XIII. NUMERICALS AND THE DEFINITE ARTICLE

Expressing numericals in predicate logic.
We are interested here in expressive capacity:

property P can be expressed in predicate logic iff there is a predicate logical formula φ such that for every predicate logical model M: P holds in M iff $[\![\varphi]\!]_M = 1$.

So we are not concerned *here* with interpreting compositionally a natural language expression like, say, *more than seven but less than twelve cats are smart*, but only with the question whether *the truth conditions* of that sentence can be expressed at all with some formula of predicate logic. We will see that the answer is yes for this example, but no for other examples, like *most cats are smart*.

(1) At least one cat is smart.

(2) At least two cats are smart.

(3) At least three cats are smart.

(4) At most one cat is smart.

(5) At most two cats are smart.

(6) At most three cats are smart.

(7) Exactly n cats are smart = ?

Expressing numericals in predicate logic

(1) At least one cat is smart.

$\exists x[CAT(x) \wedge SMART(x)]$

(2) At least two cats are smart.

$\exists x \exists y[CAT(x) \wedge SMART(x) \wedge CAT(y) \wedge SMART(y) \wedge \neg(x=y)]$

(3) At least three cats are smart.

$\exists x \exists y \exists z[CAT(x) \wedge SMART(x) \wedge CAT(y) \wedge SMART(y) \wedge CAT(z) \wedge SMART(z) \wedge \neg(x=y) \wedge \neg(x=z) \wedge \neg(y=z)]$

(4) At most one cat is smart.

$\forall x \forall y[CAT(x) \wedge SMART(x) \wedge CAT(y) \wedge SMART(y) \rightarrow (x=y)]$

(5) At most two cats are smart.

$\forall x \forall y \forall z[CAT(x) \wedge SMART(x) \wedge CAT(y) \wedge SMART(y) \wedge CAT(z) \wedge SMART(z) \rightarrow [(x=y) \vee (x=z) \vee (y=z)]]$

(6) At most three cats are smart.

$\forall x \forall y \forall z \forall u[CAT(x) \wedge SMART(x) \wedge CAT(y) \wedge SMART(y) \wedge CAT(z) \wedge SMART(z) \wedge CAT(u) \wedge SMART(u) \rightarrow [(x=y) \vee (x=z) \vee (x=u) \vee (y=z) \vee (y=u) \vee (z=u)]]$

(7) Exactly n cats are smart = at least n cats are smart $\wedge$ at most n cats are smart.

Russell:

(7) The cat is smart.

$\exists x[CAT(x) \wedge \forall y[CAT(y) \rightarrow (x=y)] \wedge SMART(x)]$
There is one and only one cat and that cat is smart.

Problem:

(8) The cat isn't smart

Russell:

$\neg\exists x[CAT(x) \wedge \forall y[CAT(y) \rightarrow (x=y)] \wedge SMART(x)]$
$\forall x[CAT(x) \wedge \forall y[CAT(y) \rightarrow (x=y)] \rightarrow \neg SMART(x)]$
If there is one and only one cat, that cat isn't smart

But (8) also implies that there is a cat, and the negation of Russell's formula doesn't.

Frege,Strawson: The existence and uniqueness are not asserted but presupposed.

Add to L_3:
If $P \in PRED^1$, then $\sigma(P) \in TERM$
Semantics:

$$[\![\sigma(P)]\!]_{M,g} = \begin{cases} d & \text{if } [\![P]\!]_{M,g} = \{d\} \\ \bot & \text{otherwise} \end{cases}$$

$\bot$ stands for *undefined*
If there is an individual $d \in D_M$ such that d is the one and only object in D_M that has P (rel g), i.e. $[\![P]\!]_{M,g} = \{d\}$, then $\sigma(P)$ denotes d in M (rel g), $[\![\sigma(P)]\!]_{M,g} = d$.
If for no $d \in D_M$ $[\![P]\!]_{M,g} = \{d\}$, the $\sigma(P)$ is undefined in M rel. g.

This is the case when $[\![P]\!]_{M,g} = \emptyset$ (existence failure),
or if $|[\![P]\!]_{M,g}| > 1$ (uniqueness failure)

This requires a three valued semantics which allows the truth value of expression to be undefined. Example:

(8) The cat is smart.
SMART(σ(CAT))

$$[\![SMART(\sigma(CAT)]\!]_{M,g} = \begin{cases} 1 & \text{if } [\![\sigma(CAT)]\!]_{M,g} \in F_M(SMART) \\ 0 & \text{if } [\![\sigma(CAT)]\!]_{M,g} \in D_M - F_M(SMART) \\ \bot & \text{otherwise} \end{cases}$$

(8) is undefined if there is no cat, and also if there is more than one cat.

The use of an expression to talk about a situation M presupposes that it is defined in M. Hence the use of (8) to talk about M, presupposes that F_M(CAT) is a set with exactly one element, a **singleton set**.

Existence failure: #Though I don't have a cat, the cat I have is white.
Uniqueness failure: #

Other example: You walk through an alley. There are two cats sitting on a garbage can.
You say: The cat is white. # infelicitous.

Similar modifications are needed for sentences involving n-place relations and identity statements.

Also the connectives need to be modified.

Three valued negation:

$$\neg\begin{pmatrix} 0 \to 1 \\ 1 \to 0 \\ \bot \to \bot \end{pmatrix}$$

Now it follows that both WHITE(σ(CAT)) and ¬WHITE(σ(CAT)) presuppose that there is a unique cat (in the context).

presuppositions of speech acts (Stalnaker): assertion, denial, questioning, supposition.

Assertion: Your cat is white.
Denial: Your cat isn't white
Questioning: Is your cat white?
Supposition: If your cat is white, the this is not your cat.

But cf: If he has a cat, his cat is young
(presupposition satisfied *inside* the sentence)

Further modifications of the semantics: strong Kleene three values semantics for connectives and quantifiers.
Strong Kleene three valued truth tables:

$\wedge$ φ ψ	1	0	⊥
1	1	0	⊥
0	0	0	**0**
⊥	⊥	**0**	⊥

$\vee$φ ψ	1	0	⊥
1	1	1	**1**
0	1	0	⊥
⊥	**1**	⊥	⊥

Generalization to quantifiers:

$$[\![\forall x\varphi]\!]_{M,g} = \begin{cases} 1 \text{ iff for every } d \in D_M\text{: } [\![\varphi]\!]_{M,g_x^d} = 1 \\ 0 \text{ iff for some } d \in D_M\text{: } [\![\varphi]\!]_{M,g_x^d} = 0 \\ \bot \text{ otherwise} \end{cases}$$

$$[\![\exists x\varphi]\!]_{M,g} = \begin{cases} 1 \text{ iff for some } d \in D_M\text{: } [\![\varphi]\!]_{M,g_x^d} = 1 \\ 0 \text{ iff for every } d \in D_M\text{: } [\![\varphi]\!]_{M,g_x^d} = 0 \\ \bot \text{ otherwise} \end{cases}$$

These clauses generalize the clauses for $\wedge$ and $\vee$.

Metalinguistics negation

What about the following counterexample:

(1) A letter to the Times:

Sir. In contradiction to what was written in the Times yesterday, the president of Belgium was *not* sent to prison, because, as you ought to know, Belgium is a monarchy.

No contradiction.

Metalinguistic negation. (Horn 1985)

In Dutch –n after schwa is not pronounced, except in Groningen, despite what the crazy new spelling reform rules try to make you believe.
You are ordering pancakes in an Amsterdam pancake restaurant, and you read your order aloud from the menu (in new spelling) to the waiter. The waiter says to you with a sneer:

(2) Sir, we do not have "panne**n**koek" on the menu, we only have "pannekoek" on the menu (we are in Amsterdam here).

This doesn't mean: it is not true that we have pancakes on the menu, the waiter accepts that there are pancakes on the menu, but objects to some other aspect of the utterence, like:

-a presupposition (1)
-the pronunciation (2)
-the register (3)

(3) Larry Horn's example:

No Johnny, Phideau didn't shit on the rug. he defecated on the carpet.

These cases are instances of metalinguistic negation (Horn):

Metalinguistic negation: Use negation to **object** to **some** aspect of the utterance other than truth value.

Like pronounciation, register, or indeed: presupposition.

Presupposition

Assertion, denial, questioning, supposition of φ presuposes p.

The above cases are counterexamples.
But this is not the standard use of negation.

Stalnaker: we only expect presuppositions for an utterence, ***if that utterence is intended as an assertion.***

Example with conditionals:
Conditional assertion:

> If the president opened partiament yesterday, then today is Wednesday.

Presupposition: there is a president.

Meta reasoning about what the antecendent presupposes (not conditional assertion)"

> If, as you say, the president opened partiament yesterday, then that means that there *is* a president.

No presupposition that there is a president **for the whole sentence.**

WHY *MOST* IS NOT FIRST ORDER DEFINABLE

CAT ∩ SMART	CAT – SMART

Let φ = *Every Cat is Smart*

For this example we ignore the irrelevant set D_M – CAT.
We start with CAT = Ø, so CAT ∩ SMART = Ø and CAT – SMART = Ø

D = Ø

CAT ∩ SMART	CAT – SMART

φ = *every cat is smart*

φ is true

Now we add an object d_1: CAT = {d_1}.

I can put d_1 in CAT ∩ SMART or in CAT – SMART

CAT ∩ SMART	CAT – SMART
d_1	

situation 1a: φ is true

CAT ∩ SMART	CAT – SMART
	d_1

situation 1b: φ is false

Now we add a second object d_2: CAT = {d_1, d_2}

The options that we have are:
Starting in situation 1a or in 1b we can put d_2 in the intersection or the difference

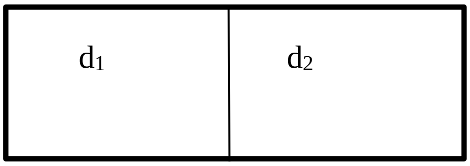
CAT ∩ SMART	CAT – SMART
d_1, d_2	

situation (1a)a φ = true

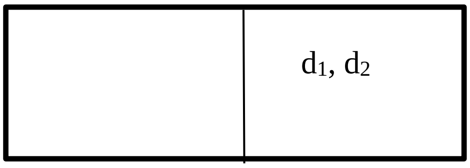
CAT ∩ SMART	CAT – SMART
d_2	d_1

Situation (1b)a φ is false

CAT ∩ SMART CAT – SMART



CAT ∩ SMART CAT – SMART



situation (1a)b φ is false

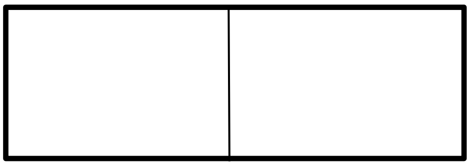
situation (1b)b φ is false

What we see is this: φ starts out as true when CAT = Ø.
As long as we put new objects in CAT ∩ SMART, φ stays true.
We can make the truthvalue **flip** by putting a new object in CAT – SMART, φ becomes false.
Once we have put an object in CAT – SMART, new objects will not affect the truth conditions any more: once false, φ stays false when adding objects.
We see:
the truth value of φ can flip in the process of adding object to the domain at most once, from true to false.

If *every cat is smart* is true on domain D, it can become false by adding a cat, but as soon as it is false on a domain, no matter how many individuals I add to the domain, *every cat is smart* stays false (i.e. one non-smart cat is enough).

The same holds for a sentence like ψ = *Some cat is smart* for inverse reasons:

D = Ø
CAT ∩ SMART CAT – SMART



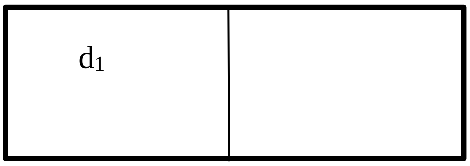
ψ = *some cat is smart*

ψ is false

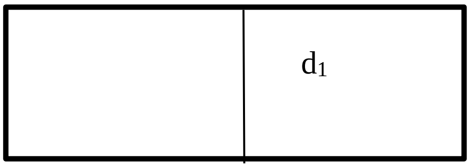
Now we add an object d_1: CAT = $\{d_1\}$.

I can put d_1 in CAT ∩ SMART or in CAT – SMART

CAT ∩ SMART CAT – SMART



CAT ∩ SMART CAT – SMART



situation 1a: ψ is true

situation 1b: ψ is false

Now we add a second object d_2: CAT = $\{d_1, d_2\}$

The options that we have are:

Starting in situation 1a or in 1b we can put d_2 in the intersection or the difference

CAT ∩ SMART	CAT – SMART
d_1, d_2	

situation (1a)a ψ = true

CAT ∩ SMART	CAT – SMART
d_2	d_1

Situation (1b)a ψ is true

CAT ∩ SMART	CAT – SMART
d_1	d_2

situation (1a)b ψ is true

CAT ∩ SMART	CAT – SMART
	d_1, d_1

situation (1b)b ψ is false

What we see is this: ψ starts out as false when CAT = Ø.
As long as we put new objects in CAT – SMART, φ stays false.
We can make the truthvalue **flip** by putting a new object in CAT ∩ SMART, ψ becomes true.
Once we have put an object in CAT ∩ SMART, new objects will not affect the truth conditions any more: once true, φ stays true when adding objects.
We see:
the truth value of ψ can flip in the process of adding object to the domain at most once, from false to true.

If *some cat is smart* is false on domain D, it can become true by adding a cat, but as soon as it is true on a domain, no matter how many individuals I add to the domain, *some cat is smart* stays true (i.e. one smart cat is enough).

Other sentences can flip **more than once.**

Take *Exactly three cats are smart*.

On a domain of less that three individuals,, the sentence is false.
I can make it true once I have three individuals (**flip one**): put d_1 in CAT ∩ SMART, then put d_2 in CAT ∩ SMART, *Exactly three cats are smart* is still false.
Now put d_3 in CAT ∩ SMART, and *Exactly three cats are smart* becomes true.
We can postpone making it flip from false to true by adding as many objects as you want to CAT – SMART, but as soon as you add one more to CAT ∩ SMART, *Exactly three cats are smart* becomes true.
After that, I can keep it true by adding objects only to CAT – SMART.
But I can make it false again, by adding one more object to CAT ∩ SMART: **flip two**.
But that's is, once I have added a fourth object to CAT ∩ SMART, the sentence will stay false what everymore you do.

A sentence like *exactly 3 cats or exactly 10 cats are smart* can flip four times.

This leads to the question:
For an arbitrary predicate logical sentence, how many times can it flip?

The answer is given in a theorem:

Theorem: Every predicate logical sentence can flip maximally **a finite number of times**, meaning: for each predicate logical sentence φ there is a boundary number $n_φ$, which is the number of times that φ can flip (this number can actually be computed for each sentence)

Barwise and Cooper 1980 Linguistics and Philosophy
Now look at φ = *Most cats are smart*.
The truth conditions say: $|CAT \cap SMART| > |CAT - SMART|$

We start out with a domain on which φ is true.

-Add non-smart cats to make the numbers equal:	φ flips: φ is false.
-Add a smart cat:	φ flips: φ is true
-Add a non-smart cat:	φ flips: φ is false.
-Add a smart cat:	φ flips: φ is true

etc…

Hence, for φ = *most cats are smart* the truth value of φ can continue to flip: there is no number $n_φ$ where n is the maximal number of flips that φ makes.

This means, by the theorem, that there is no predicate logical formula which is equivalent to *Most cats are smart,* because for all predicate logical formulas there is such a number.
This means that *most* is not first order definable.

The proof of the above theorem is nasty, it involves keeping track of quantifier depth, quantifiers embedded into other quantifiers.

(1) a. ***Finitely many*** angels stand on the tip of a pin.
b. ***Infinitely many*** angels stand on the tip of a pin

The following theorem is a straighforward consequence of the basic completeness theorem for predicate logic, the theorem that says that every valid inference can be derived in the proof theory of predicate logic.

If Δ is a set of sentences we say that M is a model for Δ if all the sentences in Δ are true on M.
We say that that a model M has cardinality n iff $|D_M| = n$
Thus, a finite model is a model with a finite domain.

Theorem: If a set of sentences has arbitrarily large finite models, it has an infinite model.

We use this theorem to prove that, while we can express all sorts of cardinality statements in predicate logic, we cannot express that the domain is finite or that the domain is infinite.

We look at the following three sentences:

(a) $\forall x[\text{Angel}(x)]$
(b) $\forall x[\text{Angel}(x) \rightarrow \text{Stand-on-this-pin}(x)]$
(c) Only finitely many angels stand on this pin.

Our set of sentences is Δ = {a,b,c}
Hence in any model for Δ there are only angels (by a), and they all stand on this pin by (b).
Obviously, if I take a domain with one object, specify that it is an angel and that it is standing on this pin, I have a model for Δ, because it is also true that only finitely many angels stand on this pin in this model, namely one.
Now, obviously, I can do this for any finite domain: if I interpret all the objects as angels standing on this pin, I have a model for Δ.
This means that Δ has arbitrarily large finite models.

By the theorem, it means that **if (c) is definable in predicate logic**, then Δ has an infinite model M^{inf},whose domain is the infinite set D^{inf}.

Since this model is a model for Δ, everything in it is an angel standing on this pin and since D^{inf} is infinite, infinitely many angels stand on this pin.
But that means that it is false that only finitely many angels stand on this pin, so M^{inf} isn't a model for Δ after all. This can only be, if there is no predicate logical sentence defining (c), and that means that the notions *finiteness/infinite* are not definable.

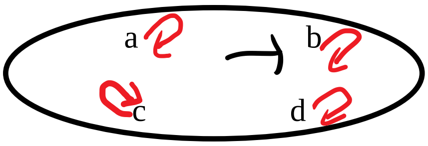
So the determiners *finitely many* and *infinitely many* are not definiable.

All cats but at most 12 are smart

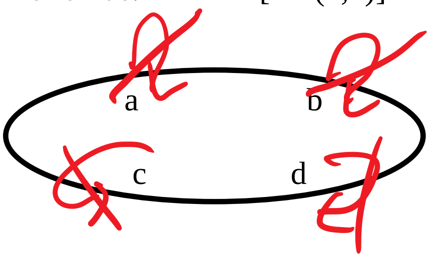
XIV. ORDER RELATIONS

Let R be a two-place relation.

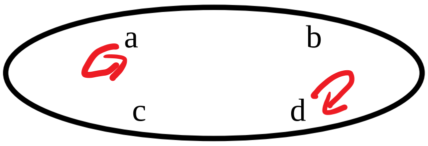
R is **reflexive**: $\forall x[R(x,x)]$ *resemble, be as old as*



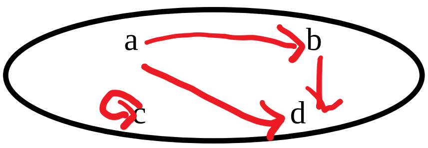
R is **irreflexive**: $\forall x[\neg R(x,x)]$ *precede, sit next to, be younger than*



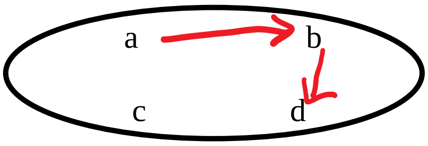
R is **non-reflexive**: $\exists x[\neg R(x,x)]$ *love*
R is **not irreflexive**: $\exists x[R(x,x)]$



R is **transitive**: $\forall x \forall y \forall z[R(x,y) \wedge R(y,z) \rightarrow R(x,z)]$ *precede, be taller than*
be part of



R is **intransitive:** $\forall x \forall y \forall z[R(x,y) \wedge R(y,z) \rightarrow \neg R(x,z)]$ *be one year older than*

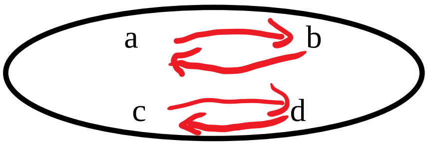


R is **non-transitive**: $\exists x \exists y \exists x[R(x,y) \land R(y,z) \land \neg R(x,z)]$

love, fit in, resemble, is a neighbour of

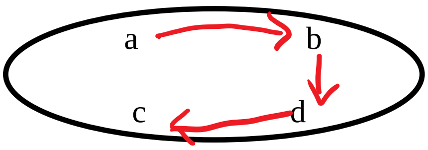
R is **symmetric:** $\forall x \forall y[R(x,y) \rightarrow R(y,x)]$

resemble, is a neighbour of
sit next to (between people)
stand next to (houses)
is equally old as



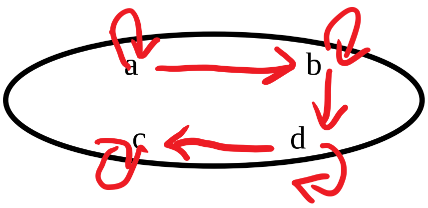
R is **asymmetric:** $\forall x \forall y[R(x,y) \rightarrow \neg R(y,x)]$

is younger than



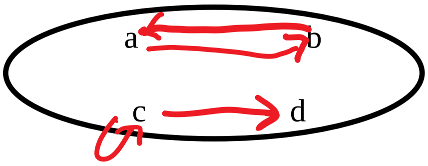
R is **antisymmetric:** $\forall x \forall y[R(x,y) \wedge R(y,x) \rightarrow (x=y)]$

is part of

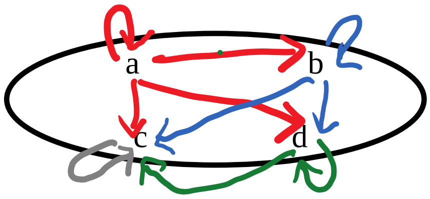


R is **non-symmetric:** $\exists x \exists y[R(x,y) \land \neg R(y,z)]$

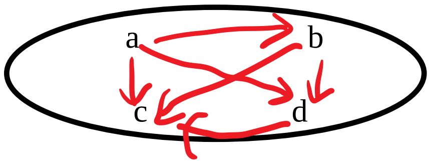
love
sit next to (unlike categories)
I sit next to an empty chair
The house stands next to the lake.



R is **connected:** $\forall x \forall y[R(x,y) \vee R(y,x)]$



R is **s-connected**: $\forall x \forall y[R(x,y) \vee R(y,x) \vee (x=y)]$

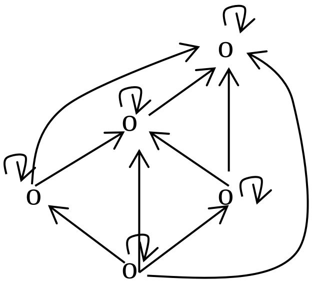


R is a **pre-order:** R is reflexive and transitive
R is a **partial order**: R is reflexive and transitive and antisymmetric.
R is a **strict partial order:** R is irreflexive and transitive and asymmetric.

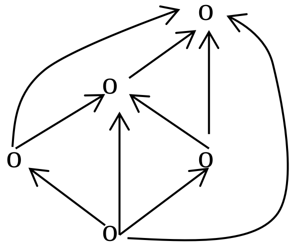
R is a **total or linear order:** R is a connected partial order.
R is a **strict total order**: R is an s-connected partial order.

R is an **equivalence relation**: R is reflexive and transitive and symmetric.

