_1 and Q_2 be two non-trivial GQs. Assume that there are live on sets A_1 and A_2 for Q_1 and Q_2 respectively s.t. A_1 and A_2 are disjoint. Then, the following hold:*

1. *If $Q_1 \cap Q_2$ satisfies* $(\mathrm{DCC})/(\mathrm{FIN})/(\mathrm{FLT})$*, then both Q_1 and Q_2 satisfy* $(\mathrm{DCC})/(\mathrm{FIN})/(\mathrm{FLT})$ *respectively.*

2. *$Q_1 \cap Q_2$ does not satisfy* (U).

3. *If $Q_1 \cup Q_2$ satisfies* $(\mathrm{DCC})/(\mathrm{FIN})/(\mathrm{U})$*, then both Q_1 and Q_2 satisfy* $(\mathrm{DCC})/(\mathrm{FIN})/(\mathrm{U})$ *respectively.*

4. *$Q_1 \cup Q_2$ does not satisfy* (FLT).

These two facts together guarantee that the four properties that are relevant for characterizing scope dominance can be completely determined recursively once the nouns in the sentence are assigned pairwise disjoint set denotations, and the requirements generalizing those in (15) are met for these set denotations. Hence, when it is possible to assign the nouns disjoint sets of the cardinalities that are required in (15), the problem of coordinate NPs is reducible to the problem of simple NPs.

However, it is sometimes impossible to guarantee that sets that are assigned in this way are disjoint. This is problematic, since the disjointness assumption in Fact 8 is necessary for this fact to hold. Consider for instance the following noun phrase.

(16) *every teacher and at least two authors*

Suppose that we know that there are at least three people in the model who are both authors and teachers. The conjoined NP in (16) is equivalent to *every teacher* under any such model, and therefore this NP denotes a filter although the second conjunct *at least two authors* does not. Thus, the conditions in Fact 8 do not necessarily characterize the four relevant properties in situations where the live on sets of conjuncts in coordinate NPs are not disjoint. Thus, we will not be able to use this fact for characterizing scope dominance in such situations. The limitations that ensue for the generality of the algorithm we present will be discussed, and partly reduced, in Section 3.7.

The algorithm itself works in four stages:

- *Preparation:* Collecting the required noun cardinalities and cardinality presuppositions from the sentence, for each noun that appears in it.

- *Stage 1*: Assigning "representative" sets and minimum cardinalities to the nouns in the sentence.

- *Stage 2*: Checking whether the sets and minimum cardinalities that are assigned in Stage 1 support scope dominance in the sentence.

- *Stage 3*: In case not— constructing a finite model that shows this, if such a model exists.

The following subsections give more details about each of these stages and give their correctness claims, which are proven in Appendix A.

3.3 Preparation stage

Let s be the input sentence, and let $X \subseteq$ *NOM* be the (finite) set of common nouns in s. For any noun $\mathsf{n} \in X$, let us define the following set:[8]

$$LGC(\mathsf{n}) \stackrel{def}{=} \{lgc(\mathsf{d}) : [_{\mathrm{NP}}[_{\mathrm{DetP}}[_{\mathrm{Det'}}\ \mathsf{d} \ldots [_{\mathrm{N}}\ \mathsf{n}\ ]]]] \text{ appears as an NP in } \mathsf{s}\}.$$

We compute the following values for each $\mathsf{n} \in X$:

1. A "recommended" minimal cardinality *card*(n) for the denotation of n, defined as $\max(LGC(\mathsf{n}))$.

2. A "recommended" minimal *finite* cardinality *fincard*(n) for the denotation of n, defined as $\max(\{0\} \cup LGC(\mathsf{n}) \setminus \{\aleph_0\})$.

   Thus, when *card*(n) is finite, *fincard*(n) is equal to *card*(n). When *card*(n) is infinite, *fincard*(n) is the maximal finite lgc for determiners appearing with n, if such a finite lgc exists. If not, *fincard*(n) is zero.

3. A "presupposed" cardinality *pres*(n) for the denotation of n, defined as having one of three values:

   - $n \in \mathbb{N}$ – if all the occurrences of NUM that appear in *Det' of the NUM* n constructions in the sentence have the same value n;
   - FIN – if there are no such occurrences, and the noun n appears in determiner-noun constructions dn in the sentence where inf_{det} is undefined for d (in the given lexicon, this is the case only for *more than half* and *at least half*).
   - ϕ ("don't care") – if all occurrences of n are in constructions d n, where inf_{det} is defined for d.

   In other cases the sentence exhibits a presupposition failure, as in the sentence *the three students saw one of the four students*. When such a failure is recognized, the system halts with a proper output.

   We henceforth assume an order '$<$' on the possible values of *pres*(n) that satisfies $n <$ FIN $< \aleph_0$ for all $n \in \mathbb{N}$.

Example 1: Consider for example the following sentence, which will be used as a running example throughout the rest of this section.

(17) All of the three persons admire more than two teachers or some dog.

In this sentence, *pres*$(person) = 3$, while *pres*$(teacher)$ and *pres*(dog) are ϕ. The *card* values (which are identical to the *fincard* values) are taken directly from Table 2, and are: *card*$(person) = 2$, *card*$(teacher) = 4$, and *card*$(dog) = 2$.

[8] Similarly to the fact that was mentioned above in footnote 7, in the given grammar, different parse trees of a sentence always agree on the $LGC(\mathsf{n})$ value of a noun $\mathsf{n} \in X$.

3.4 Stage 1: Set assignment

In this stage it is checked if it is possible to assign each noun n in the input sentence a set $S(\mathsf{n})$ such that the restrictions imposed by *cont* and *pres* are preserved. If the presuppositions in *pres* do not agree with the containment relation *cont*, then set assignment fails. If they do, a set assignment is generated that, under certain limitations, encodes a model that is indicative of scope dominance in the input sentence.

Formally, the algorithm generates a domain E and a function $S : X \rightarrow \wp(E)$ that satisfy the following conditions:

(18) (i) For all $\langle \mathsf{n}, \mathsf{m} \rangle \in$ *cont*: $S(\mathsf{n}) \subseteq S(\mathsf{m})$.

(ii) For every $\mathsf{n} \in X$: if *pres*(n) is a number (i.e. it is not "don't care" or FIN) then $|S(\mathsf{n})| =$ *pres*(n); if *pres*$(\mathsf{n}) = \mathrm{FIN}$ then $|S(\mathsf{n})| < \aleph_0$.

Clauses (i) and (ii) describe the restrictions imposed by *cont* and *pres* respectively (cf. (11)). In addition, the set assignment S and the function $mincard : X \rightarrow \mathbb{N}$ that the algorithm computes satisfy the following conditions:

(18) (iii) For every $\mathsf{n} \in X$ and for all domains E' and set assignments $S' : X \rightarrow \wp(E')$ that satisfy (i) and (ii): if $|S''(\mathsf{n})| = \aleph_0$ for some domain E'' and set assignment $S'' : X \rightarrow \wp(E'')$ that satisfy (i) and (ii), then $|S(\mathsf{n})| \geq \min\{\textit{card}(\mathsf{n}), |S'(\mathsf{n})|\}$. Otherwise, $|S(\mathsf{n})| \geq \min\{\textit{fincard}(\mathsf{n}), |S'(\mathsf{n})|\}$.

(iv) If for every $\mathsf{n} \in X$: *card*$(\mathsf{n}) < \aleph_0$, then for every $\mathsf{m} \in X$: $|S(\mathsf{m})| < \aleph_0$.

(v) If $|S(\mathsf{n})| = \aleph_0$ for some $\mathsf{n} \in X$, then for every $\mathsf{m} \in X$: if for some domain E'' and set assignment $S'' : X \rightarrow \wp(E'')$ that satisfy (i) and (ii): $|S''(\mathsf{m})| = \aleph_0$, then $|S(\mathsf{m})| = \aleph_0$.

(vi) For every $\mathsf{n} \in X$ and for all domains E' and set assignments $S' : X \rightarrow \wp(E')$ that satisfy (i) and (ii): $|S'(\mathsf{n})| \geq mincard(\mathsf{n})$.

(vii) There exist some domain E' and some set assignment $S' : X \rightarrow \wp(E')$ that satisfy (i) and (ii) and for all $\mathsf{n} \in X$: $|S'(\mathsf{n})| = mincard(\mathsf{n})$.

These additional conditions are necessary in order to guarantee that the set assignment S is indicative of scope dominance in the sentence as to be computed in Stage 2 of the algorithm. Clause (iii) adopts the following policy for "recommended" cardinalities:

- If it is possible to reach an infinite cardinality for a noun n given the restrictions in Clauses (i) and (ii), adopt *card*(n) as the "recommended" cardinality for n.

- Otherwise, adopt *fincard*(n) as the "recommended" cardinality for n.

Then, the cardinality of the set $S(\mathsf{n})$ that is assigned to n is required to be be either greater or equal than the recommended cardinality (if this is possible given Clauses (i) and (ii)) or the maximal cardinality possible below the recommended cardinality (otherwise).

Clause (iv) helps to ensure finite sets and a finite domain when possible, in order to allow the generation of a finite counter-example. Clause (v) ensures that when an infinite domain is generated, all the nouns that can be assigned an infinite set (given the restrictions in Clauses (i) and (ii)) are indeed assigned such infinite sets. This is required because under infinite domains, assigning non-maximal finite denotations to nouns may lead to non-indicative models.[9] Clause (vi) ensures that $mincard(\mathsf{n})$ will be a lower bound for the size of assigned sets, and Clause (vii) ensures that this bound is attainable. This minimal bound is used to check dominance when both the subject and the object NPs denote trivial quantifiers.

