Big Picture

Semantics – how do we figure out the situations in which sentences are true or false?
Compositionality = pieces + composing them

• **Lexical semantics** = what do we know about word meanings?

• **Compositional semantics** = how do we put the pieces together?
Compositional semantics

So far:

• **Sentence-meaning**
  • truth, truth-conditions, possible worlds

• **Meaning of NPs** (noun phrases)
  • constant/variable reference, naming game

• **Meaning of predicates**
  (verbs, nouns, adjectives)
  • set theory, relations, functions

• **Putting things together**
  • lambdas, types
Compositional semantics: NPs

From now on: More about NPs & predicates
Keep comparing theory & data!

• NPs that don’t refer to objects
  sets of sets, patterns of meaning: polarity

• Different types of NPs, & what they do
  definite, indefinite, quantificational

• A unified theory
  kinds, objects, mass, count, different languages
To motivate further theory

Either John is in that room or Mary is, and possibly they both are.

• What are some problems in translating this into predicate logic?
To motivate further theory

(1) Either John is in that room or Mary is, and possibly they both are.

• What are some problems in translating this into predicate logic?

Stuff we need: “that” (to make “that room”) representing “either...or” “possibly”
(2) Sam wants a dog, but Alice wants cats
(3) A dog is a quadruped

• *What are some problems with translating this?*
(2) Sam wants a dog, but Alice wants cats
(3) A dog is a quadruped

• What are some problems with translating this?

Stuff we need: plural vs singular phrases
what to do with bare plurals?
Is “a dog” ambiguous?
“but” vs. “and”
representing generic meanings
New kind of ambiguity?
Scope ambiguity

Lexical or structural?

- Sam wants a dog
- Everything is black or white
- Someone loves everyone
Semantic theory so far:

- Sentence = predicate saturated with all its arguments (so, smth True or False)
- Sentences can be composed from other sentences using “no”, “and”, “or”, ”if-then”
- Predicates can have arity=valency of zero (to rain), one (to run), two (to devour), three (to give) etc. arguments.
- Arguments can be of any type, including entities, other predicates, and whole sentences
Semantic theory so far (cont’d):

- NPs can represent entities (John), predicates (“a dog” in “Fido is a dog”), or expressions with quantifiers (“a dog” in “Sam wants a dog”)

- We can make new predicate expressions using lambdas; also new semantic rules.
Semantic rules

Lambda abstraction

- Used when something moves
  \[ John \lambda x \ I \ like \ x = I \ like \ John \]

- Used for making relative clauses & questions
  \[ Who \lambda x x \ does \ it = set \ of \ people \ who \ do \ it \]

- Used for representing predicates
  \[ Not \ smoking \ is \ healthy = Healthy (\lambda x \lnot \ smoke(x)) \]
Semantic rules (cont’d)

Function application:
● Used to put predicates and arguments together

  John runs

  John \( \lambda x \ I \ like \ x \)

  Someone runs

Conjunction and other ‘connective’ rules:
● Take predicates that you want to conjoin
● Fully saturate them using variables
● Conjoin the resulting sentences
● Lambda abstract over the variables to get the new predicate of correct type
Generalised Quantifiers

- Try applying conjunction schema to “John and Mary”
- What is the type of these expressions?
  “Every guy but John”
  “Some apples and this pear”
Even worse: one might initially think that a unicorn is referential (refers to a particular individual) e.g., *A unicorn was there. He was beautiful.*

However, as Bertrand Russel noted, indefinites are also non-referential: *Nobody has seen a unicorn, because there aren't any.*
Generalised Quantifier Theory

- Basic idea:
  All NPs are of the same type
  - each is a set of sets.

\[[Jane] = \{\text{snore, run, talk, girl}\}\]
Generalised Quantifier Theory

- Basic idea:
  All NPs are of the same type
  – each is a set of sets.

$$[[\text{Jane}]] = \{\text{snore, run, talk, girl}\}$$

$$\lambda P. P(j)$$
Generalised Quantifier Theory

• Basic idea:
  All NPs are of the same type
  – each is a **set of sets**.

\[
[[\text{Jane}]] = \{\text{snore, run, talk, girl}\}
\]

\[
\lambda P. P(j) \quad (e \rightarrow t) \rightarrow t
\]
Syntax and Semantics

\[ \text{[Jane snores]} = \lambda P. P(j) \quad (\text{snore}) = (e \rightarrow t) \rightarrow t \quad e \rightarrow t \]
Syntax and Semantics

\[ \text{\{Jane snores\}} = \lambda P.P(j) \quad (\text{snore}) = \text{snore} (j) \]
\[ (e \rightarrow t) \rightarrow t \quad e \rightarrow t \quad t \]
Syntax and Semantics

