Quantificational DPs, Part 1: Their Basic Semantics [1]

1. Introduction

Presently, our system can interpret sentences where the following sorts of DPs occupy subject position:

- Proper Names: *Barack, Joe, Michelle*
- Definite DPs: *the man, the president*
- Pronouns: *he, she, it*

(1) **Observation 1:** All of the above DPs are of type e

(2) **Observation 2:**

- These aren't the only kinds of DPs that can occupy subject position
- All the following DPs, for which we don't yet have a semantics, can *also* occupy subject position.

(3) **Some Quantificational DPs**

a. *A / some man* **A / some man** smokes.

b. *No man* **No man** smokes.

c. *Every man* **Every man** smokes.

d. *Three men* **Three men** smoke.

e. *Many men* **Many men** smoke.

f. *Few men* **Few men** smoke.

g. *Most men* **Most men** smoke.

(4) **Observation 3:**

If we extend our system to provide a treatment of the DPs above, then we'll have the tools to interpret just about any DP in the English language!

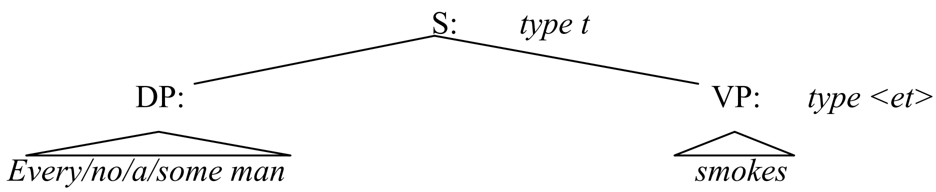
… So let's try to develop a semantics for these DPs…
… We'll start with the first three (3a,b,c), and then work our way out…

[1] These notes are based upon the material in Heim & Kratzer (1998: Chapter 6).

2. The Type of 'Quantificational DPs'

Looking at the structure of sentences (3a,b,c), there are two logical possibilities regarding the semantic type of these DPs:

(5) **The Possible Semantic Types of these DPs**



a. *Possibility 1:* The DPs are type e

b. *Possibility 2:* The DPs are type <et,t>

(6) **Question:**
- Thus far, we've treated DPs in subject position as expressions of *type e*.
- Can we continue to do so here?

(7) **The Classic Answer**
NO! For the following reasons:

2.1 Evidence Against a '*Type e*' Analysis: Argument 1

(8) **Generalization**

- Suppose that we have two VPs – *VP1* and *VP2* – such that for all x, if [[VP1]](x) = T then [[VP2]](x) = T. Moreover, suppose that we have a DP of type e.

- It follows that if [[DP VP1]] = T, then [[DP VP2]] = T.

(9) **Illustration**

- Clearly, for all x, if [[smokes Marlboros]](x) = T, then [[smokes]](x) = T
- And, intuitively, if [[Barack smokes Marlboros]] = T then [[Barack smokes]] = T.

(10) **Key Observation**

The following DOESN'T hold:
if [[**no man** smokes Marlboros]] = T, then [[**no man** smokes]] = T

(11) **Key Conclusion:** The DP *no man* is **not of *type e*.**

2.2 Evidence Against a '*Type e*' Analysis: Argument 2

(12) **Generalization**

- Suppose that we have two VPs – *VP1* and *VP2* – such that for all x, [[VP1]](x) = T *iff* [[VP2]](x) = F. Moreover, suppose that we have a DP of type e.

- It follows that [[DP VP1]] = T *iff* [[DP VP2]] = F.

(13) **Illustration**

- Clearly, for all x, [[smokes]](x) = T *iff* [[doesn't smoke]](x) = F
- And, intuitively, [[Barack smokes]] = T *iff* [[Barack doesn't smoke]] = F

(14) **Key Observation**

The following DOESN'T hold:
[[**a/some man** smokes]] = T *iff* [[**a/some** man doesn't smoke]] = F

That is, the following CAN hold:
[[**a/some man** smokes]] = T *and* [[**a/some** man doesn't smoke]] = T

(15) **Key Conclusion:** The DP *a / some man* is **not of *type e.***

2.3 Evidence Against a '*Type e*' Analysis: Argument 3

(16) **Generalization**

- Suppose that we have two VPs – *VP1* and *VP2* – such that for all x, [[VP1]](x) = T *iff* [[VP2]](x) = F. Moreover, suppose that we have a DP of type e.

