Formal Semantics of Natural Language

Yoad Winter

General Course Info

Topics: demonstrate basic principles and tools in formal approaches to natural language meaning

Relevance: linguists, logicians, computer scientists, and philosophers

Prior knowledge:
- sets, relations, functions
- aims of theoretical linguistics

Reading: Edinburgh University Press, April 2016

www.phil.uu.nl/~yoad/efs/main.html

Sessions

1. **Formal semantics:** the study of logical meaning in natural language
2. **Types and meaning composition**: How to describe properties of meanings? How does language allow us to combine them?
3. **Tutorial 1** (Sunday, 9:30-10:00)
4. **Generalized quantifiers:** How to describe meanings of expressions involving quantity, like *all*, *some* and *most?*
5. **Tutorial 2** (Sunday, 2:00-2:30)
6. **Spatial expressions:** Meanings of relations in space, such as *above*, *in* and *between*.
7. **Abstract categorial grammar / ExpSem**: long distance dependencies / experiments reciprocals

Exercises

Will be given before tutorial sessions.

It is recommended to try to solve them before the tutorial.

Soundbites

Intersective vs. non-intersective adjectives:

Tina is a *Chinese* pianist and a biologist

←→ Tina is a pianist and a *Chinese* biologist

Tina is a *skillful* pianist and a biologist

←/→ Tina is a pianist and a *skillful* biologist

Monotonicity and Negative Polarity Items:

If John *ever* goes to Moscow he will have fun

*If John goes to Moscow he will *ever* have fun

Spatial reasoning:

Dan is close to a gas station

←→ *Dan is close to some gas station*

Dan is far from a gas station

←→ *Dan is far from every gas station*

Formal Semantics of Natural Language

NASSLLI2016, Rutgers University, 9-10 July 2016

Yoad Winter, Utrecht University, The Netherlands

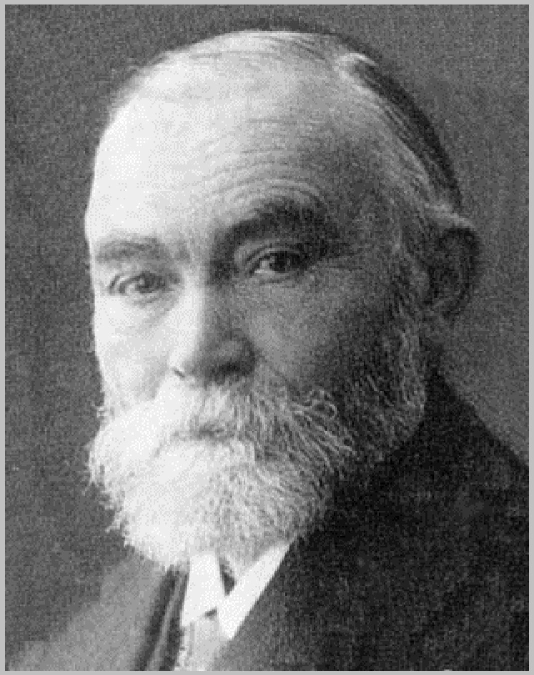
Session 1:

Basic Notions

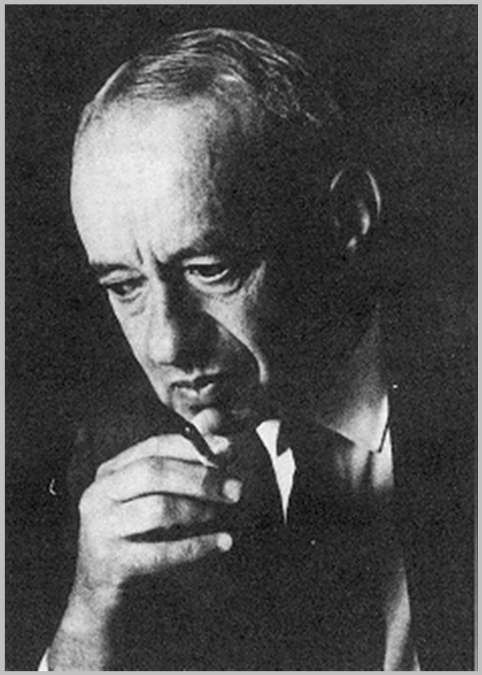
Formal Semantics:

History and Principles

Two prominent logicians on meaning



Gottlob Frege (1848-1925)



Alfred Tarski (1902-1983)

Frege: Meanings are *composed to each other*.

Tarski: Meanings can be described as objects in a *mathematical world*, external to language itself.

Meanwhile in Cognitive Science

"It seems clear, then that undeniable, though only imperfect correspondences hold between formal and semantic features in language."

(*Syntactic Structures*, 1957)

"...Chomsky would say [that] the semantic purposes do not determine the form of the syntax or even influence it in any significant way."

(*Chomsky's Revolution in Linguistics*, John R. Searle, 1972)

Noam Chomsky (1928)

Towards a Synthesis

Richard Montague
(1930-1971)

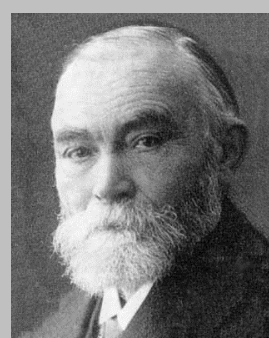
"There is in my opinion no important theoretical difference between natural languages and the artificial languages of logicians; indeed, I consider it possible to comprehend the syntax and semantics of both kinds of language within a single natural and mathematically precise theory. On this point I differ from a number of philosophers, but agree, I believe, with Chomsky and his associates."
(*Universal Grammar*, 1970)

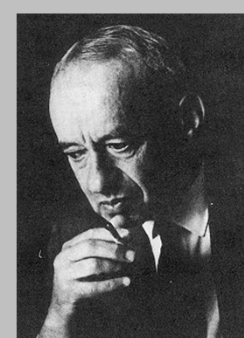
The Key to Montague's Program

Frege's Principle of Compositionality

The meaning of a compound expression is a function of the meanings of its parts, and the ways they combine with each other.

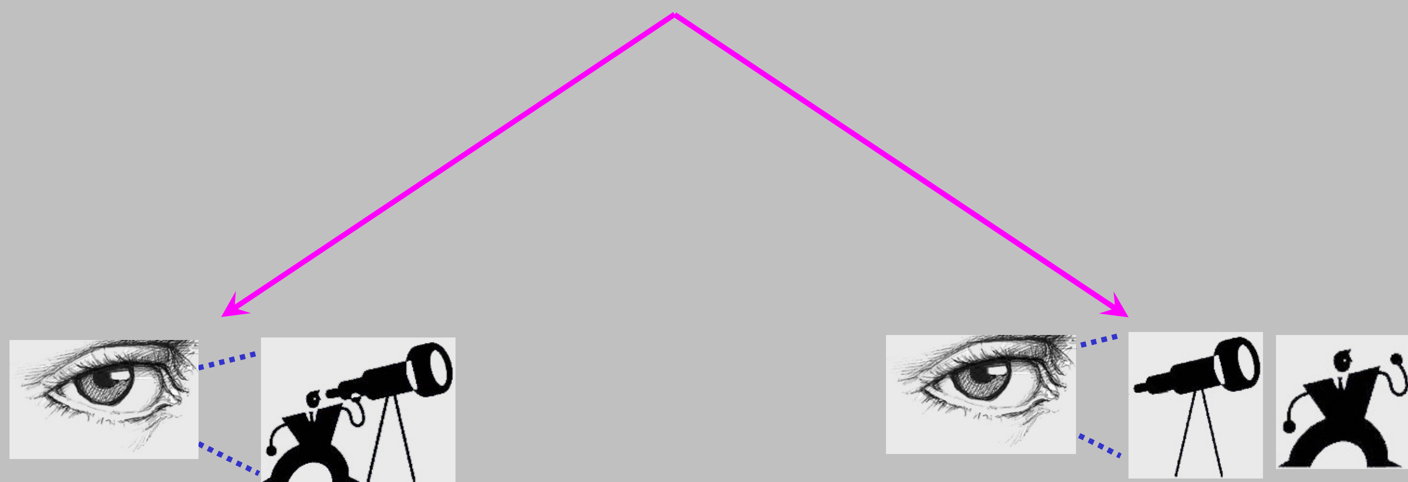
Form ⟷ Compositionality ⟷ **Meaning**

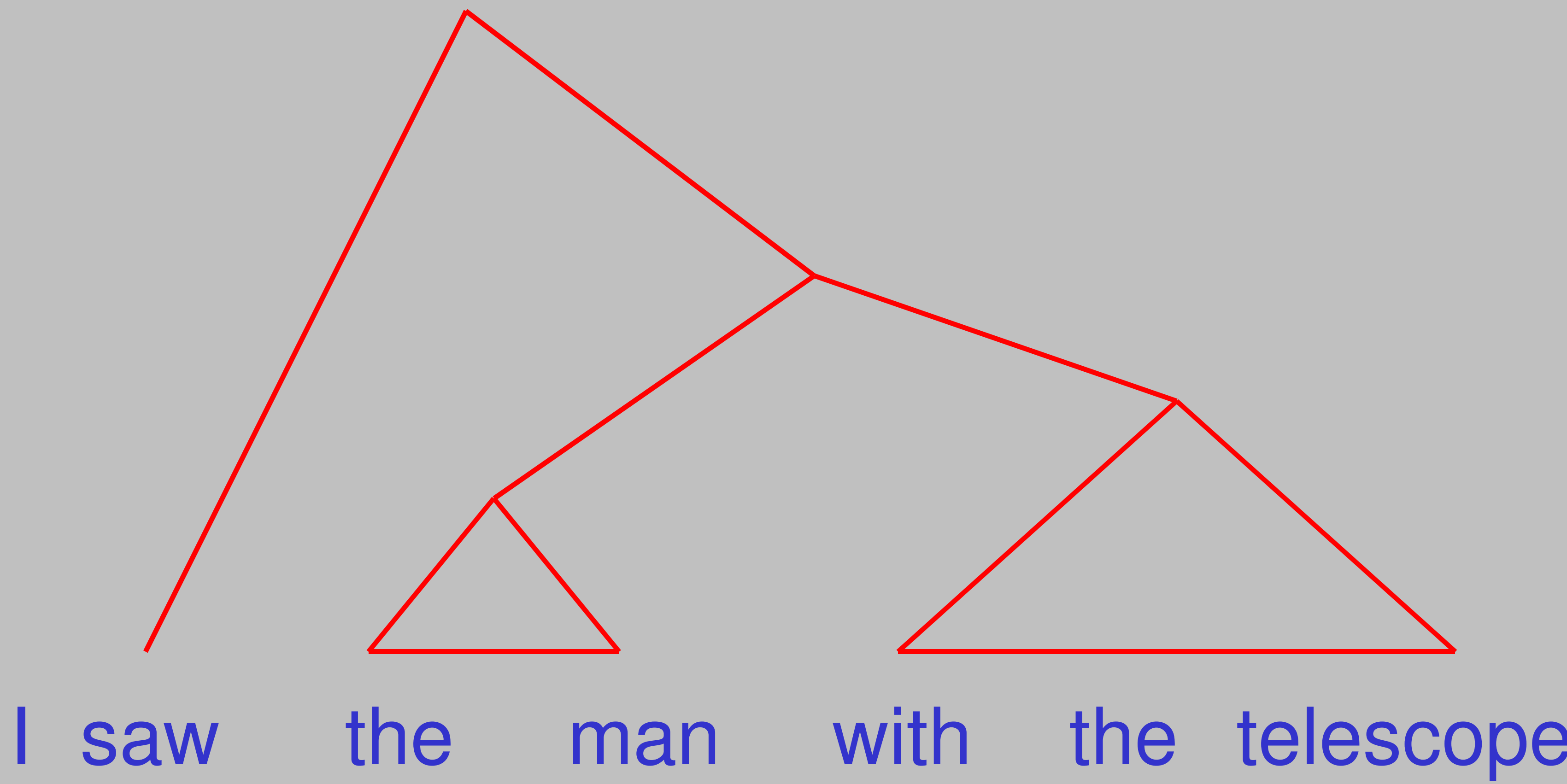


Ambiguous Expressions

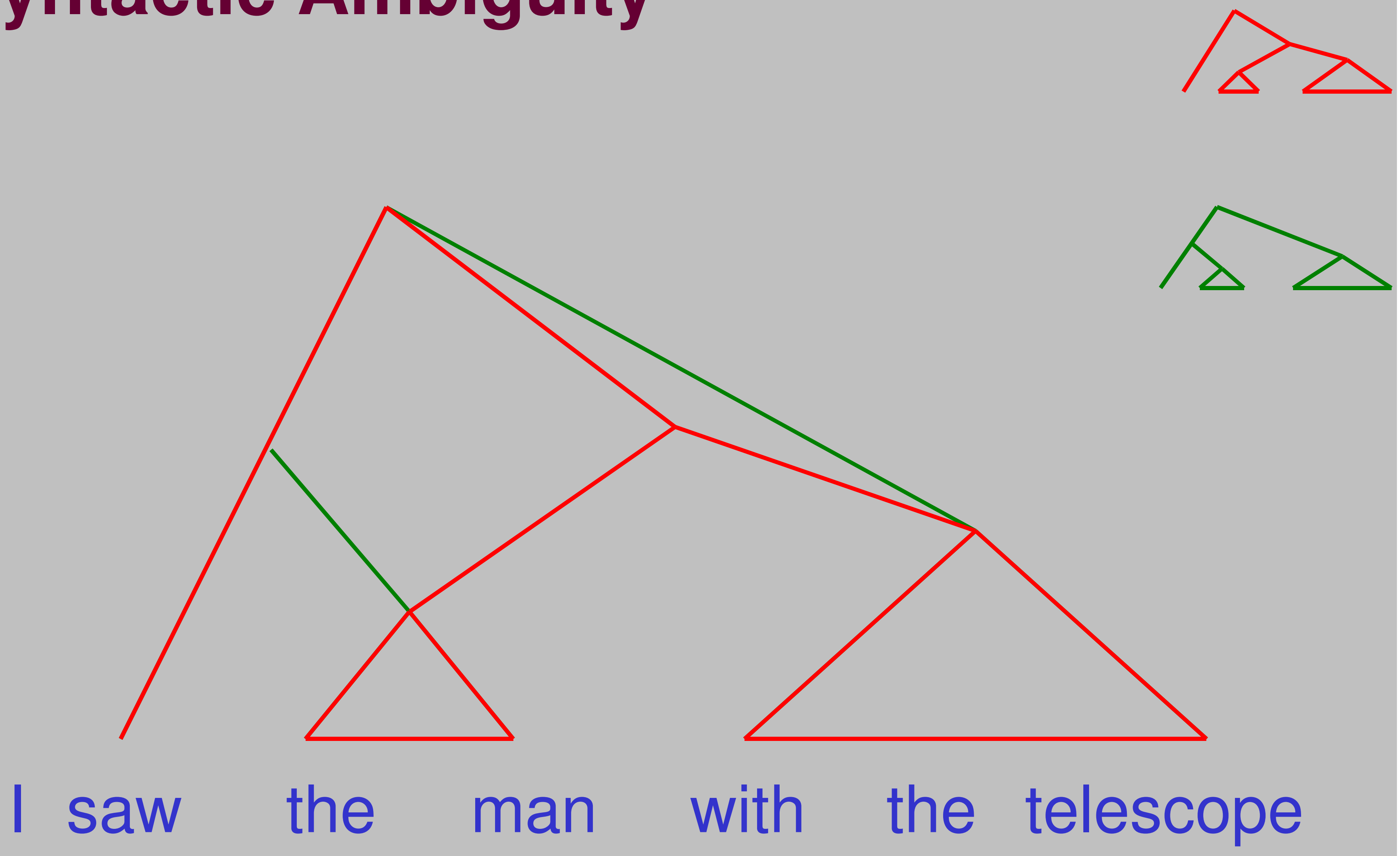
I saw the man with the telescope



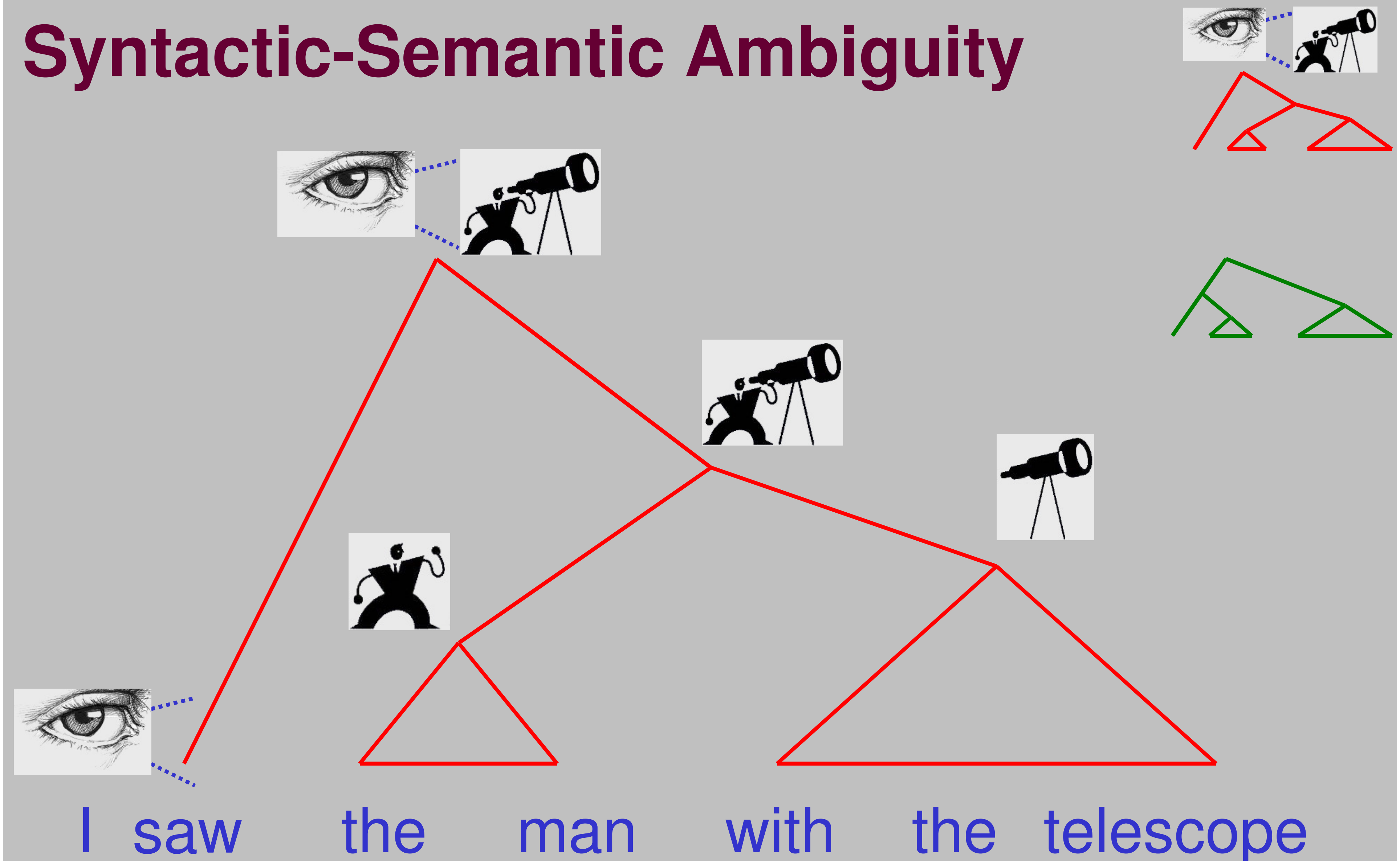
Syntactic Ambiguity



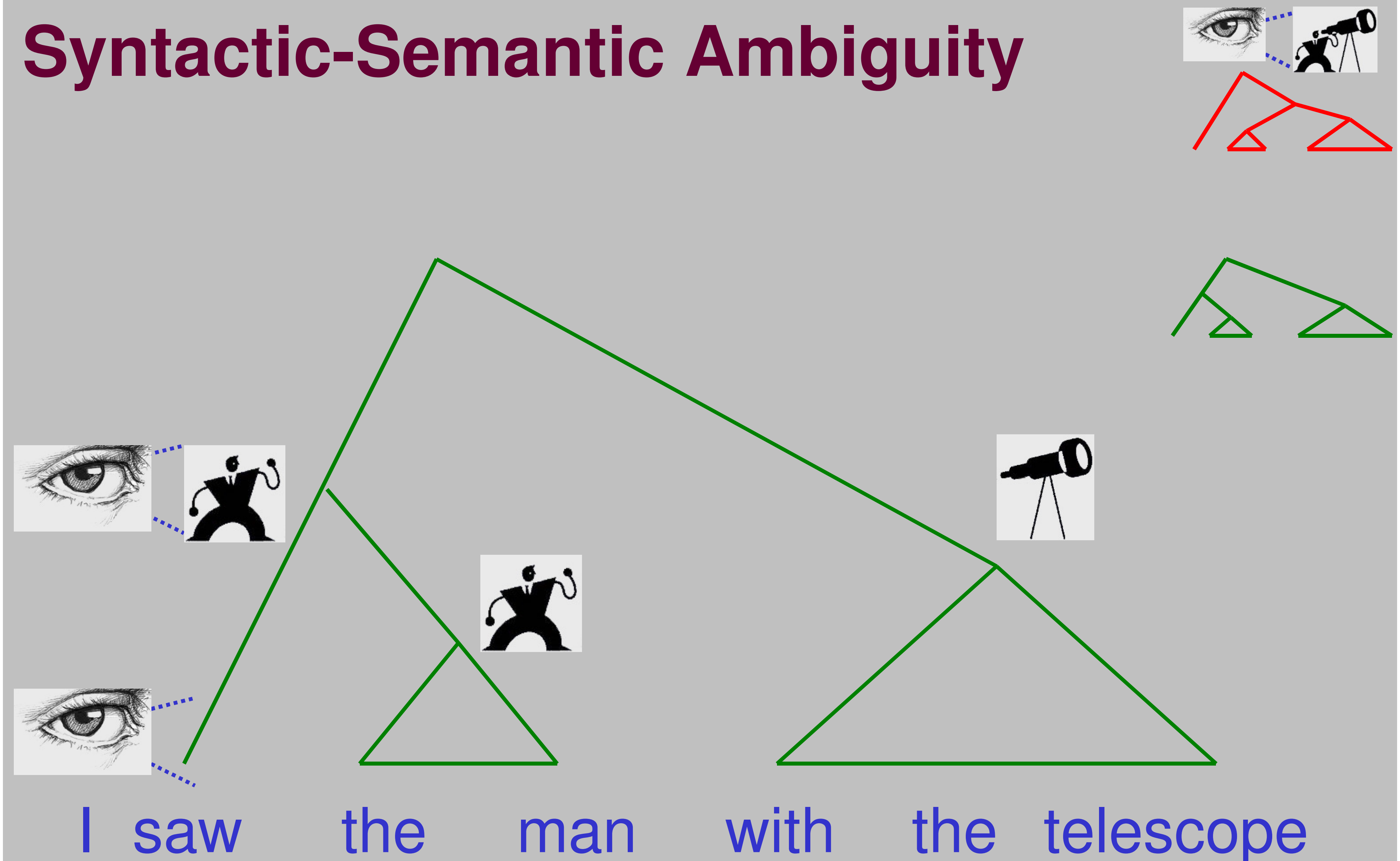
Syntactic Ambiguity



Syntactic-Semantic Ambiguity



Syntactic-Semantic Ambiguity



Notions in Set Theory

Notions in Set Theory (1)

Description of sets and their members:

Explicit: $\{1,2,3,4\}$

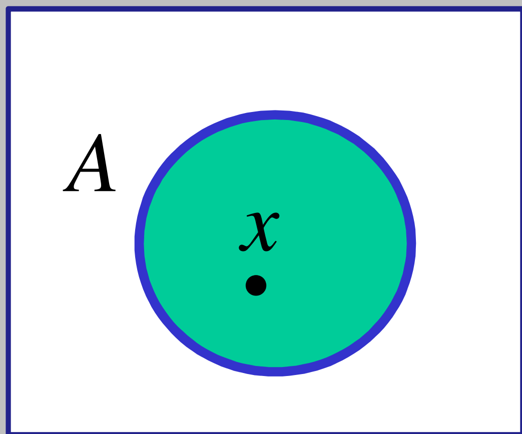
Implicit: $\{x$ is a natural number **:** x is between 1 and 4$\}$