An algorithm for computing a set assignment function that satisfies the requirements in (18) is given below. Let us briefly describe the operation of this algorithm. Step 1 verifies that the presuppositions can be met given the *cont* relation. Step 2 eliminates circuits in *cont*, which represent synonyms. The directed acyclic graph (DAG) G that this circuit elimination creates is sorted by increasing and decreasing topological sorts (Step 3), and then the minimum and maximum cardinality limitations as imposed by the presuppositions are applied (Steps 4-6). This is done by first (Step 4) updating the minimal and maximal cardinalities that are allowed by *pres*(n) for each node n. These values are used in Steps 5 and 6 to compute the minimal and maximal cardinalities imposed by the graph structure, in the order of the two increasing and decreasing topological sorts, respectively. Step 7 splits the set M of nodes in G that are affected by presuppositions into connected components. In each of these components, the algorithm assigns the maximally allowed number of elements to each node while making sure that sets of cardinality c contain all sets of cardinality less than c (Steps 8-9). Step 10 guarantees that if an infinite domain is required, all nouns are assigned a maximal number of elements, taking into account finiteness presuppositions. Step 11 completes the algorithm by assigning sets to the nodes in $\overline{M}$ (the nodes in G that are not in M). These nodes do not have any maximal cardinality required for them. Thus, all that is needed is to add a sufficient number of elements to the union of elements already assigned to nodes that represent subsets of the sets for the nodes in $\overline{M}$. Note that this may require assigning infinite sets. The representation of such infinite sets is described in Section 4.

Algorithm 1 (Set Assignment):
Let $X \subseteq$ *NOM* be the set of nouns in the sentence, with the functions *pres* from X to $\mathbb{N} \cup \{\mathrm{FIN}, \phi\}$ and *card* from X to $\mathbb{N} \cup \{\aleph_0\}$. Let $\overline{\textit{cont}}$ be the transitive-reflexive closure of the containment relation *cont* $\subseteq$ *NOM* $\times$ *NOM*.

1. If there is $\langle \mathsf{n}, \mathsf{m} \rangle \in \overline{\textit{cont}}$ s.t. both *pres*(n) and *pres*(m) have numerical values and *pres*$(\mathsf{n}) >$ *pres*(m), output "impossible" and stop.

2. Create a DAG G from X and *cont*: unify circuits $C \subseteq X$ in *cont* into single nodes $n(C)$ in G, and create an arc from $n(C_1)$ to $n(C_2)$ in G whenever $\langle \mathsf{n}, \mathsf{m} \rangle \in$ *cont* for some $\mathsf{n} \in C_1$ and $\mathsf{m} \in C_2$.

3. Sort G into two lists using topological sorts:

   L_{subset}: each element n in L_{subset} is a node in G s.t. for all nodes m in G for which there is an arc from m to n: m precedes n in L_{subset}

   L_{supset}: each element n in L_{supset} a node in G s.t. for all nodes m in G for which there is an arc from n to m: m precedes n in L_{supset}

[9] Consider for example the following sentence: *some circle contains infinitely many dots*. If we assume a finite number of circles, then the two readings of the sentence are equivalent. However, if there can be an infinite number of circles, each dot may be contained in a different circle, and thus the OWS reading of the sentence does not entail its ONS reading.

4. For each node n in G, define $minpres(\mathsf{n})$ and $maxpres(\mathsf{n})$ as follows:

$$minpres(\mathsf{n}) \quad := \quad \begin{cases} 0 & \textit{pres}(\mathsf{n}) \in \{\text{FIN}, \phi\} \\ \textit{pres}(\mathsf{n}) & \text{otherwise} \end{cases}$$

$$maxpres(\mathsf{n}) \quad := \quad \begin{cases} \aleph_0 & \textit{pres}(\mathsf{n}) = \phi \\ \textit{pres}(\mathsf{n}) & \text{otherwise} \end{cases}$$

5. Update $mincard(\mathsf{n})$ for each node n in G, in the order of L_{subset}: $mincard(\mathsf{n}) := \max(\{minpres(\mathsf{n})\} \cup \{mincard(\mathsf{m}) : \mathsf{m} \in Y\})$, where Y is the set of nodes m with an outgoing arc into n.

6. (Similarly) update $maxcard(\mathsf{n})$ for each node n in G, in the order of L_{supset}: $maxcard(\mathsf{n}) := \min(\{maxpres(\mathsf{n})\} \cup \{maxcard(\mathsf{m}) : \mathsf{m} \in Y\})$, where Y is the set of nodes m with an incoming arc from n.

7. Let M be the set of all nodes n in G s.t. $maxcard(\mathsf{n}) < \text{FIN}$. Let G_M be the subgraph induced by M in the undirected version of G. Let $C_1, C_2, \dots C_n \subseteq M$ be the sets of nodes in each connected component of G_M.

8. For each connected component C_i, let $m_i := \max\{maxcard(\mathsf{n}) : \mathsf{n} \in C_i\}$. Update S_i as an ordered set of m_i fresh elements.

9. For each connected component C_i, for every node $\mathsf{n} \in C_i$: let $S(\mathsf{n})$ be the set of the first $maxcard(\mathsf{n})$ elements from S_i.

10. If for some $\mathsf{n} \in X$, $\textit{card}(\mathsf{n}) = maxcard(\mathsf{n}) = \aleph_0$, then update for all $\mathsf{n} \in X$:

$$\textit{card}'(\mathsf{n}) := \begin{cases} \textit{fincard}(\mathsf{n}) & maxcard(\mathsf{n}) = \text{FIN} \\ \aleph_0 & \text{otherwise.} \end{cases}$$

Otherwise, update for all n:

$$\textit{card}'(\mathsf{n}) := \begin{cases} \textit{fincard}(\mathsf{n}) & maxcard(\mathsf{n}) = \text{FIN} \\ \textit{card}(\mathsf{n}) & \text{otherwise.} \end{cases}$$

11. Update $S(\mathsf{n})$ for each node n in G for which $S(\mathsf{n})$ is undefined, in the order of L_{subset}:

    (a) $U := \bigcup_{\text{there is an arc from } \mathsf{m} \text{ to } \mathsf{n}} S(\mathsf{m})$

    (b) $S(\mathsf{n}) := U$

    (c) if $\textit{card}'(\mathsf{n}) > |U|$, add $\textit{card}'(\mathsf{n}) - |U|$ fresh elements to $S(\mathsf{n})$.

Example 2: Consider for example the graph in Figure 2, which represents a containment relation *cont*, with values for *pres* and *card*. Note that $mincard(\mathsf{n}) \leq maxcard(\mathsf{n})$ for all nodes n. The connected components of Step 7 with more than one node are marked as C_1 and

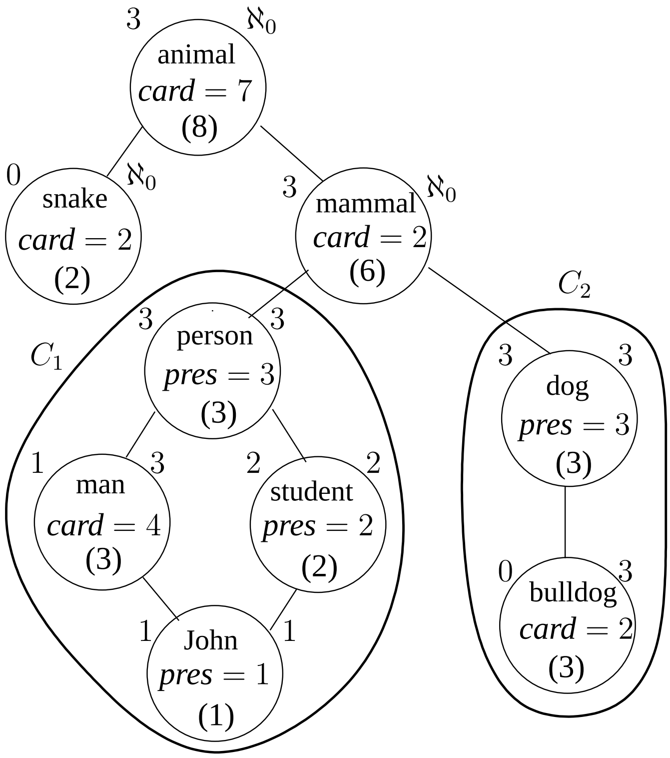

Figure 2: Example of the set assignment algorithm

Each node in the graph represents a noun, with the value of *pres*(n) (or *card*(n) if *pres*(n) $= \phi$). The values of $mincard$ and $maxcard$ are above each node on left and right respectively. The number in parenthesis (at the bottom of each node) is the cardinality of the set ultimately assigned to the noun.

C_2. In C_1, the maximum *pres* is 3, and thus $m_1 = 3$, and three fresh elements are assigned as $S_1 = \{1, 2, 3\}$ in Step 8. Similarly, $m_2 = 3$, and three different elements are issued for $S_2 = \{4, 5, 6\}$. In Step 9, the nodes are assigned sets according to their $maxcard$ value. Thus, for C_1: $S(person) = S(man) = \{1, 2, 3\}$, $S(student) = \{1, 2\}$, and $S(John) = \{1\}$, and for C_2: $S(dog) = S(bulldog) = \{4, 5, 6\}$. Finally, in Step 11, sets are assigned to "free" nodes (unaffected by presuppositions), and thus $S(snake) = \{7, 8\}$, $S(mammal) = \{1, 2, 3, 4, 5, 6\}$, and $S(animal) = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

Example 1 (cont.): In sentence (17), the nouns affected by presuppositions are *teacher* and *person*. Both are assigned the same three elements: $S(person) = S(teacher) = \{1, 2, 3\}$. The noun *dog* is then assigned *card*$(dog) = 2$ fresh elements, thus $S(dog) = \{4, 5\}$. The $mincard$ values are $mincard(person) = 3$ and $mincard(teacher) = mincard(dog) = 0$.

That the set assignment algorithm satisfies the requirements in (18) is formally claimed below.

Claim 9 *For given X, card, fincard, pres, and cont, Algorithm 1 generates a set assignment that satisfies (18) if and only if such a set assignment function exists. Otherwise, Algorithm 1 outputs "impossible".*

For the proof see Appendix A.

