MM3 Syntax-Semantic Interface
Malte Zimmermann
Universität Potsdam, WS07

Barwise & Cooper (1981). Generalized Quantifiers and Natural Language.

Linguistics & Philosophy 5: 159-219.

1. Introduction

- *Objectives & Claims:*

  i. A semantic characterization of (English) determiners and DPs that *pays close attention to their syntactic structure* by deriving the meaning of DPs in compositional fashion from the meaning of a determiner and its DP-complement and

  ii. formulation of linguistic universals that are based on this semantic characterization.

  Semantically, all DPs are generalized quantifiers of type <<e,t>,t> (as in Montague (1973)): The determiner itself takes a set (<e,t>) and maps it onto a set (or: *family*) of sets (<<e,t>,t>). Determiners thus establish a relation between two sets.

  The range of logically possible relations between sets that can be expressed by natural language determiners is restricted by the semantic property of *conservativity* (or: *live-on property*).

→ The focus on the syntax and on the meaning contribution of DPs, and in particular of the determiner are new features in the discussion of (quantified) noun phrases:

- Russell (1905): *On denoting*

  „This is the principle of the theory of denoting I wish to advocate: that denoting phrases [i.e. complex DPs, MZ] never have any meaning in themselves, but that every proposition in whose verbal expression they occur has a meaning.[...]

  "This leaves 'a man', by itself, wholly destitute of meaning, but gives a meaning to every proposition in whose verbal expression 'a man' occurs."

  "but we agreed that denoting phrases have no meaning in isolation."

- Montague (1973): *PTQ*

  Montague does assign a meaning to denoting expressions, or *terms*: They are functional expressions of type <<e,t>,t>. However, Montague assigns no independent syntactic or semantic status to the determiner itself. Determiners are introduced syncategorematically by the family of rules in S2, and their quantificational interpretation is brought about by the corresponding translation rules in T2.

→ B&C's treatment of generalized quantifiers is an elaboration of Montague's work with a focus on natural language rather than on intensional logic:

p.160: „Our hope is to develop Montague's treatment of noun phrases further in a straightforward way (without lambdas), and to show some of its implications for a theory of natural language."

2. The Basic System: Generalized Quantifiers in Natural Language

2.1 Motivation: Problems with logical approaches to the meaning of DPs

- *The inadequacy of first order PL*

  Standard first order predicate logic with its two quantifiers ∀x(...x...) and ∃x(...x....) cannot express the meaning of certain natural language determiners:

(1) *most*, *more than half*, *finitely many*, ...

(2) Most children are asleep.

(3) a. $\forall x$ [child'(x) $\rightarrow$ sleep'(x)] = All children are asleep
b. $\exists x$ [child'(x) $\wedge$ sleep'(x) = Some child is asleep.
c. $\forall x$ [child'(x) $\wedge$ sleep'(x)] = All enitites are sleeping children.
d. $\exists x$ [child'(x) $\rightarrow$ sleep'(x) = There is an entity such that, if it is a child, it sleeps.

→ no combination of the available logical operators gives the right result

→ the introduction of a new operator MOST does not give the correct result either.

(4) MEIST x [φ(x)] = 1 iff φ is true for more than half of the entities in the domain

(5) a. *Possibility 1:* MOST x [child'(x) $\wedge$ sleep'(x)]
= most entities are sleeping children

b. *Possibility 2:* MOST x [child'(x)$\rightarrow$ sleep'(x)]
= For most entities x: if x is a child, then x is asleep.

(6) a. S1: 10 persons, among them 5 adults that are awake and 5 childen **four of which are asleep**.

→ (2) intuitively true, (4) false (unter the interpretation of *most* in (5a))

b. S2: 10 persons, among them 5 adults that are awake and 5 childen **only one of which is asleep**.

→ (2) intuitively false, but (4) true (unter the interpretation of *most* in (5b))

→ a major problem with (4) is that its structure does not reflect the syntactic structure of (2) because operators must take the entire formula in their scope in PL, but natural language determiners does not.

p.159:"[...], the syntactic structure of quantified sentences in predicate calculus is completely different from the syntactic structure of quantified sentences in natural language."

