Un-interpreted function symbols

- Checking satisfiability, interpretation for
  - variables
  - function and predicate symbols
- Conjunction of literals in quantifier-free FOL
  - SAT solver handles disjunctions
- Flatten the literals, so they become one of:

$x=y$

$x \neq y$

$x=f(y_1,\ldots,y_n)$

where x,y, y_i are free variables.

Check satisfiability of

Conjunction of $x{=}y$, $x \neq y$, $x{=}f(y_1,\ldots,y_n)$

relevant terms T are variables and $f(y_1,\ldots,y_n)$.

$|T|$ is finite (at most 2n for n conjuncts)

Let $r_0 = \{(t_1,t_2) \mid (t_1{=}t_2)$ is an input conjunct$\}$

Closure: add elements to satisfy

- reflexivity, symmetry, transitivity (preserves that $r_0 \subseteq T^2$)
- congruence on elements of T: if arguments are related, so should the result

- Iterate until fixpoint (monotonic operator)

=> least congruence $\subseteq T^2$ containing r_0 : $CC(r_0)$

Satisfiability for Function Symbols

- If $x \neq y$ appears but (x,y) in $CC(r_0)$, then we have contradiction, since $(x=y)$ is a consequence of equalities. Assume not.
- Let the domain be the equivalence classes of $CC(r_0)$, union any extra disjoint set
- Define interpretation of variable x to be [x]
- If $f(x_1,\ldots,x_n)$ is in T (in the formula), define $f([x_1],\ldots, [x_n]) = [f(x_1,\ldots,x_n)]$
- otherwise, define it arbitrarily
- This interpretation satisfies the formula

Term algebras

- Checking satisfiability, find interpretation for variables, which range over ground terms

Node(Node(x,y),z) = Node(u,Node(p,q))

holds iff

  Node(x,y) = u

  z = Node(p,q)

Technique: unification

Satisfiability for term algebras

- Conjunction of $x=y$, $x \neq y$, $x=f(y_1,\ldots,y_n)$ where this time f is a constructor (with 0 or more args)
- Bidirectional closure $f(x)=f(y)$ iff $x=y$
- Let T be the terms appearing in input
- Find most-general unifier U for positive ones
- Check whether for each $x \neq y$, we have $U(x)=U(y)$ (identical substitution). If yes, contradiction
- Otherwise, we can pick the variables so that disequality does hold, by ***Independence of disequations lemma*** *(see* **Lemma 2 in [Decision Procedures for Algebraic Data Types with Abstractions](#) , POPL'10)**