A partir de hoc4 se usan dos etapas en hoc. ¿Cuales son y que hacen?

1. Generación de código: Genera el código a ser ejecutado.
2. Ejecución de código: Ejecuta el código generado.

Escriba 3 cosas importantes que se almacenan usualmente en un marco (o registro de activación) de función.

1. Variables
2. Parámetros
3. Nombre de la función

**Falso o Verdadero (F/V)**

|  |  |
| --- | --- |
| 1. En lenguaje C los parámetros formales no tienen nombre | (F) |
| 1. En lenguaje C las variables locales (no estáticas) se crean cuando se entra a una función y se destruyen cuando se sale de la función | (V) |
| 1. En hoc los parámetros formales no tienen nombre | (V) |
| 1. No es posible definir funciones recursivas en hoc6 | (V) |
| 1. En hoc no hay variables locales | (V) |
| 1. Es imposible que la pila de hoc se desborde (Stack Overflow) | (F) |
| 1. En hoc6 cuando una función termina su ejecución se saca su marco de la pila de llamadas | (V) |
| 1. En hoc6 cuando una función termina su ejecución se sacan sus parametros de la pila | (V) |
| 1. En hoc6 los parámetros reales son listas de expresiones | (V) |
| 1. En hoc6 $n es el enesimo parametro | (V) |
| 1. En hoc4 el código que ejecuta la maquina virtual de pila esta en prefijo (considere como se ejecuta una operacion de suma | (F) |
| 1. Los valores de los atributos sintetizados se calculan a partir de los valores de atributos de su nodo padre o sus nodos hermanos. | (F) |
| 1. En hoc4 la variable pc se usa en la etapa de ejecucion | (V) |
| 1. En hoc4 la variable progp se usa en la etapa de generacion de codigo | (V) |
| 1. La variable progp indica la posicion de la RAM donde esta la sig. Instruccion a ejecutar | (F) |
| 1. La variable pc indica cual es la sig. posicion de la RAM donde se almacenara una instruccion | (F) |
| 1. La funcion code retorna la direccion donde se guardo la instruccion que recibe como parametro | (V) |
| 1. La funcion define pone la direcccion de la 1a instrucccion de una funcion definida por el usuario en la tabla de simbolos | (V) |
| 1. La variables en HOC son de tipo entero | (F) |
| 1. La variables en HOC son de tipo doble | (V) |
| 1. En hoc las variables son globales | (V) |
| 1. En hoc4 la funcion yyparse() genera el codigo | (V) |
| 1. En hoc4 la funcion execute() genera el codigo | (F) |
| 1. En hoc4 el arreglo prog es la RAM de la maquina virtual de pila | (V) |
| 1. En hoc4 el arreglo prog es la pila de la maquina virtual de pila | (F) |
| 1. En hoc6 el arreglo frame es la RAM de la maquina virtual de pila | (F) |

1.-Un \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ es [A→ α . β , a] donde A→ αβ es una producción y a es un terminal o $.

a) mango

b) prefijo viable

***c) elemento LR (1)***

d) elemento LR (0)

2.-Es una producción de G con un punto en cierta posición del lado derecho.

a) mango

b) prefijo viable

c) elemento LR (1)

***d) elemento LR (0)***

3.-Son prefijos de las formas de frase derecha que pueden aparecer en la pila

a) mango

b) elemento LR (0)

c) ***prefijo viable***

d) elemento LR (1)

4.-Un \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ de una forma de frase derecha gamma es una producción S → β y una posición de gamma donde la cadena β podría encontrarse y sustituirse por A para producir la forma de frase derecha previa en una derivación por la derecha de gamma.

a) prefijo viable

***b) mango***

c) elemento LR (0)

d) elemento LR (1)

1-Ejecutar la funcion (poner el contador de programa igual a la direccion de su primera instrucción y ejecutar la instrucción a la que apunta el contador de programa) y meter el valor de retorno de la funcion en la pila.

2-Meter los parametros en la pila y meter el marco de la funcion en la pila de llamadas.

3-Poner el contador de programa igual a la direccion de retorno y ejecutar la instrucción a la que apunta el contador de programa.

4-Sacar parametros de la pila y sacar marco de la pila de llamadas.

