

CS 510 HW 10

Ex 7.13

Rules for let expressions:

(let-exp var exp body):

$t_{\text{var}} = t_{\text{exp}}$

$t_{\text{(let-exp var exp body)}} = t_{\text{body}}$

1: let x = 4 in (x 3)

Expression	Type variable
x	t_x
let x = 4 in (x 3)	t_0
(x 3)	t_1

Expression	Equations
let x = 4 in (x 3)	$t_0 = t_1$
x = 4	$t_x = \text{int}$
(x 3)	$t_x = \text{int} \rightarrow t_1$

Equations	Substitution
$t_0 = t_1$	
$t_x = \text{int}$	
$t_x = \text{int} \rightarrow t_1$	

Equations	Substitution
$\text{int} = \text{int} \rightarrow t_1$	$t_0 = t_1$
	$t_x = \text{int}$
	$t_x = \text{int} \rightarrow t_1$

Since $\text{int} \neq \text{int} \rightarrow t_1$, no such type exists for this let expression

2: let f = proc (z) z in proc (x) -((f x), 1)

Expression	Type variable
x	t_x
z	t_z
f	t_f
let f = proc (z) z in proc (x) -((f x), 1)	t_0
proc (x) -((f x), 1)	t_1
-((f x), 1)	t_2
(f x)	t_3
proc (z) z	t_4

Expression	Equations
let f = proc (z) z in proc (x) -((f x), 1)	$t_0 = t_1$
proc (x) -((f x), 1)	$t_1 = t_x \rightarrow t_2$
-((f x), 1)	$t_2 = \text{int}$
	$t_3 = \text{int}$

(f x)
 proc (z) z

$t_f = t_x \rightarrow t_3$
 $t_4 = t_z \rightarrow t_z$
 $t_f = t_4$
 $t_z = t_x$

Equations

$t_4 = t_z \rightarrow t_z$
 $t_f = t_4$
 $t_z = t_x$

Substitution

$t_0 = t_x \rightarrow \text{int}$
 $t_1 = t_x \rightarrow \text{int}$
 $t_2 = \text{int}$
 $t_3 = \text{int}$
 $t_f = t_x \rightarrow \text{int}$

Equations

$t_z = t_x$

Substitution

$t_0 = t_x \rightarrow \text{int}$
 $t_1 = t_x \rightarrow \text{int}$
 $t_2 = \text{int}$
 $t_3 = \text{int}$
 $t_f = t_x \rightarrow \text{int}$
 $t_f = t_4 = t_z \rightarrow t_z$

Equations

$t_x = \text{int}$

Substitution

$t_0 = t_x \rightarrow \text{int}$
 $t_1 = t_x \rightarrow \text{int}$
 $t_2 = \text{int}$
 $t_3 = \text{int}$
 $t_f = t_x \rightarrow \text{int}$
 $t_f = t_4 = t_x \rightarrow t_x$

Equations

Substitution

$t_0 = \text{int} \rightarrow \text{int}$
 $t_1 = \text{int} \rightarrow \text{int}$
 $t_2 = \text{int}$
 $t_3 = \text{int}$
 $t_f = \text{int} \rightarrow \text{int}$
 $t_x = \text{int}$

Thus the type of this let-exp should be $(\text{int} \rightarrow \text{int})$

3: let p = zero?(1) in if p then 88 else 99

Expression	Type variable
let p = zero?(1) in if p then 88 else 99	t_0
p	t_p
if p then 88 else 99	t_1

Expression	Equations
let p = zero?(1) in if p then 88 else 99	t_0 = t_1
p = zero?(1)	t_p = bool
if p then 88 else 99	t_1 = int

Equations	Substitution
t_0 = t_1	
t_p = bool	
t_1 = int	

Equations	Substitution
	t_0 = int
	t_p = bool

Thus the type of this let-exp is int

4: let p = proc (z) z in z in if p then 88 else 99

Expression	Type variable
let p = proc (z) z in z in if p then 88 else 99	t_0
z	t_z
p	t_p
if p then 88 else 99	t_1
proc (z) z	t_2

Expression	Equations
let p = proc (z) z in z in if p then 88 else 99	t_0 = t_1
if p then 88 else 99	t_1 = int; t_p = bool
p = proc (z) z	t_p = t_2; t_2 = t_z → t_z

Equations	Substitution
t_0 = t_1	
t_1 = int; t_p = bool	
t_p = t_2; t_2 = t_z → t_z	

Equations	Substitution
Bool = t_z → t_z	t_0 = int
	t_1 = int
	t_p = bool
	t_p = t_2
	t_p = t_z → t_z

$\text{bool} \neq t_z \rightarrow t_z$

Thus, there is no such type for this let expression