- It follows that [[<u>[DP VP1] or [DP VP2]</u>]] is necessarily true.

(17) **Illustration**
Intuitively, the following is necessarily T: "Barack smokes or Barack doesn't smoke."

(18) **Key Observation**

The following doesn't seem to be necessarily T:
"**Every man** smokes or **every man** doesn't smoke."

(19) **Key Conclusion:** The DP *every man* is **not of *type e.***

3. The Semantics of Quantificational DPs and Ds

From the preceding arguments, we come to the following general conclusion:

(20) **General Conclusion**

- None of the following DPs are of *type e*: "a/some man", "no man", "every man"
- Therefore, given (5), it follows that these DPs are of type <et,t>

<u>Side-Note:</u> These arguments can also be used to show that DPs in (3d-g) are also *type <et,t>*

OK… so we know their abstract type…
But exactly what kind of type <et,t> function do these DPs denote???

(21) **Core Intuition**

- As an <et,t> function, the extension of *no man / some man/ every man* takes an <et> property and returns a T-value.
- Thus, the extensions of these DPs are *predicates of predicates* (so-called 'second order predicates/properties)… they 'say things about' their <et> arguments.

(22) **Illustration of Core Intuition**

a. "No man" says that its VP argument is true of no man

- [[no man]]([[VP]]) = T *iff* there is no man x such that [[VP]](x) = T
- [[No man]]([[smokes]]) = T *iff* there is no man x s.t. [[smokes]](x) = T

b. "A / Some man" says that its VP argument is true of some man

- [[a/some man]]([[VP]]) = T *iff* there is some man x such that [[VP]](x) = T
- [[A man]]([[smokes]]) = T *iff* there is some man x such that [[smokes]](x) = T

c. "Every man" says that its VP argument is true of every man

- [[every man]]([[VP]]) = T *iff* for all x, if x is a man, then [[VP]](x) = T
- [[Every man]]([[smokes]]) = T *iff* for all x, if x is a man, then [[smokes]](x) = T

Given the logical formulations above of our 'core intuition', it's clear how we can represent these hypotheses using our lambda notation.

(23) **Hypothesized Extensions of Our Quantificational DPs**

a. [[no man]] = [$\lambda f_{<et>}$: there is no man x such that f(x) = T]

b. [[a/some man]] = [$\lambda f_{<et>}$: there is some man x such that f(x) = T]

c. [[every man]] = [$\lambda f_{<et>}$: for all x, if x is a man, then f(x) = T]

BUT WAIT!!!
Before we go any further with the hypothesized extensions in (23)…
… let's confirm that they do indeed avoid the problems noted for a *type e* analysis in Sections 2.1 – 2.3!!

3.1 Evidence Against a '*Type e*' Analysis: Argument 1

(24) **Earlier Observation**

Contrary to the predictions of a *type e* analysis, the following can hold:

- [[**no man** eats fish]] = T and [[**no man** eats]] = F

(25) **Question:** Does our *type <et,t>* analysis in (23a) predict that (24) can hold?

(26) **Answer: Yes!**

a. Predicted T-conditions for "no man eats fish"

[[no man eats fish]] = T *iff* there is no man x such that x eats fish.

b. Predicted T-conditions for "no man eats"

[[no man eats]] = T *iff* there is no man x such that x eats.

c. *Conclusion:*
Clearly, under these T-conditions, "no man eats fish" can be T at the same time that "no man eats" is F.
(as the world can be such that no man eats fish, but that every man eats *something else other than fish*).

3.2 Evidence Against a '*Type e*' Analysis: Argument 2

(27) **Earlier Observation**

Contrary to the predictions of a *type e* analysis, the following can hold:

- [[**a/some man** eats fish]] = T *and* [[**a/some** man doesn't eat fish]] = T

(28) **Question:** Does our *type* <*et,t*> analysis in (23b) predict that (27) can hold?

