Semantic models: ontologies (II)

MSc 2008/2009

Lecture 11/12

Summary of the previous lecture

- Ontologies.

- Introduction to the Semantic Web, RDF, RDFS, OWL.

Today‘s lecture

- More or Web ontology languages RDFS and OWL.

Logics

STI · INNSBRUCK

- Logics are formal languages for representing information such that conclusions can be drawn.
- Syntax defines the sentences in the language.
- Semantics defines the "meaning" of sentences.
  - more precisely, defines the truth of each sentence w.r.t. each possible world (model)

- E.g., the language of arithmetic
  - $x+2 \geq y$ is a sentence; $x2+y > \{\}$ is not a sentence
  -
  - $x+2 \geq y$ is true iff the number $x+2$ is no less than the number y
  -
  - Sentence $x+2 \geq y$ is true in a world where $x = 7$, $y = 1$
  - Sentence $x+2 \geq y$ is false in a world where $x = 0$, $y = 6$
  -

Entailment

- Entailment means that one thing follows from another

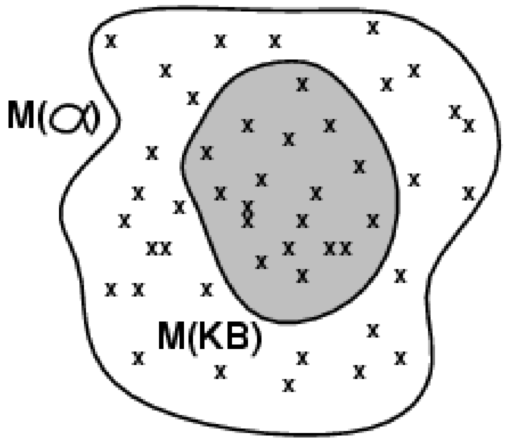
$$KB \models \alpha$$

- Knowledge base KB entails sentence α if and only if α is true in all worlds where KB is true
  - E.g., the KB containing “Innsbruck is in Tirol” and “Tirol is in Austria” entails “Innsbruck is in Austria”
  - E.g., the KB containing x+y = 4 entails 4 = x+y
- Entailment is a relationship between sentences (i.e., syntax) that is based on semantics

Models

STI · INNSBRUCK

- Possible worlds – environments in which the agent might or might not be in
- Models – mathematical abstractions, formally structured worlds with respect to which truth of sentences can be evaluated
- m is a model of a sentence α if α is true in m
- M(α) is the set of all models of α
- KB ⊨ α iff $M(KB) \subseteq M(\alpha)$
- Knowledge base KB entails sentence α if and only if all models of the KB are models of α



Inference

- *KB* $\vdash_i \alpha$: sentence α can be derived from KB by procedure i, i proves α

- Soundness: i is sound if whenever KB $\vdash_i \alpha$, it is also true that KB $\models \alpha$
- Completeness: i is complete if whenever KB $\models \alpha$, it is also true that KB $\vdash_i \alpha$

- A sound and complete procedure answers any question whose answer follows from what is known by the *KB* correctly.

Propositional logic: Syntax

STI · INNSBRUCK

- Propositional logic is the simplest logic and illustrates basic ideas of logic

- The proposition symbols P_1, P_2,…are sentences (formulae)

  - If S is a sentence, $\neg S$ is a sentence (negation)
  - If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (conjunction)
  - If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (disjunction)
  - If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication)
  - If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Propositional logic: Semantics

STI · INNSBRUCK

- Each model specifies true/false values for each proposition symbol (8 possible models)
  - E.g.

	$P_{1,2}$	$P_{2,2}$	$P_{3,1}$
	false	true	false

- Rules for evaluating truth with respect to an interpretation *m*:

$\neg S$	is true	iff	S is false
$S_1 \wedge S_2$	is true	iff	S_1 is true and S_2 is true
$S_1 \vee S_2$	is true	iff	S_1 is true or S_2 is true
$S_1 \Rightarrow S_2$	is true	iff	S_1 is false or S_2 is true
$S_1 \Rightarrow S_2$	is false	iff	S_1 is true and S_2 is false
$S_1 \Leftrightarrow S_2$	is true	iff	$S_1 \Rightarrow S_2$ is true and $S_2 \Rightarrow S_1$ is true

