Semantics 1

May 10, 2012

Gerhard Jäger

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Type driven interpretation

Regelformate

- so far, we had three types of semantic rules:
  - $X \to Y, Z :: \|X\| = \|Y\|(\|Z\|)$
  - $X \to Y, Z :: \|X\| = \|Z\|(\|Y\|)$
  - $X \to Y, Z, W :: \|X\| = \|Z\|(\|Y\|)(\|W\|)$
- Commonalities:
  - one element on the right hand side denotes a function
  - the other elements on the right hand side denote arguments for this function
  - meaning of the mother node: result of applying the function to its arguments
  - **semantic operation is always function application**
  - **There is always exactly one way hot wo apply the meaning of the daughter node to the meaning(s) of the other daughter node(s).**

$\Rightarrow$ semantic operation is determined by domain of the functions involved

Type driven interpretation

- type of a function: *domain, range*
- general semantic composition rule:

Principle of type driven interpretation
The meaning of the mother node is the result of applying the meaning of one of the daughter nodes to the meaning(s) of the other daughter node(s). Due to the types of the functions involved, this operation is always uniquely defined.

- semantic rule is always uniquely defined by syntactic rule

$\leadsto$ semantic rules are redundant

Argument structure and λ-prefixes

- verbs — examples:
  - *rain* $\rightsquigarrow \lambda s.\text{RAIN'}(s)$
  - *sleep* $\rightsquigarrow \lambda x \lambda s.\text{SLEEP'}(s, x)$
  - *read* $\rightsquigarrow \lambda y \lambda x \lambda s.\text{READ'}(s, x, y)$
  - *give* $\rightsquigarrow \lambda z \lambda y \lambda x \lambda s.\text{GIVE'}(s, x, y, z)$
- pattern: The interpretation of an n-place verb always has $n + 1$-many λs (one λ per argument place, plus one λ for the situation variable).
- argument structure can be read off from the meaning

Diathesis and lexical rules

Indefinite ellipsis

- for some transitive verbs, the object can be omitted, e.g.
  - *Peter read Anna Karenina.* $\Rightarrow$
  - *Peter read.*
- Elided sentence always follows logically from non-elided version

Diathesis and lexical rules

Indefinite ellipsis

- There are two verbs *read*, a transitive and an intransitive one. They are semantically related.
- **Lexical Rule:** If V is a transitive verbs with the meaning α, then V is also an intransitive verb with the meaning $\lambda x \lambda s.\exists y(\alpha(y)(x)(s))$
- hence:
  - meaning of transitive *read*: $\lambda y \lambda x \lambda s.\text{READ'}(s, x, y)$
  - meaning of *read* as an intransitive verb is

$$\lambda x \lambda s.\exists y(\text{READ'}(s, x, y))$$

Indefinite ellipsis

- S: $\lambda s.\text{READ'}(s, \text{P'}, \text{A'})$
  - NP: P'
    - N: P'
      - Peter
  - VP: $\lambda x \lambda s.\text{READ'}(s, x, \text{A'})$
    - V: $\lambda y \lambda x \lambda s.\text{READ'}(s, x, y)$
      - liest
    - NP: A'
      - N: A'
        - Anna Karenina

Diathesis and lexical rules

Indefinite ellipsis

S

$\lambda s.\exists y(\text{READ'}(s, \text{P'}, y))$

NP

P'

N

P'

Peter

VP

$\lambda x \lambda s.\exists y(\text{READ'}(s, x, y))$

V

$\lambda x \lambda s.\exists y(\text{READ'}(s, x, y))$

liest

$$\|\textit{Peter read Anna Karenina}\| \subseteq \|\textit{Peter read}\|$$

$$\textit{Peter read Anna Karenina} \Rightarrow \textit{Peter read}$$

Diathesis and lexical rules

Passive

- Passive:
  - *Peter read Anna Karenina*
  - *Anna Karenina was read*
- Passive transforms a transitive (two-place) verb into an intransitive (one-place) participle.
- For syntactic reasons, participle must co-occur with an auxiliary verb.

