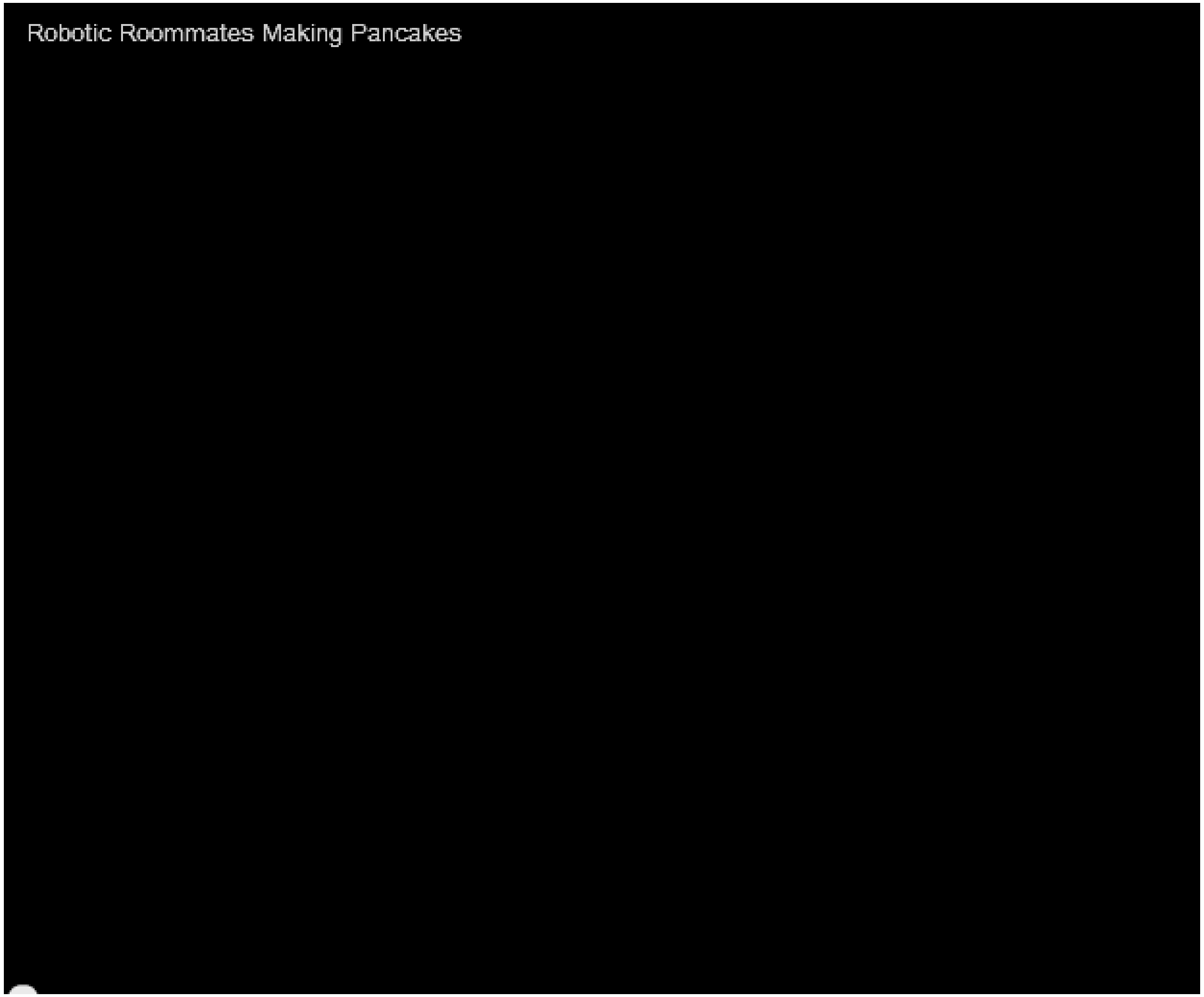
Wissensverarbeitung

- Knowledge Representation & Reaspning I -

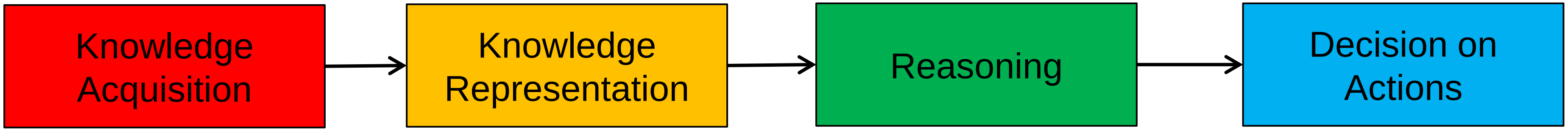
Alexander Felfernig and Gerald Steinbauer
Institute for Software Technology
Inffeldgasse 16b/2
A-8010 Graz
Austria