3.5 Stage 2: Checking scope dominance

Stage 1 computes for each noun n in the sentence a set $S(\mathsf{n})$ and a value $mincard(\mathsf{n})$. In checking whether the sentence exhibits scope dominance, the set $S(\mathsf{n})$ is used as indicative for the "most general" behavior of the determiner-noun constructions d n in which n appears. The value $mincard(\mathsf{n})$ is used to test whether these NPs are trivial under any model. Using these two values for the nouns in the sentence, we shall see that it is possible, under certain limitations, to determine whether the sentence exhibits scope dominance.

Step 2 of Algorithm 2 below, which is its main part, computes the features (U), (FLT), (FIN), (DCC), (TRIV_0) and (TRIV_1) of the NPs in the sentence under the set assignment S of Stage 1. In addition, this step computes the following two features of NPs, using the $mincard$ values of Stage 1:

$$\mathrm{TRIV}_0^{\neg}(NP) \overset{def}{\Leftrightarrow} \mathrm{TRIV}_0([\![NP]\!]_{\mathcal{M}}) \text{ does not hold for any model } \mathcal{M}.$$

$$\mathrm{TRIV}_1^{\neg}(NP) \overset{def}{\Leftrightarrow} \mathrm{TRIV}_1([\![NP]\!]_{\mathcal{M}}) \text{ does not hold for any model } \mathcal{M}.$$

To determine these six features for simple NPs of the form *DetP N*, Sub-step 2a directly uses the lexical information on determiners in Table 2. In order to determine the same features for NP conjunctions and disjunctions, Sub-step 2a also keeps for each simple noun phrase its minimal live-on set under S, which is denoted $\mathsf{mlos}[\mathrm{NP}]$. This, due to the conservativity of determiners, is simply the set that S assigns to the noun within the NP.

Sub-steps 2b and 2c use Facts 7 and 8 for computing the above properties for NP conjunctions and disjunctions. The minimal live-on sets of constituent NPs are used to determine whether Fact 8 applies, and they are recursively computed (correctly) by unioning the live-on sets of constituent NPs.

The rest of the algorithm is a direct application of Theorem 3 for deciding whether the sentence exhibits scope dominance under the computed set assignment S. Whenever the minimal live-on sets of coordinated NPs are disjoint under S, it follows from Fact 8 that the quantifiers in the sentence (under S) exhibit the relevant properties iff Step 2 assigns these properties to them. On the other hand, when non-disjoint live-on sets are generated, it is only guaranteed (by Fact 7) that the properties are exhibited if (but not only if) Step 2 computes them. Consequently under *non*-disjoint live-on sets, the set assignment S can be used only as indicative for existence of scope dominance in the sentence, and not for non-existence of scope dominance. Thus, when the live-on sets are not disjoint, a "yes" answer is given if the computed properties support scope dominance, but if the computed properties do not support scope dominance, an "indeterminate" answer is given.

The algorithm for checking scope dominance is given below.

Algorithm 2 (Checking scope dominance):
Let S be a set assignment function satisfying the conditions in (18).

1. Initialize disjoint $:= 1$. $\backslash\backslash$ a flag for marking that all coordinated quantifiers live on disjoint sets.

2. For each parse of the input sentence according to the grammar in Figure 1, assign the features $\mathrm{TRIV}_0, \mathrm{TRIV}_1, \mathrm{TRIV}_0^{-}, \mathrm{TRIV}_1^{-}, \mathrm{U}, \mathrm{FLT}, \mathrm{FIN}, \mathrm{DCC}$ and a set mlos to the constituent NPs in the sentence according to the following rules.

   (a) For $NP \rightarrow DetP\ N$:
   Let d be the determiner represented by $DetP$, with the given lexical values g_{det} and inf_{det}. Let n be the noun represented by N, let c be $|S(\mathsf{n})|$, and let m be

$mincard(\mathsf{n})$. Then:

$$
\begin{aligned}
\mathrm{TRIV}_0[NP] &\overset{def}{\Leftrightarrow} \begin{cases} inf_{det} = \mathsf{all\_but\_lt\_0} & c = \aleph_0 \\ g_{det}(c) > c & \text{otherwise} \end{cases} \\
\mathrm{TRIV}_1[NP] &\overset{def}{\Leftrightarrow} \begin{cases} inf_{det} = \mathsf{at\_least\_0} & c = \aleph_0 \\ g_{det}(c) < 1 & \text{otherwise} \end{cases} \\
\mathrm{TRIV}[NP] &\overset{def}{\Leftrightarrow} \mathrm{TRIV}_0[NP] \vee \mathrm{TRIV}_1[NP] \\
\mathrm{U}[NP] &\overset{def}{\Leftrightarrow} \mathrm{TRIV}[NP] \vee \\
&\quad \begin{cases} inf_{det} \in \{\ \mathsf{at\_least\_1}, \ \mathsf{infinitely\_many}\} & c = \aleph_0 \\ g_{det}(c) = 1 & \text{otherwise} \end{cases} \\
\mathrm{FLT}[NP] &\overset{def}{\Leftrightarrow} \mathrm{TRIV}[NP] \vee \\
&\quad \begin{cases} inf_{det} \in \{\ \mathsf{all\_but\_lt\_1}, \ \mathsf{all\_but\_fin\_many}\} & c = \aleph_0 \\ g_{det}(c) = c & \text{otherwise} \end{cases} \\
\mathrm{FIN}[NP] &\overset{def}{\Leftrightarrow} \mathrm{TRIV}[NP] \vee c < \aleph_0 \vee \\
&\quad inf_{det} \in \{\mathsf{at\_least\_}n : n \in \mathbb{N}\} \\
\mathrm{DCC}[NP] &\overset{def}{\Leftrightarrow} \mathrm{TRIV}[NP] \vee c < \aleph_0 \vee \\
&\quad inf_{det} \in \{\mathsf{all\_but\_lt\_}n : n \in \mathbb{N}\} \\
\mathrm{TRIV}_0^{\neg}[NP] &\overset{def}{\Leftrightarrow} g_{det}(m) \leq m \\
\mathrm{TRIV}_1^{\neg}[NP] &\overset{def}{\Leftrightarrow} g_{det}(m) \geq 1 \\
\mathsf{mlos}[NP] &\overset{def}{=} S(\mathsf{n})
\end{aligned}
$$

(b) For $NP \rightarrow NP_1$ and NP_2:

i. If $\mathsf{mlos}[NP_1] \cap \mathsf{mlos}[NP_2] \neq \emptyset$, set disjoint $:= 0$.

ii. If either $\mathrm{TRIV}_0[NP_1]$ or $\mathrm{TRIV}_0[NP_2]$ is 1, then $\mathrm{TRIV}_0[NP] := \mathrm{TRIV}_1^{\neg}[NP] := \mathrm{U}[NP] := \mathrm{FLT}[NP] := \mathrm{FIN}[NP] := \mathrm{DCC}[NP] := 1$, $\mathrm{TRIV}_1[NP] := \mathrm{TRIV}_0^{\neg}[NP] := 0$, and $\mathsf{mlos}[NP] := \emptyset$.

iii. Otherwise, if $\mathrm{TRIV}_1[NP_1]$ is 1, then the features of NP_2 are copied to NP.

iv. Otherwise, if $\mathrm{TRIV}_1[NP_2]$ is 1, then the features of NP_1 are copied to NP.

v. Otherwise:

$$\begin{aligned}
\mathrm{TRIV}_0[NP] &:= \mathrm{TRIV}_1[NP] := 0 \\
\mathrm{U}[NP] &:= 0 \\
\mathrm{FLT}[NP] &:= \mathrm{FLT}[NP_1] \wedge \mathrm{FLT}[NP_2] \\
\mathrm{FIN}[NP] &:= \mathrm{FIN}[NP_1] \wedge \mathrm{FIN}[NP_2] \\
\mathrm{DCC}[NP] &:= \mathrm{DCC}[NP_1] \wedge \mathrm{DCC}[NP_2] \\
\mathrm{TRIV}_0^{\neg}[NP] &:= \mathrm{TRIV}_0^{\neg}[NP_1] \wedge \mathrm{TRIV}_0^{\neg}[NP_2] \\
\mathrm{TRIV}_1^{\neg}[NP] &:= \mathrm{TRIV}_1^{\neg}[NP_1] \vee \mathrm{TRIV}_1^{\neg}[NP_2] \\
\mathsf{mlos}[NP] &:= \mathsf{mlos}[NP_1] \cup \mathsf{mlos}[NP_2]
\end{aligned}$$

(c) For $NP \rightarrow NP_1$ or NP_2:

i. If $\mathsf{mlos}[NP_1] \cap \mathsf{mlos}[NP_2] \neq \emptyset$, set disjoint $:= 0$.

ii. If either $\mathrm{TRIV}_1[NP_1]$ or $\mathrm{TRIV}_1[NP_2]$ is 1, then $\mathrm{TRIV}_1[NP] := \mathrm{TRIV}_0^{\neg}[NP] := \mathrm{U}[NP] := \mathrm{FLT}[NP] := \mathrm{FIN}[NP] := \mathrm{DCC}[NP] := 1$, $\mathrm{TRIV}_0[NP] := \mathrm{TRIV}_1^{\neg}[NP] := 0$, and $\mathsf{mlos}[NP] := \emptyset$.

iii. Otherwise, if $\mathrm{TRIV}_0[NP_1]$ is 1, then the features of NP_2 are copied to NP.

iv. Otherwise, if $\mathrm{TRIV}_0[NP_2]$ is 1, then the features of NP_1 are copied to NP.

v. Otherwise,

$$\begin{aligned}
\mathrm{TRIV}_0[NP] &:= \mathrm{TRIV}_1[NP] := 0 \\
\mathrm{FLT}[NP] &:= 0 \\
\mathrm{U}[NP] &:= \mathrm{U}[NP_1] \wedge \mathrm{U}[NP_2] \\
\mathrm{FIN}[NP] &:= \mathrm{FIN}[NP_1] \wedge \mathrm{FIN}[NP_2] \\
\mathrm{DCC}[NP] &:= \mathrm{DCC}[NP_1] \wedge \mathrm{DCC}[NP_2] \\
\mathrm{TRIV}_0^{\neg}[NP] &:= \mathrm{TRIV}_0^{\neg}[NP_1] \vee \mathrm{TRIV}_0^{\neg}[NP_2] \\
\mathrm{TRIV}_1^{\neg}[NP] &:= \mathrm{TRIV}_1^{\neg}[NP_1] \wedge \mathrm{TRIV}_1^{\neg}[NP_2] \\
\mathsf{mlos}[NP] &:= \mathsf{mlos}[NP_1] \cup \mathsf{mlos}[NP_2]
\end{aligned}$$

3. Let NP_1 and NP_2 be the parse trees of the subject and the object NPs in the sentence.

(a) If $\mathrm{TRIV}_1^{\neg}[NP_1] = 0$ and $\mathrm{TRIV}_0^{\neg}[NP_2] = 0$, output “no” and stop.

