The Representation of Meaning

Sections 17.1 - 17.2 in Textbook

hannes@ru.is

T-725-MALV

Performing Language Tasks

- **Language** is central to many **tasks**
  - Answering a question
  - Interrogating a suspect
  - Summarizing a story
  - Learning from written instructions
  - Understanding a joke
- But those tasks cannnot be properly performed without a **link from linguistic elements to non-linguistic world knowledge**

Meaning Representations

- ...are formal structures that can capture the **meaning of linguistic expressions**
- Created through **semantic analysis** of linguistic input
- Facilitate further **semantic processing**
- Their syntax and semantics specified by **meaning representation languages**

Sample Representations

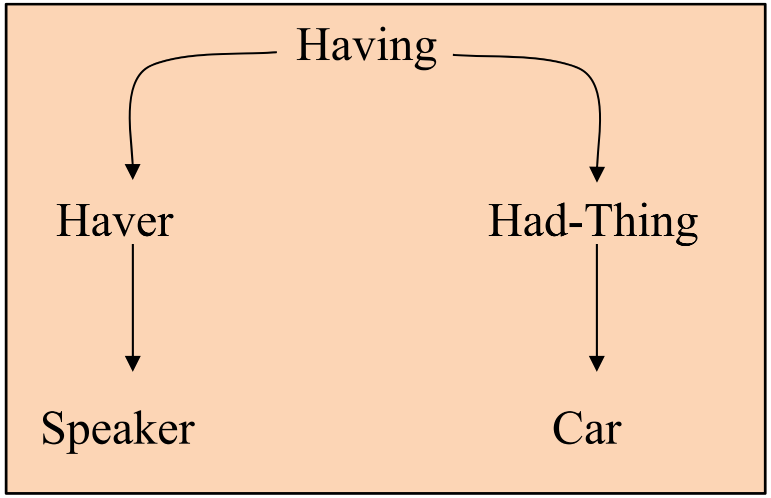
Linguistic input: "I have a car"

Meaning representation (4 examples):

$$\exists\ e,y\,Having(e) \wedge Haver(e,S) \wedge HadThing(e,y) \wedge Car(y)$$

Car
⇑ POSS-BY
Speaker

Having
Haver: Speaker
HadThing: Car



Meaning Representations

- All consists of **structures** composed from a **set of symbols** (vocabulary)
- When symbols are **properly arranged**, the structures **correspond to**
  - objects
  - properties of objects
  - relations among objects

in some **state of affairs being represented**

Two Perspectives

- Meaning Representations can be viewed as:

  (1) Representation of the **meaning of the linguistic input**

  (2) Representation of the **state of affairs** in some world

- This allows us to **link linguistic input to world**!

Literal Meaning

- For now we are only concerned with the literal meaning of sentences
  - **not** highly **contextual** or implied meaning, including idioms and metaphors

Representation Requirements

- To be useful, our meaning representation needs to **fulfil certain requirements**:
  - Be verifiable
  - Be unambiguous
  - Provide canonical form
  - Support inference
  - Support variables
  - Be expressive (enough)

Verifiability

- Can the meaning representation of a linguistic expression be **compared to a representation of state of affairs** in some world as modeled in a knowledge base?
- We need to be able to determine the truth of our representations

Does Maharani serve vegetarian food?

Serves(Maharani, Vegeterian) (Input Representation)

MATCHES

Serves(Maharani, Vegetarian) (Knowledge Representation)

Unambiguous

- **Linguistic** expressions can have **different meaning** representations assigned to them:

  I wanna eat someplace that's close to ICSI

- **Final representation** of an input's meaning should be **free from any ambiguity**
- **Vagueness** however can be useful, and therefore **ok**

  I want to eat Italian food

Canonical Form

- There may be multiple ways to express something, but this should result in **one common meaning representation**

1. Maharani has vegetarian dishes

2. They have vegetarian food at Maharani

Serves(Maharani, Vegetarian)

Inference

- Need to be able to **draw conclusions** about the truth of linguistic expressions that are not explicitly represented in the world state representation (knowledge base) but are **logically derivable** from it

Maharani serves vegetarian food (in knowledge base)

Vegetarians eat vegetarian food (in knowledbe base)

Can vegetarians eat at Maharani? (verifiable?)