Diathesis and lexical rules

Passive

- **Lexical Rule:** If V is a transitive veb with the meaning α, then the past participle of V has the meaning $\lambda x \lambda s.\exists y(\alpha(x)(y)(s))$
- $\|read_{prtc}\| = \lambda x \lambda s.\exists y(\text{READ'}(s, y, x))$
- The auxiliary does not contribute anything to the meaning:[1]

$$\|\textit{is/was}\| = \lambda P \lambda x.P(x)$$

- syntactic category of auxiliaries: T
- Syntactic Rule:

$$S \rightarrow NP, T, VP$$

[1] Apart from tense and mood information, which we ignore for the time being.

Diathesis and lexical rules

Passive

S

$\lambda s.\exists y(\text{READ'}(s, y, \text{A'}))$

NP	T	VP
A'	$\lambda P \lambda x.P(x)$	$\lambda x \lambda s.\exists y(\text{READ'}(s, y, x))$
N	was	V
A'		$\lambda x \lambda s.\exists y(\text{READ'}(s, y, x))$
Anna Karenina		read

$\|$*Peter read Anna Karenina*$\| \subseteq \|$*Anna Karenina was read*$\|$

Peter read Anna Karenina $\Rightarrow$ *Anna Karenina was read.*

Quantifiers

Introduction

- So far, we only had one class of NPs: proper nouns (*Peter, John, Anna Karenina, ...*)
- There are many other NPs in English:
  - *nobody, everybody, somebody, ...*
  - *every woman, some women, most women, three women, a woman, many women, few women, the three women*
- such NPs are called **generalized quantifiers** (or simply **quantifiers**, when no confusion with the quantifiers of logic can arise)

Quantifiers

Generalized Quantifiers Certain inference patterns that hold for proper nouns do not hold for GQs:

(1) a. Hans read Anna Karenina $\Rightarrow$ Anna Karenina was read.
    b. Nobody read Anna Karenina $\not\Rightarrow$ Anna Karenina was read.

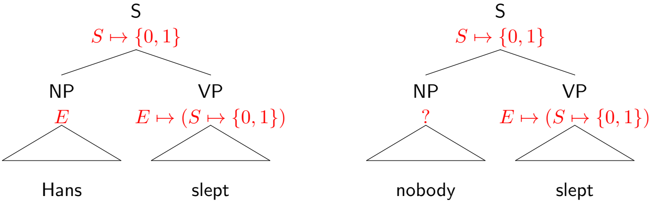
(2) a. Hans knows Anna and Hans likes Maria $\Leftrightarrow$ Hans likes Anna and likes Maria.
    b. A man knows Anna and a man likes Maria $\not\Leftrightarrow$ A man knows Anna and likes Maria.

(3) a. Hans knows Anna or Hans likes Maria $\Leftrightarrow$ Hans knows Anna or likes Maria.
    b. Every man knows Anna or every man likes Maria $\not\Leftrightarrow$ Every man knows Anna or likes Maria.

Generalized Quantifiers

- If the meaning of GQs was an individual, these inference patterns should hold!

$\leadsto$ Meaning of a GQ is not an individual.



Quantifiers

Generalized Quantifiers

- If meaning composition is driven by function application, the meaning of a quantifier must have the following type:

$$(E \mapsto (S \mapsto \{0,1\})) \mapsto (S \mapsto \{0,1\})$$

  i.e., a function from VP meanings to sentence meanings

- If we implicitly assume Schönfinkelization and the equivalence of sets and their characteristic functions, is is equivalent to *properties of properties*:

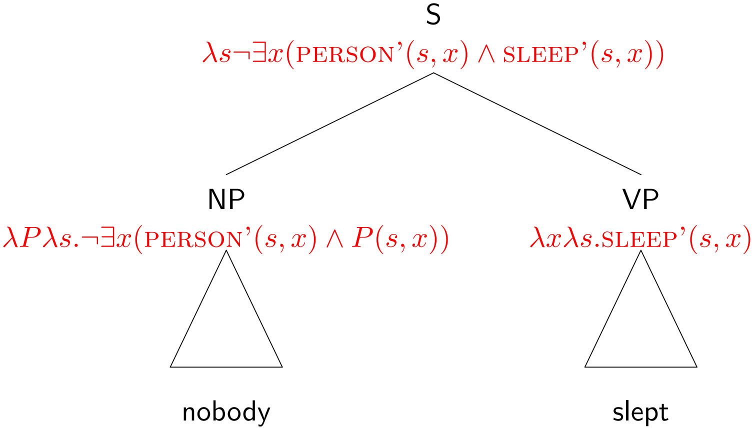
$$POW(S \times POW(S \times E))$$

Quantifiers

Generalized Quantifiers

- meaning of some GQs:
  - every, alls: $\lambda P \lambda s.\forall x(\text{PERSON'}(s,x) \rightarrow P(s,x))$
  - nobody: $\lambda P \lambda s.\neg\exists x(\text{PERSON'}(s,x) \wedge P(s,x))$
  - somebody: $\lambda P \lambda s.\exists x(\text{PERSON'}(s,x) \wedge P(s,x))$
- General pattern: the meaning of a quantifier is obtained by
  - starting with the meaning of sentence with the quantifier in question as subject,
  - replacing the VP meaning by a variable, and
  - λ-abstracting over that variable.

Generalized Quantifiers



Generalisierte Quantifiers

S

$\lambda s \exists x(\textsc{person}'(s, x) \wedge \textsc{sleep}'(s, x))$

NP

$\lambda P \lambda s.\exists x(\textsc{person}'(s, x) \wedge P(s, x))$

somebody

VP

$\lambda x \lambda s.\textsc{sleep}'(s, x)$

slept

Generalized Quantifiers

S

$\lambda s \forall x(\text{PERSON'}(s,x) \rightarrow \text{SLEEP'}(s,x))$

NP

$\lambda P \lambda s.\forall x(\text{PERSON'}(s,x) \rightarrow P(s,x))$

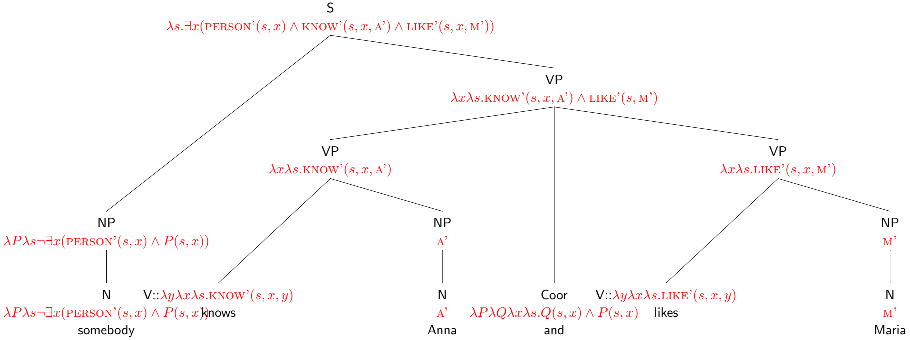
everybody

VP

$\lambda x \lambda s.\text{SLEEP'}(s,x)$

slept

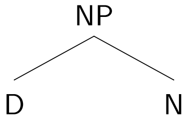
Generalized Quantifiers



Quantifiers

Determiner

- How do we compute the meaning of syntactically complex GQs?



- meaning of a noun: property of entities (just like intransitive verbs)
  $\rightsquigarrow$ subset of $S \times E$, i.e., an element of $E \mapsto (S \mapsto \{0,1\})$
- meaning of a determiner: function from noun meaning to GQ meaning

$$(E \mapsto (S \mapsto \{0,1\})) \mapsto (E \mapsto (S \mapsto \{0,1\})) \mapsto (S \mapsto \{0,1\})$$

- equivalent to

$$POW(S \times POW(S \times E) \times POW(S \times E))$$