$= \{ x \in \mathbf{N} : 1 \leq x \leq 4 \}$

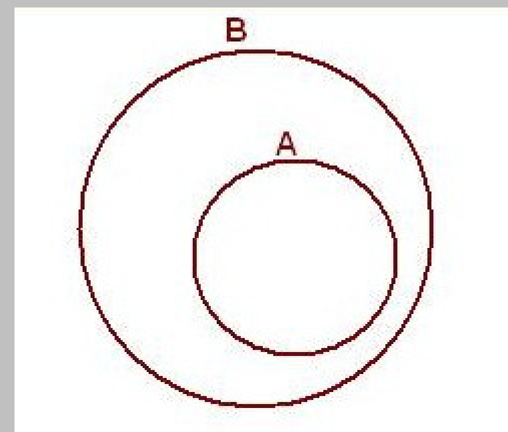
$\{x$ is a natural number **:** x is bigger than 90$\}$

$= \{ x \in \mathbf{N} : 90 \leq x \}$

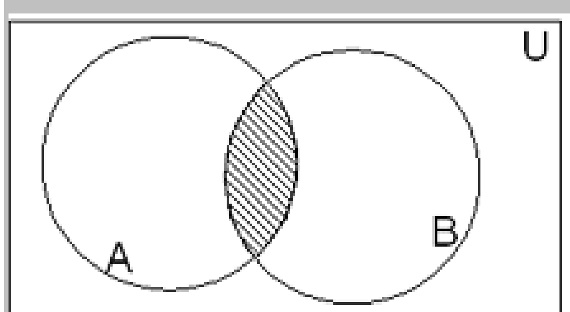
Notions in Set Theory (2)



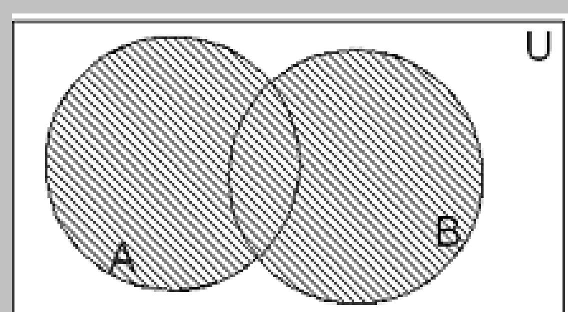
Element of: $x \in A$



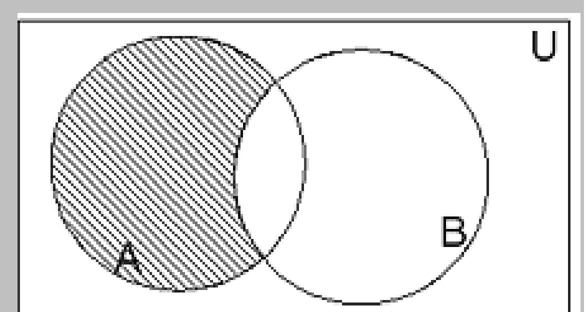
Subset: $A \subseteq B$



Intersection: $A \cap B$

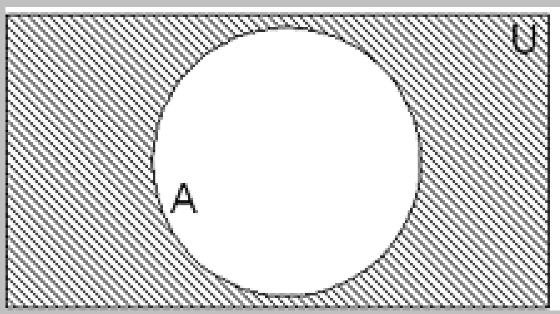


Union: $A \cup B$



Set difference: $A - B$

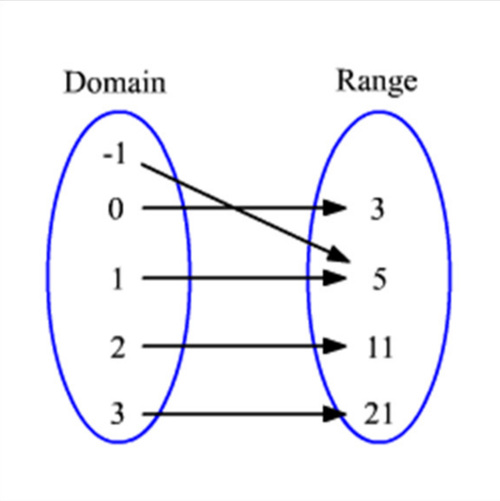
Notions in Set Theory (3)

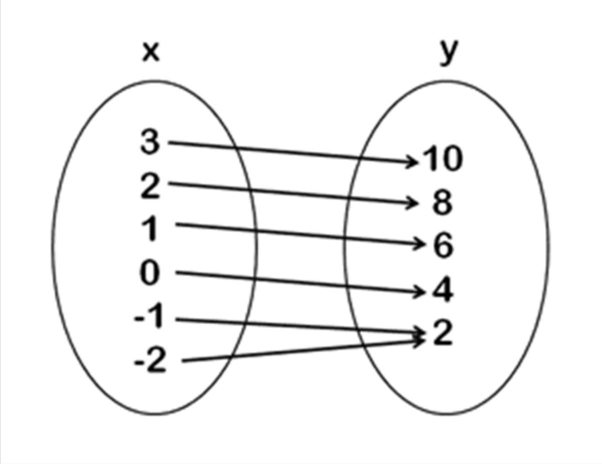


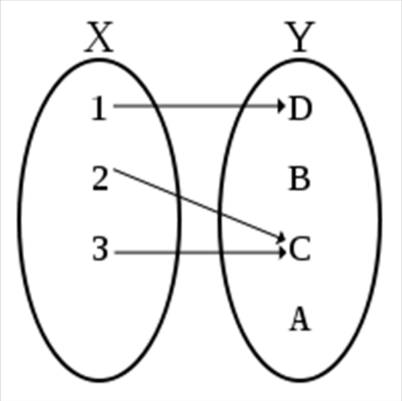
Complement: $\overline{A}$

Notions in Set Theory (4)

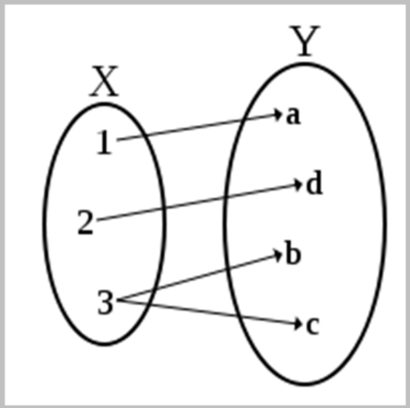
Functions:







But this is not a function:



Notions in Formal Semantics

Mentalist vs. Linguistic Meaning Relations

(1) a. What is common to the objects that people categorize as being *red*?

b. How do people react when they are addressed with the request *please pick a blue card from this pack*?

c. What emotions are invoked by expressions like *my sweetheart*, *my grandmother* or *my boss*?

(2) a. How do speakers identify relations between pairs of words like *red-color*, *dog-animal* and *chair-furniture*?

b. What are the relations between the use of the imperative sentence *please pick a <u>blue</u> card from this pack* and the use of the similar sentence *please pick a card from this pack*?

c. How are the descriptions *my grandmother* and *my only living grandmother* related to each other in language use?

(3) Red is a color / ?Red is an animal

(4) The color red annoys me / ?The animal red annoys me

(5) Every red thing has a color / ?Every red thing has an animal

Entailment

Tina is tall and thin ==> Tina is thin

premise/antecedent conclusion/consequent

Jeremy knows more than four aunts of mine $\Rightarrow$ Jeremy knows at least five aunts of mine.

A dog entered the room $\Rightarrow$ An animal entered the room.

John picked a blue card from this pack $\Rightarrow$ John picked a card from this pack.

Tina is a bird =/=> Tina can fly

Tina is a bird but she cannot fly *vis a vis*

#Tina is tall and thin, but she is not thin

Entailment *is the indefeasible relation, denoted $S_1 \Rightarrow S_2$, between a premise S_1 and a valid conclusion S_2 expressed as natural language sentences.*

More entailments

Tina is tall and thin ==> Tina is thin

(3) a. Tina is excited and she (Tina) is joyful or amazed $\Rightarrow$ Tina is excited and joyful or Tina is excited or amazed.

b. Tina is excited and joyful or Tina is excited or amazed $\Rightarrow$ Tina is excited and she (Tina) is joyful or amazed.

(4) a. Tina is tall, and Ms. Turner is not tall $\Rightarrow$ Tina is not Ms. Turner.

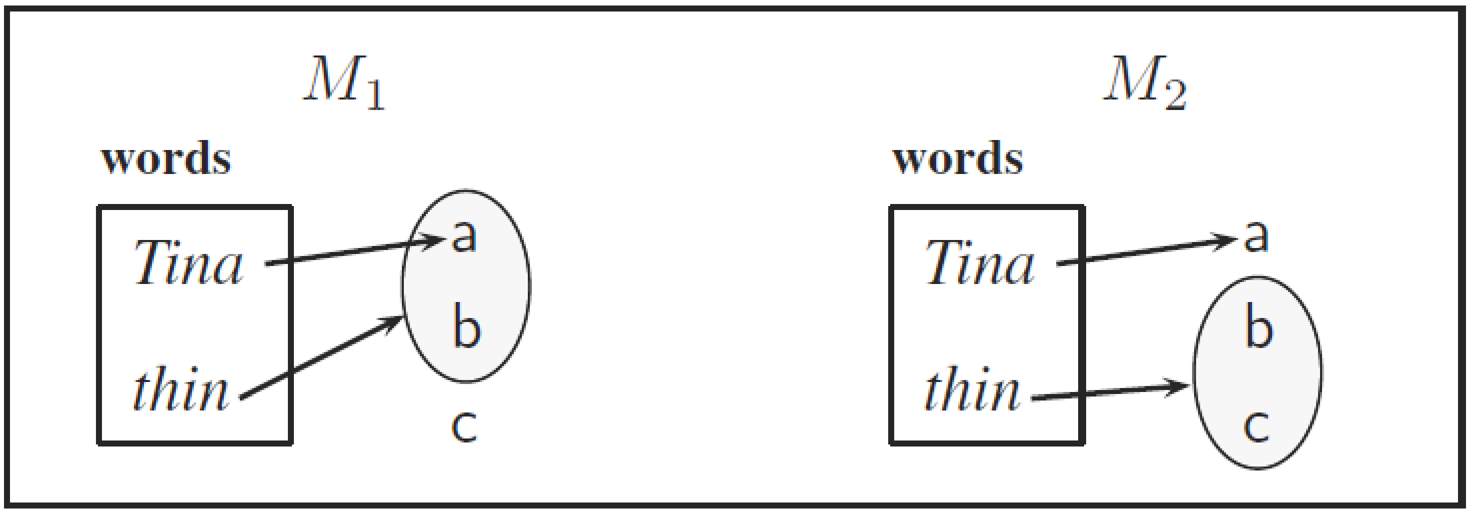
b. Tina is tall, and Tina is not Ms. Turner $\not\Rightarrow$ Ms. Turner is not tall.

(5) a. Ms. Turner is tall, and Tina is Ms. Turner or Ms. Charles $\Rightarrow$ Tina is tall or Tina is Ms. Charles.

b. Ms. Turner is tall, and Tina is Ms. Turner or Ms. Charles $\not\Rightarrow$ Tina is tall.

Models

Let exp *be a language expression, and let M be a model. We write* $[\![\text{exp}]\!]^M$ *when referring to the denotation of* exp *in the model M.*



Models and Entailment: the Truth-Conditionality Criterion

A semantic theory T is said to satisfy the **truth-conditionality criterion** (TCC) *if for all sentences S_1 and S_2, the following two conditions are equivalent:*

I. *Sentence S_1 intuitively entails sentence S_2.*

II. *For all models M in T: $[[S_1]]^M \leq [[S_2]]^M$.*

	$y = 0$	$y = 1$
$x = 0$	**yes**	**yes**
$x = 1$	**no**	**yes**

$$\underset{\text{ENTAILMENT}}{\boxed{S_1 \Rightarrow S_2}} \quad \overset{\text{TCC}}{\longleftrightarrow} \quad \underset{}{\overset{\text{MODELS}}{\boxed{\begin{array}{c} [[S_1]]^{M_1} \leq [[S_2]]^{M_1} \\ [[S_1]]^{M_2} \leq [[S_2]]^{M_2} \\ [[S_1]]^{M_3} \leq [[S_2]]^{M_3} \\ \ldots \end{array}}}}$$

Assumptions about our models

Every model has a set *E* of **entities**.

In every model, [[Tina]] is an entity.

In every model, [[tall]] and [[thin]] are sets of entities.

$$\text{IS}(x, A) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

$\text{AND}(A, B) = A \cap B$ = the set of all members of E that are both in A and in B

Thus:

$$[[\textit{Tina is thin}]]^M = \text{IS}(\mathbf{tina}, \mathbf{thin})$$

$$[[\textit{Tina is tall and thin}]]^M = \text{IS}(\mathbf{tina}, \text{AND}(\mathbf{tall}, \mathbf{thin}))$$

Convention:

Let blik *be a word in a language. When the denotation* $[[\text{blik}]]^M$ *of* blik *is arbitrary, we mark it* **blik**, *and when it is constant across models we mark it* BLIK. *In both notations the model M is implicit.*

TCC - example

Expression	Cat.	Type	Abstract denotation	Denotations in example models with $E = \{a,b,c,d\}$		
				M_1	M_2	M_3
Tina	PN	entity	$\mathbf{tina}$	a	b	b
tall	A	set of entities	$\mathbf{tall}$	$\{b,c\}$	$\{b,d\}$	$\{a,b,d\}$
thin	A	set of entities	$\mathbf{thin}$	$\{a,b,c\}$	$\{b,c\}$	$\{a,c,d\}$
tall and thin	AP	set of entities	$\textsc{and}(\mathbf{tall},\mathbf{thin})$	$\{b,c\}$	$\{b\}$	$\{a,d\}$
Tina is thin	S	truth-value	$\textsc{is}(\mathbf{tina},\mathbf{thin})$	1	1	0
Tina is tall and thin	S	truth-value	$\textsc{is}(\mathbf{tina},\textsc{and}(\mathbf{tall},\mathbf{thin}))$	0	1	0

In all three models we have:

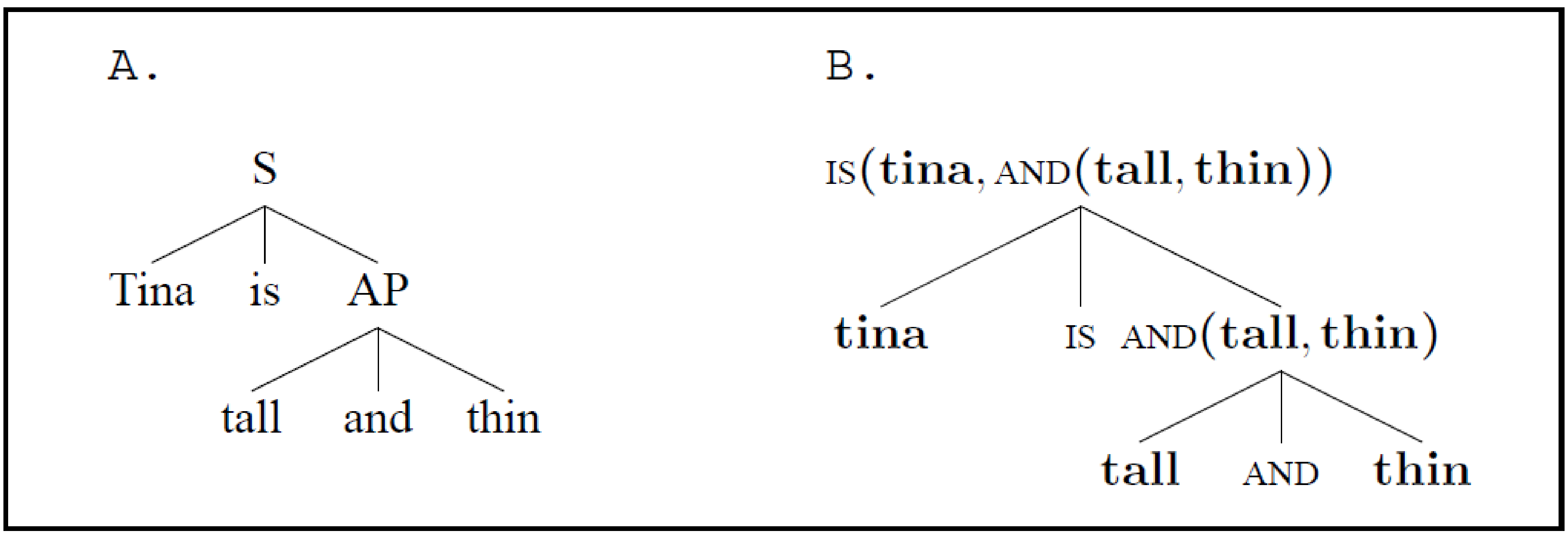
$$\textsc{is}(\mathbf{tina},\textsc{and}(\mathbf{tall},\mathbf{thin})) \leq \textsc{is}(\mathbf{tina},\mathbf{thin})$$

Compositionality

All pianists are composers, and Tina is a pianist.

All composers are pianists, and Tina is a pianist.

Compositionality: *The denotation of a complex expression is determined by the denotations of its immediate parts and the ways they combine with each other.*



Structural ambiguity (1)

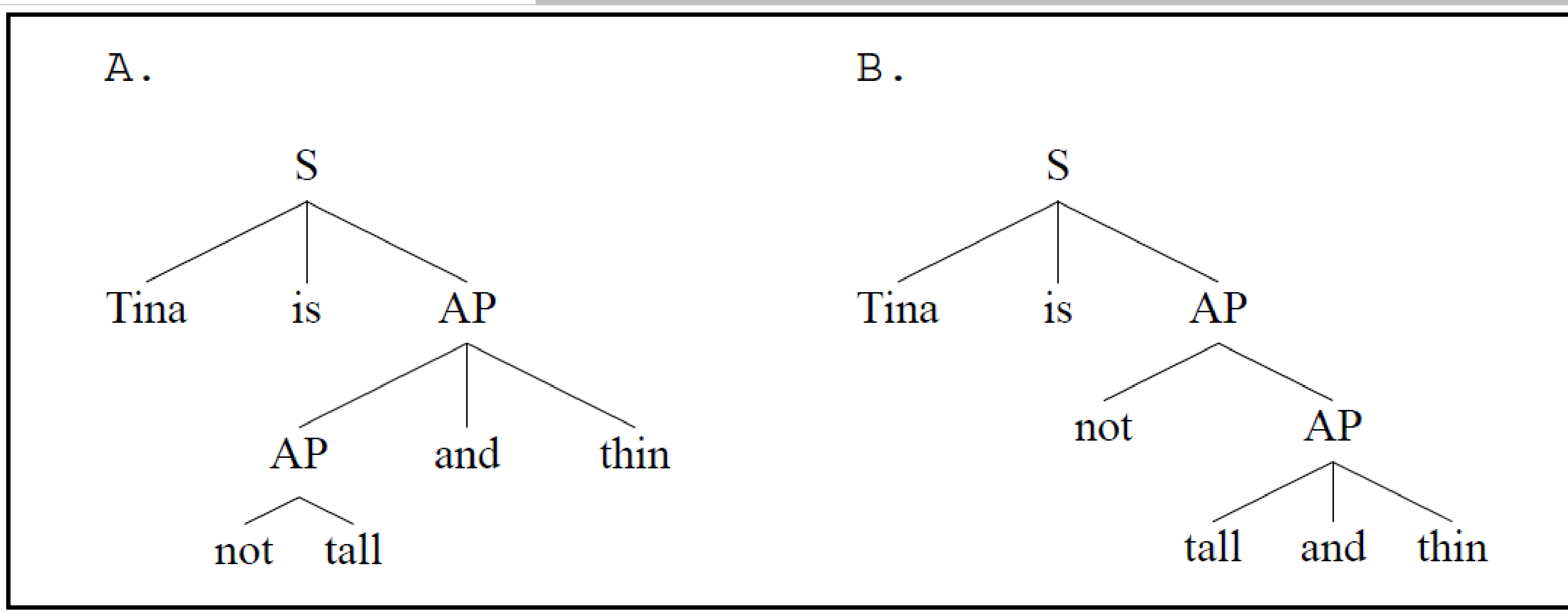
Tina is not tall and thin.

Tina is not tall, and thin.

AP $\longrightarrow$ *tall*, *thin*, ...

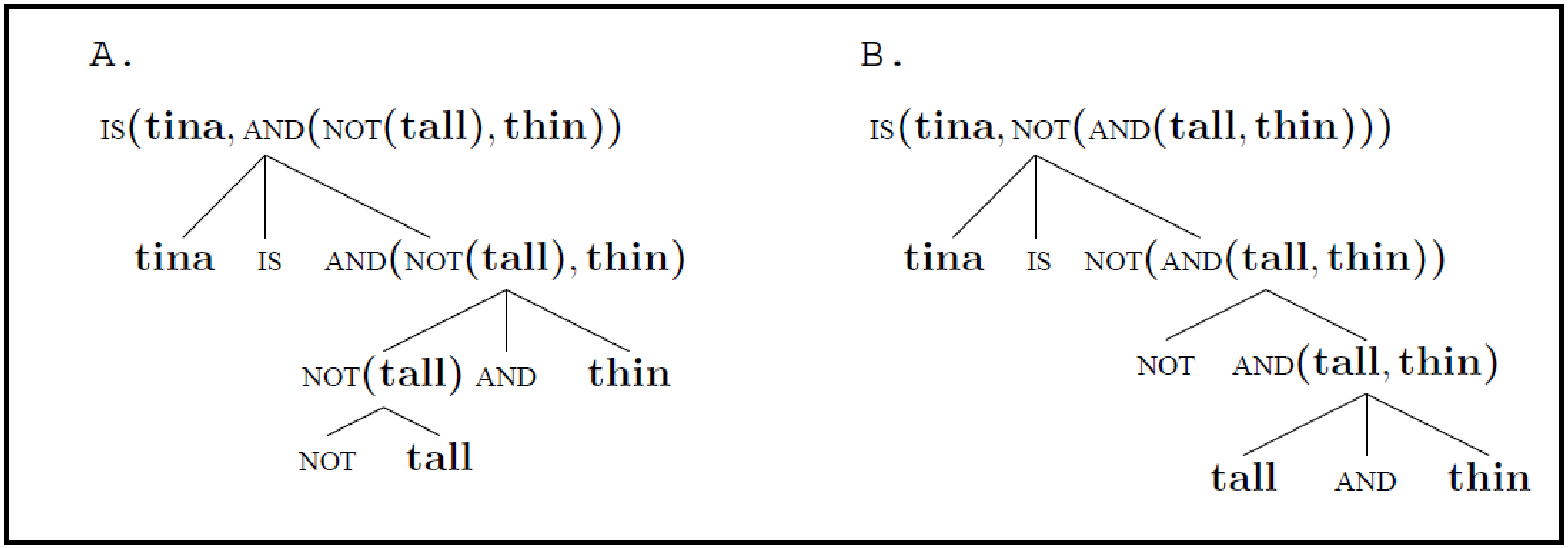
AP $\longrightarrow$ AP *and* AP

AP $\longrightarrow$ *not* AP



Structural ambiguity (2)

$\text{NOT}(A) = \overline{A} = E \setminus A$ = the set of all the members of E that are not in A



$\text{IS}(\mathbf{tina}, \text{AND}(\text{NOT}(\mathbf{tall}), \mathbf{thin})) = 1$, i.e. $\mathbf{tina} \in \overline{\mathbf{tall}} \cap \mathbf{thin}$

$\text{IS}(\mathbf{tina}, \text{NOT}(\text{AND}(\mathbf{tall}, \mathbf{thin}))) = 1$, i.e. $\mathbf{tina} \in \overline{\mathbf{tall} \cap \mathbf{thin}}$

Note: Ambiguity vs. vagueness

Summary – Main Notions

Facts:

Entailment (indefeasible)

Theory:

Model

Denotation in model

Adequacy:

Truth-Conditionality Criterion

Compositionality

Structural ambiguity

Session 2:

Types and Meaning composition

This Lecture

A general theory of model structure; a general semantic practice for meeting the TCC.

1 – working with types and denotations

2 – using lambda notation

3 – restricting denotations

Working with types and denotations

Basic/Complex Types and Domains

A **type** is a label for part of a model that is called a **domain**.

Basic types and domains:

e	:	D_e	*- arbitrary*	*- of entities*
t	:	D_t	$= \{0,1\}$	*- of truth-values*

Complex types and domains: defined inductively from basic types and domains.

Complex Types – Example

E = D_e = the set of entities {t,j,m}

[[*thin*]] = T = {t,j}

We can also define T as a *function* from D_e to D_t:

t→1
j→1
m→0

This function **characterizes** T in E = D_e.

D_{et} of the complex type ***et*** is the **domain** of such functions.

