PL Final Solution

1 a) Ch06, p14

b) Ch06, pp16~17

2 a) Ch06, pp25~27

b) Ch06, p29

3 a) Ch06, pp33~34

b) Ch06, pp31~32

Ch06, p32

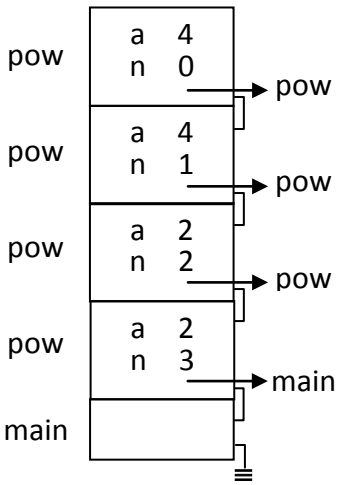
4 a) Chap10, pp5,9

b) Chap10, p.29

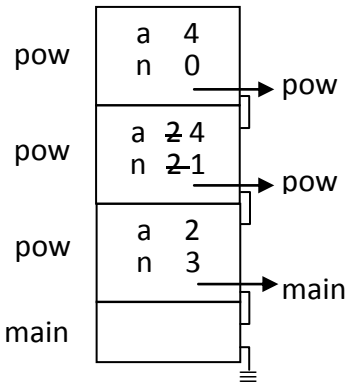
5 a)

```
int pow(int a,int n)
{
E: if (n==0) return 1;
   else if (n%2==1) return a*pow(a,n-1);
   else { a*=a; n/=2; goto E; }
}
```

b)



c)



