UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

Escuela Técnica Superior de Ingeniería Informática

Procesadores de Lenguajes

Tema 3

Parte II

Análisis Sintáctico Descendente

Javier Vélez Reyes
jvelez@lsi.uned.es

Objetivos del Tema

- Entender la técnica de análisis descendente
- Presentar los analizadores con retroceso
- Aprender el cálculo de los conjuntos de predicción
- Presentar las condiciones LL(1)
- Aprender la modificación de gramáticas no LL(1)
- Presentar los analizadores predictivos
  - Recursivos
  - Dirigidos por tabla

Javier Vélez Reyes jvelez@lsi.uned.es

Índice General

- Introducción
- Analizador con retroceso
- Técnica de análisis predictivo
- Analizadores predictivos

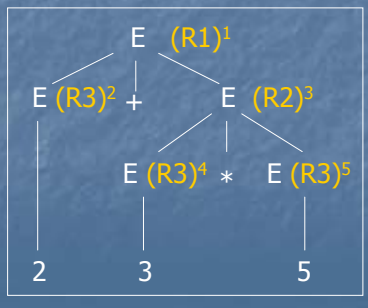
Javier Vélez Reyes jvelez@lsi.uned.es

Introducción

- Análisis sintáctico descendente
  - Partir del axioma de la gramática
  - Escoger reglas gramaticales
  - Hacer derivaciones por la izquierda
  - Procesar la entrada de izquierda a derecha
  - Obtener el árbol de análisis sintáctico o error

R1. E := E + E
R2. E := E * E
R3. E := **n**
R4. E := (E)

2 + 3 * 5



Javier Vélez Reyes jvelez@lsi.uned.es

Índice General

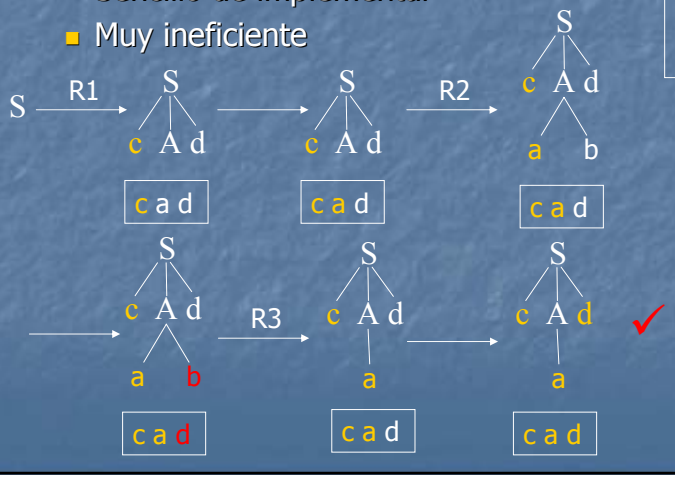
- Introducción
- Analizador con retroceso
- Técnica de análisis predictivo
- Analizadores predictivos

Javier Vélez Reyes jvelez@lsi.uned.es

Analizador con retroceso

- Analizador con retroceso
  - Usa retroceso para resolver la incertidumbre
  - Sencillo de implementar
  - Muy ineficiente

R1. S := c A d
R2. A := a b
R3. A := a



Javier Vélez Reyes jvelez@lsi.uned.es

Índice General

- Introducción
- Analizador con retroceso
- Técnica de análisis predictivo
  - Cálculo de los conjuntos de predicción
    - Conjunto Primero
    - Conjunto Siguiente
    - Conjunto Predicción
  - Condiciones LL(1)
  - Modificación de gramáticas no LL(1)
    - Eliminación de ambigüedad
    - Factorización por la izquierda
    - Eliminación de la recursividad
- Analizadores predictivos

Javier Vélez Reyes jvelez@lsi.uned.es

Técnicas de análisis predictivo

- Propósito
  - Crear un analizador descendente O(n)
  - Debe decidir qué regla aplicar según token
  - La gramática debe ser LL(1)
    - L. Análisis de izquierda a derecha
    - L. Derivaciones por la izquierda
    - 1. Un token permite decidir la regla de producción
  - Se elimina la recursividad