Characteristic functions over {t,j,m}

Subset of D_e	**Function in D_{et}**			
$\varnothing$	$f_1:$	$\mathsf{t} \mapsto 0$	$\mathsf{j} \mapsto 0$	$\mathsf{m} \mapsto 0$
$\{\mathsf{m}\}$	$f_2:$	$\mathsf{t} \mapsto 0$	$\mathsf{j} \mapsto 0$	$\mathsf{m} \mapsto 1$
$\{\mathsf{j}\}$	$f_3:$	$\mathsf{t} \mapsto 0$	$\mathsf{j} \mapsto 1$	$\mathsf{m} \mapsto 0$
$\{\mathsf{j}, \mathsf{m}\}$	$f_4:$	$\mathsf{t} \mapsto 0$	$\mathsf{j} \mapsto 1$	$\mathsf{m} \mapsto 1$
$\{\mathsf{t}\}$	$f_5:$	$\mathsf{t} \mapsto 1$	$\mathsf{j} \mapsto 0$	$\mathsf{m} \mapsto 0$
$\{\mathsf{t}, \mathsf{m}\}$	$f_6:$	$\mathsf{t} \mapsto 1$	$\mathsf{j} \mapsto 0$	$\mathsf{m} \mapsto 1$
$\{\mathsf{t}, \mathsf{j}\}$	$f_7:$	$\mathsf{t} \mapsto 1$	$\mathsf{j} \mapsto 1$	$\mathsf{m} \mapsto 0$
$\{\mathsf{t}, \mathsf{j}, \mathsf{m}\}$	$f_8:$	$\mathsf{t} \mapsto 1$	$\mathsf{j} \mapsto 1$	$\mathsf{m} \mapsto 1$

Table 2.1: Subsets of D_e and their characteristic functions in D_{et}

Characteristic Functions

Let X be any set.

Every subset $A \subseteq X$ gives us a function $f_A : X \to \{0, 1\}$ called the characteristic function of A.

It is defined as follows:

$$f_A(x) \quad = \quad \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

Example: $X = \{a, b, c, d\}, A = \{a, c\}$

Here are some examples of how this function f_A works:

- $f_A(a) = 1.$
- $f_A(b) = 0.$
- $f_A(c) = 1.$
- $f_A(d) = 0.$

Definitions: Types and domains

Definition 1. The set of **types** over the basic types e and t is the smallest set $\mathcal{T}$ that satisfies:

(i) $\{e, t\} \subseteq \mathcal{T}$

(ii) If τ and σ are types in $\mathcal{T}$ then $(\tau\sigma)$ is also a type in $\mathcal{T}$.

$$e,\ t,$$

$$ee,\ tt,\ et,\ te,$$

$$e(ee),\ e(tt),\ e(et),\ e(te),\ t(ee),\ t(tt),\ t(et),\ t(te),$$

$$(ee)e,\ (tt)e,\ (et)e,\ (te)e,\ (ee)t,\ (tt)t,\ (et)t,\ (te)t,$$

$$(ee)(ee),\ (ee)(tt),\ (ee)(et),\ (ee)(te),\ (tt)(ee),\ (tt)(tt),\ (tt)(et),\ (tt)(te)$$

Definition 2. For all types τ and σ in $\mathcal{T}$, the **domain** $D_{\tau\sigma}$ of the type $(\tau\sigma)$ is the set $D_\sigma^{D_\tau}$ – the functions from D_τ to D_σ.

$$\begin{aligned} D_{e(et)} &\text{ is the set of functions from } D_e \text{ to } D_{et} \\ &= \text{the functions from entities to } D_{et} \\ &= \text{the functions from entities to the functions from } D_e \text{ to } D_t \\ &= \text{the functions from entities to the functions from entities to truth-values.} \end{aligned}$$

Intransitive verbs

Tina smiled.

$$\mathbf{smile}_{et}(\mathbf{tina}_e)$$

Function Application

From types e and et, FA gives t (as we have seen above).

From types $(e(et))(et)$ and $e(et)$, FA gives et.

Types $(e(et))(et)$ and et cannot combine using FA: neither of these types is a prefix of the other.

$$\mathbf{tina}_e + \mathbf{smile}_{et} = \mathbf{smile(tina)} : t.$$

In more general terms, our type-based rule of function application is given below.

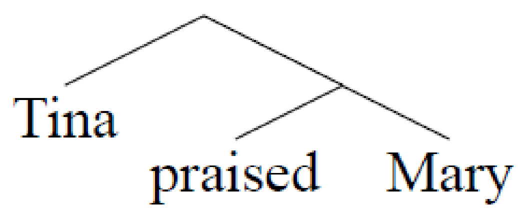
> **Function application with typed denotations**: *applying a function f of type $\tau\sigma$ to an object x of type τ gives an object $f(x)$ of type σ. In short –*
>
> $$\begin{array}{llllll} \text{Types :} & (\tau\sigma) + \tau & = & \tau + (\tau\sigma) & = & \sigma \\ \text{Denotations :} & f_{\tau\sigma} + x_\tau & = & x_\tau + f_{\tau\sigma} & = & f(x) : \sigma \end{array}$$

Intransitive and Transitive verbs

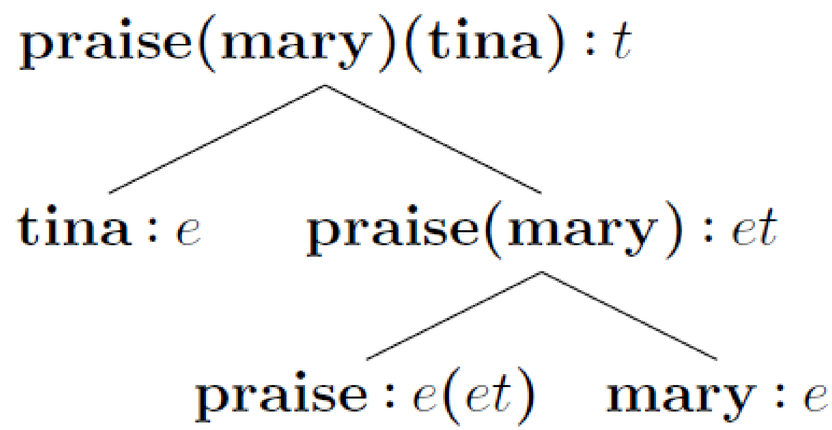
Tina smiled.

Tina [praised Mary].

A.



B.



$$\mathbf{praise}_{e(et)} + \mathbf{mary}_e \; = \; \mathbf{praise(mary)} : et$$

$$\mathbf{praise(mary)} + \mathbf{tina}_e = (\mathbf{praise(mary)})(\mathbf{tina}) : t$$

$$= \; \mathbf{praise(mary)(tina)}$$

"Curried" Relations

$$U = \{\langle \mathsf{t}, \mathsf{m}\rangle, \langle \mathsf{m}, \mathsf{t}\rangle, \langle \mathsf{m}, \mathsf{j}\rangle, \langle \mathsf{m}, \mathsf{m}\rangle\}$$

$$f_U: \begin{array}{lll} \mathsf{t} & \mapsto & [\mathsf{t} \mapsto 0 \quad \mathsf{j} \mapsto 0 \quad \mathsf{m} \mapsto 1] \\ \mathsf{j} & \mapsto & [\mathsf{t} \mapsto 0 \quad \mathsf{j} \mapsto 0 \quad \mathsf{m} \mapsto 1] \\ \mathsf{m} & \mapsto & [\mathsf{t} \mapsto 1 \quad \mathsf{j} \mapsto 0 \quad \mathsf{m} \mapsto 1] \end{array}$$

– f_U maps the entity t to the function characterizing the set $\{\mathsf{m}\}$.

– f_U maps the entity j to the function characterizing the same set, $\{\mathsf{m}\}$.

– f_U maps the entity m to the function characterizing the set $\{\mathsf{t}, \mathsf{m}\}$.

When the function f_U is the denotation of the verb *praise*, and the entities t, j and m are the denotations of the respective names, this is the situation where:

- Mary is the only one who praised Tina.
- Mary is the only one who praised John.
- Tina and Mary, but not John, praised Mary.

Currying

F: $(M \times W) \rightarrow [0,1]$

F gives any pair of man and woman (m,w) a score $F(m,w)$ indicating matching

G: $M \rightarrow (W \rightarrow [0,1])$

G gives any man m a function $G(m)$ mapping any woman w to a score $(G(m))(w)$.

Thus, we can define: $(G(m))(w) = F(m,w)$

We say that G is the Curried version of F, and that F is the deCurried version of G.

A numerical example of Currying

The function ADD sends every number N to:

the function ADD(N) that sends every number M to:

N+M

Thus: (ADD(N))(M) = N+M.

But:

ADD(1) is the *successor* function.

ADD(10) adds ten to every number.

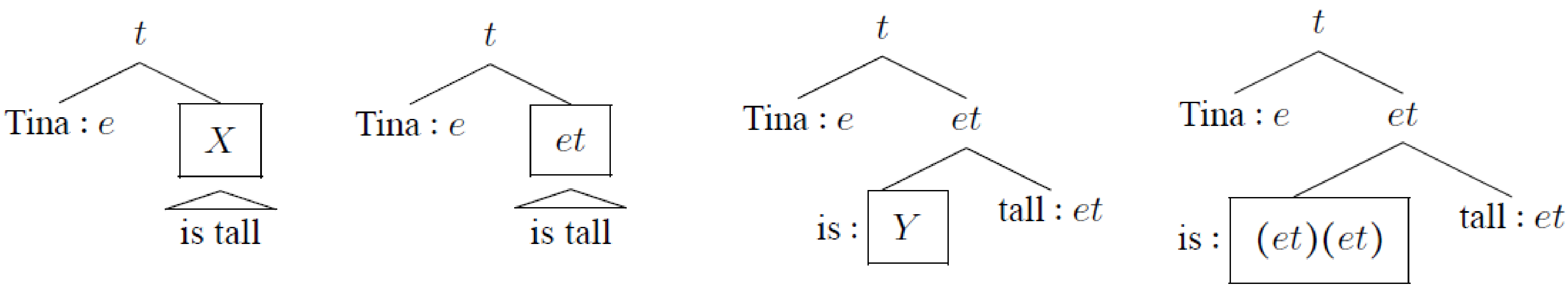
Etc.

With two-place operators like `+' it's impossible to define such functions directly.

Note: analogy between ADD(1) and **praise**(**mary**).

Solving Type Equations

$[\ \text{Tina}_e\ [\text{is}_Y\ \text{tall}_{et}]_X\ ]_t$

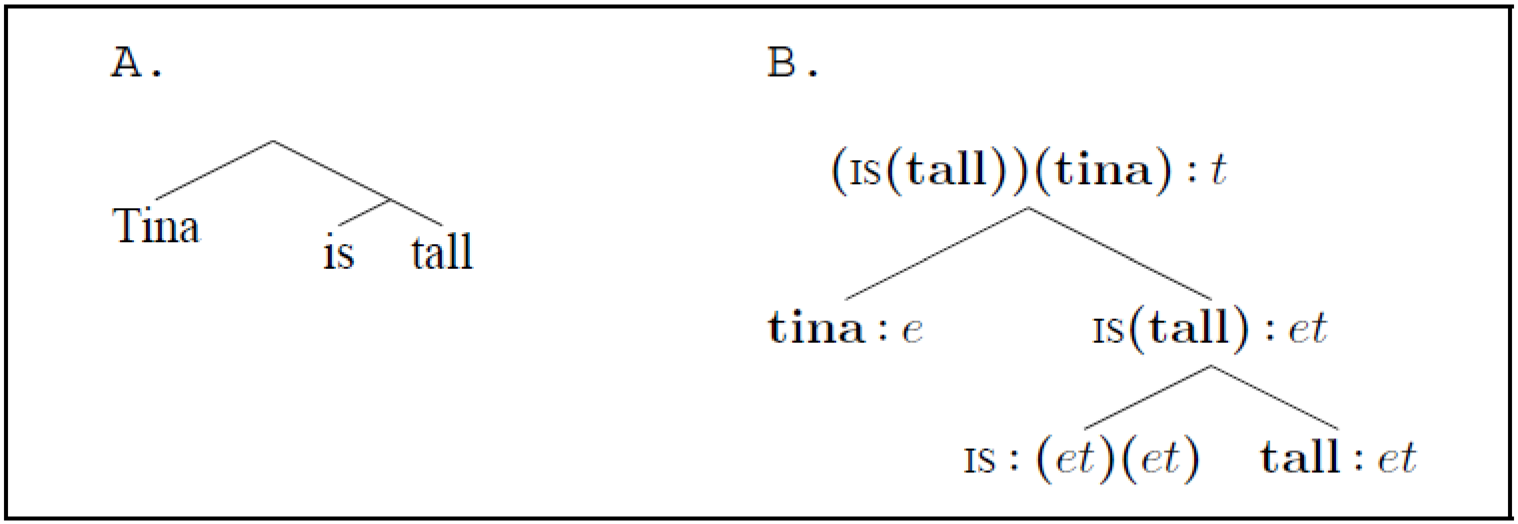


Eq. 1: $e + X = t$ **Sol. 1**: $X = et$ **Eq. 2**: $Y + et = et$ **Sol. 2**: $Y = (et)(et)$

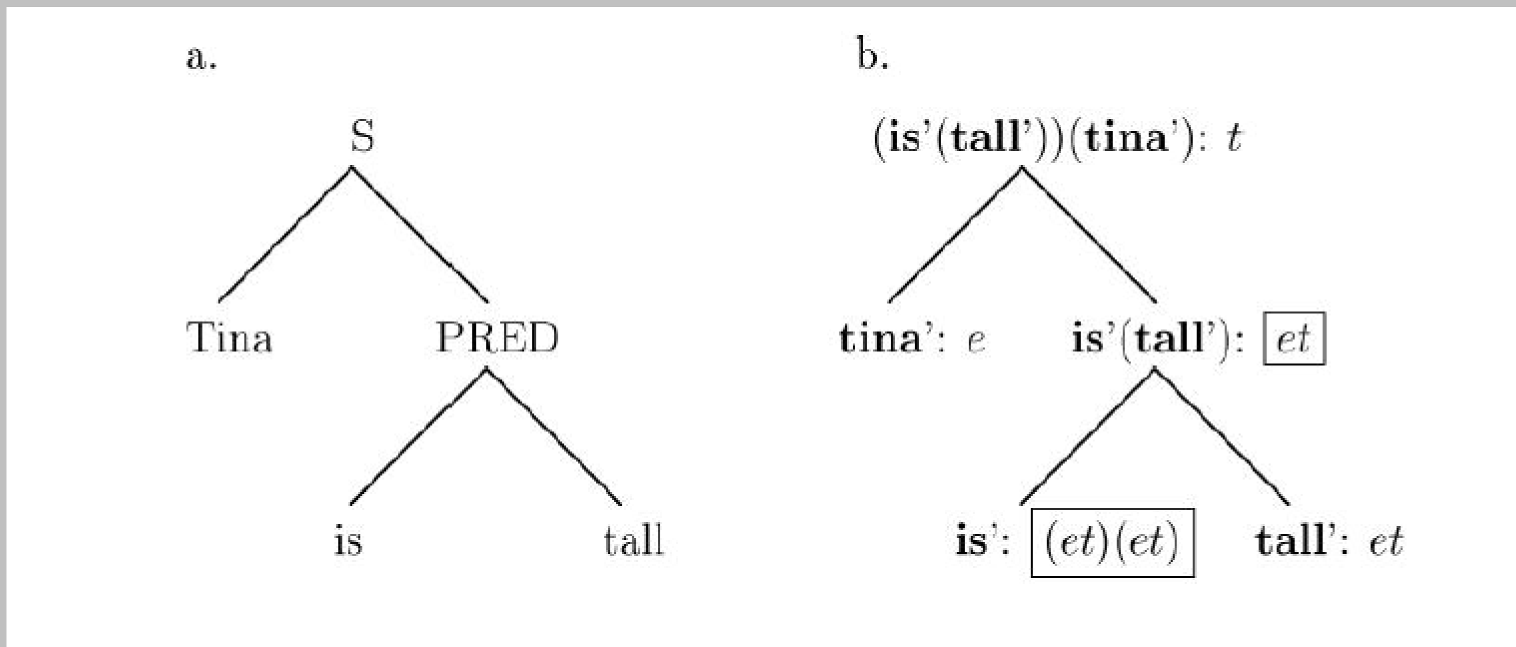
Conclusion: type of *is* should be (*et*)(*et*)

Non-arbitrary Denotations: IS

For every function f in D_{et}: $\textsc{is}(f) = f$.



Function application & constituency (1)

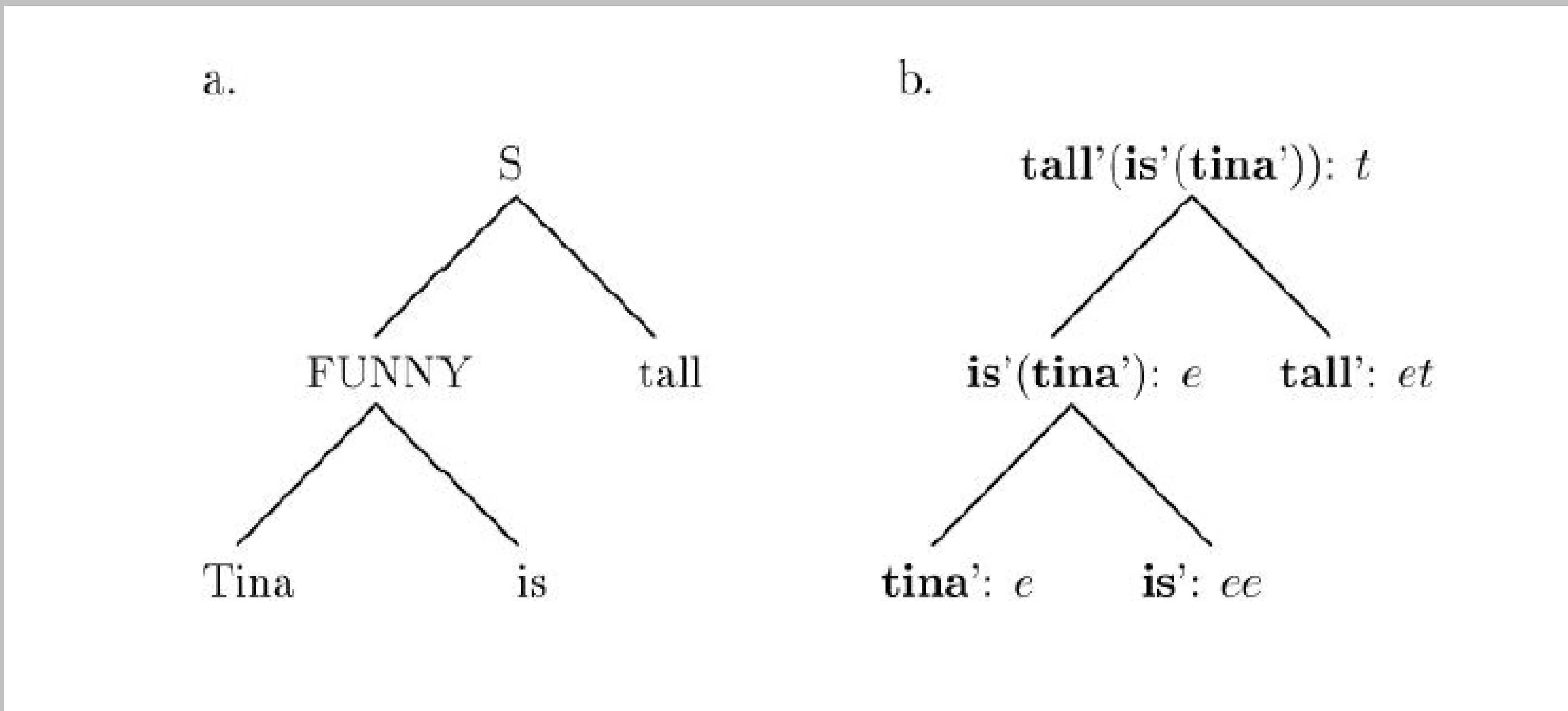


What would be the type of IS with the following (infelicitous) structure?

[Tina is] tall

What denotation would we assume for IS?

Function application & constituency (2)



Non-arbitrary Denotations: NOT

Tina [is [not tall]]

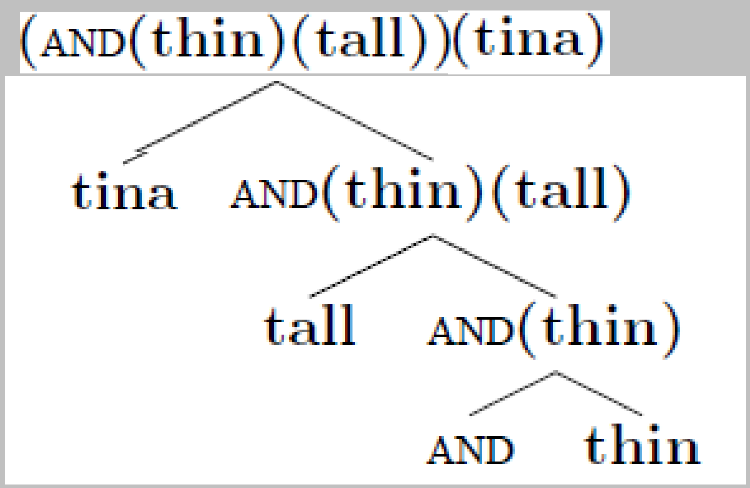
$\textsc{not}$ is the $(et)(et)$ function sending every et function g to the et function $\textsc{not}(g)$ that satisfies for every entity x:

$$(\textsc{not}(g))(x) = \begin{cases} 1 & \text{if } g(x) = 0 \\ 0 & \text{if } g(x) = 1 \end{cases}$$

Non-arbitrary Denotations: AND

For every two functions f_A and f_B in D_{et}, characterizing the subsets A and B of D_e: $(\textsc{and}(f_A))(f_B)$ is defined as the function $f_{A\cap B}$, characterizing the intersection of A and B.

Explain:
Tina is tall and thin ==> Tina is thin



Types?

In General

Types of the form

a*t*	1-place predicates	*smile*
a(a*t*)	2-place predicates	*praise*
a(a(at)))	3-place predicates	*send*
…	n-place predicates	
aa	modifiers	
	(1-place coordinators)	*is, not*
a(aa)	2-place coordinators	*X and Y*
a(a(aa))	3-place coordinators	*X, Y and Z*
…	n-place coordinators	

Using Lambda Notation

IS as Identity Function

[[Tina is tall]] = 1 -- ***tina*** denotes an entity in the set for ***tall***

With types:

$$(\mathrm{IS}_{(et)(et)}(\mathbf{tall}_{et}))(\mathbf{tina}_e)$$

Intuitively: **IS** maps any set to itself.

Formally:

$\mathrm{IS}_{(et)(et)}$ **=**

The function sending every element g of the domain D_{et} to g.

IS in lambda notation

$\text{IS}_{(et)(et)}$

The function sending every element g of the domain D_{et} to g.

Instead of writing "the function sending every element g of D_{et}" as in (55), we write "λg_{et}".

Instead of "to g" as in (55), we write "$.g$".

Thus: $\lambda g_{et}.g$ Summing up: $\text{IS} = \lambda g_{et}.g$

The letter 'λ' tells us that it is a function.

The notation 'g_{et}' before the dot introduces 'g' as an *ad hoc* name for the argument of this function. The type et in the subscript of g tells us that this argument can be any object in the domain D_{et}.

The re-occurrence of 'g' after the dot tells us that the function we define in (58) returns the value of its argument.

Lambda Notation - summary

Lambda notation: *When writing "$\lambda x_\tau.\varphi$", where τ is a type, we mean: "the function sending every element x of the domain D_τ to φ".*

Function application with Lambda's

$(\lambda g_{et}.g)(\mathbf{tall}_{et})$
$= \mathbf{tall}$

Another example:

$\text{SUCC}(x) = x + 1$

$\text{SUCC}(22) = 22 + 1$ $\quad$ $(\lambda x_n.x + 1)(22)$ = $22 + 1$

Function application with lambda terms: *The result* $(\lambda x_\tau.\varphi)(a_\tau)$ *of applying a function described by a lambda term* $\lambda x_\tau.\varphi$ *to an argument* a_τ*, is equal to the value of the expression* φ*, with all occurrences of* x *replaced by* a*.*

Reflexives in object position

Tina praised herself

$$\mathbf{praise}_{e(et)}(\mathbf{tina}_e)(\mathbf{tina})$$

[[herself]] = **tina** ???

$$(\text{HERSELF}_{(e(et))(et)}(\mathbf{praise}_{e(et)}))(\mathbf{tina}_e)$$

$$= \mathbf{praise}(\mathbf{tina})(\mathbf{tina})$$

Generalizing:

For all functions R of the domain $D_{e(et)}$, for all entities x of the domain D_e:

$$(\text{HERSELF}_{(e(et))(et)}(R))(x) = R(x)(x)$$

Reflexives in object position (cont.)

For all functions R of the domain $D_{e(et)}$, for all entities x of the domain D_e:
$(\text{HERSELF}_{(e(et))(et)}(R))(x) = R(x)(x)$

$\text{HERSELF}_{(e(et))(et)}$ is
the function sending every element R of the domain $D_{e(et)}$ to the function sending every element x of the domain D_e to $R(x)(x)$.

