Lecture 11: Knowledge Based Systems

Dragan Doder

Methods in AI Research

Before we start

Next four lectures: Reasoning

- Knowledge based reasoning
- Description logic
- Non-monotonic reasoning – *bonus lecture*
- Formal argumentation

Outline of this lecture

Knowledge representation and reasoning

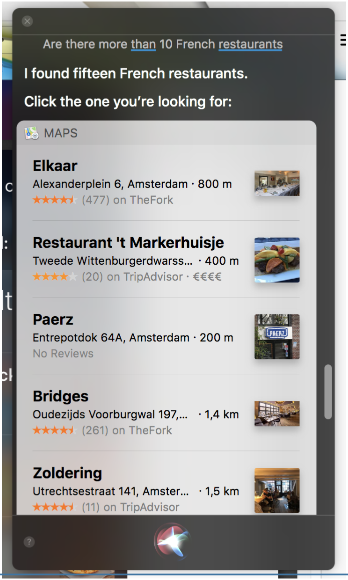
Recap – Propositional logic

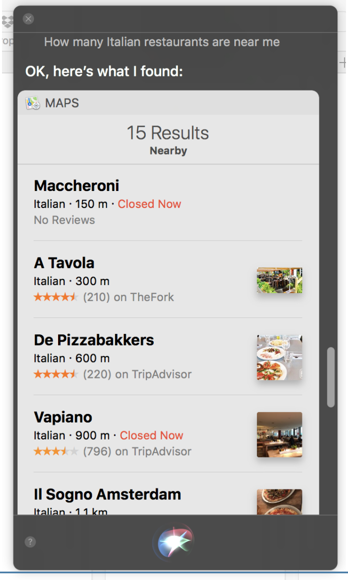
Some fundamental logical notions

First-Order logic

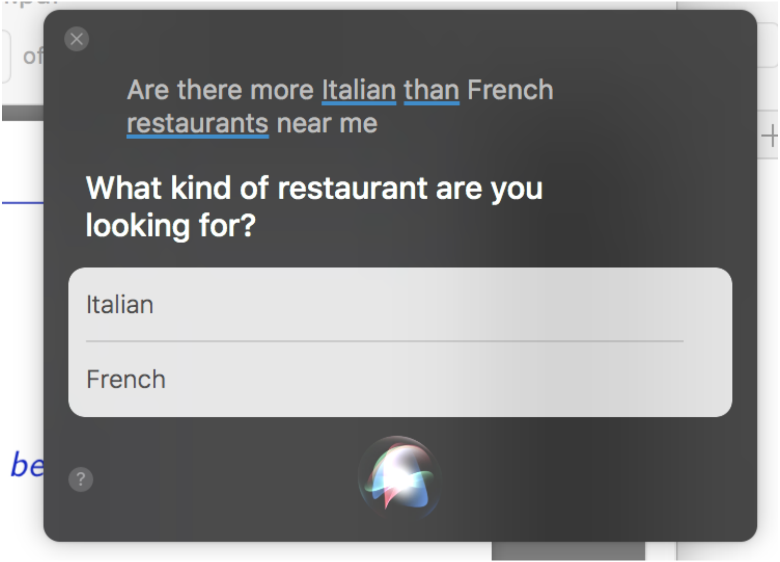
Knowledge representation and reasoning

Conversational agents - where are we now?





Conversational agents - where are we now?



Conversational agents - where would we like to go?

Conversational agents - where would we like to go?

User: *My father is called Marc.*

User: *My father likes Football.*

Conversational agents - where would we like to go?

User: *My father is called Marc.*

User: *My father likes Football.*

(A few months later)

User: *Marc was born on 22 June 1965.*

User: *Marc's favorite football team is Ajax.*

Conversational agents - where would we like to go?

User: *My father is called Marc.*

User: *My father likes Football.*

(A few months later)

User: *Marc was born on 22 June 1965.*

User: *Marc's favorite football team is Ajax.*

(Three days later)

User: *Remind me to buy football tickets for my father a month before his birthday.*

Conversational agents - where would we like to go?