- *Semantic structure resembles syntactic structure*

An adequate paraphrase of (2) would be something like ***most x such that x is a child*** *have the property of being asleep.*

→ *most* is not a quantifier (ranging over formulas), but a determiner that combines with the set-denoting expression *childen* to produce a quantifier. This semantic procedure is fully parallel to the syntactic structure:

(7) a. [Quantifier [Determiner] [Set expression]]

b. [DP [Det most] [NP children]]

- Unlike in PL, some quantifiers (*most NP*, *many NP*, *few NP*) are not logical quantifiers, i.e. their meaning is model-dependent and varies depending on context:

→ The meaning of *many* in (8ab) depends on the measurement by which we calculate what counts as relatively many under the circumstances described.

(8) Situation: In total 20 Swedes won the Nobel Price, and 20 Swedes live in Berlin.

a. Many Swedes have won the Nobel Price. → true

(compared to the overall number of Nobel price winners, the number of Nobel price winners from other countries, and given the relatively small population of Sweden)

b. Many Swedes live in Berlin. → false

(compared to the overall number of inhabitants of Berlin and the number of inhabitants from other countries)

- Natural language quantifiers can be partially defined

(9) a. the king of France → fails to denote of there is no or more than one king of France

b. both authors → fails to denote if the set of authors comprises less than or more than two authors

c. neither author → fails to denote if the set of authors comprises less than or more than two authors

[NB: German has no simplex counterpart for *neither*, which translates as *keiner der beiden Autoren*]

2.2 Generalized Quantifier Theory

- *The Syntax of Generalized Quantifiers (GQs)*

Generalized Quantifiers are typically complex DPs consisting of a determiner and an NP, cf. (7b). The complex expressions are built by applying the determiner to the NP by rule R3 (p.168):

(10) a. R3: If D is a determiner and η is a set term then $D(\eta)$ is a quantifier Q

Quantifiers (= full DPs) combine with set terms by R5

(10) b. R5: If Q is a quantifier and η is a set term $Q(\eta)$ is a formula (= a sentence)

- *The Semantics of Generalized Quantifiers*

The denotation of a generalized quantifier is derived by functionally applying the denotation of a determiner to the denotation of an NP (S5, S9, T1):

(11) $[\![DP]\!] = [\![D]\!]([\![NP]\!])$

Generalized quantifiers are sets (= families) of sets, i.e. they are of type **<<e,t>,t>**

→ *Quantifiers are second-order predicates that are used to assert that a set has some property:*

(12) a. *some boy* denotes the set of all sets X that contain at least one boy (S5a): $\{X| X \cap [\![boy]\!] \neq \emptyset\}$

b. *every boy* denotes the set of all sets X that contain every boy (possibly plus additional elements) (S5b): $\{X| [\![boy]\!] \subseteq X\}$

Determiners denote functions from sets into generalized quantifiers, i.e. sets of sets, i.e. they are of type **<<e,t>,<<e,t>,t>>**

→ *Determiners denote second-order relations between two sets A and B*

- *No semantic difference between different kinds of nominal expressions*

As in Montague, all nominal expressions, i.e. complex DPs, proper names as well as variables, are generalized quantifiers of type <<e,t>,t>. To capture the intuition that proper names and variables appear to denote individuals, B&C (p.166) postulate the following split between N- and DP-denotation (see also Elbourne 2005 for a recent proposal along these lines):

(13) a.

```
          DP    <<e,t>,t>
           |
           N    <e>
         Harry
```

b. $[[\text{Harry}_{NP}]]$ $= [[\text{the}]](\{y \mid y = \text{harry}\})$
$= [[\text{the}]](\{\text{harry}\})$
$= \{X \mid \{\text{harry}\} \subseteq X\}$
$= \{X \mid \text{harry} \in X\}$

→ the principal ultrafilter generated by [[Harry]]

2.3 Conservativity (the 'live on'-property)

The range of logically possible relations between sets that can be expressed by natural language determiners is restricted by the semantic property of *conservativity* (or: *live-on property*).

(14) *Conservativity:*

for arbitrary sets A,B: $Det(A)(B) \Leftrightarrow Det(A)(A \cap B)$

→ The result of applying the determiner meaning to its two set arguments is equivalent to applying the determiner meaning to the first set argument A (the NP-denotation) and the intersection of first and second argument $A \cap B$

→ as a result, only the NP-denotation A and the intersection of A with B, i.e. $A \cap B$, are relevant for establishing the truth-conditions of a sentence;

Elements of B that are not in A do not matter for the interpretation !