Motivation

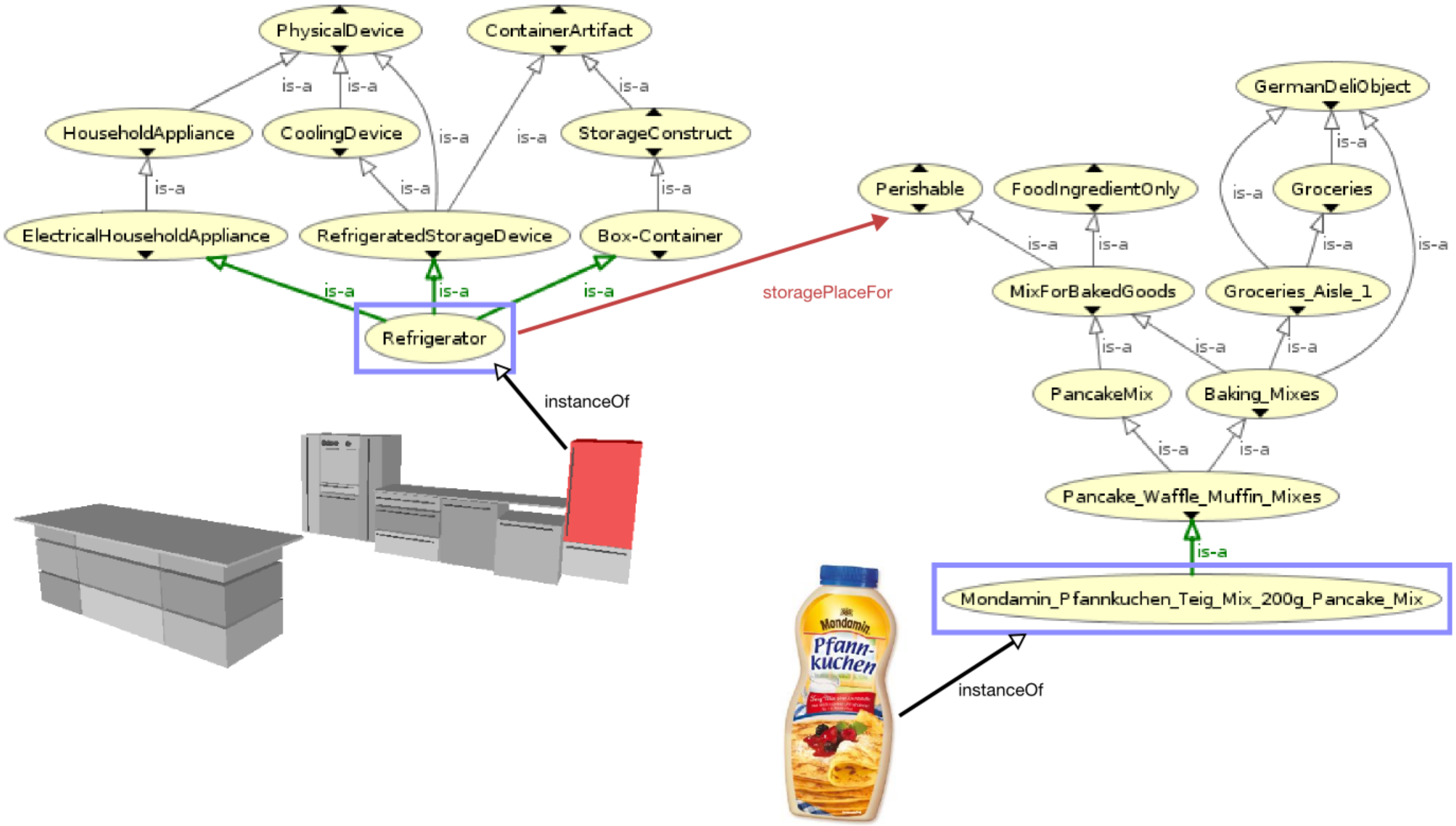


Artificial Intelligence and Knowledge Representation & Reasoning

- Artificial Intelligence (AI) can be described as:
  - the study of intelligent behavior achieved trough computational means
- Knowledge Representation & Reasoning can be viewed as:
  - the study of how to reason (compute) with knowledge in order to decide what to do.



