Procesadores de Lenguajes

Ingeniería Técnica superior de Ingeniería Informática

Departamento de Lenguajes y Sistemas informáticos

4

Análisis sintáctico

Analizadores descendentes

Javier Vélez Reyes
jvelez@lsi.uned.es
Departamento de Lenguajes Y Sistemas Informáticos
UNED

UNED

ETS de Ingeniería Informática

Objetivos

- Conocer en qué consiste el análisis descendente
- Conocer los tipos de analizadores descendentes que existen
- Entender en qué consiste el análisis descendente predictivo
- Aprender las condiciones necesarias sobre gramáticas para aplicar análisis predictivos
- Aprender a transformar gramáticas para aplicar análisis predictivos
- Entender el uso de los conjuntos de predicción
- Aprender a construir conjuntos de predicción
  - Aprender a calcular los conjuntos Primeros
  - Aprender a calcular los conjuntos Siguientes
- Aprender a construir y utilizar analizadores sintácticos predictivos
  - Aprender a construir analizadores predictivos recursivos
  - Aprender a construir analizadores predictivos dirigidos por tabla

Índice

¿Qué es el análisis descendente?

El análisis sintáctico descendente es una técnica de análisis sintáctico que intenta comprobar si una cadena x pertenece al lenguaje definido por una gramática L (G) aplicando los siguientes criterios

- Partir del axioma de la gramática
- Escoger reglas gramaticales estratégicamente
- Hacer derivaciones por la izquierda (Left Most Derivation)
- Procesar la entrada de izquierda a derecha
- Obtener el árbol de análisis sintáctico o error

R_1: E ::= E + E

R_2: E ::= E - E

R_3: E ::= E * E

R_4: E ::= E / E

R_5: E ::= (E)

R_6: E ::= n

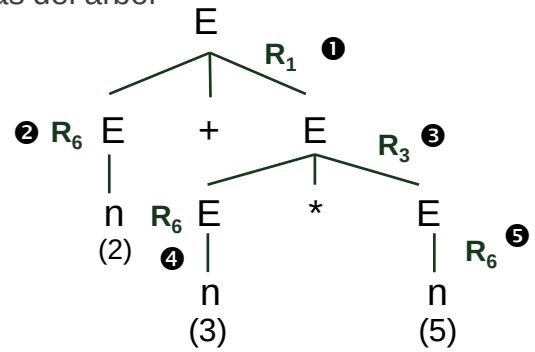
Cadena de derivación

En cada paso de derivación se transforma siempre el no terminal más a la izquierda de la forma de frase

$$E \rightarrow E + E \rightarrow$$
$$n + E \rightarrow$$
$$n + E * E \rightarrow$$
$$n + n * E \rightarrow$$
$$n + n * n$$

Árbol de análisis sintáctico

Se parte del axioma gramatical y se van aplicando reglas estratégicamente hasta alcanzar la frase X como nodos hojas del árbol



Análisis sintáctico. Analizadores descendentes

Introducción

Analizadores sintácticos descendentes

Analizadores deterministas

¿Como se selecciona el siguiente no terminal a derivar?

Analizadores descendentes

Se parte del axioma y se aplica una cadena de derivaciones para construir un árbol sintáctico

Analizadores ascendentes

Se parte de los terminales y se construye la inversa de una derivación para intentar alcanzar el axioma

¿Como se selecciona la regla de producción en cada paso de derivación?

Analizadores con retroceso

Se hace una búsqueda en profundidad con retroceso para garantizar que se encuentra la frase. Coste O (k^n)

Analizadores predictivos

Se determina qué regla aplicar a partir de un análisis de los primeros tokens a la entrada

Analizadores predictivos LL (1)

Determinan que regla de producción aplicar en cada paso en función de token que se encuentra en cada momento en la cabeza de lectura

Analizadores predictivos LL (k)

Determinan que regla de producción aplicar en cada paso en función de los k primeros tokens que se encuentra en cada momento en la cabeza de lectura

Análisis sintáctico descendente con retroceso

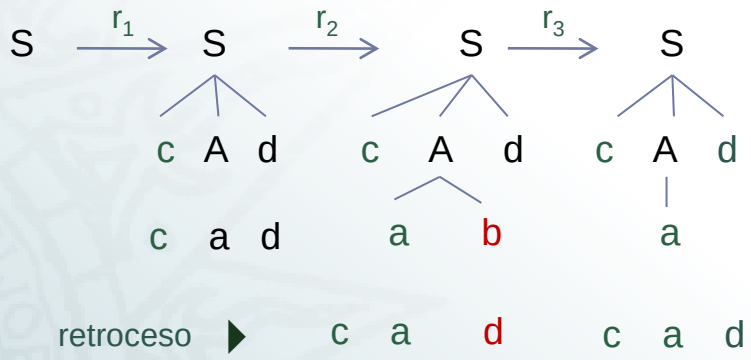
En el análisis sintáctico con retroceso se van probando una por una todas las reglas candidatas a ser aplicadas para construir el árbol de análisis sintáctico. Cuando una regla seleccionada falla, entonces se retrocede y se prueba con la siguiente regla

