First Order Logic

Pete Manolios
Northeastern

Formal Methods, Lecture 10

October 2008

First Order Logic

- Example: Group Theory
  - (G1) For all x, y, z: $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
  - (G2) For all x: $x \cdot e = x$
  - (G3) For all x there is a y such that: $x \cdot y = e$
- Theorem: For every x, there is a y such that $y \cdot x = e$
- Proof:

  By (G3) there is: a y s.t. $x \cdot y = e$ and a z s.t. $y \cdot z = e$

  Now: $y \cdot x = y \cdot x \cdot e = y \cdot x \cdot y \cdot z = y \cdot e \cdot z = y \cdot z = e$
- Is this true for all groups? Why?
- How many groups are there?
- Are there true statements about groups with no proof?

First Order Logic

- First Order Logic forms the foundation of mathematics
- We study various objects, e.g., groups
- Properties of objects captured by “non-logical” axioms
  - (G1-G3 in our example)
- Theory consists of all consequences of “non-logical” axioms
  - Derivable via logical reasoning alone
  - That’s it; no appeals to intuition
- Separation into non-logical axioms logical reasoning is astonishing: all theories use exactly same reasoning
- But, what is a proof ($\Phi \vdash \varphi$)?
- Question leads to computer science
- Proof should be so clear, even a machine can check it

First Order Logic: Syntax

- Every FOL (first order language) includes
  - Variables $v_0, v_1, v_2, \ldots$
  - Boolean connectives: $\vee, \neg$
  - Equality: $=$
  - Parenthesis: (,)
  - Quantifiers: $\exists$
- The symbol set of a FOL contains (possibly empty) sets of
  - relation symbols, each with an arity > 0
  - function symbols, each with an arity > 0
  - constant symbols
- Example: groups 2-ary function symbol $\bullet$ and constant e
- Set theory: $\in$, a 2-ary relation symbol, ...

First Order Logic: Terms

- Terms denote objects of study, e.g., group elements
- The set of *S*-terms is the least set closed under:
  - Every variable is a term
  - Every constant is a term
  - If $t_1, \ldots, t_n$ are terms and f is an n-ary function symbol, then $f(t_1, \ldots, t_n)$ is a term

First Order Logic: Formulas

- Formulas: statements about the objects of study
- An atomic formula of S is
  - $t_1 = t_2$ or
  - $R(t_1, ..., t_n)$, where t_i is an S-term and R is an n-ary relation symbol in S
- The set of S-formulas is the least set closed under:
  - Every atomic formula is a formula
  - If φ, ψ are S-formulas and x is a variable, then $\neg\varphi$, $(\varphi \vee \psi)$, and $\exists x\varphi$ are S-formulas
- All Boolean connectives can be defined in terms of $\neg$ and $\vee$
- We can define $\forall x\varphi$ to be $\neg\exists x\neg\varphi$

Definitions on Terms & Formulas

- Define the notion of a free variable for an *S*-formula
- The definition of formula depends on that of term
- So, we're going to need an auxiliary definition:

  $var(x) = \{x\}$

  $var(c) = \{\}$

  $var(f(t_1, ..., t_n)) = var(t_1) \cup \cdots \cup var(t_n)$
- Is this a definition?

  $free(t_1 = t_2) = var(t_1) \cup var(t_2)$

  $free(R(t_1, ..., t_n)) = var(t_1) \cup \cdots \cup var(t_n)$

  $free(\neg\varphi) = free(\varphi)$

  $free((\varphi \vee \psi)) = free(\varphi) \cup free(\psi)$

  $free(\exists x\varphi) = free(\varphi) \setminus \{x\}$

Semantics of First Order Logic

- What does $\exists v_0 R(v_0, v_1)$ mean?
- It depends on:
  - What R means (what relation over what domain?)
  - What v_1 means (what element of the domain?)
- What if the is domain $\mathbb{N}$, R is $<$, and v_1 is 1? If v_1 is 0?
- An S-interpretation $\mathcal{J} = \langle A, a, \beta \rangle$ where
  - A is a non-empty set (domain or universe)
  - a is a function with domain S
  - β: $Var \rightarrow A$ is an assignment
  - If $c \in S$ is a constant, then $a.c \in A$
  - If $f \in S$ is an n-ary function symbol, then $a.f : A^n \rightarrow A$
  - If $R \in S$ is an n-ary relation symbol, then $a.R \subseteq A^n$