$\text{HERSELF}_{(e(et))(et)}$
$= \lambda R_{e(et)}$.the function sending every element x of the domain D_e to $R(x)(x)$

$\text{HERSELF}_{(e(et))(et)}$
$= \lambda R_{e(et)}.(\lambda x_e.R(x)(x))$ = $\lambda R_{e(et)}.\lambda x_e.R(x)(x)$

Verifying the derivation

$(\textsc{herself}_{(e(et))(et)}(\mathbf{praise}_{e(et)}))(\mathbf{tina}_e)$	▷ compositional analysis of structure (63)
$= ((\lambda R_{e(et)}.\lambda x_e.R(x)(x))(\mathbf{praise}))(\mathbf{tina})$	▷ definition (70) of HERSELF
$= (\lambda x_e.\mathbf{praise}(x)(x))(\mathbf{tina})$	▷ applying HERSELF to the argument $\mathbf{praise}$
$= \mathbf{praise}(\mathbf{tina})(\mathbf{tina})$	▷ applying $(\textsc{herself}(\mathbf{praise}))$ to the argument $\mathbf{tina}$

What have we learnt here?

- A useful notation for functions
- A useful rule for simplifying notation under function application

Restricting Denotations

Expressing NOT in lambda's

Tina [is [not tall]]

NOT is the $(et)(et)$ function sending every et function g to the et function $\textsc{not}(g)$ that satisfies for every entity x:

$$(\textsc{not}(g))(x) = \begin{cases} 1 & \text{if } g(x) = 0 \\ 0 & \text{if } g(x) = 1 \end{cases}$$

$\sim \;=\; \lambda x_t.1 - x$

$\textsc{not} \;=\; \lambda g_{et}.\lambda x_e.\sim(g(x))$

$(\textsc{is}_{(et)(et)}(\textsc{not}_{(et)(et)}(\mathbf{tall}_{et})))(\mathbf{tina}_e)$	▷ compositional analysis of structure (37)
$= ((\lambda g_{et}.g)(\textsc{not}(\mathbf{tall})))(\mathbf{tina})$	▷ definition of IS as identity function
$= (\textsc{not}(\mathbf{tall}))(\mathbf{tina})$	▷ applying identity function to $\textsc{not}(\mathbf{tall})$
$= ((\lambda g_{et}.\lambda x_e.\sim(g(x)))(\mathbf{tall}))(\mathbf{tina})$	▷ definition (55) of NOT
$= ((\lambda x.\sim(\mathbf{tall}(x))))(\mathbf{tina})$	▷ applying definition of NOT to $\mathbf{tall}$
$= \sim(\mathbf{tall}(\mathbf{tina}))$	▷ application to $\mathbf{tina}$

Expressing ANDs in lambda's

[Tina [is tall]] [and [Tina [is thin]]]

For any two truth-values x and y: the truth-value $x \wedge y$ is $x \cdot y$, the multiplication of x by y.

$$\text{AND}^{t} = \lambda x_t . \lambda y_t . y \wedge x$$

Tina [is [tall [and thin]]]

For every two functions f_A and f_B in D_{et}, characterizing the subsets A and B of D_e: $(\text{AND}(f_A))(f_B)$ is defined as the function $f_{A \cap B}$, characterizing the intersection of A and B.

$$\text{AND}^{et} = \lambda f_{et} . \lambda g_{et} . \lambda x_e . g(x) \wedge f(x)$$

Attributive adjectives (1) - Intersective

Tina is a *tall* woman; the *tall* engineer visited us; I met five *tall* astronomers.

Tina is a Chinese pianist ⇔ Tina is Chinese and Tina is a pianist.

My doctor has a white Volkswagen ⇔ My doctor's Volkswagen is white.

Mary saw three carnivorous animals ⇔ Three animals that Mary saw are carnivorous.

Tina [is [a pianist]]

$\textsc{a}_{(et)(et)} = \textsc{is} = \lambda g_{et}.g$

$(\textsc{is}(\textsc{a}(\mathbf{pianist})))(\mathbf{tina})$

$= \mathbf{pianist}(\mathbf{tina})$

Attributive adjectives (2) - Intersective

Tina [is [a [Chinese pianist]]]

$$\textbf{chinese}^{\text{mod}}_{(et)(et)} = \lambda f_{et}.\lambda x_e.\textbf{chinese}(x) \wedge f(x)$$

For any two truth-values x and y: the truth-value $x \wedge y$ is $x \cdot y$, the multiplication of x by y.

$(\textsc{is}(\textsc{a}(\textbf{chinese}^{\text{mod}}(\textbf{pianist}))))(\textbf{tina})$	▹ compositional analysis of (73)
$= (\textbf{chinese}^{\text{mod}}(\textbf{pianist}))(\textbf{tina})$	▹ applying IS and A (identity functions)
$= ((\lambda f_{et}.\lambda x_e.\textbf{chinese}(x) \wedge f(x))(\textbf{pianist}))(\textbf{tina})$	▹ definition (74) of **chinese**$^{\text{mod}}$
$= (\lambda x_e.\textbf{chinese}(x) \wedge \textbf{pianist}(x))(\textbf{tina})$	▹ applying modificational denotation to **pianist**
$= \textbf{chinese}(\textbf{tina}) \wedge \textbf{pianist}(\textbf{tina})$	▹ applying result to **tina**

Conclusion: with adjectives like *Chinese* the attributive (*et*)(*et*) denotation can be systematically derived from the predicative *et* denotation.

Note: this is not the case with all adjectives (cf. *skillful*).

Attributive adjectives (3) - Subsective

Jan is a <u>Chinese</u> surgeon & Jan is a violinist

➔ *Jan is a <u>Chinese</u> violinist*

Jan is a <u>skillful</u> surgeon & Jan is a violinist

➔ *Jan is a <u>skillful</u> violinist*

Conclusion 1: *skillful* is not intersective.

However, skillful has a weaker property, which we call <u>restrictivity</u>.

Jan is a <u>skillful</u> surgeon

➔ *Jan is a <u>surgeon</u>*

Attributive adjectives (4) - Subsective

Formally: ***M is subsective (or "restrictive") if for every set of entities A,*** **M(A) $\subseteq$ A.**

Conclusion 2: *skillful* is subsective.

In Lambdas:

skillful$_{(et)(et)}$ **=** $\lambda A.\ \lambda y.$ **(skillful1**$_{(et)(et)}$**(A))(y)** $\wedge$ **A(y)**

Summary – restrictions on denotations

Constant, Combinatorial:

IS, A, HERSELF

Constant, Logical:

ANDs, NOTs

Arbitrary:

tina, smile, praise, pianist, chinese (predicative use)

Logical operator on arbitrary:

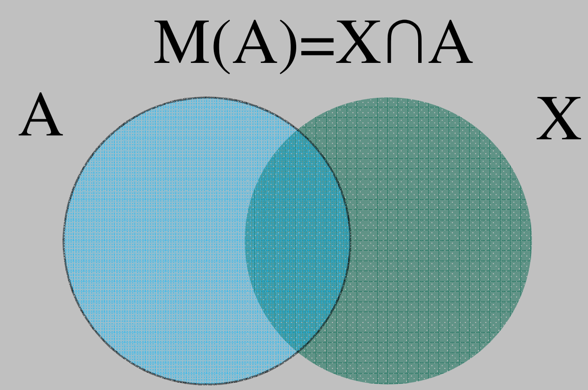
chinese*mod*, skillful*mod* (attributive use)

Further: bachelor ➔ unmarried …

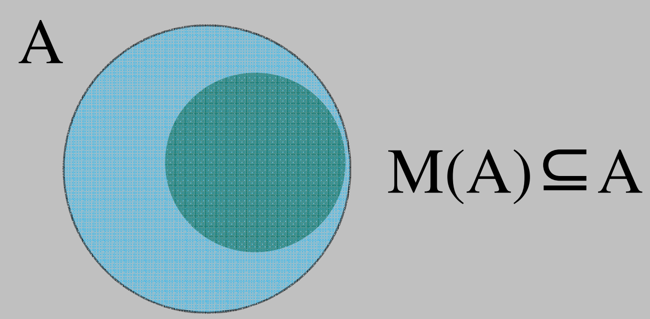
More on the classifictaion of adjectives

I - Subsective functions

Intersective functions

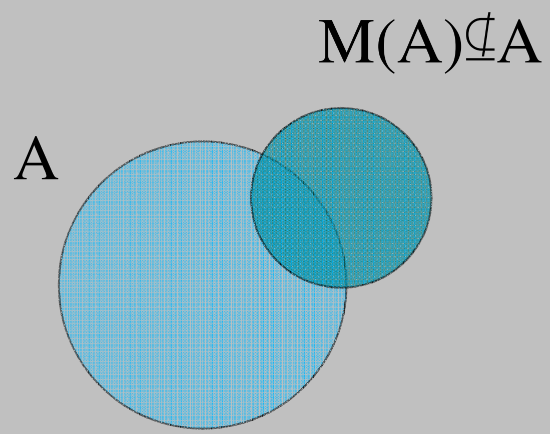


Subsective functions



Note: any intersective function is subsective

II - Non-Subsective functions



Non-subsective adjectives

Jan is an alledged surgeon

=/=> *Jan is a* *<u>surgeon</u>*

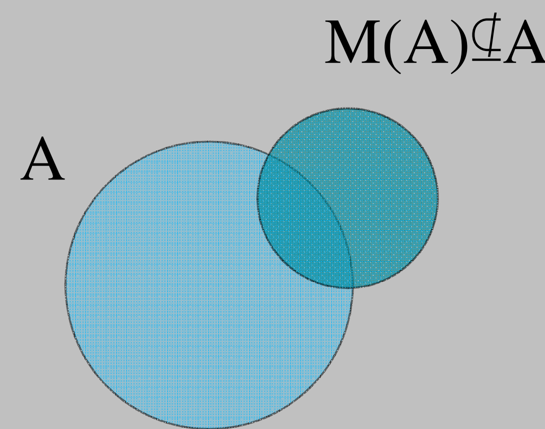
Conclusion: *alleged* is not subsective.

More examples (Partee):

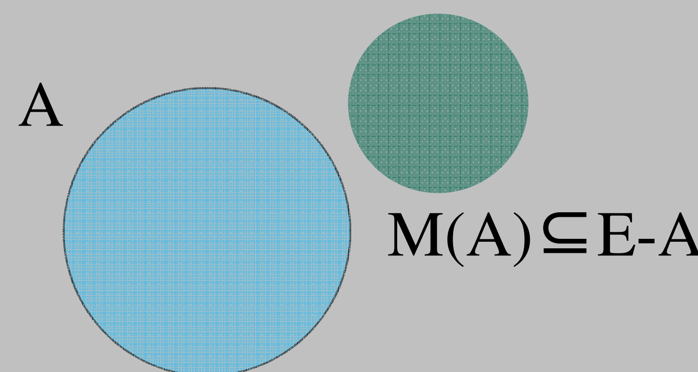
potential, alleged, arguable, likely, predicted, putative, questionable, disputed.

III - Non-subsective but co-subsective

Non-subsective (general)



Co-subsective



Note: any (non-trivial) co-subsective function is non-subsective

Co-subsective adjectives

This is a false diamond
=/=> *This is a diamond*

Conclusion 1: *false* is not subsective.

However:
This is a false diamond
➔ *This is not a diamond*

Conclusion 2: *false* is co-subsective.

Formally: M is *co-subsective* (or "privative") if for every set of entities A, M(A) $\subseteq$ E-A.

More co-subsective adjectives (Partee)

non-subsective and privative: *wouldbe, past, spurious, imaginary, fictitious, fabricated* (in one sense), *mythical* (maybe debatable); there are prefixes with this property too, like *ex, pseudo, non*.

Note:
John is an alleged criminal, and indeed he is a criminal.
Conclusion: *alleged* is not co-subsective.

Adjectives - summary

- **Subsective**
  - **Intersective**
    - **Dutch**
  - **¬ Intersective**
    - **skillful**
- **¬ Subsective**
  - **Co-subsective**
    - **false**
  - **¬ Co-subsective**
    - **alleged**

Intersective ➔ **Subsective**

Co-restrictive ➔ **¬Subsective** (ignoring trivial cases)

These properties can be generalized (and studied) for modifiers of other types besides (*et*)(*et*).

Note on (non)extensionality (1)

Jan is a <u>skillful</u> surgeon & Jan is a violinist

➔ *Jan is a <u>skillful</u> violinist*

Conclusions we had: *skillful* is not intersective (but subsective).

Adjectives like *skillful* lack another property, which we <u>extensionality</u>.

Jan is a <u>Dutch</u> surgeon &

the surgeons are the same as the lawyers

➔ *Jan is a Dutch <u>lawyer</u>*

Informally: An adjective ADJ is called <u>extensional</u> if for every two nouns N1 and N2 that denote the same set of entities, we have: [[ADJ N1]] = [[ADJ N2]].

The adjective *Dutch* is extensional, but *skillful* is not:

Jan is a <u>skillful</u> surgeon &

the surgeons are the same as the lawyers

=/=> *Jan is a skillful <u>lawyer</u>*

Note on (non)extensionality (2)

It has been postulated that the non-intersectivity of many adjectives is derived from their non-extensionality, but the ways the two properties are related is unclear.

Session 3:
Generalized Quantifiers

Quantifiers in different domains

John *rarely/usually* eats meat.

We are *close to/far from* Beijing.

There is *little/a lot of* work to do today.

Many/few people admire Richard Wagner.

We focus on quantificational *NPs*:
NPs containing determiners like *many, few, every, some, most etc.*

Quantifiers – main claims

In order to describe the meaning of NPs with *determiners* (*every, some, most* etc.), we should let such NPs denote sets of subsets of E – type $(et)t$.

The same type is needed for describing *NP coordination* in a general way.

Montague's hypothesis about the matching between syntactic categories and semantic types leads us to adopt a uniform type for all NPs.

Some hard syntactic questions can then be given interesting semantic answers.

Keenan's typology of determiners (1)

Lexical Dets
every, each, all, some, a, no, several, neither, most, the, both, this, my, these, John's, ten, a few, a dozen, many, few

Cardinal Dets
exactly/approximately/more than/fewer than/at most/only ten, infinitely many, two dozen, between five and ten, just finitely many, an even/odd number of, a large number of

Approximative Dets
approximately/about/nearly/around fifty, almost all/no, hardly any, practically no

Definite Dets
the, that, this, these, my, his, John's, the ten, these ten, John's ten

Exception Dets
all but ten, all but at most ten, every...but John, no...but Mary,

Bounding Dets
exactly ten, between five and ten, most but not all, exactly half the, (just) one...in ten, only SOME (= some but not all; upper case = contrastive stress), just the LIBERAL, only JOHN's

Possessive Dets
my, John's, no student's, either John's or Mary's, neither John's nor Mary's

Value Judgment Dets
too many, a few too many, (not) enough, surprisingly few, ?many, ?few

Proportionality Dets
exactly half the/John's, two out of three, (not) one...in ten, less than half the/John's, a third of the/John's,

Keenan's typology (2)

Partitive Dets
most/two/none/only some of the/John's, more of John's than of Mary's, not more than two of the ten

Negated Dets
not every, not all, not a (single), not more than ten, not more than half, not very many, not quite enough, not over a hundred, not one of John's

Conjoined Dets
at least two but not more than ten, most but not all, either fewer than ten or else more than a hundred, both John's and Mary's, at least a third and at most two thirds of the, neither fewer than ten nor more than a hundred

Adjectively Restricted Dets
John's biggest, more male than female, most male and all female, the last...John visited, the first ...to set foot on the Moon, the easiest...to clean, whatever...are in the cupboard

Function-Argument flip-flop
(an argument between FLIP and FLOP)

FLIP: **NP:*e* + VP:*et* = S:*t***
(subject as argument)

***FLOP*:** *But can all those NPs denote entities?* ☹

NP:(*et*)*t* + VP:*et* = S:*t*
(subject as function)

FLIP: NP denotes (*et*)*t* function??? ☹

***FLOP*:** *Yes! let's do some work on it!* ☺

Generalized Quantifiers - example

(1) Every man ran.

Let *every man* denote a *set of sets*: the set of subsets of E that include the set of men:

(2) $\{B \subseteq E : \mathbf{man}' \subseteq B\}$

In type-theoretical terms: *every man* denotes an $(et)t$ function.
Application of this function to the VP denotation:

(3) $\mathbf{run}' \in \{B \subseteq E : \mathbf{man}' \subseteq B\}$
$\Leftrightarrow \mathbf{man}' \subseteq \mathbf{run}'$
"every member of the set $\mathbf{man}'$ is a member of the set $\mathbf{run}'$"

For instance, if $E = \{a, b, c, d\}$, $\mathbf{man}' = \{a, b\}$ and $\mathbf{run}' = \{a, b, c\}$, then:

$[\![\text{every man}]\!] = \{B \subseteq E : \mathbf{man}' \subseteq B\}$
$= \{B \subseteq \{a, b, c, d\} : \{a, b\} \subseteq B\}$
$= \{\{a, b\}, \{a, b, c\}, \{a, b, d\}, \{a, b, c, d\}\}$,
and thus $\mathbf{run}' = \{a, b, c\} \in [\![\text{every man}]\!]$.

GQs - more examples

(4) Some man ran.

(5) $\mathbf{run}' \in \{B \subseteq E : \mathbf{man}' \cap B \neq \emptyset\}$
$\Leftrightarrow \mathbf{man}' \cap \mathbf{run}' \neq \emptyset$
"there is an entity that is a member of both $\mathbf{man}'$ and $\mathbf{run}'$"

(6) No man ran.

(7) $\mathbf{run}' \in \{B \subseteq E : \mathbf{man}' \cap B = \emptyset\}$
$\Leftrightarrow \mathbf{man}' \cap \mathbf{run}' = \emptyset$

(8) *exactly five men*: $\{B \subseteq E : |\mathbf{man}' \cap B| = 5\}$

(9) *most men*: $\{B \subseteq E : |\mathbf{man}' \cap B| > |\mathbf{man}' \setminus B|\}$

GQs - definition

NP:(*et*)*t* + VP:*et* = S:*t*

(*et*)*t* functions ~= sets of sets of entities

Terminology: Any set $Q \subseteq \wp(E)$ (a set of subsets of E) is called a *generalized quantifier* (GQ) over E.

GQ Monotonicity

(10) a. Some man ran quickly $\Rightarrow$ Some man ran

b. Some man ran quickly $\not\Leftarrow$ Some man ran

(11) a. No man ran quickly $\not\Rightarrow$ No man ran

b. No man ran quickly $\Leftarrow$ No man ran

(12) a. Exactly five men ran quickly $\not\Rightarrow$ Exactly five men ran

b. Exactly five men ran quickly $\not\Leftarrow$ Exactly five men ran

How do we show non-monotonicity entailments?

Some man is called an *upward monotone* (mon↑) noun phrase.
No man is called a *downward monotone* (mon↓) noun phrase.
Exactly five men is called a *non-monotone* (neither mon↑ nor mon↓) noun phrase.

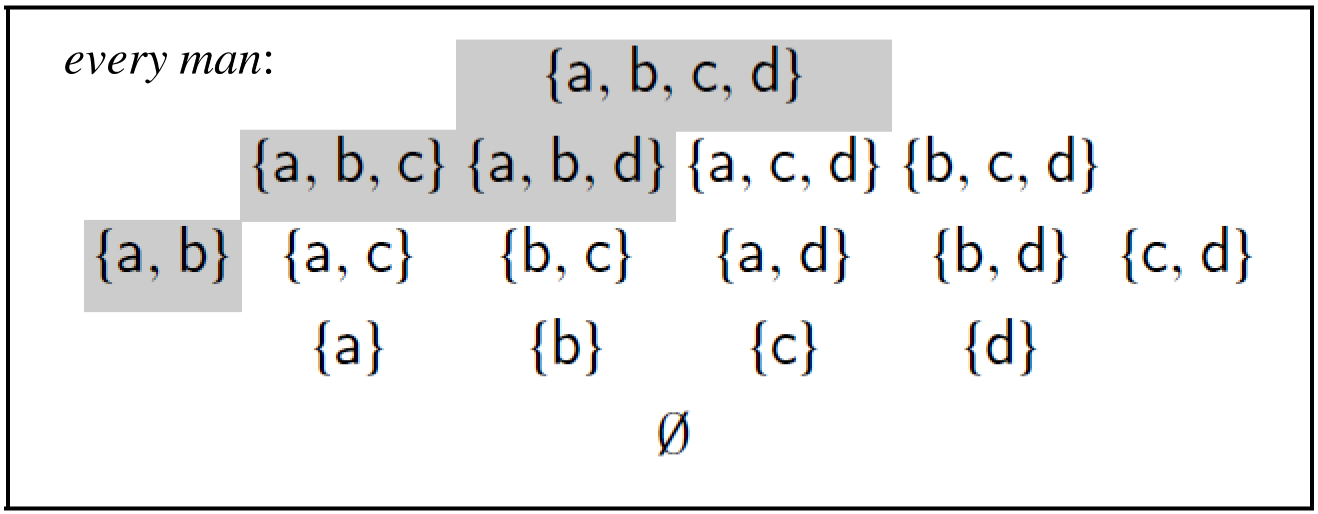
Definition 1 (quantifier monotonicity) *A generalized quantifier* $Q \in \wp(\wp(E))$ *is:*

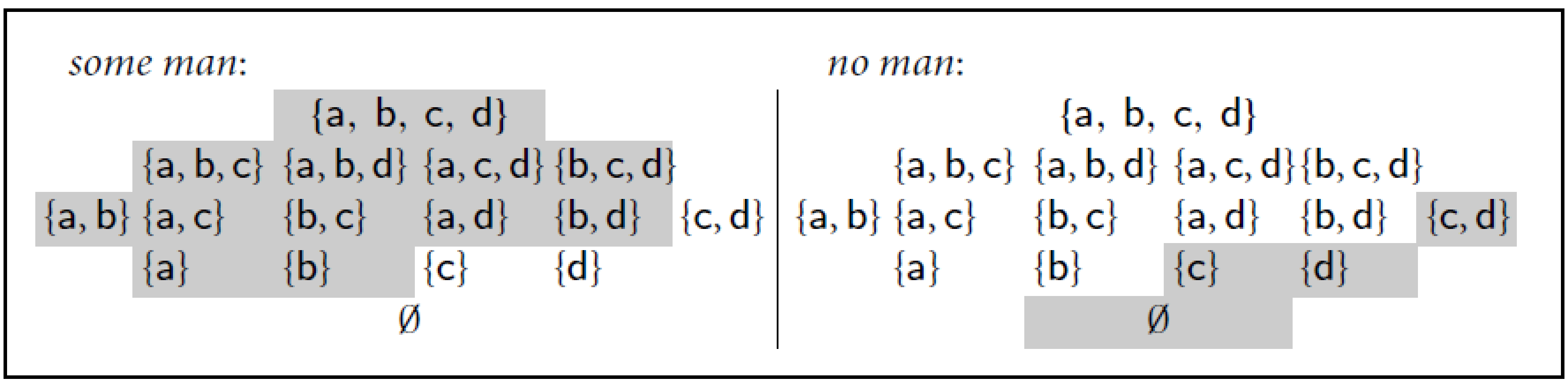
1. *mon↑ iff for all* $A \subseteq B \subseteq E$*: if* $A \in Q$ *then* $B \in Q$.
2. *mon↓ iff for all* $A \subseteq B \subseteq E$*: if* $B \in Q$ *then* $A \in Q$.

Quantifiers and monotonicity - summary

	Noun phrase	Generalized quantifier	Monotonicity
I	*every man*	$\{B \subseteq E : \mathbf{man}^* \subseteq B\}$	MON↑
II	*some man*	$\{B \subseteq E : \mathbf{man}^* \cap B \neq \emptyset\}$	MON↑
	no man	$\{B \subseteq E : \mathbf{man}^* \cap B = \emptyset\}$	MON↓
	exactly one man	$\{B \subseteq E : \mid \mathbf{man}^* \cap B \mid = 1\}$	MON¬
	at least three men	$\{B \subseteq E : \mid \mathbf{man}^* \cap B \mid \geq 3\}$	MON↑
	fewer than five men	$\{B \subseteq E : \mid \mathbf{man}^* \cap B \mid < 5\}$	MON↓
	between six and eleven men	$\{B \subseteq E : 6 \leq \mid \mathbf{man}^* \cap B \mid \leq 11\}$	MON¬
III	*at least half the men*	$\{B \subseteq E : \mid \mathbf{man}^* \cap B \mid \geq 1/2 \cdot \mid \mathbf{man}^* \mid \}$	MON↑

Quantifiers in models





Back to proper names

(23) Tina ran.