→ conservativity implies that the NP-denotation A is more important than the second set B (typically the VP-denotation): *quantifiers live on A*

- *Empirical test for conservativity*

There is a simple empirical test for conservativity. A determiner Det applied to an NP and a VP is conservative if the following equivalence holds:

(15) *Det NP VP* is true iff *Det NP is a/ are NP(s) that VP* holds

(16) a. Some students smoke. ⇔ Some students are students that smoke.

b. Every student smokes. ⇔ Every student is a student that smokes.

c. No student smokes. $\Leftrightarrow$ No student is a student that smokes.

d. Two students smoke. $\Leftrightarrow$ Two students are students that smoke.

- *Formal Proof for Conservativity:* some

(17) some (A)(B) = 1 iff $A \cap B \neq \emptyset$ (meaning of *some*)
$\Leftrightarrow$ $A \cap A \cap B \neq \emptyset$ (set theory: $A = A \cap A$)
$\Leftrightarrow$ $A \cap (A \cap B) \neq \emptyset$
= 1 iff some(A)($A \cap B$) (meaning of *some*)

→ The criterion of conservativity makes a clear prediction as to which of the logically possible quantifiers can occur as quantifiers in natural language.

By doing so, it restricts the number of logically possible determiner denotations from **65536** to **512** in a model with only two individuals.

- *Prediction*

There are no equivalences of the form Det(A)(B) $\Leftrightarrow$ Det ($A \cap B$)(B), where the meaning of the NP-complement A in its entirety does not play a role for the semantic interpretation:

(18) Every beer drinker is a student. $\neq$ Every beer drinking student is a student.

Example: The logically possible quantifier *schmevery* in (19a) with the meaning in (19b) is not attested in any natural language, even though the meaning is plausible and not difficult to compute, cf. (20):

(19) a. **Schmevery** student drinks beer = 1 iff
b. every beer drinker is a student: [[beer drinker]] $\subseteq$ [[student]])

(20) a. [[schmevery]] = $\lambda A \in \wp(D).\ \lambda B \in \wp(D).\ B \subseteq A$

b. [[schmevery student]] = $\lambda B \in \wp(D).\ B \subseteq$ [[student]]

c. [[schmevery student drinks beer]] = 1 iff [[beer drinker]] $\subseteq$ [[student]]

- Formal proof that *schmevey* is not conservative:

(21) i. the inference from left to right in (14) is valid:

schmevery(A)(B) = 1 iff $B \subseteq A$ (meaning of *schmevery*)
$\Rightarrow$ $A \cap B \subseteq A$ (set theory)
iff schmevery(A)($A \cap B$) =1 (meaning of *schmevery*)

ii. the inference from right to left in (15) is invalid:
schmevery (A)($A \cap B$) = 1 iff $A \cap B \subseteq A$ (meaning of *schmevery*)
// $\Rightarrow$ $\mathbf{B \subseteq A}$
iff schmevery(A)(B) = 1

From $A \cap B \subseteq A$ it does not follow automatically that $B \subseteq A$!

Question: What about the semantics of *only* in *Only Students are beer drinkers*?

2.4 The syntax and semantics of sentences containing generalized quantifiers

General interpretation scheme of sentences containing quantifiers:

(22) ⟦S⟧ = D(A)(B)

Generalized quantifiers in subject position take the VP-denotation as argument and deliver a truth value:

(23) a. $[_{DP}$ Every $[_{NP}$ man]] $[_{VP}$ sneezes]

→ **every (man) sneeze** (by R3 and R5)

b. $[_{DP}$ some $[_{NP}$ thing]] $[_{VP}$ runs]

→ **some (thing) run** (by R3 and R5)

→ Quantifiers in non-subject position are always interpreted by means of quantifying in (T2, SD6, T6). The quantifying-in rule SD6/T6 is equivalent to Montague's rule $F10_n$ (S14/T14)

→ Unlike in Montague (1973), there is no option to raise the type of the verb such that it can combine with a generalized quantifier in situ.

(23) c. Five or more woman kiss the man.

→ **5(woman) [λx. the(man) [λy. kiss(x,y)]]**

d. Most man kiss a (particular) woman.

→ **some(woman) [λy. most(men) [λx. kiss(x,y)]]**

→ (23d) is the semantic representation of the inverse scope reading.

3. Universals and different semantic properties of generalized quantifiers

Section 4 of Barwise & Cooper (1981) discusses possible linguistic applications of the formal apparatus developed in sections 2 and 3. B&C postulate a number of universals for the semantic behavior of quantifiers in natural language that are not logically necessary.