(b) Otherwise, compute:

$$\mathrm{val}[S] := (\mathrm{FIN}[NP_1] \vee \mathrm{DCC}[NP_2]) \wedge (\mathrm{U}[NP_1] \vee \mathrm{FLT}[NP_2])$$

4. If disjoint $= 1$: if val$[S] = 1$ output “yes”, otherwise output “no”.

5. If disjoint $= 0$: if val$[S] = 1$ output “yes”, otherwise output “indeterminate”.

Example 1 (cont.): In sentence (17), under the set assignment from Stage 1, the NP “*All of the three persons*” satisfies $\mathrm{TRIV}_0^{\neg}$, $\mathrm{TRIV}_1^{\neg}$, (FLT), (FIN), and (DCC); “*more than two teachers*” satisfies $\mathrm{TRIV}_1^{\neg}$, (FLT), (FIN), and (DCC); and “*some dog*” satisfies $\mathrm{TRIV}_1^{\neg}$, (U), (FIN), and

(DCC). Based on the sets and denotations assigned above, the NPs "*more than two teachers*" and "*some dog*" live on the sets $\{1, 2, 3\}$ and $\{4, 5\}$ respectively. Therefore, their *or* coordination is an NP that satisfies $\mathrm{TRIV}_1^{\neg}$, (FIN) and (DCC) and no other of the relevant features. Applying Theorem 3, we conclude that under the set assignment from Stage 1, the object quantifier is dominant over the subject quantifier, but not vice versa. As will be shown below, the way that the set assignment is chosen guarantees that the OWS reading entails the ONS reading, but not vice versa, as it is indeed the case.

The following two claims state that whenever Stage 2 gives a determinate ("yes"/"no") response, this is the correct response, and the sentence exhibits or does not exhibit scope dominance accordingly. Claim 10 follows directly from the set assignment stage and Theorem 3. The proof of Claim 11 is more intricate. Both proofs are given in the appendix.

Claim 10 *Given a parse of the input sentence NP_1 V NP_2 and the domain E, set assignment S, and $mincard$ from Algorithm 1, if Algorithm 2 returns a "no" response, then the parse of the input sentence does not exhibit scope dominance.*

Claim 11 *Given a parse of the input sentence NP_1 V NP_2, and the domain E and set assignment S from Algorithm 1, if Algorithm 2 responds with a "yes" response, then the parse of the input sentence exhibits scope dominance.*

3.6 Stage 3: Generating a counterexample

In this stage we generate a finite model $\mathcal{M}$, under which the ONS reading of the input sentence does not entail its OWS reading, if such finite model exists. This stage is performed only if the result of the previous stage is "no". According to Claim 10 above, this occurs only if there is a model where the sentence does not support scope dominance. Such a model is either simply the model that is generated in Stage 1, or (in case the condition in Step 3a of Algorithm 2 holds), it is a model where the subject and object denote trivial quantifiers.

An algorithm for generating such model is given below. We assume a simple procedure **handle_triv** that assigns features to coordinations involving trivial quantifiers as in Step 2 of Algorithm 2. We also assume that the TRIV_0 and TRIV_1 features from Stage 2 are kept. The algorithm itself works as follows. Step 1 handles the case where the counterexample involves trivial subject and object. Step 2 ensures a finite model by re-running the previous stages if one of the determiners in the sentence has an infinite *lgc*. If this is the case, all such determiners are reassigned a *zero lgc* value. The assignment of a finite model is a direct result of requirement (18iv) on the set assignment algorithm. This model is indicative of scope dominance with regard to *finite* domains, because we assume that every determiner d in the lexicon with $lgc(\mathsf{d}) = \aleph_0$ is trivial over finite domains. Hence the behavior of such determiners with $lgc(\mathsf{d}) = 0$ is indicative of their behavior over finite domains. Step 3 generates a counter-example based on the direct proof of Fact 1, and the properties of conjunction and disjunction of Fact 8. Recall that the denotation of the verb for the counter-example is a binary relation $R = (\{a\} \times B_1) \cup ((A_1 \setminus \{a\}) \times B_2)$, where A_1 is a minimal set in $[\![NP_1]\!]$ s.t. $|A_1| > 1$, $a \in A_1$ is some element, and B_1, B_2 are sets such that $B_1, B_2 \in [\![NP_2]\!]$ but $B_1 \cap B_2 \notin [\![NP_2]\!]$. This relation is assigned as the denotation of the main verb in Step 4.

Algorithm 3 (Generating a counter-example):
Assume a parse for the input sentence *NP1 V NP2* for which Algorithm 2 responded "no".

1. If the condition in Step 3a of Algorithm 2 holds, return the following model $\mathcal{M} := \langle E, [\![\cdot]\!]\rangle$, where $E := \{1, 2, \ldots, \max_{\mathsf{n} \in X} mincard(\mathsf{n})\}$, and

$$[\![\mathsf{x}]\!] := \begin{cases} \emptyset & \mathsf{x} \text{ is the verb } V \text{ in the sentence} \\ \{1, 2, \ldots, mincard(\mathsf{x})\} & \text{otherwise } (\mathsf{x} \text{ is a noun}). \end{cases}$$

2. Otherwise, if the domain assigned in Stage 1 is infinite, re-run the preparation stage and Stages 1 and 2, while assigning $LGC(\mathsf{d}) = 0$ for any determiner d for which $LGC(\mathsf{d}) = \aleph_0$. If Stage 2 did not respond with "no", stop.

3. Let S be the set assignment from the latest run of Stage 1. Recursively construct the sets $A[NP]$, $B_1[NP]$, and $B_2[NP]$ for each NP in the sentence as listed below. If a set cannot be constructed at some step, it is marked as undefined.

   (a) If NP is of the form $D\ N$, then let n be the noun represented by N, and let d be the determiner represented by D, with a lexical value g_{det}. $A[NP]$ is an arbitrary subset of $S(\mathsf{n})$ of cardinality $g_{det}(|S(\mathsf{n})|)$, and the sets $B_1[NP]$ and $B_2[NP]$ are arbitrary subsets of $S(\mathsf{n})$ of cardinality $g_{det}(|S(\mathsf{n})|)$, s.t. $B_1 \neq B_2$, if such sets exist (otherwise – they are undefined).

   (b) If NP is of the form *NP1 and NP2*:

   - If either $\mathrm{TRIV}[NP1]$ or $\mathrm{TRIV}[NP2]$ is 1, call **handle_triv(*NP1 and NP2*)**. Otherwise:
     - $A[NP] := A[NP1] \cup A[NP2]$.
     - If $B_1[NP1]$ is defined then: $B_1[NP] := B_1[NP1] \cup \mathsf{mlos}[NP2]$. Otherwise: $B_1[NP] := B_1[NP2] \cup \mathsf{mlos}[NP1]$, and similarly for $B_2[NP]$.

   (c) If NP is of the form *NP1 or NP2*:

   - If either $\mathrm{TRIV}[NP1]$ or $\mathrm{TRIV}[NP2]$ is 1, call **handle_triv(*NP1 or NP2*)**. Otherwise:
     - $A[NP]$ is the set between $A[NP1]$ and $A[NP2]$ which is of bigger cardinality.
     - $B_1[NP] := \mathsf{mlos}[NP1]$.
     - $B_2[NP] := \mathsf{mlos}[NP2]$.

4. Let a be some element in $A[NP1]$. Return a model $\mathcal{M} := \langle E, [\![\cdot]\!]\rangle$, where E is the domain returned by the latest run of Stage 1, and $[\![\cdot]\!]$ assigns to each noun n in X the set $S(\mathsf{n})$, and for the main verb V in the sentence, $[\![V]\!] := (\{a\} \times B_1[NP2]) \cup ((A[NP1] \setminus \{a\}) \times B_2[NP2])$.

Example 1 (cont.): The sets A, B_1, B_2, and mlos generated for the NPs in sentence (17) are displayed in Figure 3, with the resulting denotation for the verb *admire*. Indeed, this denotation for the verb *admire* with the assigned sets from Stage 1 demonstrate a situation where the ONS reading is true while the OWS reading is false.

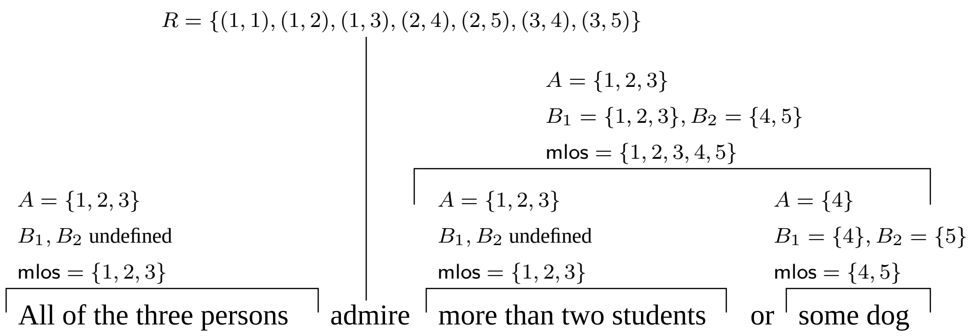

Figure 3: Generation of a counter-example

In the following correctness claim for Algorithm 3, we call a global determiner D *trivial for finite domains* if for all $A \subseteq E$ s.t. E is finite: $D_E(A)$ is a trivial quantifier. Clearly, the two determiners in the lexicon (*finitely many* and *all but finitely many*) for which $lgc = \aleph_0$ are both trivial for finite domains.