(29) **Answer: Yes!**

a. Predicted T-conditions for "some man eats fish"

[[some man eats fish]] = T *iff* there is some man x such that x eats fish.

b. Predicted T-conditions for "some man doesn't eat fish"

[[some man doesn't eat fish]] = T *iff* there's some man x such that x don't eat fish.

c. *Conclusion:*
Clearly, under these T-conditions, "some man eats fish" can be T at the same time that "some man doesn't eat fish" is T.
(...the world can be such that there are two men, one who eats fish and one who doesn't!)

3.3 Evidence Against a '*Type e*' Analysis: Argument 3

(30) **Earlier Observation**

Contrary to the predictions of a *type e* analysis, the following can hold:

- [[[**Every man** eats fish] or [**every man** doesn't eat fish.]]] = F

(31) **Question:** Does our *type* <*et,t*> analysis in (23c) predict that (30) can hold?

(32) **Answer: Yes!**

a. Predicted T-conditions for "every man eats fish or every man doesn't eat fish"

[[S]] = T *iff* for all x, if x is a man, then x eats fish, or
for all x, if x is a man, then x doesn't eat fish.

b. *Conclusion:*
Clearly, under these T-conditions, the sentence in question *can* be false.
(…the world can be such that some men eat fish while others don't)

4. The Semantics of Quantificational Determiners

So far, it seems like the equations in (33) are something our theory should capture…

(33) **Hypothesized Extensions of Our Quantificational DPs**

a. [[no man]] = [$\lambda f_{\langle et\rangle}$: there is no man x such that f(x) = T]
b. [[a/some man]] = [$\lambda f_{\langle et\rangle}$: there is some man x such that f(x) = T]
c. [[every man]] = [$\lambda f_{\langle et\rangle}$: for all x, if x is a man, then f(x) = T]

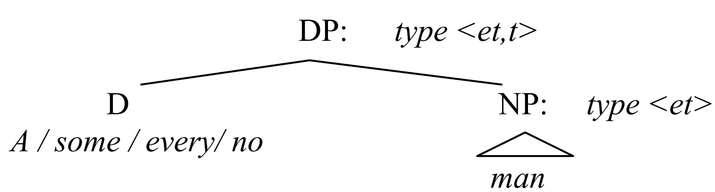
…However, it's also clear that our theory should be able to ***derive*** *these equations…*
... as there seems to be an ***infinite*** *number of such quantificational DPs:*

(34) **Some Other Quantificational DPs**

[[no woman]] = [$\lambda f_{\langle et\rangle}$: there is no woman x such that f(x) = T]
[[no man who smokes]] = [$\lambda f_{\langle et\rangle}$: there is no man who smokes x such that f(x) = T]
[[no dog who barks]] = [$\lambda f_{\langle et\rangle}$: there is no dog who barks x such that f(x) = T]
(*etc., etc.*)

(35) **Goal**: Develop lexical entries for the Ds *no, some, every* that will derive (33).

(36) **The Type of Quantificational Determiners**



CONCLUSION: The semantic type of *a, some, every* and *no* is **< et , < et, t> >**

OK… but what type <et <et,t>> functions are these?

(37) **First Step: Adjusting Our Equations**

If we restate the equations in (33) to the following equivalent forms, it's a little bit clearer how they might be derived via FA.

a. [[no man]] = [$\lambda f_{\langle et\rangle}$: there is no x such that **x is a man** and f(x) = T]
b. [[a/some man]] = [$\lambda f_{\langle et\rangle}$: there is an x such that **x is a man** and f(x) = T]
c. [[every man]] = [$\lambda f_{\langle et\rangle}$: for all x, if **x is a man**, then f(x) = T]

(38) **Second Step: Applying Our Knowledge of Types**

Given that *no*, *some* and *every* are of type <et, <et, t>>, it follows that the equations below are a consequence of our rule of FA:

a. [[no man]] = [[no]]([[man]])
b. [[a/some man]] = [[a/some]]([[man]])
c. [[every man]] = [[every]]([[man]])

