LIN241

Introduction to Semantics

Lecture 6

Predicate Logic 1

What we missed in Propositional Logic

- Argument structure of sentences of English
  - **Gertrude likes Marina** vs **Marina likes Gertrude**
- Quantification
  - **Every human is mortal** vs **Some human is mortal**

What we missed in Propositional Logic

- A valid argument with quantifiers:

  Every human is mortal.

  Socrates is human.

  $\therefore$ **Socrates is mortal.**

What we missed in Propositional Logic

- An invalid argument with quantifiers:

  Some humans are mortal.

  Socrates is human.

  ∴ **Socrates is mortal.**

What we missed in Propositional Logic

- A valid argument with argument structure:

  Kingston is in Ontario.

  Ontario is in Canada.

  ∴ **Kingston is in Canada.**

What we missed in Propositional Logic

- An invalid argument with argument structure:

  Kingston is in Canada.

  Ontario is in Canada.

  ∴ **Kingston is in Ontario.**

Plan

- Today:
  - Basics of Predicate Logic, without quantifiers
  - Application to compositional semantics
- Next week
  - Quantification

Informal Introduction to Predicate Logic

Predicates and Arguments

- In English, intransitive verbs combine with a subject:
  - **Kelly wrestles.**
- The verb is a predicate, and the subject is its argument:
- This structure is preserved in Predicate Logic (PredL):
  - WRESTLE(k)
- By convention, we will write predicates in capital letters.
- The arguments of a predicate are written between parentheses after the predicate.

Predicates and Arguments

- With transitive verbs, we have both a subject and an object:
  - **Kelly knows Henry.**
- Here, the subject and object are both arguments of the verb.
- In PredL, we write:
  - KNOW(k, h)
- In the list of arguments, we put the active voice subject first, and the active voice object second.

Predicates and Arguments

- In the following formula, we say that 'h' and 'l' are individual constants:
  - KNOW(k, h)
- Individual constants are expressions that denote individuals, and whose denotation is not variable.
- We translate proper names using individual constants
  - E.g. we can translate **Francis** as f

Types of Predicates

- Like verbs of English, predicates of PredL can have different numbers of arguments.
    - Unary predicates take one argument, like intransitive verbs.
    - Binary predicates take two arguments, like transitive verbs.
    - In general, n-ary predicates take n-arguments.
- The number of arguments a predicate takes is called its **arity**.

Types of Predicates

- Predicates of PredL are not used only to translate verbs.
- They are also translations of any other predicative words of English, e.g. predicative nouns and predicative adjectives:
    - **Marilyn is a chemist.**

      CHEMIST(m)
    - **Marilyn is Irish.**

      IRISH(m)

Statements and sentence connectives

- A predicate with all its arguments forms a statement, which has a truth-value:

  CHEMIST(m)

  IRISH(m)

- We can combine such statements with the connectives of Propositional Logic:

  IRISH(m) & CHEMIST(m)

  LINGUIST(m) → KNOW(m,n)

Translating English into PredL

Basic Principles

- For simple active sentences without quantifiers:
  - Translate the verb, nominal predicate or adjectival predicate.
  - Translate proper names as individual constants.
  - Put the translation of the subject as the first argument of the predicate.
  - Put the translation of the object as the second argument of the predicate.

Basic Principles

- For passive sentences without quantifiers:
  - Use the same principles, but reverse the order of arguments.
- How would you translate the following sentences?
  - **Joana kicked Ana.**
  - **Pia helped Jess.**
  - **Jess was helped by Pia.**

Basic Principles

- How would you translate the following sentences?
    - **Joana kicked Ana.**

      KICK(j, a)
    - **Pia helped Jess.**

      HELP(p, j)
    - **Jess was helped by Pia.**

      HELP(p, j)

Basic Principles

- When a noun phrase is introduced by a preposition, but it is an argument of the verb, we ignore the preposition in the translation:
  - **Marcus talked to Alexandra.**

    TALK(m,a)
  - **Alexandra introduced Tara to Mika.**

    INTRODUCE(a,t,m)

Basic Set Theory

What is a set?

- A set is a collection of objects.
- The simplest way to define a set is simply to give a list of its members, also called its elements.
- We do this using curly brackets.
- This is the set whose members are the numbers 1, 2 and 3:
  - {1, 2, 3}
- This is called the list notation for sets.