User: *My father is called Marc.*

User: *My father likes Football.*

(A few months later)

User: *Marc was born on 22 June 1965.*

User: *Marc's favorite football team is Ajax.*

(Three days later)

User: *Remind me to buy football tickets for my father a month before his birthday.*

(On May 22nd)

CA: Hey, Marc's birthday is in a month and you wanted to buy him football tickets.

Conversational agents - where would we like to go?

User: *My father is called Marc.*

User: *My father likes Football.*

(A few months later)

User: *Marc was born on 22 June 1965.*

User: *Marc's favorite football team is Ajax.*

(Three days later)

User: *Remind me to buy football tickets for my father a month before his birthday.*

(On May 22nd)

CA: Hey, Marc's birthday is in a month and you wanted to buy him football tickets.

CA: How about I order today the tickets for Ajax–Utrecht happening on July 3rd?

Conversational agents - What do we need for that?

Various types of logical and commonsense reasoning:

- Coreference resolution
- Reasoning about superlatives
- Reasoning about family relations
- Time reasoning

⋮

Conversational agents - What do we need for that?

Various types of logical and commonsense reasoning:

- Coreference resolution
- Reasoning about superlatives
- Reasoning about family relations
- Time reasoning

⋮

Major components of AI:

knowledge, reasoning, language understanding, learning.

Conversational agents - What do we need for that?

Various types of logical and commonsense reasoning:

- Coreference resolution
- Reasoning about superlatives
- Reasoning about family relations
- Time reasoning

⋮

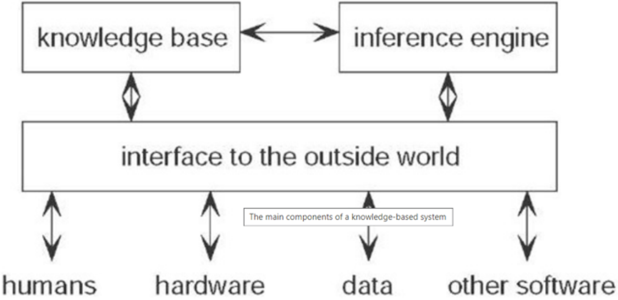
Major components of AI:

knowledge, reasoning, language understanding, learning.

$\Uparrow$ $\Uparrow$

Lectures 11–14

Knowledge based systems



Knowledge base: represents facts about the world

Inference engine: applies logical rules to KB to deduce new facts

Knowledge Representation and Reasoning

Knowledge representation

- use formal symbols to represent knowledge of an agent
- not all the knowledge is explicitly represented

Reasoning

- bridges the gap between what is represented and what is known
- formal manipulation of the symbols

The role of logic

- Logic is study of entailment relations
  - languages, truth conditions, inference rules

MYCIN - sample rule

MYCIN ('70s, Stanford U.): Reasoning about bacterial infections

IF the stain of the organism is gram negative
AND the morphology of the organism is rod
AND the aerobiocity of the organism is gram anaerobic
THEN the there is strongly suggestive evidence (0.8) that the class of the organism is enterobacteriaceae

Propositional logic

Propositions

Which sentences are propositions?

Any statement which can be evaluated as *true* or *false*.

Examples of propositions:

- Chris is a lecturer of MAIR.
- The Earth is flat.
- $15 + 3 \leq 20$
- "P=NP"

Propositions

Which sentences are propositions?

Any statement which can be evaluated as *true* or *false*.

Examples of propositions:

- Chris is a lecturer of MAIR.
- The Earth is flat.
- $15 + 3 \leq 20$
- "P=NP"

The following sentences are *not* propositions:

- Are you a lecturer of MAIR?
- Don't cheat at the exam!
- $2x + 3 = 10$

Composite propositions

Examples:

- It is not the case that the Earth is flat.
- Chris is a lecturer of MAIR or the Earth is flat.
- Chris is a lecturer of MAIR and the Earth is flat.
- If the Earth is flat, then "P=NP"
- $15 + 3 \leq 20$ if and only if $2(15 + 3) \leq 2 \cdot 20$

Composite propositions

Examples:

- It is not the case that the Earth is flat.
- Chris is a lecturer of MAIR or the Earth is flat.
- Chris is a lecturer of MAIR and the Earth is flat.
- If the Earth is flat, then "P=NP"
- $15 + 3 \leq 20$ if and only if $2(15 + 3) \leq 2 \cdot 20$

Course manual:

If all three individual grades are at least 5.0 and the weighted final grade is at least 5.5, then you passed the course.