Thus, we can rewrite the equations in (37) to the following:

c. [[no]]([[man]]) = [$\lambda f_{<et>}$: there is no x such that **x is a man** and f(x) = T]
d. [[a/some]]([[man]]) = [$\lambda f_{<et>}$: there is an x such that **x is a man** and f(x) = T]
e. [[every]]([[man]]) = [$\lambda f_{<et>}$: for all x, if **x is a man**, then f(x) = T]

Moreover, given that we know what the extension of "man" is, we can rewrite each of these equations to the following:

f. [[no]]([λy_e : y is a man]) =
[$\lambda f_{<et>}$: there is no x such that **x is a man** and f(x) = T]

g. [[a/some]]([λy_e : y is a man]) =
[$\lambda f_{<et>}$: there is an x such that **x is a man** and f(x) = T]

h. [[every]]([λy_e : y is a man]) = [$\lambda f_{<et>}$: for all x, if **x is a man**, then f(x) = T]

(39) **Third Step: Rewriting the Extension to Contain the Argument**

We can rewrite the targeted extensions so that the explicitly contain the <et> argument of the determiners. This will help us see what the extensions of the determiners should be.

a. [[no]]([λy_e : y is a man]) =

[$\lambda f_{<et>}$: there is no x such that **[λy_e : y is a man](x) = T** and f(x) = T]

b. [[a/some]]([λy_e : y is a man]) =

[$\lambda f_{<et>}$: there is an x such that **[λy_e : y is a man](x) = T** and f(x) = T]

c. [[every]]([λy_e : y is a man])

[$\lambda f_{<et>}$: for all x, if **[λy_e : y is a man](x) = T**, then f(x) = T]

(40) **Fourth Step: Stating the Generalizations**

Given the equations in (39), the following generalizations are apparent:

a. [[no]] takes as argument some <et> function g and returns the following function:

[$\lambda f_{<et>}$: there is no x such that **g(x) = T** and f(x) = T]

b. [[a/some]] takes as argument some <et> function g and returns the following function:

[$\lambda f_{<et>}$: there is an x such that **g(x) = T** and f(x) = T]

c. [[every]] takes as argument some <et> function g and returns the following function:

[$\lambda f_{<et>}$: for all x, if **g(x) = T**, then f(x) = T]

(41) **Fifth Step: Writing it Out as a Lambda Expression**

Given the generalizations in (40), it's clear what the extensions of the Ds should be:

a. [[no]] = [$\lambda g_{<et>}$: [$\lambda f_{<et>}$: there is no x such that **g(x) = T** and f(x) = T]

b. [[a/some]] = [$\lambda g_{<et>}$: [$\lambda f_{<et>}$: there is an x such that **g(x) = T** and f(x) = T]

c. [[every]] = [$\lambda g_{<et>}$: [$\lambda f_{<et>}$: for all x, if **g(x) = T**, then f(x) = T]

(42) **Predicted T-Conditions**

The following T-conditional statements follow from the lexical entries in (41):

a. "No man smokes" is T *iff* there is no x such that x is a man and x smokes.

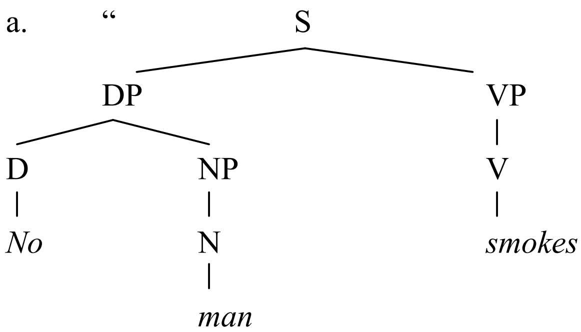
b. "A/some man smokes" is T *iff* there is some x such that x is a man and x smokes.

c. "Every man smokes" is T *iff* for all x, if x is a man, then x smokes.