Claim 12 *Let* NP_1 V NP_2 *be a sentence where every determiner* d *for which* $lgc(\mathsf{d}) = \aleph_0$ *is trivial for finite domains. Assume the result from Stage 2 was "no", and there exists a finite model* $\mathcal{M}$ *that satisfies the conditions in (11) s.t. for* $Q_1 = [\![NP_1]\!]_{\mathcal{M}}$, $Q_2 = [\![NP_2]\!]_{\mathcal{M}}$, *and* $R = [\![V]\!]_{\mathcal{M}}$*:* $Q_1 Q_2 R$ *is true, but* $Q_2 Q_1 R^{-1}$ *is false. Then, Stage 3 finds such a model* $\mathcal{M}$.

3.7 Limitations of the algorithm and a heuristic improvement

The main limitation of the proposed procedure for computation of scope dominance appears when the set assignment algorithm generates non-disjoint sets for nouns that appear in the same coordination. As we remarked in Section 3.5, if in such cases the generated model is classified as refuting scope dominance, this is not necessarily indicative of scope dominance across models. Therefore, in such cases the algorithm fails to decide on whether the sentence exhibits scope dominance. For instance, consider the following sentence, with the noun phrase of (16):

(19) Some student saw every teacher and at least two authors.

Provided that there are at least three people who are both teachers and authors, this sentence exhibits scope dominance. However, our algorithm fails to identify this, as it is based on the conditions in Fact 7, which are not necessary conditions for scope dominance when the live on sets of two quantifiers in a coordination are not disjoint.

A simple sufficient condition for the set assignment algorithm to assign disjoint sets to any two nouns is the following.

Fact 13 *Algorithm 1 satisfies* $S(\mathsf{n}) \cap S(\mathsf{m}) = \emptyset$ *for any two different nodes* n, m *in* G *whenever the following conditions hold:*

- n *and* m *have no common ancestor in* G*; and*
- *For all nodes* n', m' *in* M *(i.e.* $maxcard(\mathsf{n}'), maxcard(\mathsf{m}') < \text{FIN}$*) s.t.* ($\mathsf{n}' = \mathsf{n}$ *or* n' *is an ancestor of* n) *and* ($\mathsf{m}' = \mathsf{m}$ *or* m' *is an ancestor of* m)*:* n' *and* m' *are not in the same connected component* C_i *of* G_M.

Non-disjoint sets may be assigned due to two major reasons. One reason is that a containment relation directly forces two sets in the same NP coordination not to be disjoint. For example, consider the following sentence.

(20) Some student and every person saw at least two dogs.

Since any student is a person, these two nouns must be assigned non-disjoint sets. Thus, we cannot deduce the semantic features of the subject NP under *any* set assignment that is indicative of the behavior of the noun phrase *some student*. A different kind of problem arises when a noun has a cardinality presupposition, and two nouns whose denotation must be contained in it are forced not to be disjoint by the set assignment algorithm. For example, consider the following sentence.

(21) The four persons saw some man and some woman.

In this case, Algorithm 1 assigns the same set to the nouns *man*, *woman*, and *person*, and thus the sets for *man* and *woman* are not disjoint. This situation can be improved: in this case there is no reason not to let *man* and *woman* denote disjoint doubleton sets, which are still indicative of scope dominance, as the *card* value of *some* is 2. Of course, other cases may be more problematic. If *the four persons* in (21) is replaced by *the three persons*, the sets assigned to *man* and *woman* cannot be disjoint while satisfying the *card* requirement.

As mentioned above, dealing with conjunctions of quantifiers that require non-disjoint minimal live-on sets in order to achieve an indicative model is an open problem. However, cases like (21), which do not raise such a non-disjointness requirement, can be handled using the following simple heuristics. This heuristics is designed to allow more disjoint sets to be generated. In the cases where it fails to meet maximum cardinality requirements, the set assignment stage is restarted using Algorithm 1.

Algorithm 4 (Set Assignment Heuristics):

1. Perform Steps 1–6 of Algorithm 1.
2. Assign a set $S(\mathsf{n})$ to each node n in G, in the order of L_{subset}:
   (a) $U := \bigcup_{\text{there is an arc from } \mathsf{m} \text{ to } \mathsf{n}} S(\mathsf{m})$
   (b) if $|U| > maxcard(\mathsf{n})$, restart with Algorithm 1.
   (c) $S(\mathsf{n}) := U$
   (d) let $asscard(\mathsf{n}) := \min\{\textit{card}(\mathsf{n}), maxcard(\mathsf{n})\}$. If $maxcard(\mathsf{n}) = \text{FIN}$ and $\textit{card}(\mathsf{n}) = \aleph_0$, assign $asscard(\mathsf{n}) := \textit{fincard}(\mathsf{n})$ instead.
   (e) if $asscard(\mathsf{n}) > |U|$, add $asscard(\mathsf{n}) - |U|$ fresh elements to $S(\mathsf{n})$.

It is easy to see that this heuristics is correct when Step 2b in it does not fail. To improve the implementation further, Step 2 could be applied separately for each equivalence class. In this way, failure in Step 2b for a node n would only entail resorting to Algorithm 1 for the nodes in n's component within the graph, and not for the whole graph. In experimenting with the working implementation, the heuristics above proves rather useful in overcoming some of the main shortcomings of the more principled set assignment algorithm above.

4 Notes on the demo implementation

A system that is based on the algorithms of the previous section was implemented in Standard ML of New Jersey,[10] and a web interface to the system (illustrated in Figure 4) is accessible at `http://yeda.cs.technion.ac.il/~alon_a/`, where the full source code is also available.

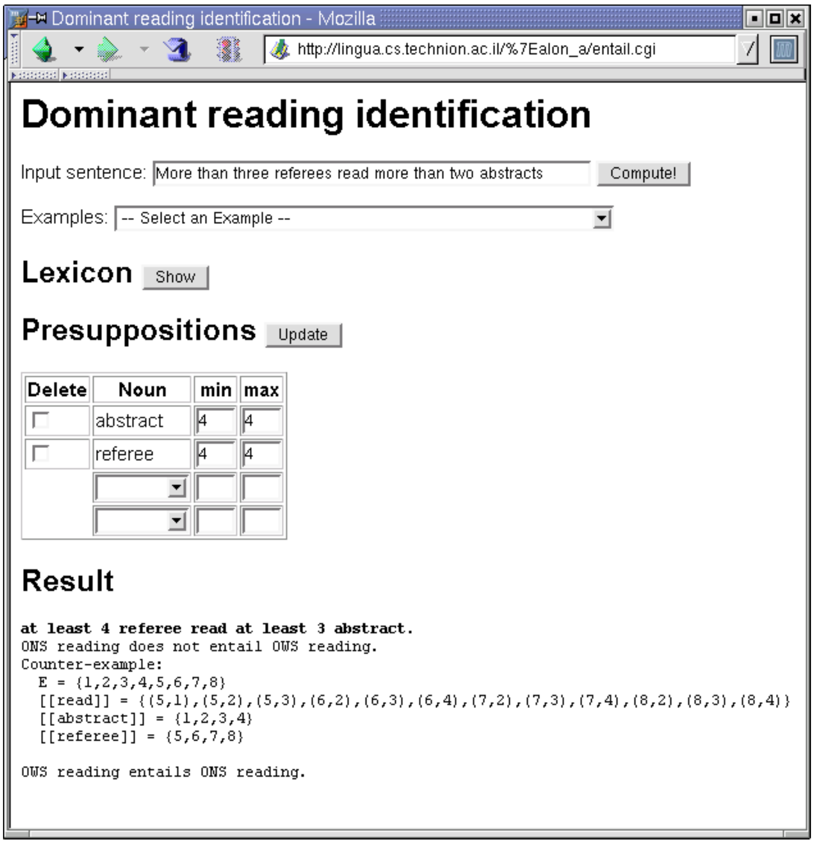

Figure 4: Screenshot of the system's web interface

The actual grammar that is used in the implementation is given below. This is a simplified version of the grammar in Figure 1, which also includes proper names. The parsing of determiners such as "*at least three*" is performed at the lexical level.

1. $\mathrm{S} \rightarrow \mathrm{Q}_1\ \mathrm{V}\ \mathrm{Q}_2$
2. $\mathrm{Q} \rightarrow \mathrm{D\ N'}\ |\mathrm{D\ OF\ NUM\ N'}\ |\mathrm{PN}\ |\mathrm{Q}_1\ \mathrm{C}\ \mathrm{Q}_2$
3. $\mathrm{N'} \rightarrow \mathrm{N}$

The lexicon in the system is:

D — all but finitely many, some and every, some or every, some, a, an, every, all, each, most, infinitely many, at least half, the, at least n, more than n, all but at most n, all but n, the n

[10] This is a variant of standard ML (Milner et al. (1997)) that is available at `http://www.smlnj.org/`.

N	dog, cat, student, teacher, person, man, woman, city, priest, animal, mamal, fish, referee, abstract, officer, outpost, circle, square, dot, shape, child
PN	john, mary, bill, sue, dave, eve, alice, bob
V	saw, visited, graded, likes, occupied, occupies, admire, admires, read, reads, like, love, loves, contain, contains
C	and, or
NUM	one, two, three, four, five, six, seven, eight, nine, ten, 0, 1, 2, ...
OF	of the

Infinite sets are not directly represented in the system, but rather are implemented as a list of elements included in the set, combined with a list of all subsets of the set. This representation is easily generated during the set assignment stage, while allowing to check for disjointness and unify sets, as required for the implementation of Stage 2.