Defining sets

- We can put anything inside a set:

    1. {a, g, t}
    2. {1, 9, 234}
    3. {John, Mary, Sue}
    4. {a, John, 9}

Defining sets

- The order of members of a set doesn't matter.
- The four sets that follow are the same set:

    1. {a, g, t}
    2. {g, a, t}
    3. {g, t, a}
    4. {a, t, g}

- That is to say:

  {a, g, t} = {g, a, t} = {g, t, a} = {a, t, g}

Defining sets

- Repeating a member of a set doesn't create a new set.
- The four sets that follow are the same set:

  1. {a, g, t}
  2. {a, g, t, t, t}
  3. {a, g, t, a}
  4. {a, a, t, g}

- That is to say:

  {a, g, t} = {a, g, t, t, t} = {a, g, t, a} = {a, a, t, g}

Membership

- Membership is a relation between a set and something that is a member of this set.
- The relation is written using the symbol '$\in$', which is a stylized version of the Greek letter epsilon.
- The following statements are equivalent:
    - 3 is a member of {1, 2, 3}
    - $3 \in \{1, 2, 3\}$

Membership

- We may want to say that something is not a member of a set.
- We express this relation with the symbol '$\notin$'.
- The following statements are equivalent:
  - 4 is not a member of {1, 2, 3}
  - $4 \notin \{1, 2, 3\}$

Predicate notation

- Another way to define sets is using predicate notation, like this:
  - {x : x is a natural number}

    (the set of all natural numbers)
- This notation should be read as follows:
  - 'the set of every x such that x is a natural number'
- Here, 'x' is a variable, that can take many different values:
  - we will use the letters x, y and z for variables

Predicate notation

- Be mindful of the way the notation is used:
    - {x : x is a natural number}
- Before the colon symbol ':', we write a variable (x, y or z).
- After the colon symbol, we write a condition on this variable.
- A set defined in this way contains every value of the variable that satisfies this condition.

Identical sets

- Two sets are identical if and only if they have the same members.
- These two sets are identical:
  - {1, fork } = {fork, 1}
- And of course:
  - {1, fork } = { 1, fork}

Binary relations

- Binary relations are relations that hold between two things.
- The following expressions of English express binary relations:
  - **like**, **be greater than**, **inside**
- We represent binary relations as sets of ordered pairs:
  - $\{ \langle 1,1 \rangle , \langle a,b \rangle , \langle 7,6 \rangle \}$
  - $\{ \langle x,y \rangle$: x and y are integers, and x is less than y$\}$
- Note that these pairs are ordered: $\langle 1,2 \rangle \neq \langle 2,1 \rangle$

Binary relations

- We can represent the relation of liking as follows:
  - { 〈x,y〉 : x likes y}
- This the set of all ordered pairs 〈x,y〉 such that x likes y.
- If we consider a set of individuals U, we can ask who likes who in this set:
  - this can be represented as a set of ordered pairs of individuals in U
  - this set is the relation of liking defined on U

Binary relations

- Let us assume that we have the following set of individuals:
  - U = {Alex, Chris, Jess}
- Let us assume that among the individuals in this set:
  - Jess likes Chris, Jess likes herself, Alex likes Chris, and nobody else likes anybody else.
- Then, the relation of liking defined on the set U is:
  - {〈Jess, Chris〉, 〈Jess, Jess〉, 〈 Alex, Chris〉}

Semantics of Predicate Logic

Models

- We interpreted statements of Propositional Logic with Truth Tables.
- In Predicate Logic, instead of Truth Tables, we will use models for Predicate Logic.
- Informally speaking, a model is an abstract representation of the world, or of a situation.
- A model contains individuals that we can organize in sets and relations.
- A model also contains information about the denotation of different expressions of Predicate Logic.

Models

- To interpret expressions of PredL in a model M, we will use:
  - a universe, written U, which is a set of individuals,
  - an interpretation $[\![\ ]\!]^M$, which tells us the **denotation** of expressions of PredL in the model.

Models

- In order to avoid confusion between individuals in U and their names in English (or other languages), we use the following conventions in these slides:
    - **Ian**: name of an individual, in English.
    - i: an individual constant, which can be used as the translation of **Ian** into PredL
    - Ian: an individual, which can be a member of U

Models

- Here is a simple model, call it M^1:
    - $U = \{Ian, Rebecca, Lola, Cooper\}$
    - $[\![i]\!]^{M^1}$ = Ian; $[\![r]\!]^{M^1}$ = Rebecca; $[\![l]\!]^{M^1}$ = Lola; $[\![c]\!]^{M^1}$ = Cooper
    - $[\![DOG]\!]^{M^1} = \{Cooper\}$
    - $[\![PERSON]\!]^{M^1} = \{Ian, Rebecca, Lola\}$
    - $[\![HAPPY]\!]^{M^1} = \{Cooper\}$
    - $[\![LIKE]\!]^{M^1} = \{\langle Cooper, Ian \rangle, \langle Cooper, Rebecca \rangle, \langle Cooper, Lola \rangle, \langle Cooper, Cooper \rangle\}$

Interpreting simple statements

- How do we know whether a statement is true in a model like this one?
  - DOG(c) is true in a model M iff $[\![c]\!]^M \in [\![DOG]\!]^M$
  - LIKE(c,i) is true in a model M iff $\langle [\![c]\!]^M, [\![i]\!]^M \rangle \in [\![LIKE]\!]^M$
- Where:
  - $[\![DOG]\!]^M$ is the set of dogs in M
  - $[\![LIKE]\!]^M$ is the set of pairs $\langle d, d' \rangle$ such that d likes d' in M.