Variables

- We may get references to **unknown** and unnamed entities

I'd like to find a restaurant serving vegetarian

Serves(x, Vegetarian)

Matching succeeds if **substitution** maintains truth of full proposition

Expressiveness

- Expressive enough to handle an extremely **wide range** of natural language expressions and subject matter

Model-Theoretic Semantics

- **Model** is a formal construct that stands for the **particular state** of affairs in the world that we are trying to represent
- We can **map expressions** in a meaning representation language **to elements** of the model
- Thus representations can do the work we require of them

Model-Theoretic Semantics

- Vocabulary of a meaning representation consists of
  - **non-logical vocabulary**: Open-ended set of names for the objects, properties and relations
  - **logical vocabulary**: Closed-set of symbols, operators, quantifiers, etc.

Model-Theoretic Semantics

- Each element of the non-logical vocabulary (objects, properties and relations) has a **denotation** (a fixed, well-defined part) in the model

Model-Theoretic Semantics

- The **domain of a model** is the set of **objects** that are part of the world being represented
  - **Objects** denote elements of the domain
  - **Properties** denote sets of elements of the domain
  - **Relations** donote sets of tuples of elements of the domain

Model Example

Domain
Matthew, Franco, Katie and Caroline
Frasca, Med, Rio
Italian, Mexican, Eclectic

$D = \{a,b,c,d,e,f,g,h,i,j\}$
a,b,c,d
e,f,g
h,i,j

Properties
Noisy
Frasca, Med, and Rio are noisy

$Noisy = \{e,f,g\}$

Relations
Likes
Matthew likes the Med
Katie likes the Med and Rio
Franco likes Frasca
Caroline likes the Med and Rio

$Likes = \{\langle a,f\rangle,\langle c,f\rangle,\langle c,g\rangle,\langle b,e\rangle,\langle d,f\rangle,\langle d,g\rangle\}$

Serves
Med serves eclectic
Rio serves Mexican
Frasca serves Italian

$Serves = \{\langle e,j\rangle,\langle f,i\rangle,\langle e,h\rangle\}$

Matthew likes Med	TRUE
a *Likes* *f*	$\langle a,f\rangle$ *found in Likes*

Model-Theoretic Semantics

- Veryfing the truth value of simple expressions like "Matthew likes Med" is straight-forward
- However, expressions can be **more complex**
    1. *Katie likes the Rio and Matthew likes the Med*
    2. *Katie and Caroline like the same restaurants*
    3. *Not every body likes Frasca*
- Need method for determining truth of complex expressions from **meaning of parts** and **meaning of logical operators**, e.g. conjunction

First-Order Logic

Section 17.3 in Textbook

hannes@ru.is

T-725-MALV

First-Order Logic

- **FOL** is a **Meaning Representation Language** that provides sound computational basis for
  - verifiability
  - inference
  - expressiveness

  and a sound model-theoretic semantics
- The represented world consists of
  - objects, properties of objects and relations among objects

FOL Context-free Grammar

Formula → AtomicFormula

| Formula Connective Formula

| Quantifier Variable, ... Formula

| $\neg$ Formula

| (Formula)

AtomicFormula → Predicate(Term, ...)

Term → Function(Term, ...)

| Constant

| Variable

Connective → $\wedge$ | $\vee$ | $\Rightarrow$

Quantifier → $\forall$ | $\exists$

Constant → A | Vegetarian | Maharani ...

Variable → x | y | ...

Predicate → Serves | Near | ...

Function → LocationOf | CuisineOf | ...

Basic Elements: Objects

Formula → AtomicFormula
| Formula Connective Formula
| Quantifier Variable, ... Formula
| $\neg$ Formula
| (Formula)

AtomicFormula → Predicate(Term, ...)