The set assignment algorithm that is implemented is the heuristic improvement of Algorithm 1 that was described in Subsection 3.7. The implementation allows the user to specify additional cardinality restrictions on noun denotations. These are minimum and/or maximum cardinalities for lexical nouns. These values are collected in the preparation stage and used to initialize $mincard$ and $maxcard$ for the set assignment stage.

More details about the implementation appear in the website of the system.

5 Conclusions and directions for further research

In this paper, some recent results on logical relations with scopally ambiguous sentences were used in order to develop a system that computes such relations in a small fragment of English. As we saw, the logical characterization of scope dominance between GQs in one model can be used as the key for identifying entailments between different readings of a sentence. In addition to this characterization, the algorithm that was developed takes into account containment relations between nominals and presuppositional/contextual information about cardinality. The main part of the algorithm, the set assignment stage, uses this information in order to construct a model that is indicative of scope dominance – scope dominance between the quantifiers in this model implies (under certain conditions) an entailment between the different readings of the sentence. Well-established properties of determiners from GQ theory such as Extension, Isomorphism-invariance and Conservativity, in addition to a new assumption concerning the *downward consistency* of natural language determiners, were crucial for establishing the correctness of the algorithm. The SML-based system that we developed is a demo that illustrates one possible way to implement the main ideas of this work.

We believe that the possible contribution of this work is in two different but inter-related research avenues. One is the logical study of scope dominance between generalized quantifiers. As mentioned in Section 2, previous works have studied only partial aspects of this general question. The characterization of scope dominance between quantifiers that are not necessarily upward monotone is still an open question, which has recently received some more attention in Ben-Avi and Winter (2004). Another avenue for pursuing our approach is to study further the usage of entailments between readings as a way for improving underspecified representations and techniques for reasoning under ambiguity. One especially interesting question is entailments between different readings of ambiguous sentences that are ambiguous but not with respect to their quantifier scope. For instance, in a sentence like *Mary and John or Bill smiled* one of the two bracketings (*Mary and [John or Bill]*) for the coordination generates a stronger reading than the other bracketing. The system that we developed cannot find such dominant readings. This calls for a more general characterization of entailments between readings of other kinds of ambiguity besides scope ambiguity.

This research leaves some other open questions as well. One of them is the characterization of scope dominance between NP coordinations that involve non-disjoint live-on sets. Another question involves scope dominance relations that appear due to logical properties of n-ary predicates. For instance, in the sentence *at least one of the two students is taller than every girl*, the OWS reading entails the ONS reading (only) due to the transitivity of the relation *taller than*. Since such properties of predicates were not taken into account in this paper, the system that we developed does not detect such scope dominance relations.

The questions that were mentioned above and related ones pose interesting challenges for formal and computational semantics of natural language. We hope to have shown that these are tenable challenges, and that by addressing them we may improve our understanding of computable logical relations in natural language that go beyond first order logic.

Acknowledgments
The authors were partly supported by grant no. 1999210 ("Extensions and Implementations of Natural Logic") from the United States-Israel Binational Science Foundation (BSF), Jerusalem, Israel. The second author was also supported by an NWO grant for visiting the Utrecht University in summer 2003. We are grateful to Gilad Ben-Avi, Nissim Francez, Fred Landman and Ya'acov Peterzil for remarks and discussions. The second author is also grateful to the ILLC of the University of Amsterdam, where part of this research was carried out. We are grateful to two anonymous reviewers of *Research on Language and Computation* for their helpful comments on this paper.

A Appendix: proofs of correctness claims

Proof of Claim 9 If Algorithm 1 outputs "impossible", it is easy to see that there are presuppositions that conflict with the containment relation (Step 1), hence there is no set assignment that satisfies (18i) and (18ii). Otherwise, we shall see that Algorithm 1 finds a set assignment S that satisfies (18).

1. For Clause (18i), let n and m be two elements in X. Assume $\langle \mathsf{n}, \mathsf{m}\rangle \in \overline{\mathit{cont}}$. If both n and m were assigned sets in Step 9, n and m must be of the same connected component C_i, and by Step 6, $|S(\mathsf{n})| = maxcard(\mathsf{n}) \leq maxcard(\mathsf{m}) = |S(\mathsf{m})|$, thus $S(\mathsf{n}) \subseteq S(\mathsf{m})$ (by Step 9). Otherwise, m was assigned in Step 11, and by the definition of U, $S(\mathsf{n}) \subseteq U \subseteq S(\mathsf{m})$.
2. For Clause (18ii), if $\mathit{pres}(\mathsf{n}) \in \mathbb{N}$, then $maxpres(\mathsf{n}) = \mathit{pres}(\mathsf{n})$ and from Steps 1 and 6, $maxcard(\mathsf{n}) = \mathit{pres}(\mathsf{n})$ as well, thus the correct number of elements is assigned in Step 9. For $\mathit{pres}(\mathsf{n}) = \text{FIN}$, the presupposition is satisfied because for all $\mathsf{m} \in X$ s.t. $\langle \mathsf{m}, \mathsf{n}\rangle \in \overline{\mathit{cont}}$: either $maxcard(\mathsf{m}) < \text{FIN}$ or (by Steps 6 and 10) $\mathit{card}'(\mathsf{m}) = \mathit{fincard}(\mathsf{m}) < \aleph_0$, and thus (by Step 11) $|S(\mathsf{m})| < \aleph_0$, and consequently $|S(\mathsf{n})| < \aleph_0$.
3. For Clause (18iii), let n be some element in X. Let S' be a set assignment that satisfies (18i) and (18ii). If $S(\mathsf{n})$ was assigned in Step 11, we know that $|S(\mathsf{n})| \geq \mathit{card}'(\mathsf{n})$. If $maxcard(\mathsf{n}) = \aleph_0$, then $|S(\mathsf{n})| \geq \mathit{card}'(\mathsf{n}) \geq \mathit{card}(\mathsf{n})$ as required. Otherwise, there is some $\mathsf{m} \in X$ s.t. $\langle \mathsf{n}, \mathsf{m}\rangle \in \overline{\mathit{cont}}$ and $\mathit{pres}(\mathsf{m}) = maxcard(\mathsf{n}) = \text{FIN}$. Thus for all domains E'' and for all set assignments S'' that satisfy (18i) and (18ii), $|S''(\mathsf{m})| < \aleph_0$, and (due to Step 11) $|S(\mathsf{n})| \geq \mathit{card}'(\mathsf{n}) = \mathit{fincard}(\mathsf{n})$ as required for this case in Clause (18iii).
   If $S(\mathsf{n})$ was assigned in Step 9, $|S(\mathsf{n})| = maxcard(\mathsf{n})$. By Step 6, there exists some $\mathsf{m} \in X$ s.t. $\langle \mathsf{n}, \mathsf{m}\rangle \in \overline{\mathit{cont}}$ and $\mathit{pres}(\mathsf{m}) = maxcard(\mathsf{n})$. Because S' satisfies (18i) and (18ii), $|S'(\mathsf{n})| \leq |S'(\mathsf{m})| = \mathit{pres}(\mathsf{m}) = maxcard(\mathsf{n}) = |S(\mathsf{n})|$, as required.

4. For Clause (18iv), let n be some element in X. If $S(\mathsf{n})$ was assigned in Step 9, $|S(\mathsf{n})| \leq maxcard(\mathsf{n}) < \aleph_0$, as required. Otherwise, $maxcard(\mathsf{n}) \in \{\text{FIN}, \aleph_0\}$ and $S(\mathsf{n})$ was assigned in Step 11. If $maxcard(\mathsf{n}) = \text{FIN}$, from (18ii), $|S(\mathsf{n})| < \aleph_0$. If $maxcard(\mathsf{n}) = \aleph_0$, assume for contradiction that $|S(\mathsf{n})| = \aleph_0$. By Step 11, there is some $\mathsf{m} \in X$ s.t. $\langle \mathsf{m}, \mathsf{n} \rangle \in \overline{cont}$ and $card'(\mathsf{m}) = \aleph_0$. By Step 10, there is some $\mathsf{r} \in X$ s.t. $card(\mathsf{r}) = maxcard(\mathsf{r}) = \aleph_0$, in contradiction to the assumption that $card(\mathsf{r}) < \aleph_0$.

5. For Clause (18v), assume that there exists $\mathsf{n} \in X$ s.t. $|S(\mathsf{n})| = \aleph_0$. $S(\mathsf{n})$ was assigned in Step 11 where $maxcard(\mathsf{n}) = card'(\mathsf{n}) = \aleph_0$. In this case, $card'(\mathsf{m}) = \aleph_0$ for all $\mathsf{m} \in X$ s.t. $maxcard(\mathsf{m}) \neq \text{FIN}$, and $card'(\mathsf{m}) = fincard(\mathsf{m})$ for all $\mathsf{m} \in X$ s.t. $maxcard(\mathsf{m}) = \text{FIN}$. Now Clause (18v) follows directly from the proof of Clause (18iii) applied to $card'$ instead of *card*.

6. For Clause (18vi), let n be some element in X. Let S' be a set assignment that satisfies (18i) and (18ii). If $mincard(\mathsf{n}) > 0$, then by Steps 1, 4 and 5, there is some m s.t. $\langle \mathsf{m}, \mathsf{n} \rangle \in \overline{cont}$ and $pres(\mathsf{m}) = mincard(\mathsf{n})$. As S' satisfies (18i) and (18ii), $|S'(\mathsf{n})| \geq |S'(\mathsf{m})| = pres(\mathsf{m}) = mincard(\mathsf{n})$.

