PRÁCTICA PL: SEGUNDA FASE I

Tiny(0)

Integrantes:

David Davó Laviña

Ela Katherine Shepherd Arévalo Grupo 12

PASO 1: Especificación sintáctica

Los nombres subrayados son clases léxicas determinadas en la especificación léxica. El símbolo inicial o raíz de la gramática es `Programa.`

```
Programa -> Decs && Insts
Decs -> Dec
Decs -> Decs ; Dec
Dec -> TypN VarN
Insts -> Inst
Insts -> Insts ; Inst
Inst -> VarN = E0
TypN -> int
TypN -> real
TypN -> bool
VarN -> Variable
E0 -> E1 OpIn0AsocD E0
E0 -> E1 OpIn0NoAsoc E1
E0 -> E1
E1 -> E1 OpIn1AsocI E2
E1 -> E2
E2 -> E2 OpIn2AsocI E3
E2 -> E3
E3 -> E4 OpIn3NoAsoc E4
E3 -> E4
E4 -> OpPre4NoAsoc E5
E4 -> OpPre4Asoc E4
E4 -> E5
E5 -> ( E0 )
E5 -> LitEnt
E5 -> LitReal
E5 -> true
E5 -> false
E5 -> Variable
OpIn0AsocD -> +
OpIn0NoAsoc-> -
OpIn1AsocI -> and
OpIn1AsocI -> or
OpIn2AsocI -> <
OpIn2AsocI -> <=
OpIn2AsocI -> >
OpIn2AsocI -> >=
OpIn2AsocI -> ==
OpIn2AsocI -> !=
OpIn3NoAsoc -> *
OpIn3NoAsoc -> /
OpPre4NoAsoc -> -
OpPre4Asoc -> not
```

PASO 2: Acondicionamiento de la gramática

En el paso anterior, las reglas en negrita tienen recursión a izquierdas y las reglas en cursiva tienen un factor común.

```
Programa -> Decs && Insts
Decs -> Dec RDecs
RDecs -> ; Dec RDecs
RDecs -> ε
Dec -> TypN VarN
Insts -> Inst RInsts
RInsts -> ; Inst RInsts
RInsts -> ε
Inst -> VarN = E0
TypN -> int
TypN -> real
TypN -> bool
VarN -> Variable
E0 -> E1 RE0
RE0 -> OpIn0AsocD E0
RE0 -> OpIn0NoAsoc E1
RE0 -> ε
E1 -> E2 RE1
RE1 -> OpIn1AsocI E2 RE1
RE1 -> ε
E2 -> E3 RE2
RE2 -> OpIn2AsocI  E3 RE2
RE2 -> ε
E3 -> E4 RE3
RE3 -> OpIn3NoAsoc E4
RE3 -> ε
E4 -> OpPre4NoAsoc E5
E4 -> OpPre4Asoc E4
E4 -> E5
E5 -> ( E0 )
E5 -> LitEnt
E5 -> LitReal
E5 -> true
E5 -> false
E5 -> Variable
OpIn0AsocD -> +
OpIn0NoAsoc-> -
OpIn1AsocI -> and
OpIn1AsocI -> or
OpIn2AsocI -> <
OpIn2AsocI -> <=
OpIn2AsocI -> >
```

```
OpIn2AsocI -> >=
OpIn2AsocI -> ==
OpIn2AsocI -> !=
OpIn3NoAsoc -> *
OpIn3NoAsoc -> /
OpPre4NoAsoc -> -
OpPre4Asoc -> not
```

PASO 3: Directores de cada regla de la gramática

La herramienta Proletool nos proporciona los primeros y siguientes de cada no terminal, y los directores de cada regla.

