1 The standard quantifiers in FOL

- First order logic includes the two quantifiers $\forall x$ and $\exists x$, for any variable $x$.

  - If $M$ is a model with domain $U$, interpretation function $\llbracket \cdot \rrbracket$, and $g$ is an assignment function into $U$:

    (1) $\llbracket \forall x \Phi \rrbracket^M,g = 1$ iff: for all $d \in U$, $\llbracket \Phi \rrbracket^M,g[x/d] = 1$

    (2) $\llbracket \exists x \Phi \rrbracket^M,g = 1$ iff: there is a $d \in D$ such that $\llbracket \Phi \rrbracket^M,g[x/d] = 1$

- An assignment is a function that maps variables onto members of the domain.

  For any assignment $g$, $g[x/d]$ is an assignment which differs from $g$ at most in assigning variable $x$ to domain member $d$.

2 Using the first-order quantifiers for NL

- With just FOL (including identity), you can do a decent amount of semantic analysis.

  (3) a. Every professor smokes.
  b. $\forall x(P(x) \rightarrow S(x))$

  (4) a. Some professor smokes.
  b. $\exists x(P(x) \land S(x))$

  (5) a. No professor smokes.
  b. $\neg \exists x(P(x) \land S(x)) \land x \neg (P(x) \rightarrow S(x))$

  (6) a. Exactly two professors smoke.
  b. $\exists x \exists y (P(x) \land P(y) \land S(x) \land S(y) \land x \neq y \land \forall z ((P(z) \land S(z)) \rightarrow (z = x \lor z = y)))$
The addition of $\lambda$ abstraction then allows compositional derivations, wherein the quantificational elements are contributed lexically by determiners.

- $\lambda$ Abstraction

Syntax:
If $v$ is a variable of type $\alpha$, $\phi$ is an expression of type $\beta$, then $\lambda v \phi$ is an expression of type $\langle \alpha, \beta \rangle$.

Semantics:
$$\llbracket \lambda v(\alpha) \phi(\beta) \rrbracket = f : \forall d(\alpha) ( f(d) = [\phi]_{M,g[v/d]} )$$

I.e. $\lambda v \phi$ is the function that, given any $d$, outputs the result of replacing any $v$ in $\phi$ with $d$.

Given the semantic representation (3b) for (3a), for example, we isolate the semantic contribution of every by abstracting over what it clearly does not contribute:

$$\text{translation}(\text{every}) = \lambda S \lambda R. \forall x(R(x) \rightarrow S(x))$$

The general result is a function of type $\langle \langle e, t \rangle, \langle e, t \rangle, t \rangle \rangle$. More on this below.

### 2.1 Montague’s rule S8,n (Quantifying-in)

- The grammar in Montague 1973 (PTQ) includes the following syntactic rule:

  Rule S8,n: (Approximate!)
  An expression in category NP can be combined with an expression in category S, to yield a larger S where a pronoun with index $n$ is replaced with NP.

  This allows (e.g.) the NP “every flavor” to combine with the S “he$_7$ likes it$_3$” to yield the S “He$_7$ likes every flavor.”

  
  $[\text{ Every flavor }] [\text{ He}_7 \text{ likes it}_3] \xrightarrow{S8,3} [\text{ He}_7 \text{ likes every flavor }]

  - The semantic correlate of S8,n is the translation rule T8,n:

  Rule T8,n: (Extensionalized)
  If $\beta = S8,n(\alpha, \phi)$; and if $\alpha$ translates to $\alpha'$ and $\phi$ to $\phi'$; then:

  $$\text{translation}(\beta) = T8,n(\alpha', \phi') = \alpha' (\lambda x_n \phi')$$
So, given these translations for “every flavor” and “he\textsubscript{7} likes it\textsubscript{3}” (simplified from what Montague had):

\[ \alpha' = \text{translation(every flavor)} = \lambda S. \forall z (\text{flavor}(z) \rightarrow S(z)) \]
\[ \phi' = \text{translation(he\textsubscript{7} likes it\textsubscript{3})} = \text{likes}(x_7, x_3) \]

We have the following translation, corresponding to the application of S8,n:

\[ T_8, 3(\alpha', \phi') \]
\[ = \alpha'(\lambda x_3. \phi') \]
\[ = \lambda S. \forall z (\text{flavor}(z) \rightarrow S(z)) (\lambda x_3. \text{likes}(x_7, x_3)) \]
\[ = \forall z (\text{flavor}(z) \rightarrow \text{likes}(x_7, z)) \]

• Such rules allowed Montague to give quantificational NPs wide scope with respect to an S, and any other scope-bearing elements within it.

