Lecture 19

Pete Manolios
Northeastern

Herbrand Interpretations

- 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)
- Let ψ be a universal FO formula w/out equality
- Let H be the Herbrand universe (all ground terms in language of ψ, as before)
- If G (all ground instances of ψ) is propositionally UNSAT then ψ is UNSAT (universal formulas imply all their instances)
- If G is propositionally SAT, say with assignment v, then ψ is SAT
  - Let $\mathcal{I}$ be a canonical interpretation where the universe is H and
  - constants are interpreted autonomously: $a(c) = c$
  - functions are interpreted autonomously: $a(f\ t_1 \ldots t_n) = f\ t_1 \ldots t_n$
  - relations are interpreted as follows: $\langle t_1, \ldots, t_n \rangle \in a.R$ iff $v(R\ t_1, \ldots, t_n) = \text{true}$
  - variables are mapped to terms (how doesn't matter)
- Notice that $\mathcal{I} \vDash ψ$. We need to check that for all vars $x_1, \ldots, x_n$ in ψ, and for all $t_1, \ldots, t_n$ in H, $\mathcal{J}\frac{t_1 \ldots t_n}{x_1 \ldots x_n} \vDash \psi$ iff $\mathcal{J}\frac{\mathcal{J}(t_1) \ldots \mathcal{J}(t_n)}{x_1 \ldots x_n} \vDash \psi$ iff $\mathcal{J} \vDash \psi\frac{t_1 \ldots t_n}{x_1 \ldots x_n}$ which holds by construction since G contains all ground instances

FOL Checking

- 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. $g \subseteq G_n$. $\exists n$ s.t. Unsat G_n iff Unsat ψ (and Valid ϕ)
- Question 1: SAT checking
  - Gilmore (1960): Maintain conjunction of instances so far in DNF, so SAT checking is easy, but there is a blowup due to DNF
  - Davis Putnam (1960): Convert ψ to CNF, so adding new instances does not lead to blowup
  - In general, any SAT solver can be used, eg, DPLL much better than DNF
- Question 2: How should we generate G_i?
  - Gilmore: Instances over terms with at most 0, 1, … , functions
  - Any such “naive” method leads to lots of useless work, eg, the book has code for minimizing instances and reductions can be drastic

Unification

- Better idea: intelligently instantiate formulas. Consider the clauses

  $\{P(x, f(y)) \vee Q(x, y), \neg P(g(u), v)\}$
- Instead of blindly instantiating, use $x=g(u)$, $v=f(y)$ so that we can resolve

  $\{P(g(u), f(y)) \vee Q(g(u), y), \neg P(g(u), f(y))\}$
- Now, resolution gives us

  $\{Q(g(u), y)\}$
- Much better than waiting for our enumeration to allow some resolutions
- Unification: Given a set of pairs of terms $S = \{(s_1,t_1), \ldots, (s_n,t_n)\}$ a *unifier* of S is a substitution σ such that $s_i|\sigma = t_i|\sigma$
- We want an algorithm that finds a *most general* unifier if it exists
  - σ is *more general* than τ, $\sigma \leq \tau$, iff $\tau = \delta \circ \sigma$ for some substitution δ
  - Notice that if σ is a unifier, so is $\delta \circ \sigma$
- Similar to solving a set of simultaneous equations, e.g., find unifiers for
  - $\{(P(f(w), f(y)), P(x, f(g(u)))), (P(x,u), P(v,g(v)))\}$ and $\{(x, f(y)), (y, g(x))\}$

Exam 1 Review