NO TERMINAL	PRIMEROS	SIGUIENTES
Programa	bool, real, int	EOF
Decs	bool, real, int	&&
RDecs	;, ε	&&
Dec	bool, real, int	;, &&
Insts	Variable	EOF
RInsts	;, ε	EOF
Inst	Variable	;, EOF
TypN	bool, real, int	Variable
VarN	Variable	;, &&, =
E0	not, LitReal, Variable, -, LitEnt, true, false, (	;, EOF,)
RE0	-, +, ε	;, EOF,)
E1	not, LitReal, Variable, -, LitEnt, true, false, (	;, EOF,), -, +
RE1	or, and, ε	;, EOF,), -, +
E2	not, LitReal, Variable, -, LitEnt, true, false, (	or, ;, EOF,), and, -, +
RE2	>, <, ==, !=, <=, >=, ε	or, ;, EOF,), and, -, +
E3	not, LitReal, Variable, -, LitEnt, true, false, (	or, <, ==, EOF,), -, <=, >, ;, !=, and, >=, +

RE3	/, *, ε	or, <, ==, EOF,), -, <=, >, ;, !=, and, >=, +
E4	not, LitReal, Variable, -, LitEnt, true, false, (	or, <, ==, EOF,), -, <=, /, *, >, ;, !=, and, >=, +
E5	LitReal, Variable, LitEnt, true, false, (	or, <, ==, EOF,), -, <=, /, *, >, ;, !=, and, >=, +
OpIn0AsocD	+	not, LitReal, Variable, -, LitEnt, true, false, (
OpIn0NoAsoc	-	not, LitReal, Variable, -, LitEnt, true, false, (
OpIn1AsocI	or, and	not, LitReal, Variable, -, LitEnt, true, false, (
OpIn2AsocI	>, <, ==, !=, <=, >=	not, LitReal, Variable, -, LitEnt, true, false, (
OpIn3NoAsoc	/, *	not, LitReal, Variable, -, LitEnt, true, false, (
OpPre4NoAsoc	-	LitReal, Variable, LitEnt, true, false, (
OpPre4Asoc	not	not, LitReal, Variable, -, LitEnt, true, false, (

REGLA	**DIRECTORES**
Programa -> Decs && Insts	{bool, int, real}
Decs -> Dec RDecs	{bool, int, real}
RDecs -> ; Dec RDecs	{;}
RDecs -> ε	{&&}
Dec -> TypN VarN	{bool, int, real}
Insts -> Inst RInsts	{Variable}
RInsts -> ; Inst RInsts	{;}

RInsts -> ε	{EOF}
Inst -> VarN = E0	{Variable}
TypN -> int	{int}
TypN -> real	{real}
TypN -> bool	{bool}
VarN -> Variable	{Variable}
E0 -> E1 RE0	{false, LitEnt, LitReal, -, not, (, true, Variable}
RE0 -> OpIn0AsocD E0	{+}
RE0 -> OpIn0NoAsoc E1	{-}
RE0 -> ε	{), ;, EOF}
E1 -> E2 RE1	{false, LitEnt, LitReal, -, not, (, true, Variable}
RE1 -> OpIn1AsocI E2 RE1	{and, or}
RE1 -> ε	{+, -,), ;, EOF}
E2 -> E3 RE2	{false, LitEnt, LitReal, -, not, (, true, Variable}
RE2 -> OpIn2AsocI E3 RE2	{!=, ==, >=, >, <=, <}
RE2 -> ε	{and, +, -, or,), ;, EOF}
E3 -> E4 RE3	{false, LitReal, LitEnt, -, not, (, true, Variable}
RE3 -> OpIn3NoAsoc E4	{/, *}
RE3 -> ε	{and, !=, ==, +, >=, >, <=, <, -, or,), ;, EOF}
E4 -> OpPre4NoAsoc E5	{-}
E4 -> OpPre4Asoc E4	{not}
E4 -> E5	{false, LitEnt, LitReal, (, true,

	Variable}
E5 -> (E0)	{(}
E5 -> LitEnt	{LitEnt}
E5 -> LitReal	{LitReal}
E5 -> true	{true}
E5 -> false	{false}
E5 -> Variable	{Variable}
OpIn0AsocD -> +	{+}
OpIn0NoAsoc-> -	{-}
OpIn1AsocI -> and	{and}
OpIn1AsocI -> or	{or}
OpIn2AsocI -> <	{<}
OpIn2AsocI -> <=	{<=}
OpIn2AsocI -> >	{>}
OpIn2AsocI -> >=	{>=}
OpIn2AsocI -> ==	{==}
OpIn2AsocI -> !=	{!=}
OpIn3NoAsoc -> *	{*}
OpIn3NoAsoc -> /	{/}
OpPre4NoAsoc -> -	{-}
OpPre4Asoc -> not	{not}

