

Desarrollo manual de un analizador sintáctico

FACULTAD DE INFORMÁTICA



UNIVERSIDAD
COMPLUTENSE
MADRID

Asignatura de Procesadores de Lenguajes

Curso 2021-2022

Grupo 20

Adrián Martín Tiscar
Gema Blanco Núñez

1. Especificación sintáctica

Programa \rightarrow Decs '&&'Instrucciones

Decs \rightarrow Decs ';' Dec

Decs \rightarrow Dec

Dec \rightarrow int id

Dec \rightarrow bool id

Dec \rightarrow real id

Instrucciones \rightarrow Instrucciones ';' Inst

Instrucciones \rightarrow Inst

Inst \rightarrow Id '=' Expresion

Expresion \rightarrow E0

E0 \rightarrow E1 '+' E0

E0 \rightarrow E1 '-' E1

E0 \rightarrow E1

E1 \rightarrow E1 OP1 E2

E1 \rightarrow E2

E2 \rightarrow E2 OP2 E3

E2 \rightarrow E3

E3 \rightarrow E4 OP3 E4

E3 \rightarrow E4

E4 \rightarrow '-' E5

E4 \rightarrow not E4

E4 \rightarrow E5

E5 \rightarrow numeroEntero

E5 \rightarrow numeroReal

E5 \rightarrow id

E5 \rightarrow true

E5 \rightarrow false

E5 \rightarrow (E0)

OP1 \rightarrow and

OP1 \rightarrow or

OP2 \rightarrow '>'

OP2 \rightarrow '>='

$OP2 \rightarrow '<'$
 $OP2 \rightarrow '<='$
 $OP2 \rightarrow '=='$
 $OP2 \rightarrow '!='$

$OP3 \rightarrow '*'$
 $OP3 \rightarrow '/'$

2. Acondicionamiento de la gramática

Con el fin de construir un analizador sintáctico descendente predictivo recursivo se aplican dos transformaciones:

- Eliminación de factores comunes
- Eliminación de recursión a izquierdas

En la siguiente tabla se muestran las reglas sin acondicionar y sus respectivas transformaciones.

| Eliminación de recursión a izquierdas | |
|---|---|
| Sin acondicionar | Acondicionada |
| $Decs \rightarrow Decs \text{ ',' } Dec$ $Decs \rightarrow Dec$ | $Decs \rightarrow Dec RDecs$ $RDecs \rightarrow \text{ ',' } Dec RDecs$ $RDecs \rightarrow \epsilon$ |
| $Instrucciones \rightarrow Instrucciones \text{ ',' } Inst$ $Instrucciones \rightarrow Inst$ | $Instrucciones \rightarrow Inst RInstrucciones$ $RInstrucciones \rightarrow \text{ ',' } Inst RInstrucciones$ $RInstrucciones \rightarrow \epsilon$ |
| $E1 \rightarrow E1 OP1 E2$ $E1 \rightarrow E2$ | $E1 \rightarrow E2 RE1$ $RE1 \rightarrow OP1 E2 RE1$ $RE1 \rightarrow \epsilon$ |
| $E2 \rightarrow E2 OP2 E3$ $E2 \rightarrow E3$ | $E2 \rightarrow E3 RE2$ $RE2 \rightarrow OP2 E3 RE2$ $RE2 \rightarrow \epsilon$ |
| Eliminación de factores comunes | |
| Sin acondicionar | Acondicionada |
| $E0 \rightarrow E1 \text{ '+' } E0$ $E0 \rightarrow E1 \text{ '-' } E1$ $E0 \rightarrow E1$ | $E0 \rightarrow E1 RE0$ $RE0 \rightarrow \text{ '+' } E0$ $RE0 \rightarrow \text{ '-' } E1$ $RE0 \rightarrow \epsilon$ |
| $E3 \rightarrow E4 OP3 E4$ | $E3 \rightarrow E4 RE3$ |

| | |
|--|---|
| $E3 \rightarrow E4$ | $RE3 \rightarrow OP3 E4$ $RE3 \rightarrow \epsilon$ |
| $OP2 \rightarrow '<'$ $OP2 \rightarrow '<='$ $OP2 \rightarrow '>'$ $OP2 \rightarrow '>='$ | $OP2 \rightarrow '<' ROP2$ $OP2 \rightarrow '>' ROP2$ $ROP2 \rightarrow '='$ $ROP2 \rightarrow \epsilon$ |

