

Análisis Léxico

Palabras Reservadas y operadores +,++

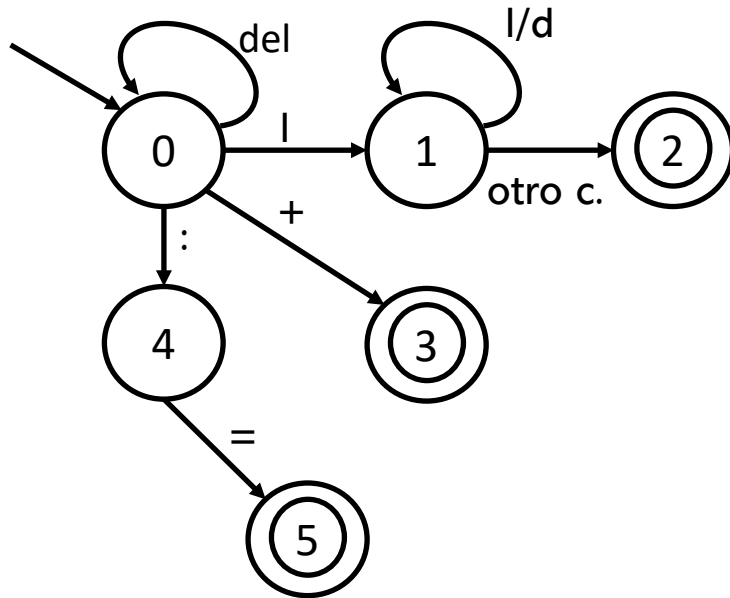
Juan Pedro Caraça-Valente Hernández

septiembre 2020

1) Tokens: Palabras reservadas (if, int, in, then, else, ...); Identificadores

2) G.Reg.: ~~$S \rightarrow \text{del } S \mid \text{"i"} A \mid \text{"t"} B \mid \text{"e"} C \dots$~~
 ~~$A \rightarrow f \mid n A' \mid \lambda \dots$~~ Demasiado Complicado \longrightarrow $S \rightarrow \text{del } S \mid I A \mid \lambda$
 $A \rightarrow I A \mid \lambda$ $A \rightarrow d A$

3) Automata



4-5) Acciones Semánticas y Errores

0-0: Leer

0-1: lexema:=I; Leer

1-1: lexema:=lexema⊕I/d; Leer

0-3: Gen_Token (op_aritmético, 1); Leer

0-4: Leer

4-5: Gen_Token (op_asignación, -); Leer

1-2: If Buscar-Lexema-Lista-Palabras-Reservadas (lexema)

Then Gen-Token (lexema, -) // Token con la Palabra reservada como TipoToken

Else {If Zona_Declaración=true then

{ p=buscar_TS (lexema);

If p=null then { p:=insertar_TS(lexema)

Gen_Token (identificador, p) }

Else /* p<>null */ error ("identificador ya declarado") }

Else /* Zona_Declaración=false */

{ p=buscar_TS (lexema)

If p=null then error ("identificador NO declarado")

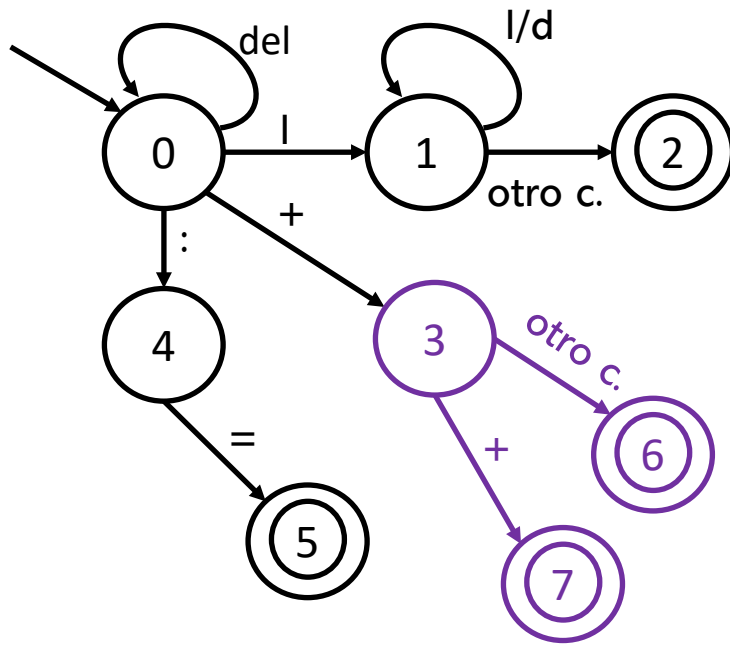
Else /* p<>null */ Gen_Token (identificador, p) } } Leer

