

HW 7

肖桐 PB18000037

2020 年 11 月 8 日

解 1. (1). $pointer(array(3, int))$
(2). $array(3, pointer(int))$
(3). $pointer(array(3, int))$
(4). $pointer(array(3, pointer(int)))$
(5). $array(3, pointer(pointer(int)))$

解 2. 翻译方案为：

表 1: 翻译方案

$P \rightarrow D ; E$
$D \rightarrow D_1 ; D_2 \mid id : T \{addtype(id.entry, T.type); \}$
$T \rightarrow list\ of\ T_1 \{T.type = list(T_1.type); \} \mid char \{T.type = char; \} \mid integer \{T.type = integer; \}$
$E \rightarrow (L) \{E.type = list(L.type); \} \mid literal \{E.type = char; \}$ $\mid num \{E.type = integer; \} \mid id \{E.type = gettype(id.entry); \} \mid nil \{E.type = NIL; \}$
$L \rightarrow E , L_1 \{if(E.type \neq L_1.type) \{L.type = type_error; \} else \{L.type = L_1.type; \} \} \mid E \{L.type = E.type; \}$

解 3. (a). $(pointer(\alpha)) \times (\beta \rightarrow \gamma)$

(b). $\beta \times (\gamma \rightarrow \delta)$

这两个表达式最一般的合一替换为：

$S(\alpha) \rightarrow \alpha, S(\beta) \rightarrow pointer(\alpha), S(\gamma) \rightarrow pointer(\alpha), S(\delta) \rightarrow pointer(\alpha)$

得到合一表达式为： $(pointer(\alpha)) \times (pointer(\alpha) \rightarrow pointer(\alpha))$

若将 (b) 中 δ 换为 α , 则不存在合一替换.

解 4. 记 f 类型为 β , l 类型为 γ , map 类型为 θ . 则可得类型推导过程如下：

表 2: 类型推导过程

行	定型断言	代换	规则
(1)	$f : \beta$		(Exp id)
(2)	$l : \gamma$		(Exp id)
(3)	$map : \theta$		(Exp id)
(4)	$map(f, l) : \delta$	$\theta = \beta \times \gamma \rightarrow \delta$	(Exp FunCall)
(5)	$null : list(\alpha_n) \rightarrow boolean$		(Exp Id Fresh)
(6)	$null(l) : boolean$	$\gamma = list(\alpha_n)$	(Exp FunCall)
(7)	$nil : list(\alpha_i)$		(Exp Id Fresh)
(8)	$hd : list(\alpha_p) \rightarrow \alpha_p$		(Exp Id Fresh)
(9)	$hd(l) : \alpha_n$	$\alpha_p = \alpha_n$	(Exp FunCall)
(10)	$f(hd(l)) : \eta$	$\beta = \alpha_n \rightarrow \eta$	(Exp FunCall)
(11)	$tl : list(\alpha_q) \rightarrow \alpha_q$		(Exp Id Fresh)
(12)	$tl(l) : list(\alpha_n)$	$\alpha_q = \alpha_n$	(Exp FunCall)
(13)	$map(f, tl(l)) : \delta$	$\gamma = \alpha \times list(\alpha_n) \rightarrow \delta$	(Exp FunCall)
(14)	$cost : \alpha_r \times list(\alpha_r) \rightarrow \alpha_r$		(Exp Id Fresh)
(15)	$cost(f(hd(l)), map(f, tl(l))) : list(\alpha_r)$	$\eta = \alpha_r$ $\delta = list(\alpha_r)$ $\beta = \alpha_n \rightarrow \alpha_r$ $\theta = ((\alpha_n \rightarrow \alpha_r) \times list(\alpha_n)) \rightarrow list(\alpha_r)$	(Exp FunCall)
(16)	$if : boolean \times \alpha_s \times \alpha_s \rightarrow \alpha_s$		(Exp Id Fresh)
(17)	$if(...) : \alpha_r$	$\alpha_i = \alpha_r$	(Exp FunCall)
(18)	$=: \alpha_t \times \alpha_t \rightarrow \alpha_t$		(Exp Id Fresh)
(19)	$map(f, l) = ... : \alpha_r$	$\alpha_t = \alpha_r$	(Exp FunCall)

因此可以看出, map 的类型为: $((\alpha_n \rightarrow \alpha_r) \times list(\alpha_n)) \rightarrow list(\alpha_r)$.

因为 α_n, α_r 是未知类型的, 因此在前面加上任意符号, 可得:

$$\forall \alpha \forall \beta ((\alpha \rightarrow \beta) \times list(\alpha)) \rightarrow list(\beta)$$