Propositional logic – Syntax

Alphabet: the symbols of propositional logic are:

- propositional letters: p, q, r etc.
- (Boolean) connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$
- brackets "(" and ")"

Propositional logic – Syntax

Alphabet: the symbols of propositional logic are:

- propositional letters: p, q, r etc.
- (Boolean) connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$
- brackets "(" and ")"

Formulas: the well defined propositional formulas are:

- propositional letters: $p, q, r \ldots$
- If α and β are formulas, then the following are also formulas.

$\neg\alpha$	negation	"not ..."
$(\alpha \wedge \beta)$	conjunction	"... and ..."
$(\alpha \vee \beta)$	disjunction	"... or ..."
$(\alpha \rightarrow \beta)$	implication	"if ..., then ..."
$(\alpha \leftrightarrow \beta)$	equivalence	"... iff ..."

Propositional logic – Semantics

A *model* (of a logic) is a precisely described situation in which the truth of each formula is determined.
A model m is a *model of a formula* α, if α is true in m.

$$m \models \alpha$$

- *Models* of propositional logic – *valuations*.
- *A valuation* assigns a truth value to every propositional variable

Example:

	p	q	r
model m:	*true*	*true*	*false*

Given m, for every formula we can determine its truth value:

- Propositional variables have their values directly assigned by m
- *[Truth functionality:]* The truth value of a compound formula is a function of the truth values of its subformulas.

Truth table

The following *truth table* characterizes the Boolean connectives.

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

Truth table

The following *truth table* characterizes the Boolean connectives.

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

	p	q	r
model m:	*true*	*true*	*false*

$\neg p \wedge (q \vee r)$

Truth table

The following *truth table* characterizes the Boolean connectives.

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

	p	q	r
model m:	*true*	*true*	*false*

$\neg p \wedge (q \vee r) = \neg true \wedge (true \vee false)$

Truth table

The following *truth table* characterizes the Boolean connectives.

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

	p	q	r
model m:	*true*	*true*	*false*

$\neg p \wedge (q \vee r) = \neg \mathit{true} \wedge (\mathit{true} \vee \mathit{false}) = \mathit{false} \wedge \mathit{true}$

Truth table

The following *truth table* characterizes the Boolean connectives.

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

	p	q	r
model m:	*true*	*true*	*false*

$\neg p \wedge (q \vee r) = \neg true \wedge (true \vee false) = false \wedge true = false$

Some fundamental logical notions

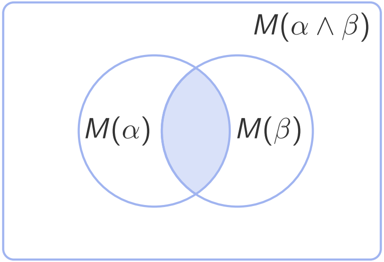
Model of a KB

$M(\alpha)$ – the set of all models of α $\quad (M(\alpha) = \{m \mid m \models \alpha\})$

Model of a KB

$M(\alpha)$ – the set of all models of α $\quad (M(\alpha) = \{m \mid m \models \alpha\})$

$m \models \alpha \wedge \beta$ iff $m \models \alpha$ and
$m \models \beta$

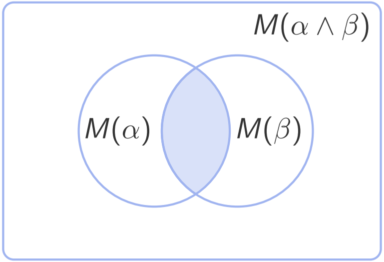


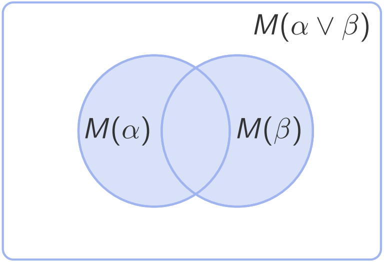
Model of a KB

$M(\alpha)$ – the set of all models of α $\quad (M(\alpha) = \{m \mid m \models \alpha\})$