Aplicando las anteriores transformaciones. obtenemos la siguiente gramática resultante acondicionada:

Programa \rightarrow Decs '&&' Instrucciones

Decs \rightarrow Dec RDecs

RDecs \rightarrow ',' Dec RDecs

RDecs $\rightarrow \epsilon$

Dec \rightarrow int id

Dec \rightarrow bool id

Dec \rightarrow real id

Instrucciones \rightarrow Inst RInstrucciones

RInstrucciones \rightarrow ',' Inst RInstrucciones

RInstrucciones $\rightarrow \epsilon$

Inst \rightarrow Id '=' Expresion

Expresion \rightarrow E0

E0 \rightarrow E1 RE0

RE0 \rightarrow '+' E0

RE0 \rightarrow '-' E1

RE0 $\rightarrow \epsilon$

E1 \rightarrow E2 RE1

RE1 \rightarrow OP1 E2 RE1

RE1 $\rightarrow \epsilon$

E2 \rightarrow E3 RE2

RE2 \rightarrow OP2 E3 RE2

RE2 $\rightarrow \epsilon$

E3 \rightarrow E4 RE3

RE3 \rightarrow OP3 E4

RE3 $\rightarrow \epsilon$

$E4 \rightarrow '-' E5$
 $E4 \rightarrow \text{not } E4$
 $E4 \rightarrow E5$

$E5 \rightarrow \text{numeroEntero}$
 $E5 \rightarrow \text{numeroReal}$
 $E5 \rightarrow \text{id}$
 $E5 \rightarrow \text{true}$
 $E5 \rightarrow \text{false}$
 $E5 \rightarrow (E0)$

$OP1 \rightarrow \text{and}$
 $OP1 \rightarrow \text{or}$
 $OP2 \rightarrow '<' ROP2$
 $OP2 \rightarrow '>' ROP2$
 $ROP2 \rightarrow '='$
 $ROP2 \rightarrow \epsilon$
 $OP2 \rightarrow '=='$
 $OP2 \rightarrow '!='$
 $OP3 \rightarrow '*'$
 $OP3 \rightarrow '/'$

3. Directores de la gramática acondicionada

Mediante la herramienta de análisis de gramáticas Proletool se han obtenido los siguientes directores para cada regla de la gramática:

| Regla | Directores |
|--|-----------------------|
| Programa \rightarrow Decs '&&' Instrucciones | {'bool','int','real'} |
| Decs \rightarrow Dec RDecs | {'bool','int','real'} |
| RDecs \rightarrow ';' Dec RDecs | {';'} |
| RDecs $\rightarrow \epsilon$ | {'&'} |
| Dec \rightarrow int id | {'int'} |
| Dec \rightarrow bool id | {'bool'} |
| Dec \rightarrow real id | {'real'} |
| Instrucciones \rightarrow Inst RInstrucciones | {'id'} |