1. Se selecciona un nuevo no terminal a derivar por la estrategia Left Most Derivation
2. Se selecciona el conjunto de todas las reglas con antecedente igual al no terminal
3. Se ordena el conjunto de reglas arbitrariamente
4. Se selecciona una regla
    1. Se extrae la regla del conjunto
    2. Se aplica la derivación por dicha regla
    3. Si la regla encaja volver a 1 sino retroceder la derivación y volver a 4

r_1. S ::= c A d

r_2. A ::= a b

r_3. A ::= a



Análisis sintáctico descendente predictivo

Los analizadores sintácticos predictivos son capaces de decidir qué regla de producción aplicar a cada paso en función de los elementos terminales que se encuentran en la cabeza de lectura de la cadena de entrada. Como consecuencia se consigue un proceso de análisis con complejidad lineal O (n) con respecto al tamaño del problema. Estos analizadores son llamados genéricamente analizadores LL (K)

LL (k)

- k: Número de terminales consultados para determinar la regla de producción a aplicar
- L: Lectura de izquierda a derecha
- L: Derivación más a la izquierda (Left Most Derivatión)

Analizadores predictivos

Se determina qué regla aplicar a partir de un análisis de los primeros tokens a la entrada

Analizadores predictivos LL (1)

Determinan que regla de producción aplicar en cada paso en función de token que se encuentra en cada momento en la cabeza de lectura

Analizadores predictivos LL (k)

Determinan que regla de producción aplicar en cada paso en función de los k primeros tokens que se encuentra en cada momento en la cabeza de lectura

Análisis sintáctico descendente predictivo

Los analizadores sintácticos predictivos son capaces de decidir qué regla de producción aplicar a cada paso en función de los elementos terminales que se encuentran en la cabeza de lectura de la cadena de entrada. Como consecuencia se consigue un proceso de análisis con complejidad lineal O (n) con respecto al tamaño del problema. Estos analizadores son llamados genéricamente analizadores LL (K)

Conjuntos de predicción

Para aplicar el análisis descendente predictivo LL (1) es necesario asociar a cada regla de producción un conjunto de predicción. El conjunto de predicción de una regla está formado por la colección de todos los posibles terminales que es necesario encontrar en la cabeza de lectura para poder aplicar dicha regla

r_1. A ::= a b B	{ a }
r_2. A ::= B	{ b, c }
r_3. B ::= b	{ b }
r_4. B ::= c	{ c }

Entrada: **a** b c — a **b** c — a b **c**

Derivación: A → a b B — A → a b B — A → a b B → a b c

r_1. A ::= a b B	{ **a** }
r_2. A ::= B	{ b, c }
r_3. B ::= b	{ b }
r_4. B ::= c	{ c }

r_1. A ::= a b B	{ a }
r_2. A ::= B	{ b, c }
r_3. B ::= b	{ b }
r_4. B ::= c	{ **c** }

De todas las reglas candidatas para el no terminal en curso se escoge aquella que contiene en su conjunto de predicción el terminal a la entrada

Análisis sintáctico descendente predictivo

Los analizadores sintácticos predictivos son capaces de decidir qué regla de producción aplicar a cada paso en función de los elementos terminales que se encuentran en la cabeza de lectura de la cadena de entrada. Como consecuencia se consigue un proceso de análisis con complejidad lineal O (n) con respecto al tamaño del problema. Estos analizadores son llamados genéricamente analizadores LL (K)

$r_{1.}$	A ::= a b B	{ a }
$r_{2.}$	A ::= B	{ b, c }
$r_{3.}$	B ::= b	{ b }
$r_{4.}$	B ::= c	{ c }

Condiciones LL (1)

Para poder aplicar el análisis descendente predictivo LL (1) es necesario que se cumplan las condiciones LL (1) que garanticen la predictibilidad absoluta a la hora de seleccionar una regla de producción:

Para poder aplicar el análisis predictivo LL (1) es necesario que los conjuntos de predicción de todas las reglas con un mismo antecedente sean disjuntas entre sí

$r_{1.}$	A ::= a b B	{ a }
$r_{2.}$	A ::= B	{ b, c }
$r_{3.}$	B ::= b	{ b }
$r_{4.}$	B ::= c	{ c }

{ a } ∩ { b, c } = Ø LL (1) ✓

{ b } ∩ { c } = Ø LL (1) ✓

LL (1) ✓

Análisis sintáctico descendente predictivo

Los analizadores sintácticos predictivos son capaces de decidir qué regla de producción aplicar a cada paso en función de los elementos terminales que se encuentran en la cabeza de lectura de la cadena de entrada. Como consecuencia se consigue un proceso de análisis con complejidad lineal O (n) con respecto al tamaño del problema. Estos analizadores son llamados genéricamente analizadores LL (K)

Condiciones LL (1)

Para cumplir las condiciones LL (1) la gramática debe satisfacer necesariamente 3 requisitos:

- No ambigua
- Factorizada por la izquierda
- No recursiva a izquierdas

Consúltese el tema 3 para obtener una descripción acerca de cómo se puede transformar una gramática para que cumpla estas restricciones