$m \models \alpha \wedge \beta$ iff $m \models \alpha$ and $m \models \beta$

$m \models \alpha \vee \beta$ iff $m \models \alpha$ or $m \models \beta$



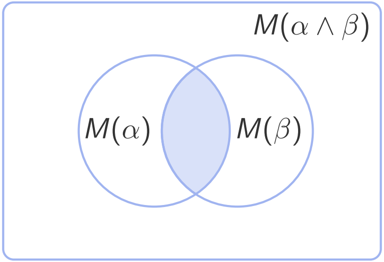


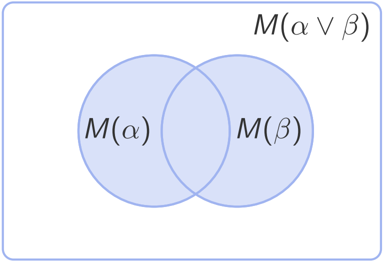
Model of a KB

$M(\alpha)$ – the set of all models of α $\quad (M(\alpha) = \{m \mid m \models \alpha\})$

$m \models \alpha \wedge \beta$ iff $m \models \alpha$ and $m \models \beta$

$m \models \alpha \vee \beta$ iff $m \models \alpha$ or $m \models \beta$





$M(KB)$ – the set of all models of *every* formula from KB

If $KB = \{\alpha, \beta, \gamma\}$, then $M(KB) = M(\alpha) \cap M(\beta) \cap M(\gamma)$

Some semantic concepts: satisfiability and validity

Satisfiability

A formula α is satisfiable iff there is a model m such that $m \models \alpha$ holds (e.g. $p \wedge q$)

α is satisfiable iff $Mod(\alpha) \neq \emptyset$

Validity

A formula α is valid (notation $\models \alpha$) iff $m \models \alpha$ holds for every model m (e.g. $p \vee \neg p$)

Some semantic concepts: Entailment

Entailment

A formula α is a logical consequence of a set of formulas KB ($KB \models \alpha$) iff α is true in every model in which KB is true. (e.g. $\{p, p \rightarrow q\} \models q$)

Entailment is a relationship between sentences (i.e., *syntax*) that is based on *semantics*.

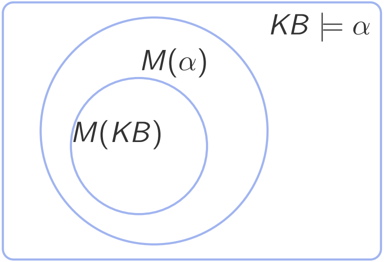
Some semantic concepts: Entailment

Entailment

A formula α is a logical consequence of a set of formulas KB ($KB \models \alpha$) iff α is true in every model in which KB is true. (e.g. $\{p, p \rightarrow q\} \models q$)

Entailment is a relationship between sentences (i.e., *syntax*) that is based on *semantics*.

- $KB \models \alpha$ iff $M(KB) \subseteq M(\alpha)$
- $\beta \models \alpha$ iff $\models \beta \rightarrow \alpha$
- An unsatisfiable formula entails every formula



Inference

$KB \vdash \alpha$ – "α is *inferred* from KB" (by...)

Two main groups of inference algorithms:

- Model-based: (e.g., truth table enumeration)
- Proof-based: apply inference rules to infer new knowledge (e.g., modus ponens:

$$\frac{\alpha \quad \alpha \rightarrow \beta}{\beta})$$

Inference in logic

Logical language, formulas

- inference system
  (inference rules)
- derivations ($\vdash$)

- semantics
  (models, satisfiability)
- entailment ($\models$)

Soundness: $KB \vdash \alpha \Longrightarrow KB \models \alpha$

Completeness: $KB \models \alpha \Longrightarrow KB \vdash \alpha$

First-Order logic

On expressiveness of Propositional logic

Question: Can we formalize this kind of reasoning in propositional logic?

- All humans are mortal
- Socrates is a human
- Therefore: Socrates is mortal

On expressiveness of Propositional logic

Question: Can we formalize this kind of reasoning in propositional logic?