(24) $\mathbf{run}' \in \{B \subseteq E : \mathbf{tina}' \in B\}$
"the set of runners is in the set of subsets of E that contain $\mathbf{tina}'$"
$\Leftrightarrow \mathbf{tina}' \in \mathbf{run}'$

NP Coordination (1)

Linguistic fact: Coordination applies freely to proper names and other NPs alike.

(25) Mary and/or John, neither Mary nor John, every woman or every man, most women and most men, many students but few teachers, one student and five teachers, the teacher and every student etc.

The denotation of these NPs is easily derived using GQs and the boolean treatment of coordination.

(26) a. Mary and John smiled.
$\mathbf{smile}' \in \{A \subseteq E : \mathbf{m}' \in A\} \cap \{A \subseteq E : \mathbf{j}' \in A\}$
$\Leftrightarrow \mathbf{m}' \in \mathbf{smile}' \wedge \mathbf{j}' \in \mathbf{smile}'$

b. Mary smiled and John smiled.
$\mathbf{smile}' \in \{A \subseteq E : \mathbf{m}' \in A\} \wedge \mathbf{smile}' \in \{A \subseteq E : \mathbf{j}' \in A\}$
$\Leftrightarrow \mathbf{m}' \in \mathbf{smile}' \wedge \mathbf{j}' \in \mathbf{smile}'$

NP Coordination (2)

(27) a. Mary or John smiled.

$$\mathbf{smile}' \in \{A \subseteq E : \mathbf{m}' \in A\} \cup \{A \subseteq E : \mathbf{j}' \in A\}$$
$$\Leftrightarrow \mathbf{m}' \in \mathbf{smile}' \vee \mathbf{j}' \in \mathbf{smile}'$$

b. Mary smiled or John smiled.

$$\mathbf{smile}' \in \{A \subseteq E : \mathbf{m}' \in A\} \vee \mathbf{smile}' \in \{A \subseteq E : \mathbf{j}' \in A\}$$
$$\Leftrightarrow \mathbf{m}' \in \mathbf{smile}' \vee \mathbf{j}' \in \mathbf{smile}'$$

(28) a. Neither Mary nor John smiled.

$$\mathbf{smile}' \in \overline{\{A \subseteq E : \mathbf{m}' \in A\}} \cap \overline{\{A \subseteq E : \mathbf{j}' \in A\}}$$
$$\Leftrightarrow \mathbf{m}' \notin \mathbf{smile}' \wedge \mathbf{j}' \notin \mathbf{smile}'$$

b. Mary didn't smile and John didn't smile.

$$\overline{\mathbf{smile}'} \in \{A \subseteq E : \mathbf{m}' \in A\} \wedge \overline{\mathbf{smile}'} \in \{A \subseteq E : \mathbf{j}' \in A\}$$
$$\Leftrightarrow \mathbf{m}' \notin \mathbf{smile}' \wedge \mathbf{j}' \notin \mathbf{smile}'$$

Conjunction Reduction???

Quantifiers with VP coordination

This is in agreement with the old (and ill-defined) transformational rule of *conjunction reduction* (CR). However, consider the following:

(29) a. NP sang and danced $\not\Leftrightarrow$ NP sang and NP danced
NP = some man, no man, not every man, Mary or John, at least/most five women, exactly five women, most women

b. NP sang and danced $\Leftrightarrow$ NP sang and NP danced
NP = every man, Mary, Mary and John

(30) a. NP sang or danced $\not\Leftrightarrow$ NP sang or NP danced
NP = every man, no man, not every man, Mary and John, at least/most five women, exactly five women, most women

b. NP sang or danced $\Leftrightarrow$ NP sang or NP danced
NP = some man, Mary, Mary or John

Against conjunction reduction

(31) Some man danced and sang.
$\mathbf{dance}' \cap \mathbf{sing}' \in \{A \subseteq E : \mathbf{man}' \cap A \neq \emptyset\}$
$\Leftrightarrow \mathbf{man}' \cap (\mathbf{dance}' \cap \mathbf{sing}') \neq \emptyset$

This can be false when both $\mathbf{man}' \cap \mathbf{dance}'$ and $\mathbf{man}' \cap \mathbf{sing}'$ are non-empty.

(32) Every man danced and sang.
$\mathbf{dance}' \cap \mathbf{sing}' \in \{A \subseteq E : \mathbf{man}' \subseteq A\}$
$\Leftrightarrow \mathbf{man}' \subseteq \mathbf{dance}' \cap \mathbf{sing}'$
This holds *iff* $\mathbf{man}' \subseteq \mathbf{dance}'$ and $\mathbf{man}' \subseteq \mathbf{sing}'$.

Conclusion: The boolean semantics of GQs is much more fine-grained than any syntactic account of the semantics of coordination.

Determiner expressions

$X + et = (et)t$ What is *X*?

Determiners like the ones considered above denote functions from sets to GQs:

(13) $\mathbf{every}'(A) \;=\; \{B \subseteq E : A \subseteq B\}$

Thus, determiners denote functions of type $(et)((et)t)$.
Alternatively, we can view such functions as *relations* between subsets of E:

(14) $\mathbf{every}'(A)(B)$ iff $A \subseteq B$

(15) $\mathbf{some}'(A)(B)$ iff $A \cap B \neq \emptyset$

(16) $\mathbf{most}'(A)(B)$ iff $|A \cap B| > |A \setminus B|$

Terminology: Any set $D \subseteq \wp(E) \times \wp(E)$ (a relation between subsets of E) is called a *determiner function* over E.

Determiner Monotonicity

[Every [tall man]] [ran quickly] $\Rightarrow$ [Every [tall man]] ran

[Every [tall man from Japan]] [ran quickly]
$\Rightarrow$ [Every [tall man from Japan]] ran

[Every [tall man from Japan who sings the blues]] [ran quickly]
$\Rightarrow$ [Every [tall man from Japan who sings the blues]] ran

[Every N] [ran quickly] $\Rightarrow$ [Every N] ran

A determiner relation $\mathbf{D}$ *over* E *is called* **right upward monotone** (MON$\uparrow$) *if and only if for all* $A \subseteq E$ *and* $B_1 \subseteq B_2 \subseteq E$: $\mathbf{D}(A, B_1) \Rightarrow \mathbf{D}(A, B_2)$.

A determiner relation $\mathbf{D}$ *over* E *is called* **right downward monotone** (MON$\downarrow$) *if and only if for all* $A \subseteq E$ *and* $B_1 \subseteq B_2 \subseteq E$: $\mathbf{D}(A, B_2) \Rightarrow \mathbf{D}(A, B_1)$.

Determiner Monotonicity – left argument

[Every man] ran $\Rightarrow$ [Every [tall man]] ran.

Suppose that the relation EVERY(A_1, B) holds.
Then $A_1 \subseteq B$.
By our assumption $A_2 \subseteq A_1$, we also have $A_2 \subseteq B$.
Thus, the relation EVERY(A_2, B) holds.

Some man ran $\not\Rightarrow$ Some tall man ran.
Some tall man ran $\Rightarrow$ Some man ran.

Exactly one man ran $\not\Rightarrow$ Exactly one tall man ran.
Exactly one tall man ran $\not\Rightarrow$ Exactly one man ran.

Determiners - summary

Determiner relations			Monotonicity
EVERY (A, B)	$\Leftrightarrow$	$A \subseteq B$	$\downarrow$MON$\uparrow$
SOME (A, B)	$\Leftrightarrow$	$A \cap B \neq \emptyset$	$\uparrow$MON$\uparrow$
NO (A, B)	$\Leftrightarrow$	$A \cap B = \emptyset$	$\downarrow$MON$\downarrow$
EXACTLY_1 (A, B)	$\Leftrightarrow$	$\lvert A \cap B \rvert = 1$	$\neg$MON$\neg$
AT_LEAST_3 (A, B)	$\Leftrightarrow$	$\lvert A \cap B \rvert \geq 3$	$\uparrow$MON$\uparrow$
FEWER_THAN_5 (A, B)	$\Leftrightarrow$	$\lvert A \cap B \rvert < 5$	$\downarrow$MON$\downarrow$
BETWEEN_6_AND_11 (A, B)	$\Leftrightarrow$	$6 \leq \lvert A \cap B \rvert \leq 11$	$\neg$MON$\neg$
AT_LEAST_HALF (A, B)	$\Leftrightarrow$	$\lvert A \cap B \rvert \geq 1/2 \cdot \lvert A \rvert$	$\neg$MON$\uparrow$

Negative polarity items (1)

(33) a. John hasn't <u>ever</u> been to Moscow.
b. *John has <u>ever</u> been to Moscow.

(34) a. John didn't see <u>any</u> birds on the tree.
b. *John saw <u>any</u> birds on the tree.

(35) a. No student here has <u>ever</u> been to Moscow.
b. *Some/every student here has <u>ever</u> been to Moscow.

(36) a. Neither John nor Mary saw <u>any</u> birds on the tree.
b. *Either John or Mary saw <u>any</u> birds on the tree.

(37) a. None of John's students has <u>ever</u> been to Moscow.
b. *One of John's students has <u>ever</u> been to Moscow.

(38) a. Not a single student here has <u>ever</u> been to Moscow.
b. *A single student here has <u>ever</u> been to Moscow.

(39) a. Not more than five students here have <u>ever</u> been to Moscow.
b. *More than five students here have <u>ever</u> been to Moscow.

(40) a. Fewer than five students here have <u>ever</u> been to Moscow.
b. *More than five students here have <u>ever</u> been to Moscow.

Negative polarity items (2)

(41) a. At most four students here have ever been to Moscow.

b. *At least four students here have ever been to Moscow.

(42) a. Less than half the students here have ever been to Moscow.

b. *More than half the students here have ever been to Moscow.

(43) a. Neither any students nor any teachers attended the meeting.

b. *Either any students or any teachers attended the meeting.

(44) a. John neither praised nor criticized any student.

b. *John either praised or criticized any student.

(45) a. Every/no/at most one student who has ever been to Moscow knows about the weather there.

b. *Some/at least one student who has ever been to Moscow knows about the weather there.

(46) If John ever goes to Moscow he will know about the weather there.

The Ladusaw-Fauconnier Generalization: Negative polarity items occur within arguments of monotonic decreasing functions but not within arguments of monotonic increasing functions.

Determiner Conservativity

(20) Every man ran $\Leftrightarrow$ Every man is a man who ran

(21) Some man ran $\Leftrightarrow$ Some man is a man who ran

(22) Exactly five men ran $\Leftrightarrow$ Exactly five men are men who ran

... and so on (allegedly) for all determiners!

Definition 3 (conservativity) *A determiner function* $D \subseteq \wp(E) \times \wp(E)$ *is called* conservative *iff for all* $A, B \subseteq E$*:* $D(A)(B) \Leftrightarrow D(A)(A \cap B)$.

Hypothesized universal: All natural language determiners (simple and complex) denote conservative determiner functions.

Formal Semantics of Natural Language

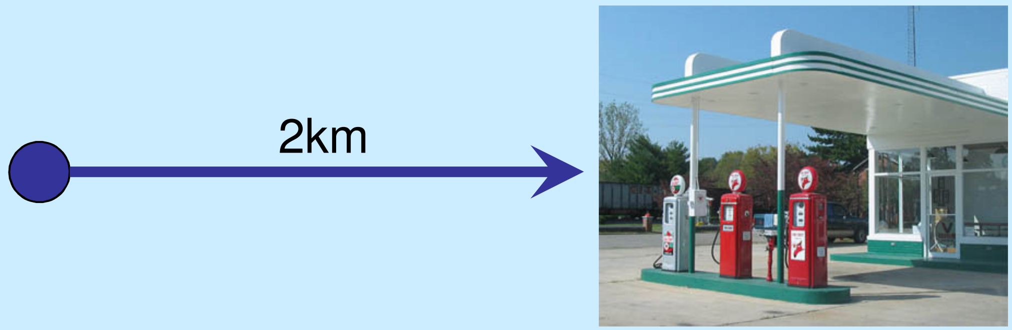
ESSLLI2016, Bolzano - Bolzen, 22-26 August 2016

Yoad Winter, Utrecht University, The Netherlands

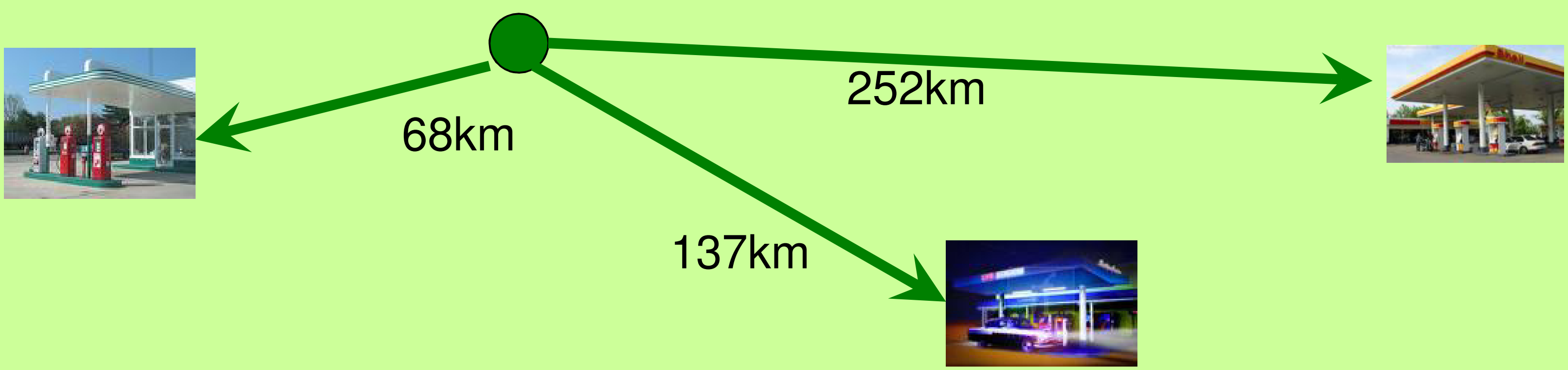
Session 4:
Spatial Semantics

The Gas Station Puzzle (Iatriduo 2003)

Michael is close to a gas station. "$\exists$"



Michael is far from a gas station. "$\forall$"



Analyzing ``a gas station´´ --
Background on indefinites

Indefinites as predicates:

Michael is *a driver*.

London is *a city*.

Indefinites as arguments:

A driver showed me his car.

A city was built here.

Partee (1987):

Predicate indefinites are type *et*: [[*a driver*]] = **driver**

Argument quantifiers are type (*et*)*t*: $\lambda P_{et}.\exists x_e.\mathbf{driver}(x) \wedge P(x)$

Analyzing ``far from/close to´´ --
Background on locatives

Locatives are two-place predicates:

Michael is *far from/close to* London.

$$\mathbf{far\_from}(\mathbf{m}, \mathbf{l})$$

Locatives are applied to entities that have <u>locations</u>:

$$\textsc{Loc}(\mathbf{m}) = m; \quad \textsc{Loc}(\mathbf{l}) = L$$

A deeper semantics as two-place <u>locative</u> predicates:

$$\text{FAR\_FROM}(\textsc{Loc}(\mathbf{m}), \textsc{Loc}(\mathbf{l}))$$

$$= \text{FAR\_FROM}(m, L)$$

Basic Account

The gas station puzzle reflects the *set denotation* of indefinites, together with the part-whole properties of spatial prepositions.

(1) Michael is **close to** London ←→

For **some** part of London *x*, Michael is close to *x*.

(2) Michael is **far from** London ←→

For **every** part of London *x*, Michael is far from *x*.

The same principles about locatives that account for (1) and (2) will be used to account for the gas station puzzle.

"A gas station" – analyzed as an *et* predicate; distances are measured from the union location of the members of this predicate.

$$\text{FAR\_FROM}\left(m, \bigcup\left\{\text{LOC}(x) : x \in \mathbf{gs}\right\}\right)$$

Background - Types of locatives

Topological locatives:

The car is *in* the garage.

The camp is *outside* the city.

Distal locatives:

The car is *far from* the garage.

The camp is *close to* the city.

Beijing is *1,318km from* Shanghai.

Projective locatives:

The car is *left of* the garage.

The bird is *close to* the house.

Topological Locatives

INSIDE: *in*, *inside (of)*, *within*

OUTSIDE: *out of*, *outside (of)*, *without*

a. The visitor is inside the building.
b. The visitor is outside the building.

a. The visitor is inside SOME$_\exists$ part of the building.
b. The visitor is outside EVERY$_\forall$ part of the building.

For all points x and regions A:

$$\text{INSIDE}(x, A) \Leftrightarrow x \in A$$

$$\text{OUTSIDE}(x, A) \Leftrightarrow x \notin A$$

Topological Locatives

INSIDE: *in*, *inside (of)*, *within*

OUTSIDE: *out of*, *outside (of)*, *without*

a. The visitor is inside the building.
b. The visitor is outside the building.

a. The visitor is inside SOME$_{\exists}$ part of the building.
b. The visitor is outside EVERY$_{\forall}$ part of the building.

Locative indefinites:

Every vehicle or trailer which is parked *outside of a garage* shall display license plates with current registration tabs. ***outside every garage***

I personally find the planets that formed *outside of a star system* more fascinating than ejecta. ***outside every star system***

One-third of the funded proposals shall serve schools *within a Metropolitan County*, and at least one-third shall serve schools *outside of a Metropolitan County*. ***outside every MC***

Topological Locatives - formally

a. The school is within a Metropolitan County.

b. The school is outside of a Metropolitan County.

$$\textsc{inside}(x, \cup\mathcal{A}) \quad \Leftrightarrow \quad \exists A \in \mathcal{A}.\textsc{inside}(x, A)$$

$$\textsc{outside}(x, \cup\mathcal{A}) \quad \Leftrightarrow \quad \forall A \in \mathcal{A}.\textsc{outside}(x, A)$$

a. $\textsc{inside}(s, \textsc{loc}(\textit{MC}))$

$\Leftrightarrow \textsc{inside}(s, \bigcup\{\textsc{loc}(x) : x \in \mathbf{mc}\})$

$\Leftrightarrow \exists x \in \mathbf{mc}.\textsc{inside}(s, \textsc{loc}(x))$

b. $\textsc{outside}(s, \textsc{loc}(\textit{MC}))$

$\Leftrightarrow \textsc{outside}(s, \bigcup\{\textsc{loc}(x) : x \in \mathbf{mc}\})$

$\Leftrightarrow \forall x \in \mathbf{mc}.\textsc{outside}(s, \textsc{loc}(x))$

Distal Locatives

Upward monotone, downward monotone and non-monotone distal locatives:

$\text{DIST}_{M\uparrow}$: *far from, away from, more than/at least 20km (away) from*

$\text{DIST}_{M\downarrow}$: *close to, near (to), less than/at most 20km (away) from*

$\text{DIST}_{M\neg}$: *exactly 20km (away) from, between 20km and 30km (away) from*

a. Michael is at least 20km from London.
b. Michael is at most 20km from London.

a. Michael is at least 20km from $\text{EVERY}_{\forall}$ part of London.
b. Michael is at most 20km from $\text{SOME}_{\exists}$ part of London.

Locative indefinites:

a. Michael is at least 20km from a gas station.
b. Michael is at most 20km from a gas station.

a. Michael is at least 20km from $\text{EVERY}_{\forall}$ gas station.
b. Michael is at most 20km from $\text{SOME}_{\exists}$ gas station.

Measuring distances

Definition 2 (*metric function*) Let M be a non-empty set, and let d be a function from the cartesian product $M \times M$ to non-negative real numbers in $\mathcal{R}$. The function d is called a *metric function* if it satisfies the following requirements for all elements $x, y \in M$:

$$d(x, y) = d(y, x) \qquad \text{(symmetry)}$$

$$d(x, y) + d(y, z) \geq d(x, z) \qquad \text{(triangle inequality)}$$

$$d(x, y) = 0 \textit{ iff } x = y \qquad \text{(identity of indiscernibles)}$$

Definition 3 (*distance*) For every non-empty closed region $A \subseteq M$ and a point $x \in M$ not in A, the *distance* between x and A is defined by:

$$\mathbf{d}(x, A) = min(\{d(x, y) : y \in A\}).$$

$$\text{FAR\_FROM}(x, A) \Leftrightarrow \mathbf{d}(x, A) > r$$

$$\text{CLOSE\_TO}(x, A) \Leftrightarrow \mathbf{d}(x, A) < r$$

"Far from" vs. "Close to"

$$\text{FAR\_FROM}(x, \cup\mathcal{A}) \quad \Leftrightarrow \forall A \in \mathcal{A}.\text{FAR\_FROM}(x, A)$$

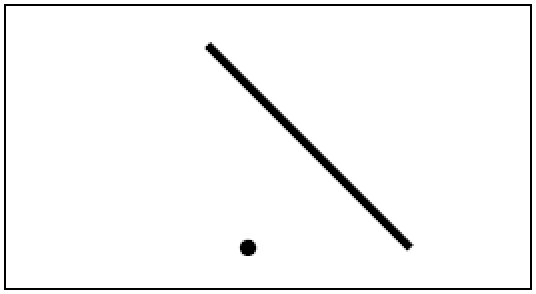
$$\text{CLOSE\_TO}(x, \cup\mathcal{A}) \quad \Leftrightarrow \exists A \in \mathcal{A}.\text{CLOSE\_TO}(x, A)$$

Projective Locatives

above, behind, north of, left of

The dot is left of the line.

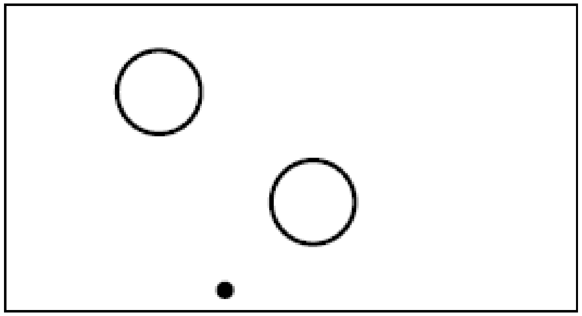
The dot is right of the line.



Non-existential locative indefinites:

The dot is left of a circle.

The dot is right of a circle.



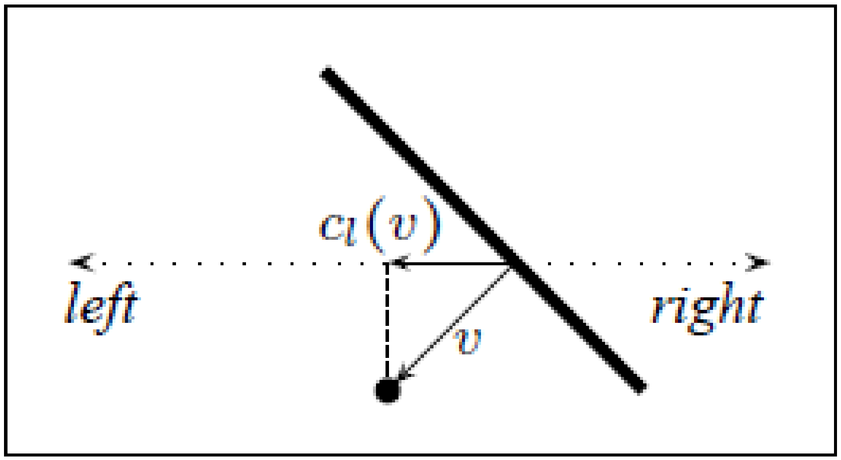
Experimental work with Robert Grimm, Eva Poortman and Choonkyu Lee (SALT 2014)

Projective Locatives – formally

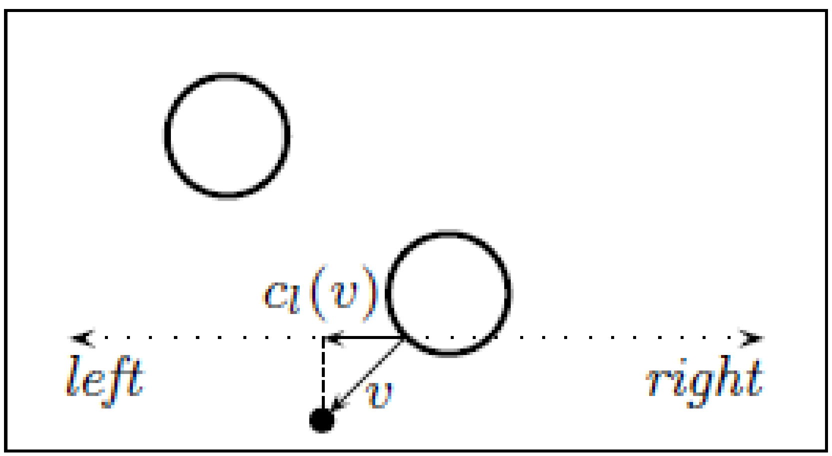
For any region A and point $x \notin A$:

$\text{LEFT\_OF}(x, A) \quad \Leftrightarrow$

the shortest vector from A to x has a non-zero 'left of' component



the dot is left of the line



the dot is left of a circle

Summary

Topological (*inside, outside*), **distal** (*far from, close to*), and **projective** (*behind, above*) locatives.