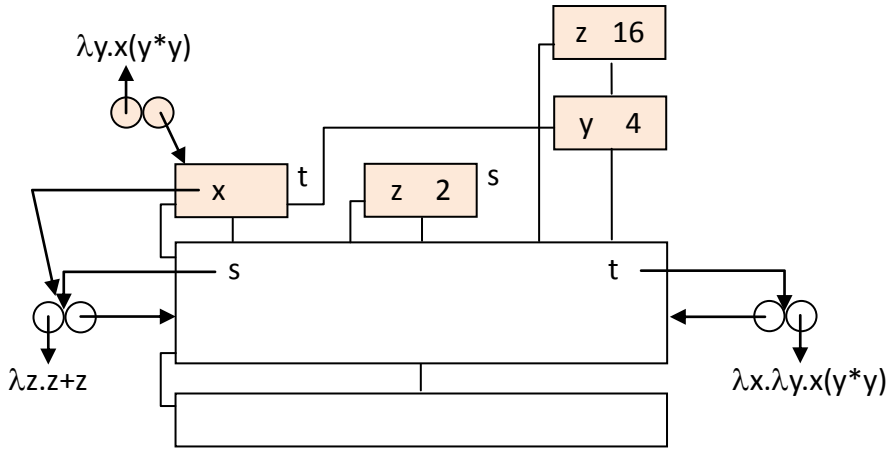
6 a) 3 0 1 3

b) 2 0 2 2

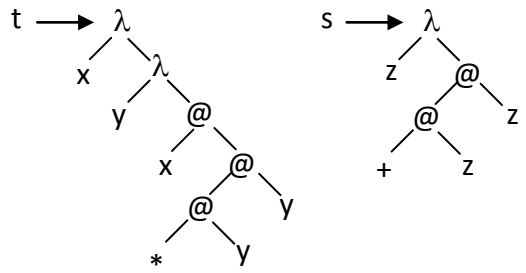
c) 2 1 1 2

d) 2 0 1 3

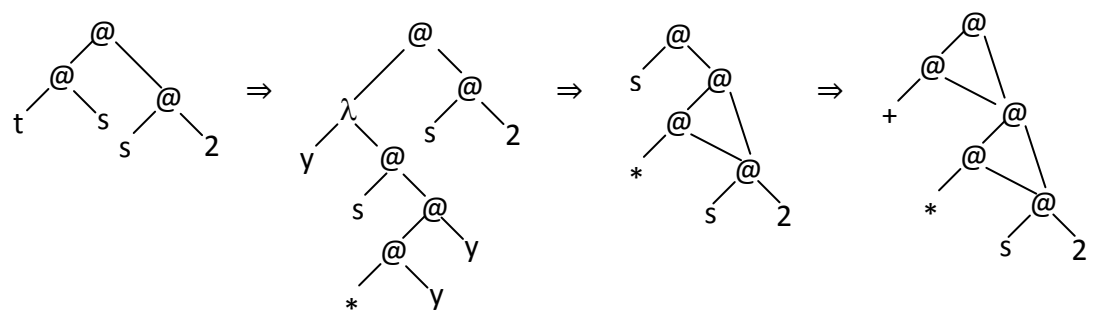
7

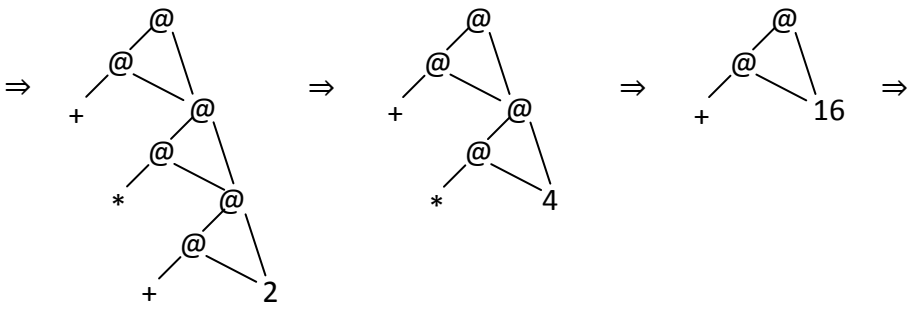

λy.x(y*y)
z 16
y 4
x
t
z 2
s
s
t
λz.z+z
λx.λy.x(y*y)

8 a)


t
λ
x
λ
y
@
x
@
@
y
*
y
s
λ
z
@
@
z
+
z

b)

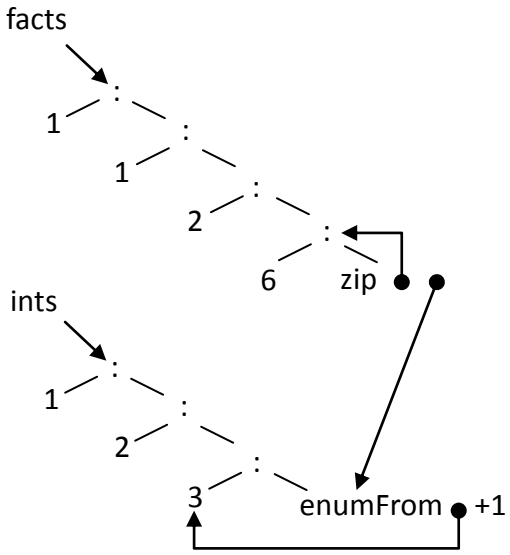

@
@
t
s
@
s
2
⇒
@
λ
@
y
s
2
@
s
@
@
y
*
y
⇒
@
s
@
@
*
@
s
2
⇒
@
@
+
@
@
*
@
s
2


⇒
@
@
+
@
@
*
@
@
+
2
⇒
@
@
+
@
@
*
4
⇒
@
@
+
16
⇒

32

9 a) `facts = 1:[x*y|(x,y)<-zip facts ints]`

b)

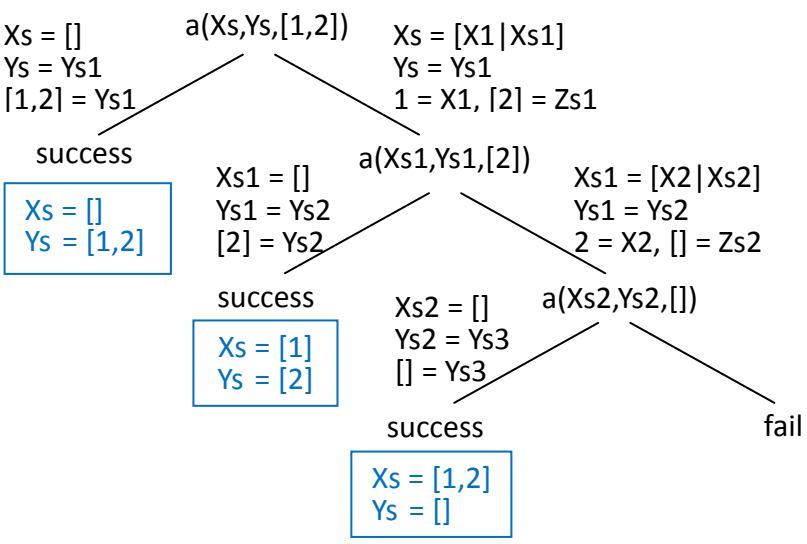


c) 1) Neither additions nor multiplications are needed.