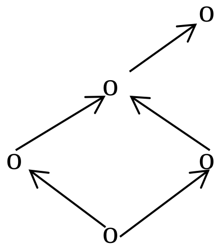
Partial order:



(ir)reflexivity understood:



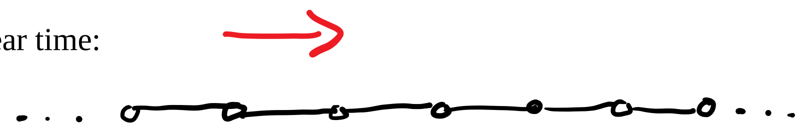
Transitivity understood:



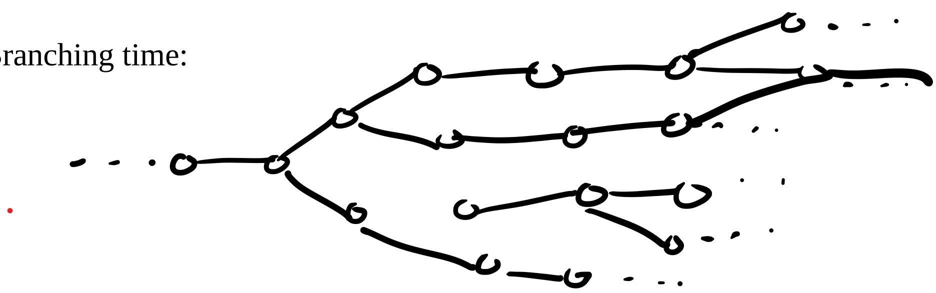
Direction of the graph understood:

Typical examples:

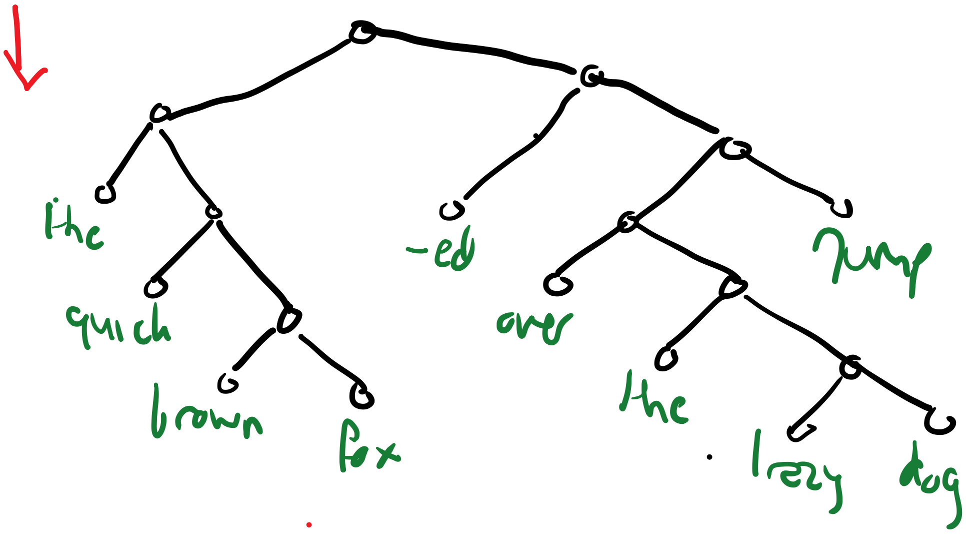
Linear time:



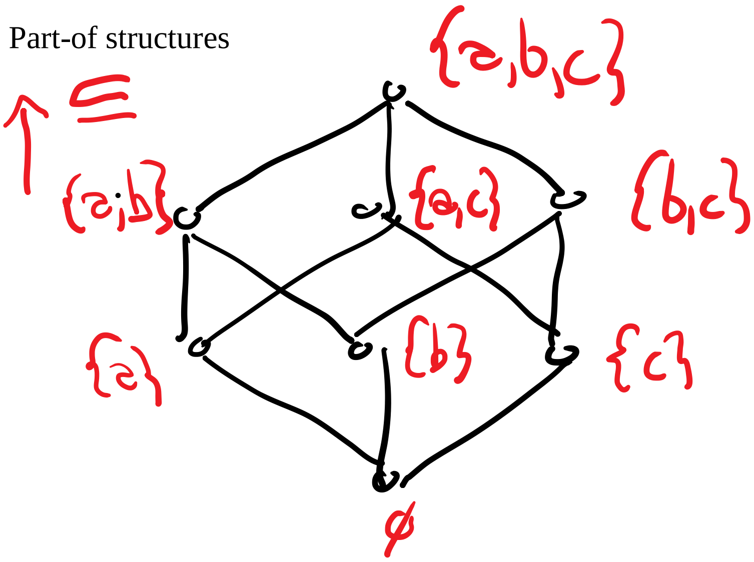
Branching time:



Trees:



Part-of structures



We indicate partial orders with variants of $\leq$, $\sqsubseteq$, $\subseteq$ and strict partial orders with variants of $<$, $\sqsubset$, $\subset$.
Let $<$ be a (strict) linear order.

$<$ has a **begin point** iff $\exists x \forall y[\neg(y < x)$
$<$ is **left continuing** iff $<$ has no begin point
$<$ has an **end point** iff $\exists x \forall y[\neg(x < y)$
$<$ is **right continuing** iff $<$ has no end point

$<$ is **dense** iff $\forall x \forall y[(x < y) \rightarrow \exists z[(x < z) \wedge (z < y)]]$
Between every two points there is a third point

$<$ is **discrete** iff $\forall x[\exists y[x < y] \rightarrow \exists y[(x < y) \wedge \neg \exists z[(x < z) \wedge (z < y)]] \wedge$
$\forall x[\exists y[y < x] \rightarrow \exists y[(y < z) \wedge \neg \exists z[(y < z) \wedge (z < x)]] \wedge$

In a linear order we call $\{y: x < y\}$ the set of **successors** of x in $<$ and
$\{y: y < x\}$ the set of **predecessors** of x in $<$.

If x has successors in $<$ then it has a direct successor, and if x has predecessors then it has a direct predecessor.

The order of **natural numbers** $\mathbb{N} = \{0,1,2,3,...\}$ ordered by $<$.

The **first order theory of natural numbers** (ordered by $<$) is:
$<$ is a discrete linear order which has a begin point and is right continuing.

The order of natural numbers is called the **Standard Model** of the first order theory of natural numbers, and in fact, of any theory of natural numbers.

We would like to find a set of axioms that define the natural numbers, i.e. that are true at the standard model and only at the standard model. But:

Fact: there is no first order theory that defined the order of natural numbers.
This means:
Any first order theory is going to be true on **non-standard models** as well.

The reason: The order of natural numbers is **continuous**, it allows **no gaps** (two sets approaching each other but never reaching). Continuity cannot be defined in first order predicate logic.

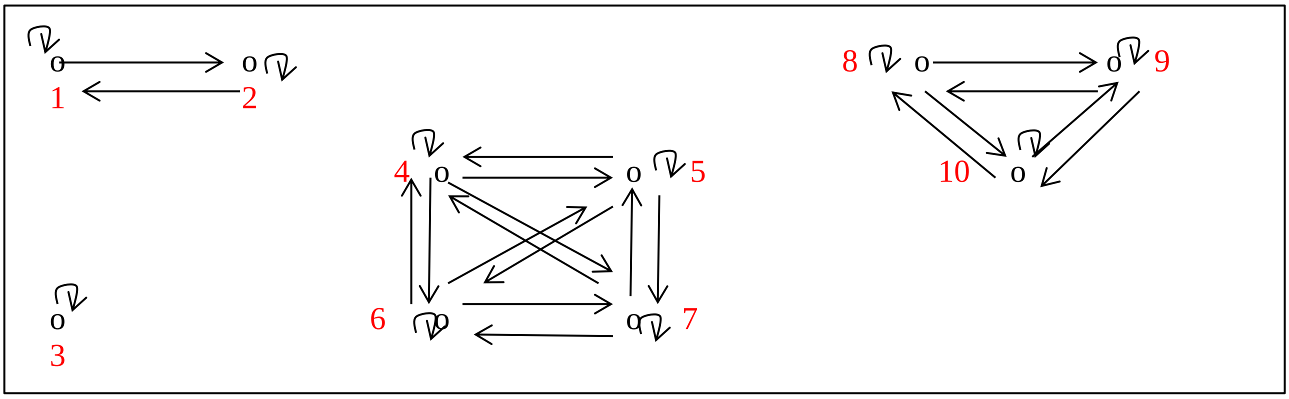
Standard model:
0 1 2 3 4 5 6 7 8 ....

Non-standard model:
0 1 2 3 4 5 6 7 8 ....–4' –3' –2' –1' 0' 1' 2'.....

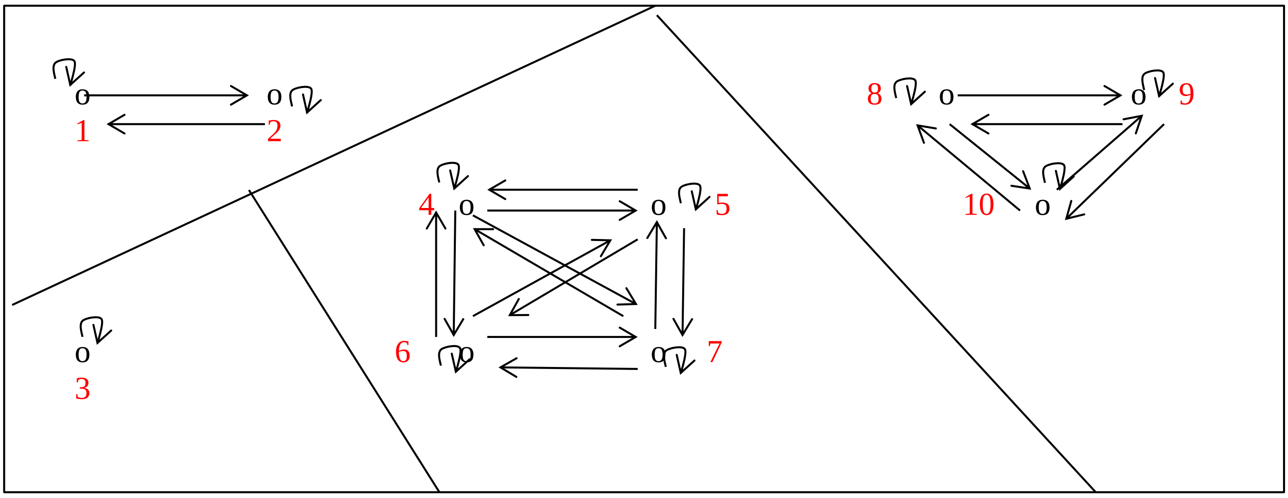
The natural numbers with a copy of the integers after it satisfies all the same first order axioms as the standard model

Equivalence relations and partitions: *is as old as* + age classes

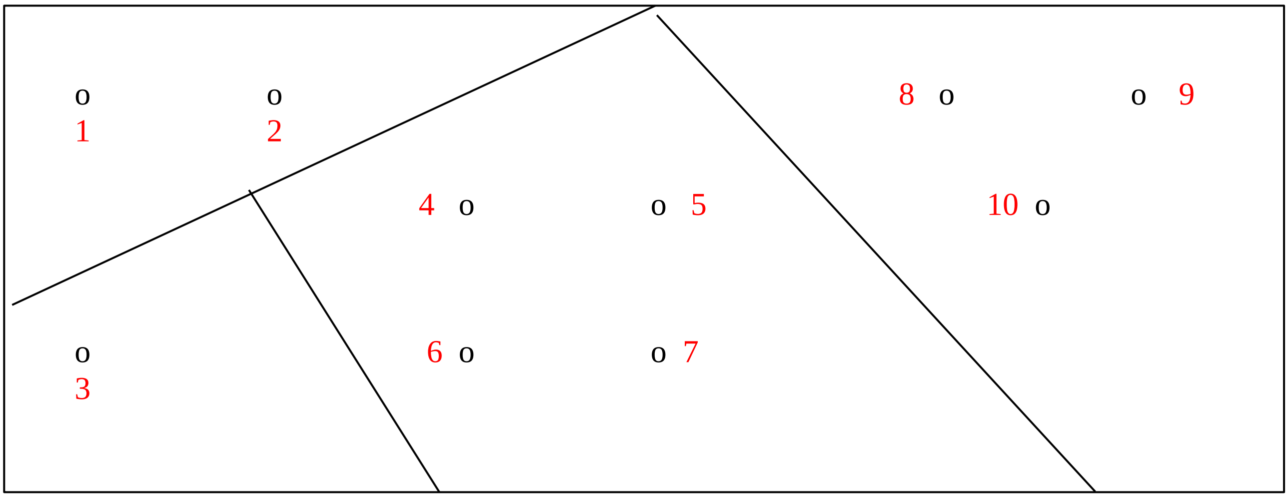
An equivalence relation on {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}:



Drawing lines around the islands:



A partition on {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}



XV. AMBIGUITY

1 Lexical ambiguity.

(1) I took my money to the bank.	Reading 1: And deposited it there.
	Reading 2: And buried it there.

Ambiguity of the lexical meaning of *bank*.

A. Lexical drift: adjective *knap*

Flemish: *knap* =	1. Intelligent	Een knap meisje – an intelligent girl
	2. Admirable	Een knappe prestatie – an admirable achievement
	3. Skilful	Een knappe chirurg – a skillful surgeon
Dutch: *knap* =	*0. Pretty*	Een knap meisje – a pretty girl
	1. Intelligent	
	2. Admirable	
	3. Skilful	
In certain idioms	*4. Narrow*	Een knap halfuurtje – 'a narrow half an hour'
		Narrowly, half an hour

German: *knapp* = 4 By a narrow margin, Narrow, Short, Brief,
i..e *Knapp verfehlt* - Just failed (exam result)
lexicography

B. Systematic Lexical Ambiguity

MASS AND COUNT NOUNS
(using recent work by Susan Rothstein, by me and others)

1. Count nouns and mass nouns

	singular	*plural*	*Counting*	
Count Nouns:	girl	girls	one girl	three girls
	rabbit	rabbits	one rabbit	three rabbits
Mass Nouns:	mud	#muds	#one mud	#three mud(s)
	furniture	#furnitures	#one furniture	#three furniture(s)
Ambiguous:	hair	My hair is getting grey		
		My hairs are getting grey		

2. Bare nouns

Bare nouns are nouns that occur in argument position without determiners.

ENGLISH

English has Bare Plural Count nouns:

(1) a. ***Dogs*** were running around.
b. ***Dogs*** play with each other when they are cheerful
c. There were ***dogs*** running around all afternoon.
d. I am afraid of ***dogs*.**

English has Bare Mass nouns:

(2) a. ***Mud*** was thrown at the prime minister.
b. There was ***mud*** on my shoe.
c. You shouldn't eat ***mud***.

.
English does not have Bare Singular Count nouns:

(3) a. #***Dog*** was running around.
b. #***Dog*** likes each other.
c. #There was ***dog*** running around all afternoon.
d.#I am afraid of ***dog***.

3. Grinding (down shifting)

(4) a. After the accident with the fan, there was ***rabbit*** all over the wall.
b. When Fred stopped trying to repair the watch, there was ***watch*** all over the table.

David Lewis: Universal grinder.
The meaning can **shift** from count to mass by grinding.

-A shift from the singular count meaning of the noun *rabbit*
to a ground mass reading: *rabbit stuff*
(Downshifting (Landman 2020) is a better term, since the watch in (4b) is not ground.)

In English: grinding is by and large a Last Resort Mechanism (Rothstein 2017)

In (4) we find the bare singular count noun *rabbit*.
English doesn't have bare singular count nouns.
We get a grammar **conflict**.
The conflict can be resolved *in some positions* in English by grinding.
Not in normal argument position, but in the position that the subject is in in *there*-insertion contexts:

Compare (5a) with (5b):

(5) a. When I opened the door of the New York appartment, there were ***cockroaches*** all over the wall.
b. After an emergency action with a towel, there was ***cockroach*** all over the wall.

(5a) does not have a grinding reading, (5b) does have a grinding reading.

This is explained if grinding is indeed a last resort mechanism:
English **does** have bare plurals, so we get a perfectly felicitous plural interpretation in (5a): no conflict, no resolution of a conflict, no grinding.

MANDARIN CHINESE

Mandarin Chinese:
1. No number; no lexical distinction between mass and count nouns.
2. Number expressions cannot directly modify nouns.

#Liăng	*níu*	*#Liăng*	*ròu*
two	cow	two	meat

3. Classifiers are used to mediate this relation.

#sān	rén	#sān	jiŭ
three	man	three	wine
✓sān **gè**	rén	✓sān **píng**	jiŭ
three $Cl_{general}$	man	three bottles of wine”	

Classifiers exist and are normal in many languages:

English: collection classifiers: a *flight of* birds, a *school of* fish
container classifiers: a *bottle of* milk

Classifiers are like packagers (the opposite of grinders)

English: classifiers are derived from lexical nouns: exception *head* in *head of cattle*.

Cassifier languages: classifiers are generally underived and obligatory for counting.

Mandarin **does** distinguish count and mass nouns grammatically at **this** level
(i.e. not at the level of N, but yes at the level of NP).
Also, Mandarin **does** recognize the conceptual distinction between prototypically mass (messy) and propotypically count (objects) at the level of lexical nouns:

The general individual classifier *ge* goes with nouns that are prototypically count, but not with nouns that are prototypically mass:

✓*Liăng ge níu* #*Liăng ge ròu*
two CL cow two CL meat

This means that while there is no lexical mass/count distinction, the mass/count distinction does exist in the language.
So it is not the case that Chinese speakers do not have the conceptual disitinction.

4. Mandarin Chinese allows all nouns as bare nouns.

What about the grinding context?

(6) Qiang-shang dou shi ***gou***
wall-topic all copula ***dog***

- This means: There are dogs all over the wall (doggy wallpaper)
- This does **not** mean: There is dog all over the wall (grinding reading)
- To express the ground reading, you need to use ***gou-rou/dog meat***

Conclusion:
With respect to grinding, Chinese works like English:
- grinding is a last resort device which comes into play in the case of a conflict.
- since in Chinese all nouns occur bare anyway, a plural interpretation is in principle available in (6).
-No grammatical conflict, no grinding.

BRASILIAN PORTUGUESE

Brasilian Portuguese has number, and bare plural count nouns (like English), but also allows **bare singular count nouns**.

(7) Eu vi ***criança*** na sala. ✓E ***ela estava*** ouvindo / E ***elas estavam*** ouvindo.
I saw ***child*** in the room. And ***she was*** listening/ And ***they were*** listening

(8) Elefante anda *um atrás do outro*.
Elephant**[sing]** walk**[sing]** one behind of the other.
Elephant**s** walk one after the other.

Elephant *walk after each other* meaning: ***Elephants*** *walk after each other*

In Brasilian Portuguese, what readings you get is dependent on **aspect**:

In the perfective aspect, the facts are the same as in Chinese: no grinding.
In the imperfective aspect, you get an ambiguity between a plural and a grinding reading:

(9) a. Depois do acidente, ***teve*** ***cachorro*** na parede in teira
After the accident ***was***[***perfective***] ***dog*** in the wall whole

Only reading: There were dogs on the wall
No grinder reading

b. Depois do acidente, ***tinha*** ***cachorro*** na parede in teira
After the accident ***was***[***imperfective***] ***dog*** in the wall whole

Ambiguous: There was dog-stuff on the wall/ there were dogs on the wall.

Three languages, three different patterns to do with how bare nouns work in the grammar of these languages.

(Yet different:

YUDJA (Western Amazon Language)
No mass nouns, only count nouns. Countextually available portioning allows counting for all nouns:

(1) Txabiu apeta pe
Three blood dripped Three puddles/spots, etc of blood dripped)

FOODSTUFF NOUNS

English:

(9) a. There is ***apple*** in the salad.
b. There is ***pig*** in the salad Grinding

cf: In the restaurant with three Michelin stars I take a hair out of my soup and say:

(10) Yeagh, there is ***Cordonbleu cook*** in this soup.

Where has the big apple gone that was lying here?
(11) #There is ***big apple*** in the salad

The contrast:

(12) a. There is big dog in the salad (Labrador)
= grinding: there is stuff in the salad derived from *big dog*
b. There is big banana in the salad
= Salad with a whole banana in it
≠ grinding: there is stuff in the salad derived from *big banana*

the big boys the boys are big
the noisy boys the boys are noisy

Why is (11) weird and doesn't it have reading (12b) in analogy to (12a)?

Answer: (Landman and others)
Foodstuff nouns like *apple* are **ambiguous** between a count and a mass interpretation
-***big*** does not modify prototypical mass nouns

Distributive adjectives:

(1) a. The noisy boys – The boys are noisy distributive: individually noisy
collective: noisy as a group
b. The big boys – the boys are big only distributive: individually big

Distrivutive adjectives can modify neat mass nouns, aggregate mass nouns,
but not mess mass nouns, prototypical mass nouns:

(2) a. The big furniture – The furniture is big.
b. #The big mud – The mud is big.

-Specific assumption: The *head* noun of the bare noun (i.e. *apple*) determines whether the grinding strategy is available or not.

So:

(1) The fooodstuff noun *apple* is lexically ambiguous between a count and a mass noun interpretation.
The complex *big apple* is **not** ambiguous in the same way: there is no food stuff: *big apple*.
(2) In the bare noun *big apple*, *apple* is the head, and it allows a mass interpretation, hence no conflict, no grinding (because bare mass nouns **are** allowed in English).
(3) However, *big* can not modify the mass interpretation of *apple*, only the count interpretation.
(4) We get a conflict after all

But, by assumption, we can no longer resolve this conflict by grinding
(because it is the head that determines whether you will grind or not).

In other words:
The last resort nature of grinding predicts that since the head noun *banana/apple* has a felicitous **mass** reading in this context, there is no grinding.
But then *big* must apply to the mass interpretation, which is weird.

Dog does not have a mass interpretation, so *dog* can be ground in this context. If you can grind *dog*, you can grind *big dog*, hence the felicity of (12a).

Does this make sense?
Yes, when we look cross-linguistically.

Mandarin Chinese

(12) a. Shala li you ***pinggui*** foodstuff
Salad inside have ***apple***
b. Shala li you ***zhu*** conceptually count
Salad inside have ***pig***

(12a) is ambiguous:
-There are ***apples*** in the salad
-There is ***apple-stuff*** in the salad
(12b) is **not** ambiguous:
(12b) means: there is ***a whole pig*** in the salad
(12b) does **not** mean: there is ***pig-meat*** in the salad

Foodstuff nouns. Rothstein 2017 calls them *horeca* nouns. Horeca is the dutch acronym for: ***ho****tels,* ***re****staurants,* ***ca****fés.*

(So, *apple* counts as a foodstuff noun in Mandarin, but *pig* does not, even though pigs are, of course, eaten a lot in China. But there is a word for pork, pigmeat. Even if there isn't such a word, say, for dog meat, the fact that dogs are eaten doesn't mean that *dog* is a foodstuff noun. In fact tests like the above tell us which nouns *are* foodstuff nouns in the language.)

Brasilian Portuguese

Change *dog* to *apple*:
Both in the imperfective **and in the perfective** do you get two readings:
-There are ***apples*** in the salad
-There is ***apple-stuff*** in the salad

This suggests indeed that foodstuff nouns are systematically ambiguous between mass and count readings.

Conclusion:
Cross linguistic variation, but systematic ambiguities and systematic connections:

For singular count nouns in English we find two readings:
-a lexical count reading
-a derived ground reading, derivable in contexts of conflict

For foodstuff nouns in English we find two readings:
-a lexical count reading
-a lexical mass reading

And there is reason to think that there are two distinct mass readings:
-lexical mass vs. ground mass

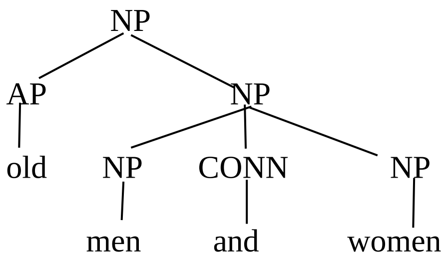
We see that Mandarin Chinese and Brasilian Portuguese in essence bring out the same distinctions, with some differences:
-Mandarin Chinese does not have grinding (well, at least not in the examples studied here), because there isn’t a conflict to be resolved.
-In Brasilian Portuguese, grinding is not a last resort option, but generally available (in the imperfective).

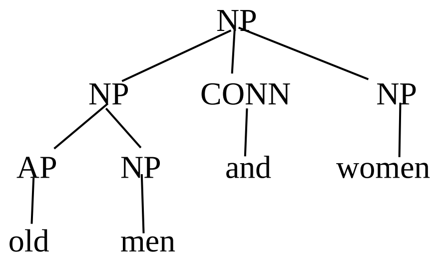
Thus, the ambiguities form regular patterns: there is method in this madness.

2. Syntactic ambiguity.

(1) Old men and women danced. Reading 1 entails: Old women danced.
Reading 2 doesn't entail: Old women danced.

This is an ambiguity of the scope of *old*.
Usual assumption: represented in syntactic constituent structure (at surface structure):





RELATIVE CLAUSES

(2) a. In this opera, the prince is in love with { a girl / the girl / every girl } who doesn't love him

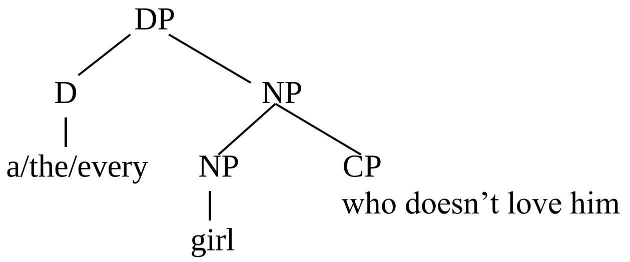
b. In this opera, the prince is in love with { a girl / the girl / #every girl } , who doesn't love him

(2a) is a restrictive relative, (2b) a non-restrictive relative, an appositive.
The data shows a similarity between non-restrictive relatives and discourse anaphora:

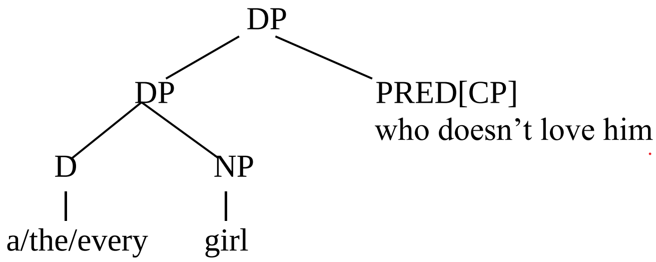
(3) a. In this opera, if { a girl / the girl / every girl } hides in the cupboard, it is because { she / she / #she } doesn't want to meet the prince.

Syntactic ambiguity of the relatives:

Restricted relative:



Non-restricted relative:



The syntactic ambiguity accounts for the discourse anaphora facts:
-in the restrictive relatives there is normal binding
-the non-restrictive relative patterns with discourse anaphora. It adjoines to a full DP, which functions as its discourse anaphora antecedent.

3. Scope ambiguity: quantifiers and negation (English)

(3) Everybody isn't smart.
Reading 1: $\forall x[\neg SMART(x)]$ $\neg$ in the scope of $\forall$
Reading 2: $\neg\forall x[SMART(x)]$ $\forall x$ in the scope of $\neg$

Usual assumption: not represented in syntactic constituent structure (at surface structure).