For all these locatives, we see pseudo-quantificational effect w.r.t. part-whole relations:

> ***close to London*** = *close to* <u>*some*</u> *part of London*
>
> ***far from London*** = *far from* <u>*every*</u> *part of London*

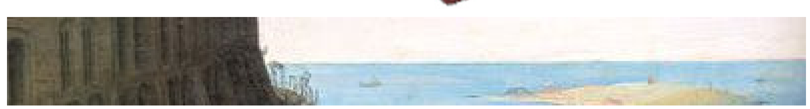
We explain the similarity between this behavior and "strange" effects with locative indefinites:

> ***close to a gas station*** = *close to* <u>*some*</u> *gas station*
>
> ***far from a gas station*** = *far from* <u>*every*</u> *gas station*

Thank you!

Sela Mador-Haim

Joost Zwarts



Modified Locatives

With distal modifiers: *far outside of, 10km north of, deep under*

With projective modifiers: *diagonally above, straight in front of, right beneath*

Modified *outside* (topological use) – pseudo-intersective:

The hotel is *far outside* the city center.

$$\text{FAR\_OUTSIDE}(x, A) \Leftrightarrow \text{FAR\_FROM}(x, A) \wedge \text{OUTSIDE}(x, A)$$

Non-existential locative indefinites:

a. This scene shows Roamer ships encountering a huge, derelict alien city in space, *far outside of* a star system.

b. Lightning can strike up to *10 miles outside of* a thunderstorm.

c. The participants were primarily Caucasian and lived *at least 25 miles outside of* a town of 12,500 or more people.

Modified Locatives (cont.)

Modified *outside* (topological use) – pseudo-intersective:

Fido is less than 5m outside of a doghouse

There is a doghouse X such that Fido is less than 5m from X

and for every doghouse Y Fido is outside Y

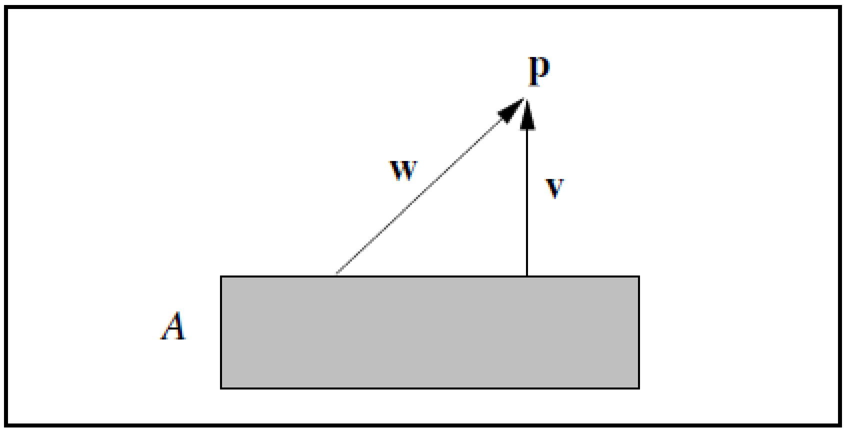
Hence it is not truly existential

Modified Locatives (cont.)

Projective locatives + projective modifier:

a. The bird is straight above the house.

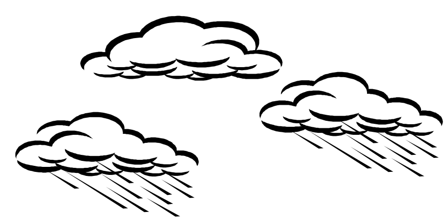
b. The bird is diagonally above the house.



Non-intersective, using shortest vector (Z&W)

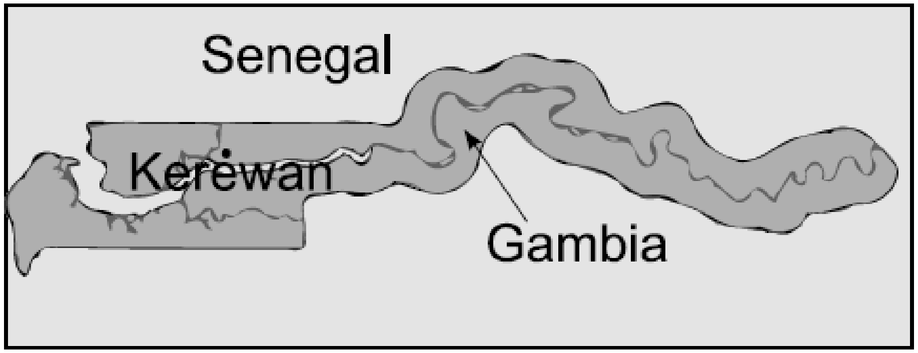
Non-existential locative indefinites:

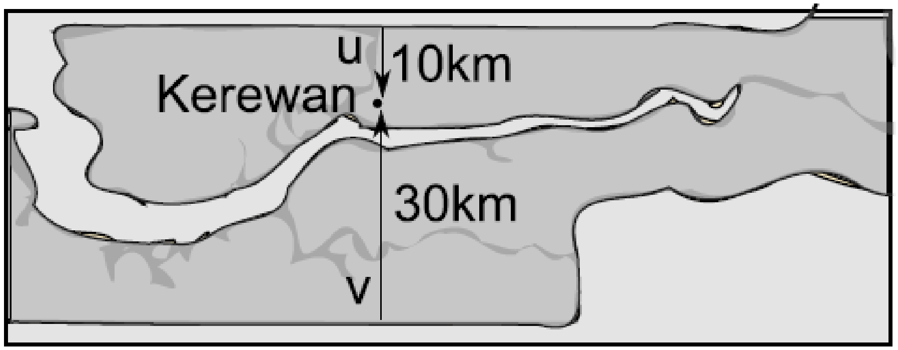
The bird is diagonally above a cloud.



Modified Locatives (cont.)

Projective locatives + distal modifier:





Kerewan is **10km south of** the Senegalsese Border.

Kerewan is **30km north of** the Senegalsese Border.

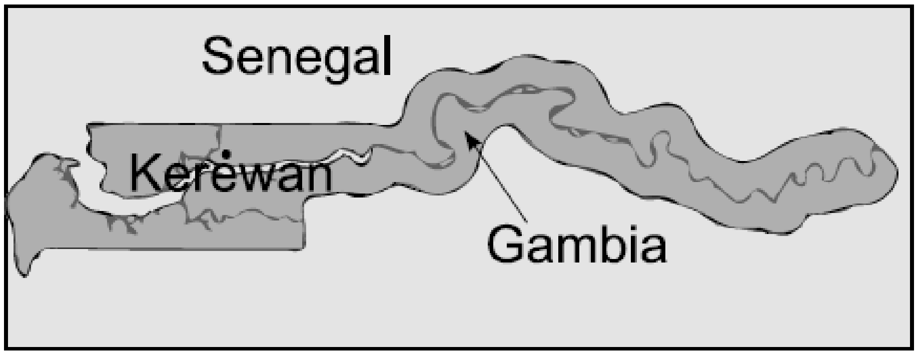
We are interested in the length of the shortest vector(s) among the **vectors north of** the border, and of the shortest vector(s) among the **vectors south of** the border.

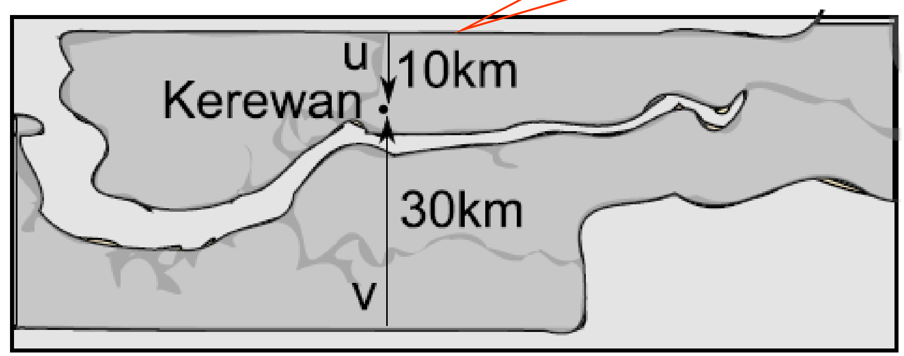
Not the shortest vectors from the border (contra Z&W).

Modified Locatives (cont.)

Projective locatives + distal modifier:

Don’t care…





Kerewan is **30km north of** the Senegalsese Border.

=

Of the vectors **pointing northbound** from the Senegalese border to Kerewan, the shortest vector is 30km.

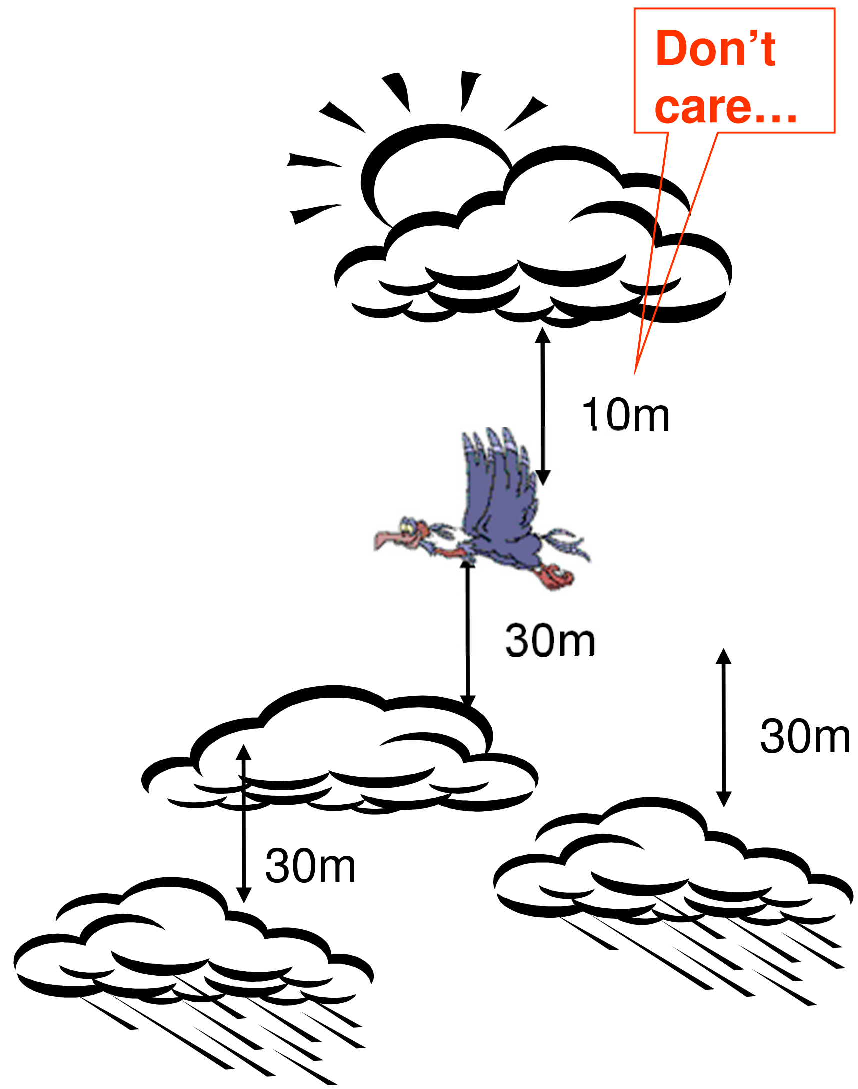
Modified Locatives (cont.)

Now with locative indefinites:

Tweety is **30m above** a cloud.

=

Of the vectors **pointing upward** from a cloud to Tweety, the shortest vector is 30km.



An alternative account - decomposition

Michael is far from a gas station.

a. Michael is [[NEG [close to]] [a gas station]]

b. Michael is [NEG [[close to] [a gas station]]]

Problems:

1. Why decomposing *far from* and not *close to*? (cf. Heim/Büring).
2. Radical decomposition: *exactly* ➔ *at least and at most*?
3. How to modify *3m outside*? As **3m not inside*?
4. How to account for non-existential effects that are not obtained by any decomposition?

   The dot is **left of** a circle.

   Tweety is **30m/diagonally above** a cloud.

Remarks and Speculations

- Existentiality
- Extensionality
- Specificity
- *Some* vs. *a*
- Kind readings
- NPIs
- Collectivity

Eigenspace readings and existentiality

Existential entailment/implication?

- *During a car race in the desert, Michael's Ferrari F2002 is running out of oil. Unfortunately, the oil used by Michael's Ferrari is extremely hard to find.*

  Michael is far from a gas station that sells this type of oil.

- Leonhard is far from a proof of his conjecture.

Our suggestion: only ***spatial*** eigenspace readings are existential.
– LOC function triggers existence requirement

Eigenspace readings and extensionality

a. John is far from a school.

b. John is far from a church.

Context: *All schools are churches and all churches are schools.*

a. The world is far from a social revolution.

b. The world is far from a solution to the inequality between people.

Context: *Every social revolution is (or would be) a solution to the inequality between people, and every solution to the inequality between people is (or would be) a social revolution.*

Our suggestion: all eigenspace readings are extensional.

Eigenspace readings and specificity

We're far from a gas station that I read about in the guide.

Our suggestion: indefinites are ***ambiguous*** *between properties and E-quantifiers.*

Eigenspace readings and *some*

a.	John is a teacher.	(predication, #identity)
b.	John is a teacher that I read about in the press.	(#predication, identity)
c.	#John is some teacher.	(*predication, #identity)
d.	John is some teacher that I read about in the press.	(*predication, identity)

a.	A dog barks.	(generic, #existential)
b.	A dog that I know barks.	(#generic, existential)
c.	Some dog barks.	(*generic, #existential)
d.	Some dog that I know barks.	(*generic, existential)

) a.	We're far from a gas station.	(universal, #existential)
b.	We're far from a gas station that I read about in the guide. (=(22))	(#universal, existential)
c.	We're far from some gas station.	(*universal, #existential)
d.	We're far from some gas station that I read about in the guide.	(*universal, existential)

Our suggestion: ***some*** is only existential. There is a connection between genericity and eigenspace readings.

Eigenspace readings and kind readings

McNally:

a. There was every kind of doctor at the convention.

b. Martha has been every kind of doctor.

a. *There was every doctor at the convention.

b. *Martha has been every doctor.

a. There was a doctor at the convention.

b. Martha has been a doctor.

Michael is far from *the kind of* gas station that sells this type of oil.

Our suggestion: following McNally – *kind* nouns trigger the property analysis.

Eigenspace readings and NPI/monotonicity

a. Michael is far from any gas station.
b. ?Michael is close to any gas station.

a. This park is outside any urban area.
b. ?This park is inside any urban area.

a. My country is far from Eurasia $\Rightarrow$ My country is far from Asia
b. My country is close to Eurasia $\not\Rightarrow$ My country is close to Asia

a. My country is outside Eurasia $\Rightarrow$ My country is outside Asia
b. My country is inside Eurasia $\not\Rightarrow$ My country is inside Asia

Our suggestion: the eigenspace analysis naturally exploits the monotonicity properties of spatial Ps.

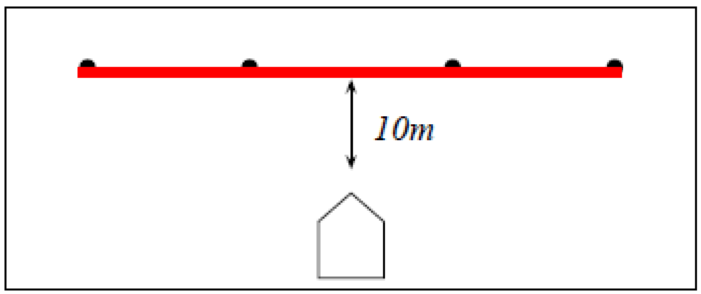
a. Michael is far from a gas station $\Rightarrow$ Michael is far from a big gas station
b. Michael is close to a gas station $\not\Rightarrow$ Michael is close to a big gas station

a. This park is outside an urban area $\Rightarrow$ This park is outside an industrial urban area
b. This park is inside an urban area $\not\Rightarrow$ This park is inside an industrial urban area

The Impure-Atom analysis of Plurals

a. The house is (exactly) 10m away from the (row of) utility poles.

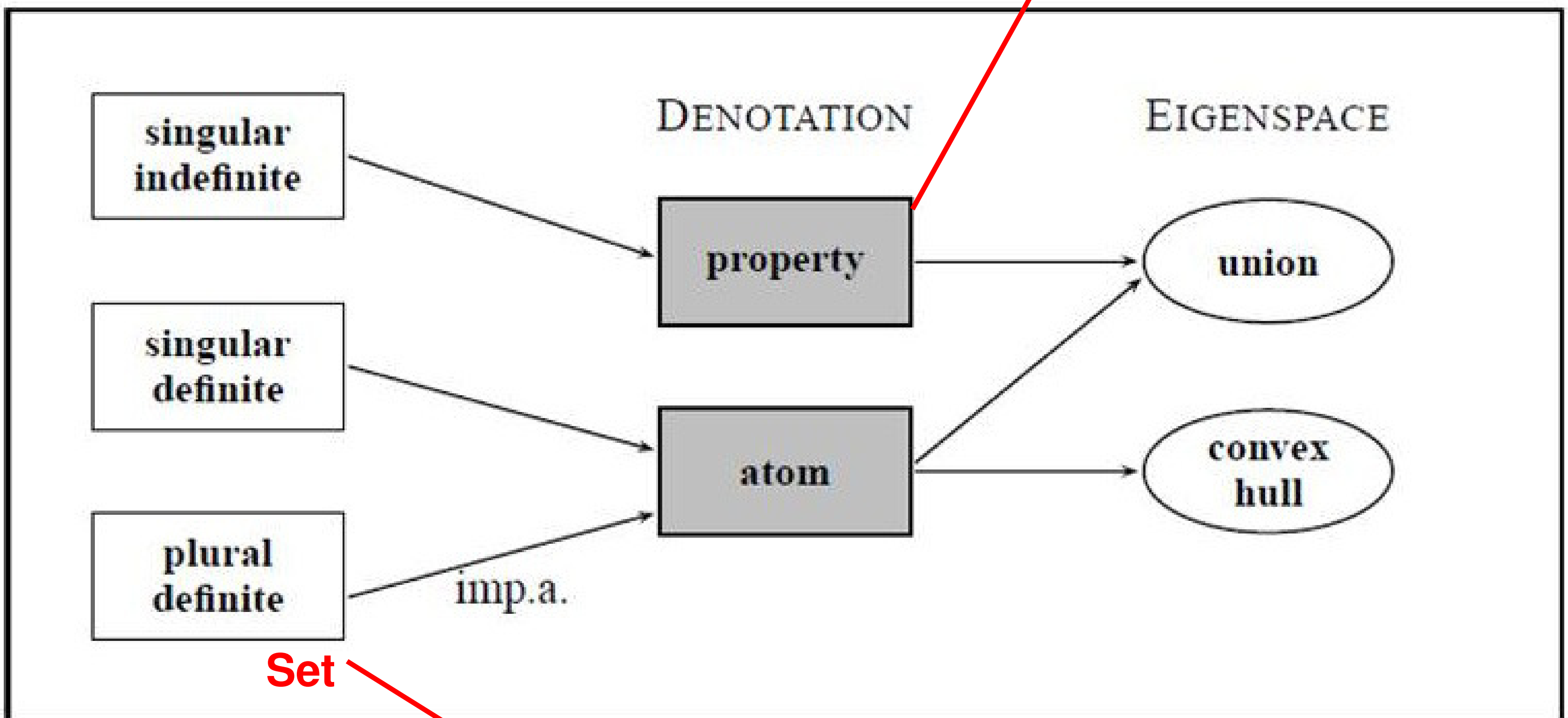
b. The house is (exactly) 10m away from a utility pole.



Our suggestion: the eigenspace of a plural can be a **convex hull of the union** of eigenspaces.

Big Picture

Existential quantification (specificity)



Distributive quantification

Assumptions

1. **Indefinites in PPs are derivationally ambiguous** (Partee 87).

Michael is far from a gas station.

a. Michael is [[far from] [E [a gas station]]]

b. Michael is [[far from] [a gas station]]

2. **Two levels of analysis of locatives** (Zwarts and Winter 00).

Michael is far from London.

a. Syntactic-semantic: $\mathrm{far\_from}(\mathrm{m}, \mathrm{l})$

b. Conceptual-semantic: $\textsc{far\_from}(m, L)$

3. **Property-Eigenspace Hypothesis**: A property's eigenspace is the <u>union</u> of eigenspaces for entities in its extension.

$$\textsc{far\_from}(m, \textit{GS})$$

$$\textsc{loc}(\textit{GS}) = \bigcup\{\textsc{loc}(x) : x \in \mathrm{gs}\}$$

the property GS is located at
the union of gas station locations

Plan

- Overview of locatives with non-existential indefinites
- Their account using the PEH
- Why existential analyses fail
- Remarks and speculations:
    - Existentiality
    - Extensionality
    - Specificity
    - *Some* vs. *a*
    - Kind readings
    - NPIs
    - Collectivity

Lexical Reciprocity as a Typicality Preference: Experimental Evidence

Yoad Winter

Joint work with Imke Kruitwagen and Eva Poortman

ESSLLI2016, Bolzano - Bolzen, 22-26 August 2016

To appear in *NELS 2016*

Reciprocal verbs

Focus: verbs like *hug, kiss, collide*

Two usages:

A and B hug

A hugs B

Old assumption:

Reciprocity = Symmetric Participation

A and B hug ⇔ A hugs B and B hugs A

Newer assumption:

Reciprocity *entails* Symmetric Participation

A and B hug ➔ A hugs B and B hugs A

Claim: Neither assumption is correct. The two entries are logically independent, but related through typicality.

“They are hugging” in Google Images

Hypothesis: for *A&B hug,* and with many other verbs, symmetric participation is not required.

Aim

Examine whether a substantial percentage of speakers accepts reciprocity without symmetric participation above chance level, for a substantial number of reciprocal verbs.

Materials - Verbs

knuffelen – “hug”

botsen (tegen) – “collide (with)”

appen – “send WhatsApp message to (each other)”

praten (tegen) – “talk (to)”

spreken (tegen) – “speak (to)”

kletsen (tegen) – “chat (to)”

roddelen (tegen) – “gossip (to)”

vechten (tegen) – “fight (against)”

Why not “*talk with*” etc.?

Materials – target items

One side is **active**; the other side is (visibly) **passive**.

Passive side shows **collaboration**.

