The Representation of Meaning

Chapter 17 J&M'09

Meaning

- **Linguistic Input**
  John called a friend from Australia.
- **Morphosyntactic analysis**
  S(NP(PN(John)),VP(TV(call),NP(DT(a),N(N(friend),PP(P(from),NP(PN(Australia)))))))
  S(NP(PN(John)),VP(VP(TV(call),NP(DT(a),N(friend))),PP(P(from),NP(PN(Australia)))))
- **Semantic analysis**
  `call(John,from(friend,australia))`
  `from(call(John,friend),australia))`

Meaning

Why (attempt to) represent meaning?

- To detect discourse-level dependencies:

  (1) a. The monkey ate the banana because it was hungry.
  b. The monkey ate the banana because it ripe.
  c. The monkey ate the banana because it was dinner time.

- To compute inferences:

  (2) Anyone who smokes snores. John smokes.
  $\Rightarrow$ John snores.

- To handle questions:

  (3) Q: Who did John call?
  A: A friend from Australia.

- To do consistency checking:

  (4) Mia is married. Mia is single. (inconsistent)

- Informativity checking

  (5) Mia is married. Mia is not single. (not new information)

Shallow vs. deep semantics

You get what you pay for.

- Cheap, fast, low-level techniques are appropriate in domains where speed and volume matter more than accuracy
- More computationally expensive, higher-level techniques are appropriate when higher-quality results are required.

Meaning

The language of First-order logic

- Non-logical symbols
  - Constants: *john*, *mary*, *kim*, ...
  - Predicates: *sing*, *cat*, *happy*, ...
  - Relations: *see*, *give*, ...
- Variables: $x_1, ..., x_n, y_1, ..., y_n, ...$
- Boolean connectives
  - Negation: $\neg$
  - Conjunction: $\wedge$
  - Disjunction: $\vee$
  - Implication: $\Rightarrow$
- Quantifiers: $\forall$ (universal), $\exists$ (existential)
- Punctuation marks:) (,

Everyone who smokes snores = *Quem fuma ressona* =
$\forall x((person(x) \wedge smoke(x)) \Rightarrow snore(x))$

The Syntax of First-order logic (defined via a CFG)

1. Formula $\rightarrow$ Predicate(α)
   (where α is a variable or a constant)
2. Formula $\rightarrow$ Relation($\alpha_1, ..., \alpha_n$)
3. Formula $\rightarrow$ Formula $\wedge$ Formula
4. Formula $\rightarrow$ Formula $\vee$ Formula
5. Formula $\rightarrow$ $\neg$ Formula
6. Formula $\rightarrow$ Formula $\Rightarrow$ Formula
7. Formula $\rightarrow$ $\exists x$(Formula)
8. Formula $\rightarrow$ $\forall x$(Formula)

Examples of formulas:

$\exists x(cat(x) \wedge \neg snore(x))$

$\forall x(cat(x) \Rightarrow \exists y(dog(y) \wedge likes(x, y)))$

Examples of non-formulas:

$\exists x(cat(x) \wedge \neg \vee snore(x))$

$\exists x(\Rightarrow cat(x))$

Translation examples (ignoring events and tense, for now)

(6) a. John smiled.
$smile(john)$

b. Fred likes Mia.
$like(fred, mia)$

c. Each cat yawned.
$\forall x(cat(x) \Rightarrow yawn(x))$

d. No cat yawned.
$\neg\exists x(cat(x) \land yawn(x))$

e. A cat didn't yawn.
$\exists x(cat(x) \land \neg yawn(x))$

f. Not every cat didn't yawn.
$\neg\forall x(cat(x) \Rightarrow \neg yawn(x))$

g. Either a dog yawned or a cat sneezed.
$(\exists x(dog(x) \land yawn(x))) \lor (\exists y(cat(y) \land sneeze(y)))$

Meaning

The interpretation of linguistic expressions is context-dependent.

(7) I saw a cat.

The correct interpretation of this sentence is contingent on:

- Who the speaker is.
- The utterance time and location.
- The speaker's knowledge of the world.

So, we need a way to represent **contexts** (situations).
In other words, a knowledge base.