Alternative approaches:

I. **Movement.** Ambiguity is represented in constituent structure at a different level: Logical Form.
-Build one surface structure.
-Allow scope taking operators to be moved, creating logical forms (For quantifiers, this is in essence what Frege did). This allows two logical representations.
-Interpret these two logical representations.
Theoretical Claim: There is a level of Logical Form ordered **after** the surface syntax: Semantic interpretation takes place after the surface structure is fully derived.`

II. **Storage.** Ambiguity is represented in semantic derivation: the same syntactic constituent structure at surface structure is derived in two different ways:
-the semantic operations for building the meaning of one surface structure for (1) can be applied in two different orders, or, more commonly, combining the interpretation of a scopal expression can be delayed in the derivation, with the interpretation **stored** and **retrieved** at a later stage of the derivation.
This allows for different derivations of the same surface structure with different scopal interpretations.
Theoretical Claim:
You don't need to wait with interpreting till you have derived surface structure, there is no independent level of logical form.

III. **Type shifting**. The effect of storage or movement can be captured by a semantic operation which shifts the normal, minimal interpretation of a scopal expression to an expression of a higher logical type which will give it wide scope.
With the choise of not applying the shift and applying the shift, you derive one surface structure with two interpretations.
Theoretical Claim:
You don't need movement or storage for this.

Evaluating these approaches requires more semantic technique than we have here, they are discussed in more detail in Advanced Semantics.

Much harder to get in other languages (Dutch, Hebrew). Although, I heard someone say the following in the tram in Amsterdam one day:

Elke verandering is geen verbetering
Every change is no improvement = Not every change is an improvement

So it exists, even though it sounds like a translation from English to me.

-Scope ambiguity: multiple quantifiers.

(4) Every man admires a woman. Reading 1: His mother. (or a list…)
Reading 2: Madonna.

$\forall x[MAN(x) \rightarrow \exists y[WOMAN(y) \wedge ADMIRE(x,y)]]$

$\exists y[WOMAN(y) \wedge \forall x[MAN(x) \rightarrow ADMIRE(x,y)]]$

(5) Some man admires every woman.

Inverse reading is a bit harder to get (but try intonation: no stress on *some man* + stress on *every woman*).
But the inverse reading is easy to get in other cases:

(6) A flag hung in front of every window.
A flag spanned every window from left to right

cf. the contrast in (7):

(7) a. At the finish, a bus is waiting for every participant from Tietjerksteradeel.
Preferred reading: $\exists x[BUS(x) \wedge \forall y[P(y) \rightarrow AWAIT(x,y)]]$

b. At the finish, a medal is waiting for every participant from Tietjerksteradeel.
Preferred reading: $\forall y[P(y) \rightarrow \exists x[M(x) \wedge AWAIT(x,y)]]$
Inverse scope: easy to get because *medal* naturally has a **relational** interpretation ('her medal'), and the implicit argument is easily bound by the other quantifier: but that requires inverse scope:
$\forall y[P(y) \rightarrow \exists x[\mathbf{M(x,y)} \wedge WAIT(x,y)]]$

(8) a. In New York City a pedestrian is run over by a car every 3 minutes.
b. In Soviet Russia a tour guide accompanied every foreign visitor.

The readings in scope ambiguities with multiple quantifiers seen so far are not independent: one reading entails the other: i.e. $\exists y\forall x[R(x,y)]$ entails $\forall x\exists y[R(x,y)]$, but not vice versa.

In general, if one reading α entails the other β, you have to take into account the possibility that the grammar generates only the weaker reading β, and derives in context the interpretation as a special case.
This is what we assume for *three cats are smart*: most of us let the grammar generate an *at least* interpretation and see the stronger *exactly interpretation* as a special case: we don't assume an ambiguity.
This strategy has been attempted for scope ambiguities as well (by Tanya Reinhart in the seventies), but not succesfully.
If the readings are logically independent, such a pragmatic strategy will not work.
In that case you either have to argue that the grammar derives a third weaker reading γ that is entailed both by α and by β and treat both α and β as special cases (this has been attempted by Kempson and Cormack in the early eighties, also not succesfully, I think), or you have to accept that there is indeed an ambiguity that the grammar must derive.

4. Collective-distributive ambiguity.

Predicates of individuals: *have blue eyes*:
Distributive interpretation:

(8) a. John and Bill have blue eyes iff John has blue eyes and Bill has blue eyes iff *each of* John and Bill has blue eyes.
b. Three boys have blue eyes iff there is a group of three boys and *each of* those three boys has blue eyes.

Predicates of groups of individuals: *meet in the park*:

In simple cases: collective interpretation:

(9) a. John and Bill met in the park.
does not mean: John met in the park and Bill met in the park.
does not mean: each of John and Bill met in the park.

The intransitive predicate *meet in the park* is not a predicate of individuals.

b. Three boys met in the park.
means: there is a group of three boys and that group met in the park.
does not mean: there is a group of three boys and *each of* those three boys met in the park.

Predicates of individuals or groups of individuals: *carry the piano upstairs*:
Collective/distributive ambiguity:

(10) a. John and Bill carried the piano upstairs.
Reading 1: John and Bill together carried the piano upstairs,
John and Bill carried the piano upstairs as a group. (Collective)

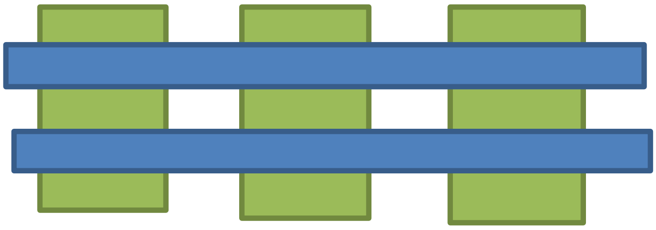
Diagnostics of collective reading: weak involvement of the group members:
the boys carried the piano upstairs allows a boy that doesn't do any carrying but walks in front with a flag.

Reading 2: John carried the piano upstairs and (after that) Bill carried the piano upstairs. (Distributive)
b. Three boys carried the piano upstairs.
Reading 1: There is a group of three boys, and as a group, they carried the piano upstairs. (Collective)
Reading 2: There is a group of three boys, and *each of* those three boys carried the piano upstairs (Distributive).

FACT: For sentences with multiple noun phrases we find **scopal** and **non-scopal** interpretations.

Example:

(11) Two flags hung in front of three windows.



Non-scopal reading: Representation something like the following:

$\exists X[FLAG(X) \wedge |X|=2 \wedge \exists Y[WINDOW(Y) \wedge |Y|=3 \wedge HIFO(X,Y)]]$

$f_1+f_2 \rightarrow w_1+w_2+w_3$
Two flags hung in front of three windows.
We went into town, and saw two flags sown together spanning three windows.

Theories of plurality discuss whether there is one non-scopal reading or several (the question is: do we need to distinguish: group f_1+f_2 spans $w_1+w_2+w_3$ from say: f_1 spans $w_1+w_2+w_3$ **and** f_2 spans $w_1+w_2+w_3$?)

Models for non-scopal readings involve maximally **two** flags and **three** windows.

Cumulative readings (total-total)
20 Chickens laid 140 eggs last week.
20 CH + 140 eggs + every one of these chickens laid some of these eggs
+ every one of these eggs was laid by one of these chickens

These readings are not collective: *laying*, *give birth to* are non-collective relations.)
(argument from Landman 1994, 2000)

Collective:
(1) a. Five women *met* with ten children 5 – 10
b. Ten women *met* with five children 10 – 5
Cumulative:
(2) a. Five women *gave birth to* ten children 5 – 10
b. #Ten women *gave birth to* five children #10 - 5

Why the infelicity of (2b)?
Because *give birth to* does not allow a collective interpretation.
But then the felicitous (2a) is not collective either.
So cumulative readings and collective readings are not the same thing.

Note: I say *infelicity* of (2b), but I am not saying that (2b) is strictly speaking infelicitous. Rather it is uncomfortable. Why? Because it seems to treat *giving birth* as something that can be treated as the responsibility of the whole group of *ten women*. The point is: that is a collectivity effect and often uncomfortable (as group responsibiity often is).

Why do we get this effect in (2b)? Because (2b) cannot have a cumulative reading (because the numbers don't allow a cumulative reading).
Why don't we get this effect in (2a)? Because (2a) **does** allow a cumulative reading. If there were **only** a collective reading, then (2a) should be as uncomfortable as (2b), but it is not. The existence of cumulative readings explains the contrast.

Scopal readings

Every theory needs to distinguish **non-scopal** readings from **scopal** readings, which associate with distributive interpretations.
Models for scopal readings involve a maximum of **two** flags and **six** windows, or **six** flags and **three** windows.

The most natural **scopal** interpretations of (12) are:

Distributive-flag takes scope over collective-window: RECTO SCOPE

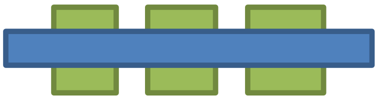
$\exists X[\text{FLAG}(X) \wedge |X|=2 \wedge$ 3 windows per flag

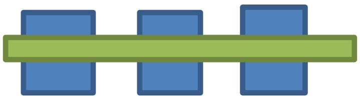
$\quad\quad \boldsymbol{\forall x \in X}$**:** $\exists Y[\text{WINDOW}(Y) \wedge |Y|=3 \wedge \text{HIFO}(x,Y)]]$

$f_1 \rightarrow w_1+w_2+w_3$

$f_2 \rightarrow w_4+w_5+w_6$

Two flags hung in front of three windows:
We found two three-window spanning flags.





Distributive-window takes scope over collective-flag: INVERSE SCOPE

$\exists Y[\text{WINDOW}(Y) \wedge |Y|=3 \wedge$ 2 flags per window

$\quad\quad \boldsymbol{\forall y \in Y}$**:** $\exists X[\text{FLAG}(X) \wedge |X|=2 \wedge \text{HIFO}(X,y)]]$

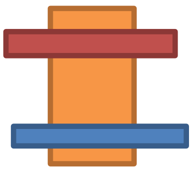
$f_1+f_2 \rightarrow w_1$

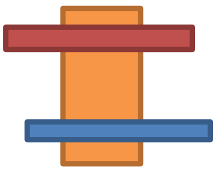
$f_3+f_4 \rightarrow w_2$

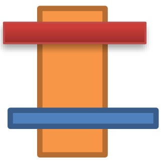
$f_5+f_6 \rightarrow w_3$

Two flags hung in front of three windows.
Of windows with two flags, we found three.

> We've seen many windows with one flag. What about two flags?
> Well, two flags…hm…two flags….let me count….
> Ok, two flags hung in front of three windows.







In this case, the recto-scope reading and the inverse scope reading are logically independent, neither entails the other.

This is evidence that a mechanism for recto and inverse scope must be part of the grammar.

-Scope islands

A medal was given to every girl
A medal *that was given to every girl* was put in the museum.

Unavailable reading: (wide scope of *every girl*)
$\forall x[Girl(x) \rightarrow \exists y[Medal(y) \wedge \exists z[Give(z,y,x) \wedge PIM(y)]]]$
For every girl, there is a medal that was given to her and put in the museum.

Available reading: (narrow scope of *every girl*)
$\exists y[Medal(y) \wedge \forall x[Girl(x) \rightarrow \exists z[Give(z,y,x)]] \wedge PIM(y)]$
Some medal was put in the museum and each girl was given that medal (say, in turn).

Observation:
-*every girl* can take wide scope over *a medal* if the latter is a co-argument of the verbal predicate
-*every girl* cannot take wide scope out of a relative clause over the head of the relative *a medal*.
Relative clauses are scope islands.

5. **De dicto-de re- ambiguity.**
Intensional contexts have scope.
-Modals: *may*

(12) As far as I know, everybody may have done it.
a. $\forall \mathbf{x}[\boldsymbol{may}(DONE(x,it))]$
b. $\boldsymbol{may}(\forall \mathbf{x}[DONE(x,it)])$

Reading a.: Beginning of a detective novel.
Reading b.: Towards the end in a famous detective novel by Agatha Christie.

-Intensional verbs: *try*
(13) John tries to find a unicorn

Representation, something like the following:
a. $\mathbf{TRY}(j,\exists \mathbf{y}[\mathbf{UNICORN(y)} \wedge FIND(j,y)])$ [*de dicto*]
b. $\exists \mathbf{y}[\mathbf{UNICORN(y)} \wedge \mathbf{TRY}(j,FIND(j,y))]$ [*de re*]

The *de dicto* reading does not entail that there **is** a unicorn:
TRY-TO-FIND is not a relation between John and an actual unicorn, but a relation between John and the unicorn-property:
John tries to bring himself in a situation where he has found an instance of the unicorn-property.

Situation 1: you see John with a unicorn-detector searching the beach.
You ask me: what is he doing. I say: (13) John tries to find a unicorn.

(13) is true on the *de dicto* reading, false on the *de re* reading

The *de re* reading **does** entail that there **is** a unicorn:
The sentence expresses that there is an actual unicorn, say, Fido, and John tries to bring himself in a situation where he has found Fido.

Situation 2. We are inside a Harry Potter style novel. Unicorn Fido has escaped. John has always thought that Fido is Tricorn, he is too vain to wear glasses. But we are all looking for Fido. A passerby asks me: what is John doing. I say
(13) John tries to find a unicorn.

(13) is true on the *de re* reading, false on the *de dicto* reading

de dicto/*de re* readings are generally logically independent, although it may require some work to construct models that show that.

-**Propositional attitude verbs**: *know*, *believe*:

(14) John believes that a former soccer player was elected Governer.
a. $\mathbf{BELIEVE}(j, \exists \mathbf{y}[\mathbf{FSP(y)} \wedge EG(y)])$ [*de dicto*]
b. $\exists \mathbf{y}[\mathbf{FSP(y)} \wedge \mathbf{BELIEVE}(j,EG(y))]$ [*de re*]

Reading a:
John reads in the newspaper: "The newly elected governer used to play Rambo."
He thinks Rambo is a soccer team, and he tells me: "A former soccer player got elected governer." I report what he told me to you: I say (14).
(14) John believes that a former soccer player was elected Governer.
I report a belief of John about the property *former soccer player*: in the world according to John, the newly elected governer is a former soccer player.
(14) is true, even though John has no belief **about** any actual individual that that individual got elected governer.

Reading b:
John watched the Governer election, and saw there Arnold getting elected. But he wasn't wearing his glasses, and he thought it was Johan Cruyff. He thinks that Johan Cruyff got elected governer. Not knowing any Dutch, but having seen Johan Cruyff on Dutch television a lot while zapping, he thinks that Johan Cruyff is the Dutch prime minister.
John says to me: "Johan Cruyff got elected governer."
Now, I know very well who Johan Cruyff is, and that he is a famous former soccer player, but I don't know that John doesn't know that, and I **do** know that **you** don't know who Johan Cruyff is. For the latter reason, I report what John said to me to you by saying (14).
(14) John believes that a former soccer player was elected Governer.
In this case, John would not himself accept: "A former soccer player got elected governer." (He would accept: 'The Dutch prime minister got elected governer.").
What I report to you by saying (14) is a belief of John **about** Johan Cruyff, **about** someone who **actually** is a former soccer player.

The situations were chosen in such a way that in the first one the *de dicto* reading is true, but the *de re* reading false, while in the second situation the *de re* reading true, but the *de dicto* reading false. So indeed, the two readings are logically independent (neither entails the other).
This means that if we agree that (14) can be truthfully said in those two types of situations, there is an ambiguity that the grammar must account for.

XVI. GENERALIZED QUANTIFIERS

I Quantifiers don't bind variables.

Frege/Tarski:
Quantifier $\forall x$ or $\exists x$ does **two things simultaneously**:
1. Frege: It binds the occurrences of variable x free in its scope.
   Tarski: It sets up a variation range for the truth value of its scope along the variation of the value for variable x.
2. Frege: It expresses its lexical meaning.
   Tarski: It expresses a constraint according to its lexical meaning on this variation range.

Modern semantic theories for natural language starting in the 1960s with the work of Richard Montague, reported in the posthumously published paper: Montague 1973: 'The proper treatment of quantification in ordinary English.'
Very similar ideas were developed roughly simultaneously in David Lewis' paper 'General Semantics', published in 1970.
The linguistic aspects of this work were strongly influenced by Montague's interaction at the time with Barbara Partee, who, in the years after Montague's death, put Montague style semantics on the map as a field in linguistics, and is pretty much 'the mother of our field'.

Montague-Lewis: Successful compositional semantic analysis of natural language quantification becomes possible only when we realize that for natural language quantification the Frege/Tarski theory is **wrong.**
(Note: Montague and Lewis do not **say** this explicitly, but it follows from their work)

And what is wrong, is part **one** of the Frege/Tarski analysis of quantification:

Montague-Lewis: Natural language quantifiers do not bind variables.
(Montague doesn't **say** this explicitly, but it follows from the theory in Montague 1973. Lewis *is* explicit about this.)

For quantification in natural language, we must separate the setting up of Tarski's variable range from the lexical restriction on the variable range: quantifiers only do the latter.

As it turns out, this separation is linguistically motivated **both** from the perspective of variable binding, **and** from the perspective of quantification.

Variable Binding: quantifiers do not bind variables, because variables are already bound inside the scope of the quantifier.

Some linguistic evidence.

1. Evidence from variables: reflexives.

1a. Reflexives without quantificational binders.

(1) Every boy admires himself.
$\forall x[BOY(x) \rightarrow ADMIRE(x,x)]$
Frege/Tarski: The quantifier $\forall x$ binds the interpretation of the reflexive, the third occurrence of x.

Problems:
-Non-quantificational subjects.

(2) John admires himself.
ADMIRE(j,j)

Intuitively, the interpretation of the reflexive is bound in (2) in the same way as it is in (1). (i.e. we **do** have something of the form ADMIRE(α,α)) in the semantics).
But there is no quantifier in (2) and no variable, and hence no binding operator.

-No subjects.
(3) a. *To admire oneself too much* is regarded as vanity.
b. *Excessive admiration of oneself* is regarded as vanity.

Intuitively, the reflexive is bound in the infinitive and in the noun phrase in the same way as it is in (1) and (2).
(We need, in the semantics, something of the form ADMIRE(α,α)).
But there is no subject, let alone a quantificational subject binding the reflexive.

1b. Reflexives in VP-ellipsis.

VP-ellipsis:

(4) John is smart and Mary **is too**.
be smart ⟶ *be smart*
John is smart and Mary is smart

(5) John kissed Ronya and Mary **did too**.
kiss Ronya ⟶ *kiss Ronya*
John kissed Ronya and Mary kissed Ronya.

(6) John likes himself and Mary **does too**. (sloppy identity reading)
like yourself ⟶ *like yourself*
John likes himself and Mary likes herself.

(6) Every boy likes himself and Every girl **does too**.
like yourself ⟶ *like yourself*
Every boy likes himself and Every girl likes herself.

Think about (6):

$\forall x[BOY(x) \rightarrow \mathbf{LIKE(x,x)}] \wedge \forall y[GIRL(y) \rightarrow ?]$

What we want is a one-place predicate, the interpretation of *like yourself* in which the variable is bound:

$\forall x[BOY(x) \rightarrow$ ***like yourself***$(x)] \wedge \forall y[GIRL(y) \rightarrow ?(y)]$
$\forall x[BOY(x) \rightarrow$ ***like yourself***$(x)] \wedge \forall y[GIRL(y) \rightarrow$ ***like yourself*** $(y)]$

Lambda Notation (to be defined shortly):

$\lambda x.\varphi(x)$
The property that you have if φ is true of you.

like yourself : $\lambda z.LIKE(z,z)$
The property that you have if you like yourself.

Equivalences:

$\forall x[BOY(x) \rightarrow LIKE(x,x)] \wedge \forall y[GIRL(y) \rightarrow ?(y)]$

Equivalent in property form:

$\forall x[BOY(x) \rightarrow \boldsymbol{\lambda z.LIKE(z,z)}\ (x)] \wedge \forall y[GIRL(y) \rightarrow ?(y)]$
For every x if x is a boy then x has the like-yourself property

$\forall x[BOY(x) \rightarrow \boldsymbol{\lambda z.LIKE(z,z)}\ (x)] \wedge \forall y[GIRL(y) \rightarrow \boldsymbol{\lambda z.LIKE(z,z)}\ (y)]$
For every x if x is a boy then x has the like-yourself property and
For every y if y is a girl then y has the like-yourself property.

Every boy likes himself:
$\forall x[BOY(x) \rightarrow LIKE(x,x)]$
$\forall x[\mathbf{BOY}(x) \rightarrow \boldsymbol{\lambda z.LIKE(z,z)}\ (x)]$
every: ∀x[------(x) → --------------(x)]
BOY λz.LIKE(z,z)

every relates two one-place predicates:

$EVERY(BOY, \lambda z.LIKE(z,z))$

But then, the crucial observation is:

The quantifier doesn't bind any variables,
because variables like reflexives are already bound (in the predicate, by the λ-operator).

A footnote on strict readings

Background: pronouns

(1) John likes his mother and Mary does too.

1. Sloppy reading: John likes John's mother and Mary likes Mary's mother.
2. Strict reading: John likes John's mother and Mary likes John's mother.

Our argument is not directly concerned with the interpretation of strict/sloppy identity for pronouns. We *are* interested in reflexives because these need to be bound. We saw:

(2) John likes himself, and Mary does too

1. Sloppy reading:

$\lambda z.LIKE(z,z)(j) \wedge ?(m)$

$\lambda z.LIKE(z,z)(j) \wedge \lambda z.LIKE(z,z)(m)$

$LIKE(j,j) \wedge LIKE(m,m)$

(3) Every boy likes himself, and Every girl does too

1. Sloppy reading:

$\forall x[Boy(x) \rightarrow \lambda z.LIKE(z,z)(x)] \wedge \forall y[GIRL(y) \rightarrow ?(y)]$

$\forall x[Boy(x) \rightarrow \lambda z.LIKE(z,z)(x)] \wedge \forall y[GIRL(y) \rightarrow \lambda z.LIKE(z,z)((y)]$

$\forall x[Boy(x) \rightarrow LIKE(x,x)] \wedge \forall y[GIRL(y) \rightarrow LIKE(y,y)]$

The question is: are there strict readings for reflexives?
We don't expect any such reading for examples like (3), but what about (2):

(2) John likes himself, and Mary does too

2. Strict reading:

$LIKE(j,j) \wedge LIKE(m,j)$

Strict readings are often harder to get, but they *are* possible.
From the literature:
(4) Bill defended himself before John did.

Huge literature in syntax and semantics. But note the following:

John likes himself

A $\lambda z.LIKE(z,z)(j)$ $\Leftrightarrow$

B $LIKE(j,j)$ $\Leftrightarrow$

C $\lambda z.LIKE(x,j)(j)$

The C form is not simply an entailment but is equivalent to A and B.
While we don't assume that what can be reconstructed as the meaning of the elided VP can be *any* entailed property of the subject, the relation between A and C is, of course, much closer. If we assume that, under contextual stress, the equivalence between A, B and C can be used to reconstruct the VP property, we can derive:

John likes himself and Mary does too

$\lambda z.LIKE(x,j)(j) \wedge \lambda z.LIKE(x,j)(m)$

$LIKE(j,j) \wedge LIKE(m,j)$

1c. Pronouns 'bound' by quantifiers that cannot bind them: Functional readings.

As before, we add the definite operator to the logical language:

If $P \in PRED^{1}$, then $\sigma(P) \in TERM$

$$[\![\sigma(P)]\!]_{M,g} = \begin{cases} d & \text{if } [\![P]\!]_{M,g} = \{d\} \\ \text{undefined} & \text{otherwise} \end{cases}$$

(7) The woman that ***every Englishman*** adores most is ***his*** mother.