| | |
|--|--|
| RInstrucciones \rightarrow ';' Inst RInstrucciones | {';'} |
| RInstrucciones $\rightarrow \epsilon$ | {'⊢'} |
| Inst \rightarrow id igual Expresion | {'id'} |
| Expresion \rightarrow E0 | {'(', '-', 'false', 'true', 'id', 'not', 'numeroEntero', 'numeroReal'} |
| E0 \rightarrow E1 RE0 | {'(', '-', 'false', 'true', 'id', 'not', 'numeroEntero', 'numeroReal'} |
| RE0 \rightarrow '+' E0 | {'+'} |
| RE0 \rightarrow '-' E1 | {'-'} |
| RE0 $\rightarrow \epsilon$ | {')', ',', '⊢'} |
| E1 \rightarrow E2 RE1 | {'(', '-', 'false', 'true', 'id', 'not', 'numeroEntero', 'numeroReal'} |
| RE1 \rightarrow OP1 E2 RE1 | {'and', 'or'} |
| RE1 $\rightarrow \epsilon$ | {')', '+', '-', ',', 'and', 'or', '⊢'} |
| E2 \rightarrow E3 RE2 | {'(', '-', 'false', 'true', 'id', 'not', 'numeroEntero', 'numeroReal'} |
| RE2 \rightarrow OP2 E3 RE2 | {'!', '<', '>', '='} |
| RE2 $\rightarrow \epsilon$ | {')', '+', '-', ',', 'and', 'or', '⊢'} |
| E3 \rightarrow E4 RE3 | {'(', '-', 'false', 'true', 'id', 'not', 'numeroEntero', 'numeroReal'} |
| RE3 \rightarrow OP3 E4 | {'*', '/'} |
| RE3 $\rightarrow \epsilon$ | {'!', ')', '+', '-', ',', '<', '>', '=', 'and', 'or', '⊢'} |
| E4 \rightarrow - E5 | {'-'} |
| E4 \rightarrow not E4 | {'not'} |
| E4 \rightarrow E5 | {'(', 'false', 'true', 'id', 'numeroEntero', 'numeroReal'} |
| E5 \rightarrow numeroEntero | {'numeroEntero'} |
| E5 \rightarrow numeroReal | {'numeroReal'} |
| E5 \rightarrow id | {'id'} |
| E5 \rightarrow true | {'true'} |
| E5 \rightarrow false | {'false'} |
| E5 \rightarrow (E0) | {'('} |

| | |
|-----------------------------|--|
| OP1 \rightarrow and | {'and'} |
| OP1 \rightarrow or | {'or'} |
| OP2 \rightarrow '<' ROP2 | {'>'} |
| OP2 \rightarrow '>' ROP2 | {'<'} |
| ROP2 \rightarrow '=' | {'='} |
| ROP2 $\rightarrow \epsilon$ | {'(', '-', 'true', 'false', 'id', 'not', 'numeroEntero', 'numeroReal'} |
| OP2 \rightarrow '==' | {'='} |
| OP2 \rightarrow '!= | {'!'} |
| OP3 \rightarrow '*' | {'*'} |
| OP3 \rightarrow '/' | {'/'} |

4. Símbolos para diagnóstico de errores

| No terminales | Símbolos de diagnóstico | Justificación |
|----------------|-------------------------|--|
| Programa | {'bool', 'int', 'real'} | Son símbolos de diagnóstico porque son terminales. |
| Decs | {'bool', 'int', 'real'} | Son símbolos de diagnóstico porque son terminales. |
| RDecs | {',', '&'} | ',' es símbolo de diagnóstico porque es un primero. |
| | | '&' es símbolo de diagnóstico porque a pesar de ser un siguiente en todas las apariciones de RDecs va seguido de '&' en el contexto de la regla <i>Programa \rightarrow Decs '&&' Instrucciones</i> . |
| Dec | {'bool', 'int', 'real'} | Son símbolos de diagnóstico por ser terminales- |
| Instrucciones | {'id'} | 'id' es símbolo de diagnóstico porque es un primero. |
| RInstrucciones | {',', '⊢'} | ',' es símbolo de diagnóstico por ser un primero. |
| | | '⊢' es símbolo de diagnóstico porque a pesar de ser un siguiente siempre continúa a la estructura |