7. For Clause (18vii), let $E' = \mathbb{N}$ and let S' be the following set assignment:

$$S'(\mathsf{n}) = \{1, 2, \ldots, mincard(\mathsf{n})\}.$$

Trivially, $|S'(\mathsf{n})| = mincard(\mathsf{n})$. For $\langle \mathsf{n}, \mathsf{m} \rangle \in cont$, $mincard(\mathsf{n}) \leq mincard(\mathsf{m})$, and thus $S'(\mathsf{n}) \subseteq S'(\mathsf{m})$. If $pres(\mathsf{n})$ is a number then $mincard(\mathsf{n}) = pres(\mathsf{n})$, and thus $|S'(\mathsf{n})| = pres(\mathsf{n})$. Thus, S' satisfies Clauses (18i) and (18ii), so Clause (18vii) is satisfied.

□

Proof of Claim 10 Let NP_1 and NP_2 be the parses of the subject and the object in the sentence. There are two cases in which the algorithm outputs "no":

1. Step 2 of the algorithm assigns $\text{TRIV}_1^-[NP_1] = \text{TRIV}_0^-[NP_2] = 0$. In this case, let $\mathcal{M}$ be a model in which the domain is $\mathbb{N}$ (the natural numbers) and the interpretation function $[\![\cdot]\!]$ satisfies: For each noun n in the sentence: $[\![\mathsf{n}]\!] = \{1, 2, \ldots, mincard(\mathsf{n})\}$. From condition (18vii) on $mincard$, the model $\mathcal{M}$ satisfies (11).

   By condition (18vi) on $mincard$ and the assignment of TRIV_0^- and TRIV_1^- in Step 2 of Algorithm 2, and downward consistency for (TRIV_0) and (TRIV_1), we conclude that $[\![NP_1]\!] = \wp(\mathbb{N})$ and $[\![NP_2]\!] = \emptyset$, thus the sentence does not exhibit scope dominance.

2. Step 2 of the algorithm assigns $\text{TRIV}_1^-[NP_1] \vee \text{TRIV}_0^-[NP_2] = 1$, $\text{val}[S] = 0$, and $disjoint = 1$. In this case, let $\mathcal{M}$ be a model in which the domain is E, and the interpretation function for the nouns assigns each noun n to the set $S(\mathsf{n})$ assigned by Algorithm 1. By requirements (18i) and (18ii) of Algorithm 1, $\mathcal{M}$ satisfies the requirements in (11).

   It is easy to see that the (U)/(FLT)/(FIN)/(DCC) properties assigned in Step 2a hold for the denotations under the model $\mathcal{M}$ of NPs of the form $DetPN$. Since $disjoint = 1$, all coordinations of NPs are between quantifiers in $\mathcal{M}$ that live on disjoint sets. Fact 8 thus holds for these quantifiers, and therefore each of the features in (U), (FLT), (FIN), (DCC), (TRIV_0) and (TRIV_1) holds for $[\![\text{NP}_1]\!]_{\mathcal{M}}$ and $[\![\text{NP}_2]\!]_{\mathcal{M}}$ iff they were assigned to NP_1 and NP_2 respectively. Applying Theorem 3, we conclude that $[\![\text{NP}_1]\!]_{\mathcal{M}}$ is not scopally dominant over $[\![\text{NP}_2]\!]_{\mathcal{M}}$.

□

Proof of Claim 11 Given that Algorithm 2 returned a "yes" response, we will show that the input sentence exhibits scope dominance. Let $\mathcal{M}$ be model in which the domain is $E_{\mathcal{M}}$ is the one generated by Algorithm 1, and the interpretation function assigns each noun n the set $S(\mathsf{n})$ that was generated by Algorithm 1. Let $\mathcal{M}'$ be a model that satisfies the requirements in (11). Let $S'(\mathsf{n}) = [\![\mathsf{n}]\!]_{\mathcal{M}'}$ for all $\mathsf{n} \in X$ be the set assignment function induced by $\mathcal{M}'$. Note that $S'(\mathsf{n})$ satisfies requirements (18i) and (18ii). We will now show that $[\![\text{NP}_1]\!]_{\mathcal{M}'}$ is dominant over $[\![\text{NP}_2]\!]_{\mathcal{M}'}$.

Assume that $E_{\mathcal{M}}$ is finite. Thus, Step 2 of Algorithm 2 assigns the features (FIN) and (DCC) to all the NPs in the sentence; and for every NP in the sentence, $[\![NP]\!]_{\mathcal{M}}$ trivially satisfies (FIN) and (DCC). Let us use the notation 'EXIST^1' and 'UNIV^0' to denote (over a given domain E) the classes of quantifiers $\text{EXIST} \cup \{\wp(E)\}$ and $\text{UNIV} \cup \{\emptyset\}$ respectively. By definition of g_{det} and finiteness of $E_{\mathcal{M}}$, if Step 2a assigns (U) or (FLT) to some simple NP (of the form DetP N), then $[\![\text{NP}]\!]_{\mathcal{M}}$ satisfies EXIST^1 or UNIV^0 respectively. Because S satisfies (18iii) and S' satisfies (18i) and (18ii), then for all $\mathsf{n} \in X$: $|S(\mathsf{n})| \geq \min\{card(\mathsf{n}), |S'(\mathsf{n})|\}$, or $|S(\mathsf{n})| \geq \min\{fincard(\mathsf{n}), S'(\mathsf{n})\}$

and $|S'(\mathsf{n})| < \aleph_0$. By downward consistency for (U) and (FLT), and definitions of *card* and *fincard*, we conclude that for every simple NP in the sentence, if $[\![NP]\!]_{\mathcal{M}}$ satisfies EXIST^1 or UNIV^0, then $[\![NP]\!]_{\mathcal{M}'}$ also satisfies EXIST^1 or UNIV^0 respectively. Because $E_{\mathcal{M}}$ is finite, for every simple NP in the sentence, if $[\![NP]\!]_{\mathcal{M}}$ satisfies (U) or (FLT), then $[\![NP]\!]_{\mathcal{M}'}$ satisfies EXIST^1 or UNIV^0 respectively. Therefore, if Step 2 of Algorithm 2 assigns (U) or (FLT) to some simple NP, $[\![NP]\!]_{M'}$ satisfies EXIST^1 or UNIV^0 respectively. By recursive application of Fact 7 to Steps 2b–2c of Algorithm 2, this is true for all NPs in the sentence. Since the algorithm answered with a "yes" response, it must assign (U) to NP_1 or (FLT) to NP_2, as well as TRIV_1^- to NP_1 or TRIV_0^- to NP_2. From condition (18vi) on $mincard$, $|S'(\mathsf{n})| \geq mincard(\mathsf{n})$ for all n. By the assignment of TRIV_0^- and TRIV_1^- in Step 2 of Algorithm 2 and downward consistency for (TRIV_0) and (TRIV_1), $[\![NP_1]\!]_{\mathcal{M}'} \neq \wp(E_{M'})$ or $[\![NP_2]\!]_{\mathcal{M}'} \neq \emptyset$. Thus, $[\![\mathrm{NP}_1]\!]_{\mathcal{M}'}$ is EXIST, $[\![\mathrm{NP}_2]\!]_{\mathcal{M}'}$ is UNIV, or exactly one of them is trivial, and thus the sentence exhibits scope dominance.

Now assume $E_{\mathcal{M}}$ is infinite. By the definitions of g_{det} and inf_{det}, if Step 2a of Algorithm 2 assigns a feature (U)/(FLT)/(FIN)/(DCC) to some simple NP, $[\![\mathrm{NP}]\!]_{\mathcal{M}}$ satisfies that feature. Because $E_{\mathcal{M}}$ is infinite and S satisfies (18v) and (18iii), for all $\mathsf{n} \in X$: $|S(\mathsf{n})| = \aleph_0 \geq |S'(\mathsf{n})|$, or $|S(\mathsf{n})| \geq \min\{\textit{fincard}(\mathsf{n}), |S'(\mathsf{n})|\}$ and $|S'(\mathsf{n})| < \aleph_0$. Therefore, by downward consistency, and the definition of *fincard*, if Step 2a of Algorithm 2 assigns a feature (U)/(FLT)/(FIN)/(DCC) to some simple NP, then $[\![NP]\!]_{\mathcal{M}'}$ also satisfies that feature. By recursive application of Fact 7 to Steps 2b–2c of Algorithm 2, this is true for the sub-parses of all the NPs in the parse of the sentence. Further, if Step 2 assigns TRIV_0^- or TRIV_1^- to some NP, by condition (18vi) on $mincard$, and downward consistency for (TRIV_0) and (TRIV_1), $[\![NP]\!]_{\mathcal{M}'}$ is not $\emptyset$ or $\wp(E_{\mathcal{M}'})$ respectively. Since the algorithm answered with a "yes" response, the fact that the sentence exhibits scope dominance is a direct result of Theorem 3. □

Proof of Claim 12 In this proof we refer to the re-run of Stage n in Step 2 of Algorithm 3 as Stage n_{fin}. To prove the claim there are two facts we need to establish:

A. In case Stage 2 generated an infinite domain, then its rerun (Stage 2_{fin}) will react by "no" if there is a finite model where the sentence exhibits scope dominance.

B. Whenever Stage 2 or 2_{fin} reacts with "no" and generates a finite domain, Stage 3 generates a finite model that illustrates lack of scope dominance in the input sentence.

Fact A: Assume for contradiction that Stage 2_{fin} was re-run and did not return "no". Let S be the set assignment from Stage 1 and let S_{fin} be the set assignment from Stage 1_{fin}. Note that the values of lgc affect (after the preparation stage) only the values of *card*. Thus, the runs of Stage 1 and Stage 1_{fin} would be exactly the same up to the end of Step 9. Therefore, for all n s.t. $maxcard(\mathsf{n}) < \mathrm{FIN}$: $S(\mathsf{n}) = S_{fin}(\mathsf{n})$. Let *card*($\mathsf{n}$) and *card*$'(\mathsf{n})$ be the *card*(n) and *card*$'(\mathsf{n})$ values from Stage 1, and let $card_{fin}(\mathsf{n})$ and $card'_{fin}(\mathsf{n})$ be the *card*(n) and *card*$'(\mathsf{n})$ values from Stage 1_{fin} respectively. It is easy to see that $card_{fin}(\mathsf{n}) \leq card(\mathsf{n})$ for all $\mathsf{n} \in X$. By Step 10 of Algorithm 1, $card'_{fin}(\mathsf{n}) \leq card'(\mathsf{n})$ for all $\mathsf{n} \in X$. By Step 11 of Algorithm 1, for all $\mathsf{n}, \mathsf{m} \in X$: if $S_{fin}(\mathsf{n}) \subseteq S_{fin}(\mathsf{m})$ then also $S(\mathsf{n}) \subseteq S(\mathsf{m})$.

