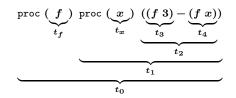
COSE212: Programming Languages

Lecture 12 — Automatic Type Inference (2)

Hakjoo Oh 2017 Fall

Finding a Solution of Type Equations

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



Equations	Solution
$t_0 = t_f \rightarrow t_1$	$t_0 = (int \to int) \to (int \to int)$
$t_1 = t_x \rightarrow t_2$	$t_1 = int o int$
$t_3 = int$	$t_2 = int$
$t_4 = int$	t_3 = int
t_{2} = int	$t_4 = int$
$t_f = int o t_3$	$t_f = \operatorname{int} o \operatorname{int}$
$t_f = t_x \rightarrow t_4$	t_x = int

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

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

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

Move the resulting equation to the substitution and update it.

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

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:

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

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

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

The final substitution is the solution of the original equations.

$$t_0 = t_f
ightarrow t_1 \ t_1 \ t_2 \ t_3 \ t_4 \ t_5 = t_1 \ t_5 \ t_1 \ t_5 \ t_1 \ t_5 \ t$$

1

Equations	Substitution
$t_0 = t_f ightarrow t_1$	
$t_f \; = \; int o t_1$	

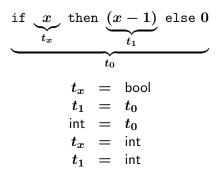
2

Equations	Substitution	
$t_f \; = \; int o t_1$	$t_0 = t_f \rightarrow t_1$	

(3)

Equations	Substitution			
	$t_0 = (int o t_1) o t_1$			
	$\mid t_f \mid = \; int o t_1$			

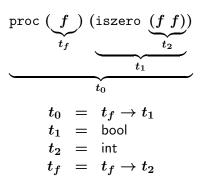
The type is *polymorphic* in t_1 .



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

Equations		Substitution				
bool	=	int				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 base types and 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)

Summary

Automatic type inference:

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