- All humans are mortal
- Socrates is a human
- Therefore: Socrates is mortal

An attempt:

p: All humans are mortal
q: Socrates is a human
r: Socrates is mortal

$$\{p, q\} \not\models r$$

First-Order Logic

Whereas propositional logic assumes world contains *facts*,
first-order logic (like natural language) assumes the world contains:

- *Objects*: people, houses, numbers, theories, MAIR lecturers, colors, centuries ...
- *Relations*: red, round, prime, boring..., sister, bigger than, inside, part of, has color, occurred after, comes between ...
- *Functions*: father of, one more than, plus ...

Syntax of FOL: Basic elements

Constant symbols	$George, \; 2, \; UU, \ldots$
Predicate symbols (relation symbols)	$Yellow, Sister, \; \geq, \ldots$
Function symbols	$Sqrt, \; FatherOf, \ldots$
Variables	$x, \; y, \; a, \; b, \ldots$
Connectives	$\wedge \;\; \vee \;\; \neg \;\; \rightarrow \leftrightarrow$
Equality	$=$
Quantifiers	$\forall \; \exists$

Atomic formulas

$$\text{Atomic formula} \quad = \quad predicate\_symbol(term_1, \ldots, term_n)$$
$$\text{or } term_1 = term_2$$

$$\text{Term} \quad = \quad function\_symbol(term_1, \ldots, term_n)$$
$$\text{or } constant\_symbol \text{ or } variable$$

E.g.

- $Spouses(FatherOf(George), MotherOf(George))$
- $\geq (Height(George), Height(FatherOf(George)))$

Complex formulas

Complex formulas are made from atomic formulas using connectives and quantifiers

$$\neg\alpha, \quad \alpha \wedge \beta, \quad \alpha \vee \beta, \quad \alpha \rightarrow \beta, \quad \alpha \leftrightarrow \beta \quad \exists x\ \alpha, \quad \forall x\ \alpha$$

E.g.:

- $Spouses(FatherOf(George), MotherOf(George)) \rightarrow Spouses(MotherOf(George), FatherOf(George))$
- $\geq(1,2) \vee \neg(\geq(1,2))$
- $\forall x\ At(x, UU) \rightarrow Smart(x)$
  "Everyone at UU is smart"
- $\exists x\ At(x, UU) \wedge Tall(x)$
  "Someone at UU is Tall"

Complex formulas

Complex formulas are made from atomic formulas using connectives and quantifiers

$$\neg\alpha, \quad \alpha \wedge \beta, \quad \alpha \vee \beta, \quad \alpha \rightarrow \beta, \quad \alpha \leftrightarrow \beta \quad \exists x\ \alpha, \quad \forall x\ \alpha$$

E.g.:

- $Spouses(FatherOf(George), MotherOf(George)) \rightarrow Spouses(MotherOf(George), FatherOf(George))$
- $\geq(1,2) \vee \neg(\geq(1,2))$
- $\forall x\ At(x, UU) \rightarrow Smart(x)$
  "Everyone at UU is smart"
- $\exists x\ At(x, UU) \wedge Tall(x)$
  "Someone at UU is Tall"
- $\exists x \exists y\ Lec(x, MAIR) \wedge Lec(y, MAIR) \wedge \neg(x = y)$
  "There are at least two lecturers og MAIR"

Truth in first-order logic

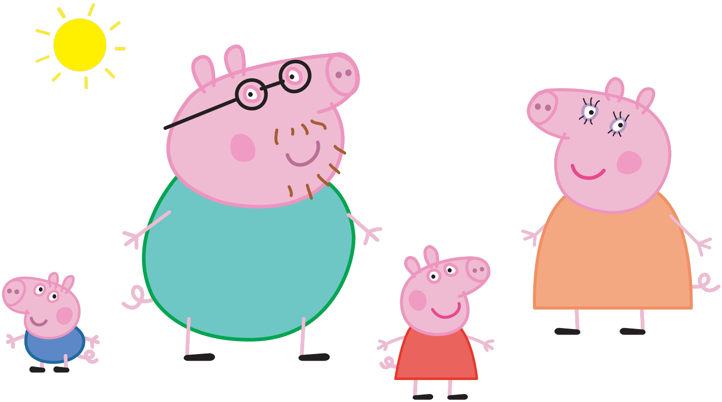
Formulas are true with respect to a *model/interpretation/structure*, which consists of