5.-De acuerdo al mecanismo de llamada a funcion cual es el orden correcto?

a) 1, 2, 3 , 4

b) 1, 3 , 4 ,2

***c) 2, 1 , 4, 3***

d) 4, 3 , 2 , 1

**Problema**.- Considere la siguiente gramática :

1) S → X

2) X → a X c

3) X → X X

4) X → b

Calcule

cerradura({X → X . X})

***X →*** ∙ ***aXc***

***X*** →∙ ***b***

cerradura({X → X X . })

e ir\_a ( { X → a . X c } , X )

***X*** → ***aX*** ∙ ***c***

***Para el Análisis LR las gramáticas se muestran con sus producciones numeradas***

***Para cada gramática:***

***- Calcule los conjuntos PRIMERO y SIGUIENTE***

***- Construya la tabla Análisis Sintáctico Predictivo no Recursivo ( LL(1) )***

***- Construya la colección de conjuntos de elementos LR (0)***

***- Construya la tabla SLR***

**Problema**

1) A → xA

2) A → yA

3) A → y

-Explicar porque esta grammatica no es LL(1).

I1=ir\_a(I0, x ) , I2=ir\_a(I0, y ) , I3=ir\_a(I0, A ), I4=ir\_a(I1, A) , I5=ir\_a(I2, A )

PRIM(A) = { x , y }

SIG(A) = { $ }

|  |  |  |  |
| --- | --- | --- | --- |
|  | x | y | $ |
| A | A → xA | A → yA |  |
|  |  | A → y |  |

**Problema**

1) S → ( S )

2) S → e

I1=ir\_a(I0, ( ) , I2=ir\_a(I0, e ) , I3=ir\_a(I0, S ), I4=ir\_a(I1, S ) , I5=ir\_a(I4, ) )

cadenas

( e )

( ( e ) )

( ( ( e ) ) )

( ( ( ( e ) ) ) )

a) PRIM (S) = { ( , e} SIG(E)={ ) , $ } I0

S → ( S )

S → ∙e

I1 ir\_a(I0, ( )

S→ ( ∙ S )

S→ ∙ ( S ) S → ∙e

I2 ir\_a(I0 , e)

S→ e∙ “reducir 2”

I3 ir\_a(I0, S)

S’ → S∙ “cadena aceptada”

I4 ir\_a(I1, S)

S→(S∙ )

I5 ir\_a(I4, ) )

S→( S ) ∙ “reducir 1”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ( | ) | e | $ | S |
| 0 | D1 |  | D2 |  | 3 |
| 1 | D1 |  | D2 |  | 4 |
| 2 |  | R2 |  | R2 |  |
| 3 |  |  |  | Aceptar |  |
| 4 |  | D5 |  |  |  |
| 5 |  | R1 |  | R1 |  |

|  |  |  |
| --- | --- | --- |
| Pila | Entrada | Acción |
| 0 | (e)$ | D1 |
| 0(1 | e)$ | D2 |
| 0(1e2 | )$ | R2 |
| 0(1S4 | )$ | D5 |
| 0(1S4)5 | $ | R1 |
| 0S3 | $ | ACEPTAR |

**Problema**

1) E → n

2) E → ( E , E )

Donde n es un entero

I1=ir\_a(I0, ( ) , I2=ir\_a(I0, n ) , I3=ir\_a(I0, E), I4=ir\_a(I1, E) , I5=ir\_a(I4, , ) ,

I6=ir\_a(I5, E), I7=ir\_a(I6, ) )

cadena ( (21 , 18) , 17 )

a) PRIM (E) = { n, (} SIG(E)={ $, )m ‘,’ }

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | N | ( | , | ) | $ |
| E | E → n | E→( E , E ) |  | A → d |  |

c)

I0

E → ∙ n

E → ∙ ( E , E )

I1 ir\_a(I0, ( )

E → ( E ∙ , E )

E → ∙ n

E → ∙ ( E, E)

I2 ir\_a(I0,n)

E → n ∙ “reducir 1”

I3 ir\_a(I0, E)

S’ → E∙ “cadena aceptada”

I4 ir\_a(I1, E)

E →(E∙ ,E)

I5 ir\_a(I4, , )

E → ( E , ∙ E)

E → ∙ n

E → ∙ ( E , E)

I6 ir\_a(I5, E)

