编译原理作业 HW10

PB20111686 黄瑞轩

5.15

 $S: lpha
ightarrow lpha, eta
ightarrow pointer(lpha), \gamma
ightarrow pointer(lpha), \delta
ightarrow pointer(lpha)$

如果 (b) 的 δ 是 α , 从已知往前推:

$$S: \gamma
ightarrow lpha, eta
ightarrow lpha, lpha
ightarrow pointer(lpha)$$

其中 $\alpha \to pointer(\alpha)$ 不成立,所以这种情况下没有最一般的合一代换。

5.17

为 f 和 l 分别引入类型变量 α, β 来表示其类型。

行	定型断言	代换	规则
1	f:lpha		(Exp Id)
2	l:eta		(Exp Id)
3	f,l: lpha imes eta		(Exp Pair)
4	$map:\delta$		(Exp Id)
5	$map(f,l):\phi$	$\delta = \gamma o \phi$	(Exp FunCall)
6	l:eta		从 (2) 可得
7	$null: list(lpha_n) ightarrow boolean$		(Exp Id Fresh)
8	null(l):boolean	$\beta = list(\alpha_n)$	(Exp FunCall)
9	$nil: list(lpha_m)$		(Exp ld Fresh)
10	$l: list(lpha_n)$		从 (8) 可得
11	$hd: list(lpha_t) ightarrow lpha_t$		(Exp Id Fresh)
12	$hd(l):lpha_n$	$lpha_t = lpha_n$	(Exp FunCall)
13	f:lpha		从 (1) 可得
14	$f(hd(l)):lpha_v$	$lpha=lpha_n ightarrowlpha_v$	(Exp FunCall)
15	$f:lpha_n olpha_v$		从 (14) 可得
16	$l: list(\alpha_n)$		从 (8) 可得
17	$tl: list(lpha_c) ightarrow list(lpha_c)$		(Exp Id Fresh)
18	$tl(l): list(lpha_n)$	$lpha_c=lpha_n$	(Exp FunCall)
19	$f,tl(l):(\alpha_n\to\alpha_v)\times list(\alpha_n)$		(Exp Pair)
20	$map: ((lpha_n ightarrow lpha_v) imes list(lpha_n)) ightarrow lpha_y$		从 (4) 可得
21	$map(f,tl(l)):lpha_y$		(Exp FunCall)
22	$f(hd(l)), map(f,tl(l)): \alpha_v \times \alpha_y$		(Exp Pair)
23	$cons:\alpha_r\times list(\alpha_r)\to list(\alpha_r)$		(Exp Id Fresh)
24	$cons(f(hd(l)), map(f, tl(l))): list(\alpha_v)$	$lpha_r = lpha_v lpha_y = list(lpha_v)$	(Exp Funcall)
25	$if:boolean imes lpha_i imes lpha_i ightarrow lpha_i$		(Exp Id Fresh)
26	$if(\dots):$	$lpha_i=lpha_v=lpha_m$	(Exp FunCall)
27	$match: (lpha_j imes lpha_j) ightarrow lpha_j$		(Exp Id Fresh)
28	$match(\dots): list(lpha_m)$	$lpha_j=lpha_m$	(Exp FunCall)

因为对类型 α_m, α_n 没有任何限制,让其称为约束变量,写成

map: orall lpha orall eta. ((lpha
ightarrow eta) imes list(lpha))
ightarrow list(eta)