- Simple recursive process evaluates an arbitrary sentence w.r.t. an interpretation, e.g.,

$$\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1}) : \ \text{true} \wedge (\text{true} \vee \text{false}) = \text{true} \wedge \text{true} = \text{true}$$

Truth tables for connectives

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

Validity and satisfiability

STI · INNSBRUCK

A sentence is valid if it is true in all models,

e.g., *True*, $A \lor \neg A$, $A \Rightarrow A$, $(A \land (A \Rightarrow B)) \Rightarrow B$

Validity is connected to inference via the Deduction Theorem:

$KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

A sentence is satisfiable if it is true in some model

e.g., $A \lor B$, C

A sentence is unsatisfiable if it is true in no models

e.g., $A \land \neg A$

Satisfibility is connected to inference via the following

$KB \models \alpha$ if and only if $(KB \land \neg \alpha)$ is unsatisfiable

Logical equivalence

- Two sentences are logically equivalent iff they are true in same models: α ≡ ß iff α ╞ β and β ╞ α

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) && \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) && \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) && \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) && \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) && \text{de Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) && \text{de Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) && \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) && \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

Expressiveness limitation of propositional logic

- KB contains "physics" sentences for every single square
- For every time *t* and every location [*x*,*y*],

$$L_{x,y}^{t} \wedge FacingRight^{t} \wedge Forward^{t} \Rightarrow L_{x+1,y}^{t+1}$$

- Rapid proliferation of clauses

Pros and cons of propositional logic

- ☺ Propositional logic is declarative
- ☺ Propositional logic allows partial/disjunctive/negated information
  - (unlike most data structures and databases)
- ☺ Propositional logic is compositional:
  - meaning of $B_{1,1} \wedge P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$
- ☺ Meaning in propositional logic is context-independent
  - (unlike natural language, where meaning depends on context)
- ☹ Propositional logic has very limited expressive power
  - (unlike natural language)
  - E.g., cannot say "pits cause breezes in adjacent squares“
    - except by writing one sentence for each square

First-order logic (FOL)

- FOL (like propositional logics) is declarative, compositional, and the meaning of its sentences is context-independent
- Propositional logic assumes the world contains facts, FOL (like natural language) assumes the world contains (ontological commitment)
  - Objects: people, houses, numbers, colors, baseball games, wars, …
  - Relations: red, round, prime, brother of, bigger than, part of, comes between, …
  - Functions: father of, best friend, one more than, plus, …
- FOL can express facts about some or all the objects in the universe
- Every sentence in FOL is either true, false or unknown to an agent (epistemological commitment)

Syntax of FOL: Basic elements

• Constants	KingJohn, 2, NUS,...
• Predicate symbols	Brother, >,...
• Function symbols	Sqrt, LeftLegOf,...
• Variables	x, y, a, b,...
• Connectives	$\neg$, $\Rightarrow$, $\wedge$, $\vee$, $\Leftrightarrow$
• Quantifiers	$\forall$, $\exists$

FOL sentences

Atomic sentence = *predicate* (*term*$_1$,...,*term*$_n$)
or *term*$_1$ = *term*$_2$

Term = *function* (*term*$_1$,...,*term*$_n$)
or *constant* or *variable*

Complex sentences are made from atomic sentences using connectives

$\neg S$, $S_1 \wedge S_2$, $S_1 \vee S_2$, $S_1 \Rightarrow S_2$, $S_1 \Leftrightarrow S_2$,

