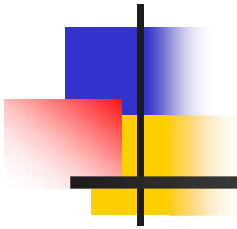


# Programming Languages and Compilers (CS 421)



Reza Zamani

<http://www.cs.illinois.edu/class/cs421/>

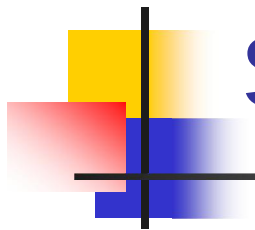
Based in part on slides by Mattox Beckman, as updated by Vikram Adve and Gul Agha



# Background for Unification

---

- n **Terms** made from **constructors** and **variables** (for the simple first order case)
- n Constructors may be applied to arguments (other terms) to make new terms
- n Variables and constructors with no arguments are base cases
- n Constructors applied to different number of arguments (arity) considered different
- n **Substitution** of terms for variables



# Simple Implementation Background

---

```
type term = Variable of string  
          | Const of (string * term list)
```

```
let rec subst var_name residue term =  
  match term with Variable name ->  
    if var_name = name then residue else term  
  | Const (c, tys) ->  
    Const (c, List.map (subst var_name residue)  
                  tys);;
```



# Unification Problem

---

Given a set of pairs of terms ("equations")

$$\{(s_1, t_1), (s_2, t_2), \dots, (s_n, t_n)\}$$

(the *unification problem*) does there exist  
a substitution  $\sigma$  (the *unification solution*)  
of terms for variables such that

$$\sigma(s_i) = \sigma(t_i),$$

for all  $i = 1, \dots, n$ ?



## Uses for Unification

---

- n Type Inference and type checking
- n Pattern matching as in OCAML
  - n Can use a simplified version of algorithm
- n Logic Programming - Prolog
- n Simple parsing



# Unification Algorithm

---

- n Let  $S = \{(s_1, t_1), (s_2, t_2), \dots, (s_n, t_n)\}$  be a unification problem.
- n Case  $S = \{ \}$ :  $\text{Unif}(S) = \text{Identity function}$  (ie no substitution)
- n Case  $S = \{(s, t)\} \cup S'$  : Four main steps



# Unification Algorithm

---

- n **Delete:** if  $s = t$  (they are the same term)  
then  $\text{Unif}(S) = \text{Unif}(S')$
- n **Decompose:** if  $s = f(q_1, \dots, q_m)$  and  $t = f(r_1, \dots, r_m)$  (same  $f$ , same  $m$ !), then  
 $\text{Unif}(S) = \text{Unif}(\{(q_1, r_1), \dots, (q_m, r_m)\} \cup S')$
- n **Orient:** if  $t = x$  is a variable, and  $s$  is not a variable,  
 $\text{Unif}(S) = \text{Unif}(\{(x, s)\} \cup S')$

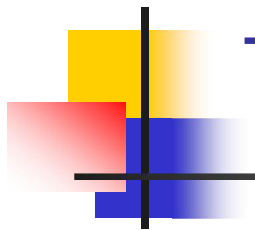


# Unification Algorithm

---

- n **Eliminate:** if  $s = x$  is a variable, and  $x$  does not occur in  $t$  (the occurs check), then
  - n Let  $\phi = x \mapsto t$
  - n Let  $\psi = \text{Unif}(\phi(S'))$
  - n  $\text{Unif}(S) = \{x \mapsto \psi(t)\} \circ \psi$ 
    - n Note:  $\{x \mapsto a\} \circ \{y \mapsto b\} = \{y \mapsto \{x \mapsto a\}(b)\} \circ \{x \mapsto a\}$  if  $y$  not in  $a$





## Tricks for Efficient Unification

---

- n Don't return substitution, rather do it incrementally
- n Make substitution be constant time
  - n Requires implementation of terms to use mutable structures (or possibly lazy structures)
  - n We haven't discussed these yet



# Example

---

n  $x, y, z$  variables,  $f, g$  constructors

n  $S = \{(f(x), f(g(y, z))), (g(y, f(y)), x)\}$



## Example

---

n  $x, y, z$  variables,  $f, g$  constructors

n Pick a pair:  $(g(y, f(y)), x)$

n  $S = \{(f(x), f(g(y, z))), (g(y, f(y)), x)\}$



## Example

---

- n  $x, y, z$  variables,  $f, g$  constructors
- n Pick a pair:  $(g(y, f(y))), x)$
- n Orient is first rule that applies
- n  $S = \{(f(x), f(g(y, z))), (g(y, f(y)), x)\}$