• N.B. For Montague, all NPs denote quantifiers, never simple individuals. Nevertheless, all NPs in object position can be interpreted in situ, since he has transitive verbs denoting functions over NP meanings (i.e. quantifiers).

2.2 Heim and Kratzer’s QR

• For Heim and Kratzer (1998) and others, transitive verbs denote functions over individuals. Thus a quantifier in object position cannot be interpreted in situ.

For this reason, and in order to generate various scopings, they deploy a rule of Quantifier Raising (QR), which is essentially the inverse of Montague’s S8,n.

• Syntax of QR:

\[ [\gamma \ldots \text{DP} \ldots] \xrightarrow{QR} [\gamma \text{ DP} [i \ [\gamma \ldots t_i \ldots]]] \]

where \( t \) is a trace in the position occupied by DP in the input, and \( i \) is an index.
• Semantics for structures generated by QR (among others).

  – Predicate Abstraction (PA)
    If \( \alpha \) has daughters \( i \) and \( \beta \), where and \( i \) is an index, then:
    \[
    \llbracket \alpha \rrbracket^g = \lambda x. \llbracket \beta \rrbracket^{g[x/i]}
    \]
    (N.B. Heim and Kratzer use the notation \( g[x/y] \) differently than it is used by (e.g.) Gamut 1991. For H&K, \( g[x/y] \) is an assignment that differs from \( g \) at most in that \( g(y) = x \). For Gamut, it means: \ldots at most in that \( g(x) = y \). This difference explains my unfortunate tendency to mix up the two conventions.)

  – Traces and Pronouns Rule (TPR)
    If \( \alpha \) is a pronoun or a trace, \( g \) is a variable assignment, and \( i \in \text{dom}(g) \), then:
    \[
    \llbracket \alpha_i \rrbracket^g = g(i)
    \]

Besides these crucial rules, we of course also have Functional Application:

  – Functional Application (FA)
    If \( \alpha \) has daughters \( \beta \) and \( \gamma \), then for any assignment \( g \), if \( \llbracket \beta \rrbracket^g \) is a function whose domain contains \( \llbracket \gamma \rrbracket^g \):
    \[
    \llbracket \alpha \rrbracket^g = \llbracket \beta \rrbracket^g(\llbracket \gamma \rrbracket)^g
    \]

EXERCISE:
Given the lexical translations in (7), calculate the translation of the LF below.

(7) a. every: \( \lambda R \lambda S. \forall x(R(x) \rightarrow S(x)) \)
    b. flavor: \( \lambda x. \text{flavor}(x) \)
    c. likes: \( \lambda y \lambda x. \text{likes}(x, y) \)

\[
\begin{array}{c}
S \\
\quad \text{DP} \\
\quad \quad \text{3} \\
\quad \quad \quad \text{S} \\
\quad \quad \quad \quad \text{DP} \\
\quad \quad \quad \quad \quad \text{VP} \\
\quad \quad \quad \quad \quad \quad \text{he} \\
\quad \quad \quad \quad \quad \quad \quad \text{V} \\
\quad \quad \quad \quad \quad \quad \quad \quad \text{likes} \\
\quad \quad \quad \quad \quad \quad \quad \quad \quad \text{t_3} \\
\end{array}
\]