→ If correct, the existence of the universals points to an important difference between natural language and formal (i.e. logical) languages. The former are restricted in that they only make use of a limited subset of the possibilities that are available from a logical point of view.

→ If correct, the validity of the universals thus qualifies as a distinguishing feature between natural language, on the one hand, and formal languages, on the other.

U1: Every natural language has DPs that denote Generalized Quantifiers

U2: Since semantic scope is the property of the entire DP rather than the determiner itself, at least DP's will occur in positions associated with variable binding

→ There should be no language that allows for determiner movement, but not DP-movement

U3: Every natural language has conservative determiners

U4: For every partially defined determiner D, there is a morphologically simple determiner D^+ that is always defined

→ There should be no language with a word for *both* but not word for *two*, or with a word for *neither* but no word for *no*.

• Strong and weak quantifiers

Milsark (1974, 1977) points out that not all determiners behave alike in all respects. For instance, only a subset of all determiners (the *weak determiners*) can freely occur in existential *there*-sentences, while others (*the strong determiners*) are infelicitous.

(24) a. *?There is every /the cat in the garden.
b. *?There are most cats(the two cats in the garden.
c. There is a / some unicorn in the garden.
d. There are two/ more than five / less than six dogs in the garden.

B&C provide a formal definition for the weakness or strength of determiners

(25) *strength/weakness of D:*

A determiner is positive strong (or negative strong) if for every model M with a domain E and every $A \subseteq E$, if the quantifier $\|D\|(A)$ is defined then $A \in \|D\|(A)$ (or $A \notin \|D\|(A)$). If D is not strong it is weak.

→ Empirical test:

(26) D N is a N/ are Ns
a. tautology → D is positive strong
b. contradiction → D is negative strong
c. contingency: → D is weak

(27) a. Every gnu is a gnu → tautology: *every* is positive strong
b. Neither gnu is a gnu → contradiction: *neither* is negative strong
c. Two gnus are gnus → true if there are gnus; false if there are no gnus: *two* is weak

Question: Why is *no* a weak determiner?

→ all definite determiners are partially defined and positive strong.

→ weak determiners are symmetrical and intersective:

(28) a. Two students smoke. = Two smokers are students.

→ strong determiners are asymmetrical and non-intersective:

(28) b. Most students smoke. ≠ Most smokers are students.

(29) a. D is symmetrical and intersective iff for all $A,B \in D_{<e,t>}$: $D(A)(B) = D(B)(A)$
b. D ist aymmetrical and non-intersective iff for all $A,B \in D_{<e,t>}$: if $D(A)(B) = D(B)(A)$ then $A = B$.

→ the arguments of symmetrical Ds can be exchanged without a change in meaning (28a), while the arguments of asymmetrical Ds can only be exchanged salva veritate if A=B (28b).

- Monotone increasing and decreasing quantifiers

monotone increasing quantifiers license the inference from subsets to supersets, (30a); monotone decreasing quantifiers license the inference from supersets to subsets, (30b):

(30) a. Every letter arrived yesterday morning → Every letter arrived yesterday.

b. Less than two letters arrived yesterday → Less than two letters arrived yesterday morning.

→ *negation* reverses the monotonicity properties of quantifiers

(31) a. Not every letter arrived yesterday morning // → Not every letter arrived yesterday.

b. Not every letter arrived yesterday → Not every letter arrived yesterday morning.

U5: There is a simple monotonous decreasing quantifier only if there is also a simple monotonous increasing quantifier with a weak non-cardinal quantifier

→ There should be no languages with a word for *no*, but no word for *some*, or a word for *few* but no word for *many*.

U6: The simple NPs of any natural language express monotone quantifiers or conjunctions of monotone quantifiers

→ There should be no languages in which the concepts *exactly three or exactly five* or *all but one* are lexicalized in a basic determiner meaning

U7: In natural languages, positive strong determiners are monotone increasing and negative strong determiners are monotone decreasing.

- *Persistence:*

A determiner is persistent if it is monotone increasing concerning the NP-argument:

(32) At least two beautiful cows give a lot of milk. → At least two cows give a lot of milk.

U8: If a determiner is monotone increasing concerning its first argument, it is also monotone increasing concerning its second argument.

Question: Which determiner is monotone increasing in its second argument, but decreasing in its first argument?