Meaning via Interpretations

- The meaning of a term in an interpretation $\mathcal{J} = \langle A, a, \beta \rangle$
  - If $v \in Var$, then $\mathcal{J}.v = \beta.v$
  - If $c \in S$ is a constant, then $\mathcal{J}.c = a.c$
  - If $f(t_1, \ldots, t_n)$ is a term, then $\mathcal{J}(f(t_1, \ldots, t_n))$ is $(a.f)(\mathcal{J}.t_1, \ldots, \mathcal{J}.t_n)$
- What it means for an interpretation to satisfy a formula:
  - $\mathcal{J} \models (t_1 = t_2)$ iff $\mathcal{J}.t_1 = \mathcal{J}.t_2$
  - $\mathcal{J} \models R(t_1, \ldots, t_n)$ iff $\langle \mathcal{J}.t_1, \ldots, \mathcal{J}.t_n \rangle \in a.R$
  - $\mathcal{J} \models \neg\varphi$ iff not $\mathcal{J} \models \varphi$
  - $\mathcal{J} \models (\varphi \vee \psi)$ iff $\mathcal{J} \models \varphi$ or $\mathcal{J} \models \psi$
  - $\mathcal{J} \models \exists x\varphi$ iff for some $b \in A$, $\mathcal{J}(x \leftarrow b) \models \varphi$

Models & Consequence

- Let Φ be a set of formulas and φ a formula
- $\mathscr{I} \models \Phi$ ($\mathscr{I}$ is a model of Φ) iff for every $\varphi \in \Phi$, $\mathscr{I} \models \varphi$
- $\Phi \models \varphi$ (φ is a consequence of Φ) iff for every interpretation, $\mathscr{I}$, which is a model of Φ, we have that $\mathscr{I} \models \varphi$

- A formula φ is satisfiable, written *Sat* φ, iff there is an interpretation which is a model of φ
- A set of formulas Φ is satisfiable (*Sat* Φ), iff there is an interpretation which is a model of all the formulas in Φ

- Lemma: For all φ, Φ: $\Phi \models \varphi$ iff not *Sat* $(\Phi \cup \{\neg\varphi\})$

Proof Theory

- $\Phi \vdash \varphi$ denotes that φ is provable from Φ
- Provability should be machine checkable
- It may seem hopeless to nail down what a proof is
  - Don’t mathematicians expand their proof methods?
- FOL has a fairly simply set of obvious rules
- There are many equivalent ways of defining proof

Sequent Calculus

- A sequent is a nonempty sequence of formulas
- Sequent rules:

$$\frac{\begin{array}{l}\Gamma \quad \neg\varphi \quad \psi \\ \Gamma \quad \neg\varphi \quad \neg\psi\end{array}}{\Gamma \quad \varphi} \qquad\qquad \frac{}{\Gamma \quad \varphi}\ \text{if } \varphi \text{ is a member of } \Gamma$$

- The left rule says if you have a proof of both $\neg\psi$ and ψ from $\Gamma \cup \{\neg\varphi\}$, that constitutes a proof of φ from Γ
- If there is a derivation of the sequent $\Gamma\ \varphi$, then we write $\vdash \Gamma\ \varphi$ and say that $\Gamma\ \varphi$ is *derivable*
- A formula φ is *formally provable* or *derivable* from a set Φ of formulas, written $\Phi \vdash \varphi$, iff there are finetely many formulas $\varphi_1, \ldots, \varphi_n$ in Φ s.t. $\vdash \varphi_1 \ldots \varphi_n\ \varphi$

