

Procesadores de Lenguaje - Examen de Junio del 2006

Enunciado:

Considera los dos siguientes operadores binarios: $\langle x \rangle$ y $\langle + \rangle$. $\langle x \rangle$ tiene mayor prioridad que $\langle + \rangle$ y asocia a izquierdas. Por su parte, $\langle + \rangle$ asocia a derechas. Se pide:

- Formaliza la sintaxis para las expresiones formadas mediante los operadores $\langle x \rangle$ y $\langle + \rangle$. Las expresiones básicas pueden ser números o variables, y, como es habitual, pueden utilizarse paréntesis para alterar las prioridades y asociatividades. La sintaxis debe reflejar las prioridades y asociatividades de los operadores siguiendo los patrones explicados en la asignatura.
- Se dota a la máquina P de las instrucciones xor y xnor, xor desapila el valor de la cima de la pila v1 y el valor de la subcima v0, y apila v0 $\langle x \rangle$ v1. Por su parte, xnor también desapila el valor de cima de la pila v1 y el valor de la subcima v0, y apila v0 $\langle + \rangle$ v1. Utilizando estas instrucciones, formaliza mediante una gramática de atributos la traducción de las expresiones caracterizadas en (a) a código de la máquina P.
- Aplicando los patrones de eliminación de recursión a izquierdas y de factorización en gramáticas de atributos, transforma la gramática de atributos obtenida en (b) para obtener una gramática de atributos equivalente en la que la gramática incontextual subyacente sea LL(1).
- Escribe el esquema de traducción orientado a la gramática de atributos obtenida en el apartado C.

Solución:

a)

$\text{Exp0} ::= \text{Exp1 Op0 Exp0}$

$\text{Exp0} ::= \text{Exp1}$

$\text{Op0} ::= \langle + \rangle$

$\text{Exp1} ::= \text{Exp1 Op1 Exp2}$

$\text{Exp1} ::= \text{Exp2}$

$\text{Op2} ::= \langle x \rangle$

$\text{Exp2} ::= \text{num}$

Exp2 ::= id

Exp2 ::= (Exp0)

b)

Exp0 ::= Exp1 Op0 Exp0

Exp1.tsh = Exp0₀.tsh
Exp0₁.tsh = Exp0₀.tsh
Exp0₀.cod = Exp1.cod || Exp0₁.cod || Op0.op

Exp0 ::= Exp1

Exp1.tsh = Exp0.tsh
Exp0.cod = Exp1.cod

Op0 ::= <+>

Op0.op = xnor

Exp1 ::= Exp1 Op1 Exp2

Exp1₁.tsh = Exp1₀.tsh
Exp2.tsh = Exp1₀.tsh
Exp1₀.cod = Exp1₁.cod || Exp2.cod || Op1.op

Exp1 ::= Exp2

Exp2.tsh = Exp1.tsh
Exp1.cod = Exp2.cod

Op1 ::= <x>

Op1.op = xor

Exp2 ::= num

Exp2.cod = apila(valorDe(num.lex))

Exp2 ::= id

Exp2.cod = apila_dir(dirID(Exp2.tsh,id.lex))

$\text{Exp2} ::= (\text{Exp0})$

$\text{Exp0.tsh} = \text{Exp2.tsh}$
 $\text{Exp2.cod} = \text{Exp0.cod}$

c)

Haciendo factorización y eliminación de recursión a izquierdas obtenemos una gramática LL(1).

Factorización:

$\text{Exp0} ::= \text{Exp1 RExp0}$

$\text{Exp1.tsh} = \text{RExp0.tsh} = \text{Exp0.tsh}$

$\text{Exp0.cod} = \text{RExp0.cod}$

$\text{RExp0.codh} = \text{Exp1.cod}$

$\text{RExp0} ::= \text{Op0 Exp0}$

$\text{Exp0.tsh} = \text{RExp0.tsh}$

$\text{RExp0.cod} = \text{RExp0.codh} \parallel \text{Exp0.cod} \parallel \text{Op0}$

$\text{RExp0} ::= \text{lambda}$

$\text{RExp0.cod} = \text{RExp0.codh}$

Eliminación de recursión a izquierdas:

$\text{Exp1} ::= \text{Exp2 RExp1}$

$\text{Exp1.tsh} = \text{Exp2.tsh} = \text{RExp1.tsh}$

$\text{Exp1.cod} = \text{RExp1.cod}$

$\text{RExp1.codh} = \text{Exp2.cod}$

$\text{RExp1} ::= \text{Op1 Exp2 RExp1}$

$\text{RExp1}_1.\text{tsh} = \text{Exp2.tsh} = \text{RExp1}_0.\text{tsh}$

$\text{RExp1}_1.\text{codh} = \text{RExp1}_0.\text{codh} \parallel \text{Exp2.cod} \parallel \text{Op1.op}$

$\text{RExp1}_0.\text{cod} = \text{RExp1}_1.\text{cod}$

$\text{RExp1} ::= \text{lambda}$

$\text{RExp1.cod} = \text{RExp1.codh}$

d)

$\text{Exp1} ::= \{ \text{Exp2.tsh} = \text{Exp1.tsh}; \}$

Exp2

$\{ \text{RExp1.tsh} = \text{Exp1.tsh}; \\ \text{RExp1.codh} = \text{Exp2.cod}; \}$

RExp1

$\{ \text{Exp1.cod} = \text{RExp1.cod}; \}$

$\text{RExp1} ::= \text{Op1}$

$\{ \text{Exp2.tsh} = \text{RExp1}_0.\text{tsh}; \}$

Exp2

$\{ \text{RExp1}_1.\text{codh} = \text{RExp1}_0.\text{codh} \parallel \text{Exp2.cod} \parallel \text{Op1.op}; \}$

RExp1

$\{ \text{RExp1}_0.\text{cod} = \text{RExp1}_1.\text{cod}; \}$

$\text{RExp1} ::= \text{lambda}$

$\{ \text{RExp1.cod} = \text{RExp1.codh}; \}$

$\text{Exp0} ::= \{ \text{RExp1.tsh} = \text{Exp0.tsh}; \}$

Exp1

$\{ \text{RExp0.tsh} = \text{Exp0.tsh}; \\ \text{RExp0.codh} = \text{Exp1.cod}; \}$

RExp0

{Exp0.cod = RExp0.cod;}

RExp0 ::= Op0

{Exp0.tsh = RExp0.tsh;}

Exp0

{RExp0.cod = RExp0.codh || Exp0.cod || Op0.op;}

RExp0 ::= lambda

{RExp0.cod = RExp0.codh;}

Exp2 ::= num

{Exp2.cod = apila(valorDe(num.lex));}

Exp2 ::= id

{Exp2.cod = apila_dir(Exp2.tsh[id.lex].dir);}

Exp2 ::= (

{Exp0.tsh = Exp2.tsh;}

Exp0

{Exp2.cod = Exp0.cod;}

)

Op0 ::= <+>

{Op0.op = xnor;}

Op1 ::= <x>

{Op1.op = xor;}