Truth-judgement task for two sentences:

Collective – *het meisje en de vrouw knuffelen*

"the girl and the woman hug"

Binary – *het meisje knuffelt de vrouw*

"the woman hugs the girl"

Materials – more target illustrations

Materials – more target illustrations

Materials - Fillers

8 target verbs

X 2 sentences (collective + binary)

= **18** target items

+ 30 fillers, of two types – to hit balance between expected true/false ratios:

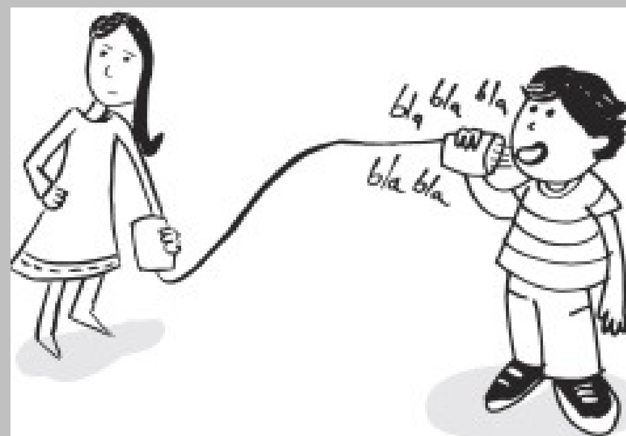
1. Collective/binary sentences, in situations where they are clearly true
2. Other types of sentences, in situations where they are <u>not</u> clearly true/false

Procedure

- 48 Dutch speakers (female 37, age M=23)
- Trials on a screen in a pseudo-random order (Open Sesame)
- green key for “true” and a red key for “false”

Control task

Appendix – 9 control items



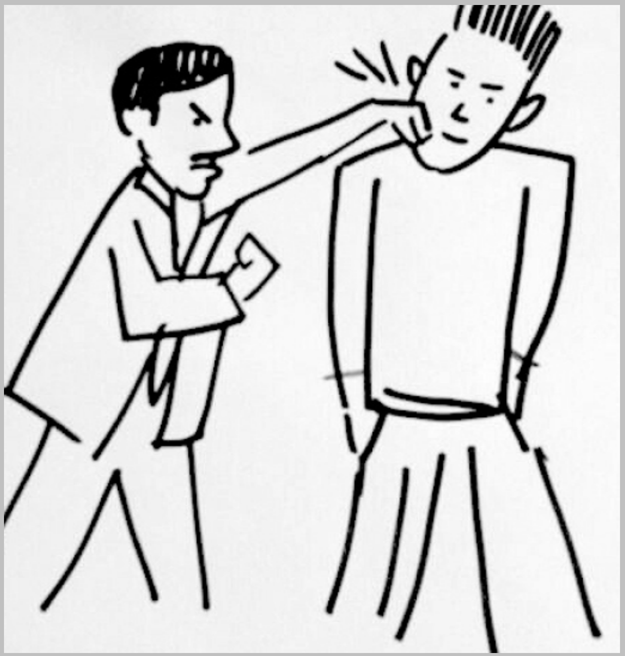
Only collective sentences:

“the girl and the woman hug”

“the boy and the girl talk”

More control drawings

More control drawings



Results summary

verb	**col+**	**bin+**	**col+bin-**	**ctrl.col+**
hug	79%	31%	48%	19%
collide	98%	2%	96%	65%
appen	94%	8%	85%	44%
talk	46%	4%	42%	13%
speak	69%	13%	56%	33%
chat	98%	17%	81%	27%
gossip	90%	6%	83%	46%
fight	73%	15%	58%	23%
MEAN	**81%**	**12%**	**69%**	**34%**

Results summary

verb	col+	bin+	col+bin-	ctrl.col+
hug	79%	31%	48%	19%
collide	98%	2%	96%	65%
appen	94%	[illegible]	85%	44%
talk	[illegible]	[illegible]	42%	13%
speak	[illegible]	[illegible]	56%	33%
chat	98%	17%	81%	27%
gossip	90%	6%	83%	46%
fight	73%	15%	58%	23%
MEAN	**81%**	**12%**	**69%**	**34%**

Changed their mind:
24-66%, *M*=40%

Pilot – video clips

knuffelen – “hug”

botsen (tegen) – “collide (with)”

appen – “send WhatsApp message to (each other)”

praten (tegen) – “talk (to)”

vechten (tegen) – “fight (against)”

After showing the film, the sentence was:

“Violet and Mark hugged/collided/apped/talked/fought”

Or: “Mark hugged/… Violet”

Results summary

Verb	Col+	Bin-	Col+Bin-	Ctrl.Col+
hug	64%	28%	36%	24%
collide	92%	0%	92%	76%
appen	20%	0%	20%	8%
talk	48%	4%	48%	8%
fight	48%	4%	48%	8%
MEAN	**54%**	**7%**	**49%**	**25%**

Discussion

- Symmetric participation is not required with collective verbs that are traditionally classified as “reciprocal”
- Attitude of passive side matters: collaboration positively affects collective judgement

Outline of theory:

For pseudo-reciprocal predicates P, an event e is *typical* for P proportionally to two values:

- **Participation**, e.g. number of hugs
- Evidence for **collective intentionality**

The higher the typicality value is, the higher the chance is that the event passes the speaker threshold for “truth”.

Acknowledgements

***Support**: NWO VICI grant Winter*

***Illustrations**: Ruth Noy Shapira*

***Mark:** Hilbert Dijkstra*

***Violet:** Lianne Zandstra*

***Ideas & judgements:** Joost Zwarts, and many other colleagues and students*

Abstract Categorial Grammar

Yoad Winter

Formal Semantics of Natural Language
NASSLLI2016, Rutgers University, 9-10 July 2016

Our minimalist formal semantics

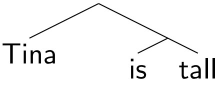
- Trees over <u>strings</u>
- Lexical <u>semantic</u> types and denotations
- Inductive interpretation of trees using <u>function application</u>

Our minimalist formal semantics

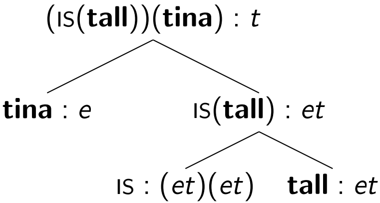
- Trees over strings
- Lexical semantic types and denotations
- Inductive interpretation of trees using function application

Example

A.



B.



Three classic problems

Quantifiers in object position

Tina praised every student

> *What do we do with types $e(et)$ and $(et)t$?*

Three classic problems

Quantifiers in object position

Tina praised every student

> *What do we do with types* $e(et)$ *and* $(et)t$*?*

Quantifier scope

Some teacher praised every student

> *How do we derive the object wide scope reading?*

Three classic problems

Quantifiers in object position

Tina praised every student

What do we do with types $e(et)$ *and* $(et)t$*?*

Quantifier scope

Some teacher praised every student

How do we derive the object wide scope reading?

Extraction

Some teacher that Mary praised smiled

How can we interpret constituents like Mary praised?

Our modified system

- **Hypothetical Reasoning**: a dual principle to Function Application.
- **Signs**: pairs of sounds and meanings replace strings as the items manipulated by the grammar.

Function Application and Modus Ponens

Function Application (FA) Rule

$$\frac{\tau\sigma \qquad \tau}{\sigma}$$

Interpretation

$$\frac{A \qquad B}{A(B)}$$

Function Application and Modus Ponens

Function Application (FA) Rule

$$\frac{\tau\sigma \qquad \tau}{\sigma}$$

Interpretation

$$\frac{A \qquad B}{A(B)}$$

Implication Elimination (Modus Ponens)

$$\frac{\varphi \to \psi \qquad \varphi}{\psi}$$

Function Application and Modus Ponens

Function Application (FA) Rule

$$\frac{\tau\sigma \qquad \tau}{\sigma}$$

Interpretation

$$\frac{A \qquad B}{A(B)}$$

Implication Elimination (Modus Ponens)

$$\frac{\varphi \to \psi \qquad \varphi}{\psi}$$

If Mary is tall then Tina is tall,
and Mary is tall
$\Rightarrow$ Tina is tall

Hypothetical Reasoning – Two Equivalent Patterns

(A) Tina is taller than Mary
$\Rightarrow$ If Mary is tall then Tina is tall

Hypothetical Reasoning – Two Equivalent Patterns

(A) Tina is taller than Mary
$\Rightarrow$ If Mary is tall then Tina is tall

(B) Tina is taller than Mary
and Mary is tall
$\Rightarrow$ Tina is tall

Hypothetical Reasoning – Two Equivalent Patterns

(A) Tina is taller than Mary
$\Rightarrow$ If Mary is tall then Tina is tall

(B) Tina is taller than Mary
and Mary is tall
$\Rightarrow$ Tina is tall

Suppose we accept entailment (A). General principles of entailment, plus a general principle of conditional reasoning – *Modus Ponens* – force us to accept (B).

Hypothetical Reasoning – Two Equivalent Patterns

(A) Tina is taller than Mary
$\Rightarrow$ If Mary is tall then Tina is tall

(B) Tina is taller than Mary
and Mary is tall
$\Rightarrow$ Tina is tall

Suppose we accept entailment (A). General principles of entailment, plus a general principle of conditional reasoning – *Modus Ponens* – force us to accept (B).

Suppose we accept entailment (B). General principles of entailment, plus a general principle of conditional reasoning – which one? – <u>should</u> force us to accept (A).

Hypothetical Reasoning – Two Equivalent Patterns

(A) Tina is taller than Mary

$\Rightarrow$ If Mary is tall then Tina is tall

Hypothetical Reasoning – Two Equivalent Patterns

(A) Tina is taller than Mary
$\Rightarrow$ If Mary is tall then Tina is tall

(B) Tina is taller than Mary
and Mary is tall
$\Rightarrow$ Tina is tall

Hypothetical Reasoning – Two Equivalent Patterns

(A) Tina is taller than Mary
$\Rightarrow$ If Mary is tall then Tina is tall

(B) Tina is taller than Mary
and Mary is tall
$\Rightarrow$ Tina is tall

Proving (B) using (A)

$$\dfrac{\dfrac{\text{Tina is taller than Mary}}{\text{If Mary is tall then Tina is tall}}\ \text{(A)} \qquad \text{Mary is tall}}{\text{Tina is tall}}\ \text{MP}$$

Hypothetical Reasoning – Two Equivalent Patterns

(A) Tina is taller than Mary
$\Rightarrow$ If Mary is tall then Tina is tall

(B) Tina is taller than Mary
and Mary is tall
$\Rightarrow$ Tina is tall

Proving (B) using (A)

$$\dfrac{\dfrac{\text{Tina is taller than Mary}}{\text{If Mary is tall then Tina is tall}}\,(A) \qquad \text{Mary is tall}}{\text{Tina is tall}}\,\text{MP}$$

Proving (A) using (B)

$$\dfrac{\dfrac{\text{Tina is taller than Mary} \qquad [\text{Mary is tall}]^1}{\text{Tina is tall}}\,(B)}{\text{If Mary is tall then Tina is tall}}\,\text{discharge hypothesis 1}$$

Implication Introduction

Implication Introduction

$$\dfrac{\dfrac{\begin{matrix} \ldots & [\varphi]^1 \\ & \vdots \end{matrix}}{\psi}}{\varphi \to \psi} \text{ discharge hypothesis 1}$$

Implication Introduction

Implication Introduction

$$\dfrac{\dfrac{\begin{matrix}\dots & [\varphi]^1 \\ & \vdots\end{matrix}}{\psi}}{\varphi \to \psi}\ \text{discharge hypothesis 1}$$

Example

$$\dfrac{\dfrac{\dfrac{\varphi_1 \to (\varphi_2 \to \psi) \quad [\varphi_1]^1}{\varphi_2 \to \psi}\ \text{MP} \qquad \varphi_2}{\psi}\ \text{MP}}{\varphi_1 \to \psi}\ \text{discharge hypothesis 1}$$

Function Abstraction

Function Introduction

$$\begin{array}{c} \ldots \qquad [\tau]^1 \\ \vdots \\ \hline \dfrac{\sigma}{\tau\,\sigma} \end{array} \quad \text{discharge hypothesis 1}$$

Function Abstraction

Function Introduction

$$\dfrac{\dfrac{\begin{matrix}\dots & & [\tau]^1\\ & \vdots & \end{matrix}}{\sigma}}{\tau\sigma}\ \text{discharge hypothesis 1}$$

Example

$$\dfrac{\dfrac{\dfrac{e(et) \quad [e]^1}{et}\ \text{APP} \qquad e}{t}\ \text{APP}}{et}\ \text{discharge hypothesis 1}$$

Function Abstraction – Interpretation

Function Introduction

$$\dfrac{\dfrac{\ldots \qquad [u : \tau]^1 \\ \vdots}{z : \sigma}}{\lambda u.z : \tau\sigma} \text{ discharge hypothesis 1}$$

Function Abstraction – Interpretation

Function Introduction

$$\dfrac{\dfrac{\begin{array}{cc} \ldots & [u : \tau]^1 \\ & \vdots \end{array}}{z : \sigma}}{\lambda u.z : \tau\sigma}\ \text{discharge hypothesis 1}$$

Example

$$\dfrac{\dfrac{\dfrac{\textbf{praise} : e(et) \quad [u : e]^1}{\textbf{praise}(u) : et}\ \text{FA} \quad \textbf{mary} : e}{\textbf{praise}(u)(\textbf{mary}) : t}\ \text{FA}}{\lambda u_e.\textbf{praise}(u)(\textbf{mary}) : et}\ \text{discharge hypothesis 1}$$

Praising Mary: an intermediate summary

The constituent *praised Mary* can be analyzed in two ways.

Praising Mary: an intermediate summary

The constituent *praised Mary* can be analyzed in two ways.

Using Application:

praise(**mary**)

mary **praise**

Praising Mary: an intermediate summary

The constituent *praised Mary* can be analyzed in two ways.

Using Application:

$$\mathbf{praise}(\mathbf{mary})$$

$$\mathbf{mary} \qquad \mathbf{praise}$$

Using Abstraction:

$$\lambda u.\mathbf{praise}(u)(\mathbf{mary})$$

$$\mathbf{mary} \qquad \mathbf{praise}$$

Praising Mary: an intermediate summary

The constituent *praised Mary* can be analyzed in two ways.

Using Application:

$$\begin{array}{c} \textbf{praise}(\textbf{mary}) \\ \diagup \quad \diagdown \\ \textbf{mary} \quad \textbf{praise} \end{array}$$

Using Abstraction:

$$\begin{array}{c} \lambda u.\textbf{praise}(u)(\textbf{mary}) \\ \diagup \quad \diagdown \\ \textbf{mary} \quad \textbf{praise} \end{array}$$

Application (Ajdukiewicz):
undergeneration – object quantifiers, wide scope, extraction
overgeneration – extraction

Praising Mary: an intermediate summary

The constituent *praised Mary* can be analyzed in two ways.

Using Application: $\textbf{praise}(\textbf{mary})$

$\textbf{mary} \quad \textbf{praise}$

Using Abstraction: $\lambda u.\textbf{praise}(u)(\textbf{mary})$

$\textbf{mary} \quad \textbf{praise}$

Application (Ajdukiewicz):
undergeneration – object quantifiers, wide scope, extraction
overgeneration – extraction

Application + Abstraction (Lambek-Van Benthem):
less undergeneration
more overgeneration

Using signs

"*The linguistic sign unites, not a thing and a name, but a concept and a sound-image.*"
(de Saussure 1916)

Using signs

"The linguistic sign unites, not a thing and a name, but a concept and a sound-image."
(de Saussure 1916)

A linguistic sign, or in short a **sign**, *is a pair* $\langle P, C\rangle$, *where* P *stands for a* perceptual *representation of sensory input and* C *stands for a* conceptual *representation of meaning.*

Using signs

"The linguistic sign unites, not a thing and a name, but a concept and a sound-image."
(de Saussure 1916)

A linguistic sign*, or in short a* **sign***, is a pair* $\langle P, C \rangle$*, where* P *stands for a* perceptual *representation of sensory input and* C *stands for a* conceptual *representation of meaning.*

Sign composition:

MARY (sign) $$\left\{\begin{array}{ll}\texttt{mary} & \text{(perception)}\\ \textbf{mary} & \text{(concept)}\end{array}\right\}$$

$+$

PRAISE (sign) $$\left\{\begin{array}{ll}\texttt{praise} & \text{(perception)}\\ \textbf{praise} & \text{(concept)}\end{array}\right\}$$

Using signs

"The linguistic sign unites, not a thing and a name, but a concept and a sound-image."
(de Saussure 1916)

> *A linguistic sign, or in short a* **sign**, *is a pair* $\langle P, C \rangle$, *where* P *stands for a* perceptual *representation of sensory input and* C *stands for a* conceptual *representation of meaning.*

Sign composition:

MARY (sign) $\qquad \left\{ \begin{array}{ll} \texttt{mary} & \text{(perception)} \\ \textbf{mary} & \text{(concept)} \end{array} \right\}$

$+$

PRAISE (sign) $\qquad \left\{ \begin{array}{ll} \texttt{praise} & \text{(perception)} \\ \textbf{praise} & \text{(concept)} \end{array} \right\}$

$=\ldots$ (two possibilities)

Strings as perceptual units

The domain of strings $D_f = F$ satisfies:

- *Closure under concatenation*. For all strings $a, b \in F$, the concatenation $a \cdot b$ is also in F.
- *Neutral element for concatenation*. F contains an element ϵ that satisfies for every $x \in F$: $x \cdot \epsilon = \epsilon \cdot x = x$.

Strings as perceptual units

The domain of strings $D_f = F$ satisfies:

- *Closure under concatenation*. For all strings $a, b \in F$, the concatenation $a \cdot b$ is also in F.
- *Neutral element for concatenation*. F contains an element ϵ that satisfies for every $x \in F$: $x \cdot \epsilon = \epsilon \cdot x = x$.

Pheno-types: f is a pheno-type (of strings). If σ and τ are pheno-types then $(\sigma\tau)$ is a pheno-type as well.

Strings as perceptual units

The domain of strings $D_f = F$ satisfies:

- *Closure under concatenation*. For all strings $a, b \in F$, the concatenation $a \cdot b$ is also in F.
- *Neutral element for concatenation*. F contains an element ϵ that satisfies for every $x \in F$: $x \cdot \epsilon = \epsilon \cdot x = x$.

Pheno-types: f is a pheno-type (of strings). If σ and τ are pheno-types then $(\sigma\tau)$ is a pheno-type as well.

Example: In a given model –

- $\mathtt{tina}_f = \mathit{tina}$
- $\mathtt{mary}_f = \mathit{mary}$
- $\mathtt{praise}_{f(ff)} \;=\; \lambda x_f.\lambda y_f.\ y \cdot \mathit{praised} \cdot x$

Strings as perceptual units

The domain of strings $D_f = F$ satisfies:

- *Closure under concatenation*. For all strings $a, b \in F$, the concatenation $a \cdot b$ is also in F.
- *Neutral element for concatenation*. F contains an element ϵ that satisfies for every $x \in F$: $x \cdot \epsilon = \epsilon \cdot x = x$.

Pheno-types: f is a pheno-type (of strings). If σ and τ are pheno-types then $(\sigma\tau)$ is a pheno-type as well.

Example: In a given model –

- $\texttt{tina}_f = \textit{tina}$
- $\texttt{mary}_f = \textit{mary}$
- $\texttt{praise}_{f(ff)} \;=\; \lambda x_f.\lambda y_f.\ y \cdot \textit{praised} \cdot x$

$$\begin{aligned} \texttt{praise}_{f(ff)}(\texttt{mary}_f)(\texttt{tina}_f) &= \textit{tina} \cdot \textit{praised} \cdot \textit{mary} \\ \texttt{praise}_{f(ff)}(\texttt{tina}_f)(\texttt{mary}_f) &= \textit{mary} \cdot \textit{praised} \cdot \textit{tina} \end{aligned}$$

Application and Abstraction using signs

Application and Abstraction using signs

Application

$$\frac{\langle \texttt{praise}_{f(ff)}, \textbf{praise}_{e(et)} \rangle \quad \langle \texttt{mary}_{f}, \textbf{mary}_{e} \rangle}{\langle \texttt{praise}(\texttt{mary}), \textbf{praise}(\textbf{mary}) \rangle} \; APP$$

Application and Abstraction using signs

Application

$$\frac{\langle \texttt{praise}_{f(ff)}, \textbf{praise}_{e(et)} \rangle \quad \langle \texttt{mary}_f, \textbf{mary}_e \rangle}{\langle \texttt{praise}(\texttt{mary}), \textbf{praise}(\textbf{mary}) \rangle}\ APP$$

In our model:

$$= \langle (\lambda x_f.\lambda y_f.\ y \cdot praised \cdot x)(\texttt{mary}_f), \textbf{praise}(\textbf{mary}_e) \rangle$$

Application and Abstraction using signs

Application

$$\frac{\langle \mathtt{praise}_{f(ff)}, \mathbf{praise}_{e(et)} \rangle \quad \langle \mathtt{mary}_f, \mathbf{mary}_e \rangle}{\langle \mathtt{praise}(\mathtt{mary}), \mathbf{praise}(\mathbf{mary}) \rangle}\ APP$$

In our model:

$= \langle (\lambda x_f.\lambda y_f.\ y \cdot \textit{praised} \cdot x)(\mathtt{mary}_f)\,, \mathbf{praise}(\mathbf{mary}_e) \rangle$

$= \langle (\lambda y_f.\ y \cdot \textit{praised} \cdot \mathtt{mary}_f)\,, \mathbf{praise}(\mathbf{mary}_e) \rangle$

Application and Abstraction using signs

Application

$$\frac{\langle \texttt{praise}_{f(ff)}, \textbf{praise}_{e(et)} \rangle \quad \langle \texttt{mary}_f, \textbf{mary}_e \rangle}{\langle \texttt{praise}(\texttt{mary}), \textbf{praise}(\textbf{mary}) \rangle}\ APP$$

In our model:

$$= \langle (\lambda x_f.\lambda y_f.\ y \cdot praised \cdot x)(\texttt{mary}_f)\ , \textbf{praise}(\textbf{mary}_e) \rangle$$
$$= \langle (\lambda y_f.\ y \cdot praised \cdot \texttt{mary}_f)\ , \textbf{praise}(\textbf{mary}_e) \rangle$$

Abstraction

$$\dfrac{\dfrac{\dfrac{\langle \texttt{praise}_{f(ff)}, \textbf{praise}_{e(et)} \rangle \quad [\langle u_f, u_e \rangle]^1}{\langle \texttt{praise}(u_f), \textbf{praise}(u_e) \rangle}\ \text{FA} \qquad \langle \texttt{mary}_f, \textbf{mary}_e \rangle}{\langle \texttt{praise}(u_f)(\texttt{mary}),\ \textbf{praise}(u_e)(\textbf{mary}) \rangle}\ \text{FA}}{\langle \lambda u_f.\texttt{praise}(u_f)(\texttt{mary}),\ \lambda u_e.\textbf{praise}(u_e)(\textbf{mary}) \rangle}\ \text{discharge hypothesis 1}$$

Application and Abstraction using signs

Application

$$\frac{\langle \texttt{praise}_{f(ff)}, \mathbf{praise}_{e(et)} \rangle \quad \langle \texttt{mary}_{f}, \mathbf{mary}_{e} \rangle}{\langle \texttt{praise}(\texttt{mary}), \mathbf{praise}(\mathbf{mary}) \rangle}\ APP$$

In our model:

$= \langle (\lambda x_f.\lambda y_f.\ y \cdot praised \cdot x)(\texttt{mary}_f)\ , \mathbf{praise}(\mathbf{mary}_e) \rangle$

$= \langle (\lambda y_f.\ y \cdot praised \cdot \texttt{mary}_f)\ , \mathbf{praise}(\mathbf{mary}_e) \rangle$

Abstraction

$$\cfrac{\cfrac{\cfrac{\langle \texttt{praise}_{f(ff)}, \mathbf{praise}_{e(et)} \rangle \quad [\langle u_f, u_e \rangle]^1}{\langle \texttt{praise}(u_f), \mathbf{praise}(u_e) \rangle}\ \text{FA} \quad \langle \texttt{mary}_f, \mathbf{mary}_e \rangle}{\langle \texttt{praise}(u_f)(\texttt{mary}),\ \mathbf{praise}(u_e)(\mathbf{mary}) \rangle}\ \text{FA}}{\langle \lambda u_f.\texttt{praise}(u_f)(\texttt{mary}),\ \lambda u_e.\mathbf{praise}(u_e)(\mathbf{mary}) \rangle}\ \text{discharge hypothesis 1}$$

In our model:

$= \langle \lambda u_f.(\lambda x_f.\lambda y_f.\ y \cdot praised \cdot x)(u_f)(\texttt{mary}),\ \lambda u_e.\mathbf{praise}(u_e)(\mathbf{mary}) \rangle$

Application and Abstraction using signs

Application

$$\frac{\langle \mathtt{praise}_{f(ff)}, \mathbf{praise}_{e(et)}\rangle \quad \langle \mathtt{mary}_f, \mathbf{mary}_e\rangle}{\langle \mathtt{praise}(\mathtt{mary}), \mathbf{praise}(\mathbf{mary})\rangle}\ APP$$

In our model:

$$= \langle (\lambda x_f.\lambda y_f.\ y \cdot \mathit{praised} \cdot x)(\mathtt{mary}_f)\ , \mathbf{praise}(\mathbf{mary}_e)\rangle$$
$$= \langle (\lambda y_f.\ y \cdot \mathit{praised} \cdot \mathtt{mary}_f)\ , \mathbf{praise}(\mathbf{mary}_e)\rangle$$

Abstraction

$$\cfrac{\cfrac{\cfrac{\langle \mathtt{praise}_{f(ff)}, \mathbf{praise}_{e(et)}\rangle \quad [\langle u_f, u_e\rangle]^1}{\langle \mathtt{praise}(u_f), \mathbf{praise}(u_e)\rangle}\ \mathrm{FA} \quad \langle \mathtt{mary}_f, \mathbf{mary}_e\rangle}{\langle \mathtt{praise}(u_f)(\mathtt{mary}),\ \mathbf{praise}(u_e)(\mathbf{mary})\rangle}\ \mathrm{FA}}{\langle \lambda u_f.\mathtt{praise}(u_f)(\mathtt{mary}),\ \lambda u_e.\mathbf{praise}(u_e)(\mathbf{mary})\rangle}\ \text{discharge hypothesis 1}$$

In our model:

$$= \langle \lambda u_f.(\lambda x_f.\lambda y_f.\ y \cdot \mathit{praised} \cdot x)(u_f)(\mathtt{mary}),\ \lambda u_e.\mathbf{praise}(u_e)(\mathbf{mary})\rangle$$
$$= \langle \lambda u_f.(\lambda y_f.\ y \cdot \mathit{praised} \cdot u_f)(\mathtt{mary}),\ \lambda u_e.\mathbf{praise}(u_e)(\mathbf{mary})\rangle$$

Application and Abstraction using signs

Application

$$\frac{\langle \mathtt{praise}_{f(ff)}, \mathbf{praise}_{e(et)}\rangle \quad \langle \mathtt{mary}_{f}, \mathbf{mary}_{e}\rangle}{\langle \mathtt{praise}(\mathtt{mary}), \mathbf{praise}(\mathbf{mary})\rangle}\ APP$$

In our model:

$$= \langle (\lambda x_f.\lambda y_f.\ y \cdot praised \cdot x)(\mathtt{mary}_f)\ , \mathbf{praise}(\mathbf{mary}_e)\rangle$$
$$= \langle (\lambda y_f.\ y \cdot praised \cdot \mathtt{mary}_f)\ , \mathbf{praise}(\mathbf{mary}_e)\rangle$$

Abstraction

$$\cfrac{\cfrac{\cfrac{\langle \mathtt{praise}_{f(ff)}, \mathbf{praise}_{e(et)}\rangle \quad [\langle u_f, u_e\rangle]^1}{\langle \mathtt{praise}(u_f), \mathbf{praise}(u_e)\rangle}\ \text{FA} \quad \langle \mathtt{mary}_f, \mathbf{mary}_e\rangle}{\langle \mathtt{praise}(u_f)(\mathtt{mary}),\ \mathbf{praise}(u_e)(\mathbf{mary})\rangle}\ \text{FA}}{\langle \lambda u_f.\mathtt{praise}(u_f)(\mathtt{mary}),\ \lambda u_e.\mathbf{praise}(u_e)(\mathbf{mary})\rangle}\ \text{discharge hypothesis 1}$$

In our model:

$$= \langle \lambda u_f.(\lambda x_f.\lambda y_f.\ y \cdot praised \cdot x)(u_f)(\mathtt{mary}),\ \lambda u_e.\mathbf{praise}(u_e)(\mathbf{mary})\rangle$$
$$= \langle \lambda u_f.(\lambda y_f.\ y \cdot praised \cdot u_f)(\mathtt{mary}),\ \lambda u_e.\mathbf{praise}(u_e)(\mathbf{mary})\rangle$$
$$= \langle \lambda u_f.\mathtt{mary} \cdot praised \cdot u_f,\ \lambda u_e.\mathbf{praise}(u_e)(\mathbf{mary})\rangle$$

What have we achieved so far?