Use Case: Expected Object Location



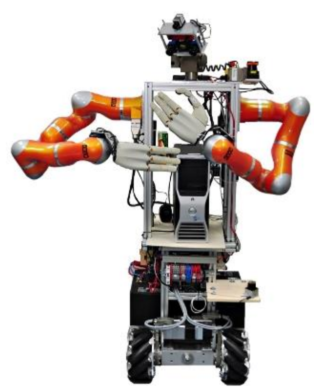
Clear Need for proper KR & R

cycorp

wikiHow
The How-To Manual That You Can Edit

WolframAlpha

Google



amazon.com

RoboEarth
connecting robots worldwide

What is Knowledge?

- easier question: how do we talk about it?
- we say “John knows that ...” and fill the blank with a proposition
  - can be true / false, right / wrong
- contrast: “John fears that ...”
  - same content, different attitude
- other forms of knowledge:
  - know how, who, what, when, ...
  - sensorimotor: typing, riding a bicycle
- belief: not necessarily true and/or held for appropriate reasons
  - and weaker yet: “John suspects that ...”

What is Representation?

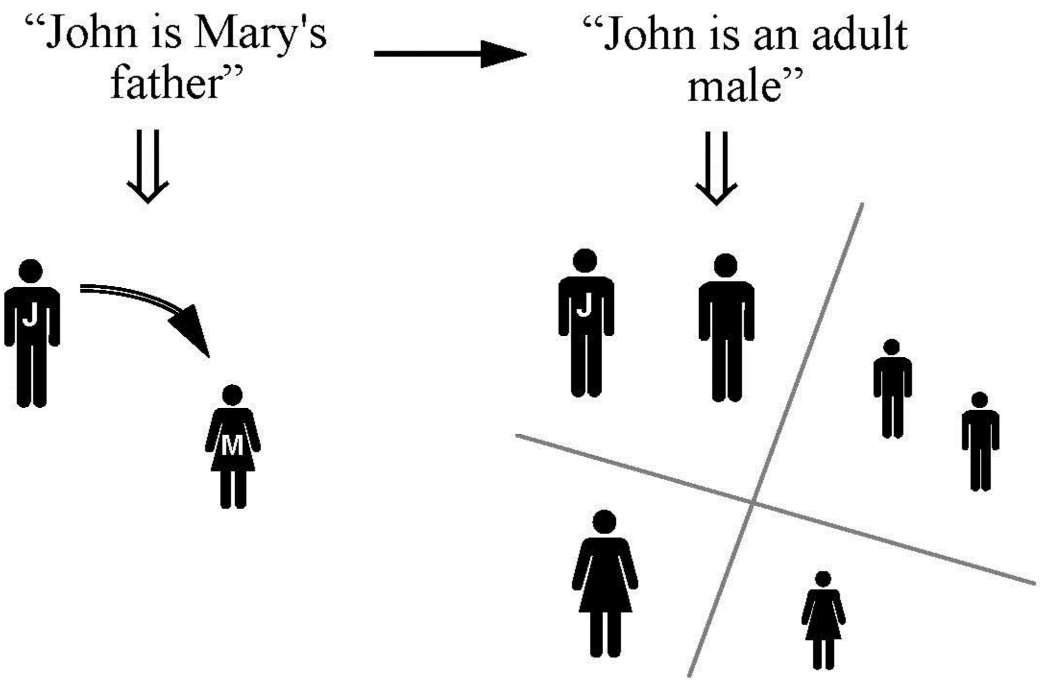
- symbols standing for things in the world

	→	first aid
	→	women
“John”	→	John
“John loves Mary”	→	the proposition that John loves Mary

- Knowledge Representation:
    - symbolic encoding of propositions believed (by some agent)

What is Reasoning?