Gramática de operadores canónica

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

❶

Gramática de operadores no ambigua

E ::= E + T | E – T | T
T ::= T * F | T / F | F
F ::= (E) | n

❷

Gramática de operadores LL (1)

E ::= TE’
E’ ::= + TE’ | – TE’ |
T ::= FT’
T’ ::= * FT’ | / FT’ |
F ::= (E) | n

❶ Eliminación de la ambigüedad
❷ Factorización por la izquierda

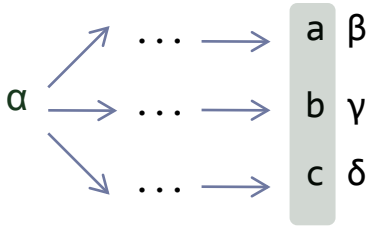
Análisis sintáctico descendente predictivo

Los analizadores sintácticos predictivos son capaces de decidir qué regla de producción aplicar a cada paso en función de los elementos terminales que se encuentran en la cabeza de lectura de la cadena de entrada. Como consecuencia se consigue un proceso de análisis con complejidad lineal O (n) con respecto al tamaño del problema. Estos analizadores son llamados genéricamente analizadores LL (K)

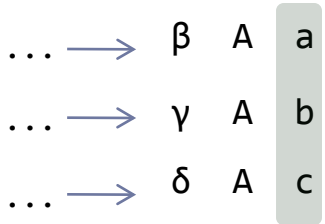
Construcción de los conjuntos de predicción

La construcción de los conjuntos de predicción de una regla A ::= α se apoya en el uso de dos conjuntos asociados respectivamente a la parte derecha e izquierda de la regla:

- Primeros (α) devuelve el conjunto de todos los terminales que se pueden encontrar a la cabeza de cualquier derivación de la frase α



- Siguientes (A) devuelve el conjunto de todos los terminales que se pueden encontrar siguiendo a A en cualquier derivación posible



Análisis sintáctico descendente predictivo

Los analizadores sintácticos predictivos son capaces de decidir qué regla de producción aplicar a cada paso en función de los elementos terminales que se encuentran en la cabeza de lectura de la cadena de entrada. Como consecuencia se consigue un proceso de análisis con complejidad lineal O (n) con respecto al tamaño del problema. Estos analizadores son llamados genéricamente analizadores LL (K)

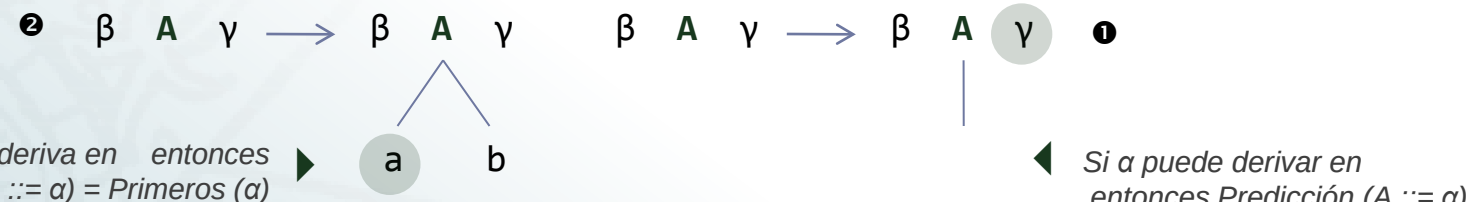
Construcción de los conjuntos de predicción

La construcción de los conjuntos de predicción de una regla A ::= α se apoya en el uso de dos conjuntos asociados respectivamente a la parte derecha e izquierda de la regla:

Predicción (A ::= α) =

❶ **Si** ϵ Primeros (α) **entonces** = Primeros (α) – { } $\cup$ Siguientes (A)

❷ **Sino** = Primeros(α)

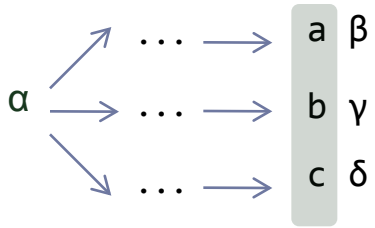


Análisis sintáctico descendente predictivo

Los analizadores sintácticos predictivos son capaces de decidir qué regla de producción aplicar a cada paso en función de los elementos terminales que se encuentran en la cabeza de lectura de la cadena de entrada. Como consecuencia se consigue un proceso de análisis con complejidad lineal O (n) con respecto al tamaño del problema. Estos analizadores son llamados genéricamente analizadores LL (K)

Conjuntos primeros

Si α es una forma de frase compuesta por una concatenación de símbolos Primeros (α) es el conjunto de terminales incluyendo potencialmente que pueden aparecer iniciando las cadenas que derivan de α



Primeros ( ) = { }

Primeros (a) = {a}

Primeros (A) = $\cup$ Primeros (α_i) Para todo A := α_i

Primeros (Aα) =

si Є Primeros (A), = Primeros (A) – { } U Primeros (α)

sino, = Primeros (A)

Ejercicio

Calcular el conjunto Primeros para todas las reglas de la siguiente gramática

E ::= TE’