- A *non empty domain D* of objects
- An *interpretation function I* linking the syntax with the domain:
  - constant symbols $\rightarrow$ *objects*
  - predicate symbols $\rightarrow$ *relations*
  - function symbols $\rightarrow$ *functions*

An atomic formula $predicate\_symbol(term_1, \ldots, term_n)$ is true iff the *objects* referred to by $term_1, \ldots, term_n$ are in the *relation* referred to by *predicate_symbol*

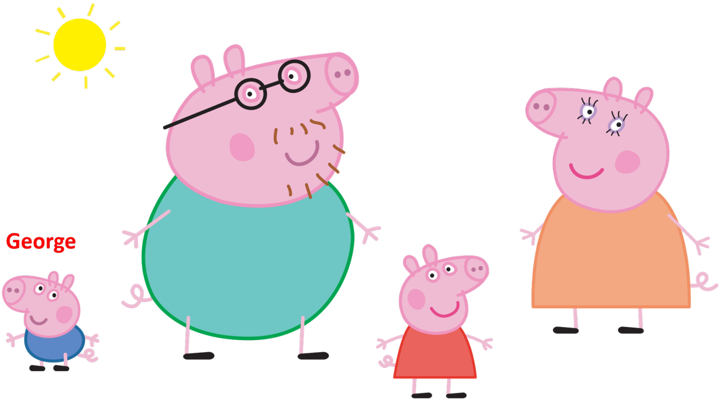
$\forall x\ \alpha$ is true in a model if α is true with x being each possible *object* in the model.

Models for FOL: Example



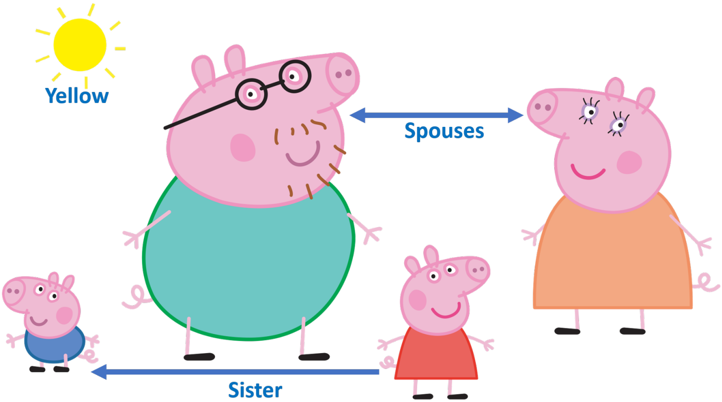
Objects: Peppa Pig, George Pig, Mammy Pig, Daddy Pig, Sun

Models for FOL: Example



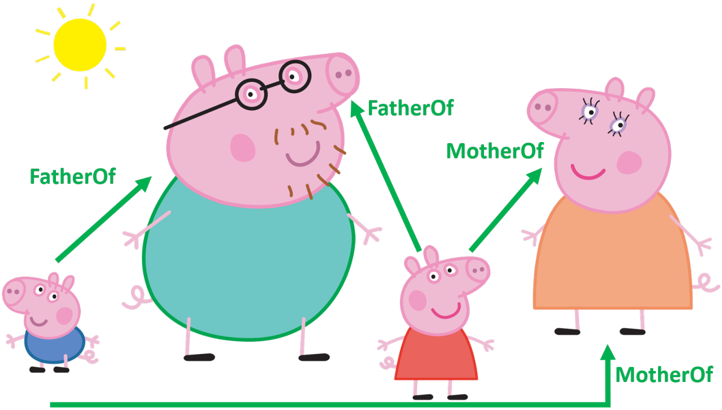
Constants: George

Models for FOL: Example



Relations: Yellow, Sister, Spouses

Models for FOL: Example



Functions: MotherOf, FatherOf

Semantics, more formally

An interpretation function I maps elements of the syntax to the domain as follows (superscripts – arity):

- For each constant symbol c, $I(c) = o \in D$
- For each n–ary function symbol f^n, $I(f^n) = \phi^n : D^n \to D$
- For each n–ary predicate symbol P^n, $I(P^n) = R^n \subseteq D^n$

So, the interpretation function maps all elements of the syntax to the domain except for variables.

