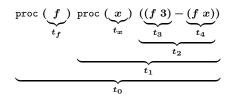
## COSE212: Programming Languages

Lecture 15 — Automatic Type Inference (3)

Hakjoo Oh 2018 Fall

## Finding a Solution of Type Equations

Find the values of type variables that make all the equations true.



			Equations			Solution
_	$t_0$	=	$t_f  ightarrow t_1$	$t_0$	=	$(int \to int) \to (int \to int)$
	$t_1$	=	$t_x  ightarrow t_2$	$t_1$	=	int  o int
	$t_3$	=	int	$t_2$	=	int
	$t_4$	=	int	$t_3$	=	int
	$t_2$	=	int		=	
	$t_f$	=	$int \to t_3$	$ t_f $	=	int  o int
	$t_f$	=	$t_x  ightarrow t_4$		=	

Static type systems find such a solution using unification algorithm.

The calculation is split into equations to be solved and substitution found so far. Initially, the substitution is empty:

Equations	Substitution
$t_0 = t_f  ightarrow t_1$	
$t_1 \; = \; t_x  ightarrow t_2$	
$t_3 = int$	
$t_4 = int$	
$t_2 = int$	
$t_f$ $=$ int $ o t_3$	
$t_f^- = t_x  ightarrow t_4$	

Consider each equation in turn. If the equation's left-hand side is a variable, move it to the substitution:

Equations	Substitution
$t_1 = t_x  ightarrow t_2$	$t_0 = t_f  ightarrow t_1$
$t_3 \; = \; int$	
$t_4 = int$	
$t_2 = int$	
$t_f \; = \; int  o t_3$	
$t_f = t_x  ightarrow t_4$	

Move the next equation to the substitution and propagate the information to the existing substitution (i.e., substitute the right-hand side for each occurrence of  $t_1$ ):

Equations	Substitution
$t_3$ = int	$t_0 = t_f  ightarrow (t_x  ightarrow t_2)$
$t_4 \; = \; int$	$t_1 = t_x  ightarrow t_2$
$t_2 \; = \; int$	
$t_f \; = \; int  o t_3$	
$t_f = t_x  ightarrow t_4$	

Same for the next three equations:

Equations	Substitution
$t_4 \; = \; int$	$\mid t_0 \mid = \mid t_f \rightarrow (t_x \rightarrow t_2)$
$t_{2}$ = int	$egin{array}{cccc} t_1 &=& t_x  ightarrow t_2 \end{array}$
$t_f$ $=$ int $ o t_3$	$t_3$ = int
$t_f^{'} = t_x  ightarrow t_4$	
Equations	Substitution
$t_2$ = int	$t_0 = t_f  ightarrow (t_x  ightarrow t_2)$
$t_f$ $=$ int $ o t_3$	$t_1 = t_x  ightarrow t_2$
$t_f^{-}=t_x ightarrow t_4$	$t_3$ = int
·	$ t_4 = {int}$
Equations	Substitution
$t_f = {\sf int}  o t_3$	$t_0 = t_f  ightarrow (t_x  ightarrow { m int})$
$t_f = t_x  ightarrow t_4$	$t_1 = t_x  ightarrow int$
	$\mid t_3 \mid = \mid$ int
	$ig  t_4 = int$
	$t_2 = int$

Consider the next equation  $t_f={\rm int} \to t_3$ . The equation contains  $t_3$ , which is already bound to int in the substitution. Substitute int for  $t_3$  in the equation. This is called *applying* the substitution to the equation.

Equations	Substitution
$t_f = int  o int$	$t_0 = t_f  ightarrow (t_x  ightarrow int)$
$t_f = t_x  ightarrow t_4$	$egin{array}{cccc} t_1 &=& t_x  ightarrow { m int} \end{array}$
	$t_3$ = int
	$t_4$ = int
	$egin{array}{lll} t_0 &=& t_f  ightarrow (t_x  ightarrow { m int}) \ t_1 &=& t_x  ightarrow { m int} \ t_3 &=& { m int} \ t_4 &=& { m int} \ t_2 &=& { m int} \end{array}$

Move the resulting equation to the substitution and update it.