Gödel's Completeness Theorem

- While we haven't shown a full proof system, the following turns out to be easy to show:
- $\Phi \vdash \varphi$ implies $\Phi \vDash \varphi$
- What about the converse?
- Gödel's completeness theorem: $\Phi \vDash \varphi$ implies $\Phi \vdash \varphi$
- Lemma: Con Φ implies Sat Φ
- Φ is consistent, written Con Φ, ff there is no formula φ such that $\Phi \vdash \varphi$ and $\Phi \vdash \neg\varphi$
- Proof: $\Phi \vDash \varphi$

  iff {previous lemma} not Sat $(\Phi \cup \{\neg\varphi\})$

  iff {above lemma, soundness} not Con $(\Phi \cup \{\neg\varphi\})$

  iff {hint: use first sequent rule} $\Phi \vdash \varphi$

Gödel's Completeness Theorem

- $\Phi \vdash \varphi$ iff $\Phi \models \varphi$
- What does this mean for group theory?
- What about new proof techniques?
- Once we show the equivalence between $\vdash \varphi$ and $\models$, we can transfer properties of one to the other
  - Compactness theorem:
    (a) $\Phi \models \varphi$ iff there is a finite $\Phi_0 \subseteq \Phi$ such that $\Phi_0 \models \varphi$
    (b) Sat Φ iff for all finite $\Phi_0 \subseteq \Phi$, Sat Φ_0
- From the proof, we get the Löwenheim-Skolem theorem: Every satisfiable and at most countable set of formulas is satisfiable over a domain which is at most countable

Gödel's 1st Incompleteness Theorem

- A set is *recursive* iff $\in$ can be decided by a Turing machine
- Assuming Con(ZF), the set $\{\varphi : ZF \vdash \varphi\}$ is not recursive
- More generally, for any consistent extension C of ZF:
  - $\{\varphi : C \vdash \varphi\}$ is not recursive
  - Intuitively clear: embed Turing machines in set theory
  - Encode halting problem as a formula in set theory
- Theorem: If C is a recursive consistent extension of ZF, then it is incomplete, i.e., there is a formula φ such that $C \nvdash \varphi$ and $C \nvdash \neg\varphi$
- Proof Outline: If not, then for every φ, either $C \vdash \varphi$ or $C \vdash \neg\varphi$. We can now decide $C \vdash \varphi$: enumerate all proofs of C. Stop when a proof for φ or $\neg\varphi$ is found

Gödel's 2nd Incompleteness Theorem

- TM_n is the n^{th} Turing machine
- TM is a Turing machine that given input n:
  - Searches for a proof in PA that "TM_n does not halt at n"
  - If it finds a proof, TM halts; otherwise TM does not halt
- Let TM be TM_k. What if we run TM_k at k?
- Case 1. There is a proof in PA that "TM_k does not halt at k", so:
  - TM_k halts at k
  - But then PA proves "TM_k halts at k"
  - Since Con(PA), this is impossible
- Case 2. (*)There is no proof in PA that "TM_k does not halt at k"
  - Then (⧺)TM_k does not halt at k
- We proved: (⧺) and (*), the 1st Incompleteness theorem for PA
- Also, if PA can prove Con(PA), then PA can prove (*), (⧺)
- Thus, PA would prove: (*) & PA proves "TM_k does not halt at k"
- Hence Inc(PA); thus PA cannot prove its own consistency

FOL Observations

- In ZF, the axiom of choice is neither provable nor refutable
- In ZFC, the continuum hypothesis is neither provable nor refutable
- By Gödel's first incompleteness theorem, no matter how we extend ZFC, there will always be sentences which are neither provable nor refutable
- There are non-standard models of $\mathbb{N}$, $\mathbb{R}$ (un/countable)
- Since any reasonable proof theory has to be decidable, and TMs can be formalize in FOL (set theory), any logic can be reduced to FOL
- Building reliable computing systems requires having programs that can reason about other programs and this means we have to really understand what a proof is so that we can program a computer to do it