Meaning

Model : a representation of a situation / context

$\mathcal{M} = \langle D, F \rangle$

- D is the domain (a non-empty set of entities)
  E.g., $D = \{a, b, c, d, e, f, g\}$
- F is an interpretation function (a set of ordered pairs)
  E.g., $F = \{\langle rat, \{a, b, c, d, e, f, g\}\rangle, \langle see, \{\langle a, b\rangle, \langle f, e\rangle\}\rangle\}$

Example 1:
$\mathcal{M} = \langle\{a, b, c, d\}, \{\langle cat, \{a, c, d\}\rangle, \langle dog, \{b\}\rangle, \langle sneeze, \{a, b, d\}\rangle\}\rangle$

Example 2:
$\mathcal{M} = \langle\{a, b, c, d\}, \{\langle cat, \{a, c, d\}\rangle, \langle dog, \{b\}\rangle, \langle chase.\{\langle b, c\rangle, \langle c, d\rangle\}\rangle\}\rangle$

Example 3:
$\mathcal{M} = \langle\{a, b\}, \{\langle tom, a\rangle, \langle mary, b\rangle, \langle love, \{\langle a, b\rangle, \langle b, a\rangle\}\rangle\}\rangle$

Meaning

Figure 17.2

Domain	$\mathscr{D} = \{a,b,c,d,e,f,g,h,i,j\}$
Matthew, Franco, Katie and Caroline	a,b,c,d
Frasca, Med, Rio	e,f,g
Italian, Mexican, Eclectic	h,i,j
Properties	
Noisy	$Noisy = \{e,f,g\}$
Frasca, Med, and Rio are noisy	
Relations	
Likes	$Likes = \{\langle a,f\rangle, \langle c,f\rangle, \langle c,g\rangle, \langle b,e\rangle, \langle d,f\rangle, \langle d,g\rangle\}$
Matthew likes the Med	
Katie likes the Med and Rio	
Franco likes Frasca	
Caroline likes the Med and Rio	
Serves	$Serves = \{\langle e,j\rangle, \langle f,i\rangle, \langle e,h\rangle\}$
Med serves eclectic	
Rio serves Mexican	
Frasca serves Italian	

PEARSON

Speech and Language Processing, Second Edition
Daniel Jurafsky and James H. Martin

Interpreting First-order logic formulas

A variable assignment g is a function from variables to individuals.

For example, the formula in (8a) is true with respect to $\mathcal{M}$ given the interpretation g.

(8) a. $\exists x(dog(x) \wedge chase(x, felix))$

b. $\mathcal{M} = \langle D, F\rangle$ where
$D = \{a, b, c, d\}$
$F = \{\langle cat, \{a, c\}\rangle, \langle dog, \{b\}\rangle, \langle felix, c\rangle, \langle chase, \{\langle b, c\rangle, \langle c, d\rangle\}\rangle\}$

c. $g = \{\langle x, b\rangle, \langle felix, c\rangle\}$

Very important: g and F are functions, so we can write
$g(x) = b$ and $F(cat) = \{a, c\}$

Meaning

Formulas, satisfaction, assignments, and models

A formula ϕ is satisfied with respect to g iff there is an extension of g that satisfies the conditions in ϕ relative to the the model $\mathcal{M}$.

What is an extension?
For example, take $g = \{\langle y, a \rangle\}$ and $g' = \{\langle y, a \rangle, \langle x, b \rangle\}$.

We say that g' extends g, relative to x.

This is written as ${}_g[x]_{g'}$

The satisfaction function ${}_g[\![\phi]\!]^{\mathcal{M}}_{g'}$ takes as argument an assignment g, a formula ϕ, and a model $\mathcal{M}$ and outputs an extension g' that satisfies ϕ, relative to $\mathcal{M}$.

Formulas, satisfaction, assignments, and models

A formula ϕ is satisfied with respect to g iff there is an extension of g that satisfies the conditions in ϕ relative to the the model $\mathcal{M}$.

What is an extension?
For example, take $g = \{\langle y, a\rangle\}$ and $g' = \{\langle y, a\rangle, \langle x, b\rangle\}$.

We say that g' extends g, relative to x.

This is written as ${}_g[x]_{g'}$

The satisfaction function ${}_g[\![\phi]\!]^{\mathcal{M}}_{g'}$ takes as argument an assignment g, a formula ϕ, and a model $\mathcal{M}$ and outputs an extension g' that satisfies ϕ, relative to $\mathcal{M}$.

One last piece of ancillary notation:
$I(x) = g(x)$ (if x is a variable)
$I(c) = F(c)$ (if c is a constant, e.g. $john$, $mary$, ...)

Satisfaction conditions for First-order logic formulas

i. ${}_g[\![P(\alpha)]\!]_g^{\mathcal{M}}$ *iff* $I(\alpha) \in F(P)$
   Example: ${}_g[\![sneeze(tom)]\!]_g^{\mathcal{M}}$ *iff* $I(tom) \in F(sneeze)$

ii. ${}_g[\![R(\alpha_1, ..., \alpha_n)]\!]_g^{\mathcal{M}}$ *iff* $\langle I(\alpha_1)\,, ...,\, I(\alpha_n)\rangle \in F(R)$
   Example: ${}_g[\![love(x_1, x_2)]\!]_g^{\mathcal{M}}$ *iff* $\langle g(x_1), g(x_2)\rangle \in F(love)$