Mismatch for syntax and semantics:
- What's the argument?
- What's the predicate?
- What is the constituent structure?
- Which individuals matter for the truth of S?

\[
\text{[[Every student danced]]} = \text{Every } x \ [ \text{student}(x) \rightarrow \text{danced}(x) ]
\]


**Syntax and Semantics**

**In English:**

- "Every student" is a unit
- It combines with "danced"

**In PC:**

- a totally different tree
- \( \forall x \ [ \text{student}(x) \rightarrow \text{danced}(x) ] \)
- \( x \ [ \text{student}(x) \rightarrow \text{danced}(x) ] \)
- \( \text{student}(x) \ \text{danced}(x) \)

- \([ [ \text{every student} ] \) and \([ \text{danced} ] \) are not constituents!
- "student danced" is a unit
- It combines with "every"
Syntax and Semantics

**In English:**

- Look in the **set of students**
  - If all members of this set danced – T
  - If not all members of this set danced – F

**In PC:**

- Look at all the entities in the **universe**
  - If the entity is not a student, T
  - If the entity is a student, then if this entity danced – T
  otherwise - F
GQ Theory

- Try semantics which is more true to syntax:

$$[[\text{Every student}]] = \{\text{dance, run, talk, student}\}$$

$$\lambda P. \text{Every(student)}(P) \quad (e \rightarrow t) \rightarrow t$$

What’s “every”? Something that combines with “student” to make “every student”
GQ Theory

- What a determiner might mean:

  "every" - something that combines with "student"

\[ e \rightarrow t \]

...to make "every student"

\[ (e \rightarrow t) \rightarrow t \]
GQ Theory

- What a determiner might mean:
  
  "every" - something that combines with "student" \( e \rightarrow t \)

\[
\lambda Q_e \rightarrow t.
\]

to make "every student"

\[
\lambda P_e \rightarrow t.\ Every(\text{student})(P)
\]

Every student danced
GQ Theory

- “every” - combines with
  
  “student” $e \rightarrow t$ $\lambda Q_{e \rightarrow t}$ to make

  “every student” $\lambda P_{e \rightarrow t} . \text{Every}(\text{student})(P)$

- SO: $[[\text{Every}]] =$

  $\lambda Q \lambda P . \text{Every}(Q)(P)$

  $(e \rightarrow t) \rightarrow ((e \rightarrow t) \rightarrow t)$

---

Student

Dance

Every student danced
GQ Theory

- Try semantics which is more true to syntax:

\[
\text{Every student danced} = (e \rightarrow t) \rightarrow ((e \rightarrow t) \rightarrow t)
\]
GQ Theory

• Try semantics which is more true to syntax:

\[
\lambda Q \lambda P. \text{Every}(Q)(P) \quad \text{student}' \quad \text{danced}'
\]
GQ Theory

- Try semantics which is more true to syntax:

\[
\text{[[Every student danced]]} =
\[(e \rightarrow t) \rightarrow ((e \rightarrow t) \rightarrow t) \quad e \rightarrow t \quad e \rightarrow t
\]
\[
\lambda Q \lambda P. \text{Every}(Q)(P) \quad \text{student}' \quad \text{danced}'
\]
GQ Theory

- Try semantics which is more true to syntax:

\[
\begin{array}{c}
\text{S} \\
\text{NP} \quad \text{VP} \\
\text{Det} \quad \text{N'} \quad \text{V} \\
\text{Every} \quad \text{student} \quad \text{danced} \\
\end{array}
\]

\[
\begin{array}{c}
(e\to t)\to((e\to t)\to t) \\
\lambda Q\lambda P.Every(Q)(P) \\
\lambda Q\lambda P.\forall x(Q(x)\to P(x)) \\
\lambda x.\text{student'}(x) \\
\lambda y.\text{danced'}(y)
\end{array}
\]

\[
\begin{array}{c}
\text{e}\to t \\
\text{student'} \\
\text{danced'}
\end{array}
\]
Several GQs

Every student

Some students

Student

Jane

j
# Generalised Quantifier Theory

## Several GQs
- $[\text{All NP}] = [\text{All } (A,B)]$
- $[\text{Some NP}] = [\text{Some } (A,B)]$
- $[\text{No NP}] = [\text{No } (A,B)]$
- $[\text{At least 5 NP}] = [\text{At least 5 } (A,B)]$
- $[\text{Most NP}] = [\text{Most } (A,B)]$

## Several Determiners
- $[\text{All } (A,B)]$
- $[\text{Some } (A,B)]$
- $[\text{No } (A,B)]$
- $[\text{At least 5 } (A,B)]$
- $[\text{Most } (A,B)]$
Generalised Quantifier Theory

Several GQs

• \[[\text{All NP}]\] = All Ling 130 students are smart.

