

IIC2323 — Construcción de Compiladores

Agú: Descripción Léxica

Primer semestre 2016

Índice

1. Notación	1
2. Estructura léxica	2
2.1. Identificadores	2
2.2. Keywords	2
2.3. Otros símbolos	2
2.4. Expresiones literales	2
2.5. Separadores	3
2.6. Comentarios	3
2.7. Espacios en blanco	3

1. Notación

En el documento, el lenguaje es definido utilizando gramáticas libres de contexto, donde al lado derecho de las producciones se utilizan expresiones regulares con la siguiente notación:

1. $[\varphi]$ corresponde a $\varphi + \varepsilon$, es decir, que φ es opcional,
2. $\{\varphi\}$ corresponde a φ^* , es decir que φ se repite una cantidad arbitraria de veces,
3. $\varphi|\psi$ es equivalente a $\varphi + \psi$, es decir que se utiliza φ o ψ
4. (φ) corresponde a φ . En caso de necesitar paréntesis como terminales, estos se indicaran como $()$.

Por ejemplo, la gramática:

$$\begin{aligned} identifier &\rightarrow [\$]letter\{letter|digit\} \\ letter &\rightarrow a|b|c \\ digit &\rightarrow 0|1 \end{aligned}$$

indica que un identificador comienza opcionalmente con \$, continúa con una letra, y termina con una cantidad arbitraria de letras y dígitos. Algunos identificadores válidos según esta son: \$a101b, b, y \$c0, mientras que 1a00c y a\$01 no lo son.

2. Estructura léxica

2.1. Identificadores

Los identificadores dan nombre a clases, tipos, variables y funciones. En el caso de tipos y clases, estos comienzan con mayúscula. Más formalmente:

$$\begin{aligned} identifier &\rightarrow (lowercase_letter|_)\{letter|digit|_|\}\{!|?\}\\ type_identifier &\rightarrow uppercase_letter\{letter|digit|_|\}\{!|?\}\\ digit &\rightarrow 0|1|2|3|4|5|6|7|8|9\\ letter &\rightarrow lowercase_letter|uppercase_letter\\ lowercase_letter &\rightarrow a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|á|é|í|ó|ú|ñ\\ uppercase_letter &\rightarrow A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|Á|É|Í|Ó|Ú|Ñ \end{aligned}$$

2.2. Keywords

Los keywords son palabras reservadas del lenguaje, que se utilizan para elementos fundamentales de este, y por lo tanto no pueden ser usados como identificadores.

En este caso, los keywords son:

alias	mut	class	set
val	var	let	def
if	then	else	when
while	do	not	

2.3. Otros símbolos

Se utilizan símbolos especiales para indicar operadores y otros elementos del lenguaje. Estos son:

+	-	*	/	()
;	{	}	,	<-	\$
:	->	==	%	&&	
=	/=	<	>	<=	>=

2.4. Expresiones literales

Las expresiones literales indican valores literales de tipos básicos. Se definen para enteros, chars, strings y booleans, de la siguiente forma:

$$\begin{aligned}
int_literal &\rightarrow 0 \\
&\rightarrow (posdigit \mid posdigit\,digit \mid posdigit\,digit\,digit)\{[-]\,digit\,digit\,digit\} \\
posdigit &\rightarrow 1|2|3|4|5|6|7|8|9 \\
digit &\rightarrow 0 \mid posdigit \\
char_literal &\rightarrow '(nonescaped|escaped)' \\
escaped &\rightarrow \backslash("|\backslash'|a|b|n|r|t|v|\backslash|oct\,oct\,oct) \\
oct &\rightarrow 0|1|2|3|4|5|6|7 \\
string_literal &\rightarrow "{nonescaped|escaped}" \\
bool_literal &\rightarrow \text{true}|\text{false}
\end{aligned}$$

donde *nonescaped* es cualquier símbolo imprimible (por lo menos ASCII), distinto de \, ', y ".

2.5. Separadores

Los separadores se utilizan para separar statements, y se definen como:

$$separator \rightarrow ;\{;\mid whitespace\}$$

donde *whitespace* es un espacio, tab o salto de línea.

2.6. Comentarios

Los comentarios comienzan con // y terminan en el siguiente salto de línea. Son ignorados.

2.7. Espacios en blanco

Los espacios en blanco que no forman parte de un separador son utilizados solo para separar tokens, y por lo tanto son ignorados.