Equations			Substitution
$t_f = t_x  o t_4$	$t_0$	=	$(int  o int)  o (t_x  o int)$
	$\mid t_1 \mid$	=	$t_{m{x}}  ightarrow int$
	$ t_3 $	=	int
	$ t_4 $	=	int
	$ t_2 $	=	int
	$\mid t_f \mid$	=	$(\operatorname{int}  o \operatorname{int})  o (t_x  o \operatorname{int})$ $t_x  o \operatorname{int}$

Apply the substitution to the equation:

Equations	Substitution
$int  o int \ = \ t_x  o int$	$t_0 = (int \to int) \to (t_x \to int)$
	$t_1 \;\; = \;\; t_x  o int$
	$t_3$ = int
	$t_4$ = int
	$t_2$ = int
	$egin{array}{lll} t_0 &=& (\operatorname{int}  ightarrow \operatorname{int})  ightarrow (t_x  ightarrow \operatorname{int}) \ t_1 &=& t_x  ightarrow \operatorname{int} \ t_3 &=& \operatorname{int} \ t_4 &=& \operatorname{int} \ t_2 &=& \operatorname{int} \ t_f &=& \operatorname{int}  ightarrow \operatorname{int}  i$

If neither side of the equation is a variable, simplify the equation by yielding two new equations:

Equations	Substitution
$int \; = \; t_x$	$t_0 = (int \to int) \to (t_x \to int)$
int = int	$t_1 = t_x  ightarrow {int}$
	$t_3$ = int
	$t_4 = int$
	$t_2 = int$
	$egin{array}{lll} t_0&=&(\operatorname{int} ightarrow\operatorname{int}) ightarrow (t_x ightarrow\operatorname{int}) \ t_1&=&t_x ightarrow\operatorname{int} \ t_3&=&\operatorname{int} \ t_4&=&\operatorname{int} \ t_2&=&\operatorname{int} \ t_f&=&\operatorname{int} ightarrow\operatorname{int}  ightarrow \operatorname{int}  ightarrow \operatorname{int} \ \end{array}$

Switch the sides of the first equation and move it to the substitution:

Equations		Substitution	
int = int	$t_0$ =	$(\text{int} \rightarrow \text{int}) \rightarrow (\text{int}$	$\rightarrow$ int)
	$t_1$ =	$int \to int$	
	$t_3$ =	int	
	$t_4$ =	int	
	$t_2$ =	int	
	$t_f$ =	$int \to int$	
	$t_x$ =	int → int int int int int int int int int	

The final substitution is the solution of the original equations.

$$t_0 = t_f 
ightarrow t_1 \ t_0 = t_f 
ightarrow t_1 \ t_f = \operatorname{int} 
ightarrow t_1$$

1

	Equations	Substitution
$t_0$	$= t_f  ightarrow t_1$	
$t_f$	$=$ int $ ightarrow t_1$	

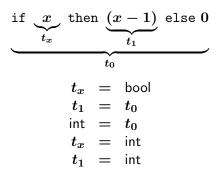
2

$$egin{array}{ccccc} {\sf Equations} & {\sf Substitution} \ \hline t_f &=& {\sf int} 
ightarrow t_1 & t_0 &=& t_f 
ightarrow t_1 \ \hline \end{array}$$

3

<b>Equations</b>	Substitution
	$t_0 = (int  o t_1)  o t_1$
	$t_f = int  o t_1$

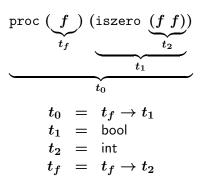
The type is polymorphic in  $t_1$ .



The equations have no solutions because, during the unification algorithm, we encounter the following contradictory state:

Equations			Substitution			
bool	=	int		$t_x$		bool
$t_1$	=	int		$t_1$	=	int
				$t_0$	=	int

Because bool and int cannot be equal, there is no solution to the equations.



Solving as usual, we encounter a problem:

Equations	Substitution			
$t_f = t_f  o  ext{int}$	$egin{array}{lll} t_0 &=& t_f  ightarrow { m bool} \ t_1 &=& { m bool} \ t_2 &=& { m int} \end{array}$			

- There is no type  $t_f$  that satisfies the equation, because the right-hand side of the equation is always larger than the left.
- If we ever deduce an equation of the form  $t=\dots t\dots$  where the type variable t occurs in the right-hand side, we must conclude that there is no solution. This is called *occurrence check*.