\( g = 1 \) iff: \ldots

1. \( \llbracket \text{every flavor} \rrbracket^g(\llbracket \text{3 he_7 likes t_3} \rrbracket^g) \) FA
2. \( \llbracket \text{every flavor} \rrbracket^g (\lambda x. (\llbracket \text{he_7 likes t_3} \rrbracket^g(x)) ) \) PA
3. \( \llbracket \text{every flavor} \rrbracket^g (\lambda x. (\llbracket \text{likes} \rrbracket^g(x)) (\llbracket \text{he_7} \rrbracket^g(x)) ) \) FA
4. \( \llbracket \text{every flavor} \rrbracket^g (\lambda x. (\llbracket \text{like} \rrbracket^g(x, y)) (\llbracket \text{he_7} \rrbracket^g(x)) ) \) Lex, TPR
5. \( \llbracket \text{every flavor} \rrbracket^g (\lambda x. (\llbracket \text{like} \rrbracket^g(x, y)) (\llbracket \text{he_7} \rrbracket^g(x)) ) \) \( g[x/3](3) = x; g[x/3](7) = g(7) \)
6. \( \llbracket \text{every flavor} \rrbracket^g (\lambda x. \text{like}(g(7), x)) \) \( \lambda \) Conversion
7. \[ \text{every}^g(\text{flavor}^g) (\lambda x. \text{like}(g(7), x)) \] \hspace{1cm} \text{FA}

8. \[ \lambda R \lambda S. \forall z(R(z) \rightarrow S(z))(\lambda k. \text{flavor}(k))(\lambda x. \text{like}(g(7), x)) \] \hspace{1cm} \text{Lex}

9. \[ \lambda S. \forall z(\text{flavor}(z) \rightarrow S(z))(\lambda x. \text{like}(g(7), x)) \] \hspace{1cm} \lambda \text{ Conversion}

10. \[ \forall z(\text{flavor}(z) \rightarrow \lambda x. \text{like}(g(7), x)(z)) \] \hspace{1cm} \lambda \text{ Conversion}

11. \[ \forall z(\text{flavor}(z) \rightarrow \text{like}(g(7), z)) \] \hspace{1cm} \lambda \text{ Conversion}

• Quantificational DPs are in \(\langle e, t \rangle\) and \(\lambda x \Phi\) is in type \(\langle e, \text{TYPE}(\Phi) \rangle\).

Thus a structure \([\text{DP} [ i \text{ XP } ]]\) formed by QR will be interpretable if \([\text{XP}]\) is in type \(\langle t \rangle\). But otherwise it may not be.

For this reason, those who adopt QR with an H&K semantics assume that QR always raises a DP to a node in \(\langle t \rangle\).

(You may recall various puzzles associated with this consequence, like how to derive the LF for the most natural reading of “No student from a foreign country was admitted”: presuming that “a student” must QR, where does it QR to, such that it still has no scope with respect to “no”?)

3 Two problems for the first-order formalization

3.1 The mismatch between logical and grammatical form

• In the usual translation, the constituency of the logical forms does not match the constituency of the translated sentence:

\[
\text{every professor} \quad \text{smokes}
\]

\[
\forall x
\]

Here the logical form includes no constituent corresponding to “every professor,” for instance.

• Perhaps it would be preferable to have a logical form whose parts matched those of the grammatical form, given our usual methodological assumptions.

Or at least, it would be good to know that there could be such a form.
3.2 The much more serious problem: non-firstorder quantifiers

- Certain quantifiers provably have no first-order representation; for example, all quantifiers whose determiner is most.

There is no way to represent, for arbitrary models, the meanings of the sentences in (8) using FOL.

(8) a. Most professors smoke.
    b. Most students drink.
    c. Most politicians fornicate.

- Here is a way to appreciated this, short of a full proof.

  - If a sentence with a quantifier can be expressed with a first-order formula, the formula will have a form like this:
    \[ W_x \Phi \]
    Where \( W_x \) is some quantifier, just like \( \forall x \) or \( \exists x \).

  - Such formulas say something about how many members of the domain \( U \) have a certain property: the property of being a \( d \) such that \( \llbracket \Phi \rrbracket_{g[x/d]} \).

1. \( \forall x \Phi \) says that for all \( d \), \( \llbracket \Phi \rrbracket_{g[x/d]} \) is true.
   (9) Every professor smokes.
   (10) For every \( d \):
   (11) If it is a professor, it smokes.
   (12) = It is either a non-professor or a smoker.
   (13) = It is not both a professor and a non-smoker.

2. \( \exists x \Phi \) says that for at least one \( d \), \( \llbracket \Phi \rrbracket_{g[d/x]} \) is true.
   (14) Some student drinks.
   (15) At least one \( d \) is such that:
       a. It is a student and it drinks.

