编译原理作业 HW10

PB20111686 黄瑞轩

5.15

$S:\alpha\to\alpha,\beta\to pointer(\alpha),\gamma\to pointer(\alpha),\delta\to pointer(\alpha)$

如果 (b) 的 δ 是 α，从已知往前推：

$S:\gamma\to\alpha,\beta\to\alpha,\alpha\to pointer(\alpha)$

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	$f:\alpha$		(Exp Id)
2	$l:\beta$		(Exp Id)
3	$f,l:\alpha\times\beta$		(Exp Pair)
4	$map:\delta$		(Exp Id)
5	$map(f,l):\phi$	$\delta=\gamma\to\phi$	(Exp FunCall)
6	$l:\beta$		从（2）可得
7	$null:list(\alpha_n)\to boolean$		(Exp Id Fresh)
8	$null(l):boolean$	$\beta=list(\alpha_n)$	(Exp FunCall)
9	$nil:list(\alpha_m)$		(Exp Id Fresh)
10	$l:list(\alpha_n)$		从（8）可得
11	$hd:list(\alpha_t)\to\alpha_t$		(Exp Id Fresh)
12	$hd(l):\alpha_n$	$\alpha_t=\alpha_n$	(Exp FunCall)
13	$f:\alpha$		从（1）可得
14	$f(hd(l)):\alpha_v$	$\alpha=\alpha_n\to\alpha_v$	(Exp FunCall)
15	$f:\alpha_n\to\alpha_v$		从（14）可得
16	$l:list(\alpha_n)$		从（8）可得
17	$tl:list(\alpha_c)\to list(\alpha_c)$		(Exp Id Fresh)
18	$tl(l):list(\alpha_n)$	$\alpha_c=\alpha_n$	(Exp FunCall)
19	$f,tl(l):(\alpha_n\to\alpha_v)\times list(\alpha_n)$		(Exp Pair)
20	$map:((\alpha_n\to\alpha_v)\times list(\alpha_n))\to\alpha_y$		从（4）可得
21	$map(f,tl(l)):\alpha_y$		(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	$cons(f(hd(l)),map(f,tl(l))):list(\alpha_v)$	$\alpha_r=\alpha_v\quad\alpha_y=list(\alpha_v)$	(Exp Funcall)
25	$if:boolean\times\alpha_i\times\alpha_i\to\alpha_i$		(Exp Id Fresh)
26	$if(\dots):$	$\alpha_i=\alpha_v=\alpha_m$	(Exp FunCall)
27	$match:(\alpha_j\times\alpha_j)\to\alpha_j$		(Exp Id Fresh)
28	$match(\dots):list(\alpha_m)$	$\alpha_j=\alpha_m$	(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

$$map:\forall\alpha\forall\beta.((\alpha\to\beta)\times list(\alpha))\to list(\beta)$$