The following derivation shows how the T-conditions in (42a) are derived…
…the reader is encouraged to also write out their own derivations for (42b) and (42c), to confirm that those T-conditions are indeed derived as stated…

(43) **Sample Derivation**

Let g be any variable assignment.

a. “  ” is T *iff* (by notation)

b. $[[\text{ S }]]^g = T$

c. **Subproof:**
(i) $[[\text{ NP }]]^g$ = (by NNx2, TN)
(ii) $[\ \lambda y_e : \text{y is a man}\ ]$

d. **Subproof:**
(i) $[[\text{ D }]]^g$ = (by NN, TN)
(ii) $[\ \lambda g_{<et>} : [\ \lambda f_{<et>} : \text{there is no x such that } g(x) = T \text{ and } f(x) = T\ ]$

e. **Subproof:**

(i) $[[\text{ DP }]]^g$ = (by FA, c, d)

(ii) $[[\text{ D }]]^g (\ [[\text{ NP }]]^g\ )$ = (by c, d)

(iii) $[\ \lambda g_{<et>} : [\ \lambda f_{<et>} : \text{there is no x such that } g(x) = T \text{ and } f(x) = T\ ]$
$(\ [\ \lambda y_e : \text{y is a man}\ ]\ )$ = (by LC)

(iv) $[\ \lambda f_{<et>} : \text{there is no x such that } [\ \lambda y_e : \text{y is a man}\ ](x) = T \text{ and } f(x) = T\ ]$
= (by LC)

(v) $[\ \lambda f_{<et>} : \text{there is no x such that x is a man and } f(x) = T\ ]$

f. **Subproof:**
(i) $[[\text{ VP }]]^g$ = (by NNx2, TN)
(ii) $[\ \lambda y_e : \text{y smokes}\ ]$

g. $[[\text{ S }]]^g = T$ *iff* (by FA, e, f)

h. $[[\text{ DP }]]^g (\ [[\text{ VP }]]^g\ ) = T$ *iff* (by e, f)

i. [$\lambda f_{<et>}$: <u>there is no x such that x is a man and f(x) = T</u>] ([λy_e : <u>y smokes</u>]) = T *iff* (by LC)

j. there is no x such that x is a man and [λy_e : <u>y smokes</u>](x) = T *iff* (by LC)

k. there is no x such that x is a man and x smokes.

(44) **Interim Summary**

a. Quantificational DPs (*no man, some man, every man*) are of type <et,t>

b. Quantificational Ds (*no, some, every*) are of type <et,<et,t>>

(45) **Some (Confusing) Terminology**

a. *Generalized Quantifiers*: Functions of type <et,t>

b. *Quantificational Determiners:* Functions of type <et,<et,t>>

(46) **Some History**

Many of the advances in semantics and logic throughout the centuries have been made via the study of quantificational DPs and Ds.

Indeed, one of the first, primary results of formal, truth-conditional semantics was the analysis it was able to provide for the compositional semantics of quantificational DPs in English (Montague 1974).

This particular treatment of English quantificational DPs has many parents, but some of the most important are:

- Mostowski (1957) "On a Generalization of Quantifiers"
- Lewis (1972) "General Semantics"
- Montague (1974) "The Proper Treatment of Quantification in Ordinary English"
- Barwise & Cooper (1981) "Generalized Quantifiers and Natural Language"

4. Some Formal Properties of Quantificational Determiners in Natural Language

(47) **Question (Barwise & Cooper 1981)**
The set of possible <et <et,t>> functions is *huge*. Are there any properties that seem to characterize the quantificational determiners that are employed by natural language?