- Notice that the condition \( \Phi \) is constructed from the N and VP expressions, using the usual connectives—but still, it imposes a single condition on some arbitrary portion of the domain.
– Now, there is in fact no way to describe a sentence with “most” this way.

(16) *(Some proportion of) d’s in the domain are such that \([\Phi]_\Phi^d/x\)

(17) Most professors smoke.
   a. Over half the elements in the domain are such that:
      i. *If they are professors, they smoke.\(^1\)
      ii. *They are professors and they smoke.
      iii. *It is not true that they are both non-professors and non-smokers.
      iv. *Either they are professors or they are smokers.
      v. *...

– Perhaps this is clear intuitively, because “Most NP VP” says nothing about any proportion of the domain in general. It says something only about those in the set \([NP]\).

(18) a. Most professors smoke.
   b. = Over half the elements in the set of professors are such that they smoke.

4 The basic idea Generalized Quantifier Theory

• GQT begins with these two ideas:
  1. DPs denote second-order properties: i.e. a sets of sets
  2. Determiners denote relations between sets.

• Happily, this perspective accommodates both those quantifiers that can be defined in FOL, and those that cannot be:
  1. “every” denotes the relation of set inclusion.
     \[[\text{every professor smokes}] = \text{EVERY}(P)(S) = P \subseteq S\]
     So \([\text{every NP}]\) is the set of all \([NP]\)’s superset.

\(^1\)Notice that this would be true in any model where most things are not professors, since a conditional is true whenever it’s antecedent is false.
2. “some” denotes the relation of having a non-empty intersection
\[[ \text{some professor smokes} ] \equiv \text{SOME}(P)(S) = (P \cap S) \neq \emptyset\]
So \[[\text{some NP}]\] is the set of all sets that overlap with \[[\text{NP}]\].

3. “most” denotes the relation that holds between sets \(A\) and \(B\) when the members of \(A \cap B\) outnumber the members of \(A - B\).
\[[ \text{most professors smoke} ] \equiv \text{MOST}(P)(S) = |(P \cap S)| > |(P - S)|\]
So \[[\text{most NPs}]\] is the set of all sets whose intersection with \[[\text{NP}]\] is larger than the intersection its complement with \[[\text{NP}]\].

- The success of this generalized perspective leads us to ask:
  1. What formal properties do the determiner relations have?
  2. Do any of these formal properties correlate with grammatical facts?
  3. Are any of these formal properties shared by all determiner meanings?

There have been important results with regard to all these questions.

5 Conservativity

- The best known result of GQT is the conjecture that all NL determiner meanings are conservative.

Conservativity \(D\) is conservative iff \(D(A)(B) \iff D(A)(A \cap B)\).

- Test: \(\text{DET N VP} \iff \text{DET N are N that VP}\)
  - \([[\text{some}]\] is conservative:
    19. a. Some professors smoke a pipe.
    b. \(\Rightarrow\) Some professors are professors that smoke a pipe.
  20. a. Some professors are professors that smoke a pipe.
    b. \(\Rightarrow\) Some professors smoke a pipe.
[every is conservative:

(21) a. Every professor smokes a pipe.
    b. ⇒ Every professor is a professor that smokes a pipe.

(22) a. Every professor is a professor that smokes a pipe.
    b. ⇒ Every professor smokes a pipe.

[most is conservative:

(23) a. Most professors smoke a pipe.
    b. ⇒ Most professors are professors that smoke a pipe.

(24) a. Most professors are professors that smoke a pipe.
    b. ⇒ Most professors smoke a pipe.

Notice that in the case of only, the inferences do not go through

(25) a. Only professors smoke a pipe.
    b. ⇒ Only professors are professors that smoke a pipe.

(26) a. Only professors are professors that smoke a pipe.
    b. \( \not\Rightarrow \) Only professors smoke a pipe.

As it happens, however, only is not at determiner, and so is not an exception to the claim that determiner meanings are conservative:

(27) a. Only the professors smoke a pipe.
    b. The professor only smokes a pipe.
    c. The professor will smoke a water-pipe only.

6 Extension

Extension \( \mathcal{D} \) satisfies extension if for \( A, B \subseteq U \subseteq U' \): \( \mathcal{D}_U(A)(B) \iff \mathcal{D}_{U'}(A)(B) \).