- manipulation of symbols encoding propositions to produce representations of new propositions
- analogy: arithmetic $1011 + 10 \rightarrow 1101$



Knowledge-Based Systems

EXAMPLE 1:

```
printColour(snow) :- !, write("It's white.").
printColour(grass) :- !, write("It's green.").
printColour(sky) :- !, write("It's yellow.").
printColour(X) :- write("Beats me.").
```

EXAMPL 2:

```
printColour(X) :- colour(X,Y), !, write("It's "), write(Y), write(".").
printColour(X) :- write("Beats me.").
colour(snow,white).
colour(sky,yellow).
colour(X,Y) :- madeof(X,Z), colour(Z,Y).
madeof(grass,vegetation).
colour(vegetation,green).
```

Advantage

- knowledge-based system most suitable for open-ended tasks
  - can structurally isolate reasons for particular behavior
- good for
  - explanation and justification
    - “Because grass is a form of vegetation.”
  - informability: debugging the KB
    - “No the sky is not yellow. It's blue.”
  - extensibility: new relations
    - “Canaries are yellow.”
  - extensibility: new applications
    - returning a list of all the white things
    - painting pictures

KR Requirements

- the following requirements are desirable for KR approaches:
  - expressiveness
  - processibility
  - flexibility
  - modularity
  - understandability
  - representation of uncertainty

Expressiveness

- approach have to be able to represent relevant issues
- approach has to have enough expressional power
  - facts, e.g.: *tom* studies *cs* or: *ann* has *fever*.
  - relations between facts, for example, *rules* such as: if *tom* studies *cs* then *tom* knows about *oo programming languages* or: if *ann* has *fever* then she is *ill*

Processibility

- approach can automatically provide result in time
  - derive new knowledge systematically
  - calculi of formal logic
- approach automatable
  - problems solvable in finite time
  - polynomial behavior ("tractable reasoning")
  - description as an algorithms
  - always tradeoff between processibility and expressiveness

Flexibility

- approach has to be general
- approach applicable in different application domains

Understandability

- used language understandable
- support acquisition of knowledge
- underlying knowledge bases maintainable

Modularity

- structuring mechanisms for knowledge bases
- knowledge changes over time
- support deletion and insertion of knowledge
- changes should have only local effects

Representation of uncertainty

- vague information
  - facts/results are imprecise, for example, *tom is big*
- uncertain information
  - we assume that *Ann is already in Graz*
- uncertain relations
  - *if I drink tap water in Ghana I might get diarrhea*

- … altogether challenges for intelligent systems

Further challenges

- inconsistent knowledge
  - e.g. a robot believes to be at location *A* and *B*
- wrong knowledge
  - e.g. spinach contains a lot of iron
- common-sense/default reasoning
  - for example, "*birds typically fly*"

Our Goals

- what are the goals we will tackle in the course
  - get to know example knowledge representations
  - use this representations for a concrete problem
  - be aware of the modelling issues of different representations
  - be able to automatically reason with different representations (manually and using tools)
  - understand the performance issues of different representations
  - be able to manage specific aspects of KR & R

Content

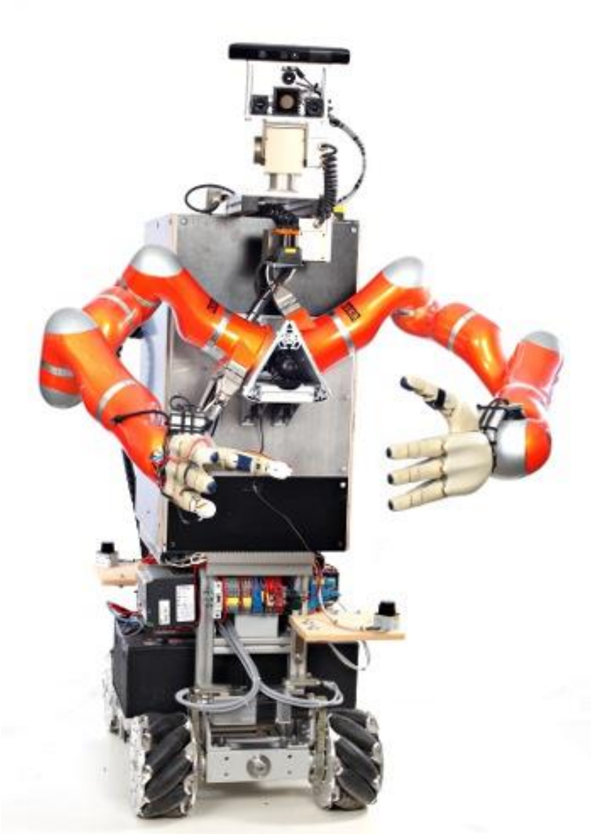
- next four units
- foundations of different representations
- representing knowledge with different representations
- automated reasoning with different representations
- special issues with different representations
- we focus on
  - First Order Logic (FOL)
  - Answer Set Programming (ASP)
  - Description Logic (DL)