E’ ::= + TE’ | – TE’ |

T ::= FT’

T’ ::= * FT’ | / FT’ |

F ::= (E) | n

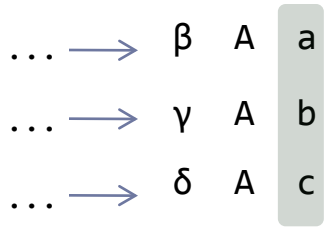
[1] $ representa el carácter fin de fichero

Análisis sintáctico descendente predictivo

Los analizadores sintácticos predictivos son capaces de decidir qué regla de producción aplicar a cada paso en función de los elementos terminales que se encuentran en la cabeza de lectura de la cadena de entrada. Como consecuencia se consigue un proceso de análisis con complejidad lineal O (n) con respecto al tamaño del problema. Estos analizadores son llamados genéricamente analizadores LL (K)

Conjuntos Siguientes

Si A es un símbolo no terminal de la gramática SIG (A) es el conjunto de terminales potencialmente incluyendo $[1] que pueden aparecer a continuación de A en alguna forma de frase derivada del axioma



1. Inicialmente

   Siguiente (A) = { }

2. Si A es el axioma

   Siguiente (A) = Siguiente (A) ∪ { $ }

3. Para cada regla B := α A β

   Siguiente (A) = Siguiente (A) ∪ { Primeros (β) – { ε } }

4. Para cada regla B := α A o B:= α A β con ε ϵ Primeros (β)

   Siguiente (A) = Siguiente (A) ∪ Siguiente (B)

5. Repetir 3 y 4 hasta que no se aumentar Siguiente (A)

Ejercicio

Calcular el conjunto Siguientes para todas las reglas de la siguiente gramática

E ::= TE'

E' ::= + TE' | – TE' |

T ::= FT'

T' ::= * FT' | / FT' |

F ::= (E) | n

Análisis sintáctico descendente predictivo

Los analizadores sintácticos predictivos son capaces de decidir qué regla de producción aplicar a cada paso en función de los elementos terminales que se encuentran en la cabeza de lectura de la cadena de entrada. Como consecuencia se consigue un proceso de análisis con complejidad lineal O (n) con respecto al tamaño del problema. Estos analizadores son llamados genéricamente analizadores LL (K)

Ejercicios

Comprobar el cumplimiento de las condiciones LL (1) para las siguientes gramáticas

E ::= TE'

E' ::= + TE' | – TE' |

T ::= FT'

T' ::= * FT' | / FT' |

F ::= (E) | n

S ::= A a | b

A ::= A c | S d

S ::= A a | b

A ::= A c | S d |

S ::= A B | s

A ::= a S c | e B f |

B ::= b A d |

A ::= A a | B C D

B ::= b |

C :: = c |

D ::= d | Ce

A ::= A a | B C D

B ::= b |

C :: = c |

D ::= d | Ce |

Analizadores sintácticos descendentes predictivos

Los analizadores sintácticos descendentes predictivos son autómatas a pila deterministas que reconocen las frases de un lenguaje por la estrategia de vaciado de pila. En esta sección estudiaremos dos tipos distintos de analizadores descendentes predictivos que se diferencian en la forma de implementar el autómata y dar soporte a la pila

Analizadores descendentes predictivos

Tratarán de encontrar la cadena de entrada partiendo del axioma y aplicando pasos de derivación. La selección de reglas está dirigida por los conjuntos de predicción

Analizadores descendentes recursivos

Para dar soporte a la pila se utilizan las capacidades de recursión del lenguaje. La gramática queda expresada a través de llamadas explicitas a distintas funciones asociadas a los no terminales

Analizadores descendentes dirigidos por tabla

La pila se da soporte a través de una representación explicita. La selección de reglas se realiza con ayuda de una tabla que representa la gramática del lenguaje

Analizadores sintácticos descendentes predictivos

Los analizadores sintácticos descendentes predictivos son autómatas a pila deterministas que reconocen las frases de un lenguaje por la estrategia de vaciado de pila. En esta sección estudiaremos dos tipos distintos de analizadores descendentes predictivos que se diferencian en la forma de implementar el autómata y dar soporte a la pila

Analizadores descendentes recursivos

Para dar soporte a la pila se utilizan las capacidades de recursión del lenguaje. La gramática queda expresada a través de llamadas explicitas a distintas funciones asociadas a los no terminales

Ejemplo

E ::= TE'

E' ::= + TE' | – TE' |

T ::= FT'

T' ::= * FT' | / FT' |

F ::= (E) | n

```
void match (Terminal token)  {
   if ( cabeza == token ) cabeza = scanner.nextToken ();
   else throw new SyntaxError ();
}
void e () {
   t ();
   ePrima ();
}
void ePrima () {
  if ( ! (cabeza in { ‘+’, ‘-’ } ) throw new SyntaxError ();
   switch ( cabeza ) {
     case ‘+’ : match (‘+’);
                t ();
                ePrima ();
                break;
```