Meaning of (7):

$\forall x[\ ENGLISHMAN(x) \rightarrow$
For every englishman
$\sigma(\lambda y.WOMAN(y) \wedge ADORE\text{-}MOST(x,y)) \quad = \quad \sigma(\lambda y.\ Mother\text{-}of(y,x\ ))\ ]$
The woman ***he*** *adores most* *is* ***his*** *mother*

To read this formula, read the predicates first:

$\lambda y.WOMAN(y) \wedge ADORE\text{-}MOST(x,y))$
the property you have if you are a woman and x *adores you most*

$\lambda y.\ Mother\text{-}of(y,x\ )$
The property that you have if you are x *'s mother*

σ is the definiteness operator, so:

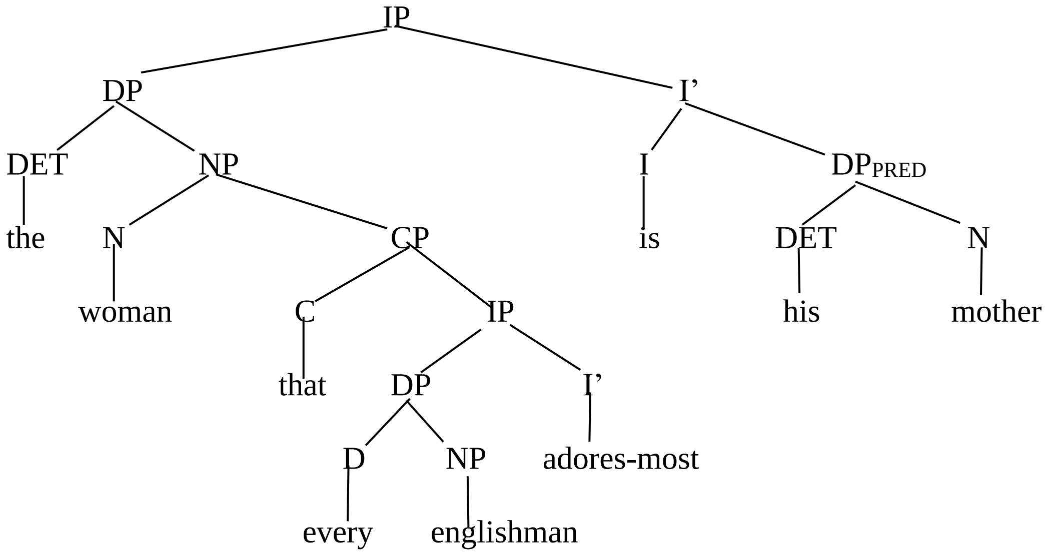
$\sigma(\lambda y.WOMAN(y) \wedge ADORE\text{-}MOST(x,y)))$
The woman that x *adores most*

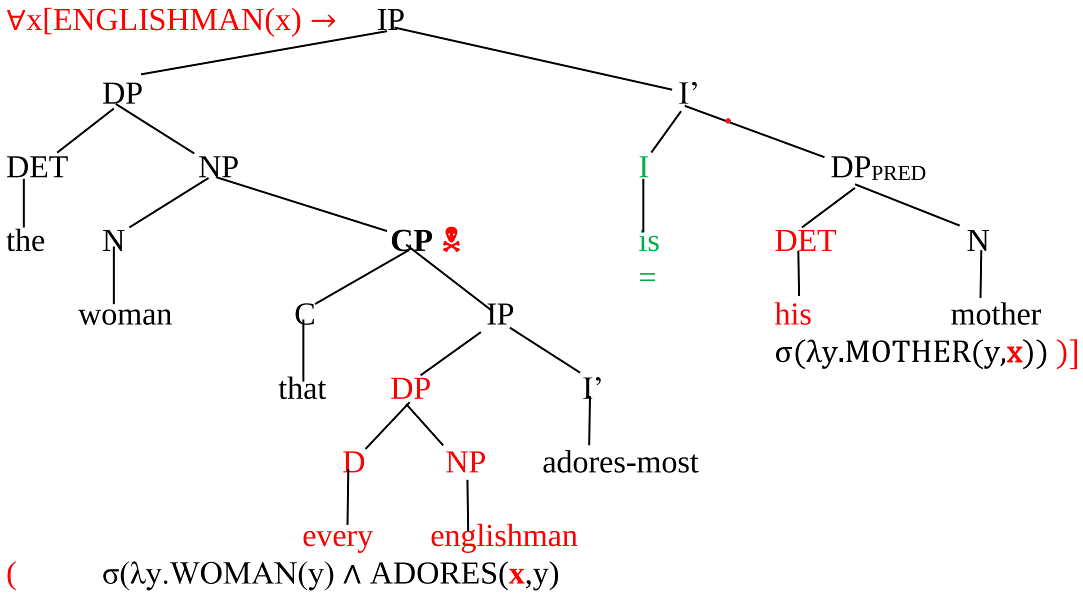
$\sigma(\lambda y.\ Mother\text{-}of(y,x\ ))$
The mother of x

$\sigma(\lambda y.WOMAN(y) \wedge ADORE\text{-}MOST(x,y))) = \sigma(\lambda y.\ Mother\text{-}of(y,x\ ))$
The woman that x *adores most is the mother of* x.

Problem: this involves scoping *every englishman* out of the relative clause ***that every Englishman adores***.
But we have seen that expressions like *every englishman* cannot scope out of relative clauses, since relative clauses are **scope islands**.

(7) The woman that ***every Englishman*** adores most is ***his*** mother.





Problem: in order to bind the variable x in *his mother*, *every englishman* must be given scope out of the relative clause scope island.

Alternative analysis: **Functional readings**.

(7) is analyzed as an equation of two functions f and g, both of which are functions from individuals to individuals:

$f,g: D_M \rightarrow D_M$

The woman that every Englishman adores most
Interpretation: The function f that maps every Englishman onto the woman that he adores most.

his mother
Interpretation: The function that maps every individual onto his/her mother.

We can represent these readings also with help of the λ-operator. We interpret the expression *his mother* (*one's mother*) as:

$g \quad \lambda x.\ \sigma(\lambda y.MOTHER(y,x))$
We read this as:
the function that maps every individual x onto $\sigma(\lambda y.MOTHER(y,x))$, the mother of x.
$f \quad \lambda x \in ENGLISHMEN: \sigma(\lambda y.WOMAN(y) \wedge ADORE(x,y)))$
We read this as:
the function that maps every englishman x onto the woman that x adores most.

(7) The woman that ***every Englishman*** adores most is ***his*** mother.

Semantics: We restrict the mother function to the common domain, englishmen:

when restricted to their common domain:

$g\restriction ENGLISHMEN = \lambda x \in ENGLISHMEN: \sigma(\lambda y.MOTHER(y,x))$
We read this as:
the function that maps every englishman onto his mother.

And (7) is interpreted as (8):

(8) $f = g\restriction ENGLISHMEN$

or explicitly:

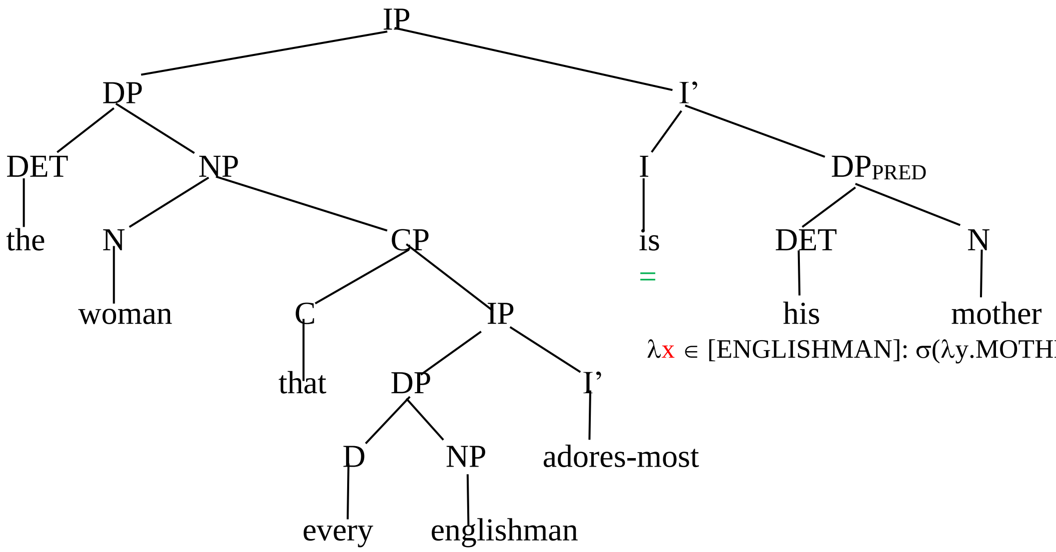
(8) $\lambda x \in ENGLISHMAN: \sigma(\lambda y.WOMAN(y) \wedge ADORE(x,y)))$

$=$

$\lambda x \in ENGLISHMAN: \sigma(\lambda y.MOTHER(y,x))$

(7) then expresses that the function that maps every Englishman onto the woman he adores is the function that maps every Englishman onto his mother.

It turns out that an elegant compositional semantics can be given that derives for *the woman that every Englishman adores* this functional interpretation f, **without** giving *every Englishman* wide scope out of the relative clause.
(7) The woman that ***every Englishman*** adores most is ***his*** mother.



λx ∈ [ENGLISHMAN]: σ(λy.MOTHER(y,x))

λx ∈ ENGLISHMAN: σ(λy.WOMAN(y) ∧ ADORE(x,y)))

An analysis in terms of functional readings along those lines is generally assumed to be the correct way of analyzing cases like (7).

But this means, again, that the pronoun ***his*** in *his mother* in (7) is **not** bound by the quantifier *every Englishman* at all. It is bound inside the expression *his mother*: The interpretation of *his mother* is the function denoted by the expression:

λx. σ(λy.MOTHER(y,x)
The function that maps every x onto x’s mother

and the pronoun ***his*** is **bound** by the λ-operator in this expression.

Which of his relatives does every Englishman admire most?
Groenendijk and Stokhof, Engdahl, ca. 1980

So, by introducing the λ-operator, we can separate quantification and variable binding.

The facts about variables suggest that we should.

2. Evidence from quantification.

Applying the Frege/Tarski's analysis of quantifiers to natural language quantifiers has well known problems.
-There is no good theory of the restricting effect of the noun:

Every cat is smart.
$\forall x[CAT(x) \rightarrow SMART(x)]$
Some cat is smart.
$\exists x[CAT(x) \wedge SMART(x)]$

Sometimes you use $\rightarrow$, sometimes you use $\wedge$. There is no **theory** of when you use the one and when the other.
For $\forall$ and $\exists$, this is not a very serious problem, since we can introduce **restricted quantifiers** (which do not increase the power of the language at all):

If x is a variable and φ a formula, P a one-place predicate, then
$\forall x \in P: \varphi$ and $\exists x \in P: \varphi$ are formulas.

$[\![\forall x \in P: \varphi]\!]_{M,g} = 1$ iff for every $d \in [\![P]\!]_{M,g}$: $[\![\varphi]\!]_{M,g_x^d} = 1$; 0 otherwise
$[\![\exists x \in P: \varphi]\!]_{M,g} = 1$ iff for some $d \in [\![P]\!]_{M,g}$: $[\![\varphi]\!]_{M,g_x^d} = 1$; 0 otherwise

Every cat is smart.
$\forall x[CAT(x) \rightarrow SMART(x)]$
$\forall x \in CAT: SMART(x)$

Some cat is smart.
$\exists x[CAT(x) \wedge SMART(x)]$
$\exists x \in CAT: SMART(x)$

But what about other quantifiers?

Most cats are smart
$Mx[CAT(x)$ **?** $SMART(x)]$
$Mx \in CAT: SMART(x)$

Try: $Mx[CAT(x) \wedge SMART(x)]$
$Mx[CAT(x) \rightarrow SMART(x)]$

You can prove that there is **no** Frege/Tarski quantifier Mx and connective ?
that get the truth conditions of *Most cats are smart* right.
You can prove that there is **no** restricted Frege/Tarski quantifier ***over individuals***
$Mx \in CAT$ that gets the truth conditions of *Most cat are smart* right.

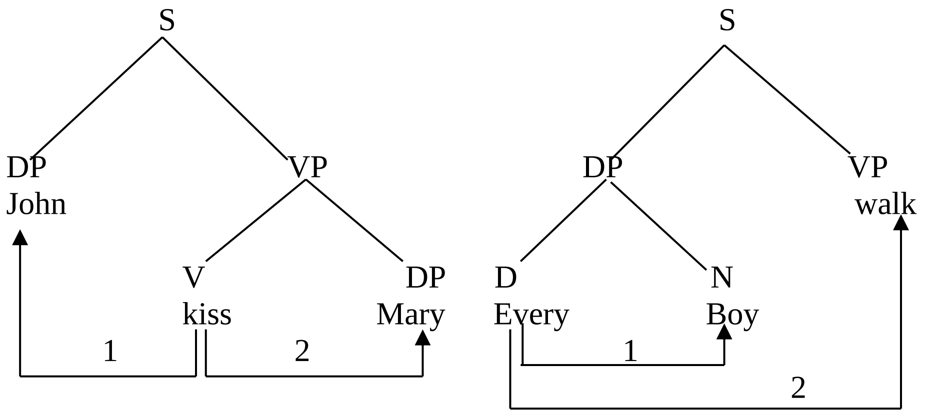
This requires, of course, a proper definition of a 'quantifier over individuals', but it reflects the intuition about the semantics of *most*: *most* **compares** the cardinalities of **two sets of individuals**.

Montague and Lewis solve these problems by using a **different perspective on quantifiers** introduced in logic in the 1950s by Andrej Mostowski, that of **generalized quantifiers**.

I will introduce the theory here as a theory of **generalized quantificational determiners**, by which we mean expressions like *every, some, no, most, at least three*, etc.
The idea is very simple:

> Determiners like *every* do not express Frege/Tarski quantifiers at all, they express relations between **sets of individuals.**

Analogy:



V is a 2 place relation between individuals

D is a 2 place relation between sets of individuals

This idea combines in the following way with the analysis of predicates discussed above. We argued that in *every boy admires himself*, the noun phrase *every boy* or the determiner *every* does not bind the reflexive variable at all, that variable is already bound in the predicate, *admires himself*.
We analyzed that with the variable binding operation λx:

> *admires himself* is interpreted as λx.ADMIRE(x,x).

We are **not** doing **without** the Frege/Tarski analysis of variable binding:
the semantic interpretation of λx.ADMIRE(x,x) is **built**, semantically**,** from Tarski's variable range.

$$\left(\begin{array}{l} <g_x^{d_1}, [\![\text{ADMIRE}(x,x)]\!]_{M,g_x^{d_1}}> \\ <g_x^{d_2}, [\![\text{ADMIRE}(x,x)]\!]_{M,g_x^{d_2}}> \\ <g_x^{d_3}, [\![\text{ADMIRE}(x,x)]\!]_{M,g_x^{d_3}}> \end{array}\right)$$

... for every $d \in D_M$

The variable range is a function from assignments g_x^d, with $d \in D_M$ to truth values. Mathematically, we can identify this with a function from objects $d \in D_M$ to truth values:

$$\begin{pmatrix} \langle d_1, [\![ADMIRE(x,x)]\!]_{M,g_x^{d_1}} \rangle \\ \langle d_2, [\![ADMIRE(x,x)]\!]_{M,g_x^{d_2}} \rangle \\ \langle d_3, [\![ADMIRE(x,x)]\!]_{M,g_x^{d_3}} \rangle \\ \text{... for every } d \in D_M \end{pmatrix}$$

And mathematically, we can identify this with the set characterized by this function:

$$\{d \in D_M: [\![ADMIRE(x,x]\!]_{M,g_x^d} = 1\}$$

But this is **precisely** the interpretation of $\lambda x.ADMIRE(x,x)$.

From this we derive the all important conclusion:

Tarski's value ranges can be identified with sets of individuals.

Now the two theories come together:

-Predication formation on ADMIRE(x) binds variable x to abstraction operator λx. This forms a set of individuals, equivalent to the Tarski value range of ADMIRE(x,x): the set of individuals that admire themselves.

-The determiner meaning *every* in *every boy* expresses a restriction on this set, a restriction which relates it to the set which is the noun interpretation, the set of boys.

In sum, then, we get:

EVERY[BOY,λx.ADMIRE(x,x)]

The semantics of determiner *every* expresses a constraint on the relation between the set of boys and the set of self-admirers.

We have now separated variable binding from quantification:
-variable binding is what Tarski assumed it was, except that it is done by operation λx, and not by quantifiers.
-quantificational determiners express relations between sets of individuals.

The advantage of this perspective for quantificational determiners is that it provides a unified theory of natural language quantification: in this perspective we can study the semantic contribution of any determiner element, and, importantly, we can formulate semantic generalizations about the meanings of classes of determiners.

While developed by Montague and Lewis, the theory was first formulated as a theory of semantic generalizations about classes of determiners by Jon Barwise and Robin Cooper in 1981 in a paper called 'Generalized quantifiers and natural language'.

II THE LANGUAGE L_5: PREDICATE LOGIC EXTENDED WITH GENERALIZED QUANTIFIERS

For comparison reasons, we don't redefine quantification along the lines indicated here, but **add** the new approach to predicate logic.
Our language L_5 has the same syntax as L_4, but with the following additions:

DET = {EVERY, SOME, NO, n, AT MOST n, AT LEAST n, EXACTLY n, MOST} where $n \in \mathbb{N}$ and $n > 0$.
DET $\subseteq$ LEX

Abstraction:
If $x \in$ VAR and $\varphi \in$ FORM, then $\lambda x.\varphi \in \text{PRED}^1$

Quantification:
If $\alpha \in$ DET and P,Q $\in \text{PRED}^1$, then α[P, Q] $\in$ FORM

EXIST:
EXIST $\in \text{PRED}^1$

The semantics for L_5 is exactly the same as for L_4 with the following additions:

For every $\alpha \in$ DET: $[\![\alpha]\!]_{M,g} = F_M(\alpha)$

If $x \in$ VAR and $\varphi \in$ FORM, then:
$[\![\lambda x.\varphi]\!]_{M,g} = \{d \in D_M: [\![\varphi]\!]_{M,g_x^d} = 1\}$

If $\alpha \in$ DET and P,Q $\in \text{PRED}^1$, then:
$[\![\alpha[P, Q]]\!]_{M,g} = 1$ iff $< [\![P]\!]_{M,g}, [\![Q]\!]_{M,g} > \in [\![\alpha]\!]_{M,g}$

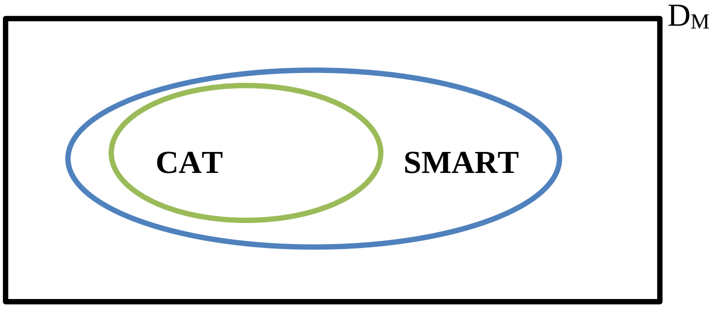
$[\![\text{EXIST}]\!]_{M,g} = D_M$

The existence predicate will be useful in some of the technical discussions below.
This leaves the specification of the new lexical items, the determiners:

For every $\alpha \in$ DET: $F_M(\alpha) \subseteq \text{pow}(D_M) \times \text{pow}(D_M)$
Every determiner is interpreted as a **relation** between **sets** of individuals.

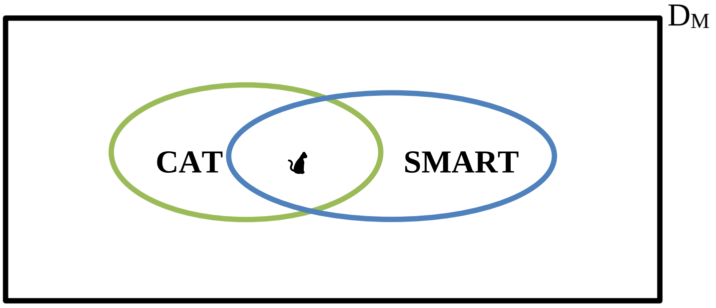
$F_M(\text{EVERY}) = \{\langle X,Y\rangle: X,Y \subseteq D_M \text{ and } X \subseteq Y\}$

EVERY[CAT, SMART]



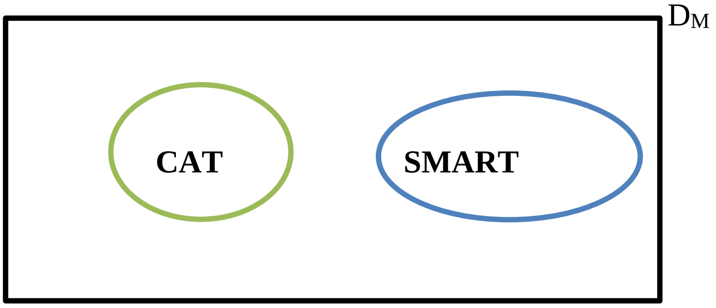
$F_M(\text{SOME}) = \{\langle X,Y\rangle: X,Y \subseteq D_M \text{ and } X \cap Y \neq \emptyset\}$

SOME[CAT, SMART]



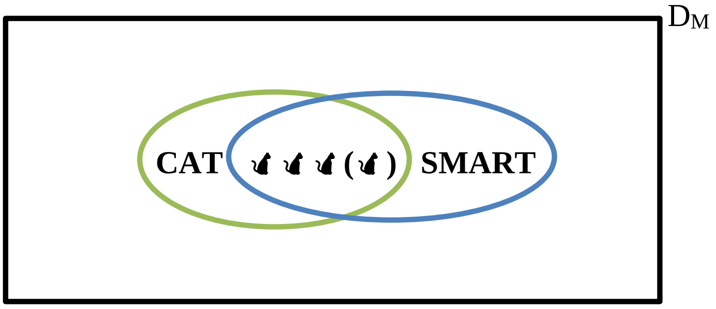
$F_M(\text{NO}) = \{\langle X,Y\rangle: X,Y \subseteq D_M \text{ and } X \cap Y = \emptyset\}$

NO[CAT, SMART]



$F_M(\text{AT LEAST n}) = \{\langle X,Y\rangle: X,Y \subseteq D_M \text{ and } |X \cap Y| \geq n\}$

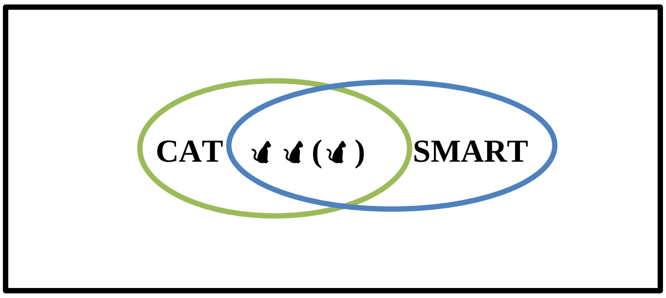
AT LEAST 3[CAT, SMART]



$F_M(\text{AT MOST } n) = \{\langle X,Y\rangle : X,Y \subseteq D_M \text{ and } |X \cap Y| \leq n\}$

AT MOST 3[CAT, SMART]

D_M

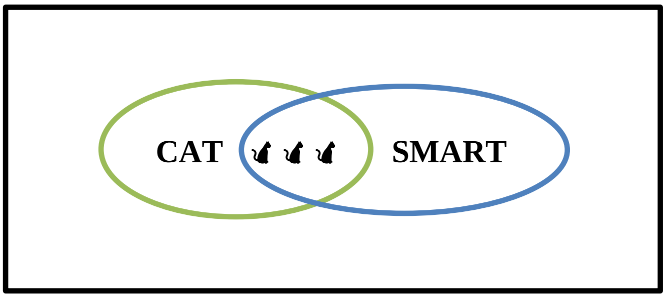


$F_M(n) = F_M(\text{AT LEAST } n)$

$F_M(\text{EXACTLY } n) = \{\langle X,Y\rangle : X,Y \subseteq D_M \text{ and } |X \cap Y| = n\}$

EXACTLY 3[CAT, SMART]

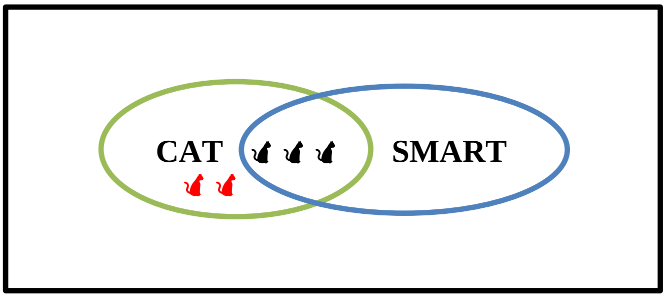
D_M



$F_M(\text{MOST}) = \{\langle X,Y\rangle : X,Y \subseteq D_M \text{ and } |X \cap Y| > |X - Y|\}$

MOST[CAT, SMART]

D_M



We can now prove useful things:

$\text{EVERY}[\text{CAT}, \text{SMART}] \Leftrightarrow \forall x[\text{CAT}(x) \rightarrow \text{SMART}(x)]$

$\text{EVERY}[\text{CAT}, \lambda x.\text{ADMIRE}(x,x)] \Leftrightarrow \forall x[\text{CAT}(x) \rightarrow \text{ADMIRE}(x,x)]$

$\text{SOME}[\text{CAT}, \text{SMART}] \Leftrightarrow \exists x[\text{CAT}(x) \wedge \text{SMART}(x)]$

$\text{NO}[\text{CAT}, \text{SMART}] \Leftrightarrow \neg\exists x[\text{CAT}(x) \wedge \text{SMART}(x)]$

$\text{AT LEAST 2}[\text{CAT}, \text{SMART}] \Leftrightarrow$
$\exists x \exists y[\text{CAT}(x) \wedge \text{CAT}(y) \wedge \text{SMART}(x) \wedge \text{SMART}(y) \wedge (x \neq y)]$
etc.

MOST[CAT, SMART] is not equivalent to any L_4 sentence.

$\text{EVERY}[\text{BOY}, \lambda x.\text{SOME}[\text{GIRL}, \lambda y\text{KISS}(x,y)]] \Leftrightarrow$
$\forall x[\text{BOY}(x) \rightarrow \exists y[\text{GIRL}(y) \wedge \text{KISS}(x,y)]]$

$\text{SOME}[\text{GIRL}, \lambda y.\text{EVERY}[\text{BOY}, \lambda x.\text{KISS}(x,y)]] \Leftrightarrow$
$\exists y[\text{GIRL}(y) \wedge \forall x[\text{BOY}(x) \rightarrow \text{KISS}(x,y)\}\}$

Excursus: A note on *most*
Our semantics:

MOST[A, B]: $|A \cap B| > |A - B|$
most A's are B's is true if there are more A's that are B's than A's that are not B.

