

Procesadores de Lenguajes

Tema 4: Análisis sintáctico Epígrafe 4.2: Análisis ascendente

Curso 2007/2008

Antonio Pareja Lora

PP.LL. – Tema 4

Análisis Sintáctico Ascendente

Sesión 6 (I y II): Introducción al análisis sintáctico ascendente

Curso 2007/2008

Antonio Pareja Lora

PP.LL. – Tema 4

Sesión 6 (I): Introducción al an. sintáctico ascendente

- Conceptos clave
 - Algoritmo genérico de reducción-desplazamiento
 - Configuraciones inicial y final de la pila
 - Acciones semánticas del traductor a pila
- Análisis sintáctico ascendente con retroceso
- Análisis sintáctico ascendente sin retroceso
 - Análisis sintáctico por precedencia de operador
 - Tipología de gramáticas (de precedencia y de operador)
 - Relaciones de precedencia de operador
 - Análisis sintáctico por precedencia simple
 - Relaciones de precedencia simple

Curso 2007/2008

Antonio Pareja Lora

PP.LL. – Tema 4 – 3

Análisis sint. ascendente: conceptos clave (1)

- Configuraciones inicial y final de la pila del autómata:



- PIVOTE de una FORMA SENTENCIAL por la derecha ($\alpha \in (N \cup T)^+$):
 - Es una producción $A \rightarrow \beta$ y una posición dentro de α en la que puede encontrarse β y ser reemplazada por A para producir la anterior forma sentencial (por la derecha) en una derivación por la derecha de α
 - Representa el siguiente subárbol COMPLETO más a la izquierda en el árbol de análisis para el cual aún no se ha generado su padre en un análisis ascendente.

Curso 2007/2008

Antonio Pareja Lora

PP.LL. – Tema 4 – 4

Análisis sint. ascendente: conceptos clave (2)

- Acciones semánticas del autómata:
 - REDUCIR (SIMPLIFICAR):
 - Sustituye un pivote (β) en la pila (últimos k símbolos) por el antecedente (A) de la regla ($A \rightarrow \beta$) con cuyo consecuente equipara
 - DESPLAZAR:
 - Scan ($token$); pila.Push($token$);
 - ÉXITO (RECONOCER):
 - Se ha leído el fin de la cadena de entrada (\$)
 - Se ha alcanzado una configuración final de la pila
 - ERROR:
 - Caso 1:
 - La pila está vacía (no hay forma de continuar ascendiendo)
 - No se ha leído aún el fin de la cadena de entrada (\$)
 - Caso 2:
 - Se ha leído el fin de la cadena de entrada (\$)
 - No se ha alcanzado la configuración final de la pila

Análisis sint. ascendente: esquema del algoritmo

- Lectura de *tokens*: de izquierda a derecha
- Construcción del árbol: de las hojas a la raíz (= ascendiendo)
 - A ser posible, habrá que eliminar de la gramática las reglas borradoras ($A \rightarrow \lambda$) y los ciclos ($A \Rightarrow^+ A$)
- ESQUEMA:
 - MIENTRAS (NO(ÉXITO) Y NO(ERROR))
 - Intenta REDUCIR
 - Si no es posible, DESPLAZA

Análisis Sintáctico Ascendente

Análisis sintáctico ascendente con retroceso

Recordatorio: Tipos de analizadores sintácticos

- Analizador descendente:
 - Analizador descendente recursivo:
 - Con retroceso
 - Sin retroceso (predictivo)
 - Analizador descendente no recursivo predictivo (\equiv tabular):
 - Analizador $LL(K)$
 - Analizador $LL(1)$
- Analizador ascendente:
 - Analizador ascendente con retroceso
 - Analizador de gramáticas de precedencia de operador
 - Analizador de gramáticas de precedencia simple
 - Analizador $LR(K)$
 - Analizadores $LR(1)$
 - Analizadores $SLR(1)$

Análisis ascendente con retroceso: algoritmo

- ENTRADA:
 - Gramática $G = (N, T, S, P)$, sin ciclos y sin reglas borradoras, con sus reglas ordenadas por partes derechas y numeradas.
 - $w = w_1 \dots w_n \in T^*$, cadena de tokens que se va a analizar.
- SALIDA:
 - SI $w \in L(G)$: ÉXITO ; EN OTRO CASO: ERROR
- PROCESO:
 - pila.Vaciar (); pila.Push (\$); Scan (token);
 - Ensayar (pila, token);
- Ensayar (pila, token):
 - $\forall k \in \{1, \dots, \text{Cardinal}(P)\}$:
 - SI Equipara (Consecuente(Regla(k)) , pila.Ventana()):
 - Guarda el estado del análisis
 - Reduce (* Sustituye Consecuente(Regla(k)) en la pila por Antecedente(Regla(k)) *)
 - SI (pila.ConfiguraciónFinal() \wedge (token = \$)): RECONOCE (FIN)
EN OTRO CASO: Ensayar (pila, token);
 - Restaura estado anterior
 - SI (token \neq \$):
 - Guarda el estado del análisis
 - Desplaza(token); Ensayar (pila, token)
 - Restaura estado anterior
 - SI NO HA HABIDO RECONOCIMIENTO: ERROR

Bibliografía

- Aho, A. V.; Sethi, R.; Ullman, J. D.: *Compilers: Principles, Techniques and Tools*. Massachusetts: Addison-Wesley Publishing Company, 1986.
- Alfonseca Cubero, E.; Alfonseca Moreno, M.; Moriyón Salomón, R. Teoría de autómatas y lenguajes formales. Madrid: Mc-Graw-Hill/Interamericana de España, S.A.U., 2007.
- Grogono, P. Programación en Pascal. Wilmington, Delaware (EE.UU.):Addison-Wesley Iberoamericana, 1996.
- Sanchís Llorca, F. J. y Galán Pascual, C. Compiladores: Teoría y construcción. Madrid: Editorial Paraninfo, 1986.