Analizadores sintácticos descendentes predictivos

Los analizadores sintácticos descendentes predictivos son autómatas a pila deterministas que reconocen las frases de un lenguaje por la estrategia de vaciado de pila. En esta sección estudiaremos dos tipos distintos de analizadores descendentes predictivos que se diferencian en la forma de implementar el autómata y dar soporte a la pila

Analizadores descendentes recursivos

Para dar soporte a la pila se utilizan las capacidades de recursión del lenguaje. La gramática queda expresada a través de llamadas explicitas a distintas funciones asociadas a los no terminales

Ejemplo

E ::= TE'

E' ::= + TE' | – TE' |

T ::= FT'

T' ::= * FT' | / FT' |

F ::= (E) | n

```
     case '-' : match ('-');
                t ();
                ePrima ();
                break;
     default  : break;
  }
}
void t () {
   f ();
   tPrima ();
}
void tPrima () {
  if ( ! (cabeza in { '*', '/', '+', '-' } ) throw new
      SyntaxError ();
   switch ( cabeza ) {
```

Analizadores sintácticos descendentes predictivos

Los analizadores sintácticos descendentes predictivos son autómatas a pila deterministas que reconocen las frases de un lenguaje por la estrategia de vaciado de pila. En esta sección estudiaremos dos tipos distintos de analizadores descendentes predictivos que se diferencian en la forma de implementar el autómata y dar soporte a la pila

Analizadores descendentes recursivos

Para dar soporte a la pila se utilizan las capacidades de recursión del lenguaje. La gramática queda expresada a través de llamadas explicitas a distintas funciones asociadas a los no terminales

Ejemplo

E ::= TE'

E' ::= + TE' | – TE' |

T ::= FT'

T' ::= * FT' | / FT' |

F ::= (E) | n

```
    case '*' : match ('*');
               f ();
               tPrima ();
               break ();
     case '/' : match ('/');
               f ();
               tPrima ();
               break ();
     default  : break;
   }
}
void f () {
  switch ( cabeza )
     case '(' : e ();
               match (')'); break;
```

Analizadores sintácticos descendentes predictivos

Los analizadores sintácticos descendentes predictivos son autómatas a pila deterministas que reconocen las frases de un lenguaje por la estrategia de vaciado de pila. En esta sección estudiaremos dos tipos distintos de analizadores descendentes predictivos que se diferencian en la forma de implementar el autómata y dar soporte a la pila

Analizadores descendentes recursivos

Para dar soporte a la pila se utilizan las capacidades de recursión del lenguaje. La gramática queda expresada a través de llamadas explicitas a distintas funciones asociadas a los no terminales

Ejemplo

E ::= TE'

E' ::= + TE' | – TE' |

T ::= FT'

T' ::= * FT' | / FT' |

F ::= (E) | n

```
     case n   : match (n);
                break;
     default  : throw new SyntaxError ();
   }
}
Token cabeza;
Scanner scanner = new Scanner (file);
main () {
   cabeza = scanner.nextToken ();
   e ();
}
```

Analizadores sintácticos descendentes predictivos

Los analizadores sintácticos descendentes predictivos son autómatas a pila deterministas que reconocen las frases de un lenguaje por la estrategia de vaciado de pila. En esta sección estudiaremos dos tipos distintos de analizadores descendentes predictivos que se diferencian en la forma de implementar el autómata y dar soporte a la pila

Analizadores descendentes recursivos

Para dar soporte a la pila se utilizan las capacidades de recursión del lenguaje. La gramática queda expresada a través de llamadas explicitas a distintas funciones asociadas a los no terminales

Ejemplo

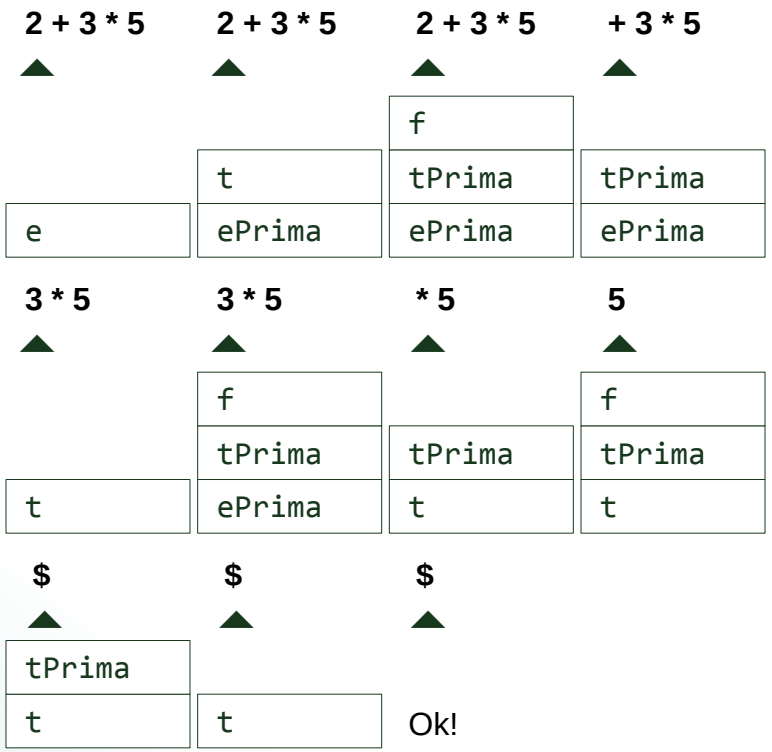
E ::= TE'