iii. ${}_{g_1}[\![\exists x(\phi)]\!]_{g_3}^{\mathcal{M}}$ *iff* there is a g_2 such that ${}_{g_1}[x]_{g_2}$ and ${}_{g_2}[\![\phi]\!]_{g_3}^{\mathcal{M}}$

iv. ${}_g[\![\neg\phi]\!]_g^{\mathcal{M}}$ *iff* there is no g_2 such that ${}_g[\![\phi]\!]_{g_2}^{\mathcal{M}}$

v. ${}_g[\![\forall x(\phi)]\!]_g^{\mathcal{M}}$ *iff* ${}_g[\![\neg\exists x(\neg\phi)]\!]_g^{\mathcal{M}}$

vi. ${}_{g_1}[\![\phi_1 \wedge \phi_2]\!]_{g_3}^{\mathcal{M}}$ *iff* there is some g_2 such that ${}_{g_1}[\![\phi_1]\!]_{g_2}^{\mathcal{M}}$ and some g_3 such that ${}_{g_2}[\![\phi_2]\!]_{g_3}^{\mathcal{M}}$

vii. ${}_{g1}[\![\phi_1 \vee \phi_2]\!]_{g2}^{\mathcal{M}}$ *iff* ${}_{g1}[\![\phi_1]\!]_{g2}^{\mathcal{M}}$ or ${}_{g1}[\![\phi_2]\!]_{g2}^{\mathcal{M}}$

viii. ${}_{g1}[\![\phi_1 \Rightarrow \phi_2]\!]_{g2}^{\mathcal{M}}$ *iff* ${}_{g1}[\![\neg(\phi_1) \vee \phi_2]\!]_{g2}^{\mathcal{M}}$

Putting it all together

Given the model

(9) $\mathcal{M} = \langle D, F \rangle$ where
$D = \{a, b, c, d\}$
$F = \{\langle cat, \{a, c\}\rangle, \langle dog, \{b\}\rangle, \langle ball, \{d\}\rangle, \langle chase, \{\langle b, c\rangle, \langle c, d\rangle\}\rangle\}$

and the proposition

(10) $\exists x(dog(x) \wedge \exists y(cat(y) \wedge chase(x, y)))$

is it the case that (10) is true in $\mathcal{M}$?

I.e. is there a g such that:
${}_{\{\}}[\![\exists x(dog(x) \wedge \exists y(cat(y) \wedge chase(x, y)))]\!]_g^{\mathcal{M}}$?

Meaning

Applying the satisfaction conditions

- ${}_{\{\}}[\![\exists x(dog(x) \wedge \exists y(cat(y) \wedge chase(x,y)))]\!]^{\mathcal{M}}_{g_1}$
- ${}_{\{\langle x,b\rangle\}}[\![dog(x) \wedge \exists y(cat(y) \wedge chase(x,y))]\!]^{\mathcal{M}}_{g_2}$
- ${}_{\{\langle x,b\rangle\}}[\![dog(x)]\!]^{\mathcal{M}}_{g_3}$ and ${}_{g_3}[\![\exists y(cat(y) \wedge chase(x,y))]\!]^{\mathcal{M}}_{g_2}$
- ${}_{\{\langle x,b\rangle\}}[\![dog(x)]\!]^{\mathcal{M}}_{\{\langle x,b\rangle\}}$ *iff* $b \in F(dog)$ i.e. $b \in \{b\}$
- ${}_{\{\langle x,b\rangle\}}[\![\exists y(cat(y) \wedge chase(x,y))]\!]^{\mathcal{M}}_{g_2}$
- ${}_{\{\langle x,b\rangle,\langle y,c\rangle\}}[\![cat(y) \wedge chase(x,y)]\!]^{\mathcal{M}}_{g_2}$
- ${}_{\{\langle x,b\rangle,\langle y,c\rangle\}}[\![cat(y)]\!]^{\mathcal{M}}_{g_4}$ and ${}_{g_4}[\![chase(x,y)]\!]^{\mathcal{M}}_{g_2}$
- ${}_{\{\langle x,b\rangle,\langle y,c\rangle\}}[\![cat(y)]\!]^{\mathcal{M}}_{\{\langle x,b\rangle,\langle y,c\rangle\}}$ *iff* $c \in F(cat)$ i.e. $c \in \{a,c\}$
- ${}_{\{\langle x,b\rangle,\langle y,c\rangle\}}[\![chase(x,y)]\!]^{\mathcal{M}}_{\{\langle x,b\rangle,\langle y,c\rangle\}}$ *iff* $\langle b,c\rangle \in F(chase)$ i.e. $\langle b,c\rangle \in \{\langle b,c\rangle,\langle c,d\rangle\}$
- $g_1 = \{\langle x,b\rangle,\langle y,c\rangle\}$

Meaning

... next up: implementing this in Prolog.