Truth in FOL

STI · INNSBRUCK

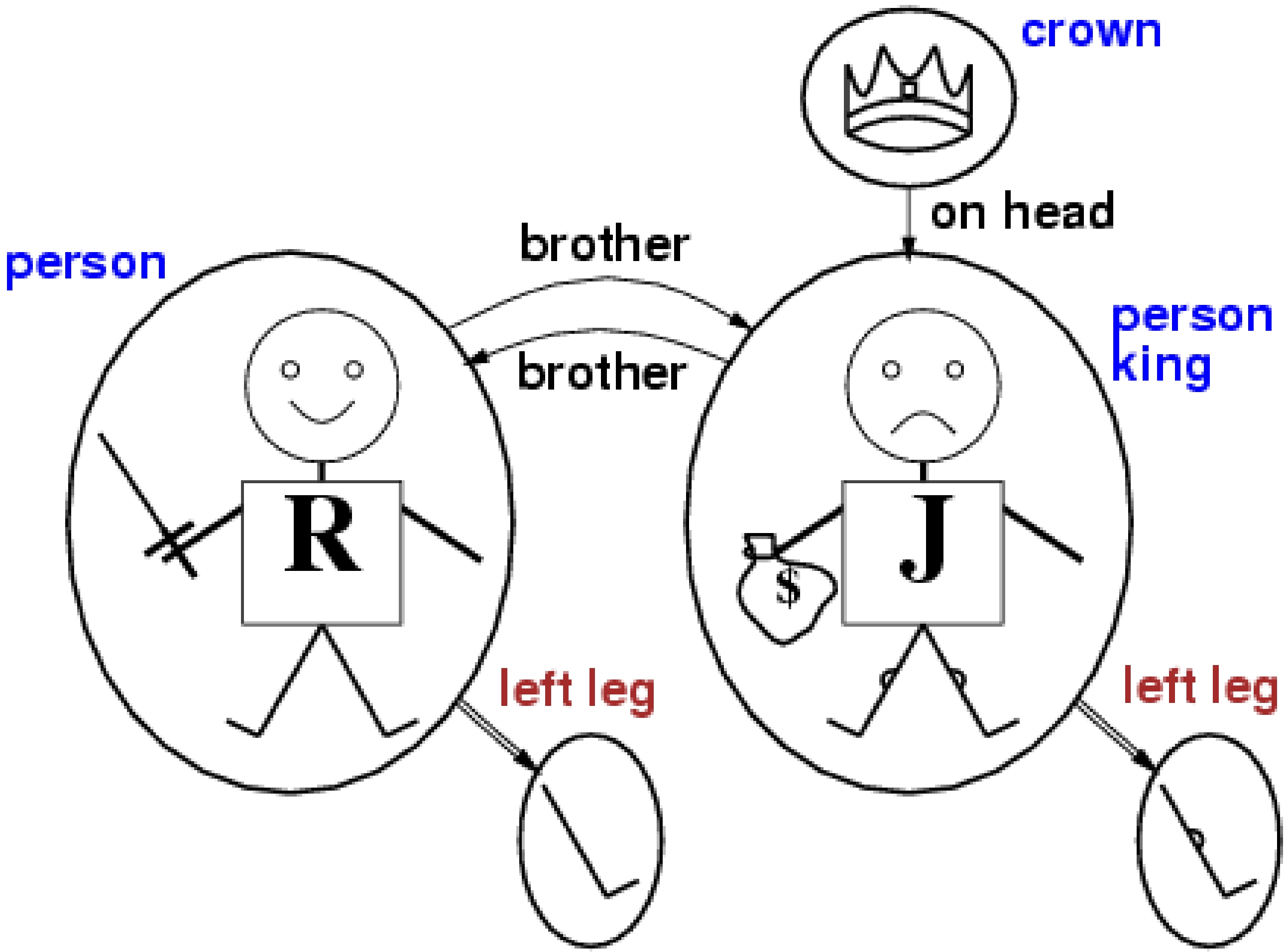
- Sentences are true with respect to a model and an interpretation
- A model contains objects (domain elements) and relations among them
- Interpretation specifies referents for

constant symbols	→	objects
predicate symbols	→	relations
function symbols	→	functions

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

Models for FOL: Example

STI · INNSBRUCK



Truth in the example

STI · INNSBRUCK

Consider the interpretation in which
$Richard$ $\rightarrow$ Richard the Lionheart
$John$ $\rightarrow$ the evil King John
$Brother$ $\rightarrow$ the brotherhood relation

Under this interpretation, $Brother(Richard, John)$ is true just in case Richard the Lionheart and the evil King John are in the brotherhood relation in the model

Universal quantification

STI · INNSBRUCK

- $\forall$<*variables*> <*sentence*>

"Everyone at UIBK is smart":
$\forall x\ At(x,UIBK) \Rightarrow Smart(x)$

- $\forall x\ P$ is true in a model *m* iff *P* is true with *x* being **each** possible object in the model

- Roughly speaking, equivalent to the conjunction of instantiations of *P*

$$\begin{array}{ll} & At(KingJohn, UIBK) \Rightarrow Smart(KingJohn) \\ \wedge & At(Richard, UIBK) \Rightarrow Smart(Richard) \\ \wedge & At(UIBK, UIBK) \Rightarrow Smart(UIBK) \\ \wedge \ldots & \end{array}$$