E' ::= + TE' | – TE' |

T ::= FT'

T' ::= * FT' | / FT' |

F ::= (E) | n



Analizadores sintácticos descendentes predictivos

Los analizadores sintácticos descendentes predictivos son autómatas a pila deterministas que reconocen las frases de un lenguaje por la estrategia de vaciado de pila. En esta sección estudiaremos dos tipos distintos de analizadores descendentes predictivos que se diferencian en la forma de implementar el autómata y dar soporte a la pila

Analizadores descendentes recursivos

Para dar soporte a la pila se utilizan las capacidades de recursión del lenguaje. La gramática queda expresada a través de llamadas explicitas a distintas funciones asociadas a los no terminales

Ventajas	Inconvenientes
› Fácil de entender e interpretar	› Pesado de desarrollar y mantener
› Gramática explícitamente representada	› Específico para cada lenguaje
› Cada función representa un no terminal	› Coste computacional (recursividad)
› Adecuado para gramáticas sencillas	› No da soporte a gran número de lenguajes

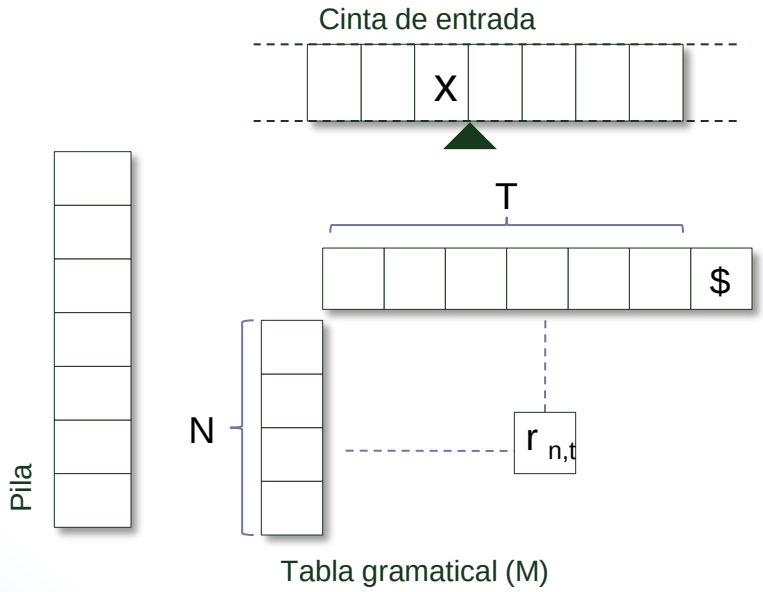
Analizadores sintácticos descendentes predictivos

Los analizadores sintácticos descendentes predictivos son autómatas a pila deterministas que reconocen las frases de un lenguaje por la estrategia de vaciado de pila. En esta sección estudiaremos dos tipos distintos de analizadores descendentes predictivos que se diferencian en la forma de implementar el autómata y dar soporte a la pila

Analizadores descendentes dirigidos por tabla

Los analizadores descendentes dirigidos por tabla están constituidos por dos elementos que se utilizan para llevar a cabo el proceso de análisis sintáctico

- Una pila, donde se almacenan símbolos gramaticales
- Una tabla de doble entrada que representa la gramática
  - En columnas el conjunto de terminales T U { $ }
  - En filas el conjunto de no terminales N
  - M (t, n) = la regla que debe aplicarse



Analizadores sintácticos descendentes predictivos

Los analizadores sintácticos descendentes predictivos son autómatas a pila deterministas que reconocen las frases de un lenguaje por la estrategia de vaciado de pila. En esta sección estudiaremos dos tipos distintos de analizadores descendentes predictivos que se diferencian en la forma de implementar el autómata y dar soporte a la pila

Analizadores descendentes dirigidos por tabla

El algoritmo de análisis descendente predictivo para este tipo de analizadores consiste en ir consultando la tabla para saber que regla aplicar y apoyarse en la pila asociada:

```
cabeza = <<primer terminal de w$>>
pila = [ $, Axioma ]
do {
   p = pila.cima ()
   a = scanner.nextToken ()
   if ( p in T U { $ } )
     id ( cabeza == a ) {
      cabeza = scanner.nextToken ()
      pila.pop ()
   } else new throw SyntaxError ();
```

```
   else {
     if ( M (p, a) == X ::= Y1 ... Yk ) {
       pop ()
       for (i = k; i >= 1; i++) push (Yi)
     } else throw new SyntaxError ();
   }
while (p != '$')
```

Analizadores sintácticos descendentes predictivos

Los analizadores sintácticos descendentes predictivos son autómatas a pila deterministas que reconocen las frases de un lenguaje por la estrategia de vaciado de pila. En esta sección estudiaremos dos tipos distintos de analizadores descendentes predictivos que se diferencian en la forma de implementar el autómata y dar soporte a la pila