PASO 4: Símbolos para el diagnóstico de errores

NO TERMINAL	SÍMBOLOS DIAGNÓSTICO	RAZONAMIENTO
Programa	{bool, real, int}	todos los símbolos de diagnóstico son primeros del no terminal
Decs	{bool, real, int}	todos los símbolos de diagnóstico son primeros del no terminal
RDecs	{;, &&}	; es un primero de RDecs, y && es un siguiente que siempre va a aparecer. RDecs aparece en Decs, y después de Decs siempre va a estar &&
Dec	{bool, real, int}	todos los símbolos de diagnóstico son primeros del no terminal
Insts	{Variable}	todos los símbolos de diagnóstico son primeros del no terminal
RInsts	{;, EOF}	; es un primero de RInsts, y RInsts solo aparece en Insts, que siempre va a tener después el fin de fichero
Inst	{Variable}	todos los símbolos de diagnóstico son primeros del no terminal
TypN	{bool, real, int}	todos los símbolos de diagnóstico son primeros del no terminal
VarN	{Variable}	todos los símbolos de diagnóstico son primeros del no terminal
E0	{not, LitReal, Variable, -, LitEnt, true, false, (}	todos los símbolos de diagnóstico son primeros del no terminal
RE0	{-, +}	- y + son primeros de RE0. RE0 es el final del lado derecho de una instrucción. Como la parte de instrucciones está

		formada por una o varias instrucciones, no podemos asegurar que después de RE0 haya siempre un fin de fichero (puede haber más instrucciones), un paréntesis (quizás no se han modificado las prioridades) o un punto y coma (puede ser la última instrucción)
E1	{not, LitReal, Variable, -, LitEnt, true, false, (}	todos los símbolos de diagnóstico son primeros del no terminal
RE1	{or, and}	or y and son primeros de RE1. En cuanto a los siguientes, RE1 aparece en E1, y después de E1 siempre aparece E0. E0 tiene tres reglas con directores distintos, por tanto no podemos asegurar que + o - aparezca siempre. Además, tenemos el mismo problema comentado en RE0.
E2	{not, LitReal, Variable, -, LitEnt, true, false, (}	todos los símbolos de diagnóstico son primeros del no terminal
RE2	{>, <, ==, !=, <=, >=}	Los operadores relacionales son primeros de RE2. Ninguno de sus siguientes son símbolos de diagnóstico por la acumulación de los problemas para añadir más símbolos en RE0 y RE1.
E3	{not, LitReal, Variable, -, LitEnt, true, false, (}	todos los símbolos de diagnóstico son primeros del no terminal
RE3	{/, *}	/ y * son los primeros de RE3. No podemos asegurar que cualquiera de sus siguientes

		vayan a aparecer siempre por lo explicado en RE0, RE1, y RE2.
E4	{not, LitReal, Variable, -, LitEnt, true, false, (}	todos los símbolos de diagnóstico son primeros del no terminal
E5	{LitReal, Variable, LitEnt, true, false, (}	todos los símbolos de diagnóstico son primeros del no terminal
OpIn0AsocD	{+}	todos los símbolos de diagnóstico son primeros del no terminal
OpIn0NoAsoc	{-}	todos los símbolos de diagnóstico son primeros del no terminal
OpIn1AsocI	{or, and}	todos los símbolos de diagnóstico son primeros del no terminal
OpIn2AsocI	{>, <, ==, !=, <=, >=}	todos los símbolos de diagnóstico son primeros del no terminal
OpIn3NoAsoc	{/, *}	todos los símbolos de diagnóstico son primeros del no terminal
OpPre4NoAsoc	{-}	todos los símbolos de diagnóstico son primeros del no terminal
OpPre4Asoc	{not}	todos los símbolos de diagnóstico son primeros del no terminal