Lecture 26

Pete Manolios
Northeastern

Dealing with Equality

- Plan for a FO validity checker w/=: Given FO ϕ, negate & Skolemize to get universal ψ s.t. Valid(ϕ) iff Unsat(ψ). Convert ψ into equivalent CNF $\mathscr{K}$. Generate ψ* in expanded language without = s.t. Sat(ψ) iff Sat(ψ*). Use U-Resolution on ψ*.
- To go from ψ to ψ*
  - Introduce a new binary relation symbol, *E*
  - Replace $t_1=t_2$ *with* $E(t_1, t_2)$ everywhere in ψ
  - Force *E* to be an equivalence relation by adding clauses
    - $\{E(x,x)\}$, $\{\neg E(x,y), E(y,x)\}$, $\{\neg E(x,y), \neg E(y,z), E(x,z)\}$
    - $\{E(x,x)\}$, $\{\neg E(x,y), \neg E(x,z), E(y,z)\}$ Can replace symmetry, transitivity with this!
  - Force *E* to be a congruence (RAP: *Equality Axiom Schema for Functions*)
    - $\{\neg E(x_1,y_1),\ldots,\neg E(x_n,y_n), E(f(x_1,\ldots,x_n), f(y_1,\ldots,y_n))\}$ for every *n*-ary *f* in ψ
    - $\{\neg E(x_1,y_1),\ldots,\neg E(x_n,y_n), \neg R(x_1,\ldots,x_n), R(y_1,\ldots,y_n)\}$ for every *n*-ary *R* in ψ
  - Clauses for *E* are positive Horn (see later slides)!

Equality Decision Procedure

- Consider a universal formula $\langle \forall x_1,\ldots,x_n\ \phi(x_1,\ldots,x_n)\rangle$ which does not contain any predicates, but can contain =, vars, functions, constants
- The formula is valid iff $\langle \exists x_1,\ldots,x_n\ \neg\phi(x_1,\ldots,x_n)\rangle$ is Unsat
- Iff $\neg\phi(c_1,\ldots,c_n)$ is Unsat, via Skolemization
- We can generate equivalent DNF: $\psi_1(c_1,\ldots,c_n) \vee \cdots \vee \psi_k(c_1,\ldots,c_n)$
- Which is Unsat iff $\psi_i(c_1,\ldots,c_n)$ is Unsat for all i (there are no vars)
- Note: $\psi_i(c_1,\ldots,c_n)$ is of the form $s_1{=}t_1 \wedge \cdots \wedge s_l{=}t_l \wedge u_1{\neq}v_1 \wedge \cdots \wedge u_m{\neq}v_m$
- Which is Unsat iff $s_1{=}t_1 \wedge \cdots \wedge s_l{=}t_l \Rightarrow u_1{=}v_1 \vee \cdots \vee u_m{=}v_m$ is Valid
- Iff for some j, $s_1{=}t_1 \wedge \cdots \wedge s_l{=}t_l \Rightarrow u_j{=}v_j$ is Valid
- So, we can reduce validity of FO formulas with no predicates to validity of equational logic with ground terms:
  - $\Phi \models s{=}t$ where $s{=}t$ and all elements of Φ are ground equations
  - By Birkhoff's theorem, equivalent to $\Phi \vdash \phi$ where we only use AXIOM, equivalence and congruence rules (see book)

Reduction to Propositional Logic

- Ackermann's idea: reduce the problem to propositional logic
- Consider: $f(f(f(c)))=c \wedge f(f(c))=c \Rightarrow f(c) = c$ (Valid or not?)
- Remove functions: Introduce variables for subterms, say $x_k=f^k(c)$ for $0 \leq k \leq 3$ and add constraints for congruence properties over subterms
  - $x_3=x_0 \wedge x_2=x_0 \wedge (x_0=x_1 \Rightarrow x_1=x_2) \wedge (x_0=x_2 \Rightarrow x_1=x_3) \wedge (x_1=x_2 \Rightarrow x_2=x_3)$
  - Check if this implies $x_1=x_0$ MP
- Remove =: replace equations, say $s=t$, with propositional atoms, say $P_{s,t}$, and add constraints for equivalence properties ($P_{s,t} \wedge P_{t,u} \Rightarrow P_{s,u}$)
- Now, we can use a propositional SAT solver
- Note: this problem is decidable