Analizadores descendentes dirigidos por tabla

El algoritmo de análisis descendente predictivo para este tipo de analizadores consiste en ir consultando la tabla para saber que regla aplicar y apoyarse en la pila asociada:

	+	-	*	/	(	)	n	$
E	-	-	-	-	TE’	-	TE’	-
E’	+TE’	-TE’	-	-	-		-	
T	-	-	-	-	FT’	-	FT’	-
T’			*FT’	/FT’	-		-	
F	-	-	-	-	(E)	-	n	-

Ejemplo

E ::= TE’

E’ ::= + TE’ | – TE’ |

T ::= FT’

T’ ::= * FT’ | / FT’ |

F ::= (E) | n

Pila	Entrada	Acción
$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 $	match (n)
$E’T’	+ n * n $	T’ ::=
$E’	+ n * n $	E’ ::= +TE’

Analizadores sintácticos descendentes predictivos

Los analizadores sintácticos descendentes predictivos son autómatas a pila deterministas que reconocen las frases de un lenguaje por la estrategia de vaciado de pila. En esta sección estudiaremos dos tipos distintos de analizadores descendentes predictivos que se diferencian en la forma de implementar el autómata y dar soporte a la pila

Analizadores descendentes dirigidos por tabla

El algoritmo de análisis descendente predictivo para este tipo de analizadores consiste en ir consultando la tabla para saber que regla aplicar y apoyarse en la pila asociada:

	+	-	*	/	(	)	n	$
E	-	-	-	-	TE’	-	TE’	-
E’	+TE’	-TE’	-	-	-		-	
T	-	-	-	-	FT’	-	FT’	-
T’			*FT’	/FT’	-		-	
F	-	-	-	-	(E)	-	n	-

Ejemplo

E ::= TE’

E’ ::= + TE’ | – TE’ |

T ::= FT’

T’ ::= * FT’ | / FT’ |

F ::= (E) | n

Pila	Entrada	Acción
$E’T+	+ n * n $	match (+)
$E’T	n * n $	T ::= FT’
$E’T’F	n * n $	F ::= n
$E’T’n	n * n $	match (n)
$E’T’	* n $	T’ ::= *FT’
$E’T’F*	* n $	match (*)

Analizadores sintácticos descendentes predictivos

Los analizadores sintácticos descendentes predictivos son autómatas a pila deterministas que reconocen las frases de un lenguaje por la estrategia de vaciado de pila. En esta sección estudiaremos dos tipos distintos de analizadores descendentes predictivos que se diferencian en la forma de implementar el autómata y dar soporte a la pila

Analizadores descendentes dirigidos por tabla

El algoritmo de análisis descendente predictivo para este tipo de analizadores consiste en ir consultando la tabla para saber que regla aplicar y apoyarse en la pila asociada:

	+	-	*	/	(	)	n	$
E	-	-	-	-	TE’	-	TE’	-
E’	+TE’	-TE’	-	-	-		-	
T	-	-	-	-	FT’	-	FT’	-
T’			*FT’	/FT’	-		-	
F	-	-	-	-	(E)	-	n	-

Ejemplo

E ::= TE’

E’ ::= + TE’ | – TE’ |

T ::= FT’

T’ ::= * FT’ | / FT’ |

F ::= (E) | n

Pila	Entrada	Acción
$E’T’F	n $	F ::= n
$E’T’n	n $	match (n)
$E’T’	$	T’ ::=
$E’	$	E’ ::=
$	$	Ok!

Analizadores sintácticos descendentes predictivos

Los analizadores sintácticos descendentes predictivos son autómatas a pila deterministas que reconocen las frases de un lenguaje por la estrategia de vaciado de pila. En esta sección estudiaremos dos tipos distintos de analizadores descendentes predictivos que se diferencian en la forma de implementar el autómata y dar soporte a la pila

Analizadores descendentes dirigidos por tabla

El algoritmo de análisis descendente predictivo para este tipo de analizadores consiste en ir consultando la tabla para saber que regla aplicar y apoyarse en la pila asociada

Ventajas	Inconvenientes
› Ágil de desarrollar y mantener	› Más complejo de interpretar
› Representación tabular de las reglas	› Gramática no explícitamente representada
› Bajo coste computacional (no recursividad)	› Reglas distribuidas en la tabla
› Adecuado para gramáticas sencillas	› No da soporte a gran número de lenguajes

Gestión de errores

La labor de un analizador sintáctico no se limita exclusivamente a reconocer que una frase pertenece al lenguaje generado por una gramática. Adicionalmente, en el caso de que existan errores sintácticos (o léxicos no reconocidos en la fase de análisis léxico), el analizador debe ser capaz de reconocer errores y emitir mensajes de información asociados

I. Identificación de errores

El reconocimiento de la existencia de un error debe realizarse lo más temprano posible dentro del proceso de análisis sintáctico. No hacerlo así puede provocar que el analizador abandone el contexto sintáctico donde éste se ha producido lo cual provoca mensajes de error inadecuados