An obvious alternative:

MOST[A, B]: $|A \cap B| > ½|A|$
most A's are B's is true if more than half of the A's are B's

Is there a difference? Not on finite domains, obviously.
But do we native speakers have intuitions about infinite domains?
Cantor told us that there are as many even natural numbers as there are natural numbers,
but do we have an **intuition** that (1) below is false (as it is according to our semantics), rather than infelicitous (as it is, if we assume that ½|A| is not defined, if |A| is infinite)?

(1) Most natural numbers are even.

I don't think we do,
but – interestingly enough – we **do** have intuitions about comparison between finite and infinite sets, as in (2):

(2) Most prime numbers are odd.

In (2) we are comparing the cardinality of the set of odd primenumbers (infinite) and the cardinality of the set of even primenumbers (one).
We have no problem counting (2) as true.

This is predicted by our semantics of *most*,
but interestingly enough, not by an analysis that assume that ½|A| is infelicitous if |A| is infinite, or an analysis that assumes that for infinite sets ½|A| = |A| .
Either analysis predicts incorrectly that (2) is infelicitous or false.

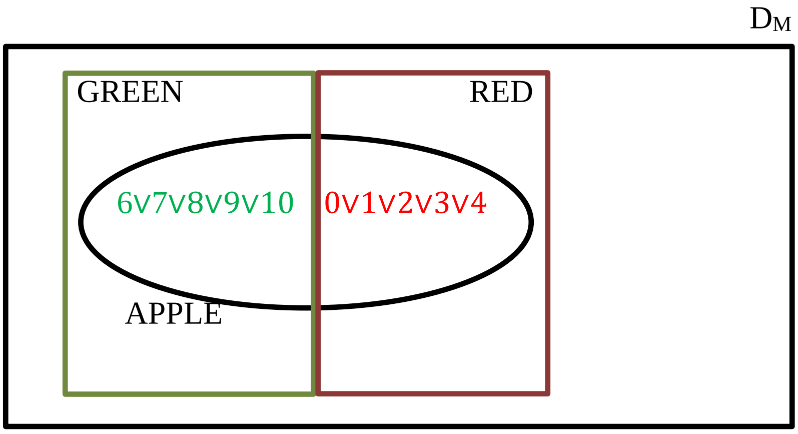
End of excurses

With this new logical language, we can now analyze many new inference patterns, like:

$\{(1),(2),(3)\} \Rightarrow (4)$

(1) There are exactly 10 apples
(2) Every apple is either green or red, not both
(3) Most apples are green
Hence:
(4) At most 4 apples are red

(1) EXACTLY 10[APPLE, EXIST]
(2) EVERY[APPLE, $\lambda x.(\text{GREEN}(x) \vee \text{RED}(x)) \wedge \neg(\text{GREEN}(x) \wedge \text{RED}(x))$]
(3) MOST[APPLE, GREEN]
hence:
(4) AT MOST 4[APPLE, RED]



SKETCH OF THE SEMANTICS FOR PARTIAL DETERMINERS.

Note: the analysis is tailored to later discussion in this chapter. It would be better formulated in a theory that also deals with semantic plurality, but such a theory is only sketched at the end of this class. The present analysis does not treat collective readings at all.

We add to the lexicon a special set of determiners:

DET^p = {THE. BOTH, NEITHER}

We have the same syntactic rule for DET^p as for DET:

If $\alpha \in DET^p$ and $P,Q \in PRED^1$, then $\alpha[P, Q] \in FORM$

We add to the models an interpretation function **pair** $<F_M^+,F_M^->$, where F_M^+ and F_M^- are functions from DET^p to pow(pow(D_M) × pow(D_M), specified below, we call them the *positive extension* and the *negative extension*.

We add the following interpretation rules:

If $\alpha \in DET^p$ and $P,Q \in PRED^1$, then:

$$[\![\alpha[P, Q]]\!]_{M,g} = \begin{cases} 1 & \text{if } <[\![P]\!]_{M,g}, [\![Q]\!]_{M,g}> \in F_M^+(\alpha) \\ 0 & \text{if } <[\![P]\!]_{M,g}, [\![Q]\!]_{M,g}> \in F_M^-(\alpha) \\ \bot \text{ (undefined) otherwise} & \end{cases}$$

Now we specify the lexical meanings of the partial determiners.
In fact, we give here a **schema** for their interpretation:

If $\alpha \in DET^P$ then:

$F_M^+(\alpha) = \{<X,Y>: X,Y \subseteq D_M$ and $\varphi(X,Y)$ **and** $pres_X\}$
$F_M^-(\alpha) = \{<X,Y>: X,Y \subseteq D_M$ and $\neg\varphi(X,Y)$ **and** $pres_X\}$

So, both the positive extension and the negative extension have *the same* presuppositional clause which depends on the noun argument that the determiner combines with. When the presupposition is satisfied, the constraint on $F_M^+(\alpha)$ is that some condition $\varphi(X,Y)$ holds, and on $F_M^-(\alpha)$ that that clause $\varphi(X,Y)$ does not hold.

This means for the truth conditions that:

$[\![\alpha[P, Q]]\!]_{M,g} = 1$ iff $\varphi([\![P]\!]_{M,g},[\![Q]\!]_{M,g})$ and $pres_P$

$[\![\alpha[P, Q]]\!]_{M,g} = 0$ iff $\neg\varphi([\![P]\!]_{M,g},[\![Q]\!]_{M,g})$ and $pres_P$

$[\![\alpha[P, Q]]\!]_{M,g} = \bot$ iff $\neg pres_P$

We start with partial determiner THE:

$F_M^{+}(\text{THE}) = \{<X,Y>: X,Y \subseteq D_M \text{ and } X \subseteq Y \text{ and } pres_X\}$
$F_M^{-}(\text{THE}) = \{<X,Y>: X,Y \subseteq D_M \text{ and } X \not\subseteq Y \text{ and } pres_X\}$

The semantics if THE is the same as that of EVERY: $\varphi(X,Y) = X \subseteq Y$.
As we see in the plural cases, this means that the interpretation we generate is the ***distributive*** interpretation.

The presupposition of the partial determiner THE depends on the interpretation of the noun. To analyse these noun interpretations properly, we need a theory of plurality, which, as said, I am not giving here. But the idea is quite simple:

*the **cat*** **pres**$_{\text{CAT}}$ $= |\text{CAT}| = 1$

THE[CAT, SMART] is true if every cat is smart and there is exactly one cat.
THE[CAT, SMART] is false if not every cat is smart and there is exactly one cat.
(meaning: that cat isn't smart)
THE[CAT, SMART] is undefined if there isn't exactly one cat.

*the **two cats*** **pres**$_{\text{TWO CATS}}$ $= |\text{CAT}| = 2$

THE[TWO CATS, SMART] is true if every cat is smart and there are exactly 2 cats.
THE[TWO CATS, SMART] is false if not every cat is smart and there are exactly 2 cats.
THE[TWO CATS, SMART] is undefined if there aren't exactly 2 cats.

*the **more than two cats*** **pres**$_{\text{MORE THAN TWO CATS}}$ $= |\text{CAT}| > 2$

THE[MORE THAN TWO CATS, SMART] is true if
every cat is smart and there are more than 2 cats.
THE[MORE THAN TWO CATS, SMART] is false if
not every cat is smart and there are more than 2 cats.
THE[MORE THAN TWO CATS, SMART] is undefined if there aren't more than 2 cats.

We will see later that in the semantics for plurality these presuppositions fall out of the theory naturally.

$F_M^+(\text{BOTH}) = \{<X,Y>: X,Y \subseteq D_M \text{ and } X \subseteq Y \text{ and } |X|=2\}$
$F_M^-(\text{BOTH}) = \{<X,Y>: X,Y \subseteq D_M \text{ and } X \not\subseteq Y \text{ and } |X|=2\}$

φ and ψ are **strongly equivalent** iff they are true in the same models and false in the same models.

We can show:

BOTH[CATS, SMART] and THE[TWO CATS, SMART] are strongly equivalent
THE[CAT, SMART] and THE[ONE CAT, SMART] are strongly equivalent.

We saw that *both* has the φ(X,Y)-clause of *every*.
Neither has the φ(X,Y)-clause of *no*:

$F_M^+(\text{NEITHER}) = \{<X,Y>: X,Y \subseteq D_M \text{ and } X \cap Y = \emptyset \text{ and } |X|=2\}$
$F_M^-(\text{NEITHER}) = \{<X,Y>: X,Y \subseteq D_M \text{ and } X \cap Y \neq \emptyset \text{ and } |X|=2\}$

NEITHER[CAT, SMART] is true if no cat is smart and there are exactly two cats.
NEITHER[CAT, SMART] is false if some cat is smart and there are exactly two cats.
NEITHER[CAT, SMART] is undefined if there aren't exactly two cats.

FEW AND MANY.
Lots of literature. Here, unsatisfactory analysis that only deals with the simplest cases.

$F_M(\text{FEW}) = \{<X,Y>: X,Y \subseteq D_M \wedge |X \cap Y| < \mathbf{f}_C(X,Y)\}$
$F_M(\text{MANY}) = \{<X,Y>: X,Y \subseteq D_M \wedge |X \cap Y| > \mathbf{m}_C(X,Y)\}$

Here **f** is a **contextual function** that determines, in context a number that **counts as few**. Which number this is is contextually determined, and can depend on X, on Y, on both, or even on a comparison set C distinct from X and Y.
Similarly, **m** is a contextual function that determines, in context, a number that **counts as many**.

Given this semantics,
we expect
FEW[CAT, SMART] to **pattern semantically** in some ways like
AT MOST n[CAT, SMART],
and we expect
MANY[CAT, SMART] to pattern semantically in some ways like
AT LEAST n[CAT, SMART],
and this is the prediction that interests us here.

There is much more to be said and done about the semantics of *few* and *many* (i.e. readings that are harder to fit in). The semantics given here is introduced here mainly for comparison reasons later.

III GENERAL CONSTRAINTS ON DETERMINER INTERPRETATION.
[Jon Barwise, Robin Cooper, Ed Keenan, Johan van Benthem]

With some notorious problematic cases, discussed in the literature (eg. *few*, *many*, *only* as in *only cats are smart*), natural language determiners all satisfy the following principles of **extension**, **conservativity** and **quantity** (van Benthem 1983)

EXTENSION

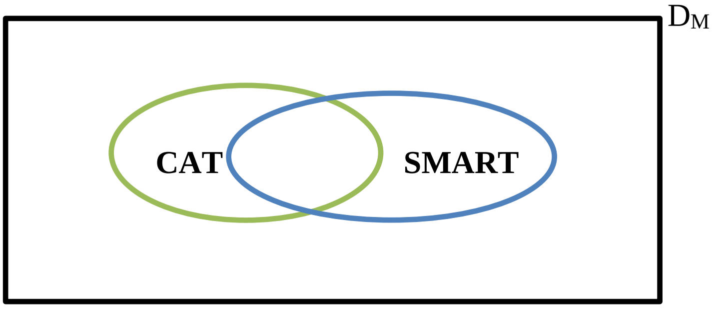
> Determiner α satisfies **extension** iff for all models M_1, M_2 and for all sets X,Y such that $X,Y \subseteq D_{M_1}$ and $X,Y \subseteq D_{M_2}$:
> $<X,Y> \in F_{M_1}(\alpha)$ iff $<X,Y> \in F_{M_2}(\alpha)$
>
> *If you assign* CAT *and* SMART ***the same*** *interpretation in models* M_1 *and* M_2, *then* α[CAT,SMART] *has the same truth value in* M_1 *and* M_2.

Let $F_{M_1}(P) = F_{M_2}(P) = X$ and $F_{M_1}(Q) = F_{M_2}(Q) = Y$.
If α satisfies extension, then the truthvalue of α[P, Q] depends only on what is in $X \cup Y$, not on what is in $D_{M_1} - (X \cup Y)$ or in $D_{M_2} - (X \cup Y)$.

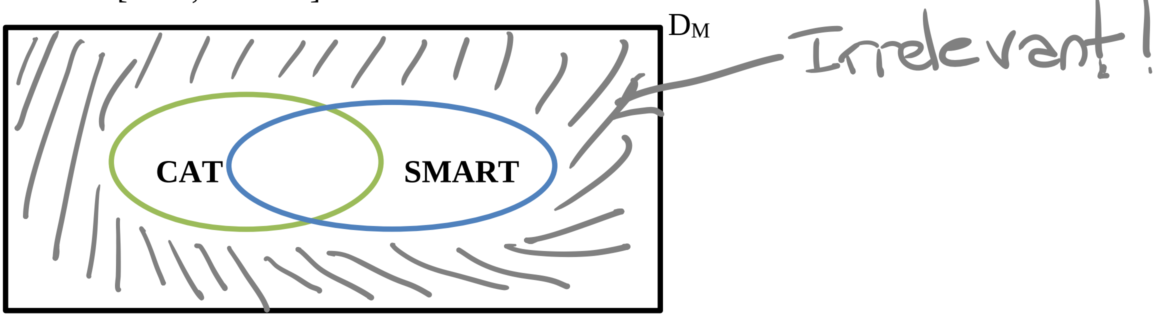
The intuition is the following:
If α satisfies extension then, **if** we **only** specify of a model F_M(CAT) and F_M(SMART), the truth value of α[CAT, SMART] in M is already determined.

This is a natural constraint on natural language determiners:
The truth value of *every cat/some cat/no cat/most cats…is/are smart* does **not** depend on the presence or absence of objects that are neither cats nor smart.

In a picture:



If α satisfies extension then only what is inside CAT ∪ SMART is relevant for the truth of α[CAT,SMART]



So: relevant is:

Extension: if we extend the domain with stupid dogs, the truth value of α[CAT,SMART] is unaffected.

Note: the context dependency of *many* affects judgements for examples like *many cats are smart*. The more stupid dogs you add, the fewer cats you need to say: well, actually many *cats* are smart.

CONSERVATIVITY

Determiner α is **conservative** iff for every model M and for all sets $X,Y \subseteq D_M$:
$<X,Y> \in F_M(\alpha)$ iff $<X, X\cap Y> \in F_M(\alpha)$

(Barwise and Cooper terminology: α is conservative iff in α[X, Y] α **lives on** X)

There is another formulation of conservativity and extension, which is useful:

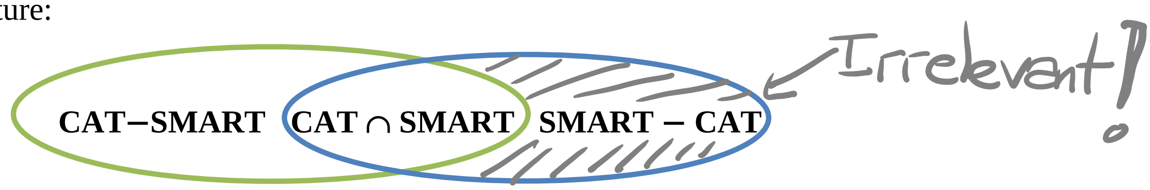
Determiner α satisfies **extension** and **conservativity** iff
for all models M_1,M_2, and all sets X_1,Y_1,X_2,Y_2 such that
$X_1, Y_1 \subseteq D_{M_1}$ and $X_2, Y_2 \subseteq D_{M_2}$:
If $X_1 \cap Y_1 = X_2 \cap Y_2$ and $X_1 - Y_1 = X_2 - Y_2$ then
$<X_1,Y_1> \in F_{M_1}(\alpha)$ iff $<X_2,Y_2> \in F_{M_2}(\alpha)$.

If you let $\lambda x.CAT(x) \wedge SMART(x)$ *and* $\lambda x.CAT(x) \wedge \neg SMART(x)$ *have* ***the same*** *interpretation in* M_1 *and* M_2 *then* α[CAT, SMART] *has the same truth value in* M_1 *and* M_2.

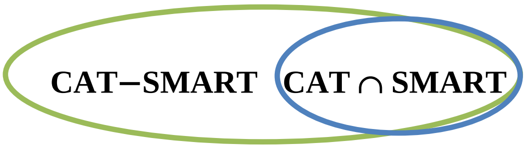
Let $F_{M_1}(P) = X_1$ and $F_{M_2}(P) = Y_1$ and $F_{M_1}(Q) = X_2$ and $F_{M_2}(Q) = Y_2$.
If α satisfies extension, and conservativity, then the truthvalue of α[P, Q] depends only on what is in $X_1 \cap Y_1$ (= $X_2 \cap Y_2$) and in $X_1 - Y_1$ (= $X_2 - Y_2$).

The intuition is the following:
If α satisfies extension and conservativity, then if we specify of a model M, **not even** what $F_M(CAT)$ and $F_M(SMART)$ are, **but only** what $F_M(CAT) \cap F_M(SMART)$ and $F_M(CAT) - F_M(SMART)$ are, then **still** the truth value of α[CAT, SMART] in M is already determined.

This is a natural constraint on natural language determiners:
The truth value of *every cat/some cat/no cat/most cat…is/are smart* does **not** depend on the presence or absence of objects that are neither cats nor smart, **and also not** on the presence or absence of smart cookies that are not cats:
it **only** depends on
what is in the set of cats that are smart,
and
what is in the set of cats that are not smart.
In a picture:



If α satisfies extension and monotonicity, then only CAT ∩ SMART and CAT – SMART are relevant for the truth of α[CAT,SMART]



Conservativity can be checked in the following pattern:

α is conservative iff α[CAT, SMART] is equivalent to
α[CAT, λx.CAT(x) ∧ SMART(x)]

α cat is smart iff α cat is a cat that is smart
α cats are smart iff α cats are cats that are smart
cf:

Every cat is smart iff Every cat is a smart cat
Most cats are smart iff Most cats are smart cats
No cats are smart iff No cats are smart cats

Again problematic are context dependent quantifiers like *many*:
cf. Surprisingly many Swedes are Nobelprize winners. ≠
Surprisingly many Swedes are Swedish Nobelprize winners.

Other classical problem: *only*

Only cats purr ≠
Only cats are purring cats

But only here is probably a DP modifier. *Only the boys* came to the party.

QUANTITY (Independent definition technically complex, see literature)

> Determiner α satisfies **extension** and **conservativity** and **quantity** iff
> for all models M_1, M_2, and all sets X_1, Y_1, X_2, Y_2 such that
> $X_1, Y_1 \subseteq D_{M_1}$ and $X_2, Y_2 \subseteq D_{M_2}$:
> If $|X_1 \cap Y_1| = |X_2 \cap Y_2|$ and $|X_1 - Y_1| = |X_2 - Y_2|$ then
> $\langle X_1, Y_1\rangle \in F_{M_1}(\alpha)$ iff $\langle X_2, Y_2\rangle \in F_{M_2}(\alpha)$.
>
> *If you let each of* $\lambda x.CAT(x) \wedge SMART(x)$ *and* $\lambda x.CAT(x) \wedge \neg SMART(x)$
> *have **the same cardinality** in* M_1 *as it has in* M_2*, then* α[CAT, SMART]
> *has the same truth value in* M_1 *and* M_2.

Let $F_{M_1}(P) = X_1$ and $F_{M_2}(P) = Y_1$ and $F_{M_1}(Q) = X_2$ and $F_{M_2}(Q) = Y_2$.
If α satisfies extension, and conservativity and extension , then the truthvalue of α[P,Q] depends only on **the cardinality of** $X_1 \cap Y_1$ (= $|X_2 \cap Y_2|$) and **the cardinality of** $X_1 - Y_1$ (= $|X_2 - Y_2|$).

The intuition is the following:
If α satisfies extension and conservativity and quantity, then if we specify of a model M, **not even** what F_M(CAT) and F_M(SMART) are, **and not even** what $F_M(CAT) \cap F_M(SMART)$ and $F_M(CAT) - F_M(SMART)$ are, **but only** what $|F_M(CAT) \cap F_M(SMART)|$ and $|F_M(CAT) - F_M(SMART)|$ are
then **still** the truth value of α[CAT,SMART] in M is already determined.

This is a natural constraint on natural language determiners:
The truth value of *every cat/some cat/no cat/most cats…is/are smart* does **not** depend on the presence or absence of objects that are neither cats nor smart, **and also not** on the presence or absence of smart cookies that are not cats; **it doesn't even** depend on **what** is in the set of smart cats, and **what** is in the set of non-smart cats,
but only on
how many things there are in the set of *smart cats*
and on
how many things there are in the set of *non-smart cats*.

In a picture:

where n = |CAT ∩ SMART|
where m = |CAT – SMART|

For determiners that satisfy extension, conservativity and quantity we can set up the semantics in the following more general way.

The independent definition of quantity is formulated in terms of permutations: if you take objects out of CAT ∩ SMART and replace them by **the same number of** other objects, the truth value stays the same, the same for CAT ─ SMART.

With these constraints we are now in a position to characterize the meanings of determiners more narrowly.

We let the model M associate with every determiner α that satisfies extension, conservativity and quantity a **relation r_α between numbers**.
We associate for every model the same relation r_α with α.

In terms of this, we **define** $F_M(\alpha)$ in the following schema for **all natural language determiners that satisfy extension, conservativity and quantity:**
(excluding possessive determiners phrases)

Determiner α is an **ECQ determiner** iff α satisfies extension, conservativity and quantity

Let α be an ECQ determiner.

$$F_M(\alpha) = \{ \langle X,Y\rangle : X,Y \subseteq D_M \text{ and } \langle |X \cap Y|, |X - Y|\rangle \in r_\alpha \}$$

Given this, the meaning of the determiner α is now reduced to the relation r_α between numbers. These meanings we specify as follows:

r_{EVERY}	=	$\{\langle n,0\rangle : n \in \mathbb{N}\}$	
r_{SOME}	=	$\{\langle n,m\rangle : n,m \in \mathbb{N} \text{ and } n \neq 0\}$	
r_{NO}	=	$\{\langle 0,m\rangle : m \in \mathbb{N}\}$	
$r_{AT\ LEAST\ k}$	=	$\{\langle n,m\rangle : n,m \in \mathbb{N} \text{ and } n \geq k\}$	for $k \in \mathbb{N}$
$r_{AT\ MOST\ k}$	=	$\{\langle n,m\rangle : n,m \in \mathbb{N} \text{ and } n \leq k\}$	for $k \in \mathbb{N}$
$r_{EXACTLY\ k}$	=	$\{\langle k,m\rangle : m \in \mathbb{N}\}$	for $k \in \mathbb{N}$
r_{MOST}	=	$\{\langle n,m\rangle : n,m \in \mathbb{N} \text{ and } n > m\}$	

Let $F_M(\textit{cat})$= CAT and $F_M(\textit{smart})$ = SMART, with CAT, SMART $\subseteq D_M$

$[\![\alpha[\textit{cat, smart}]]\!]_{M,g} = 1$ iff $\langle CAT, SMART\rangle \in F_M(\alpha)$ iff
$\langle |CAT \cap SMART|, |CAT - SMART|\rangle \in \mathbf{r}_\alpha$

$[\![\textit{every}\,[\textit{cat, smart}]]\!]_{M,g} = 1$

$<CAT, SMART> \in F_M(\textit{every})$ iff

$<|CAT \cap SMART|, |CAT - SMART|> \in r_{EVERY}$ iff

$<|CAT \cap SMART|, |CAT - SMART|> \in \{<n,0>: n \in \mathbb{N}\}$ iff

$|CAT \cap SMART| \in \mathbb{N}$ and $|CAT - SMART| = 0$ iff

$|CAT - SMART| = 0$ iff

$CAT \subseteq SMART$

$[\![\textit{some}\,[\textit{cat, smart}]]\!]_{M,g} = 1$

$<CAT, SMART> \in F_M(\textit{some})$ iff

$<|CAT \cap SMART|, |CAT - SMART|> \in r_{SOME}$ iff

$<|CAT \cap SMART|, |CAT - SMART|> \in \{<n,m>: n \in \mathbb{N}, m \in \mathbb{N} \text{ and } n \neq 0\}$ iff

$|CAT \cap SMART| \in \mathbb{N}$ $|CAT - SMART| \in \mathbb{N}$ and $|CAT \cap SMART| \neq 0$ iff

$|CAT \cap SMART| \neq 0$ iff

$CAT \cap SMART \neq \emptyset$

$[\![\textit{most}\,[\textit{cat, smart}]]\!]_{M,g} = 1$

$<CAT, SMART> \in F_M(\textit{most})$ iff

$<|CAT \cap SMART|, |CAT - SMART|> \in r_{MOST}$ iff

$<|CAT \cap SMART|, |CAT - SMART|> \in \{<n,m>: n \in \mathbb{N}, m \in \mathbb{N} \text{ and } n > m\}$ iff

$|CAT \cap SMART| > |CAT - SMART|$

$r_{EVERY}(CAT, SMART)$	iff	$\|CAT - SMART\| = 0$	iff	$CAT \subseteq SMART$
$r_{SOME}(CAT, SMART)$	iff	$\|CAT \cap SMART\| \neq 0$	iff	$CAT \cap SMART \neq \emptyset$
$r_{NO}(CAT, SMART)$	iff	$\|CAT \cap SMART\| = 0$	iff	$CAT \cap SMART = \emptyset$
$r_{AT\ LEAST\ k}(CAT, SMART)$	iff	$\|CAT \cap SMART\| \geq k$		
$r_{AT\ MOST\ k}(CAT, SMART)$	iff	$\|CAT \cap SMART\| \leq k$		
$r_{EXACTLY\ k}(CAT, SMART)$	iff	$\|CAT \cap SMART\| = k$		
$r_{MOST}(CAT, SMART)$	iff	$\|CAT \cap SMART\| > \|CAT - SMART\|$		

Some facts about the cardinality of the set of all relations between sets of individuals and the cardinality of the set of all such relations satisfying extension, conservativity and quantity:

$k! = 1 + \ldots + k = \frac{k \times (k+1)}{2}$ (Gauss 100! = 5,050)

$REL_M = pow(pow(D_M) \times pow(D_M))$
REL_M is the set of all relations between sets of individuals in D_M

If $|D_M| = n$
Then $|pow(D_M)| = 2^n$ 2^n distinct properties

Then $|pow(D_M) \times pow(D_M)| = 2^{(2^n)}$

Then $|pow(pow(D_M) \times pow(D_M))| = 2^{(2^{(2^n)})}$

So:

$\lvert D_M\rvert = 1$	$\lvert REL_M\rvert = 16$	distinct relations between sets on a domain of 1 ind.
$\lvert D_M\rvert = 2$	$\lvert REL_M\rvert = 65.536$	2 ind
$\lvert D_M\rvert = 3$	$\lvert REL_M\rvert = 2^{64}$	(Famous from the Chinese chessboard)

So this is the **total number of two place relations** between individuals in a domain of 3 elements

We look at relations satisfying extension, conservativity and quantity.

