

Procesadores de Lenguaje. Examen Septiembre 2006

Ejercicio 1 [4 punto]

Considera los tres siguientes operadores binarios: $\&$, \otimes y \oplus . \oplus tiene más prioridad que $\&$ y que \otimes , y asocia a izquierdas. Por su parte, $\&$ y \otimes tienen la misma prioridad. \otimes asocia a derechas, y $\&$ no es asociativo. Se pide:

- a) [0,5 puntos] Formaliza la sintaxis para las expresiones formadas mediante los operadores $\&$, \otimes y \oplus . Las expresiones básicas son únicamente las constantes lógicas **true** y **false** (este lenguaje no contiene otro tipo de constantes, ni tampoco 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.

b) [1 punto] Se dota a la máquina P de las instrucciones **and**, **xor**, **xnor**

- **and** desapila el valor de la cima de la pila v_1 y el valor de la subcima v_0 , y apila $v_0 \& v_1$.
- **xor** desapila el valor de la cima de la pila v_1 y el valor de la subcima v_0 , y apila $v_0 \otimes v_1$.
- **xnor** desapila el valor de cima de la pila v_1 y el valor de la subcima v_0 , y apila $v_0 \oplus v_1$.

Utilizando estas instrucciones, formaliza mediante una gramática de atributos la traducción de las expresiones caracterizadas en (a) a código de la máquina P. En esta traducción, **true** debe representarse como **1** y **false** como **0**.

c) [1 punto] 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).

d) [1,5 puntos] Calcula los conjuntos *primero* y *siguiente* de los no terminales de la gramática incontextual subyacente a la gramática de atributos del apartado (c) y construye la tabla de análisis para el analizador descendente no recursivo asociado con esa gramática.

Ejercicio 2 [5 puntos]

Considera el programa listado en el reverso de la hoja. Se pide:

a) [0,5 puntos] Haz un esquema de la estructura de la memoria de una máquina P con memoria dinámica y que utiliza un *display* para ejecutar un lenguaje con procedimientos recursivos anidados y con parámetros, explicando la finalidad de cada segmento de memoria.

b) [1,25 puntos] Haz un esquema y explica el contenido de la tabla de símbolos en el punto (*1*). Supondremos que enteros y punteros ocupan una posición en la memoria de la máquina P.

c) [1 punto] Escribe el código P necesario para realizar la llamada al procedimiento y el paso de parámetros en (*2*).

d) [1,25 puntos] La ejecución del programa comienza realizando el bucle (*3*), donde se construye el árbol binario que se muestra en la figura del reverso. Después se invoca al procedimiento *preorden*. Desde allí a *listaEnPreorden...* En esta instante, y una vez activado *listaEnPreorden* tras esta primera llamada desde *preorden*, haz un esquema detallado de la memoria de la máquina P en el punto (*1*). Incluye detalles de los marcos de activación y de la memoria dinámica.

e) [1 punto] Formaliza mediante una gramática de atributos la traducción de las instrucciones **REPEAT** del tipo de la de (*3*). Haz primero un esquema que muestre cómo se organiza la traducción y después formalízalo mediante una gramática de atributos.

Ejercicio 3 [1 punto]

Responde a las siguientes cuestiones:

a) [0,5 puntos] ¿Cuál es la propiedad básica de los lenguajes incontextuales en la que se basan los analizadores ascendentes LR?

b) [0,5 puntos] Considera un autómata LR(0) que incluye el siguiente estado:

A ::= $\alpha.a\beta$
B ::= $\alpha.a\gamma$

donde α , β y γ son formas sentenciales, y a es un terminal. Suponiendo que los únicos conflictos pueden ser debidos a este estado, las tablas generadas por el método LR(0), ¿tienen algún conflicto? ¿Y las generadas por el método SLR? Razona tu respuesta.

(PROGRAMA EJERCICIO 2)

```
programa examen;

tipo tarbol = ^tcelda;
tipo tcelda = record
    int item;
    tarbol der;
    tarbol izq
end;
tipo tvect = array [100] of int;

tvect vect;
tarbol arbol;
tarbol arbolaux;
int i;

proc preorden(tarbol arbol);
int i;

proc listaEnPreorden(tarbol arbol);
begin (* <== 1 *)
    if (arbol <> NIL) AND (i < 99) then
    begin
        vect[i] := arbol^.item;
        i := i+1;
        listaEnPreorden(arbol^.der); (* <== 2 *)
        listaEnPreorden(arbol^.izq)
    end
end; (* fin proc listaEnPreorden*)

begin (* proc preorden *)
    i := 0;
    listaEnPreorden(arbol);
    vect[i] := -1
end; (* fin proc preorden*)

begin (* programa principal *)
    i := 0;
    arbol:= NIL;
    repeat (* <== 3 *)
    begin
        new(arbolaux);
        arbolaux^.item := i;
        arbolaux^.izq := arbol;
        arbolaux^.der := NIL;
        arbol := arbolaux;
        i := i+1
    end
until i > 60;
preorden(arbol)

end.
```

(FIGURA EJERCICIO 2)