That is, if some things are added to the domain that are neither As nor Bs, the value of \( \mathcal{D}(A)(B) \) is unchanged.

• Example:

\( U = \{\text{Al Gore, Bill Clinton, Condeleeza Rice}\} \)

\( U' = \{\text{Al Gore, Bill Clinton, Condeleeza Rice, Djibouti, East Timor, Ferghana Valley, ...}\} \)

\([\text{man/men}]^U = [\text{man/men}]^{U'} = \{\text{Al Gore, Bill Clinton}\}\)

\([\text{politician/politicians}]^U = [\text{politician/politicians}]^{U'} = U\)
(28) Truth preserved under domain extension
   a. Some men are politicians.
   b. Every man is a politician.
   c. ?? Many/few men are politicians.

7 Isomorphy

Isomorphy If $f$ is a bijection from $U$ to $U'$, then $D_U(A)(B) \iff D_{U'}(fA)(fB)$.

• Relations that satisfy isomorphy are insensitive to the identity of the elements in the domain. All that could matter is what sets $d$ is in.

• If you think of possessives as determiners, semantically, then these are nonisomorphic.

8 Conservativity + Extension + Isomorphy

• If $D$ satisfies Conservativity and Extension, then:
  1. $D_U(A)(B) \iff D_A(A)(B)$;
  2. and $D(A)(B)$ depends only two sets: $A \cap B$ and $A - B$.

• If $D$ satisfies Conservativity, Extension and Isomorphy, then $D(A)(B)$ depends only the cardinalities of two sets: $A \cap B$ and $A - B$.

• The determiners whose meanings satisfy these three conditions are sometimes called the *logical* determiners; or sometimes, the *quantifiers* proper.
9 Symmetry and Intersectivity

Symmetry \( \mathcal{D} \) is *symmetrical* if \( \mathcal{D}(A)(B) = \mathcal{D}(B)(A) \), and *nonsymmetrical* otherwise.

- **Test:** DET Pred1 Pred2 \( \iff \) DET Pred2 Pred1
  - *Some* denotes a symmetrical relation:
    
    (29) a. Some professors smoke a pipe.
    b. \( \iff \) Some pipe-smokers are professors.
  - *Every* denotes an asymmetrical relation:
    
    (30) a. Every professor smokes a pipe.
    b. Every pipe-smoker is a professor.

Intersective \( \mathcal{D} \) is *intersective* iff \( \mathcal{D}(A)(B) \iff \mathcal{D}(A \cap B)(B) \).

- **Fact:** Conservative \( \mathcal{D} \) is symmetric iff it is intersective
  - **Proof:**
    \[
    \mathcal{D}(A)(B) \\
    \iff \mathcal{D}(A \cap B)(B) \text{ (intersectivity)} \\
    \iff \mathcal{D}(A \cap B)((A \cap B) \cap B) \text{ (conservativity)} \\
    \iff \mathcal{D}(A \cap B)(A \cap B) \text{ (equivalent to last line)} \\
    \iff \mathcal{D}(A \cap B)((A \cap B) \cap U) \text{ (since } X \cap U = X) \\
    \iff \mathcal{D}(A \cap B)(U) \text{ (conservativity)}
    \]
    By the same token: \( \mathcal{D}(B)(A) \iff D(B \cap A)(U) \iff D(A \cap B)(U) \)
    Thus if \( \mathcal{D} \) is intersective: \( \mathcal{D}(A)(B) \iff \mathcal{D}(A \cap B)(U) \iff D(B)(A) \)
    Which shows that \( \mathcal{D} \) is intersective iff symmetric.

- **Note that** only expresses an intersective relation
  
  (31) a. Only professors smoke a pipe.
  b. \( \Rightarrow \) Only professors that smoke a pipe smoke a pipe.
  (32) a. Only professors that smoke a pipe smoke a pipe.
  b. \( \Rightarrow \) Only professors smoke a pipe.

- Intersectivity is parallel to conservativity, but over the “left” argument. Therefore it might be true that all NL relations in \( A \times A \) are *either* (right-)conservative or left-conservative.
10  Strength

- Milsark (1977) discusses the fact that the acceptability of an NP in an existential sentence depends on its determiner:

\[
(33) \quad \begin{align*}
\text{a. & There are no professors.} \\
\text{b. & There are some professors.} \\
\text{c. & # There is every professor.} \\
\text{d. & # There is not every professor.}
\end{align*}
\]

Determiners that are unacceptable in existential sentences, he called \textit{strong}. Those that are acceptable, he called \textit{weak}.