Interpreting simple statements

- How would you translate the following statements into PredL, and are the translations true in M^1?
  - **Cooper likes Ian.**
  - **Cooper is a dog.**
  - **Ian is a dog.**

Interpreting simple statements

- Translations:
  - **Cooper likes Ian.**
    - LIKE(c,i)
  - **Cooper is a dog.**
    - DOG(c)
  - **Ian is a dog.**
    - DOG(i)

Interpreting simple statements

- The statement 'LIKE(c,i)' is true in M^1:
  - U = {Ian, Rebecca, Lola, Cooper}
  - $[\![i]\!]^{M^1}$ = Ian; $[\![r]\!]^{M^1}$ = Rebecca; $[\![l]\!]^{M^1}$ = Lola; $[\![c]\!]^{M^1}$ = Cooper
  - $[\![DOG]\!]^{M^1}$ = {Cooper}
  - $[\![PERSON]\!]^{M^1}$ = {Ian, Rebecca, Lola}
  - $[\![HAPPY]\!]^{M^1}$ = {Cooper}
  - $[\![LIKE]\!]^{M^1}$ = { ⟨Cooper, Ian⟩ , ⟨Cooper, Rebecca⟩ , ⟨Cooper, Lola⟩ , ⟨Cooper, Cooper⟩ }

Interpreting simple statements

- The statement 'DOG(c)' is true in M^1:
  - U = {Ian, Rebecca, Lola, Cooper}
  - $[\![i]\!]^{M^1}$ = Ian; $[\![r]\!]^{M^1}$ = Rebecca; $[\![l]\!]^{M^1}$ = Lola; $[\![c]\!]^{M^1}$ = Cooper
  - $[\![\text{DOG}]\!]^{M^1}$ = {Cooper}
  - $[\![\text{PERSON}]\!]^{M^1}$ = {Ian, Rebecca, Lola}
  - $[\![\text{HAPPY}]\!]^{M^1}$ = {Cooper}
  - $[\![\text{LIKE}]\!]^{M^1}$ = { ⟨Cooper, Ian⟩, ⟨Cooper, Rebecca⟩, ⟨Cooper, Lola⟩, ⟨Cooper, Cooper⟩ }

Interpreting simple statements

- The statement 'DOG(i)' is false in M^1:
  - U = {Ian, Rebecca, Lola, Cooper}
  - $[\![i]\!]^{M^1}$ = Ian; $[\![r]\!]^{M^1}$ = Rebecca; $[\![l]\!]^{M^1}$ = Lola; $[\![c]\!]^{M^1}$ = Cooper
  - $[\![\text{DOG}]\!]^{M^1}$ = {Cooper}
  - $[\![\text{PERSON}]\!]^{M^1}$ = {Ian, Rebecca, Lola}
  - $[\![\text{HAPPY}]\!]^{M^1}$ = {Cooper}
  - $[\![\text{LIKE}]\!]^{M^1}$ = { 〈Cooper, Ian〉, 〈Cooper, Rebecca〉, 〈Cooper, Lola〉, 〈Cooper, Cooper〉 }

Interpreting simple statements

- We can formulate general rules of interpretations for statements with individual constants:
  - If t is an individual constant and P is a unary predicate, then:

    $[\![P(t)]\!]^M = T$ iff $[\![t]\!]^M \in [\![P]\!]^M$

  - If t_1, ..., t_n are individual constants and R is an n-ary predicate with n ≥ 2, then:

    $[\![R(t_1, ..., t_n)]\!]^M = T$ iff $\langle [\![t_1]\!]^M, ..., [\![t_n]\!]^M \rangle \in [\![R]\!]^M$

Connectives

- Sentence connectives are interpreted in the same way as in propositional logic.

- Let F and F' be formulas for PredL:

  - $[\![\neg F]\!]^M = T$ iff $[\![F]\!]^M = F$

  - $[\![F \;\&\; F']\!]^M = T$ iff $[\![F]\!]^M = T$ and $[\![F']\!]^M = T$

  - $[\![F \vee F']\!]^M = T$ iff $[\![F]\!]^M = T$ or $[\![F']\!]^M = T$

  - $[\![F \rightarrow F']\!]^M = T$ iff $[\![F]\!]^M = F$ or $[\![F']\!]^M = T$

  - $[\![F \leftrightarrow F']\!]^M = T$ iff $[\![F]\!]^M = [\![F']\!]^M$

Appendix: compositional interpretation of English

Rules of Translation

- We could apply the semantics of PredL directly to English.
- In this appendix: a simple illustration of this procedure.

