

ANALIZADOR SINTÁCTICO DESCENDENTE CON RETROCESO

Algoritmo:

Entrada: Gramática de contexto libre no recursiva por la izquierda

Cadena de entrada: $w = a_1, a_2, \dots, a_n, n \geq 0$

Salida: Un *parse* a izquierdas para w , si existe; en otro caso, error

Procedimiento:

1. Ordenar y numerar todas y cada una de las reglas de la gramática
2. Comenzar la generación del árbol con el axioma de la gramática. Éste será el nodo activo, inicialmente

Ejecutar:

- a. Si el nodo activo es un no terminal A :
 - Escoger la primera alternativa para A y crear sus descendientes
 - Hacer el de más a la izquierda el nodo activo
 - Si A no tiene más descendientes, hacer como nodo activo el siguiente a la derecha de A
 - b. Si el nodo activo es un terminal a :
 - Comparar el *token* de entrada con a
 - Si son iguales, el nodo activo será el siguiente a la derecha de a
 - Si no lo son, regresar al nodo donde se aplicó la regla anterior previamente, y tratar la siguiente alternativa (otra derivación)
 - Si no hay más alternativas, volver al nodo anterior y seguir así
- Hasta llegar al final de w (si no se puede llegar a construir el árbol, error)
3. El *parse* será la secuencia de los números de las reglas utilizadas

ANALIZADOR ASCENDENTE CON RETROCESO (REDUCCIÓN-DESPLAZAMIENTO)

Algoritmo:

Entrada: Gramática de contexto libre sin reglas λ y sin ciclos

Cadena de entrada: $w=a_1, a_2, \dots, a_n, n \geq 1$

Las reglas de la gramática están numeradas

Salida: Un *parse* a derechas para w , si existe; en otro caso, error

Procedimiento:

1. Se considera que la cadena a analizar se encuentra en una pila de entrada
2. Explorar la cabeza de la pila de trabajo
 - a. Comprobar si hay un lado derecho de alguna regla que pueda equipararse con los símbolos de la cabeza
 - Si es así:
 - Hacer una reducción, esto es, reemplazar estos símbolos por el lado izquierdo de la producción
 - Si puede hacerse más de una reducción, éstas estarán ordenadas de alguna forma para determinar cuál aplicar primero
 - Si no es posible la reducción:
 - Desplazar el siguiente *token* a la pila de trabajo y volver al paso 2
 - b. Si se llega al final de la cadena w y no ha sido posible una reducción, retroceder hasta el último movimiento en el cual se hizo una reducción. Si hay otra reducción posible en este punto, hacerla y volver al paso 2. Si no la hay, seguir retrocediendo
 - c. Si se llega a un punto en que no se puede seguir avanzando, error
 - d. Si se ha leído toda la cadena w y en la pila de trabajo sólo está el axioma, entonces el *parse* será la secuencia de los números de las reglas utilizadas en las reducciones