(48) **Some Formal Properties of Relations Between Sets**

a. Symmetry:
D is symmetric *iff* for all A,B, if D(A)(B) = T, then D(B)(A) = T
(*example:* 'some') (*non-example:* 'every')

b. Transitive:
D is transitive *iff* for all A,B,C, if D(A)(B) =T and D(B)(C) =T, then D(A)(C) = T
(*example:* 'every') (*non-example:* 'some')

c. Left Upward Monotone:
D is left upward monotone *iff* for all A,B, if D(A)(B) = T and
$\{x: A(x) = T\} \subseteq \{y: C(y) = T\}$, then D(C)(B) = T.
(*example:* 'some') (*non-example*: 'no')

d. Right Upward Monotone:
D is right upward monotone *iff* for all A,B, if D(A)(B) = T and
$\{x: B(x) = T\} \subseteq \{y: C(y) = T\}$, then D(A)(C) = T.
(*example:* 'some') (*non-example*: 'no')

e. Left Downward Monotone:
D is left downward monotone *iff* for all A,B, if D(A)(B) = T and
$\{y: C(y) = T\} \subseteq \{x: A(x) = T\}$, then D(C)(B) = T.
(*example:* 'no') (*non-example:* 'some')

f. Right Downward Monotone:
D is right downward monotone *iff* for all A, B, if D(A)(B) = T and
$\{y: C(y) = T\} \subseteq \{x: B(x) = T\}$, then D(A)(C) = T.
(*example:* 'no') (*non-example:* 'some')

What's the point of this sort of classification?

- As you'll see in the next problem set, there are linguistic phenomena that seem to be sensitive to some of these properties…

- Also, there is one formal property that has been argued to be a universal property of all natural language determiners…

(49) **Conservativity**
A quantificational D is conservative *iff* for all A, B,
D(A)(B) = T *iff* D(A)([λx : A(x) = T and B(x) = T]) = T

i.e., D holds for A and B iff D holds for A and ***all those As that are B***

(50) **Claim (Barwise & Cooper 1981, Keenan & Stavi 1986):**

All natural language determiners are conservative.

(51) **The Interest of the Claim**

a. No Clear Counter Examples
All the (uncontroversial) determiners in English seem to be 'conservative'.

(i) "some cats breathe" = T *iff* "some cats are cats who breathe" = T
(ii) "no cats breathe" = T *iff* "no cats are cats who breathe" = T
(iii) "every cat breathes" = T *iff* "every cat is a cat who breathes" = T

b. Possible Counter-Examples are Imaginable

(i) The DP 'only cats' in English is non-conservative.
- "only cats breathe" = F
- "only cats are cats who breathe" = T

(ii) However, the element "only" in English is not a D.
- You cannot stack Ds in English (**every the cat*), but "only" can take a DP as sister (*only the cats*)
- "only" can take all sorts of things as sister, including VPs (*Dave only* [*kissed Mary*])

c. Has a Kind of 'Functionalist' Rationale

(i) If D is *conservative*, then to know whether 'D(A)(B)' holds, you need only look at A and *those As which are B.*

(ii) Thus, to know if 'D(A)(B)' = T, you don't need to ever look beyond the set A. (You look at A, and then consider how the whole set relates to that subset of A which is B).

(iii) *…and there seems to be something rather computationally efficient about that (maybe)…*

5. Presuppositions and Quantificational Determiners

It seems that some quantificational Ds in natural language carry presuppositions regarding the extensions of their NP arguments.

(52) **First Case Study: *Neither***

a. Observation 1:
Suppose that there are two cats. Then "Neither cat is hungry" seems T iff "no cat is hungry" is T, iff there is no x such that x is a cat and x is hungry.

b. Observation 2:
- "Neither cat is hungry" implies that there are two cats.
- "It's not the case that neither cat is hungry" *still* implies that there's two cats. (it still feels 'wrong' if there is either only one cat or more than two cats)
- *So,* "Neither cat is hungry" seems to *presuppose* that there are (exactly) two cats…

(53) **Some Notation: The 'Cardinality' of a Set**
The *cardinality* of a set *A* – which we can represent with the notation '|*A*|' – is the number of members that A has.

a. Examples
(i) $| \{a, b, c\} | = 3$
(ii) $| \{a, b, c , d, a\} | = 4$
(iii) | { x : x is a first year linguistics graduate student } | = 5

(54) **Lexical Entry for *Neither***

[[neither]] = [λg : g ∈ $D_{<et>}$ and **| { x : g(x) = T } | = 2** .
[$\lambda f_{<et>}$: there is no x such that g(x) = T and f(x) = T]]

(55) **More Controversial Example: *Every***

a. Fact 1:
- "Every unicorn is beautiful" implies that there are unicorns.
- "It's not the case that every unicorn is beautiful" *still* implies that there are unicorns.
- *So*. "Every unicorn is beautiful" seems (via this test) to *presuppose* that there do exist unicorns.

b. Fact 2:
There is a significant population who find the following to be a valid inference:
If "Every NP VP" is T, then "Some NP VP" is T.
(e.g., Aristotle thought this was valid.)