Tool Support

- we will use state-of-the-art tools
- please install them for the lecture and the practical work
- FOL
  - Automated Theorem Prover – Prover9 (GUI)
  - https://www.cs.unm.edu/~mccune/mace4/
- ASP
  - Potsdam Answer Set Solving Collection – clingo (command line)
  - http://potassco.sourceforge.net/
- DL
  - ontology editor - Protégé (GUI)
  - http://protege.stanford.edu/

Practical Work

- support ROSIE in setting the table



Logic-based Knowledge Representation

- **logic**: properties of the world (domain) are represented in the form of propositions and sentences.
- **syntax**: expressions (formulae) φ; admissible/syntactically correct sentences
- **semantics**:
  - meaning of the sentences
  - true sentences $\rightarrow$ basis for logical consequences, for example, from φ_1, φ_2, …, φ_n we can derive ψ.
  - inference systems (calculi): allow „calculations“
  - natural representation of facts and rules, for example:

```
from_Crete(Epimenides).
∀x(from_Crete(x) → lies(x)).
logical conclusion: lies(Epimenides).
```

Logic-based Knowledge Representation

- **reference language**
  - first order (predicate) logic
  - most important formalism of logic
  - expressive: all computable functions specifiable (Church‘s thesis)
  - simple, natural syntax and intuitive semantics
- **disadvantages**
  - non-decidable but decidable fragments
  - high computational complexity → further restrictions of expressivity needed

First Order Logic (FOL)

- also known as Predicate Logic
- popular in AI
- more expressive power than others
- FOL handles
  - objects
  - properties
  - relations

Syntax FOL

- FOL builds up formulas
- FOL comprises the following vocabulary
- constant symbols
  - refers to a single object
  - e.g. *Homer*
  - constant symbols need to be interpreted
  - can be used as a name for an object
  - e.g. *Bart*

Syntax FOL

- function symbols
  - represent intuitively a *n-ary* function
  - take *n* arguments (objects)
  - assign exact one object *o* to the arguments, $o=f(o_1,\ldots,o_n)$
  - e.g. *Marge=mother_of*(*Lisa*)
- variable symbols
  - stand for an object which will be bound later at evaluation of a formula
  - can be used in quantifiers
  - e.g *x,y,z*

Syntax FOL

- predicate symbols
  - represent a *n-ary* predicate
  - take *n* arguments (objects)
  - if *n=1* it represents the property of an object, e.g. *smug*(*Lisa*)
  - if *n>1* it represents a relation of objects, e.g. *sibling*(*Maggie*,*Bart*)
  - if *n=0* it represents a property independent to objects, similar to propositions, e.g. *the_simpsons_are_cool*()

Syntax FOL

- terms
  - refer to an object
  - each constant symbol *c* and variable symbol *x* is a term
  - if $t_1,\ldots,t_n$ are terms then $f(t_1,\ldots,t_n)$ is a term, e.g. *father_of*(*Bart*)
- atomic formulae
  - if *P* is a n-ary predicate symbol and $t_1,\ldots,t_n$ are terms then $P(t_1,\ldots,t_n)$ is an atomic formula (atom)

Syntax FOL

- formulas
  - each atomic formula is a formula in *F*
  - if φ_1 and φ_2 are formulae from *F and x* is a variable symbol, then also
    - $\varphi_1 \wedge \varphi_2$ (conjunction), $\varphi_1 \vee \varphi_2$ (disjunction)
    - $\varphi_1 \rightarrow \varphi_2$ (implication), $\varphi_1 \leftrightarrow \varphi_2$ (equivalence)
    - $\neg \varphi_1$ (negation)
    - $\forall x \varphi_1$ (universal quantifier, φ_1 area of binding)
    - $\exists x \varphi_1$ (existential quantifier)
    - e.g. $\forall x[married(x) \rightarrow \exists y(spouse(x,y))]$