We shall now prove that any disjoint sets assigned in Stage 1 must also be disjoint in Stage 1_{fin}. Let $\mathsf{n}, \mathsf{m} \in X$ be nouns s.t. $S_{fin}(\mathsf{n}) \cap S_{fin}(\mathsf{m}) \neq \emptyset$. If $S_{fin}(\mathsf{n})$ was assigned in Step 11, then because any set containing the freshly added elements contains U as well, there is some $\mathsf{n}' \in X$ s.t. $S_{fin}(\mathsf{n}') \cap S_{fin}(\mathsf{m}) \neq \emptyset$, $\emptyset \subset S_{fin}(\mathsf{n}') \subseteq S_{fin}(\mathsf{n})$ and either $S_{fin}(\mathsf{n}')$ was assigned in Step 9, or all elements in $S_{fin}(\mathsf{n}')$ are fresh elements (i.e. $U = \emptyset$). If n was assigned in Step 9, the same claim is trivially true for $\mathsf{n}' = \mathsf{n}$. Similarly, there is $\mathsf{m}' \in X$ s.t. $S_{fin}(\mathsf{n}') \cap S_{fin}(\mathsf{m}') \neq \emptyset$, $\emptyset \subset S_{fin}(\mathsf{m}') \subseteq S_{fin}(\mathsf{m})$ and either $S_{fin}(\mathsf{m}')$ was assigned in Step 9, or all elements in $S_{fin}(\mathsf{m}')$ are fresh elements. If both $S_{fin}(\mathsf{n}')$ and $S_{fin}(\mathsf{m}')$ were assigned in Step 9, then $S_{fin}(\mathsf{n}') = S(\mathsf{n}')$ and $S_{fin}(\mathsf{m}') = S(\mathsf{m}')$, and thus $S(\mathsf{n}') \cap S(\mathsf{m}') \neq \emptyset$. So also, $S(\mathsf{n}) \cap S(\mathsf{m}) \neq \emptyset$. Otherwise, $S_{fin}(\mathsf{n}')$ or $S_{fin}(\mathsf{m}')$ are assigned in Step 11, and thus only contain fresh elements. Therefore, $S_{fin}(\mathsf{n}') \subseteq S_{fin}(\mathsf{m}')$ or $S_{fin}(\mathsf{m}') \subseteq S_{fin}(\mathsf{n}')$, so there is some $\mathsf{r}' \in \{\mathsf{n}', \mathsf{m}'\}$ s.t. $S_{fin}(\mathsf{r}') \subseteq S_{fin}(\mathsf{n}') \cap S_{fin}(\mathsf{m}') \subseteq S_{fin}(\mathsf{n}) \cap S_{fin}(\mathsf{m})$. Because $|S(\mathsf{r}')| \geq |S_{fin}(\mathsf{r}')| > 0$ and $S(\mathsf{r}') \subseteq S(\mathsf{n}) \cap S(\mathsf{m})$, $S(\mathsf{n}) \cap S(\mathsf{m}) \neq \emptyset$.

Therefore, any disjoint sets assigned in Stage 1 must also be disjoint in Stage 1_{fin}. Thus, given that in Stage 2 all the coordinated quantifiers live on disjoint sets, it follows that the same holds for Stage 2_{fin}. Therefore, the result of Stage 2_{fin} must be "yes" (rather than "indeterminate"). Thus, by application of the proof of Claim 11 to finite models, and by the triviality for finite domains of the determiners for which the lgc value was modified, the sentence exhibits scope dominance for any finite model.

Fact B: We show that Stage 3 finds a finite model $\mathcal{M}$ for which Algorithm 2 returned a "no" response. Let Stage 2_{last} be Stage 2_{fin} (if there was a second run), or Stage 2 (if not). From Step 3 in Algorithm 2 and the finiteness of $\mathcal{M}$, we conclude that in Stage 2_{last}, either the feature (U) was not assigned to NP_1 and the feature (FLT) was not assigned to NP_2, or neither TRIV_0^- was assigned to NP_1, nor TRIV_1^- was assigned to NP_2. In the latter case, by the proof of Claim 10, $\mathcal{M}$ trivially demonstrates lack of dominance. For the former case, we will prove that the sets $A[NP_1]$, $B_1[NP_2]$, and $B_2[NP_2]$ fulfill the requirements for the relation from the proof of Fact 1, thus ensuring that Q_1Q_2R is true while $Q_2Q_1R^{-1}$ is false as required.

We will show first that for every NP for which the feature (U) was not assigned in Stage 2_{last}, the set $A[NP]$ is a minimal set in $Q = [\![\text{NP}]\!]_{\mathcal{M}}$ s.t. $|A[NP]| \geq 2$:

- By the assignment of (U) in Step 2a of Algorithm 2 and the definition of g_{det}, the cardinality of each minimal set in Q is at least 2, and thus the sets $A[NP]$ assigned by Step 3a in Algorithm 3 satisfies the requirement.
- For Step 3b in Algorithm 3, the union of minimal sets in two nontrivial quantifiers is a minimal set in their intersection, and assuming the coordinated quantifiers live on disjoint sets, the cardinality of the union must be at least 2, thus the set $A[NP]$ that is assigned by Step 3b satisfies the requirement $|A[NP]| \geq 2$.
- For Step 3c, each minimal set in the two quantifiers $Q_1 = [\![\text{NP}_1]\!]_{\mathcal{M}}$ and $Q_2 = [\![\text{NP}_2]\!]_{\mathcal{M}}$ is a minimal set in their union $Q_1 \cup Q_2$, and thus if at least one of the quantifiers has a minimal set A' s.t. $|A'| \geq 2$, then also the cardinality of the set $A[NP]$ assigned by Step 3c is at least 2, thus satisfying the requirement.

Now, we will show that for every NP for which the feature (FLT) was not assigned in Stage 2_{last}, the sets $B_1[NP]$ and $B_2[NP]$ are in $Q = [\![\text{NP}]\!]_{\mathcal{M}}$, but $B_1[NP] \cap B_2[NP] \notin Q$.

- By the assignment of (FLT) in Step 2a of Algorithm 2 and the definition of g_{det}, there are at least two distinct minimal sets in Q, and thus sets $B_1[NP], B_2[NP]$ assigned by Step 3a satisfy the requirement.
- For Step 3b in Algorithm 3, let Q_1 and Q_2 be $[\![NP_1]\!]_{\mathcal{M}}$ and $[\![NP_2]\!]_{\mathcal{M}}$ respectively. For every set $B \in Q_1$: $B \cup \mathsf{mlos}(Q_2) \in Q_1 \cap Q_2$, and because Q_1 and Q_2 live on disjoint sets, for every set $C \notin Q_1$: $C \cup \mathsf{mlos}(Q_2) \notin Q_2$, thus sets $B_1[NP], B_2[NP]$ assigned by Step 3b satisfy the requirement. The sets are not undefined because of the application of Fact 8 in Stage 2_{last}.
- For Step 3c, from nontriviality, for every Q', $\mathsf{mlos}(Q') \in Q'$, so $B_1[NP], B_2[NP] \in Q$. Because the quantifiers denoted by the NPs live on disjoint sets, $B_1[NP] \cap B_2[NP] = \emptyset$, and from nontriviality, $\emptyset \notin Q$, and thus the sets $B_1[NP], B_2[NP]$ assigned by Step 3c satisfy the requirement.

□

References

Altman, A., Peterzil, Y., and Winter, Y. (2002). Scope dominance with upward monotone quantifiers. Unpublished ms., Technion and Haifa University. Downloadable at `http://www.cs.technion.ac.il/~winter/`.

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

Ben-Avi, G. and Winter, Y. (2004). Scope dominance with monotone quantifiers over finite domains. To appear in *Journal of Logic, Language and Information*.

Comfort, W. W. and Negrepontis, S. (1974). *The Theory of Ultrfilters*. Springer-Verlag, Berlin.

Keenan, E. and Stavi, J. (1986). A semantic characterization of natural language determiners. *Linguistics and Philosophy*, 9:253–326.

Keenan, E. and Westerståhl, D. (1996). Generalized quantifiers in linguistics and logic. In Van Benthem, J. and Ter Meulen, A., editors, *Handbook of Logic and Language*. Elsevier, Amsterdam.

Milner, R., Tofte, M., Harper, R., and MacQueen, D. (1997). *The Definition of Standard ML – Revised*. MIT Press, Cambridge, Massachusetts.

Väänänen, J. and Westerståhl, D. (2001). On the expressive power of monotone natural language quantifiers over finite sets. To appear in *Journal of Philosophical Logic*.

Van Benthem, J. (1984). Questions about quantifiers. *Journal of Symbolic Logic*, 49:443–466.

Westerståhl, D. (1986). On the order between quantifiers. In Furberg, M. et al., editors, *Acta Universitatis Gothoburgensis*, pages 273–285. Göteborg University.

Westerståhl, D. (1996). Self-commuting quantifiers. *The Journal of Symbolic Logic*, 61:212–224.

Zimmermann, T. E. (1993). Scopeless quantifiers and operators. *Journal of Philosophical Logic*, 22:545–561.

Computer Science
Technion - Israel Institute of Technology
Haifa 32000
Israel
E-mail: `{alon@vipe,winter@cs}.technion.ac.il`