1) Tokens: Palabras reservadas (if, int, in, then, else, ...); Identificadores

2) G.Reg.: $S \rightarrow \text{del } S \mid I A \mid \lambda$ Como haría si quiero reconocer también los operadores + y ++
 $A \rightarrow I A \mid \lambda \mid d A$

3) Automata

4-5) Acciones Semánticas y Errores



0-3: Leer

3-6: Gen_Token (op_aritmético, 1)

3-7: Gen_Token (op_aritmético, 5); Leer

Análisis Léxico

Varios tipos de comentarios

Juan Pedro Caraça-Valente Hernández

septiembre 2020

Comentarios (// Esto es un comentario y /* Esto es un comentario */);

Operador /

2) G.Reg.: $S \rightarrow \text{del } S \mid / A \mid \dots$

l: letra c: todo carácter menos <eol>

$A \rightarrow / B \mid * C \mid \lambda$

c': todo carácter menos *

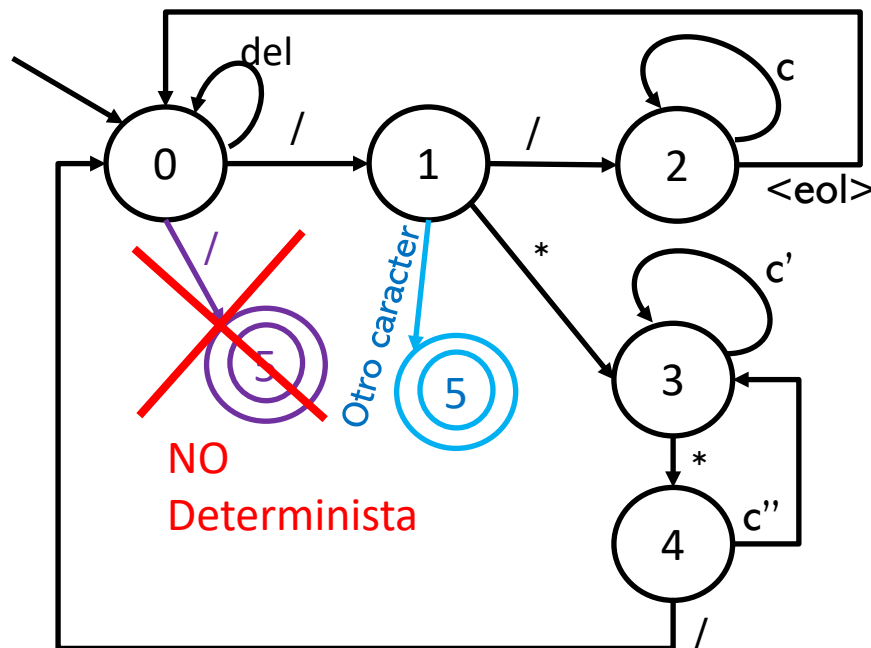
$B \rightarrow c B \mid <\text{eol}> S$

c'': todo carácter menos /

$C \rightarrow c' C \mid * D$

$D \rightarrow / S \mid c'' C$

3) Automata



4-5) Acciones Semánticas y Errores

0-0: Leer

0-1: Leer

1-2: Leer

1-3: Leer

2-2: Leer

2-0: Leer

3-3: Leer

3-4: Leer

4-3: Leer

4-0: Leer

1-5: Gen_Token (op_aritmetico, 4); Leer

Análisis Léxico

Strings y Estrategias de Diseño

Juan Pedro Caraça-Valente Hernández

septiembre 2020

1) Tokens: Cadenas (“Esto es una cadena”) con carácter de escape “Esta /” esta dentro de la cadena”

2) G.Reg.: $S \rightarrow \text{del } S \mid \text{“ } A \mid \dots$

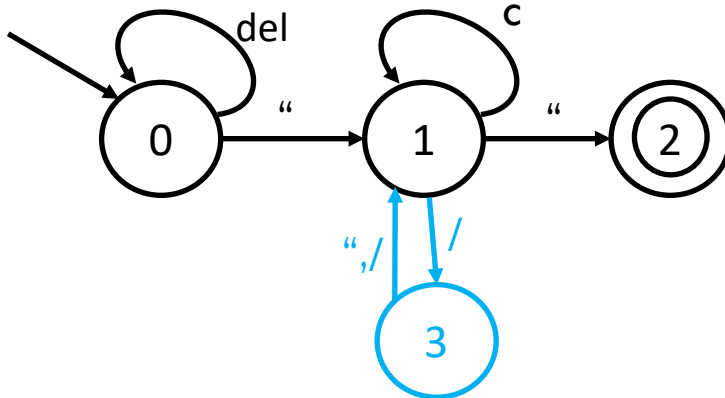
l: letra c: todo carácter menos “ y “/”