Definitions FOL

- free variables
  - a occurrence of a variable x in φ is named free if x is not bound by a quantifier ($\forall$, $\exists$)
  - a variable x is named free in φ if it occurs freely at least once in φ
  - a formula φ with free variables $x_1,..,x_n$ is written as $a(x_1,..,x_n)$
  - is there a free variable ?
    - $loves(Mary,x)$, $p(a) \rightarrow q(a)$
    - $\exists x(p(x) \vee \forall y(p(x,y)))$
    - $\forall x(p(x) \vee \exists x(q(x)))$

Definitions FOL

- sentence
  - is a formula without free variables
  - it is a closed proposition
  - e.g. $\forall y \exists x \neg likes(x,y)$
- universal sentence
  - a sentence of the from $\forall x_1 \forall x_2 \ldots \forall x_n \psi$ and ψ is free of quantifiers
- literal
  - L is an atomic formula (*loves*(*Marge*,*x*)) or its negation (¬*likes*(*Selma*,*Homer*))
- clause
  - C is an disjunction of literals, $C=L_1 \vee \ldots \vee L_n$

Semantics FOL

- defines the meaning of a formula
- defined over interpretations and models
- the interpretation of a vocabulary *V* needs
- a non-empty set *D* of objects (domain, universe)
- an assignment of all symbols over *D*
  - constant symbol *c*: object $I(c) \in D$
  - n-ary function symbol *f*: function $I(f): D^n \rightarrow D$
  - n-ary predicate symbol *P*: relation $I(P) \subseteq D^n$

Semantics FOL

- evaluation of a variable-free term t
  - if t is a constant symbol c then $I(t)=I(c)$
  - if t is of form $f(t_1,\ldots,t_n)$ then $I(t)=I(f)(I(t_1),\ldots,I(t_n))$
- evaluation of a sentence φ
  - a sentence φ can be evaluated to *true* or *false* in I
  - if φ is variable-free atom $P(t_1,\ldots,t_n)$ then $I(\varphi)=true$ if the tuple $(I(t_1),\ldots,I(t_n))$ is in $I(P)$

Semantics FOL

- evaluation of a sentence φ continued
  - if φ is a composition of formulae φ_1 and φ_2 $(\varphi_1 \circ \varphi_2)$ then I(φ) is a propositional composition of $I(\varphi_1)$ and $I(\varphi_2)$ $[I(\varphi_1) \circ I(\varphi_2)]$
    - $I(\varphi_1)$ and $I(\varphi_2)$ are similar to propositions
    - evaluation uses the truth table
  - if $\varphi = \forall x(\psi)$ then $I(\varphi) = true$ if $I(\psi[x/o]) = true$ for all $o \in D$
    - $\psi[x/o]$) is the resulting formula if all occurrences of free variable x in ψ are replaced by o

Semantics FOL

- evaluation of a sentence φ continued
  - if $\varphi = \exists x(\psi)$ then $I(\varphi) = true$ if $I(\psi[x/o]) = true$ for at least one $o \in D$
- evaluation of a FOL sentence φ is in general undecideable
  - e.g. infinite number of objects in D
  - no algorithm for finite time

FOL Definitions

- model: I is a model for φ (written as $I \models \varphi$) iff $I(\varphi)=t$
- I is a model for a set of formulas S (written as $I \models S$) iff $I(\varphi)=t$ for all $\varphi \in S$
- the set of all models for φ or S is named $Mod(\varphi)$ or $Mod(S)$
- a formula φ is a tautology if every interpretation I is a model for φ, $Mod(\varphi)=Int(\varphi)$
  - $((x \equiv y) \wedge (y \equiv z)) \rightarrow (x \equiv z)$

Elementary Properties FOL

- useful for proofs and reasoning
- $S \models \varphi \rightarrow \psi \Leftrightarrow S \cup \{\varphi\} \models \psi$ (deduction theorem)
- $S \models \varphi \Leftrightarrow S \cup \{\neg \varphi\} \models \bot$ (proof by contradiction)
- $\{\varphi, \varphi \rightarrow \psi\} \models \psi$ (modes ponens)
- $S \cup \{\varphi_1 \vee \varphi_2\} \models \psi \Leftrightarrow S \cup \{\varphi_1\} \models \psi$ and $S \cup \{\varphi_2\} \models \psi$ (case analysis)
- $\forall x \, \psi(x) \models \psi[x/t]$, t some term (specialization)

Elementary Properties FOL

- example (Epimenides Paradox)
- given
  - $\forall x(from\_crete(x) \rightarrow lies(x))$
  - $from\_crete(Epimenides)$
- to prove
  - $lies(Epimenides)$