- Barwise and Cooper (1981) suggest this semantic definition of Milsark’s categories:

  - \( \mathcal{D} \) is positive strong iff reflexive: \textbf{for every model} \( M \) and every \( A \subseteq U \), if \( \mathcal{D}(A) \) is defined, then \( \mathcal{D}(A)(A) \).
  
  - \( \mathcal{D} \) is negative strong iff irreflexive: \textbf{for every model} \( M \) and every \( A \subseteq U \), if \( \mathcal{D}(A) \) is defined, then \( \neg \mathcal{D}(A)(A) \).
  
  - \( \mathcal{D} \) is weak iff it is neither positive nor negative strong.

\[
(34) \quad \text{“every” is positive strong, since this is true, regardless of whether the domain includes professors:} \\
\text{a. Every professor is a professor.}
\]

\[
(35) \quad \text{“no” is weak, because neither of the following statements is necessarily true. The first is false in a domain that includes professors, and the second is false in a domain without professors:} \\
\text{a. No professor is a professor.} \\
\text{b. It’s not true that no professor is a professor.}
\]

\[
(36) \quad \text{“some” is weak, because neither of the following statements is necessarily true. The first is false in a domain without professors, and the second is false in a domain with professors:} \\
\text{a. Some professor is a professor.} \\
\text{b. It’s not true that some professor is a professor.}
\]
Here is one story about why the strong determiners should be bad in existential contexts:

– Assume that “There is/are \([DP \text{ Det NP }]\)” is true iff \([\llbracket \text{Det} \rrbracket ((\llbracket NP \rrbracket))(U)\).

– Then by conservativity, “There is/are \([DP \text{ Det NP }]\)” is true iff \([\llbracket \text{Det} \rrbracket ((\llbracket NP \rrbracket))(\llbracket NP \rrbracket)\).

– Of course \([\llbracket \text{Det} \rrbracket ((\llbracket NP \rrbracket))(\llbracket NP \rrbracket)\) is always true when \([\llbracket \text{Det} \rrbracket\) is reflexive, and always false when it is irreflexive.

– Hence “There is/are \([DP \text{ Det NP }]\)” will be informative only when the Det (meaning) is weak.

Barwise and Cooper’s definition of strength fails with certain proportional determiners, at least if you agree with the following somewhat delicate judgment:

\[(37) \quad \# \text{ There are 5\% of the professors in the room.}\]

Such “n\%” determiners do count as weak for Barwise and Cooper.

\[(38) \quad \begin{array}{l}
\text{a. Not true without professors:} \\
\quad 5\% \text{ of professors are professors.} \\
\text{b. Not true unless there are enough professors:} \\
\quad \text{It is not true that 5\% of professors are professors.}
\end{array}\]

Keenan (1987) suggests instead that the weak determiners are those that denote intersective relations.\(^3\)

As desired, the proportional determiners are not intersective, since \(\mathcal{D}(A \cap B)(B) \not\rightarrow \mathcal{D}(A)(B)\):

\[(39) \quad \begin{array}{l}
\text{a. 5\% of professors smoke a pipe.} \\
\text{b. } \not\rightarrow 5\% \text{ of professors that smoke a pipe smoke a pipe.}
\end{array}\]

\[(40) \quad \begin{array}{l}
\text{a. 5\% of professors that smoke a pipe smoke a pipe.} \\
\text{b. } \not\rightarrow 5\% \text{ of professors smoke a pipe.}
\end{array}\]

Dyadic intersective relations are symmetrical. Hence in the usual case, the weak determiners are the ones that denote symmetric relations.

