COP-5555

PROGRAMMING LANGUAGE PRINCIPLES

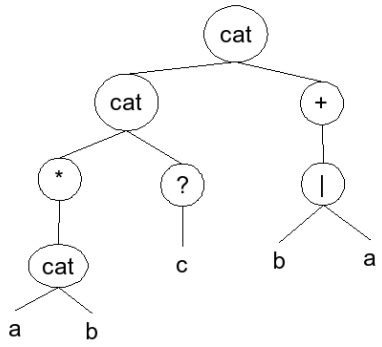
Summer 2007

Exam 1 Solutions

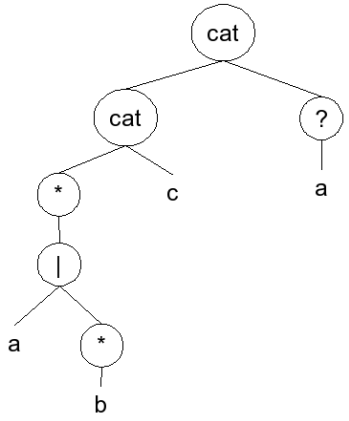
Problem 1: Regular Expression Grammar

a) Abstract Syntax Trees

i)



ii)

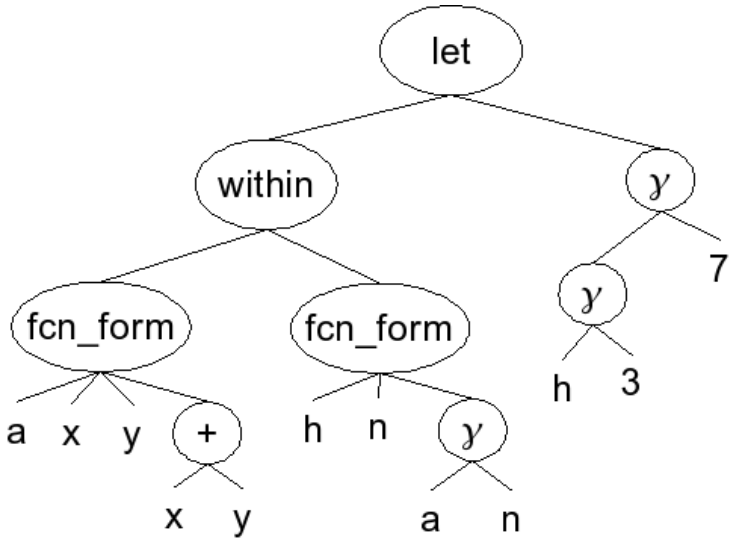


b) Parser Pseudocode

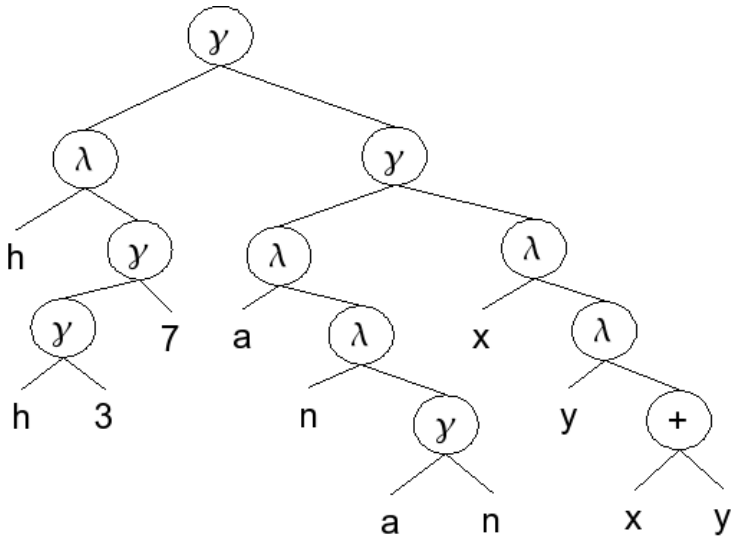
```
proc Expression
  Term
  if (nextToken = T_|)
    n := 1
    while (nextToken = T_|)
      read(T_|)
      Term
      n++
    BuildTree('|', n)

proc Term
  List
  while (nextTerm = T_ID or nextTerm = T_( )
    List
    BuildTree('cat',2)

proc List
  Primary
  if (nextToken = T_*)
    read(T_*)
    BuildTree('*',1)
  else if (nextToken = T_+)
    read(T_+)
    BuildTree('+',1)
  else if (nextToken = T_?)
    read(T_?)
    BuildTree('?',1)

proc Primary
  if (nextToken = T_( )
    read(T_()
    Expression
    read(T_))
  else read(T_ID)
```