Two ways of combining the signs `MARY` and `PRAISE`:

- *Direct application*:

$\texttt{PRAISE}(\texttt{MARY}) =$

string	*praised Mary*
denotation	$\{x \in E : x \text{ praised Mary}\}$

What have we achieved so far?

Two ways of combining the signs $\texttt{MARY}$ and $\texttt{PRAISE}$:

- *Direct application*:

  $\texttt{PRAISE}(\texttt{MARY}) =$

string	*praised Mary*
denotation	$\{x \in E : x \text{ praised Mary}\}$

- *With abstraction*:

  $\lambda\texttt{U}.\texttt{PRAISE}(\texttt{U})(\texttt{MARY}) =$

string	*Mary praised*
denotation	$\{y \in E : \text{Mary praised } y\}$

What have we achieved so far?

Two ways of combining the signs MARY and PRAISE:

- *Direct application*:

  PRAISE(MARY) =

string	*praised Mary*
denotation	$\{x \in E : x$ praised Mary$\}$

- *With abstraction*:

  λU.PRAISE(U)(MARY) =

string	*Mary praised*
denotation	$\{y \in E :$ Mary praised $y\}$

No overgeneration!

What have we achieved so far?

Two ways of combining the signs MARY and PRAISE:

- *Direct application*:

  PRAISE(MARY) =

string	*praised Mary*
denotation	$\{x \in E : x$ praised Mary$\}$

- *With abstraction*:

  λU.PRAISE(U)(MARY) =

string	*Mary praised*
denotation	$\{y \in E :$ Mary praised $y\}$

No overgeneration!

Hypothesis The Lambek-Van Benthem Calculus (Application + Abstraction) is a suitable logical apparatus for manipulating the composition of signs in natural language grammar.

Relative clauses (1): using empty strings

Consider the string *that ran* in *some man* <u>*that ran*</u> *smiled* .

$$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon)$$

Relative clauses (1): using empty strings

Consider the string *that ran* in *some man* <u>*that ran*</u> *smiled* .

$$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot that \cdot P(\epsilon)$$

$$\texttt{run} \;=\; \lambda u_f.u \cdot ran$$

Relative clauses (1): using empty strings

Consider the string *that ran* in *some man* <u>*that ran*</u> *smiled* .

$$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon)$$

$$\texttt{run} \;=\; \lambda u_f.u \cdot \mathit{ran}$$

$$\texttt{that}(\texttt{run})$$

Relative clauses (1): using empty strings

Consider the string *that ran* in *some man* <u>*that ran*</u> *smiled* .

$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot \textit{that} \cdot P(\epsilon)$

$\texttt{run} \;=\; \lambda u_f.u \cdot \textit{ran}$

$\texttt{that}(\texttt{run})$
$= (\lambda P_{ff}.\lambda y_f.\, y \cdot \textit{that} \cdot P(\epsilon))(\texttt{run})$

Relative clauses (1): using empty strings

Consider the string *that ran* in *some man <u>that ran</u> smiled* .

$$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon)$$

$$\texttt{run} \;=\; \lambda u_f.u \cdot \mathit{ran}$$

$$\begin{aligned}&\texttt{that}(\texttt{run})\\ &= (\lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon))(\texttt{run})\\ &= \lambda y_f.\, y \cdot \mathit{that} \cdot \texttt{run}(\epsilon)\end{aligned}$$

Relative clauses (1): using empty strings

Consider the string *that ran* in *some man <u>that ran</u> smiled* .

$$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot that \cdot P(\epsilon)$$

$$\texttt{run} \;=\; \lambda u_f.u \cdot ran$$

$$\begin{aligned}
&\texttt{that}(\texttt{run}) \\
&= (\lambda P_{ff}.\lambda y_f.\, y \cdot that \cdot P(\epsilon))(\texttt{run}) \\
&= \lambda y_f.\, y \cdot that \cdot \texttt{run}(\epsilon) \\
&= \lambda y_f.\, y \cdot that \cdot ((\lambda u_f.u \cdot ran)(\epsilon))
\end{aligned}$$

Relative clauses (1): using empty strings

Consider the string *that ran* in *some man* <u>*that ran*</u> *smiled* .

$$\mathtt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon)$$

$$\mathtt{run} \;=\; \lambda u_f.u \cdot \mathit{ran}$$

$$\begin{aligned}
&\mathtt{that}(\mathtt{run})\\
&= (\lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon))(\mathtt{run})\\
&= \lambda y_f.\, y \cdot \mathit{that} \cdot \mathtt{run}(\epsilon)\\
&= \lambda y_f.\, y \cdot \mathit{that} \cdot ((\lambda u_f.u \cdot \mathit{ran})(\epsilon))\\
&= \lambda y_f.\, y \cdot \mathit{that} \cdot (\epsilon \cdot \mathit{ran})
\end{aligned}$$

Relative clauses (1): using empty strings

Consider the string *that ran* in *some man* <u>*that ran*</u> *smiled* .

$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon)$

$\texttt{run} \;=\; \lambda u_f.u \cdot \mathit{ran}$

$\texttt{that}(\texttt{run})$
$= (\lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon))(\texttt{run})$
$= \lambda y_f.\, y \cdot \mathit{that} \cdot \texttt{run}(\epsilon)$
$= \lambda y_f.\, y \cdot \mathit{that} \cdot ((\lambda u_f.u \cdot \mathit{ran})(\epsilon))$
$= \lambda y_f.\, y \cdot \mathit{that} \cdot (\epsilon \cdot \mathit{ran})$
$= \lambda y_f.\, y \cdot \mathit{that} \cdot \mathit{ran}$

Relative clauses (2): using empty strings

Consider the string *that Mary praised*.

$$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon)$$

Relative clauses (2): using empty strings

Consider the string *that Mary praised*.

$$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot that \cdot P(\epsilon)$$

$$\texttt{mp} \;=\; \lambda u_f. mary \cdot praised \cdot u$$

Relative clauses (2): using empty strings

Consider the string *that Mary praised*.

$$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon)$$

$$\texttt{mp} \;=\; \lambda u_f.\mathit{mary} \cdot \mathit{praised} \cdot u$$

$$\texttt{that}(\texttt{mp})$$

Relative clauses (2): using empty strings

Consider the string *that Mary praised*.

$$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot \textit{that} \cdot P(\epsilon)$$

$$\texttt{mp} \;=\; \lambda u_f.\textit{mary} \cdot \textit{praised} \cdot u$$

$$\texttt{that}(\texttt{mp})$$
$$= (\lambda P_{ff}.\lambda y_f.\, y \cdot \textit{that} \cdot P(\epsilon))(\texttt{mp})$$

Relative clauses (2): using empty strings

Consider the string *that Mary praised*.

$$\texttt{that} \ = \ \lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon)$$

$$\texttt{mp} \ = \ \lambda u_f.\mathit{mary} \cdot \mathit{praised} \cdot u$$

$$\begin{aligned}&\texttt{that}(\texttt{mp})\\ &= (\lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon))(\texttt{mp})\\ &= \lambda y_f.\, y \cdot \mathit{that} \cdot \texttt{mp}(\epsilon)\end{aligned}$$

Relative clauses (2): using empty strings

Consider the string *that Mary praised*.

$$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon)$$

$$\texttt{mp} \;=\; \lambda u_f.\mathit{mary} \cdot \mathit{praised} \cdot u$$

$$\begin{aligned}
&\texttt{that}(\texttt{mp}) \\
&= (\lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon))(\texttt{mp}) \\
&= \lambda y_f.\, y \cdot \mathit{that} \cdot \texttt{mp}(\epsilon) \\
&= \lambda y_f.\, y \cdot \mathit{that} \cdot ((\lambda u_f.\mathit{mary} \cdot \mathit{praised} \cdot u)(\epsilon))
\end{aligned}$$

Relative clauses (2): using empty strings

Consider the string *that Mary praised*.

$$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon)$$

$$\texttt{mp} \;=\; \lambda u_f.\mathit{mary} \cdot \mathit{praised} \cdot u$$

$$\begin{aligned}
&\texttt{that}(\texttt{mp})\\
&= (\lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon))(\texttt{mp})\\
&= \lambda y_f.\, y \cdot \mathit{that} \cdot \texttt{mp}(\epsilon)\\
&= \lambda y_f.\, y \cdot \mathit{that} \cdot ((\lambda u_f.\mathit{mary} \cdot \mathit{praised} \cdot u)(\epsilon))\\
&= \lambda y_f.\, y \cdot \mathit{that} \cdot (\mathit{mary} \cdot \mathit{praised} \cdot \epsilon)
\end{aligned}$$

Relative clauses (2): using empty strings

Consider the string *that Mary praised*.

$$\texttt{that} \;=\; \lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon)$$

$$\texttt{mp} \;=\; \lambda u_f.\mathit{mary} \cdot \mathit{praised} \cdot u$$

$$\begin{aligned}
&\texttt{that}(\texttt{mp}) \\
&= (\lambda P_{ff}.\lambda y_f.\, y \cdot \mathit{that} \cdot P(\epsilon))(\texttt{mp}) \\
&= \lambda y_f.\, y \cdot \mathit{that} \cdot \texttt{mp}(\epsilon) \\
&= \lambda y_f.\, y \cdot \mathit{that} \cdot ((\lambda u_f.\mathit{mary} \cdot \mathit{praised} \cdot u)(\epsilon)) \\
&= \lambda y_f.\, y \cdot \mathit{that} \cdot (\mathit{mary} \cdot \mathit{praised} \cdot \epsilon) \\
&= \lambda y_f.\, y \cdot \mathit{that} \cdot \mathit{mary} \cdot \mathit{praised}
\end{aligned}$$

Quantifiers

Consider the noun phrase *someone* in *someone ran* .

$$\texttt{someone} \;=\; \lambda P_{ff}.\, P(\mathit{someone})$$

Quantifiers

Consider the noun phrase *someone* in *someone ran* .

$\texttt{someone} \;=\; \lambda P_{ff}.\, P(\mathit{someone})$

$\texttt{run} \;=\; \lambda u_{f}.u \cdot \mathit{ran}$

Quantifiers

Consider the noun phrase *someone* in *someone ran* .

$\texttt{someone} \ = \ \lambda P_{ff}.\, P(\mathit{someone})$

$\texttt{run} \ = \ \lambda u_f.u \cdot \mathit{ran}$

$\texttt{someone}(\texttt{run})$

Quantifiers

Consider the noun phrase *someone* in *someone ran* .

$\texttt{someone} = \lambda P_{ff}.\, P(someone)$

$\texttt{run} = \lambda u_f . u \cdot ran$

$\texttt{someone(run)}$
$(\lambda P_{ff}.\, P(someone))(\lambda u_f . u \cdot ran)$

Quantifiers

Consider the noun phrase *someone* in *someone ran* .

$\texttt{someone} \;=\; \lambda P_{ff}.\, P(\mathit{someone})$

$\texttt{run} \;=\; \lambda u_f.u \cdot \mathit{ran}$

$\texttt{someone}(\texttt{run})$
$(\lambda P_{ff}.\, P(\mathit{someone}))(\lambda u_f.u \cdot \mathit{ran})$
$(\lambda u_f.u \cdot \mathit{ran})(\mathit{someone})$

Quantifiers

Consider the noun phrase *someone* in *someone ran* .

$\texttt{someone} \;=\; \lambda P_{ff}.\, P(\mathit{someone})$

$\texttt{run} \;=\; \lambda u_f.u \cdot \mathit{ran}$

$\texttt{someone}(\texttt{run})$
$(\lambda P_{ff}.\, P(\mathit{someone}))(\lambda u_f.u \cdot \mathit{ran})$
$(\lambda u_f.u \cdot \mathit{ran})(\mathit{someone})$
$\mathit{someone} \cdot \mathit{ran}$

Question: How about *some* in *some man ran*?

Further: The notion *abstract type* (abstract category).

Relative clauses (3): full derivation

- $\mathtt{SOME}(\mathtt{THAT}(\lambda \mathtt{U}_{f,e}.\mathtt{PRAISE}(\mathtt{U})(\mathtt{MARY}))(\mathtt{TEACHER}))(\mathtt{SMILE})$
  - $\mathtt{SOME}(\mathtt{THAT}(\lambda \mathtt{U}_{f,e}.\mathtt{PRAISE}(\mathtt{U})(\mathtt{MARY}))(\mathtt{TEACHER}))$
    - $\mathtt{SOME}$
    - $\mathtt{THAT}(\lambda \mathtt{U}_{f,e}.\mathtt{PRAISE}(\mathtt{U})(\mathtt{MARY}))(\mathtt{TEACHER})$
      - $\mathtt{TEACHER}$
      - $\mathtt{THAT}(\lambda \mathtt{U}_{f,e}.\mathtt{PRAISE}(\mathtt{U})(\mathtt{MARY}))$
        - $\mathtt{THAT}$
        - $\lambda \mathtt{U}_{f,e}.\mathtt{PRAISE}(\mathtt{U})(\mathtt{MARY})$
          - $\mathtt{MARY}$
          - $\mathtt{PRAISE}$
  - $\mathtt{SMILE}$

Relative clauses (3): full derivation

$$\boxed{\begin{array}{c} \textit{some} \cdot \textit{teacher} \cdot \textit{that} \cdot \textit{mary} \cdot \textit{praised} \cdot \textit{smiled} \\ \mathrm{SOME}(\mathrm{THAT}(\lambda u_e.\mathbf{praise}(u)(\mathbf{mary}))(\mathbf{teacher}))(\mathbf{smile}) \end{array}}$$

$$\mathtt{SOME}(\mathtt{THAT}(\lambda \mathtt{U}_{f,e}.\mathtt{PRAISE}(\mathtt{U})(\mathtt{MARY}))(\mathtt{TEACHER}))(\mathtt{SMILE})$$

$$\mathtt{SOME}(\mathtt{THAT}(\lambda \mathtt{U}_{f,e}.\mathtt{PRAISE}(\mathtt{U})(\mathtt{MARY}))(\mathtt{TEACHER})) \qquad \mathtt{SMILE}$$

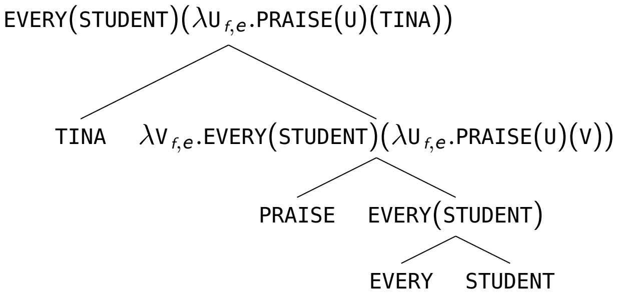
$$\mathtt{SOME} \qquad \mathtt{THAT}(\lambda \mathtt{U}_{f,e}.\mathtt{PRAISE}(\mathtt{U})(\mathtt{MARY}))(\mathtt{TEACHER})$$

$$\mathtt{TEACHER} \qquad \mathtt{THAT}(\lambda \mathtt{U}_{f,e}.\mathtt{PRAISE}(\mathtt{U})(\mathtt{MARY}))$$

$$\mathtt{THAT} \qquad \lambda \mathtt{U}_{f,e}.\mathtt{PRAISE}(\mathtt{U})(\mathtt{MARY})$$

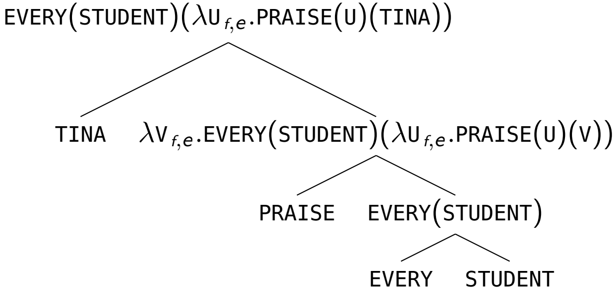
$$\mathtt{MARY} \qquad \mathtt{PRAISE}$$

Quantificational object noun phrases

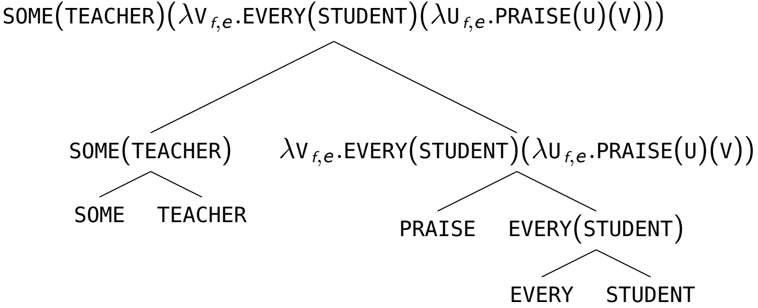


Quantificational object noun phrases

$$\boxed{\begin{array}{c} \textit{tina} \cdot \textit{praised} \cdot \textit{every} \cdot \textit{student} \\ \text{EVERY}(\textbf{student})(\lambda u_e.\textbf{praise}(u)(\textbf{tina})) \end{array}}$$



Quantifier scope (1): object narrow scope



Quantifier scope (1): object narrow scope

$$\boxed{\begin{array}{c} \textit{some} \cdot \textit{teacher} \cdot \textit{praised} \cdot \textit{every} \cdot \textit{student} \\ \text{SOME}(\mathbf{teacher})(\lambda v_e.\text{EVERY}(\mathbf{student})(\lambda u_e.\mathbf{praise}(u)(v))) \end{array}}$$

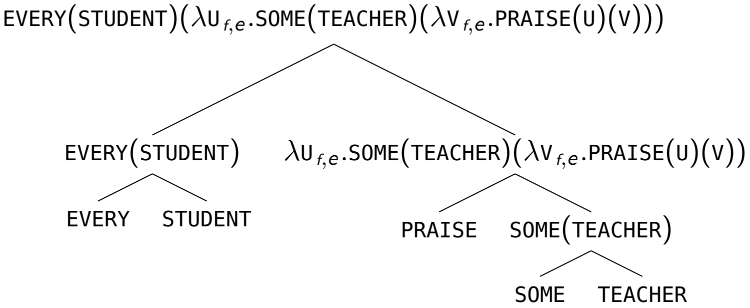
$\mathtt{SOME}(\mathtt{TEACHER})(\lambda \mathtt{V}_{f,e}.\mathtt{EVERY}(\mathtt{STUDENT})(\lambda \mathtt{U}_{f,e}.\mathtt{PRAISE}(\mathtt{U})(\mathtt{V})))$

$\mathtt{SOME}(\mathtt{TEACHER})$ $\quad$ $\lambda \mathtt{V}_{f,e}.\mathtt{EVERY}(\mathtt{STUDENT})(\lambda \mathtt{U}_{f,e}.\mathtt{PRAISE}(\mathtt{U})(\mathtt{V}))$

$\mathtt{SOME}$ $\quad$ $\mathtt{TEACHER}$ $\quad$ $\mathtt{PRAISE}$ $\quad$ $\mathtt{EVERY}(\mathtt{STUDENT})$

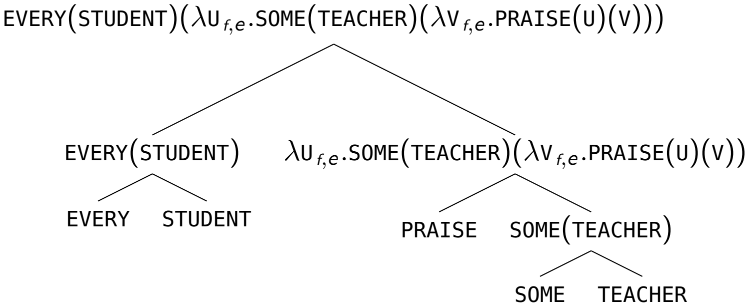
$\mathtt{EVERY}$ $\quad$ $\mathtt{STUDENT}$

Quantifier scope (2): object wide scope



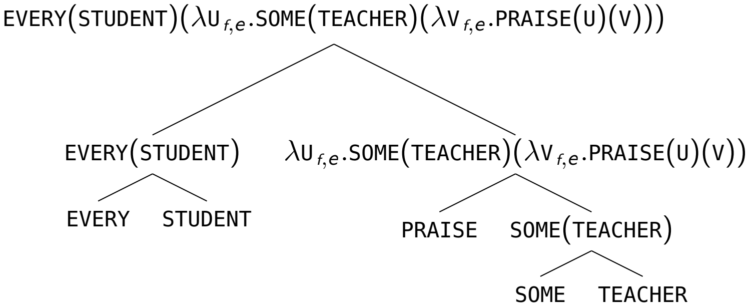
Quantifier scope (2): object wide scope

$$\boxed{\begin{array}{c} \textit{some} \cdot \textit{teacher} \cdot \textit{praised} \cdot \textit{every} \cdot \textit{student} \\ \text{EVERY}(\textbf{student})(\lambda u_e.\text{SOME}(\textbf{teacher})(\lambda v_e.\textbf{praise}(u)(v))) \end{array}}$$



Quantifier scope (2): object wide scope

$$\begin{array}{c} \textit{some} \cdot \textit{teacher} \cdot \textit{praised} \cdot \textit{every} \cdot \textit{student} \\ \text{EVERY}(\textbf{student})(\lambda u_e.\text{SOME}(\textbf{teacher})(\lambda v_e.\textbf{praise}(u)(v))) \end{array}$$



Two parameters:

- Order of composition of signs – determines semantic scope
- Sign argument saturated – determines syntactic position

Summary

- Lambek-Van Benthem Calculus – flexibility of hypothetical reasoning
- Directionality is not in tecto-level syntax, but in the pheno-level objects that it manipulates
- Saussurean signs – avoiding overgeneration
- Implications:
  - Modeltheoretic phonology
  - Free variables in grammar, not in meaning
  - Syntax and semantics hand in hand

Further usages

By extending the framework with *possible worlds*, scope mechanisms as in ACG can also deal with *de dicto*/*de re* ambiguities, such as:

Mary is looking for a secretary.

References

Curry, H. B. (1961), Some logical aspects of grammatical structure, *in* R. O. Jakobson, ed., 'Structure of Language and its Mathematical Aspects', Vol. 12 of *Symposia on Applied Mathematics*, American Mathematical Society, Providence.

de Groote, P. (2001), Towards abstract categorial grammars, *in* 'Proceedings of the 39th annual meeting of the Association for Computational Linguistics (ACL)'.

de Saussure, F. (1959), *Course in General Linguistics*, Philosophical Library, New York. Translation of *Cours de Linguistique Générale*, Payot & Cie, Paris, 1916.

Lambek, J. (1958), 'The mathematics of sentence structure', *American Mathematical Monthly* **65**, 154–169.

Muskens, R. (2003), Language, Lambdas, and Logic, *in* G.-J. Kruijff & R. Oehrle, eds, 'Resource Sensitivity in Binding and Anaphora', Studies in Linguistics and Philosophy, Kluwer, pp. 23–54.

van Benthem, J. (1991), *Language in Action: categories, lambdas and dynamic logic*, North-Holland, Amsterdam.