Elementary Properties FOL

- satisfiability
  - if a sentence is valid is undecideable (Church-Theorem)
  - FOL is semi-decidable, there are an algorithm which terminated if φ is a valid sentence
- logic inference systems (Calculi)
  - algorithms to derive valid sentence
  - finite time

Logical KR

- use of FOL
- domain theory
  - a general description of the problem domain
  - e.g. groups in mathematics, geometry
- axioms
  - general valid sentences
- facts
  - specific knowledge
  - literals, e.g., *child*(*Bart*,*Homer*), ¬*likes*(*Moe*,*Bart*)

Logical KR

- general rules
  - universal quantified sentences
  - $\forall x_1 \forall x_2 \ldots \forall x_n (\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow \psi$
  - φ_i: condition, premise
  - ψ: conclusion
  - similar to if-then statement
- universal quantified Horn Clauses
  - all φ_i *and* ψ are atoms
  - subset of FOL, reduced expressiveness
  - e.g., $\forall x (human(x) \rightarrow mortal(x))$

Logical KR

- universe of Discourse
  - in FOL in general no special domain
  - in KR a particular domain *D* and interpretation
  - e.g., arithmetic: {*0*,*1*,*2*,*3*} (constants), {+,*} (functions)

Methodology of Modeling

- what is a good way to model in logic
- no formal approach
- clarity of names, concepts
  - names of constants, predicates etc. are meaningful, e.g. *partent*(*x*,*y*)
- disclose relations
  - $\forall x(senior(x) \rightarrow discount(x))$
  - fact: *senior(Peter)*
  - problem: gender? why discount ?
  - $\forall x \forall y[(y=gender(x) \wedge age(x)>age\_limit(y)) \rightarrow discount(x)]$

Methodology of Modeling

- validation
  - why a sentence gets valid ?
- generality
  - can one express a sentence more generally ?
- requirement of predicates
  - are new predicates necessary ?
  - relation to other predicates
  - description of super/sub classes

Approach

1. conceptualization
   - decide what to represent
   - abstract concept
2. choice of vocabulary
   - translation of the abstract concept to FOL
   - the resulting vocabulary is an ontology of the problem domain
3. coding of the domain theory
   - specify all relations and rules
4. coding of the specific knowledge

Example – Alpine Club

- Tony, Mike and John belong to the Alpine Club. Every member of the Alpine Club who is not a skier is a mountain climber. Mountain climbers do not like rain, and anyone who does not like snow is not a skier. Mike dislikes whatever Tony likes, and likes whatever Tony dislikes. Tony likes rain and snow.
- Prove that the given sentences logically entail that there is a member of Alpine Club who is a mountain climber but not a skier.
- Suppose we had been told that Mike likes whatever Tony dislikes, but we had not been told that Mike dislikes whatever Tony likes. Prove that the resulting set of sentences no longer logically entails that there is a member of Alpine Club who is a mountain climber but not a skier.

Example – Simple Assignment Task

- suppose a simple assignment task
- we have n persons
- we have *m* items of *l* types, $l \leq m$
- each person needs one item of a type
- an item can only be assigned to a single person
- prove if a setup is consistent or not

Logical KR Issues I

- need further axioms for reduced interpretation
- unique names assumption (UNA)
  - different constants $c_1, \ldots, c_n$ refer to different objects
  - $c_1 \neq c_2, \ldots, c_1 \neq c_2$ for all $i \neq j$
  - with UNA *Homer≠Marge* is valid, but not in general

Logical KR Issue II

- domain closure axiom (DCA)
  - in general the domain is defined by the constant symbols $c_1,\ldots,c_n$
  - to prevent the interpretation of further objects, $\forall x(x= c_1 \vee x= c_2 \ldots \vee x= c_n)$
  - set of constant symbols C have to be finite
  - if C is finite and there are no function symbols then interpretation with DCA and UNA = H-interpretation
  - to limit a properties P to certain objects, $\forall x[P(x) \leftrightarrow (x= c_1 \vee x= c_2 \ldots \vee x= c_n)]$

Logical KR Issue II

- sometimes expected sentences are not derivable
- solution closed world assumption (CWA): represent only positive and assume all facts (not derivable) to be false
- CWA is non-monotonic: addition of new facts may limit the number of derivable negative facts

Thank You!