Javier Vélez Reyes jvelez@lsi.uned.es

Conjuntos de predicción

- Conjuntos de predicción
  - Ayudan a decidir qué regla utilizar en cada paso
- Construcción
  - Conjunto Primero PRIM (α)
  - Conjunto Siguiente SIG (A)
  - Regla

PRED (A := α) =

SI $\varepsilon \in$ PRIM (α) ENTONCES = PRIM (α) – { ε } $\cup$ SIG (A)

SINO = PRIM (α)

Javier Vélez Reyes jvelez@lsi.uned.es

Conjunto Primero

- Definición
  - Si α es una forma sentencial compuesta por una concatenación de símbolos PRIM (α) es el conjunto de terminales (o ε) que pueden aparecer iniciando las cadenas que pueden derivar de α
- Construcción

PRIM (ε) = {ε}
PRIM (a) = {a}
PRIM (A) = $\cup$ PRIM (α_i) Para todo A := α_i
PRIM (Aα) = PRIM (A) – {ε} $\cup$ PRIM$_\varepsilon$ (α)

Javier Vélez Reyes jvelez@lsi.uned.es

Conjunto Siguiente

- Definición
  - Si A es un símbolo no terminal de la gramática SIG (A) es el conjunto de terminales (y $) que pueden aparecer a continuación de A en alguna forma sentencial derivada del axioma
- Construcción

1.Inicialmente
   SIG (A) = { }
2. Si A es el axioma
   SIG (A) = SIG (A) $\cup$ {$}
3. Para cada regla B := $\alpha A \beta$
   SIG (A) = SIG (A) $\cup$ {PRIM (β) - ε}
4. Para cada regla B := αA o B:= $\alpha A \beta$ con $\varepsilon \in$ PRIM (β)
   SIG (A) = SIG (A) $\cup$ SIG (B)
5. Repetir 3 y 4 hasta que no se aumentar SIG (A)

Javier Vélez Reyes jvelez@lsi.uned.es

Condiciones LL(1)

- Determinación de la condición LL(1)
  - Se determina a partir de los conjuntos de predicción
  - Permite distinguir siempre la regla a aplicar

Los conjuntos de predicción de las reglas de producción de cada no terminal deben ser disjuntos entre sí

- Ejemplo

A := a b B	{a}
A := B b	{b, c}
B := b	{b}
B := c	{c}

Javier Vélez Reyes jvelez@lsi.uned.es

Modificación de gramáticas no LL(1)

- Condiciones necesarias para ser LL(1)
  - No ambigua
  - Factorizada por la izquierda
  - No recursiva a izquierdas

E := E + E
E := E - E
E := E * E
E := E / E
E := **n**
E := (E)

→

E := TE'
E' := +TE' | -TE' | ε
T := FT'
T' := *FT' | /FT' | ε
F := **n** | (E)

Javier Vélez Reyes jvelez@lsi.uned.es

Factorización por la izquierda

- Objetivo
  - Se trata de rescribir las producciones de la gramática con igual comienzo para retrasar la decisión hasta haber visto lo suficiente de la entrada como para elegir la opción correcta
- Procedimiento

$A := \alpha\beta_1 | \alpha\beta_2 | \ldots | \alpha\beta_n | \delta_i$

↓

$A := \alpha A' | \delta_i$
$A' := \beta_1 | \beta_2 | \ldots | \beta_n$

Javier Vélez Reyes jvelez@lsi.uned.es

Eliminación de la recursividad

- Tipos de recursividad
  - Directa. Una gramática G es recursiva si tiene alguna regla de producción que sea recursiva por la izquierda

$$A := A\alpha$$

  - Indirecta. Si, a partir de una forma sentencial que empieza por un no terminal se puede derivar una nueva forma no sentencial donde reaparece al principio el no terminal

$$A\alpha =>^* A\beta\alpha$$

Javier Vélez Reyes jvelez@lsi.uned.es

Eliminación de la recursividad

- Eliminación de la recursividad
  - Directa

$$A := A\alpha \mid \beta \longrightarrow \begin{array}{l} A := \beta A' \\ A' := \alpha A' \\ A' := \varepsilon \end{array}$$