Let DET_M be the set of all relations in REL_M satisfying extension, conservativity and quantity

If $|D_M| = n$

Then $|DET_M| = 2^{1+ \ldots +n+1} = 2^{\frac{(n+1)(n+2)}{2}}$

So:

$\lvert D_M\rvert = 1$	$\lvert DET_M\rvert = 8$
$\lvert D_M\rvert = 2$	$\lvert DET_M\rvert = 64$
$\lvert D_M\rvert = 3$	$\lvert DET_M\rvert = 1024$

So of the 2^{64} relations, only 1024 relations are candidates for the denotations of natural language determiners.

DETERMINERS AS PATTERNS ON THE TREE OF NUMBERS
(van Benthem 1983)

If |CAT| = 3, then there are four possibilities for the cardinalities in
<|CAT ∩ SMART|, |CAT – SMART |>:
<0,3> means: |CAT ∩ SMART | = 0 and |CAT – SMART | = 3
<1,2> means: |CAT ∩ SMART | = 1 and |CAT – SMART | = 2
<2,1> means: |CAT ∩ SMART | = 2 and |CAT – SMART | = 1
<3,0> means: |CAT ∩ SMART | = 3 and |CAT – SMART | = 0

We can write down a **tree of numbers** which shows for each cardinality of CAT, all the possibilities for the cardinalities of <|CAT ∩ SMART |, |CAT – SMART |>:

<0,0>	|CAT|=0
<0,1> <1,0>	|CAT|=1
<0,2> <1,1> <2,0>	|CAT|=2
<0,3> <1,2> <2,1> <3,0>	|CAT|=3
<0,4> <1,3> <2,2> <3,1> <4,0>	|CAT|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0>	|CAT|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0>	|CAT|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0>	|CAT|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0>	|CAT|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,4> <6,3> <7,2> <8,1> <9,0>	|CAT|=9
...	...

We can now study the **pattern** that each determiner meaning r_α makes on the tree of numbers, by **highlighting** (bold italic) the extension of r_α:

$\mathbf{r_{EVERY}}$

<0,0>	|CAT|=0
<0,1> ***<1,0>***	|CAT|=1
<0,2> <1,1> ***<2,0>***	|CAT|=2
<0,3> <1,2> <2,1> ***<3,0>***	|CAT|=3
<0,4> <1,3> <2,2> <3,1> ***<4,0>***	|CAT|=4
<0,5> <1,4> <2,3> <3,2> <4,1> ***<5,0>***	|CAT|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> ***<6,0>***	|CAT|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> ***<7,0>***	|CAT|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> ***<8,0>***	|CAT|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,4> <6,3> <7,2> <8,1> ***<9,0>***	|CAT|=9
...	...

r_{SOME}

<0,0> |CAT|=0
<0,1> ***<1,0>*** |CAT|=1
<0,2> ***<1,1> <2,0>*** |CAT|=2
<0,3> ***<1,2> <2,1> <3,0>*** |CAT|=3
<0,4> ***<1,3> <2,2> <3,1> <4,0>*** |CAT|=4
<0,5> ***<1,4> <2,3> <3,2> <4,1> <5,0>*** |CAT|=5
<0,6> ***<1,5> <2,4> <3,3> <4,2> <5,1> <6,0>*** |CAT|=6
<0,7> ***<1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0>*** |CAT|=7
<0,8> ***<1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0>*** |CAT|=8
<0,9> ***<1,8> <2,7> <3,6> <4,5> <5,4> <6,3> <7,2> <8,1> <9,0>*** |CAT|=9
…

r_{NO}

<0,0> |CAT|=0
<0,1> <1,0> |CAT|=1
<0,2> <1,1> <2,0> |CAT|=2
<0,3> <1,2> <2,1> <3,0> |CAT|=3
<0,4> <1,3> <2,2> <3,1> <4,0> |CAT|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0> |CAT|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0> |CAT|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0> |CAT|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0> |CAT|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,4> <6,3> <7,2> <8,1> <9,0> |CAT|=9
… …

$r_{AT\ LEAST\ 4}$

<0,0> |CAT|=0
<0,1> <1,0> |CAT|=1
<0,2> <1,1> <2,0> |CAT|=2
<0,3> <1,2> <2,1> <3,0> |CAT|=3
<0,4> <1,3> <2,2> <3,1> ***<4,0>*** |CAT|=4
<0,5> <1,4> <2,3> <3,2> ***<4,1> <5,0>*** |CAT|=5
<0,6> <1,5> <2,4> <3,3> ***<4,2> <5,1> <6,0>*** |CAT|=6
<0,7> <1,6> <2,5> <3,4> ***<4,3> <5,2> <6,1> <7,0>*** |CAT|=7
<0,8> <1,7> <2,6> <3,5> ***<4,4> <5,3> <6,2> <7,1> <8,0>*** |CAT|=8
<0,9> <1,8> <2,7> <3,6> ***<4,5> <5,4> <6,3> <7,2> <8,1> <9,0>*** |CAT|=9
… …

$\mathbf{r}_{\text{AT MOST 4}}$

<0,0> |CAT|=0
<0,1> <1,0> |CAT|=1
<0,2> <1,1> <2,0> |CAT|=2
<0,3> <1,2> <2,1> <3,0> |CAT|=3
<0,4> <1,3> <2,2> <3,1> <4,0> |CAT|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0> |CAT|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0> |CAT|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0> |CAT|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0> |CAT|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,4> <6,3> <7,2> <8,1> <9,0> |CAT|=9
… …

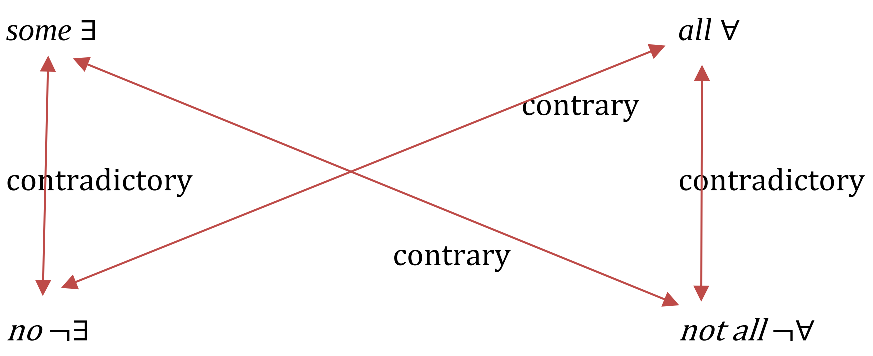
$\mathbf{r}_{\text{EXACTLY 4}}$

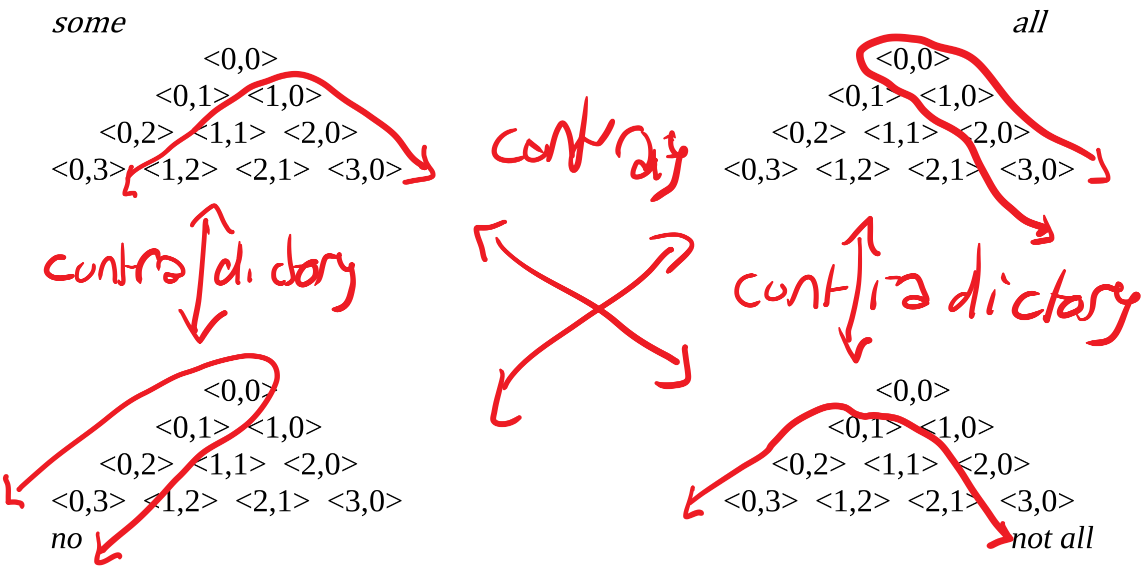
<0,0> |CAT|=0
<0,1> <1,0> |CAT|=1
<0,2> <1,1> <2,0> |CAT|=2
<0,3> <1,2> <2,1> <3,0> |CAT|=3
<0,4> <1,3> <2,2> <3,1> ***<4,0>*** |CAT|=4
<0,5> <1,4> <2,3> <3,2> ***<4,1>*** <5,0> |CAT|=5
<0,6> <1,5> <2,4> <3,3> ***<4,2>*** <5,1> <6,0> |CAT|=6
<0,7> <1,6> <2,5> <3,4> ***<4,3>*** <5,2> <6,1> <7,0> |CAT|=7
<0,8> <1,7> <2,6> <3,5> ***<4,4>*** <5,3> <6,2> <7,1> <8,0> |CAT|=8
<0,9> <1,8> <2,7> <3,6> ***<4,5>*** <5,4> <6,3> <7,2> <8,1> <9,0> |CAT|=9
… …

$\mathbf{r}_{\text{MOST}}$

<0,0> |CAT|=0
<0,1> ***<1,0>*** |CAT|=1
<0,2> <1,1> ***<2,0>*** |CAT|=2
<0,3> <1,2> ***<2,1> <3,0>*** |CAT|=3
<0,4> <1,3> <2,2> ***<3,1> <4,0>*** |CAT|=4
<0,5> <1,4> <2,3> ***<3,2> <4,1> <5,0>*** |CAT|=5
<0,6> <1,5> <2,4> <3,3> ***<4,2> <5,1> <6,0>*** |CAT|=6
<0,7> <1,6> <2,5> <3,4> ***<4,3> <5,2> <6,1> <7,0>*** |CAT|=7
<0,8> <1,7> <2,6> <3,5> <4,4> ***<5,3> <6,2> <7,1> <8,0>*** |CAT|=8
<0,9> <1,8> <2,7> <3,6> <4,5> ***<5,4> <6,3> <7,2> <8,1> <9,0>*** |CAT|=9
… …

Aristotle's square of opppositions





Fact: *some, all, no* are lexicalized in languages as determiners (not necessarily in all)
not all is not lexicalized in any language as a determiner.

However, even though *no* is lexicalized, there is evidence that the ¬ and the ∃ part are **semantically** separable.

Dutch (From Landman 2004):

(1) Wil jij een broodstok? Hm. Dat heet helemaal geen broodstok,
Want you a breadstick? Hm. That is called completely [$_{DP}$ no breadstick]
dat heet soepstengel
that is called soupstel
Do you want a breadstick? Hm. That isn't called *bread stick* at all,
that is called soup stem.

The negation takes auxiliaty scope, the DP semantically breaks up into negation + broodstock

Same in English, though determiner negation is a somewhat formal register.

(2) Seek no evil
Don't seek evil
Evil stays in the scope of the intensional context, but the negation takes auxiliary scope.

Conclusion: even though there is a lexical item *no* and not a lexical item *not all*, the two parts of the lexical item $\neg + \exists$ are semantically separable like the two parts in $\neg + \forall$.

SYMMETRY

Determiner α is **symmetric** iff for every model M and all sets X,Y $\in D_M$:
$<X,Y> \in F_M(\alpha)$ iff $<Y,X> \in F_M(\alpha)$

Pattern: α[CATS, SMART] is equivalent to α[SMART, CATS]

α cat is smart iff α smart cookie is a cat
α cats are smart iff α smart cookies are cats

	SYMMETRIC	
every	NO	Every cat is smart iff every smart cookie is a cat
some	YES	Some cat is smart iff some smart cookie is a cat
no	YES	No cat is smart iff no smart cookie is a cat
at least n	YES	At least three cats are smart iff at least three smart cookies are cats
at most n	YES	At most three cats are smart iff at most three smart cookies are cats
exactly n	YES	Exactly three cats are smart iff exactly three smart cookies are cats
many	YES	Many cats are smart iff many smart cookies are cats (on the analysis given, keeping m constant)
few	YES	Few cats are smart iff few smart cookies are cats (on the analysis given, keeping f constant)
most	NO	Most cats are smart iff most smart cookies are cats
the cat	NO	The cat is smart iff the smart cookie is a cat
the n cats	NO	The two cats are smart iff the smart cookies are two cats
both	NO	Both cats are smart iff both smart cookies are cats
neither	NO	Neither cat is smart iff neither smart cookie is a cat

Felicity in *there*-insertion contexts (Milsark 1974), definiteness effects:

(1) a. #There is *every cat* in the garden.
b. ✓ There is *some cat* in the garden.
c. ✓ There is *no cat* in the garden.
d. ✓There are *at least three cats* in the garden.
e. ✓ There are *at most three cats* in the garden.
f. ✓ There are *exactly three cats* in the garden.
g. ✓ There are *many cats* in the garden.
h. ✓ There are *few cats* in the garden.
i. #There are *most cats* in the garden.
j. #There is *the cat* in the garden.
k. #There are *the cats* in the garden.
l. #There are *the three cats* in the garden.
m. #There are *both cats* in the garden.
n. #There is *neither cat* in the garden.

✓There are *some* zoologists who don't know what a platypus is.
✓There are *not many* zoologists who don't know what a platypus is.
✓There are *no australian* zoologists who don't know what a platypus is.
#There are *all islandic* zoologists who don't know what a platypus is.
#There are *most czech* zoologists who don't know what a platypus is.

The same pattern with relational nouns like *sister* in existential *have* sentences:

John has D sister(s) in the army.

John has a sister in the army/ #John has the sister in the army.
John has at least two sisters in the army?#John has most sisters in the army

Note: exceptions:

(1) a. Who should we ask to sing Auld lang Syne at the party.
Well, there's always Fred.
b. What is there in the fridge? Well, there's the milk and the wine and the cheese.
c. There's every reason to distrust him

(= there is good reason to distrust him. (1c) does not mean:
For every reason to distrust him, there is it.)

Milsark:
[$_{DP}$ [$_D$ α] NOUN] is felicitous in *there*-insertion contexts iff
α is an indefinite determiner

But Milsark doesn't define what an indefinite determiner is.

Observation: Keenan 1987, varying Barwise and Cooper 1981:
(Keenan's actual statement is a bit more subtle, since it applies also to complex noun phrases.)

[$_{DP}$ [$_D$ α] NOUN] is felicitous in *there*-insertion contexts iff α is symmetric.
So Milsark's notion of α is indefinite is defined as: α is symmetric

Determiner α is an **ECQ determiner** iff α satisfies extension, conservativity and quantity.

Let α be an ECQ determiner.

Given conservativity, the semantics of α[X,Y] where α is a symmetric determiner only depends on X∩Y not on X −Y. The reason is that ∩ is commutative, X∩Y = Y∩X, but − is not.

If α is an ECQ determiner, its semantics depends on X∩Y, or X−Y or on both.
If the semantics of α only depends on X∩Y α is symmetric
This follows from commutativity:

$\alpha[X, Y] \Leftrightarrow \varphi_\alpha(|X\cap Y|) \Leftrightarrow \varphi_\alpha(|Y\cap X|) \Leftrightarrow \alpha[Y, X]$.
(with φ_α some numerical property not formulated in terms of X and Y)

In the other two cases there is no such equivalence, because of the non-commutative nature of −, hence you cannot prove symmetry.

This is a relatively informal proof. But it is not difficult to show this more formally.

Let us for ease write α[X, Y] for $<X,Y> \in F_M(\alpha)$.

We defined:

α is *symmetric* iff
for all models M, for all $X, Y \subseteq D_M$: $<X,Y> \in F_M(\alpha) \Leftrightarrow <Y,X> \in F_M(\alpha)$

Keenan 1987 gives another definition of symmetry (but the notion may already be in van Benthem's work):

α is *symmetric* iff
for all models M, for all $X, Y \subseteq D_M$: $<X,Y> \in F_M(\alpha) \Leftrightarrow <X\cap Y,D_M> \in F_M(\alpha)$

It will be visually easier to suppress the quantifiers and write this in the object language form:

The first definition, then, is written as:

Definition 1: α is *symmetric* iff α[A, B] ⇔ α[B, A]

And the second definition is written as:

Definition 2: α is *symmetric* iff α[A, B] ⇔ α[A∩B, EXIST]

where EXIST is the existence predicate with: $[\![EXIST]\!]_{M,g} = D_M$

Theorem: If α is an ECQ determiner then definition 1 and definition 2 are equivalent.
(it is really conservativity that matters)

Proof.

1. The easy side.
Assume that α is symmetric on definition 2.
Then we get the following equivalences:

$$\alpha[A, B] \Leftrightarrow_{[\text{definition 2}]} \alpha[A\cap B,\text{EXIST}] \Leftrightarrow_{[\text{commutativity}]} \alpha[B\cap A,\text{EXIST}] \Leftrightarrow_{[\text{definition 2}]} \alpha[B, A]$$

Hence α is symmetric on definition 1.

2. The harder side (which uses conservativity)
Assume that α is an ECQ determiner and that α is symmetric on definition 1.
Then:

$$\alpha[A, B] \Leftrightarrow_{[\text{conservativity}]} \alpha[A, A\cap B] \Leftrightarrow_{[\text{symmetry def 1}]} \alpha[A\cap B, A] \Leftrightarrow_{[\text{conservativity}]} \alpha[A\cap B, (A\cap B)\cap A] \Leftrightarrow \alpha[A\cap B, A\cap B]$$

$$\alpha[A\cap B, A\cap B] \Leftrightarrow_{[\text{determiner schema}]} r_\alpha(|(A\cap B) \cap (A\cap B)|, |(A\cap B) - (A\cap B)|) \Leftrightarrow r_\alpha(|A\cap B|,0) \Leftrightarrow r_\alpha(|(A \cap B) \cap \text{EXIST}|, |(A \cap B) - \text{EXIST}|)$$

This is because $(A \cap B) \cap \text{EXIST} = (A \cap B)$
and $(A \cap B) - \text{EXIST} = \varnothing$

But:

$$r_\alpha(|(A \cap B) \cap \text{EXIST}|, |(A \cap B) - \text{EXIST}|) \Leftrightarrow \alpha[A\cap B, \text{EXIST}]$$

We have only used equivalences, so we have derived:

$$\alpha[A, B] \Leftrightarrow \alpha[A\cap B, \text{EXIST}]$$

Hence α is symmetric on definition 2.
QED

With this we can indeed freely use the second definition of symmetry for natural language determiners:

α is **symmetric** iff $\alpha[A, B] \Leftrightarrow \alpha[A\cap B, \text{EXIST}]$

And this indeed means that the truth conditions of $\alpha[A, B]$ **only** depend on the cardinality of $A\cap B$, ie. are completely determined by that.

We don't use Milsark's notion of indefineness, rather the terms used in the GQ literature are *weak* and *strong*:

> **Strenght:**
> A determiner α is **weak** iff α is symmetric; otherwise α is **strong**.

Generalization: Weak determiners are determiners α for which the truth value of α[A, B] only depends on |A ∩ B|.

It is then the **commutativity** of ∩ (i.e. the fact that A ∩ B = B ∩ A), which brings in symmetry.
For strong determiners, the semantics of α[A,B] depends not on |A ∩ B| or on more than |A ∩ B|.

Thus, the semantics of EVERY and MOST depends on |A – B|, which makes the determiner antisymmetric (⊆ for every) or asymmetric (> for most).

The semantics of presuppositional noun phrases like *the*, *both*, *neither* have presuppositions that interfere with symmetry: the semantics of α[A, B] associates a presupposition with A, that of α[B, A] associates a presupposition with B. Obviously, this is a failure of symmetry.

The Adjectival Theory of Numericals

Intersective adjectives

ADJ = {OLD, SMART}

If $A \in ADJ$ and $P \in PRED^1_{nominal}$ then $\lambda x.P(x) \wedge A(x) \in PRED^1_{nominal}$

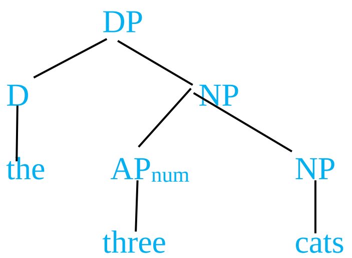
Adjectival Theory: Indefinite determiners derive from adjectival interpretations.

(1) a. The three ferocious tigers
  b. The ferocious three tigers were sent to Artis, and the meek three tigers were sent to Blijdorp.

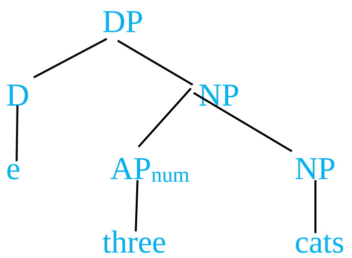
Inside NP, *three* can mingle with adjectives.

Idea: *the three cats* = THE[THREE ∩ CATS]
where THREE ∈ ADJ and CATS ∈ $PRED^1$.
three cats → $\lambda x.CATS(x) \wedge THREE(x)$

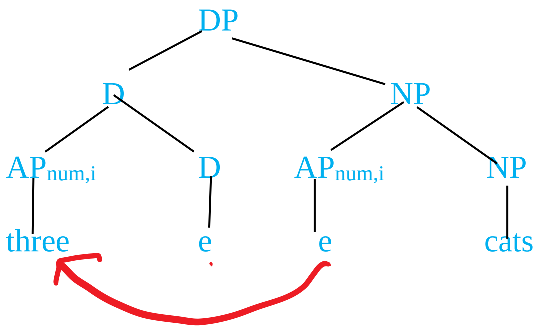
The set of pluralities that are cats and that count as three.
This predicate forms input for the generalized quantifier.



On this idea, in *three cats are smart* we see on the surface the NP *three cats* which is a one place predicate, but we don't see a determiner:



There is actually reason to assume that the numerical actually syntactically occurs in the DP-layer:



But the question is, on either analysis: what is the interpretation of the empty determiner?
The answer (given in Landman 2000) is more subtle than I will give here, and is addressed in detail at the end of Advanced Semantics, when we read my 2000 paper.

But it is generally assumed that we have here a one-place predicate, an NP interpretation, and what we need to get is a DP interpretation.
And the standard assumption is that this is done via a form of **existential closure.**

For the simple case above, we can assume that existential closure takes place by assuming that there is a null determiner $[_{DP}$ e] with determiner interpretation: EC.

As I argue in Landman 2000, defining EC so that it will deal with all cases correctly is rather tricky, but when done correctly, it will turn out to be the case, that in the case of example (1a) EC = SOME:

(1) a. [[Three cats e] are smart]
$EC[\lambda x.CATS(x) \wedge THREE(x), SMART]$
$SOME[\lambda x.CATS(x) \wedge THREE(x), SMART]$

In this analysis, *three* is just a conjoined predicate with *cats* and not a determiner.
The symmetry of the analysis comes from the interpretation of EC,
in fact, EC, as given in Landman 2000 is symmetric in all cases.

This means that on that analysis the symmetry actually does not derive from the numerical, but from the fact that the numerical is conjoined with the noun and the semantics of symmetric EC.

Moral: GQT is not a God Given Theory that gives results that cannot be changed.
It is a framework and a tool for comparing and developing semantic analyses.

A remark about *there*-insertion:
-*there is* is **not** an existential quantifier ∃:
That wouldn't work for examples like (1):

(1) There is no cat in the garden.

This doesn't have a reading with an existential quantifier taking scope over *no cat*.
(?There is something that is no cat, that isn't a cat, that ??????)

-*there is* is **not** a locative.
It looks like a locative in English, and in Dutch, but cross-linguistic variation shows that this is misleading.