# Example

---

n  $x, y, z$  variables,  $f, g$  constructors

n  $S \rightarrow \{(f(x), f(g(y, z))), (x, g(y, f(y)))\}$



## Example

---

- n  $x, y, z$  variables,  $f, g$  constructors
- n Pick a pair:  $(f(x), f(g(y, z)))$
- n  $S \rightarrow \{(f(x), f(g(y, z))), (x, g(y, f(y)))\}$



## Example

---

- n  $x, y, z$  variables,  $f, g$  constructors
- n Pick a pair:  $(f(x), f(g(y, z)))$
- n Decompose it:  $(x, g(y, z))$
- n  $S \rightarrow \{(x, g(y, z)), (x, g(y, f(y)))\}$



## Example

---

- n  $x, y, z$  variables,  $f, g$  constructors
- n Pick a pair:  $(x, g(y, f(y)))$
- n  $S \rightarrow \{(x, g(y, z)), (x, g(y, f(y)))\}$





## Example

---

- n  $x, y, z$  variables,  $f, g$  constructors
  - n Pick a pair:  $(x, g(y, f(y)))$
  - n Substitute:
  - n  $S \rightarrow \{(g(y, f(y)), g(y, z))\}$
- With  $\{x \mapsto g(y, f(y))\}$



## Example

---

n  $x, y, z$  variables,  $f, g$  constructors

n Pick a pair:  $(g(y, f(y)), g(y, z))$

n  $S \rightarrow \{(g(y, f(y)), g(y, z))\}$

With  $\{x \mapsto g(y, f(y))\}$



## Example

---

- n  $x, y, z$  variables,  $f, g$  constructors
  - n Pick a pair:  $(g(y, f(y)), g(y, z))$
  - n Decompose:  $(y, y)$  and  $(f(y), z)$
  - n  $S \rightarrow \{(y, y), (f(y), z)\}$
- With  $\{x \mapsto g(y, f(y))\}$



## Example

---

n  $x, y, z$  variables,  $f, g$  constructors

n Pick a pair:  $(y, y)$

n  $S \rightarrow \{(y, y), (f(y), z)\}$

With  $\{x \mapsto g(y, f(y))\}$



## Example

---

- n  $x, y, z$  variables,  $f, g$  constructors
  - n Pick a pair:  $(y, y)$
  - n Delete
  - n  $S \rightarrow \{(f(y), z)\}$
- With  $\{x \mapsto g(y, f(y))\}$



## Example

---

n  $x, y, z$  variables,  $f, g$  constructors

n Pick a pair:  $(f(y), z)$

n  $S \rightarrow \{(f(y), z)\}$

With  $\{x \mapsto g(y, f(y))\}$



## Example

---

- n  $x, y, z$  variables,  $f, g$  constructors
  - n Pick a pair:  $(f(y), z)$
  - n Orient
  - n  $S \rightarrow \{(z, f(y))\}$
- With  $\{x \mapsto g(y, f(y))\}$



## Example

---

n  $x, y, z$  variables,  $f, g$  constructors

n Pick a pair:  $(z, f(y))$

n  $S \rightarrow \{(z, f(y))\}$

With  $\{x \mapsto g(y, f(y))\}$





## Example

---

n  $x, y, z$  variables,  $f, g$  constructors

n Pick a pair:  $(z, f(y))$

n Substitute

n  $S \rightarrow \{ \}$

With  $\{x \mapsto \{z \mapsto f(y)\} (g(y, f(y))) \} \circ \{z \mapsto f(y)\}$



## Example

---

n  $x, y, z$  variables,  $f, g$  constructors

n Pick a pair:  $(z, f(y))$

n Substitute

n  $S \rightarrow \{ \}$

With  $\{x \mapsto g(y, f(y))\} \circ \{(z \mapsto f(y))\}$



## Example

---

$$S = \{(f(x), f(g(y,z))), (g(y,f(y)), x)\}$$

Solved by  $\{x \mapsto g(y,f(y))\} \circ \{(z \mapsto f(y))\}$

$$\underbrace{f(g(y,f(y)))}_x = f(g(y,\underbrace{f(y)})_z)$$

and

$$g(y,f(y)) = \underbrace{g(y,f(y))}_x$$



## Example of Failure

---

- n  $S = \{(f(x, g(y)), f(h(y), x))\}$
- n Decompose
- n  $S \rightarrow \{(x, h(y)), (g(y), x)\}$
- n Orient
- n  $S \rightarrow \{(x, h(y)), (x, g(y))\}$
- n Substitute
- n  $S \rightarrow \{(h(y), g(y))\}$  with  $\{x \mapsto h(y)\}$
- n No rule to apply! Decompose fails!