First Order Logic – recap

Marijke Keet

KRDB Research Centre, Faculty of Computer Science
Free University of Bozen-Bolzano, Italy
keet@inf.unibz.it

January 18, 2008

First order logic

The lexicon of a first order language contains:

- Connectives & Parentheses: $\neg$, $\rightarrow$, $\leftrightarrow$, $\wedge$, $\vee$, (and);
- Quantifiers: $\forall$ (universal) and $\exists$ (existential);
- Variables: $x, y, z, \ldots$ ranging over particulars;
- Constants: $a, b, c, \ldots$ representing a specific element;
- Functions: $f, g, h, \ldots$, with arguments listed as $f(x_1, \ldots x_n)$;
- Relations: $R, S, \ldots$ with an associated arity.

First order logic: variable assignments

- Let V be the set of all variables, function $\alpha : V \mapsto \Delta$. Then a *variable assignment* for all free occurrences of x with term d, denoted with $\alpha[x/d]$
- For *closed formulas*, α is omitted, i.e. $\mathcal{I} \models \phi$
  (or: $\mathcal{I}$ is a model of ϕ)
- ϕ is satisfied by $\mathcal{I}$ under α, denoted as: $\mathcal{I}, \alpha \models \phi$
  (or: interpretation $\mathcal{I}$ is a model of ϕ under α)
- ϕ is valid (a tautology) if every $(\mathcal{I}, \alpha)$ is a model of ϕ, denoted as: $\models \phi$
- ϕ is falsifiable if there is some $(\mathcal{I}, \alpha)$ that does not satisfy ϕ
- Σ entails ϕ, denoted as $\Sigma \models \phi$

FOL, domains, model checking

- ..
- ..
- ..

First order logic: equivalences

- Those from PL + New in FOL: see lecture slides p30. In particular,
  $\neg\forall x.\phi \equiv \exists x.\neg\phi$
  $\neg\exists x.\phi \equiv \forall x.\neg\phi$
  $\forall x.\phi \wedge \forall x.\psi \equiv \forall x.(\phi \wedge \psi)$
  $\exists x.\phi \vee \exists x.\psi \equiv \exists x.(\phi \vee \psi)$
  (the others deal with free & bound variables)
- The $\forall$ definition: $\forall x\phi(x)$ if and only if $\neg\exists x\neg\phi(x)$
- Use them for rewriting formulas, e.g for proving two formulas are equivalent, and
- Just like with tableaux for PL, we may need them for rewriting a formula suitable for a FOL tableaux

Tableau

- Finds a model for a given collection of sentences in negation normal form.
- Refutation tree, completion rules, apply rules until (a) an explicit contradiction (clash) or (b) there is a completed branch and no more rule is applicable (a completed branch gives a model of KB: the KB is satisfiable)
- $\phi \models \psi$ iff $\phi \wedge \neg\psi$ is NOT satisfiable—if it is satisfiable, we have found a counterexample

Tableau

- Basic completion rules:
  - If a model satisfies a conjunction, then it also satisfies each of the conjuncts
  - If a model satisfies a disjunction, then it also satisfies one of the disjuncts. It is a non-deterministic rule, and it generates two alternative branches.
- Completion rules for quantified formulas:
  - If a model satisfies an existentially quantified formula, then it also satisfies the formula where that quantified variable has been substituted with a new *skolem constant*: for $\exists x \phi(x)$ we get, e.g., $\phi(c)$
  - If a model satisfies a universally quantified formula, then it also satisfies the formula where the quantified variable has been substituted with *some term*: for $\forall x \phi(x)$ we get, e.g., $\phi(a)$

Overview

- FOL: puzzle (from natural language to FOL formula)
- FOL: pizza KB
- FOL: theories and interpretations with Tarski's World
- PL: equivalence game (and truth tables and tableau)

FOL puzzle

- Consider the following problem concerning houses in Potter square.
  1. The houses in Potter square are either blue, red or yellow.
  2. If a house in Potter square is close to a blue house in the same square then it cannot be blue.
  3. If two houses in Potter square are close and one of them is not blue then one of them is red.
  4. The house number 6 in Potter square is yellow and close to a house in the same square which is not blue.
  5. No house in Potter square is close to itself.
- Formalise the story in a suitable first-order language then determine if the story is consistent (that is, satisfiable).

FOL, domains, model checking

- Start Tarski's world, open Arnault.sen and create a world in which all ten sentences are true.
- Save your world, and compare with your classmates if you have made exactly the same world or if some blocks are maybe on another place on the raster etc.
- Move or resize one block at a time and check what effect it has on T/F of the sentences.

Pizzas (1/2)

- Let us have a logical theory Θ with 12 sentences
  (1) $\forall x Pizza(x)$, (2) $\forall x PizzaT(x)$, (3) $\forall x PizzaB(x)$,
  which are disjoint (4) $\forall x(Pizza(x) \rightarrow \neg PizzaT(x))$, (5)
  $\forall x(Pizza(x) \rightarrow \neg PizzaB(x))$, (6) $\forall x(PizzaT(x) \rightarrow \neg PizzaB(x))$,
  and we have two relations:
  (7) $\forall x, y(hasT(x, y) \rightarrow Pizza(x) \wedge PizzaT(y))$,
  (8) $\forall x, y(hasB(x, y) \rightarrow Pizza(x) \wedge PizzaB(y))$
  In addition, (9) $\forall x(ITPizza(x) \rightarrow Pizza(x))$ and
  (10) $\forall x(ITPizza(x) \rightarrow \neg\exists y(hasT(x, y) \wedge FruitT(y))$, where (11)
  $\forall x(VegeT(x) \rightarrow PizzaT(x))$ and (12) $\forall x(FruitT(x) \rightarrow PizzaT(x))$.
  A Pizza margherita has the necessary and sufficient conditions that it has mozzarella, tomato, basilicum and oil as toppings and has a pizza base. Add this to Θ.
- Annotate you commitments: what have you added to Θ and how?
- Hint: fruits are not vegetables, categorise the toppings, and "necessary and sufficient" $\leftrightarrow$

Pizzas (2/2)

- We want to merge our new Θ with some other theory Γ that has knowledge about fruits and vegetables. Γ contains, among other formulas, $\forall x(\mathit{Tomato}(x) \rightarrow \mathit{Fruit}(x))$. What happens? Represent the scenario formally, and prove your answer.
- Actually, a decidable fragment of FOL, $\mathcal{SHOIN}(D)$ is used for the Pizza Ontology (logical theory Θ'). To get an idea, a (non-logician) user-perspective can be found in the Pizza Ontology Tutorial: http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf.

PL: truth tables, equivalences

- Game: take some arbitrary PL sentence, obfuscate it using (at least) the PL equivalence rules for several steps. Give the start and final formula to your neighbour who has to prove the (non) equivalence. Repeat for other formulas. The one who proved most, wins.
- Recollect that, depending on your classmate's "inventiveness", making a truth table can help to prove your answer (in particular if the two formulas are not equivalent)
- The game can be easily extended to tableaux
- Or the same approach, but then for FOL formula