\(^3\)Keenan extends the definition of intersectivity to \(n\)-place relations. \(\mathcal{D} \) is intersective iff \(\mathcal{D}(A_i, \ldots, A_n, B) \iff \mathcal{D}(A_i \cap B, \ldots, A_n \cap B, B)\).
11 Monotonicity

11.1 (Right) Monotone Increasing

Monotone Increasing (for quantifiers) \( Q \) in model \( M \) is monotone increasing iff for all sets \( B, C \): if \( Q(B) \) and \( B \subseteq C \), then \( Q(C) \)

Right Monotone Increasing (for determiner meanings) \( D \) in model \( M \) is monotone increasing iff for all sets \( A, B, C \): if \( D(A)(B) \) and \( B \subseteq C \), then \( D(A)(C) \)

- Right monotone increasing determiner meaning = right upward monotone determiner meaning \( \Rightarrow \) monotone increasing quantifier

- Test: If \([VP1] \subseteq [VP2]\), then \([NP \ VP1] \Rightarrow [NP \ VP2]\).

(41) a. Some professors smoke a pipe.
    b. \( \Rightarrow \) Some professors smoke.

(42) a. Every professor smokes a pipe.
    b. \( \Rightarrow \) Every professor smokes.

11.2 (Right) Monotone Decreasing

Monotone decreasing (for quantifiers) \( Q \) in model \( M \) is monotone decreasing iff for all sets \( B, C \): if \( Q(B) \) and \( C \subseteq B \), then \( Q(C) \)

Right monotone decreasing (for determiner meanings) \( D \) in model \( M \) is right monotone decreasing iff for all sets \( A, B, C \): if \( D(A)(B) \) and \( C \subseteq B \), then \( D(A)(C) \)

- Right monotone decreasing determiner meaning = Right downward monotone determiner meaning \( \Rightarrow \) Monotone decreasing quantifier.

(43) a. No professor smokes.
    b. \( \Rightarrow \) No professor smokes a pipe.
11.3 Quantifiers that are not monotone

- Some quantifiers are neither monotone increasing, nor monotone decreasing.

\[(44)\]

a. Exactly three professors smoke a pipe.

b. \(\lnot\) Exactly three professors smoke.

c. \(\lnot\) Exactly three professors smoke a water-pipe.

- It has been proposed, as a possible NL universal, that:

Names and DPs with simple, noncompound determiners express monotone quantifiers, or conjunctions of monotone quantifiers.

Simple determiners: every, some, no

Compound determiners: exactly three, between three and six, \ldots

11.4 Monotonicity under conjunction, disjunction, and negation

- Right monotonicity of quantifiers is preserved under conjunction (intersection) and disjunction (union).

1. Suppose that \((Q_1 \cap Q_2)(B)\). Then necessarily \(Q_1(B)\) and \(Q_2(B)\). Now suppose that \(B \subseteq C\). If \(Q_1\) and \(Q_2\) are monotone increasing, then \(Q_1(C)\) and \(Q_2(C)\). Consequently \((Q_1 \cap Q_2)(C)\), which shows that \((Q_1 \cap Q_2)\) is also monotone increasing.

2. Suppose that \((Q_1 \cap Q_2)(B)\). Then necessarily \(Q_1(B)\) and \(Q_2(B)\). Now suppose that \(C \subseteq B\). If \(Q_1\) and \(Q_2\) are monotone decreasing, then \(Q_1(C)\) and \(Q_2(C)\). Consequently \((Q_1 \cap Q_2)(C)\), which shows that \((Q_1 \cap Q_2)\) is also monotone decreasing.

3. Exercise: Prove that upward monotonicity is preserved under Union.

4. Exercise: Prove that downward monotonicity is preserved under Union.

- Examples

\[(45)\]

a. At least two students smoke (a pipe).

b. Every professor smokes (a pipe).

c. At least two students and every professor smoke (a pipe).
Negation reverses monotonicity.

- Definitions
  1. $\neg Q \equiv \{ X \subseteq U \mid X \notin Q \}$
  2. $Q \neg \equiv \{ X \subseteq U \mid (U - X) \in Q \}$

- Reversal of monotonicity
  If Q is monotone increasing (decreasing), then both $\neg Q$ and $Q \neg$ are monotone decreasing (increasing).

- Example Proof:
  If Q is monotone increasing, then $\neg Q$ is monotone decreasing.
  Assume that Q is monotone increasing. Now take two arbitrary sets X, Y, with these properties: $Y \subseteq X$; and $X \in \neg Q$, hence $X \notin Q$. Given this, it cannot be that $Y \in Q$. For if it were, the upward monotonicity of Q would imply that $X \in Q$, contradicting our premise that $X \notin Q$. So it must be that $Y \notin Q$, hence that $Y \in \neg Q$. But then Q is monotone decreasing, since $Y \in \neg Q$ whenever $X \in \neg Q$, for any $Y \subseteq X$.