E → ( E , E ∙ )

I7 ir\_a ( I6 , ) )

E → ( E , E ) ∙ “reducir 2”

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ( | ) | $ | , | n | E |
| 0 | D1 |  |  |  | D2 | 3 |
| 1 | D1 |  |  |  | D2 | 4 |
| 2 |  | R1 | R1 | R1 |  |  |
| 3 |  |  | ACEPTAR |  |  |  |
| 4 |  |  |  | D5 |  |  |
| 5 | D1 |  |  |  | D2 | 6 |
| 6 |  | D7 |  |  |  |  |
| 7 |  | R2 | R2 | R2 |  |  |

|  |  |  |
| --- | --- | --- |
| Pila | Entrada | Acción |
| 0 | ((21,18),17)$ | D1 |
| 0(1 | (21,18),17)$ | D1 |
| 0(1(1 | 21,18),17)$ | D2 |
| 0(1(1n2 | ,18),17)$ | R1 |
| 0(1(1E4 | ,18),17)$ | D3 |
| 0(1(1E4,5 | 18),17)$ | D2 |
| 0(1(1E4,5n2 | ),17)$ | R1 |
| 0(1(1E4,5E6 | ),17)$ | D7 |
| 0(1(1E4,5E6,7 | ,17)$ | R2 |
| 0(1(1E4 | ,17)$ | D3 |
| 0(1E4,5 | 17)$ | D2 |
| 0(1E4,5n2 | )$ | R1 |
| 0(1E4,4E6 | )$ | D7 |
| 0(1(E4(5E6)7 | $ | R2 |
| 0E3 | $ | ACEPTAR |

**Problema**

1) S→ [ L ]

2) S → a

3) L → L , S

4) L → S

I1=ir\_a(I0, [ ) , I2=ir\_a(I0, a ) , I3=ir\_a(I0, S ), I4=ir\_a(I1, S ) , I5=ir\_a(I1, L ) , I6=ir\_a(I5, ] ),

I7=ir\_a(I5 , , ) , I8=ir\_a(I7, S)

PRIM (S) = { [ , a } SIG(S) = { $, )]. ‘,’}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | [ | , | ] | A | $ |
| S | S → [ L ] |  |  | S → a |  |
| L | L → L, S  L → L, S |  |  | L → L, S  L → S |  |

I0

S → [ L ]

S → ∙ a

I1 ir\_a(I0,[)

S→ [∙L ]

L→ ∙ L ,S

L→ ∙ S

S → ∙ a

S→ ∙[ L ]

I2 ir\_a(I0,a)

S→ a∙ “reducir 2”

I3 ir\_a(I0, S)

S’ → S∙ “cadena aceptada”

I4 ir\_a(I1, S)

L→S∙ “reducir 4”

I5 ir\_a(I1, L )

S→ [ L ∙ ]

L→L ∙ , S

E → ∙ ( E , E)

I6 ir\_a(I5, ] )

S → [ L ] ∙ “reducir 1”

I7 ir\_a ( I5 , , )

L → L , ∙ S

S → ∙ a

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | [ | A | . | ] | $ | S | L |
| 0 | D1 | D2 |  |  |  | 3 |  |
| 1 | D1 | D2 |  |  |  | 4 | 5 |
| 2 |  |  | R2 | R2 | R2 |  |  |
| 3 |  |  |  |  | ACEPTAR |  |  |
| 4 |  |  |  | R4 | R4 |  |  |
| 5 |  |  | D7 | D6 |  |  |  |
| 6 |  |  | R1 | R1 | R1 |  |  |
| 7 | D1 | D2 |  |  |  | 8 |  |
| 8 |  |  | R3 | R3 |  |  |  |

Problema 12

1) S → d c a

2) S →d A b

3) S →A a

4) A → c

I1=ir\_a(I0, d ) , I2=ir\_a(I0 , c ) , I3=ir\_a(I0, S ), I4=ir\_a(I0, A ) , I5=ir\_a(I1 , c ) ,

I6=ir\_a(I1, A), I7=ir\_a(I4, a ) , I8=ir\_a(I5, a ) , I9=ir\_a(I6, b )

PRIM (A)={ c } SIG(A)={ a , b }

PRIM (S)={ d , c } SIG(L)={ $ }

b)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | D | C | A | $ |
| S | S → dca  S → dAc | S → Aa |  |  |
| A |  |  |  |  |

c)

