Lecture 21

Pete Manolios
Northeastern

Unification Basics

- Unification Problem: 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'll write $s_i\sigma = t_i\sigma$)
- $U(S)$ is the set of all unifiers of S; notice that if σ is a unifier, so is $\tau \circ \sigma$
- σ is *more general* than τ, $\sigma \leq \tau$, iff $\tau = \delta\sigma$ ($\delta \circ \sigma$) for some substitution δ
- $\leq$ is a preorder; let δ be the identify for reflexivity
  - transitivity: if $\sigma \leq \tau$, $\tau \leq \theta$ then $\tau = \delta\sigma$, $\theta = \gamma\tau = \gamma(\delta\sigma) = (\gamma\delta)\sigma$
  - $\sigma \sim \tau$ iff $\sigma \leq \tau$, $\tau \leq \sigma$. Notice that if $\sigma = x \leftarrow y$, $\tau = y \leftarrow x$, then $\sigma \sim \tau$
  - $\sigma \sim \tau$ iff there is a *renaming* (bijection on Vars) θ s.t. $\sigma = \theta\tau$
- A *most general unifier* (mgu) is $\sigma \in U(S)$ s.t. for all $\tau \in U(S)$, $\sigma \leq \tau$
  - What is an mgu for $x=y$? $x \leftarrow y$? $y \leftarrow x$? $z \leftarrow x, z \leftarrow y$? $y \leftarrow x, w \leftarrow z, z \leftarrow w$?
- A substitution is *idempotent* if $\sigma\sigma = \sigma$ (rules out last case above)
  - σ is idempotent iff Domain(σ) is disjoint from Vars(Range(σ))
- If a unification problem has a solution, then it has an idempotent mgu
- We want an algorithm that finds an mgu, if a unifier exists

Unification Algorithm

- $S = \{(x_1,t_1), \ldots, (x_n,t_n)\}$ is in solved form if the x_i are distinct variables and don't occur in any of the t_i. Then $S{\downarrow} = \{t_1 \leftarrow x_1, \ldots, t_n \leftarrow x_n\}$
- If S is in solved form and $\sigma \in U(S)$, then $\sigma = \sigma S{\downarrow}$ (σ, $\sigma S{\downarrow}$ agree on all vars)
- If S is in solved form, then $S{\downarrow}$ is an idempotent mgu
- Algorithm: *Nondeterministic transition system* based on the following rules
  - Delete $\{t=t\} \uplus S \Longrightarrow S$ useful way of thinking about algorithms: SMT/IMT
  - Decompose $\{f(t_1, \ldots, t_n) = f(s_1, \ldots, s_n)\} \uplus S \Longrightarrow \{t_1=s_1, \ldots, t_n=s_n\} \cup S$
  - Orient $\{t=x\} \uplus S \Longrightarrow \{x=t\} \cup S$, if t is not a variable
  - Eliminate $\{x=t\} \uplus S \Longrightarrow \{x=t\} \cup S|t \leftarrow x$, if $x \in \text{Vars}(S) - \text{Vars}(t)$
- Unify(S) = apply rules nondeterministically; if solved return $S{\downarrow}$, else fail
- Try it with: $\{(x, f(a)), (g(x,x), g(x,y))\}$

Unification Algorithm

- Algorithm: Nondeterministic transition system based on the following rules
  - Delete $\{t=t\} \uplus S \implies S$
  - Decompose $\{f(t_1, \ldots, t_n) = f(s_1, \ldots, s_n)\} \uplus S \implies \{t_1=s_1, \ldots, t_n=s_n\} \cup S$
  - Orient $\{t=x\} \uplus S \implies \{x=t\} \cup S$, if t is not a variable
  - Eliminate $\{x=t\} \uplus S \implies \{x=t\} \cup S|t{\leftarrow}x$, if $x \in \text{Vars}(S) - \text{Vars}(t)$

$x=f(a), g(x,x)=g(x,y)$ $\implies$ decompose — what other rules can I use?