Term → Function(Term, ...)
| Constant
| Variable

Connective → $\wedge$ | $\vee$ | $\Rightarrow$

Quantifier → $\forall$ | $\exists$

Constant → A | Vegetarian | Maharani ...

Variable → x | y | ...

Predicate → Serves | Near | ...

Function → LocationOf | CuisineOf | ...

Constants: Harry, Susan, Sun **Functions**: HeightOf, AgeOf
Variables: w, x

Basic Elements: Relations

Formula → AtomicFormula
| Formula Connective Formula
| Quantifier Variable, ... Formula
| ¬ Formula
| (Formula)

AtomicFormula → **Predicate**(Term, ...)

Term → Function(Term, ...)
| Constant
| Variable

Connective → ∧ | ∨ | ⇒

Quantifier → ∀ | ∃

Constant → A | Vegetarian | Maharani ...

Variable → x | y | ...

Predicate → Serves | Near | ...

Function → LocationOf | CuisineOf | ...

Predicates: Likes(Harry, Susan), Enjoys(Susan, Sun)

Basic Elements: Properties

Formula → AtomicFormula
| Formula Connective Formula
| Quantifier Variable, ... Formula
| ¬ Formula
| (Formula)

AtomicFormula → **Predicate**(Term, ...)

Term → Function(Term, ...)
| Constant
| Variable

Connective → ∧ | ∨ | ⇒

Quantifier → ∀ | ∃

Constant → A | Vegetarian | Maharani ...

Variable → x | y | ...

Predicate → Serves | Near | ...

Function → LocationOf | CuisineOf | ...

Predicates: Likes(Harry, Susan), Enjoys(Susan, Sun)
Person(Harry), Smiling(Susan), Shining(Sun)

Complex Representations

Formula → AtomicFormula
| Formula Connective Formula
| Quantifier Variable, ... Formula
| ¬ Formula
| (Formula)

AtomicFormula → Predicate(Term, ...)

Term → Function(Term, ...)
| Constant
| Variable

Connective → ∧ | ∨ | ⇒

Quantifier → ∀ | ∃

Constant → A | Vegetarian | Maharani ...

Variable → x | y | ...

Predicate → Serves | Near | ...

Function → LocationOf | CuisineOf | ...

More complex: Likes(Harry, Susan) ∧ Smiling(Susan) ∧ ¬ Shining(Sun)

Variables and Quantifiers

Formula → AtomicFormula
| Formula Connective Formula
| Quantifier Variable, ... Formula
| ¬ Formula
| (Formula)

AtomicFormula → Predicate(Term, ...)

Term → Function(Term, ...)
| Constant
| Variable

Connective → ∧ | ∨ | ⇒

Quantifier → ∀ | ∃

Constant → A | Vegetarian | Maharani ...

Variable → x | y | ...

Predicate → Serves | Near | ...

Function → LocationOf | CuisineOf | ...

There exists:	∃x Person(x) ∧ Smiling(x)	(use conjunction)
For all:	∀y Person(y) ⇒ Likes(y, Susan)	(use implication)

Semantics of FOL

- The **objects** (terms) denote elements in a model **domain**
- **Atomic formulas** are sets of domain elements (properties) or sets of tuples (relations)
- Verifying these as simple as checking **membership in sets**
- What about verifying more **complex formulas?**

Semantics of Connectives

- Formulas containing connectives are interpreted based on **meaning of components** combined with **meaning of connectives** themselves
  - Semantics of connectives given in **truth table**

P	**Q**	**$\neg$ P**	**P $\wedge$ Q**	**P $\vee$ Q**	**P $\Rightarrow$ Q**
False	False	True	False	False	True
False	True	True	False	True	True
True	False	False	False	True	False
True	True	False	True	True	True

Inference

- **Modus ponens** most widely implemented inference method provided by FOL

1 α

2 $\alpha \Rightarrow \beta$ (if left hand side is true...)

---

3 β (...we can infer the right hand side)