A common mistake to avoid

STI · INNSBRUCK

- Typically, $\Rightarrow$ is the main connective with $\forall$
- Common mistake: using $\wedge$ as the main connective with $\forall$:

  $\forall x\ At(x,UIBK) \wedge Smart(x)$

  means "Everyone is at UIBK **and** everyone is smart"

- Correct: $\forall x\ At(x,UIBK) \Rightarrow Smart(x)$

Existential quantification

STI · INNSBRUCK

- $\exists$<*variables*> <*sentence*>

- "Someone at UIBK is smart":
- $\exists x$ At(x,UIBK) $\wedge$ Smart(x)

- $\exists x$ *P* is true in a model *m* iff *P* is true with *x* being **some** possible object in the model

- Roughly speaking, equivalent to the disjunction of instantiations of *P*

  At(KingJohn,UIBK) $\wedge$ Smart(KingJohn)
  $\vee$ At(Richard,UIBK) $\wedge$ Smart(Richard)
  $\vee$ At(UIBK,UIBK) $\wedge$ Smart(UIBK)
  $\vee$ ...

Another common mistake to avoid

STI · INNSBRUCK

- Typically, $\wedge$ is the main connective with $\exists$
- Common mistake: using $\Rightarrow$ as the main connective with $\exists$:

  $$\exists x \ At(x,UIBK) \Rightarrow Smart(x)$$

  is true if there is anyone who is not at UIBK!

Usually used in **Queries**:

"Is there someone in UIBK who is smart?"

Correct: $\exists x \ At(x,UIBK) \wedge Smart(x)$

Properties of quantifiers

- ∀x ∀y is the same as ∀y ∀x
- ∃x ∃y is the same as ∃y ∃x

- ∃x ∀y is not the same as ∀y ∃x
- ∃x ∀y Loves(x,y)
  - "There is a person who loves everyone in the world"
- ∀y ∃x Loves(x,y)
  - "Everyone in the world is loved by at least one person"

  - Quantifier duality: each can be expressed using the other
- ∀x Likes(x,IceCream) ¬∃x ¬Likes(x,IceCream)
- ∃x Likes(x,Broccoli) ¬∀x ¬Likes(x,Broccoli)

Equality

STI · INNSBRUCK

- $term_1 = term_2$ is true under a given interpretation if and only if $term_1$ and $term_2$ refer to the same object

- E.g., definition of *Sibling* in terms of *Parent*:

$$\forall x,y\ Sibling(x,y) \Leftrightarrow [\neg(x = y) \land \exists m,f\ \neg(m = f) \land Parent(m,x) \land Parent(f,x) \land Parent(m,y) \land Parent(f,y)]$$

Using FOL

STI · INNSBRUCK

The family domain:

- Brothers are siblings

  $\forall$x,y *Brother(x,y)* $\Rightarrow$ *Sibling(x,y)*

- One's mother is one's female parent

  $\forall$m,c *Mother(c)* = m $\Leftrightarrow$ *(Female(m)* $\wedge$ *Parent(m,c))*

- "Sibling" is symmetric

  $\forall$x,y *Sibling(x,y)* $\Leftrightarrow$ *Sibling(y,x)*

RDFS/OWL Tutorial

http://www.cs.man.ac.uk/~horrocks/ISWC2003/Tutorial/

Summary

- Predicate logics.
- First-order logics.
- RDFS.
- OWL.

Assignments 9/10

STI · INNSBRUCK

1. Translate the following sentences into FOL:
   - Everything is bitter or sweet.
   - Either everything is bitter or everything is sweet.
   - There is somebody who is loved by everyone.
   - Nobody is loved by no one.
   - If someone is noisy, everyone is annoyed.
   - Nobody can ignore her.

2. Show (p -> q) <-> (⌐q -> ⌐ p) is a tautology.

3. Create an OWL ontology (in Protégé) for the following statements:
   1. Each supplier has a unique name.
   2. More than one supplier can be located in the same city.
   3. Each part has a unique part number.
   4. Each part has a color.
   5. A supplier can supply more than one part.
   6. A part can be supplied by more than one supplier.
   7. A supplier can supply a fixed quantity of each part.

Thank You!

Questions?