Several Determiners

\[[\text{All (A,B)}]\] =
Generalised Quantifier Theory

Several GQs

- $[\text{All NP}] = \{X \subseteq U \mid A \subseteq X\}$
- $[\text{Some NP}] = \{\}$

Several Determiners

- $[\text{All (A,B)}] = 1$ iff $A \subseteq B$
- $[\text{Some (A,B)}] = \{\}$

Some Brandeis students commute.
Generalised Quantifier Theory

Several GQs

- \([\text{All NP}] = \{X \subseteq U \mid A \subseteq X\}\)
- \([\text{Some NP}] = \{X \subseteq U \mid X \cap A \neq \emptyset\}\)
- \([\text{No NP}] = \)

Several Determiners

- \([\text{All } (A, B)] = 1 \text{ iff } A \subseteq B\)
- \([\text{Some } (A, B)] = 1 \text{ iff } A \cap B \neq \emptyset\)
- \([\text{No } (A, B)] = \)

No boy(s) came to the party.
Generalised Quantifier Theory

Several GQs

- \([\text{All NP}] = \{ X \subseteq U \mid A \subseteq X \}\)
  - \([\text{Some NP}] = \{ X \subseteq U \mid X \cap A \neq \emptyset \}\)
  - \([\text{No NP}] = \{ X \subseteq U \mid X \cap A = \emptyset \}\)
  - \([\text{At least 5 NP}] = \)

Several Determiners

- \([\text{All (A,B)}] = 1 \text{ iff } A \subseteq B\)
  - \([\text{Some (A,B)}] = 1 \text{ iff } A \cap B \neq \emptyset\)
  - \([\text{No (A,B)}] = 1 \text{ iff } A \cap B = \emptyset\)
  - \([\text{At least 5 (A,B)}] = \)

At least 5 ballerinas danced there.
Generalised Quantifier Theory

Several GQs

- \([\text{All NP}] = \{X \subseteq U \mid A \subseteq X\}\)
- \([\text{Some NP}] = \{X \subseteq U \mid X \cap A \neq \emptyset\}\)
- \([\text{No NP}] = \{X \subseteq U \mid X \cap A = \emptyset\}\)
- \([\text{At least 5 NP}] = \{X \subseteq U \mid |X \cap A| \geq 5\}\)
- \([\text{Most NP}] = \frac{|X \subseteq U \mid |X \cap A| \geq \text{some threshold}|}{|X \subseteq U|}\)

Several Determiners

- \([\text{All } (A,B)] = 1 \text{ iff } A \subseteq B\)
- \([\text{Some } (A,B)] = 1 \text{ iff } A \cap B \neq \emptyset\)
- \([\text{No } (A,B)] = 1 \text{ iff } A \cap B = \emptyset\)
- \([\text{At least 5 } (A,B)] = 1 \text{ iff } |A \cap B| \geq 5\)
- \([\text{Most } (A,B)] = \frac{|A \subseteq B \mid |A \cap B| \geq \text{some threshold}|}{|A \subseteq B|}\)

Most Brandeis students live on campus.
# Generalised Quantifier Theory

## Several GQs

- **[All NP]**: \( \{ X \subseteq U \mid A \subseteq X \} \)
- **[Some NP]**: \( \{ X \subseteq U \mid X \cap A \neq \emptyset \} \)
- **[No NP]**: \( \{ X \subseteq U \mid X \cap A = \emptyset \} \)
- **[At least 5 NP]**: \( \{ X \subseteq U \mid |X \cap A| \geq 5 \} \)
- **[Most NP]**: \( \{ X \subseteq U \mid |X \cap A| > \frac{1}{2} |A| \} \)

## Several Determiners

- **[All (A,B)]**: \( 1 \text{ iff } A \subseteq B \)
- **[Some (A,B)]**: \( 1 \text{ iff } A \cap B \neq \emptyset \)
- **[No (A,B)]**: \( 1 \text{ iff } A \cap B = \emptyset \)
- **[At least 5 (A,B)]**: \( 1 \text{ iff } |A \cap B| \geq 5 \)
- **[Most (A,B)]**: \( 1 \text{ iff } |A \cap B| > |A - B| \)