Problem 2: Compilation

a) AST



b) PST



c) Control Structures

δ_0 : γ λ_1^h γ λ_2^a λ_3^x
δ_1 : γ γ h 3 7
δ_2 : λ_4^n
δ_3 : λ_5^y
δ_4 : γ a n
δ_5 : + x y

d) CSE Machine

Control	Stack	Env
e_0 δ_0	e_0	e_0 = PE
e_0 γ λ_1^h γ λ_2^a λ_3^x	e_0	
e_0 γ λ_1^h γ λ_2^a	${}^0\lambda_3^x$ e_0	
e_0 γ λ_1^h γ	${}^0\lambda_2^a$ ${}^0\lambda_3^x$ e_0	
e_0 γ λ_1^h e_1 δ_2	e_1 e_0	$e_1 = [{}^0\lambda_3^x$ /a$]e_0$
e_0 γ λ_1^h e_1 λ_4^n	e_1 e_0	
e_0 γ λ_1^h e_1	${}^1\lambda_4^n$ e_1 e_0	
e_0 γ λ_1^h	${}^1\lambda_4^n$ e_0	
e_0 γ	${}^0\lambda_1^h$ ${}^1\lambda_4^n$ e_0	
e_0 e_2 δ_1	e_2 e_0	$e_2 = [{}^1\lambda_4^n$ /h$]e_0$
e_0 e_2 γ γ h 3 7	e_2 e_0	
e_0 e_2 γ γ h 3	7 e_2 e_0	
e_0 e_2 γ γ h	3 7 e_2 e_0	
e_0 e_2 γ γ	${}^1\lambda_4^n$ 3 7 e_2 e_0	
e_0 e_2 γ e_3 δ_4	e_3 7 e_2 e_0	$e_3 = [3/\text{n}]e_1$
e_0 e_2 γ e_3 γ a n	e_3 7 e_2 e_0	
e_0 e_2 γ e_3 γ a	3 e_3 7 e_2 e_0	
e_0 e_2 γ e_3 γ	${}^0\lambda_3^x$ 3 e_3 7 e_2 e_0	
e_0 e_2 γ e_3 e_4 δ_3	e_4 e_3 7 e_2 e_0	$e_4 = [3/\text{x}]e_0$
e_0 e_2 γ e_3 e_4 λ_5^y	e_4 e_3 7 e_2 e_0	
e_0 e_2 γ e_3 e_4	${}^4\lambda_5^y$ e_4 e_3 7 e_2 e_0	
e_0 e_2 γ e_3	${}^4\lambda_5^y$ e_3 7 e_2 e_0	
e_0 e_2 γ	${}^4\lambda_5^y$ 7 e_2 e_0	
e_0 e_2 e_5 δ_5	e_5 e_2 e_0	$e_5 = [7/\text{y}]e_4$
e_0 e_2 e_5 + x y	e_5 e_2 e_0	
e_0 e_2 e_5 + x	7 e_5 e_2 e_0	
e_0 e_2 e_5 +	3 7 e_5 e_2 e_0	
e_0 e_2 e_5	10 e_5 e_2 e_0	
e_0 e_2	10 e_2 e_0	
e_0	10 e_0	
-	10	

Problem 3

a) (λa. λb. a b) (λb. b + e) 3

$\Rightarrow_\beta$ (λb. (λb. b + e) b) 3
$\Rightarrow_\beta$ (λb. b + e) 3
$\Rightarrow_\beta$ 3 + e

b) (λf. f f) (λh. h ((λf. λh. f) f))

$\Rightarrow_\beta$ (λh. h ((λf. λh. f) f)) (λh. h ((λf. λh. f) f))
$\Rightarrow_\beta$ (λh. h ((λf. λh. f) f)) ((λf. λh. f) f)
$\Rightarrow_\beta$ ((λf. λh. f) f) ((λf. λh. f) f)
$\Rightarrow_\beta$ (λh. f) ((λf. λh. f) f)
$\Rightarrow_\beta$ f

c) (λa. a a) (λa. λb. a b a)

$\Rightarrow_\beta$ (λa. λb. a b a) (λa. λb. a b a)
$\Rightarrow_\beta$ (λb. (λa. λb. a b a) b (λa. λb. a b a))
$\Rightarrow_\beta$ (λb. (λw. b w b) (λa. λb. a b a))
$\Rightarrow_\beta$ (λb. b (λa. λb. a b a) b)

Problem 4: RPAL Programming

```
let merge s1 s2 =
   rm s1 s2 ''  where
   rec rm s1 s2 r =
        s1 eq '' & s2 eq '' -> r
     |  s1 eq '' -> r @Conc s2
     |  s2 eq '' -> r @Conc s1
     | ( let c1 = Stem s1 and c2 = Stem s2 in
           c1 < c2 -> rm (Stern s1) s2 (r @Conc c1)
         |            rm s1 (Stern s2) (r @Conc c2)
       )
in
   Print (merge 'abd' 'bce')
```