More First-Order Logic

CS156

Chris Pollett

Oct 27, 2014

Outline

- Using First-order Logic
- Theorem Proving
- Unification

Introduction

- Last week, we started talking about First-Order Logic.
- When we were talking about search we switched from problem solving agents which worked on atomic domains, to constraint satisfaction problems where the domain was factored.
- In similar fashion, we have talked about logic-based agents and we have said that one way to represent their knowledge bases is with propositional logic.
- This works well for atomic domains, but now we are interested in logic-based agents that can work with factored domains, where we have real world objects whose properties can take on more than binary values.
- First-order logic is our approach for doing this.
- In first-order logic, we have constants and variables which run over domains. We can build up terms using these, and the functions of our first-order logic. We can substitute terms into the predicates of our logic to make atomic formulas, and we can build general first-order formulas out of these using $\neg, \wedge, \vee, \forall x, \exists y$.
- At the end of last day we started to give the Tarski definition of truth of a formula F in a model M, but didn't finish this.
- Today, we start with this definition and then start talking about knowledge bases, but in the first-order setting.

Semantics 1st order formulas

- To start we need to have a set M called the **universe** that our variables range over.
- For each constant c in our language, we need to give a value c^M in M.
- Here the superscript means in the model, not exponentiation.
- For each function f in our language need to give an actual function
  $f^M : M \times ... \times M \to M$ determined on M
- For each predicate symbol P,
  $P^M : M \times ... \times M \to \{T, F\}$
- All of the above information together is called a structure/model. Usually, denoted by M.
- Can almost now talk about the meaning of a formula F in a given structure M...
- Consider:
  $(\exists x)G(x, y)$
  in this formula x is called a **bound variable** as it is in the scope of an $\exists$ or $\forall$, y is **unbound**.
- To give the meaning of a formula, we also need an unbound variable assignment. Here **variable/object assignment** ν is a map from unbound variables to elements of M. We can finally write $M \models F[v]$ (M entails F with assignment ν)

What is the meaning of a formula F?

- If F is an atomic formula calculate the value in M of each of the terms in F. Plug these values into the predicate P^M and see if outputs T or F.
- If $F := \neg G$, then F is true in M, ν iff not $M \models G[\nu]$.
- If $F := G \vee H$, then $M \models F[\nu]$ iff $M \models G[\nu]$ or $M \models H[\nu]$
- If $F := G \wedge H$ then $M \models F[\nu]$ iff $M \models G[\nu]$ and $M \models H[\nu]$
- Other connectives are similar.
- If $F := (\exists x)G$ then $M \models F[\nu]$ iff there is some way to map $x \rightarrow x^M \in M$ such that ν is extended by this additional mapping give $M \models G\left[x \rightarrow x^M, v\right]$
- If $F := (\forall x)G$ then $M \models F[\nu]$ iff for all ways to map $x \rightarrow x^M \in M$, $M \models G\left[x \rightarrow x^M, \nu\right]$.
- Write $M \models F$ if for every variable assignment, $\nu, M \models F[v]$

Example

- Suppose we have the language: 0,1, $+$, $\cdot$, $=$ consider the formula $(\exists x)(1+1)\cdot x = 1+1+1+1$.
- This intuitively asserts there is some number such that twice that number is four.
- All variables in this formula are bound.
- Let our model M be the natural numbers, $\mathbb{N}$: 0,1,2,3 ...
- Recall we will use superscript to mean in the model of the natural numbers, not exponentiation.
- We interpret 0^M to be 0, zero in the natural numbers and 1^M to be 1, one in the natural numbers.
- We interpret the functions $+^M$ and $\cdot^M$ as the usual plus and times on the natural numbers.
- We interpret the predicate $=^M$ as the usual equality of two natural numbers
- From the last slide $M \models (\exists x)(1+1)\cdot x = 1+1+1+1$ iff there is some variable $x \to x^M \in M$ such that $M \models (1+1)\cdot x = 1+1+1+1\left[x \to x^M\right]$.
- Suppose we map x to 2^M. Then the statement holds if $\left(1^M +^M 1^M\right)\cdot 2^M =^M 1^M +^M 1^M +^M 1^M +^M 1^M$. Now both sides, after computing the values in the model based on 1^M, 2^M, and $+^M$, are 4^M. Thus, the statement does hold for our model.
- Consider the statement $(\exists x)(1+1)\cdot x = 1+1+1$. In our natural number model there is no assignment to $x \to x^M$ that could make this statement true. so M does not model (exists x) (1+1)cdot x = 1+1+1`

- Consider the model M' with universe $\{0, 1, 2, 3, 4\}$ and where we interpret $+$ and $\cdot$ using the integers mod 5. In this model both, $(\exists x)(1+1)\cdot x = 1+1+1+1$ and $(\exists x)(1+1)\cdot x = 1+1+1$ will hold. To see this for the second statement, notice that $2 \cdot 4 \bmod 5 \equiv 3$ This shows that a statement can be true in more than one model and it also shows just because a statement is false in one model doesn't mean it is false in all models.

KBs, First-Order, Proofs

- Write $KB \models F$ to mean for all structures M, such that $M \models KB$ we also have $M \models F$
- Proofs in 1st Order Logic are very much like proofs in propositional logic except now we have some additional axioms.
- For example, some possible additional axioms might be:
  $(\forall x)(\neg P) \Rightarrow \neg((\exists x)P)$
  (for every pig, it can't fly = not there is a pig that can fly)
  $\neg(\forall x P) \Rightarrow (\exists x)\neg P$
  (Not for every horse it is blue = there is a horse that is not blue)
  $A(t) \Rightarrow (\exists x)A(x)$
  $A(x) \Rightarrow (\forall y)A(y)$
  etc.

Quiz

Which of the following is true?

1. Every satisfiable propositional formula is valid.
2. A Horn program is allowed to have clauses with two positive literals.
3. In DPLL if there is a Pure Symbol we assign it so as to make it true.