The *there*-insertion construction is a construction in which the subject does not occur in the normal external subject position but in some lower position.
The definiteness effects are presumably related to the special properties of that lower position (at least that is what *I* argue in my 2004 book *Indefinites and the Type of Sets*).
Instead what appears in the external subject position is what we call a ***pleonastic* element** (not a great term, because the element may be null).
What appears there is open to variation.
We find the definiteness effects not just with *there be* but also with unaccusative verbs like *arrive*.

English: **There** have just now arrived three girls from Paris.
Pleonastic: ***there***

Dutch: **Er** *zijn* net drie meisjes aangekomen uit Parijs.
Misschien *zijn* (**er**) net drie meisjes aagekomen uit Parijs.

The finite verb (*zijn*) is in second position.
Pleonastic: If the subject is first position, then obligatorily ***er*** [*there*]
If the subject is not first position, it is third position
and either ***er*** or – [null] (i.e. optionally filled with *er*)

German: **Es** *sind* - gerade drei Mädchen angekommen aus Paris.
Vielleicht *sind* - gerade drei Mädchen angekommen aus Paris.
The finite verb (*sind*) is in second position.
Pleonastic: If the subject is first position, then obligatorily ***es*** [*it*]
If the subject is not first position, it is third position and obligatorily – [null]

French: **Il** sont arrivé trois filles de Paris.
Pleonastic: ***Il*** [*he*] are arrived three girls from Paris
Idiomatic: **Il y a** un chat dans le jardin — Pleonastic *il*
he there has a cat in the garden
there is a cat in the garden

MONOTONICITY.

Let α be a determiner.
In α[P, Q] we call P the **first argument** of α and Q the **second argument** of α

Terminology:
α is $\uparrow_1$: α is upward monotonic, upward entailing, on its first argument
α is $\downarrow_1$: α is downward monotonic, downward entailing, on its first argument
α is $-_1$: α is neither upward nor downward monotonic on its first argument

α is $\uparrow_2$: α is upward monotonic, upward entailing, on its second argument
α is $\downarrow_2$: α is downward monotonic, downward entailing, on its second argument
α is $-_2$ α is neither upward nor downward monotonic on its second argument

α is $\boldsymbol{\uparrow_1}$ iff for every model M and all sets $X_1, X_2, Y \subseteq D_M$:

if $\langle \mathbf{X_1}, Y\rangle \in F_M(\alpha)$ and $\mathbf{X_1} \subseteq \mathbf{X_2}$ then $\langle \mathbf{X_2}, Y\rangle \in F_M(\alpha)$

α is $\boldsymbol{\downarrow_1}$ iff for every model M and all sets $X_1, X_2, Y \subseteq D_M$:

if $\langle \mathbf{X_2}, Y\rangle \in F_M(\alpha)$ and $\mathbf{X_1} \subseteq \mathbf{X_2}$ then $\langle \mathbf{X_1}, Y\rangle \in F_M(\alpha)$

α is $\boldsymbol{-_1}$ iff α is not $\uparrow_1$ and α is not $\downarrow_1$

α is $\boldsymbol{\uparrow_2}$ iff for every model M and all sets $X, Y_1, Y_2 \subseteq D_M$:

if $\langle X, \mathbf{Y_1}\rangle \in F_M(\alpha)$ and $\mathbf{Y_1} \subseteq \mathbf{Y_2}$ then $\langle X, \mathbf{Y_2}\rangle \in F_M(\alpha)$

α is $\boldsymbol{\downarrow_2}$ iff for every model M and all sets $X, Y_1, Y_2 \subseteq D_M$:

if $\langle X, \mathbf{Y_2}\rangle \in F_M(\alpha)$ and $\mathbf{Y_1} \subseteq \mathbf{Y_2}$ then $\langle X, \mathbf{Y_1}\rangle \in F_M(\alpha)$

α is $\boldsymbol{-_2}$ iff α is not $\uparrow_2$ and α is not $\downarrow_2$

Diagnostic Tests:
For every model M for English and g: $[\![\text{GINGER CAT}]\!]_{M,g} \subseteq [\![\text{CAT}]\!]_{M,g}$
For every model M for English and g: $[\![\text{WALK}]\!]_{M,g} \subseteq [\![\text{MOVE}]\!]_{M,g}$

α is $\uparrow_1$ iff α[GINGER CAT, WALK] ⇒ α[CAT, WALK]
α is $\downarrow_1$ iff α[CAT, WALK] ⇒ α[GINGER CAT, WALK]
α is $\uparrow_2$ iff α[CAT, WALK] ⇒ α[CAT, MOVE]
α is $\downarrow_2$ iff α[CAT, MOVE] ⇒ α[CAT, WALK]

	ARGUMENT 1	ARGUMENT 2
every	↓	↑
	every ginger cat walks	every cat walks
	every cat walks	every ginger cat walks
	every cat walks	every cat moves
	every cat moves	every cat walks
some	↑	↑
	some ginger cat walks	some cat walks
	some cat walks	some ginger cat walks
	some cat walks	some cat moves
	some cat moves	some cat walks
no	↓	↓
at least n	↑	↑
at most n	↓	↓
exactly n	–	–
most	–	↑
	most ginger cats walk	most cats walk
	most cats walk	most ginger cats walk
	most cats walk	most cats move
	most cats move	most cats walk
many	↑	↑ (on the analysis given)
few	↓	↓ (on the analysis given)

(we ignore the partial determiners here)

Fact: The the determiner interpretations given earlier have exactly this monotonicity behaviour:

Examples:

EVERY

assume:	EVERY[CAT, WALK] $\downarrow_1$	
then:	CAT $\subseteq$ WALK	
then:	GINGER CAT $\subseteq$ WALK	(because GINGER CAT $\subseteq$ CAT)
hence:	EVERY[GINGER CAT, WALK]	
assume:	EVERY[CAT, WALK] $\uparrow_2$	
then:	CAT $\subseteq$ WALK	
then:	CAT $\subseteq$ MOVE	(because WALK $\subseteq$ MOVE)
hence:	EVERY[CAT, MOVE]	

AT MOST THREE

assume: AT MOST THREE[CAT, WALK] $\downarrow_1$

then: $|CAT \cap WALK| \leq 3$

then: $|GINGER\ CAT \cap WALK| \leq 3$ because $GINGER\ CAT \subseteq CAT$

hence: AT MOST THREE[GINGER CAT, WALK]

assume: AT MOST THREE[CAT, MOVE] $\downarrow_2$

then: $|CAT \cap MOVE| \leq 3$

then: $|CAT \cap WALK| \leq 3$ because $WALK \subseteq MOVE$

hence: AT MOST THREE[CAT, WALK]

MOST

$\uparrow_2$

Assume: MOST[CAT, WALK]

Then $|CAT \cap WALK| > |CAT - WALK|$

But: $|CAT \cap MOVE| \geq |CAT \cap WALK|$ because $WALK \subseteq MOVE$

$|CAT - MOVE| \leq |CAT - WALK|$

Hence: $|CAT \cap MOVE| > |CAT - MOVE|$

And hence: MOST[CAT, MOVE]

$-_1$

Assume there are 5 ginger cats and 12 non-ginger cats

Assume 3 ginger cats walk, and 2 ginger cats don't walk

$|GINGER\ CAT \cap WALK| > |GINGER\ CAT - WALK|$

Assume none of the 12 non-ginger cats walk

$|CAT \cap WALK| < |CAT - WALK|$

Then MOST[GINGER CAT, WALK] is true

but MOST[CAT, WALK] is false.

Hence MOST is not $\uparrow_1$

Assume all 12 non-ginger cats walk

$|CAT \cap WALK| > |CAT - WALK|$

And assume 2 of the 5 ginger cats don't walk.

$|GINGER\ CAT \cap WALK| < |GINGER\ CAT - WALK|$

Then MOST[CAT, WALK] is true

but MOST[GINGER CAT, WALK] is false.

Hence MOST is not $\downarrow_1$

Hence MOST is $-_1$

Polarity sensitivity items: *any*, *ever*, *a red cent*, *budge an inch*, *a damn*,…

(1) a. I don't see *anything*
b. #I see *anything*.
(2) a. I haven't ever visited him.
b. #I have *ever* visited him.
(3) a. I don't give *a damn*.
b. #I give *a damn*.
(4) a. He doesn't give *a red cent* to charity.
b. #He gives *a red cent* to charity.
(5) a. The donkey didn't *budge an inch*.
b. #The donkey *budged an inch*.

Polarity sensitivity items are licensed in the scope of negation.
But not just negation, also other contexts:
-Questions: Did you *ever* love me?
-Antecedents of conditionals: If Fred reads *anything* these days, it is Italo Calvino.
-and more…

We use *ever*.
We check: α student(s) *ever* visited Paris — *ever* in the second argument of α
α student(s) who *ever* visited Paris was/were happy — *ever* in the first argument of α

(1) a. #*Every* student ever visited Paris.	2
b. ✓*Every* student who ever visited Paris was happy.	1
(2) a. #*Some* student ever visited Paris.	2
b. #*Some* student who ever visited Paris was happy.	1
(3) a. ✓*No* student ever visited Paris.	2
b. ✓*No* student who ever visited Paris was happy.	1
(4) a. #*At least three* students ever visited Paris.	2
b. #*At least three* students who ever visited Paris were happy.	1
(5) a. ✓*At most three* students ever visited Paris.	2
b. ✓*At most three* students who ever visited Paris were happy.	1
(6) a. #*Exactly three* students ever visited Paris.	2
b. #*Exactly three* students who ever visited Paris were happy.	1
(7) a. #*Most* students ever visited Paris.	2
b.?*Most* students who ever visited Paris were happy.	1
(8) a. #*Many* students ever visited Paris.	2
b. #*Many* students who ever visited Paris were happy.	1
(9) a. ✓*Few* students ever visited Paris.	2
b ✓*Few* students who ever visited Paris were happy.	1

Results: *ever* felicitous inside:

	ARGUMENT 1	ARGUMENT 2
every	YES	NO
some	NO	NO
no	YES	YES
at least n	NO	NO
at most n	YES	YES
exactly n	NO	NO
most	NO(?)	NO
many	NO	NO
few	YES	YES

Correlation: (Ladusaw 1979) Polarity sensitivity item α is felicitous iff α occurs in a downward monotonic environment.

Results: *ever* felicitous inside:

	ARGUMENT 1		ARGUMENT 2	
every	YES	$\downarrow_1$	NO	$\uparrow_2$
some	NO	$\uparrow_1$	NO	$\uparrow_2$
no	YES	$\downarrow_1$	YES	$\downarrow_2$
at least n	NO	$\uparrow_1$	NO	$\uparrow_2$
at most n	YES	$\downarrow_1$	YES	$\downarrow_2$
exactly n	NO	$-_1$	NO	$-_2$
most	NO(?)	$-_1$	NO	$\uparrow_2$
many	NO	$\uparrow_1$	NO	$\uparrow_2$
few	YES	$\downarrow_1$	YES	$\downarrow_2$

What is it about *any* that makes it occur in DE contexts?
(Kadmon and Landman 1993)

Intensifiers:

John is a fool
John is a *damn* fool

1. What does *damn* do?
Answer: it creates a **stronger** expression

2. What does *stronger* mean?
Answer: The expression *damn* entails the expression without *damn*
(Kadmon and Landman allow also pragmatic implication here)

3. How does it create a stronger meaning?
Answer: By being a subsective/intersective adjective
(a damn fool is a fool, but not every fool is a damn fool)
i.e. DAMN FOOL $\subseteq$ FOOL

4. When will it work?
Answer: In upward entailing contexts.

Cf. John isn't a damn fool, he is only a bit of a fool (only metalinguistic negation)
Cf. a. I have always told you Jane, your husband is a DAMN fool.
b.# I have always told you Jane, your husband isn't a DAMN fool.

A damn fool is an indefinite which is stronger and more restricted than *a fool*.

5. How do you intensify in downward entailing contexs?
Answer: By finding an expression that creates a **stronger** expression in downward entailing contexts.
Adjectives **restrict** the noun interpretation: this is **weaker** in DE contexts.
So what we want is an **anti-adjective**: an expression that doesn't **restrict** the noun interpretation but **liberates it, widenes it**.

6. Polarity sensitivity items are anti-adjectives
(This is not a standard term, the term is invented for these class notes. But it is a good term.)

Out of the blue the noun *potatoes* is restricted in context.
I ask: do we have any potatoes? You say: no, we don't have potatoes, you turned them into latkes.

> *We don't have potatoes* = We don't have potatoes^{NARROW}
> potatoes^{NARROW} = potatoes for eating

I say [desparately]: what about the potatoe we used for the game of Mr. Potato Head with the kids yesterday? You say, no Fred, we don't have *any* potatoes.

> *We don't have* ***any*** *potatoes* = We don't have potatoes^{WIDE}
> potatoes^{WIDE} = potatoes for eating or for playing games

> *Any fool* is an indefinite which is **stronger and less restricted** than *a fool*.

But, of course, anti-adjectives only create a stronger expression in DE contexts.

So the restriction on DE contexts can be explained through the interaction of the two properties: widening and strenghtening.

ANOTHER CHARACTERIZATION OF $\uparrow_2$ AND $\downarrow_2$ (van Benthem 1984)

Upward monotonicity on the second argument

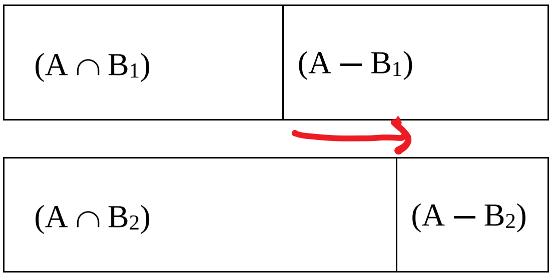
Monotonicity, again, in simpler notation:

α is $\uparrow^2$ iff if $\alpha[A, B_1]$ and $B_1 \subseteq B_2$ then $\alpha[A, B_2]$

van Benthem 1984 shows that there is an equivalent second definition of monotonicity: $\uparrow^{2*}$

α is $\uparrow^{2*}$ iff if $\alpha[A, B_1]$ and $(A \cap B_1) \subseteq (A \cap B_2)$ then $\alpha[A, B_2]$

The condition $(A \cap B_1) \subseteq (A \cap B_2)$ stands for the following situation:



[This can be seen by noting that:
1. For any set B: $(A \cap B) \cup (A - B) = A$
2. If $(A \cap B_1) \subseteq (A \cap B_2)$ then $(A - B_2) \subseteq (A - B_1)$]

On this definition of monotonicity,
α is upward monotonic on the second argument if

moving objects from the difference to the intersection doesn't affect the truth value

And van Benthem proves that these two definitions are equivalent (for ECQ determiners).

Lemma: α if $\uparrow^2$ iff α is $\uparrow^{2*}$

Proof:

(Side 1) Assume α is $\uparrow^{2*}$.
Assume that **$\alpha[A, B_1]$** and **$B_1 \subseteq B_2$**.
If $B_1 \subseteq B_2$ then $(A \cap B_1) \subseteq (A \cap B_2)$.
Then, by definition of $\uparrow^{2*}$, **$\alpha[A, B_2]$**.
Hence, indeed, α is $\uparrow^2$.

(Side 2) Assume α is $\uparrow^2$.
Assume that **$\alpha[A, B_1]$** and **$(A \cap B_1) \subseteq (A \cap B_2)$**.
By conservativity: $\alpha[A, B_1]$ iff $\alpha[A, A \cap B_1]$
Since $(A \cap B_1) \subseteq (A \cap B_2)$, by $\uparrow^2$, $\alpha[A, A \cap B_2]$.
By conservativity, $\alpha[A, A \cap B_2]$ iff $\alpha[A, B_2]$.
Hence **$\alpha[A, B_2]$**
Hence, indeed, α is $\uparrow^{2*}$.

So this version of $\uparrow^{2*}$ says:

> Assume that α[CAT, SMART] is true [situation 1]
> Decide that some non-smart cats are smart after all [situation 2]
> Then α[CAT, SMART] is true in situation 2.

Examples:
1. If ***at least three*** cats are smart, then smartening up non-smart cats **doesn't** affect the truth value. ***at least three*** is $\uparrow^{2*}$.
2. If ***at most three*** cats are smart, then smartening up non-smart cats **can easily** affect the truth value. ***most three*** is not $\uparrow^{2*}$.

Downward monotonicity on the second argument:

α is $\downarrow^{2*}$ iff if $\alpha[A, B_2]$ and $(A \cap B_1) \subseteq (A \cap B_2)$ then $\alpha[A, B_1]$

On this definition of monotonicity,
α is downward monotonic on the second argument if
> **moving objects from the intersection to the difference doesn't affect the truth value**

So this version of $\downarrow^{2*}$ says:

> Assume that α[CAT, SMART] is true [situation 1]
> Decide that some smart cats are not smart after all [situation 2]
> Then α[CAT, SMART] is true in situation 2.

Examples:
1. If ***at most three*** cats are smart, then removing smart cats **doesn't** affect the truth value. ***at most three*** is $\downarrow^{2*}$.
2. If ***at least three*** cats are smart, then removing smart cats **can easily** affect the truth value. ***at least three*** is not $\downarrow^{2*}$.

Lemma: α if $\downarrow^{2}$ iff α is $\downarrow^{2*}$ Proof: mirror image of the above proof.

$-^{2*}$ If *exactly three cats are smart* is true, it doesn't necessarily stay true if you decide that some non-smart cats are smart after all, and neither if you decide that some smart cats are not smart after all.

ANOTHER CHARACTERIZATION OF $\uparrow_1$ AND $\downarrow_1$ (van Benthem 1984)

Only an informal characterization this time:

α is $\uparrow^{1*}$ iff if α[CAT, SMART] is true then α[CAT, SMART] *stays true* if you add cats.
α is $\downarrow^{1*}$ iff if α[CAT, SMART] is true then α[CAT, SMART] *stays true* if you take away cats.

Here adding cats means: either adding them to the intersection or the difference of CAT and SMART or both, and taking away cats means: either taking them away from the intersection or from the difference or from both.

Example:
$\uparrow^{2*}$ If *at least three cats are smart* is true, it stays true if you add more cats to the domain.

$\downarrow^{2*}$ If *at most three cats are smart* is true, it stays true if you take away some cats from the domain.

$-^{2*}$ If *exactly three cats are smart* is true, it doesn't necessarily stay true if you add cats to the domain, and neither if you take away cats from the domain.

MONOTONICITY ON THE TREE OF NUMBERS

Monotonicity on the second argument

These characterisations allow us to define the patterns that monotonicity make on the tree of numbers:

$\uparrow_2^*$ if <n,m> $\in$ $\mathbf{r}_\alpha$, then every number to the right is in α

<0,0>	|CAT|=0
<0,1> <1,0>	|CAT|=1
<0,2> <1,1> <2,0>	|CAT|=2
<0,3> <1,2> <2,1> <3,0>	|CAT|=3
<0,4> <1,3> <2,2> <3,1> <4,0>	|CAT|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0>	|CAT|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0>	|CAT|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0>	|CAT|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0>	|CAT|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> <9,0>	|CAT|=9
...	...

$\downarrow_2^*$ if <n,m) $\in$ $\mathbf{r}_\alpha$ then every number to the left is in α

<0,0>	|CAT|=0
<0,1> <1,0>	|CAT|=1
<0,2> <1,1> <2,0>	|CAT|=2
<0,3> <1,2> <2,1> <3,0>	|CAT|=3
<0,4> <1,3> <2,2> <3,1> <4,0>	|CAT|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0>	|CAT|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0>	|CAT|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0>	|CAT|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0>	|CAT|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,4> <6,3> <7,2> <8,1> <9,0>	|CAT|=9
...	...

We see:

$\mathbf{r}_{\text{EVERY}}$ is $\uparrow_2^*$

<0,0> |CAT|=0

<0,1> ***<1,0>*** |CAT|=1

<0,2> <1,1> ***<2,0>*** |CAT|=2

<0,3> <1,2> <2,1> ***<3,0>*** |CAT|=3

<0,4> <1,3> <2,2> <3,1> ***<4,0>*** → |CAT|=4

<0,5> <1,4> <2,3> <3,2> <4,1> ***<5,0>*** |CAT|=5

<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> ***<6,0>*** |CAT|=6

<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> ***<7,0>*** |CAT|=7

<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> ***<8,0>*** |CAT|=8

<0,9> <1,8> <2,7> <3,6> <4,5> <5,4> <6,3> <7,2> <8,1> ***<9,0>*** |CAT|=9

... ...

$\mathbf{r}_{\text{SOME}}$ is $\uparrow_2^*$

<0,0> |CAT|=0

<0,1> ***<1,0>*** |CAT|=1

<0,2> ***<1,1> <2,0>*** |CAT|=2

<0,3> ***<1,2> <2,1> <3,0>*** |CAT|=3

<0,4> ***<1,3> <2,2> <3,1> <4,0>*** |CAT|=4

<0,5> ***<1,4> <2,3> <3,2> <4,1> <5,0>*** → |CAT|=5

<0,6> ***<1,5> <2,4> <3,3> <4,2> <5,1> <6,0>*** |CAT|=6

<0,7> ***<1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0>*** |CAT|=7

<0,8> ***<1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0>*** |CAT|=8

<0,9> ***<1,8> <2,7> <3,6> <4,5> <5,4> <6,3> <7,2> <8,1> <9,0>*** |CAT|=9

...

$\mathbf{r}_{\text{NO}}$ is $\downarrow_2^*$

<0,0> |CAT|=0

<0,1> <1,0> |CAT|=1

<0,2> <1,1> <2,0> |CAT|=2

← ***<0,3>*** <1,2> <2,1> <3,0> |CAT|=3

<0,4> <1,3> <2,2> <3,1> <4,0> |CAT|=4

<0,5> <1,4> <2,3> <3,2> <4,1> <5,0> |CAT|=5

<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0> |CAT|=6

<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0> |CAT|=7

<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0> |CAT|=8

<0,9> <1,8> <2,7> <3,6> <4,5> <5,4> <6,3> <7,2> <8,1> <9,0> |CAT|=9

... ...

$r_{\text{AT LEAST 4}}$ is $\uparrow_2^*$

<0,0> |CAT|=0
<0,1> <1,0> |CAT|=1
<0,2> <1,1> <2,0> |CAT|=2
<0,3> <1,2> <2,1> <3,0> |CAT|=3
<0,4> <1,3> <2,2> <3,1> ***<4,0>*** |CAT|=4
<0,5> <1,4> <2,3> <3,2> ***<4,1> <5,0>*** |CAT|=5
<0,6> <1,5> <2,4> <3,3> ***<4,2> <5,1> <6,0>*** |CAT|=6
<0,7> <1,6> <2,5> <3,4> ***<4,3> <5,2> <6,1> <7,0>*** |CAT|=7
<0,8> <1,7> <2,6> <3,5> ***<4,4> <5,3> <6,2> <7,1> <8,0>*** |CAT|=8
<0,9> <1,8> <2,7> <3,6> ***<4,5> <5,4> <6,3> <7,2> <8,1> <9,0>*** |CAT|=9
... ...

$r_{\text{AT MOST 4}}$ is $\downarrow_2^*$

<0,0> |CAT|=0
<0,1> <1,0> |CAT|=1
<0,2> <1,1> <2,0> |CAT|=2
<0,3> <1,2> <2,1> <3,0> |CAT|=3
<0,4> <1,3> <2,2> <3,1> <4,0> |CAT|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0> |CAT|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0> |CAT|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0> |CAT|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0> |CAT|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,4> <6,3> <7,2> <8,1> <9,0> |CAT|=9
... ...