(56) **Problem: Failure of Prediction**

Our current semantics for *every* (41c) – repeated below – does not predict the facts in (55). Note that given the semantics in (a), the sentence in (b) will have the T-conditions in (c), which will hold even in situations where *there aren't actually any unicorns.*

a. [[every]] = [$\lambda g_{<et>}$: [$\lambda f_{<et>}$: for all x, if **g(x) = T**, then f(x) = T]

b. [[every unicorn is beautiful]] = T *iff*

c. For all x, if x is a unicorn, then x is beautiful.

(57) **Common Solution: Write in the Presupposition**

The determiner *every* has as its extension a function whose domain is restricted to those <et> functions that are true of *at least one entity*.

[[every]] = [λg : $g \in D_{<et>}$ and **| { x : g(x) = T } | ≥ 1** .
[$\lambda f_{<et>}$: for all x, if g(x) = T, then f(x) = T]]

a. *First Consequence*
"Every NP VP" will now presuppose that there exists at least one x such that [[NP]](x) = T

b. *Second Consequence*
Given its new presupposition, the T-conditions of "Every NP VP" now entail "Some NP VP".

This difference between "every" and "no/some" also extends to other quantificational Ds…

(58) **The Determiner *Most* Seems to Behave like *Every***

"Most unicorns live at UMass." *Implies there are unicorns.*
"It isn't the case that most unicorns live at UMass." *Still implies that there are unicorns.*

(59) **Some Other Determiners that Seem to Behave like *Some* and *No***

a. Numerals
"It isn't the case that two unicorns live at UMass." *Straightforwardly true; Does not imply there are unicorns*

b. *Many*
"It isn't the case that many unicorns live at UMass." *Straightforwardly true; Does not imply there are unicorns*

We can similarly write this difference in presuppositional behavior into the lexical items of these determiners.

(60) **Some Lexical Entries:**

a. [[most]] = [$\lambda g : g \in D_{<et>}$ and **$|\{ x : g(x) = T \}| \geq 1$** .
[$\lambda f_{<et>}$: $\underline{|\{ x : g(x) = T \text{ and } f(x) = T \}| > |\{ y : g(y) = T \text{ and } f(y) = F \}|}$]]

‘*There are more NPs that VP than NPs that don’t VP.*’

b. [[three]] = [$\lambda g_{<et>}$: [$\lambda f_{<et>}$: $\underline{|\{ x : g(x) = T \text{ and } f(x) = T \}| \geq 3}$]]

‘*There are at least three entities for which the NP and the VP is T.*’

(61) **Key Observation**
If a determiner D requires its first argument to be true of some entities, then the following existential sentence is ill-formed: “there is/are D NP (Location)”.

a. Evidence

(i) There is a/some cat (in my yard).
(ii) There is no cat (in my yard).
(iii) There are three cats (in my yard).
(iv) There are many cats (in my yard).

(v) * There is every cat (in my yard).
(vi) * There are most cats (in my yard).
(vii) * There is neither cat (in my yard).
(viii) *There is the cat (in my yard).

(62) **Syntactic Terminology**

a. Strong Determiners (Syntactic Property):
A determiner D is ‘strong’ if the following existential sentence is ill-formed: “there is/are D NP (Location)”.

b. Weak Determiners (Syntactic Property):
A determiner D is ‘weak’ if the following existential sentence is well-formed: “there is/are D NP (Location)”.

(63) **Empirical Generalization (Falsifiable)**
A determiner is ‘strong’ *iff* it presupposes that its first argument is true of something