Semantics, more formally

An interpretation function I maps elements of the syntax to the domain as follows (superscripts – arity):

- For each constant symbol c, $I(c) = o \in D$
- For each n–ary function symbol f^n, $I(f^n) = \phi^n : D^n \rightarrow D$
- For each n–ary predicate symbol P^n, $I(P^n) = R^n \subseteq D^n$

So, the interpretation function maps all elements of the syntax to the domain except for variables.

Interpreting terms:

- $George \longrightarrow I(\text{George}) \longrightarrow George\ Pig$
- $FatherOf(George) \longrightarrow I(FatherOf)(George\ Pig)$

Semantics, more formally

An interpretation function I maps elements of the syntax to the domain as follows (superscripts – arity):

- For each constant symbol c, $I(c) = o \in D$
- For each n–ary function symbol f^n, $I(f^n) = \phi^n \colon D^n \to D$
- For each n–ary predicate symbol P^n, $I(P^n) = R^n \subseteq D^n$

So, the interpretation function maps all elements of the syntax to the domain except for variables.

Interpreting terms:

- $George \longrightarrow I(\text{George}) \longrightarrow George\ Pig$
- $FatherOf(George) \longrightarrow I(FatherOf)(George\ Pig) \to Daddy\ Pig$

Semantics, more formally

An interpretation function I maps elements of the syntax to the domain as follows (superscripts – arity):

- For each constant symbol c, $I(c){=}o \in D$
- For each $n-$ary function symbol f^n, $I(f^n) = \phi^n\colon D^n \rightarrow D$
- For each $n-$ary predicate symbol P^n, $I(P^n){=}\ R^n \subseteq D^n$

So, the interpretation function maps all elements of the syntax to the domain except for variables.

Interpreting terms:

- $George \longrightarrow I(\text{George}) \longrightarrow George\ Pig$
- $FatherOf(George) \longrightarrow I(FatherOf)(George\ Pig) {\rightarrow} Daddy\ Pig$
- $FatherOf(x)$?

Denotation – Interpreting terms

Let $\mathcal{M} = (D, I)$ be a model.

A *variable assignment* μ maps every variable to an object in D:

$$\mu : \mathit{Variables} \to D$$

Denotation – Interpreting terms

Let $\mathcal{M} = (D, I)$ be a model.

A *variable assignment* μ maps every variable to an object in D:

$$\mu : \mathit{Variables} \rightarrow D$$

Denotation maps terms to objects of D (given $\mathcal{M}$ and μ):

- $\|x\|_{\mathcal{M},\mu} = \mu(x)$
- $\|c\|_{\mathcal{M},\mu} = I(c)$
- $\|f(t_1, \ldots, t_n)\|_{\mathcal{M},\mu} = I(f)(\|t_1\|_{\mathcal{M},\mu}, \ldots, \|t_n\|_{\mathcal{M},\mu})$

Satisfiability relation $\models$

Let $\mathcal{M} = (D, I)$ be a model, and let μ be a variable assignment.

$$
\begin{array}{lcl}
\mathcal{M}, \mu \models P_i^n(t_1,\ldots,t_n) & \Leftrightarrow & I(P_i^n)(\|t_1\|_{\mathcal{M},\mu}, \ldots, \|t_n\|_{\mathcal{M},\mu}) \\
\mathcal{M}, \mu \models \neg\alpha & \Leftrightarrow & \text{not } \mathcal{M}, \mu \models \alpha \\
\mathcal{M}, \mu \models \alpha \wedge \beta & \Leftrightarrow & \mathcal{M}, \mu \models \alpha \text{ and } \mathcal{M}, \mu \models \beta \\
\vdots & & \vdots \\
\mathcal{M}, \mu \models \forall x\ \alpha & \Leftrightarrow & \text{for all } \mu' =_x \mu\text{: } \mathcal{M}, \mu' \models \alpha \\
\mathcal{M}, \mu \models \exists x\ \alpha & \Leftrightarrow & \text{for some } \mu' =_x \mu\text{: } \mathcal{M}, \mu' \models \alpha
\end{array}
$$

μ is an *x-variant* of μ' (written as $\mu =_x \mu'$) if they map all variables to the same object in D except for x.