Ackermann Example

- Consider: $f(f(f(c)))=c \land f(f(c))=c \;\Rightarrow\; f(c) = c$
- Remove functions: Introduce variables for subterms, say
  - $x_k=f^k(c)$ for $0\leq k\leq 3$, so: $x_0=c$, $x_1=f(c)$, $x_2=f(f(c))$, $x_3=f(f(f(c)))$
- Rewrite problem: $x_3=x_0 \land x_2=x_0 \;\Rightarrow\; x_1=x_0$
- Add hyps: constraints for congruence properties over subterms
  - $(x_0=x_1 \Rightarrow x_1=x_2) \land (x_0=x_2 \Rightarrow x_1=x_3) \land (x_1=x_2 \Rightarrow x_2=x_3)$
  - Note $(x_0=x_3 \Rightarrow x_1=x_4)$, etc not needed since x_4 is not a subterm
- Remove =: replace equations with propositional atoms
  - $P_{3,0} \land P_{2,0} \land (P_{0,1} \Rightarrow P_{1,2}) \land (P_{0,2} \Rightarrow P_{1,3}) \land (P_{1,2} \Rightarrow P_{2,3}) \Rightarrow P_{1,0}$
- Add equivalence properties (as hyps) *Finish the reduction*

Optimizations?

  - $P_{0,0} \land P_{1,1} \land P_{2,2} \land P_{3,3} \land$
  - $(P_{0,1}\equiv P_{1,0})\land(P_{0,2}\equiv P_{2,0})\land(P_{0,3}\equiv P_{3,0})\land(P_{1,2}\equiv P_{2,1})\land(P_{1,3}\equiv P_{3,1})\land(P_{2,3}\equiv P_{3,2})\land$
  - $(P_{1,0}\land P_{0,2}\Rightarrow P_{1,2})\land(P_{1,0}\land P_{0,3}\Rightarrow P_{1,3})\land(P_{2,0}\land P_{0,3}\Rightarrow P_{2,3})\land(P_{0,1}\land P_{1,2}\Rightarrow P_{0,2})\land$ $(P_{0,1}\land P_{1,3}\Rightarrow P_{0,3})\land(P_{2,1}\land P_{1,3}\Rightarrow P_{2,3})\land(P_{0,2}\land P_{2,1}\Rightarrow P_{0,1})\land(P_{0,2}\land P_{2,3}\Rightarrow P_{0,3})\land$ $(P_{1,2}\land P_{2,3}\Rightarrow P_{1,3})\land(P_{0,3}\land P_{3,1}\Rightarrow P_{0,1})\land(P_{0,3}\land P_{3,2}\Rightarrow P_{0,2})\land(P_{1,3}\land P_{3,2}\Rightarrow P_{1,2})$