  - Indirecta

```
Ordenar No terminales: A1, A2, … An
For i := 1 To  n Do
    For j := 1 To i – 1 Do
        Sustituir cada Ai := Aj β por Ai := α1 β | α2 β | αk β
        donde Aj := α1 | α2 | … | αk producciones actuales de Aj
        Eliminar la recursividad directa de Ai
```

Javier Vélez Reyes jvelez@lsi.uned.es

Índice General

- Introducción
- Analizador con retroceso
- Técnica de análisis predictivo
- Analizadores predictivos
  - Analizador descendente recursivo
  - Analizador descendente predictivo dirigido por tabla

Javier Vélez Reyes jvelez@lsi.uned.es

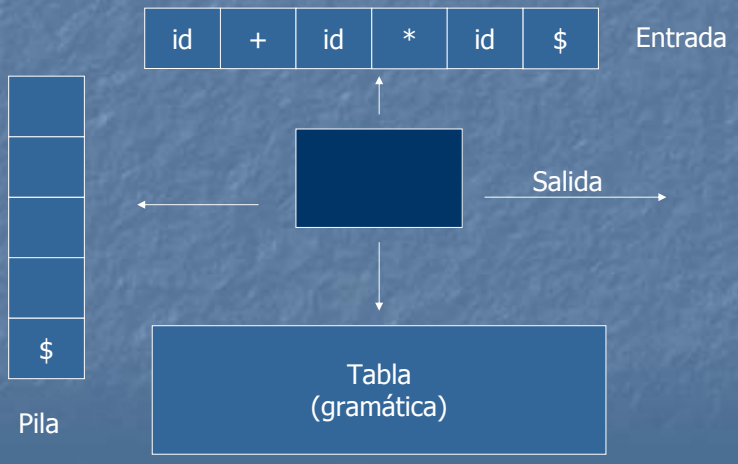
Analizador descendente Recursivo

- Elementos
  - Token
  - Preanálisis
- Funciones
  - Empareja (Token t)
  - Una por cada regla de producción
    - Recursivas
    - Siguen la parte derecha de la regla
    - Se utilizan los conjuntos de predicción para determinar la regla

Javier Vélez Reyes jvelez@lsi.uned.es

Analizador dirigido por tabla

- Utiliza una tabla para decidir



Javier Vélez Reyes jvelez@lsi.uned.es

Analizador dirigido por tabla

- Tabla
  - 1 fila por cada No terminal
  - 1 columna por cada terminal y $
  - Cada celda una regla de producción o error
- Construcción

Calcular los conjuntos de predicción de cada regla
Para cada A := α
    para cada a ϵ PRED (A := α) Tabla[A,a] = "A := α"
Rellenar los huecos vacíos con error

Javier Vélez Reyes jvelez@lsi.uned.es

Analizador dirigido por tabla

- Análisis de una cadena
  - Contruir una tabla con 3 columnas
    - Pila
    - Entrada procesada
    - Salida

PILA	ENTRADA	SALIDA
$E	n + n * n $	E := TE'
$E'T	n + n * n $	T := FT'
$E'T'F	n + n * n $	F := n
$E'T'n	n + n * n $	Emparejar(n)
$E'T'	+ n * n $	T' := ε

Javier Vélez Reyes jvelez@lsi.uned.es

Bibliografía

[AJO] AHO, SETHI, ULLMAN: *Compiladores: Principios, técnicas y herramientas*,: Addison-Wesley Iberoamericana, 1990

[GARRIDO] A. Garrido, J. Iñesta, F. Moreno y J. Pérez. 2002. *Diseño de compiladores*. Universidad de Alicante.