

Traductores Push–Down para Gramáticas LL

Extensión de Autómatas

Universidad de Cantabria

Outline

- 1 El Problema
- 2 Modelando la Tabla del Traductor
- 3 Los Símbolos de Anticipación
- 4 La Detección de Errores

El Problema

Podemos resolver el problema de la palabra para lenguajes generados por gramáticas $LL(1)$ pero todavía no tenemos un modo eficiente de hallar un árbol de derivación.

El Problema

Nos interesa construir un traductor push-down que genere una derivación de una palabra. Como nuestro algoritmo es **descendente**, la derivación que obtendremos será a la izquierda.

El Problema

Recordad que como ha sido construida la tabla de análisis sintáctico. Lo que esta claro es que es fácil definir lo que hay que escribir en la cinta de escritura.

El Problema

Quedan dos problemas abiertos:

- ¿Que debe ir a la pila en cada iteración?
- ¿Como modelamos los símbolos de anticipación?

Tabla del Traductor

Nuestro traductor estará definido por los siguientes parámetros:

- El alfabeto de la cinta de input es el alfabeto Σ de la gramática dada.
- El alfabeto Δ de la cinta de output son los números naturales $\{1, \dots, N\}$ de una enumeración.
-
- El alfabeto de la pila Γ es la unión de los alfabetos V (conjunto de variables de la gramática), Σ (el alfabeto de la cinta de input) y el símbolo \S que jugará el papel de fondo de la pila.

$$\Gamma = V \cup \Sigma \cup \{\S\}.$$

Tabla del Traductor

La tabla será construida a partir de la gramática dada y cuyas filas están indicadas por los elementos de $V \cup \Sigma \cup \{\$\}$ y cuyas columnas están indicadas por los elementos de $\Sigma \cup \{\lambda\}$.

La diferencia es que en cada casilla hay un par, para definir que debe introducirse a la pila y que debe escribirse a la cinta.

Tabla del Traductor

En cada casilla, estará la parte derecha de la producción que aparezca en la tabla de análisis sintáctico $LL(1)$ y se escribirá el número de producción que corresponda.

Los Símbolos de Anticipación

La forma de utilizar los símbolos de anticipación es poniendo un estado para cada símbolo de anticipación. Las transiciones se definirán utilizando la tabla del traductor.

Los Símbolos de Anticipación

- El estado inicial q_0 solo tendrá una transición a si mismo que añadira a la pila **la variable inicial de la gramática**.
- Para cada elemento $a \in \Sigma$, definiremos un estado q_a .
- Solo desde el estado q_0 , si se lee el fondo de pila entonces se vacia la pila.

Los Símbolos de Anticipación

- Desde el estado inicial se ira al estado correspondiente. De esta forma se simula el conocimiento del símbolo de anticipación.
- Se retorna al estado q_0 cuando en la casilla de la tabla del traductor aparezca la palabra **pop** (y en estos casos no se escribe en la cinta de output).

Los Símbolos de Anticipación

Las transiciones de cada estado dependen de la tabla del traductor, ya que las filas de la tabla del traductor dan el top de la pila.

La Detección de Errores

Además de dar una derivación, en caso de que la palabra este en el lenguaje, queremos información sobre donde ha habido un error. Puede ser interesante dar información de, en que zona el error ha aparecido.

La Detección de Errores

La información será dada por **la configuración** en el momento que se detuvo la computación.

- El primer error puede ser detectado por el top de la pila.
- Si el top de la pila es un símbolo, esta claro que ese símbolo falta en la palabra, ¿pero que pasa si es una variable?
- Esto es importante para intentar que el compilador recupere el error y siga operando.