11.5 Left Monotone Increasing (Persistent)

**Left Monotone Increasing**  $\mathcal{D}$ is left monotone increasing (or **persistent**) iff for all sets $A$, $B$, $C$: if $\mathcal{D}(A)(B)$ and $A \subseteq C$, then $\mathcal{D}(C)(B)$.

(46) **Some** denotes a left monotone increasing relation:
  a. Some professor smokes.
  b. $\Rightarrow$ Some person smokes.

(47) **Every** does **not** denote a left monotone increasing relation:
  a. Every professor smokes.
  b. $\not\Rightarrow$ Every person smokes.

- Left monotone increasing = persistent = left upward monotone
11.6 Left Monotone Decreasing (Antipersistent)

**Left Monotone Decreasing** $\mathcal{D}$ is left monotone decreasing (or antipersistent) iff for all sets $A$, $B$, $C$: if $\mathcal{D}(A)(B)$ and $C \subseteq A$, then $\mathcal{D}(C)(B)$.

(48) **Every** denotes a left monotone decreasing relation:
   a. Every professor smokes.
   b. $\not\exists$ Every linguistics professor smokes.

(49) **Some** does **not** denote a left monotone decreasing relation:
   a. Some professor smokes.
   b. $\not\exists$ Some linguistics professor smokes.

- Left monotone decreasing relation $=$ *anti-persistent* $=$ left downward monotone

11.7 Nonpersistent determiners

- Some determiner meanings are neither persistent nor antipersistent:

  (50) Most professors smoke.
   a. $\not\exists$ Most linguistics professors smoke.
   b. $\not\exists$ Most people smoke.

- Notice that even some *simple* determiners express nonpersistent relations, even when all simple determiners express (right) monotone relations.

11.8 Montonicity and NPIs

- *Negative polarity items* (NPIs) are expressions that are allowed only in a certain class of contexts, whose central exemplar is the negative context.

When it acts as the semantic counterpart to “some” (rather to to “every,” as in “Any idiot would know that”), then “any” is an NPI.

(51) John ate some apples.
(52) John didn’t eat any apples.
(53) * John ate any apples.
• But the NPI “any” can occur in many contexts that are not obviously negative:

(54)  
\begin{align*}
  & a. \text{ Every kid who ate any apples got sick.} \\
  & b. \text{ At most three sick kids ate any apples.} \\
  & c. \ast \text{ Some kid who ate any apples got sick.} \\
  & d. \ast \text{ At least three sick kids ate any apples.}
\end{align*}

• Bill Ladusaw (1979) proposed that the environments where NPI “any” occurs are exactly the “downward entailing” environments $x \ldots y$, where $xAy$ entails $xBy$, whenever $B \subseteq A$.

The downward-entailing environments include the monotone decreasing environments.

(55) Every: Left monotone decreasing
\begin{align*}
  & a. \text{ Every kid got sick.} \\
  & b. \Rightarrow \text{ Every fat kid got sick.}
\end{align*}

(56) At most: Right monotone decreasing
\begin{align*}
  & a. \text{ At most three kids got sick.} \\
  & b. \Rightarrow \text{ At most three fat kids got sick.}
\end{align*}

(57) Some: Not left monotone decreasing
\begin{align*}
  & a. \text{ Some kid got sick.} \\
  & b. \not\Rightarrow \text{ Some fat kid got sick.}
\end{align*}

(58) At least: Not right monotone decreasing
\begin{align*}
  & a. \text{ At least three kids got sick.} \\
  & b. \not\Rightarrow \text{ At least three fat kids got sick.}
\end{align*}

• Ladusaw’s idea faces some serious challenges (consider (59)), but still, it was an excellent idea.

(59)  
\begin{align*}
  & a. \text{ It’s not true that Al doesn’t want any fruit.} \\
  & b. \not\Rightarrow \text{ It’s not true that Al doesn’t want any apples.} \\
  & c. \text{ Al wants some fruit.} \\
  & d. \not\Rightarrow \text{ Al wants some apples.}
\end{align*}