Satisfiability relation $\models$

Let $\mathcal{M} = (D, I)$ be a model, and let μ be a variable assignment.

$$
\begin{array}{lcl}
\mathcal{M}, \mu \models P_i^n(t_1,\ldots,t_n) & \Leftrightarrow & I(P_i^n)(\|t_1\|_{\mathcal{M},\mu}, \ldots, \|t_n\|_{\mathcal{M},\mu}) \\
\mathcal{M}, \mu \models \neg\alpha & \Leftrightarrow & \text{not } \mathcal{M}, \mu \models \alpha \\
\mathcal{M}, \mu \models \alpha \land \beta & \Leftrightarrow & \mathcal{M}, \mu \models \alpha \text{ and } \mathcal{M}, \mu \models \beta \\
\vdots & & \vdots \\
\mathcal{M}, \mu \models \forall x\ \alpha & \Leftrightarrow & \text{for all } \mu' =_x \mu\text{: } \mathcal{M}, \mu' \models \alpha \\
\mathcal{M}, \mu \models \exists x\ \alpha & \Leftrightarrow & \text{for some } \mu' =_x \mu\text{: } \mathcal{M}, \mu' \models \alpha
\end{array}
$$

μ is an *x-variant* of μ' (written as $\mu =_x \mu'$) if they map all variables to the same object in D except for x.

α is *true* in $\mathcal{M}$ (notation $\mathcal{M} \models \alpha$) iff $\mathcal{M}, \mu \models \alpha$ holds for all μ

Sentence: a formula in which all variables are bounded by a quantifier

Satisfiability relation $\models$

Let $\mathcal{M} = (D, I)$ be a model, and let μ be a variable assignment.

$$
\begin{array}{lcl}
\mathcal{M}, \mu \models P_i^n(t_1,\ldots,t_n) & \Leftrightarrow & I(P_i^n)(\|t_1\|_{\mathcal{M},\mu}, \ldots, \|t_n\|_{\mathcal{M},\mu}) \\
\mathcal{M}, \mu \models \neg\alpha & \Leftrightarrow & \text{not } \mathcal{M}, \mu \models \alpha \\
\mathcal{M}, \mu \models \alpha \wedge \beta & \Leftrightarrow & \mathcal{M}, \mu \models \alpha \text{ and } \mathcal{M}, \mu \models \beta \\
\vdots & & \vdots \\
\mathcal{M}, \mu \models \forall x\ \alpha & \Leftrightarrow & \text{for all } \mu' =_x \mu\text{: } \mathcal{M}, \mu' \models \alpha \\
\mathcal{M}, \mu \models \exists x\ \alpha & \Leftrightarrow & \text{for some } \mu' =_x \mu\text{: } \mathcal{M}, \mu' \models \alpha
\end{array}
$$

μ is an *x-variant* of μ' (written as $\mu =_x \mu'$) if they map all variables to the same object in D except for x.

α is *true* in $\mathcal{M}$ (notation $\mathcal{M} \models \alpha$) iff $\mathcal{M}, \mu \models \alpha$ holds for all μ

Sentence: a formula in which all variables are bounded by a quantifier (evaluated in $\mathcal{M}$, assignments irrelevant)

Beyond FOL

Jon Barwise (1942–2000) an American logician

To claim that logic is first-order logic is like to claim that astronomy is the study of the telescope.

Reasoning about:

knowledge	epistemic logic
time	temporal logic
norms	deontic logic
inconsistent knowledge	paraconsistent logic
defeasible inferences	non-monotonic logic
uncertainty	probabilistic logic, ...
vague statements	fuzzy logic
⋮	⋮

Summary

This lecture:

- Knowledge-based systems apply logical inference to a knowledge base to derive new information.
- Logics:
  - Propositional logic
  - First order logic

Next lecture:

- Description logics