2) One addition to obtain 4 (= 3+1)

One multiplication to obtain 24 (= 6*4)

10 a)



b) Chap16Prolog, pp18,27~28

In Prolog, the functionality of a parameter may be in or out, depending on the goal. Thus, the append relation ***alone*** may be used to answer various goals, say

?- append([a,b],[c],[a,b,c]).

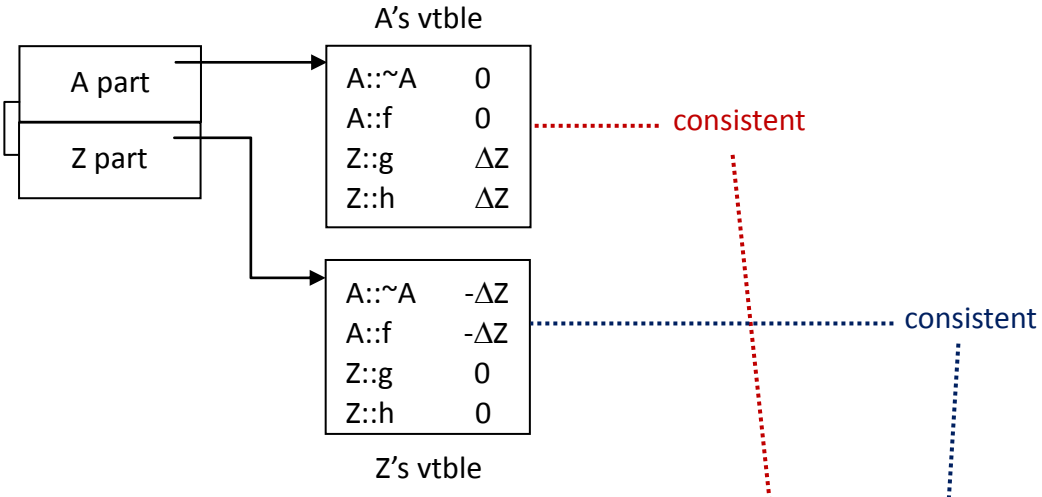
?- append([a,b],[c],Zs).

?- append(Xs,Ys,[a,b,c]).

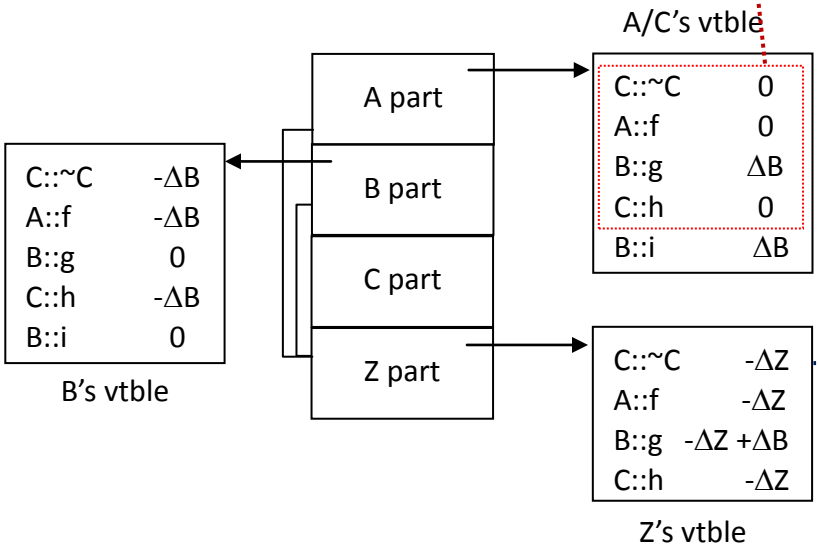
and so on.

However, these goals must be handled by ***distinct*** procedural programs, since the functionality of a parameter in such programs is fixed.

11 a)


A's vtble
A part
Z part
A::~A 0
A::f 0
Z::g ΔZ
Z::h ΔZ
consistent
A::~A -ΔZ
A::f -ΔZ
Z::g 0
Z::h 0
consistent
Z's vtble

b)


A/C's vtble
A part
B part
C part
Z part
C::~C 0
A::f 0
B::g ΔB
C::h 0
B::i ΔB
C::~C -ΔB
A::f -ΔB
B::g 0
C::h -ΔB
B::i 0
B's vtble
C::~C -ΔZ
A::f -ΔZ
B::g -ΔZ +ΔB
C::h -ΔZ
Z's vtble