II. Localización de errores

La gestión de errores conlleva la localización de los mismos dentro del código fuente de la manera más precisa posible. El mensaje de error generado deberá contener información detallada y correcta acerca del número de fila y columna donde se encuentra el error

III. Emisión de mensajes

La existencia de errores se comunica al programación a través de la emisión de mensajes de error. Estos mensajes deben ser lo más específicos posibles de manera que el propio mensaje de información acerca de que acción correctiva debe ser aplicada

II. Recuperación de errores

El analizador sintáctico no debe abortar el proceso de compilación al encontrar un error sino que debe escapar del contexto sintáctico de error para avanzar a una nueva situación estable. Esta técnica se reconoce por el nombre recuperación de errores

Gestión de errores

La labor de un analizador sintáctico no se limita exclusivamente a reconocer que una frase pertenece al lenguaje generado por una gramática. Adicionalmente, en el caso de que existan errores sintácticos (o léxicos no reconocidos en la fase de análisis léxico), el analizador debe ser capaz de reconocer errores y emitir mensajes de información asociados

Recuperación de errores en análisis sintáctico descendente predictivo

Al identificar un error sintáctico se entra en estado de pánico y el analizador empieza a consumir tokens hasta encontrar un contexto de compilación estable, escapando del error. La identificación del contexto estable se dirige por conjuntos de sincronización asociados a cada tipo de error. El compilador consumirá tokens hasta encontrar un token perteneciente al conjunto de sincronización. El conjunto de sincronización suele construirse a partir de Primeros y Siguientes

```
Wile (a>b) a++ ;
```

Recuperación de errores en análisis sintáctico descendente dirigido por tabla

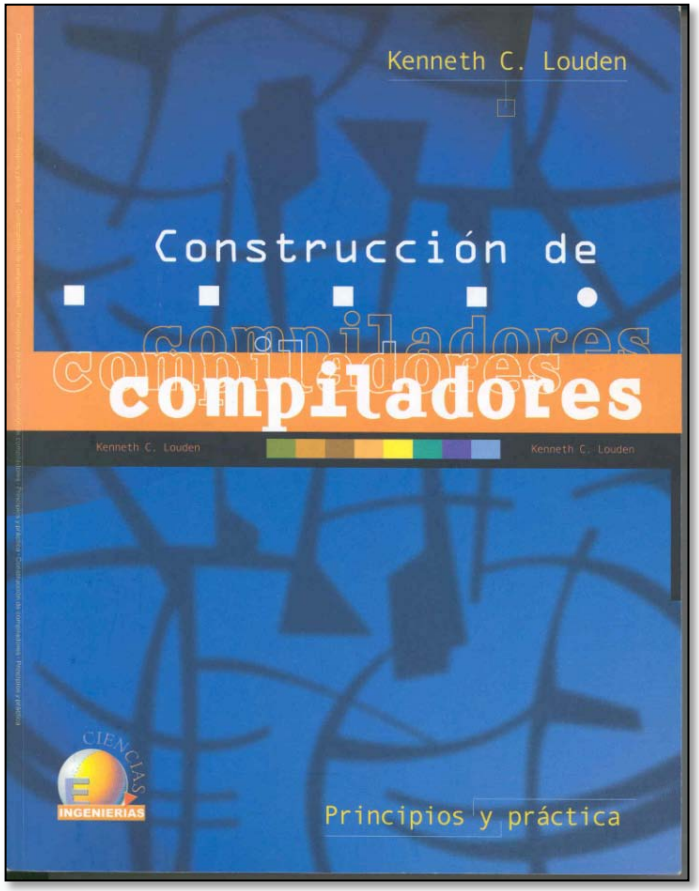
El proceso de recuperación de errores es esencialmente el mismo que en los analizadores descendentes recursivos. La tabla contiene ahora información para gestionar los errores. En las celdas de error pueden aparecer 3 acciones de recuperación

- Extraer el no terminal de la cima de la pila
- Consumir tokens en estado de pánico dirigido por conjunto de sincronización
- Insertar un nuevo terminal en la pila

Material de estudio

Bibliografía básica

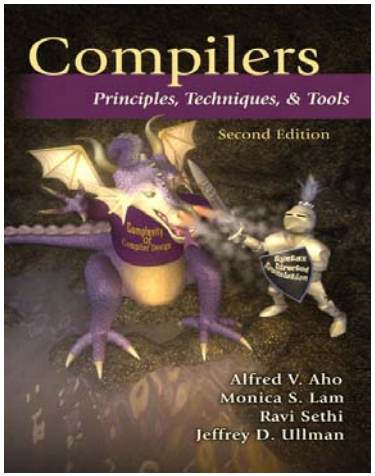
Construcción de compiladores: principios y práctica
Kenneth C. Louden International Thomson Editores, 2004 ISBN 970-686-299-4



Material de estudio

Bibliografía complementaria

Compiladores: Principios, técnicas y herramientas.
Segunda Edición Aho, Lam, Sethi, Ullman
Addison – Wesley, Pearson Educación, México 2008



Diseño de compiladores. A. Garrido, J. Iñesta, F. Moreno y J. Pérez. 2002. Edita Universidad de Alicante