$r_{\text{EXACTLY 4}}$ is neither $\uparrow_2^*$ nor $\downarrow_2^*$

<0,0> |CAT|=0
<0,1> <1,0> |CAT|=1
<0,2> <1,1> <2,0> |CAT|=2
<0,3> <1,2> <2,1> <3,0> |CAT|=3
<0,4> <1,3> <2,2> <3,1> ***<4,0>*** |CAT|=4
<0,5> <1,4> <2,3> <3,2> ***<4,1>*** <5,0> |CAT|=5
<0,6> <1,5> <2,4> <3,3> ***<4,2>*** <5,1> <6,0> |CAT|=6
<0,7> <1,6> <2,5> <3,4> ***<4,3>*** <5,2> <6,1> <7,0> |CAT|=7
<0,8> <1,7> <2,6> <3,5> ***<4,4>*** <5,3> <6,2> <7,1> <8,0> |CAT|=8
<0,9> <1,8> <2,7> <3,6> ***<4,5>*** <5,4> <6,3> <7,2> <8,1> <9,0> |CAT|=9
... ...

r_{MOST} is $\uparrow_2^*$

<0,0>	|CAT|=0
<0,1> ***<1,0>***	|CAT|=1
<0,2> <1,1> ***<2,0>***	|CAT|=2
<0,3> <1,2> ***<2,1> <3,0>***	|CAT|=3
<0,4> <1,3> <2,2> ***<3,1> <4,0>***	|CAT|=4
<0,5> <1,4> <2,3> ***<3,2> <4,1> <5,0>***	|CAT|=5
<0,6> <1,5> <2,4> <3,3> ***<4,2> <5,1> <6,0>***	|CAT|=6
<0,7> <1,6> <2,5> <3,4> ***<4,3> <5,2> <6,1> <7,0>***	|CAT|=7
<0,8> <1,7> <2,6> <3,5> <4,4> ***<5,3> <6,2> <7,1> <8,0>***	|CAT|=8
<0,9> <1,8> <2,7> <3,6> <4,5> ***<5,4> <6,3> <7,2> <8,1> <9,0>***	|CAT|=9
…	…

Monotonicity on the first argument

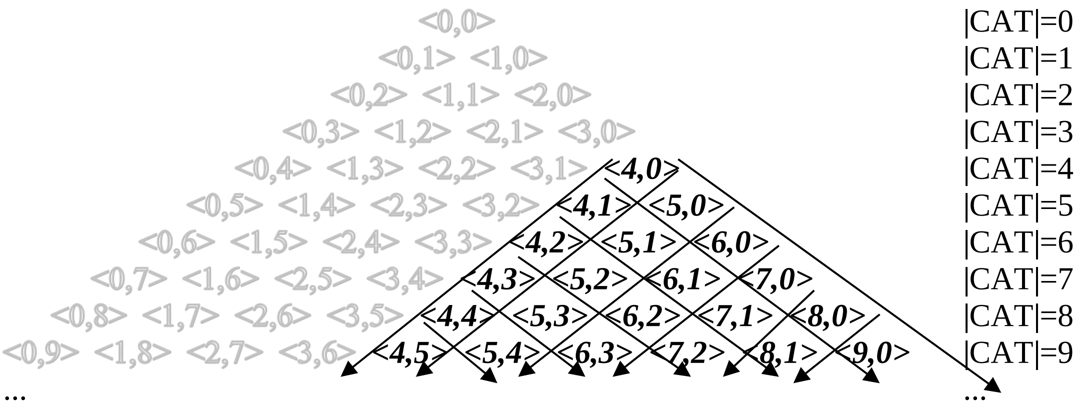
We can give similar definitions on the tree for ↑$_1$ and ↓$_1$
I will here state the facts about the trees:

r_α is ↑$_1$ iff if <n,m> ∈ **r_α** then <n+1,m>, <n,m+1> ∈ **r_α**

This means that **r_α is ↑$_1$** iff if <n,m> ∈ **r_α**
then **the whole triangle with top <n,m>** is in r_α.
Adding a new object to A∪B doesn't affect the truth conditions

Example: $r_{\text{AT MOST 4}}$ is ↑$_1$:

$r_{\text{AT LEAST 4}}$

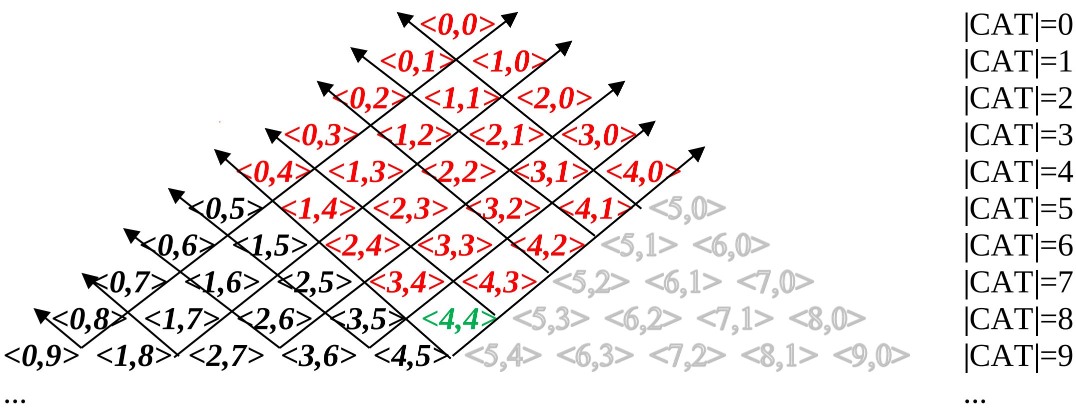


r_α is ↓$_1$ iff if <n,m> ∈ **r_α** then <n−1,m>, <n,m−1> ∈ **r_α**
(when n or m is 0, set n−1, m−1 to 0 as well)

This means that **r_α is ↓$_1$** iff if <n,m> ∈ r_α then **the whole inverted triangle with bottom <n,m>** is in r_α.
Taking away objects from A∪B doesn't affect the truth conditions.

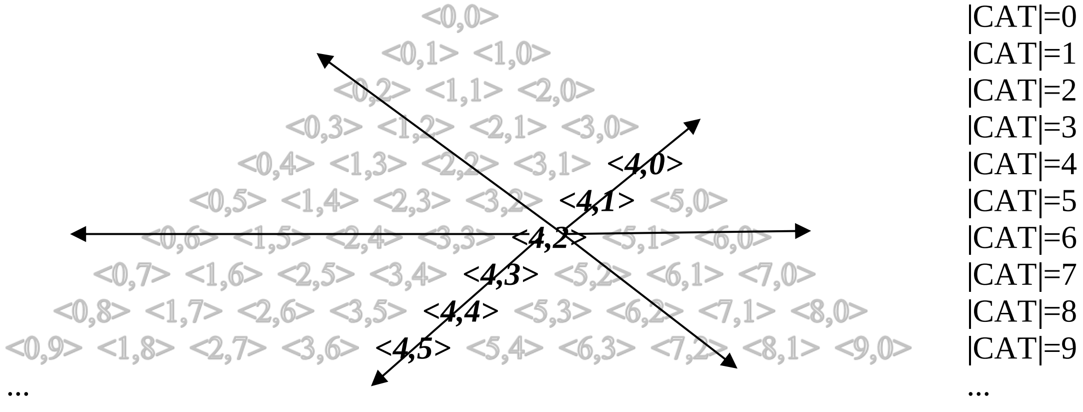
Example: $r_{\text{AT LEAST 4}}$ is ↓$_1$:

$r_{\text{AT MOST 4}}$



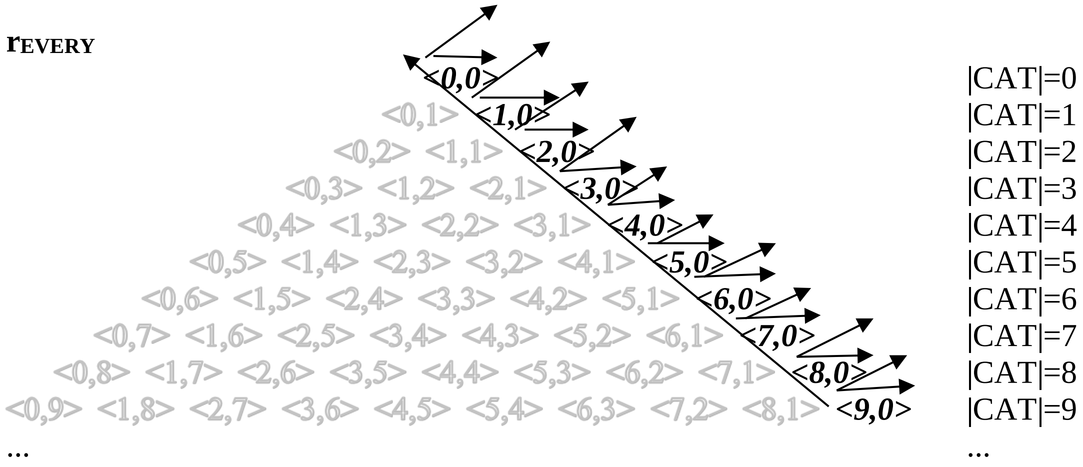
It is easy to check that $r_{\text{EXACTLY 4}}$ is none of the above:

rEXACTLY 4



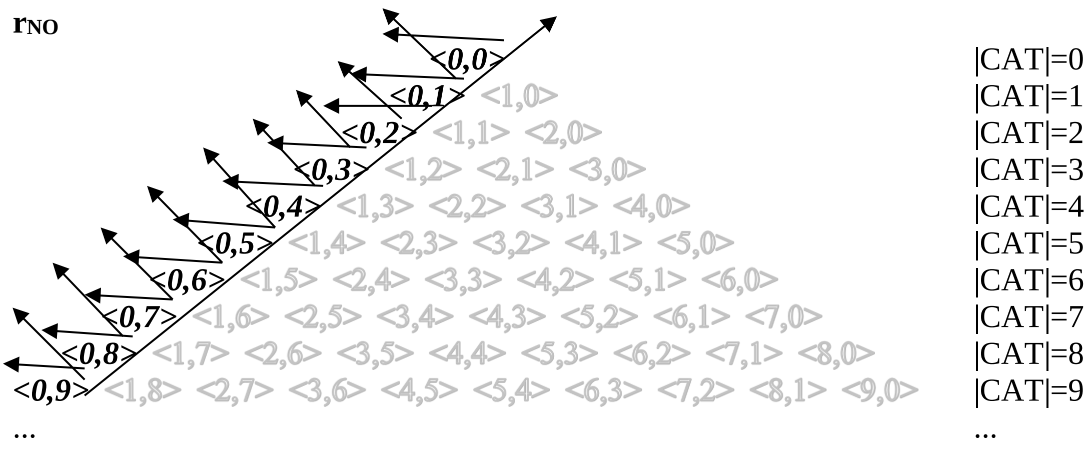
r_{every} is clearly not $\uparrow_1$, since the downward triangles are not preserved.
r_{every} is $\downarrow_1$, since the upward inverted triangle is just the right edge.

rEVERY



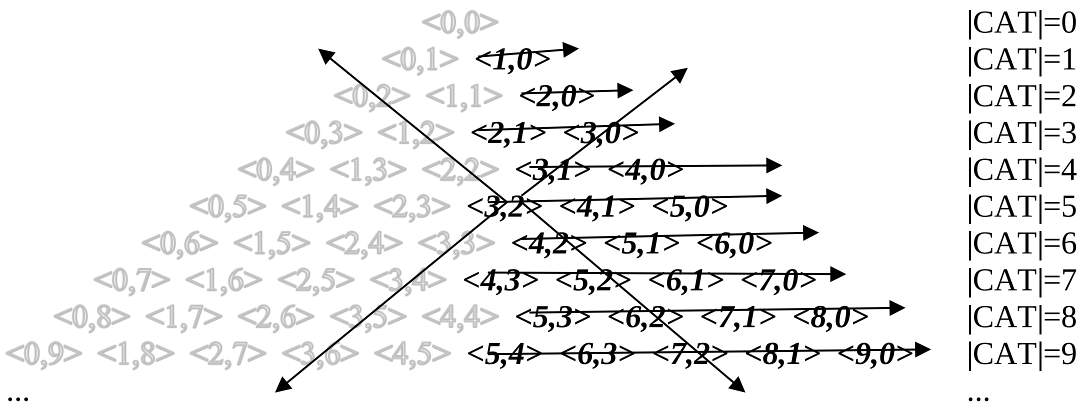
r_{no} is again clearly not $\uparrow_1$, but it is $\downarrow_1$, because, again, the upward inverted triangle is just the left edge.

rNO



r_{most} is $\uparrow_2$, but neither $\uparrow_1$ nor $\downarrow_1$: for no point in r_{most} is the downward triangle completely in r_{most} and for no point is the upward triangle completely in r_{most} (because <0,0> is not).

r_{MOST}



SYMMETRY ON THE TREE OF NUMBERS

We have shown that for symmetric determiners the truth value of α[A, B] only depends on A∩B not on the difference. This means in terms of the numbers <n,m>, that the truthvalue only depends on n, not on m.

We assume that:

α is **symmetric** iff r_α **is symmetric**

What does it mean for $\mathbf{r}_\alpha$ to be symmetric?
It means that the number m varies without affecting the truthvalue of α[A, B]
This means that:

$\mathbf{r}_\alpha$ is **symmetric** iff

If for some number n ∈ ℕ there is a number k ∈ ℕ such that <n,k> ∈ $\mathbf{r}_\alpha$
then for all numbers m ∈ ℕ: <n,m> ∈ $\mathbf{r}_\alpha$
If for some number n ∈ ℕ there is a number k ∈ ℕ such that <n,k> ∉ $\mathbf{r}_\alpha$
then for all numbers m ∈ ℕ: <n,m> ∉ $\mathbf{r}_\alpha$

In terms of the tree of numbers this means the following.
For number n, {<n,k>:k ∈ ℕ} is a **diagonal line** in the tree going from left below to right up:
Like, for n = 3:

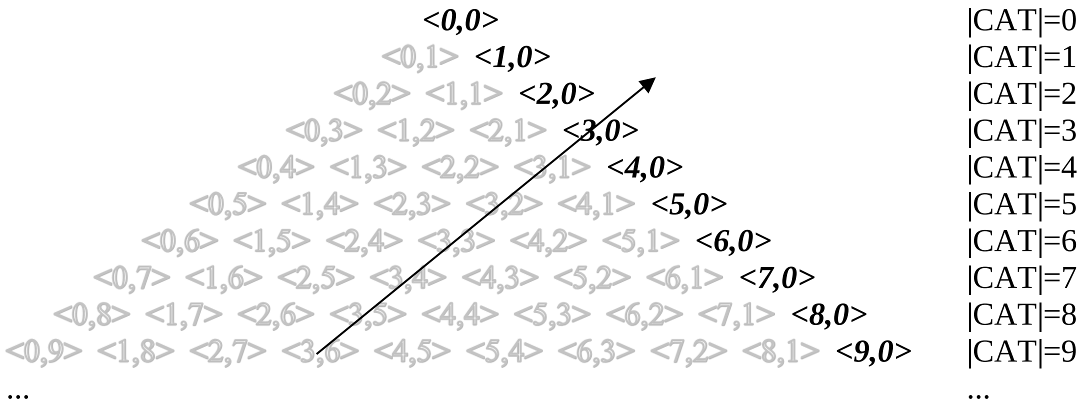
<0,0> — |CAT|=0
<0,1> <1,0> — |CAT|=1
<0,2> <1,1> <2,0> — |CAT|=2
<0,3> <1,2> <2,1> <3,0> — |CAT|=3
<0,4> <1,3> <2,2> <3,1> <4,0> — |CAT|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0> — |CAT|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0> — |CAT|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0> — |CAT|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0> — |CAT|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,4> <6,3> <7,2> <8,1> <9,0> — |CAT|=9
... — ...

r_α is **symmetric** iff every such diagonal line is **either completely inside $\mathbf{r}_\alpha$ or completely outside $\mathbf{r}_\alpha$.**

With this we can check straighforwardly in the trees which r_α's are symmetric:

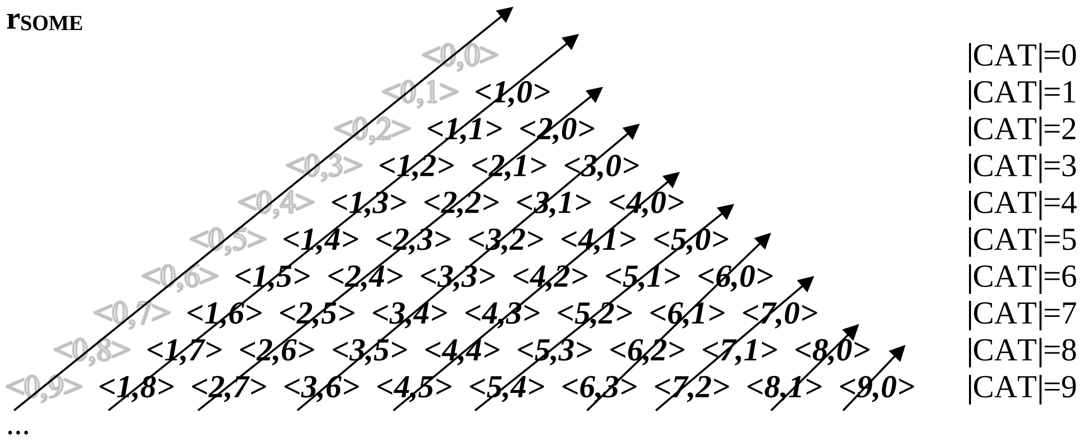
r_{every} is **not** symmetric:

rEVERY



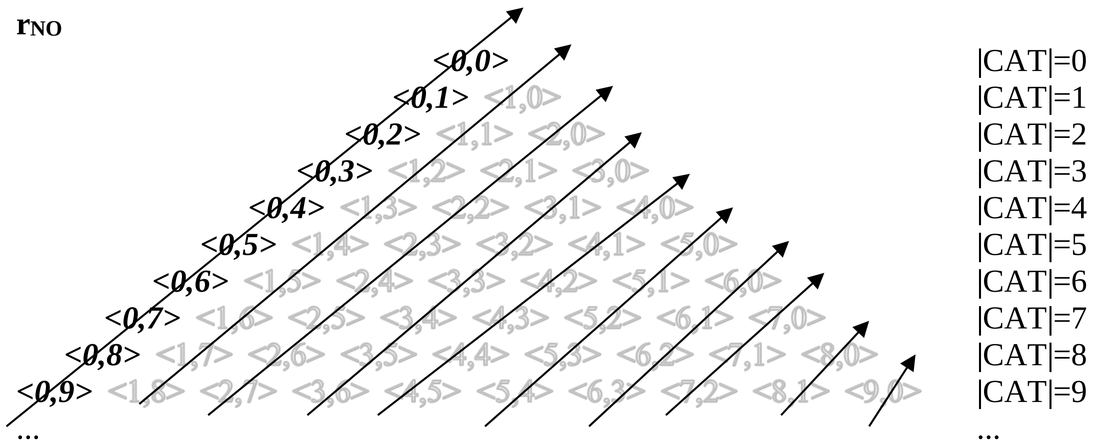
r_{some} **is** symmetric:

rSOME



r_{no} **is** symmetric:

rNO



It is easy to check that $r_{\text{AT LEAST } n}$, $r_{\text{AT MOST } n}$, $r_{\text{EXACTLY } n}$ are symmetric, but that r_{most} is not symmetric.

SEMANTIC AUTOMATA, JUST A TASTE (again van Benthem)

CAT CAT ∩ CALICO CAT − CALICO

(A calico cat is three coloured: typically black, red, and white)

Give every individual in CAT a collar with the letter *i* (for *i*ntersection) or *d* (for *d*ifference):

ronya has label *i* because ronya ∈ CAT ∩ CALICO
pim has label *d* because pim ∈ CAT − CALICO

In going through the set of cats, we can write a sequence:

i i i d i d i i

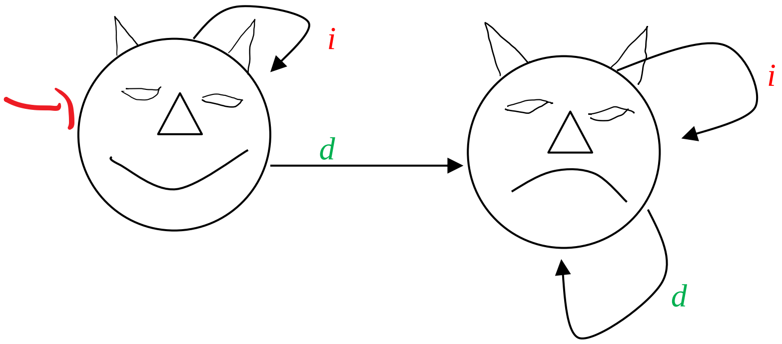
a string of labels indicating that |CAT ∩ CALICO| = 6 and
|CAT − CALICO| = 2

the set of strings α in alphabet {*i*, *d*} such that *d* doesn't occur in α
{*e*, *i*, *ii*, *iii*, *iiii*,….} (*e* is the empty string)

The *some* language is
the set of strings α in alphabet {*i*, *d*} such that i does occur in α
{*i*, *id*, *di*, *idd*, *did*, *ddi*, *iid*, *idi*, *dii*,*iid*,…}

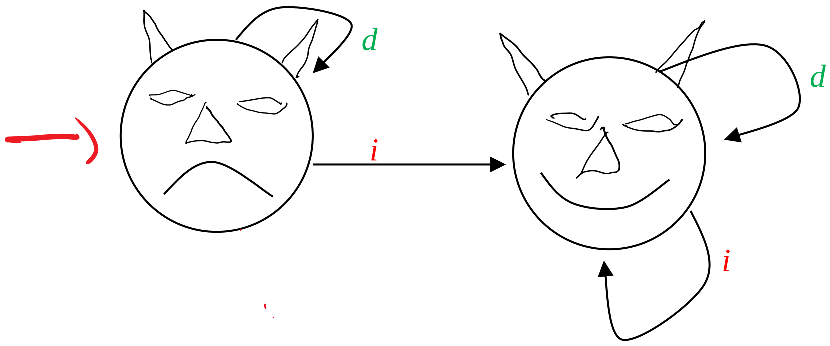
The *most* language is the set of strings with more *d*'s and *i*'s.
ddiiidiii is in the *most* language, *ddddddiid* is not.
etc.

The *every* automaton (smiley's are accepting states):



Accepted: *iiiii*
Rejected: *iiiidi*
(one cat isn't three coloured)

The *some* automaton:



Accepted: *ddddid*
Rejected: *dddd*

Fact: For every determiner definable in predicate logic, there is a finite state automaton accepting its language (regular)

-Some 'determiners' that are not definable in predicate logic have a language accepted by a finite state automaton (*an even number of*)

- the *most* language is not accepted by a finite state automaton. the *most* language is accepted by a pushdown storage automaton (context free).

Push down storage automaton: while reading you can push symbols onto a memory store or pop symbols from the store, where the store is a first-in last-out memory.

The *most* automaton:

Start: You start reading the first symbol of the string and an empty store.

Move: 1. If the top of the store is empty, push what you read on the input on top of the store.

2. If what you read on the input is the same as what is on top of the store, then push what you read on the input on the top of the store.

3. If the store is not empty and what you read on the input is different from what is on top of the store, then pop the topsymbol off the store.

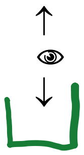
4. In each case move to reading the next right symbol on the input tape.

End: When you reach the end symbol # on the input tape stop and accept the string if there are *i*'s in the store, otherwise reject the string.

Example. We read the string: *idiiddiddii*

We start:

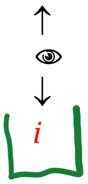
\# *i* *d* *i* *i* *d* *d* *i* *d* *d* *i* *i* #



You read *i* on the input tape, nothing in the store.

Read the next symbol on the input tape and add the *i* that you read on the input to the top of the store:

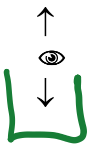
\# *i* *d* *i* *i* *d* *d* *i* *d* *d* *i* *i* #



You read *d* on the input tape and *i* on the store. They are different. So:

Read the next symbol on the input tape and remove the *i* from the top of the store:

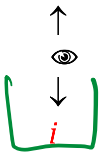
\# *i* *d* *i* *i* *d* *d* *i* *d* *d* *i* *i* #



You read *i* on the input tape, nothing in the store.

Read the next symbol on the input tape and add the *i* that you read on the input to the top of the store:

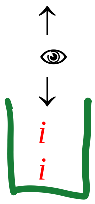
\# *i* *d* *i* *i* *d* *d* *i* *d* *d* *i* *i* #



You read *i* on the input tape, and *i* on the top of the store. They are the same.

Read the next symbol on the input tape and add the *i* that you read on the input to the top of the store:

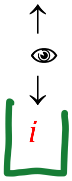
\# *i* *d* *i* *i* *d* *d* *i* *d* *d* *i* *i* #



You read *d* on the input tape and *i* on the store. They are different. So:

Read the next symbol on the input tape and remove the *i* from the top of the store:

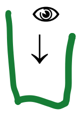
i *d* *i* *i* *d* *d* *i* *d* *d* *i* *i*



You read *d* on the input tape and *i* on the store. They are different. So:

Read the next symbol on the input tape and remove the *i* from the top of the store:

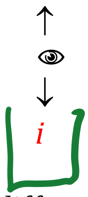
i *d* *i* *i* *d* *d* *i* *d* *d* *i* *i*



You read *i* on the input tape, nothing in the store.

Read the next symbol on the input tape and add the *i* that you read on the input to the top of the store:

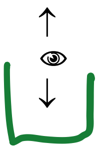
i *d* *i* *i* *d* *d* *i* *d* *d* *i* *i*



You read *d* on the input tape and *i* on the store. They are different. So:

Read the next symbol on the input tape and remove the *i* from the top of the store:

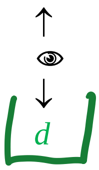
i *d* *i* *i* *d* *d* *i* *d* *d* *i* *i*



You read *d* on the input tape, nothing in the store.

Read the next symbol on the input tape and add the *d* that you read on the input to the top of the store:

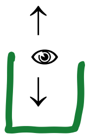
i *d* *i* *i* *d* *d* *i* *d* *d* *i* *i*



You read *i* on the input tape and *d* on the store. They are different. So:

Read the next symbol on the input tape and remove the *d* from the top of the store:

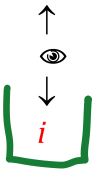
\# *i* *d* *i* *i* *d* *d* *i* *d* *d* *i* *i* #



You read *i* on the input tape, nothing in the store.

Read the next symbol on the input tape and add the *i* that you read on the input to the top of the store:

\# *i* *d* *i* *i* *d* *d* *i* *d* *d* *i* *i* #



You read end symbol # on the input tape, *i* on top of the store.

You stop and you accept the string, since there are *i*'s in the store.
This means that the automaton accepts the string *idiiddiddii*.

This means that it represents a situation in which MOST[CAT, CALICO] is true, which is good, since there are 6 CATS that are calico and 5 cats that are not.