| | | |
|-----------|--|--|
| | | RInstrucciones independientemente del sitio donde aparezca. |
| Inst | {'id'} | Es símbolo de diagnóstico por ser un terminal. |
| Expresion | {'(', '-', 'false', 'true', 'id', 'not', 'numeroEntero', 'numeroReal'} | Todos son símbolos de diagnóstico porque en una expresión son esperables cualquiera de estos símbolos. |
| E0 | {'(', '-', 'false', 'true', 'id', 'not', 'numeroEntero', 'numeroReal'} | Todos son símbolos de diagnóstico porque en una expresión son esperables cualquiera de estos símbolos. |
| RE0 | {'+', '-'} | '+' y '-' son símbolos de diagnóstico porque son primeros. Los símbolos {'}', ';', '⊢'} no se consideran símbolos de diagnóstico porque son siguientes y su aparición depende del contexto, es decir, no son esperables en cualquier situación. |
| E1 | {'(', '-', 'false', 'true', 'id', 'not', 'numeroEntero', 'numeroReal'} | Todos son símbolos de diagnóstico porque en una expresión son esperables cualquiera de estos símbolos. |
| RE1 | {'and', 'or'} | 'and' y 'or' son símbolos de diagnóstico porque son primeros en el contexto de la regla $RE1 \rightarrow OP1 E2 RE1$ ya que aparecen en la parte derecha de la regla $OP1 \rightarrow and$ y $OP1 \rightarrow or$. Los símbolos {'}', '+', '-', ';', '⊢'} no se consideran símbolos de diagnóstico porque son siguientes y su aparición depende del contexto, es decir, no son esperables en cualquier situación. |
| E2 | {'(', '-', 'false', 'true', 'id', 'not', 'numeroEntero', 'numeroReal'} | Todos son símbolos de diagnóstico porque en una expresión son esperables cualquiera de estos símbolos. |
| RE2 | {'>', '<', '!', '=', 'and', 'or'} | '>', '<', '!' y '=' son símbolo de diagnóstico porque son primeros en el contexto de la regla $RE2 \rightarrow OP2 E3 RE2$ ya que aparecen en la parte derecha de las reglas para OP2. 'and' y 'or' también son símbolos de diagnóstico porque RE2 está dentro del contexto de E1. |
| E3 | {'(', '-', 'false', 'true', 'id', 'not', 'numeroEntero', 'numeroReal'} | Todos son símbolos de diagnóstico porque en una expresión son esperables cualquiera de estos símbolos. |
| RE3 | { '*', '/', '!', '<', '>', '=', 'and', 'or' } | '*' y '/' son símbolos de diagnóstico porque son primeros en el contexto de la regla $RE3 \rightarrow OP3 E2$ ya que aparecen en la parte derecha de las reglas para OP3. '!', '<', '>', '=', 'and' y 'or' también son símbolos de diagnóstico porque RE3 puede estar dentro del contexto de E1 o de RE2 y son primeros dentro de estos contextos. |

| | | |
|------|---|--|
| E4 | {'(', '-', 'false', 'true', 'id', 'not', 'numeroEntero', 'numeroReal'} | Todos son símbolos de diagnóstico porque en una expresión son esperables cualquiera de estos símbolos. |
| E5 | {'(', 'false', 'true', 'id', 'numeroEntero', 'numeroReal'} | Todos son símbolos de diagnóstico porque en una expresión son esperables cualquiera de estos símbolos. En este caso '-' y not no son símbolos de diagnóstico porque no son primeros ya que irían precedidos de un '('. |
| OP1 | {'and', 'or'} | Son símbolos de diagnóstico por ser terminales. |
| OP2 | {'>', '<', '!', '='} | Son símbolos de diagnóstico por ser terminales. |
| ROP2 | {'=', '(', '-', 'true', 'false', 'id', 'not', 'numeroEntero', 'numeroReal'} | '=' es símbolo de diagnóstico por ser primero. El resto también son símbolos de diagnóstico porque ROP2 está dentro del contexto de E2 y estos son primeros en dicho contexto. |
| OP3 | {'*', '/'} | Son símbolos de diagnóstico por ser terminales- |