Lecture 18

Pete Manolios
Northeastern

EXAM 1

- Tomorrow
- In class
- One page of notes is allowed
- Topics
  - ACL2s: language, defdata, definec, proofs, termination, induction, rewriting, simplification, etc.
  - Propositional logic results & algorithms (2SAT, BDDs, CNF, DNF, Resolution, DP, DPLL, etc)
  - FOL: syntax, semantics, formalization, results, Prenex Normal Form, Skolemization

Reduce FOL to Propositional SAT

- We reduced FOL SAT to SAT of the universal fragment
- We now go one step further

ground: quantifier/variable free

- Theorem: A universal FO formula (w/out =) is SAT iff all finite sets of ground instances are (propositionally) SAT (eg $P(x) \vee \neg P(x)$ is propositionally SAT)
- Corollary: A universal FO formula (w/out =) is UNSAT iff some finite set of ground instances is (propositionally) UNSAT
- FO validity checker: Given FO ϕ, negate & Skolemize to get universal ψ s.t. Valid(ϕ) iff UNSAT(ψ). Let G be the set of ground instances of ψ (possibly infinite, but countable). Let G_1, G_2 …, be a sequence of finite subsets of G s.t. $\forall g \subseteq G, |g| < \omega$, $\exists n$ s.t. $g \subseteq G_n$. If $\exists n$ s.t. Unsat G_n, then Unsat ψ and Valid ϕ
- The SAT checking is done via a propositional SAT solver!
- If ϕ is not valid, the checker may never terminate, i.e., we have a semi-decision procedure and we'll see that's all we can hope for
- How should we generate G_i? One idea is to generate all instances over terms with at most 0, 1, … , functions. We'll explore that more later.

Example

$\langle \exists x \langle \forall y\ P(x) \Rightarrow P(y)\rangle\rangle$ **is Valid iff** $\langle \forall x \langle \exists y\ P(x) \wedge \neg P(y)\rangle\rangle$ **is UNSAT**

iff $\langle \forall x\ P(x) \wedge \neg P(f_y(x))\rangle$ **is UNSAT**

with smart Skolemization **iff** $\langle \forall x\ P(x) \wedge \neg P(c)\rangle$ **is UNSAT**

- *Herbrand universe* of FO language L is the set of all ground terms of L, except that if L has no constants, we add c to make the universe non-empty.
- For our example we have $H = \{c, f_y(c), f_y(f_y(c)), \ldots\}$
- So $G = \{P(t) \wedge \neg P(f_y(t)) \mid t \in H\}$
- Notice that $\Delta = \{P(c) \wedge \neg P(f_y(c)), P(f_y(c)) \wedge \neg P(f_y(f_y(c)))\}$ is UNSAT
  - the SAT solver will report UNSAT for: $P(c) \wedge \neg P(f_y(c)) \wedge P(f_y(c)) \wedge \neg P(f_y(f_y(c)))$
- So, for the first G_i that has both $\neg P(f_y(c))$ and $P(f_y(c))$ will lead to termination
- BTW, why do we restrict ourselves to FO w/out equality?
  - Consider $P(c) \wedge \neg P(d) \wedge c{=}d$
  - $H = \{c,d\}$
  - $G = \{P(c) \wedge \neg P(d) \wedge c{=}d\}$, which is propositionally SAT, but FO UNSAT
- This is why smart Skolemization is useful

Propositional Compactness

- A set Γ of propositional formulas is SAT iff every finite subset is SAT
- This is a key theorem justifying the correctness of our FO validity checker
- Proof: Ping is easy. Let p_1, p_2, …, be an enumeration of the atoms (assume the set of atoms is countable). Define Δ_i as follows
  - $\Delta_0 = \Gamma$
  - $\Delta_{n+1} = \Delta_n \cup \{p_{n+1}\}$ if this is finitely SAT
  - $\Delta_{n+1} = \Delta_n \cup \{\neg p_{n+1}\}$ otherwise

  Note: for all i, Δ_i is finitely SAT as is $\Delta = \cup_i \Delta_i$ (any finite subset is in some Δ_i)

  Here is an assignment for Γ: $v(p_i)$ = true iff $p_i \in \Delta$