$A \rightarrow c A \mid \text{“ } \mid / B$

$B \rightarrow \text{“ } A \mid / A$

3)Automata

4-5) Acciones Semánticas y Errores



0-0: Leer

0-1: Lexema= ϕ ; Leer

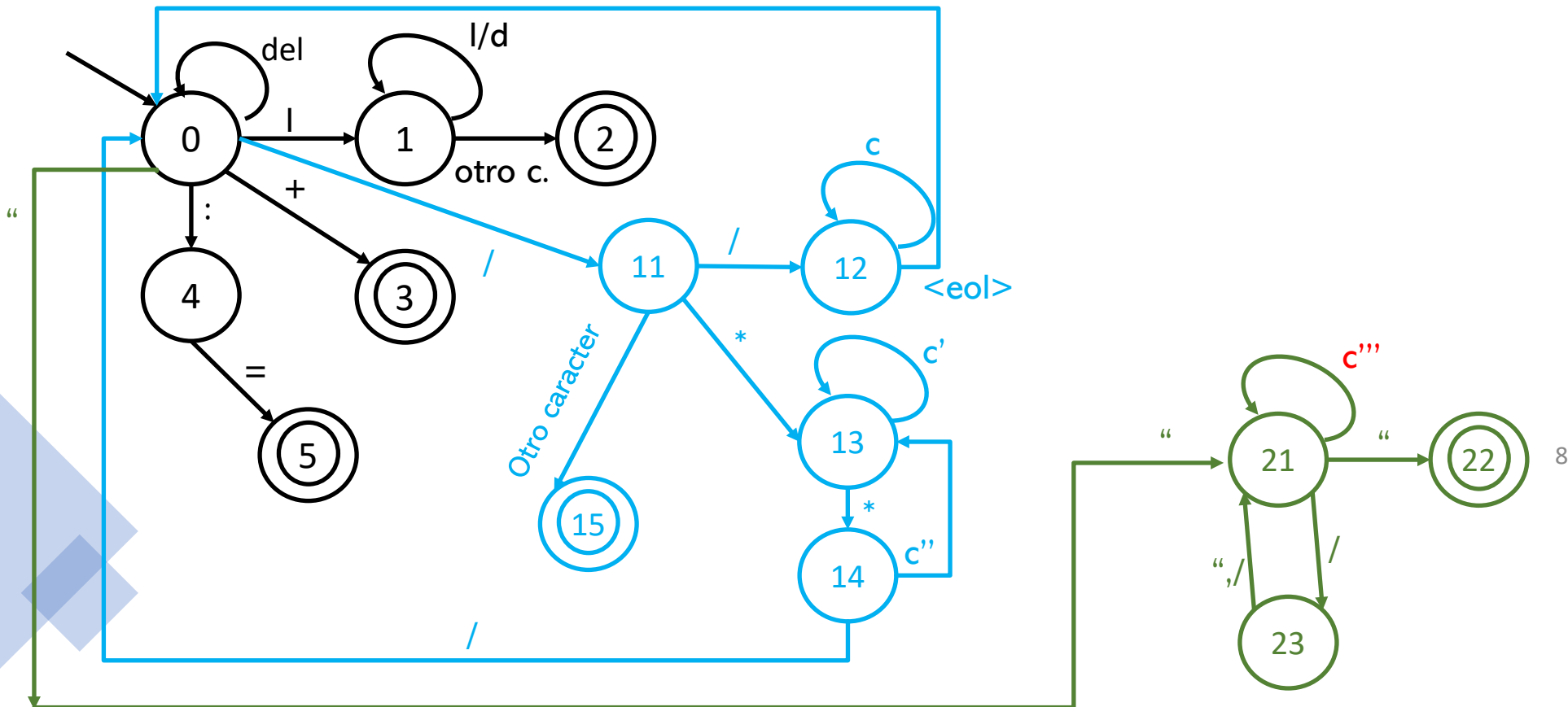
1-1: Lexema=Lexema \oplus c; Leer

1-2: Gen-Token (Cadena, Lexema); Leer

1-3: Leer

3-1: Lexema=Lexema \oplus “ ó / ; Leer

- Puedo usar esta estrategia de crear la Gramática y el autómata de cada tipo de token por separado y luego juntarlo?
 - En general no se hace, hay que tener mucho cuidado y no suele facilitar el diseño sino lo contrario. Tengo además que cambiar los nombres a los No terminales excepto el axioma, posiblemente redefinir los “pseudoterminales” como “c”, y los estados del autómata.
 - Pero para los 3 casos anteriores si puedo juntarlos sin problema porque los 3 tipos de token empiezan siempre por caracteres distintos entre si (letra, /, “)



- En general construiremos la Gramática teniendo en cuenta todos los tipos de token. Cuidado especial con los tipos de token que puedan tener lexemas que empiecen igual.
- Lo mismo para el autómatata, siguiendo lo definido en la Gramática