$x=f(a), x=x, x=y$ $\implies$ delete — can't use eliminate on $x=x$; why?

$x=f(a), x=y$ $\implies$ eliminate x — can't use orient on $x=y$; why? can eliminate using $x=f(a)$

$y=f(a), x=y$ $\implies$ eliminate y

$y=f(a), x=f(a)$ $\implies$ return $S\downarrow$

- Try it with: $\{(x, f(y)), (y, g(x))\}$
- Try it with: $\{(P(f(w), f(y)), P(x, f(g(u))), (P(x,u), P(v,g(v))\}$
- Try it with: $\{(f(a,b,g(x,x),g(y,y),z), f(g(v,v),g(a,a),y,z,b))\}$

Unification Algorithm Termination

- Algorithm: Nondeterministic transition system based on the following rules
  - Delete $\{t=t\} \uplus S \implies S$
  - Decompose $\{f(t_1, \ldots, t_n) = f(s_1, \ldots, s_n)\} \uplus S \implies \{t_1=s_1, \ldots, t_n=s_n\} \cup S$
  - Orient $\{t=x\} \uplus S \implies \{x=t\} \cup S$, if t is not a variable
  - Eliminate $\{x=t\} \uplus S \implies \{x=t\} \cup S|t \leftarrow x$, if $x \in \mathrm{Vars}(S) - \mathrm{Vars}(t)$
- Termination: our measure function will be on ordinals (infinite numbers)
  - 0,1, 2, …, ω the first infinite ordinal (why stop with the naturals?)
  - Keep going: $\omega+1,\ \omega+2,\ \ldots,\ \omega+\omega = \omega 2,\ \omega 2+1,\ \ldots,\ \omega 3,\ \ldots,\ \omega\omega = \omega^2,$ $\ldots,\ \omega^3,\ \ldots,\ \omega^\omega,\ \ldots, \omega^{\omega^{\omega^{\cdots}}} = \epsilon_0$ ACL2s measures can use ordinals
  - Lexicographic ordering on tuples of natural numbers is $\approx \omega^\omega$
    - $\langle x_0, \ldots, x_{n-1}, x_n \rangle \longmapsto \omega^n x_0 + \cdots + \omega x_{n-1} + x_n$
    - There is an order-preserving bijection from n+1-tuples of Nats to ω^n
    - There is a theorem of this in the ACL2 ordinals books; you can define a relation, prove it is well-founded and use it in termination proofs

Unification Algorithm Termination

- Algorithm: Nondeterministic transition system based on the following rules
  - Delete $\{t=t\} \uplus S \Longrightarrow S$
  - Decompose $\{f(t_1, \ldots, t_n) = f(s_1, \ldots, s_n)\} \uplus S \Longrightarrow \{t_1=s_1, \ldots, t_n=s_n\} \cup S$
  - Orient $\{t=x\} \uplus S \Longrightarrow \{x=t\} \cup S$, if t is not a variable
  - Eliminate $\{x=t\} \uplus S \Longrightarrow \{x=t\} \cup S|t\leftarrow x$, if $x \in \text{Vars}(S) - \text{Vars}(t)$
- Termination: our measure function will be on ordinals (infinite numbers)
  - x is solved in S iff $x=t \in S$ and x only appears once in S
  - Measure: $\langle$vars in S not solved, size of S, # of equations $t=x$ in $S\rangle$

	vars in S not solved	size of S	# of equations $t=x$ in S
Delete	$\leq$ why not =?	$<$	Maybe $x \in t$, $x \notin S$
Decompose	$\leq$	$<$	
Orient	$\leq$	$=$	$<$
Eliminate	$<$		

for every rule we have $(\leq \mid =)^*<$, so the lexicographic order is decreasing (and well-founded), i.e., any algorithm based on these rules terminates