Proper Names

- A proper name like **Jess** is a type of Noun Phrase.
- We represent it as follows: $[_{NP}$ **Jess**$]$
- This NP denotes an individual:

$\llbracket\ [_{NP}\ \textbf{Jess}\ ]\ \rrbracket^{M}$ = Jess

Intransitive Verbs

- An intransitive verb like **smokes** is a type of Verb Phrase.
- We represent it as follows: $[_{VP}$ **smokes** $]$
- This VP denotes a set, for instance:

  $⟦ [_{VP} \textbf{smokes}] ⟧^M = \{\text{Jess, Charlie, Brian}\}$

Simple Sentences

- Typically, simple sentences consist of:
    - a Noun Phrase subject
    - a Verb Phrase predicate
- We represent this as follows: $[_S$ NP VP$]$
- The sentence is true if and only if the individual denoted by the subject NP is a member of the set denoted by the VP:
    - $\llbracket [_S \text{ NP VP}] \rrbracket^M = T$ iff $\llbracket \text{ NP } \rrbracket^M \in \llbracket \text{ VP } \rrbracket^M$

Simple Sentences

- Let S be $[_S\ [_{NP}$ **Jess**$]\ [_{VP}$ **smokes** $]\ ]$
- $[\![$ S $]\!]^M$ = T in the following model M:
    - U = {Jess, Lola, Cooper}
    - $[\![$**Jess**$]\!]^M$ = Jess; $[\![$**Lola**$]\!]^M$ = Lola; $[\![$**Cooper**$]\!]^M$ = Cooper
    - $[\![$**person**$]\!]^M$ = {Cooper, Jess, Lola}
    - $[\![$**smokes**$]\!]^M$ = {Jess, Cooper}
    - $[\![$**likes**$]\!]^M$ = { ⟨Cooper, Jess⟩ , ⟨Cooper, Cooper⟩ }

Transitive Verbs

- Transitive verbs denote sets of pairs of individuals.
- For instance the verb **likes** denotes:
  - the set of pairs of individuals 〈d, d'〉 such that d likes d'
- By combining a transitive verb with an object NP, we get a VP that denotes a set of individuals.
- For instance the VP **likes** denotes:
  - the set of individuals who like Jess.

Transitive Verbs

- We can formulate a rule of interpretation for transitive verbs:
  - $[\![\, [_{VP} \text{ V NP}] \,]\!]^M = \{x: \langle x, [\![\text{ NP }]\!]^M \rangle \in [\![\text{ V }]\!]^M \}$
- We can apply this rule to the VP **likes Jess**:
  - $[\![\, [_{VP} \text{ likes } [_{NP} \text{ Jess }] \,] \,]\!]^M =$
    $\{x: \langle x, [\![\text{ Jess }]\!]^M \rangle \in [\![\text{ likes }]\!]^M\}$

Transitive Verbs

- Let S be [$_S$ [$_{NP}$ **Cooper**] [$_{VP}$ **likes** [$_{NP}$ **Jess**]]]
- $[\![$ S $]\!]^M$ = T in the following model M:
    - U = {Jess, Lola, Cooper}
    - $[\![$**Jess**$]\!]^M$ = Jess; $[\![$**Lola**$]\!]^M$ = Lola; $[\![$**Cooper**$]\!]^M$ = Cooper
    - $[\![$**person**$]\!]^M$ = {Cooper, Jess, Lola}
    - $[\![$**smokes**$]\!]^M$ = {Jess, Cooper}
    - $[\![$**likes**$]\!]^M$ = { 〈Cooper, Jess〉, 〈Cooper, Cooper〉 }

Connectives

- We can import the rules for connectives in English:
    - $[\![$ **it is not the case that** S $]\!]^M$ = T iff $[\![$ S $]\!]^M$ = F
    - $[\![$ S **and** S' $]\!]^M$ = T iff $[\![$ S $]\!]^M$ = T and $[\![$ S' $]\!]^M$ = T
    - $[\![$ S **or** S' $]\!]^M$ = T iff $[\![$ S $]\!]^M$ = T or $[\![$ S' $]\!]^M$ = T
    - $[\![$ **if** S **then** S' $]\!]^M$ = T iff $[\![$ S $]\!]^M$ = F or $[\![$ S' $]\!]^M$ = T
    - $[\![$ S **if and only if** S' $]\!]^M$ = T iff $[\![$ S $]\!]^M$ = $[\![$ S' $]\!]^M$

Compositionality

- We have just sketched how we can use the semantics of PredL to build a compositional theory of natural language semantics:

Principle of Compositionality:
The meaning of a composite expression is a function of the meaning of its immediate constituents and the way these constituents are put together.