Congruence Closure

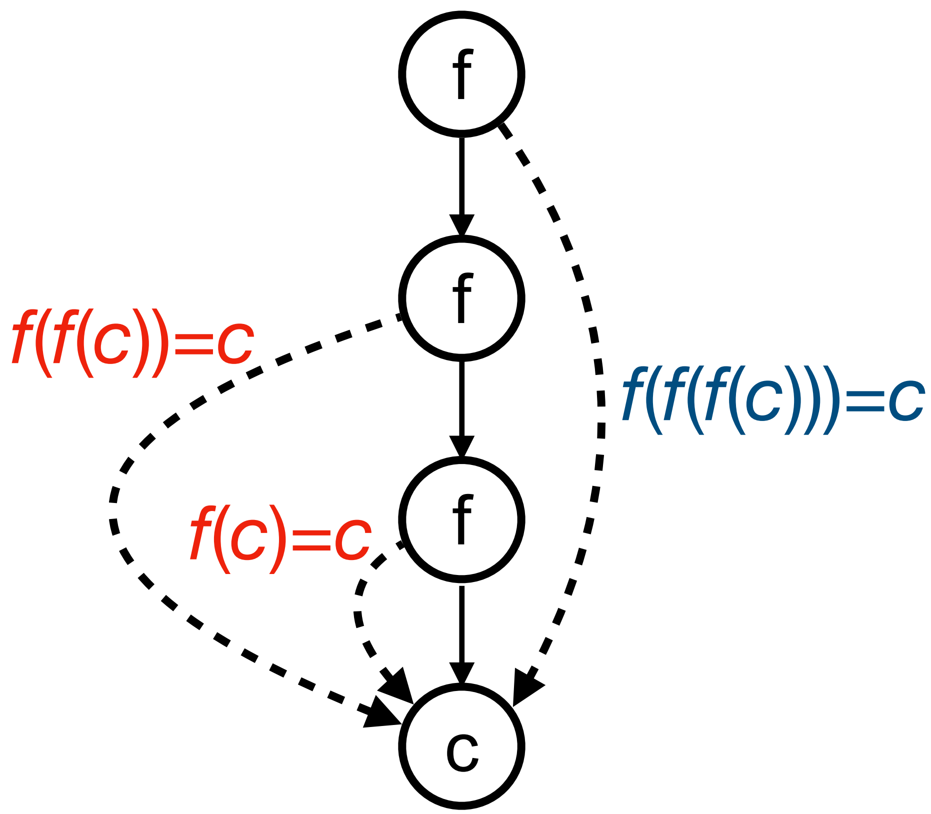
- Decision procedure for $\Phi \models s{=}t$ where $s{=}t$ and all elements of Φ are ground equations
- Let G be a set of terms closed under subterms
  - If $t \in G$ and s is a subterm of t, then $s \in G$
- $\sim$ is a congruence on G: an equivalence, congruence on terms in G
- For $R \subseteq G \times G$, the *congruence closure* of R on G is the smallest congruence on G extending R
  - Start with R and apply equivalence, congruence rules until fixpoint
- Let $\Phi = \{s_1{=}t_1, \ldots, s_n{=}t_n\}$, G is the minimal set closed under subterms of $\{s_1, t_1, \ldots, s_n, t_n, s, t\}$, $\sim$ the congruence closure of Φ on G. Then:
  - $\Phi \models s{=}t$ iff $s \sim t$
  - Can do this in P-time

Congruence Closure Algorithm

- Decision procedure for $\Phi \models s{=}t$ where $s{=}t$ and all elements of Φ are ground equations
- Main idea: use a graph with structure sharing to represent terms
- Start with ~ being the identity
- Each node (term) is mapped to its equivalence class
- For each assumption, $s_i{=}t_i$,
  - merge equivalence classes $[s_i]$, $[t_i]$
  - propagate congruences efficiently (using predecessor pointers)
- Check is $[s] = [t]$ after processing all hypotheses
- $O(m^2)$ algorithm due to Nelson, Oppen (m is the # edges in graph) in book
- $O(m\ log(m)^2)$ algorithm due to Downey, Sethi & Tarjan

Congruence Closure Example

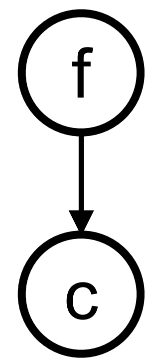
Consider: $f(f(f(c)))=c \wedge f(f(c))=c \;\Rightarrow\; f(c) = c$



$[f(f(c))]=[c]$, so $[f(f(f(c)))]=[f(c)]$ ie,

$[c]=[f(c)]$

congruence propagation



corresponds to $f(c)$

Graph representation allows structure sharing

corresponds to c

equivalence class of term

So, when we extend the congruence, by *union*ing [*s*] [*t*], we also have to union any terms of the form f(…s…) and f(…t…) if the rest of the arguments are in same class

Decidable Fragments of FOL

- Propositional Satisfiability: DP, DPLL, etc.
- LP, ILP, MILP: In NP, simplex, interior point methods, cutting planes, etc.
- $\langle \forall x_1,\ldots,x_n\ \phi(x_1,\ldots,x_n)\rangle$, with =, vars, functions, constants, no predicates
  - Uninterpreted functions: congruence closure
- How do we combine decision procedures? SMT/IMT (more later)
- Tools/language support
  - ACL2s/Z3 interface: query a solver on supported decidable fragments
  - Z3, other solvers, provide interfaces for various languages
  - A powerful, new programming paradigm: arbitrary interleaving of computation, constraint solving
  - Widely used, lots of potential applications