Compile Principle assignment

Author Haihan Gao

第一题

(1)$pointer(array(3,int))$

(2)pointer(3,pointer(int))

(3)pointer(array(3,int))

(4)pointer(array(3,pointer(int)))

(5)pointer(3,pointer(pointer(int)))

教材5.6

P->D;E

D->D$_1$;D$_2$

D->id:T   {addtype(id.entry,T.type);}

T->list of T$_1$ {T.type=T$_1$.type;}

T->char {T.type=char;}

T->integer {T.type=integer;}

E->(L) {E.type=L.type;};

E->literal   {E.type=literal;}

E->num {E.type=num;}

E->id   {E.type=lookup(id.entry);}

E->nil {E.type=nil}

L->E,L$_1$ {L.type=L$_1$.type;}

L->E   {L.type=E.type;}

教材5.15

$\beta = (pointer(\alpha))$ $\gamma = pointer(\alpha)$ $\delta = pointer(\alpha)$

将β修改为α

教材5.17

这里面函数参数为f和l，假设f和l的类型是α β

Step1 列出类型声明

$f : \alpha$

$l : \beta$

$null : \forall\ \alpha. list(\alpha)->boolean$

$nil : \forall\ \alpha. list(\alpha)$

$cons : \forall\alpha.(\alpha \times list(\alpha))->list(\alpha)$

$hd : \forall\alpha. list(\alpha)->\alpha$

$\forall\alpha. list(\alpha)->list(\alpha)$

$if : \forall boollean \times list(\alpha) \times list(\alpha)->(\forall\alpha. list(\alpha)->boolean)$

match:$\forall\alpha\forall\beta->list(\alpha)$

```
match(
    match(f,l);
    if null(l) then nil
    else cons(f(hd(l)),map(f,tl(l)));
)
```

Step2 代换推导

为规则编号

(1)Exp id

(2)Exp Funcall

(3)Exp id Fresh

类型断言	代换	规则
$f : \alpha$		(1)
$l : \beta$		(1)
$map : \gamma$		(1)
$map(f.l) : \delta$	$\gamma = (\alpha \times \beta)- > \delta$	(2)
$null : list(\alpha i)- > boolean$		(3)
$null(l) : boolean$	$\beta = list(\alpha i)$	(2)
$nil : list(\alpha i + 1)$		(3)
$l : list(\alpha i)$		
$hd : list(\alpha i + 2)- > \alpha i + 2$		(3)
$hd : list(\alpha i + 2)- > \alpha i + 2$		(3)
$hd(l) : \alpha i$	$\alpha i + 2 = \alpha i$	(2)
$f(hd(l)) : \alpha i + 3$	$\alpha = \alpha i- > \alpha i + 3$	(1)
$f : \alpha- > \alpha i + 3$		
$tl : list(\alpha i + 4)- > list(\alpha i + 4)$		(3)
$tl(l) : list(\alpha i)$	$\alpha i + 4- > \alpha i$	(2)
$map : ((\alpha 0- > \alpha i + 3) \times list(\alpha i))- > \delta$		
$map(f, tl(l)) : \delta$		(2)
$cons : \alpha i + 5 \times list(\alpha i + 5)- > list(\alpha i + 5)$		(3)
$cons() : list(\alpha i + 3)$	$\alpha i + 5 = \alpha i + 3, \delta = list(\alpha i + 3)$	(2)
$if : boolean \times (\alpha i + 6) \times list(\alpha i + 6)- > list(\alpha i + 6)$		(3)
$if() : list(\alpha i + 1)$	$\alpha i + 6 = \alpha i + 1, \alpha i + 3 = \alpha i + 1$	(2)
$match : \alpha i + 7 \times \alpha i + 7- > \alpha i + 7$		(3)
$match() : list(\alpha i + 1)$	$\alpha i + 7 = list(\alpha i + 1)$	(2)