## Unification Algorithm

For each equation in turn,

- Apply the current substitution to the equation.
- If the equation is always true (e.g. int = int), discard it.
- If the left- and right-hand sides are contradictory (e.g. bool = int), the algorithm fails.
- If neither side is a variable (e.g. int  $\rightarrow t_1 = t_2 \rightarrow$  bool), simplify the equation, which eventually generates an equation whose left- or right-hand side is a variable.
- If the left-hand side is not a variable, switch the sides.
- If the left-hand side variable occurs in the right-hand side, the algorithm fails.
- Otherwise, move it to the substitution and substitute the right-hand side for each occurrence of the variable in the substitution.

$$let x = 4 in (x 3)$$

let 
$$f = \text{proc}(z) z \text{ in proc}(x) ((f x) - 1)$$

let p = iszero 1 in if p then 88 else 99

let f = proc(x) x in if (f (iszero0)) then (f 11) else (f 22)

#### Substitution

Solutions of type equations are represented by substitution:

$$S \in Subst = \mathit{TyVar} o T$$

Applying a substitution to a type:

$$S(\mathsf{int}) = \mathsf{int}$$
 $S(\mathsf{bool}) = \mathsf{bool}$ 
 $S(lpha) = egin{cases} t & \mathsf{if} \ lpha \mapsto t \in S \ lpha & \mathsf{otherwise} \end{cases}$ 
 $S(T_1 o T_2) = S(T_1) o S(T_2)$ 

Applying the substitution

$$S = \{t_1 \mapsto \mathsf{int}, t_2 \mapsto \mathsf{int} \to \mathsf{int}\}$$
 to to the type  $(t_1 \to t_2) \to (t_3 \to \mathsf{int})$ :  $S((t_1 \to t_2) \to (t_3 \to \mathsf{int}))$   $= S(t_1 \to t_2) \to S(t_3 \to \mathsf{int})$   $= (S(t_1) \to S(t_2)) \to (S(t_3) \to S(\mathsf{int}))$   $= (\mathsf{int} \to (\mathsf{int} \to \mathsf{int})) \to (t_3 \to \mathsf{int})$ 

#### Unification

Update the current substitution with equality  $t_1 \doteq t_2$ .

$$\mathsf{unify}: T \times T \times Subst \to Subst$$

$$\begin{array}{rcl} & \mathsf{unify}(\mathsf{int},\mathsf{int},S) & = & S \\ & \mathsf{unify}(\mathsf{bool},\mathsf{bool},S) & = & S \\ & \mathsf{unify}(\alpha,\alpha,S) & = & S \\ & \mathsf{unify}(\alpha,t,S) & = & \left\{ \begin{array}{l} \mathsf{fail} & \alpha \; \mathsf{occurs} \; \mathsf{in} \; t \\ \mathsf{extend} \; S \; \mathsf{with} \; \alpha \doteq t \; \mathsf{otherwise} \end{array} \right. \\ & \mathsf{unify}(t,\alpha,S) & = \; \mathsf{unify}(\alpha,t,S) \\ & \mathsf{unify}(t_1 \to t_2,t_1' \to t_2',S) & = \; \mathsf{let} \; S' = \mathsf{unify}(t_1,t_1',S) \; \mathsf{in} \\ & \mathsf{let} \; S'' = \mathsf{unify}(S'(t_2),S'(t_2'),S') \; \mathsf{in} \\ & S'' \\ & \mathsf{unify}(\cdot,\cdot,\cdot) & = \; \mathsf{fail} \end{array}$$

- $\operatorname{unify}(\alpha, \operatorname{int} \to \operatorname{int}, \emptyset) =$
- unify( $\alpha$ , int  $\rightarrow \alpha$ ,  $\emptyset$ ) =
- unify( $\alpha \to \beta$ , int  $\to$  int,  $\emptyset$ ) =
- $\operatorname{unify}(\alpha \to \beta, \operatorname{int} \to \alpha, \emptyset) =$

# Solving Equations

$$\begin{array}{rcl} \text{unifyall}: TyEqn \rightarrow Subst \rightarrow Subst \\ & \text{unifyall}(\emptyset,S) &=& S \\ \text{unifyall}((t_1 \doteq t_2) \ \land \ u,S) &=& \text{let } S' = \text{unify}(S(t_1),S(t_2),S) \\ & \text{in unifyall}(u,S') \end{array}$$

Let  $\mathcal{U}$  be the final unification algorithm:

$$\mathcal{U}(u) = \mathsf{unifyall}(u,\emptyset)$$

## $\mathsf{typeof}: E \to T$

The final type inference algorithm that composes equation derivation  $(\mathcal{V})$  and equation solving  $(\mathcal{U})$ :

$$\begin{array}{l} \operatorname{typeof}(E) = \\ \operatorname{let} S = \mathcal{U}(\mathcal{V}(\emptyset, E, \alpha)) \quad (\text{new } \alpha) \\ \operatorname{in} S(\alpha) \end{array}$$

- typeof((proc(x) x) 1)
- typeof(let x = 1 in proc(y) (x + y))

## Summary

#### Automatic type inference:

- derive type equations from the program text, and
- solve the equations by unification algorithm.