I0

S’ → ∙ S

S → dca

S → dAb

S → Aa

A → ∙c

I1 ir\_a(I0, d)

S → d ∙ ca

S → d∙Ab

A→ ∙ c

I2 ir\_a(I0,c)

S→ c∙ “reducir 4”

I3 ir\_a(I0, S)

S’ → S∙ “cadena aceptada”

I4 ir\_a(I1, A)

S → A ∙ a “reducir 4”

I5 ir\_a(I1, c )

S→ dc ∙ a

A→ c ∙ “reducir 4”

I6 ir\_a(I1, A )

S →dA ∙ b “reducir 1”

I7 ir\_a ( I4 , a )

S →Aa ∙ “reducir 3”

I8 ir\_a (I5 , a )

S →dca ∙ “reducir 1”

I9 ir\_a (I6 , b)

S →dAb ∙

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | d | a | b | c | $ | S | A |
| 0 | D1 |  |  | D2 |  | 3 | 4 |
| 1 |  |  |  | D5 |  |  | 6 |
| 2 |  | R4 | R4 |  |  |  |  |
| 3 |  |  |  |  | ACEPTAR |  |  |
| 4 |  | D7 |  | R4 | R4 |  |  |
| 5 |  | D8 / R4 | R4 |  |  |  |  |

Problema 14

1) S → c S A d

2) S → d

3) A → a B

4) A →a

5) B → a

6) B → b

I1=ir\_a(I0 , c ) , I2=ir\_a(I0, d ) , I3=ir\_a(I0, S), I4=ir\_a(I1, S) , I5=ir\_a(I4, a ) ,

I6=ir\_a(I4, A), I7=ir\_a(I5, a ) , I8=ir\_a(I5, b ) , I9=ir\_a(I5, B), I10=ir\_a(I6, d)

cadena: cdad

PRIM (S) = {d,c} SIG(S) = {$ , a}

PRIM(A) = {a} SIG(A) = { d }

PRIM(B) = { a, b } SIG(B) = { d }

I0 S’ → ∙S

S’ → ∙cSAd

S →∙ d

I1 ir\_a(I0,c)

S’ → c∙SAd

S’ → ∙cSAd

S→ ∙ d

I2 ir\_a(I0,s)

S→ d ∙ “reducir 2”

I3 ir\_a(I0,d)

S’→ S∙ “cadena aceptada”

I4 ir\_a(I0,S)

S’ → cS∙Ad

A → aB

A → a

I5 ir\_a(I1,a)

A→a∙ B

A→a∙

B → ∙a

B →∙b

I6 ir\_a(I4,A)

S→ cSA ∙d

I7 ir\_a(I2,a)

B → a∙

I8 ir\_a(I5,6)

B→b∙ “reducir 6”

I9 ir\_a(I1,B)

A→aB∙ “reducir 3”

I10 ir\_a(I6,c)

S→ cSAd ∙ “reducir 1”

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c | d | $ | S | A | B |
| 0 |  |  | D1 | D2 |  | 3 |  |  |
| 1 |  |  | D1 | D2 |  | 4 |  |  |
| 2 | R2 |  |  |  | R2 |  |  |  |
| 3 |  |  |  |  | ACEPTAR |  |  |  |
| 4 | D5 |  |  |  |  |  | 6 |  |
| 5 | D7 | D8 |  | R4 |  |  |  | 9 |
| 6 |  |  |  | D10 |  |  |  |  |
| 7 |  |  |  | R5 |  |  |  |  |
| 8 |  |  |  | R6 |  |  |  |  |
| 9 |  |  |  | R3 |  |  |  |  |
| 10 | R1 |  |  |  | R1 |  |  |  |

|  |  |  |
| --- | --- | --- |
| Pila | Entrada | Acción |
| 0 | cdad$ | D1 |
| 0c1 | dad$ | D2 |
| 0c1d2 | ad$ | R2 |
| 0c1S4 | ad$ | D5 |
| 0c1S4a5 | d$ | R4 |
| 0c1S4A6 | d$ | D10 |
| 0c1S4a5d10 | $ | R1 |
| 0S3 | $d | ACEPTAR |