

Teoría de la Computación

Sesión 13

Edgar Andrade, Ph.D.

Matemáticas Aplicadas y Ciencias de la computación

Última revisión: Agosto de 2021



MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Contenido

Scanners y Parsers

Métodos de parsing

Autómatas de Knuth

Autómata DK



Contenido

Scanners y Parsers

Métodos de parsing

Autómatas de Knuth

Autómata DK



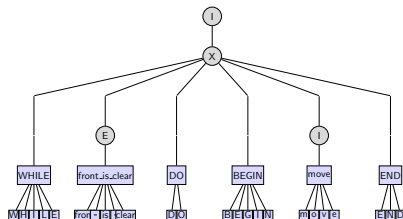
Sea G una CFG y suponga que $w \in L(G)$.

Definición

El problema de parsing para w es encontrar un árbol de análisis de w en G .



WHILE front_is_clear **DO BEGIN** move **END**

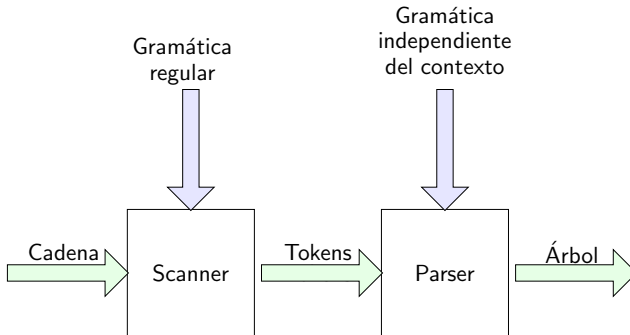


Sea G una CFG y suponga que $w \in L(G)$.

Definición

El problema de parsing para w es encontrar un árbol de análisis de w en G .





Contenido

Scanners y Parsers

Métodos de parsing

Autómatas de Knuth

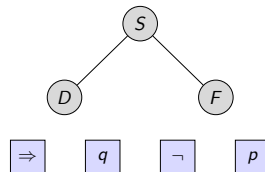
Autómata DK



Top-down parser

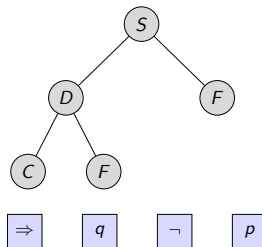
$$\begin{aligned} G = \{ & S \rightarrow NF \mid DF \mid p \mid q \\ & F \rightarrow NF \mid DF \mid p \mid q \\ & D \rightarrow CF \\ & N \rightarrow \neg \\ & C \rightarrow \wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow \} \end{aligned}$$
 \Rightarrow q \neg p 

Top-down parser

$$\begin{aligned} G = \{ & \textcolor{red}{S} \rightarrow NF \mid \textcolor{red}{DF} \mid p \mid q \\ & F \rightarrow NF \mid DF \mid p \mid q \\ & D \rightarrow CF \\ & N \rightarrow \neg \\ & C \rightarrow \wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow \} \end{aligned}$$


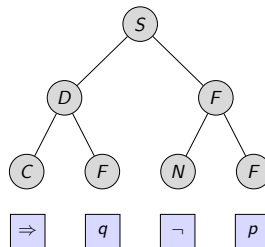
Top-down parser

$G = \{ S \rightarrow NF \mid DF \mid p \mid q$
 $F \rightarrow NF \mid DF \mid p \mid q$
 $D \rightarrow CF$
 $N \rightarrow \neg$
 $C \rightarrow \wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow \}$



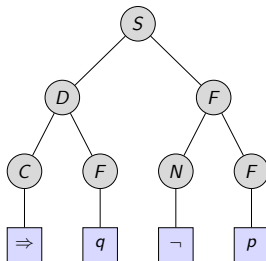
Top-down parser

$G = \{ S \rightarrow NF \mid DF \mid p \mid q$
 $\quad F \rightarrow NF \mid DF \mid p \mid q$
 $\quad D \rightarrow CF$
 $\quad N \rightarrow \neg$
 $\quad C \rightarrow \wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow \}$



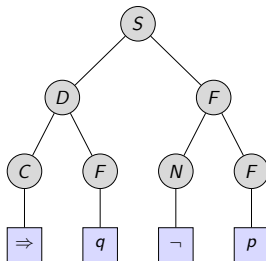
Top-down a ciegas

¿Cuántos recursos consume una búsqueda top-down a ciegas?



Top-down a ciegas

¿Cuántos recursos consume una búsqueda top-down a ciegas?

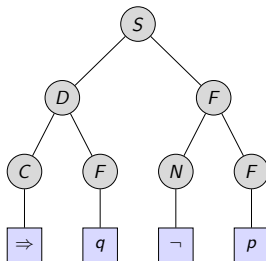


👉 Supongamos G en Forma normal de Chomsky.



Top-down a ciegas

¿Cuántos recursos consume una búsqueda top-down a ciegas?



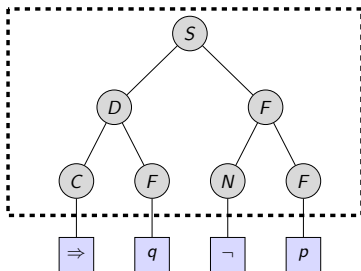
Conteo de reglas

Para una cadena de longitud n se requieren n aplicaciones de reglas de tipo $A \rightarrow a$.



Top-down a ciegas

¿Cuántos recursos consume una búsqueda top-down a ciegas?



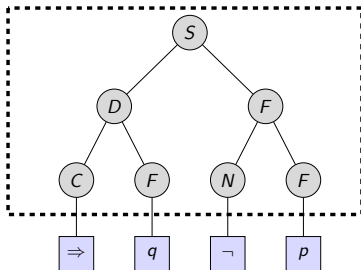
Conteo de reglas

Número de reglas de tipo $A \rightarrow BC$ es igual a la mitad de aristas del árbol.



Top-down a ciegas

¿Cuántos recursos consume una búsqueda top-down a ciegas?



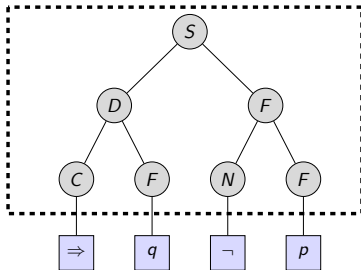
Conteo de reglas

El número de aristas de un árbol binario con n hojas es igual a $2(n - 1)$.



Top-down a ciegas

¿Cuántos recursos consume una búsqueda top-down a ciegas?



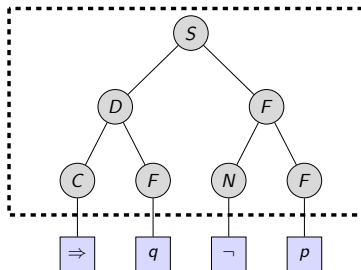
Conteo de reglas

Se requieren $n - 1$ reglas de tipo $A \rightarrow BC$.



Top-down a ciegas

¿Cuántos recursos consume una búsqueda top-down a ciegas?



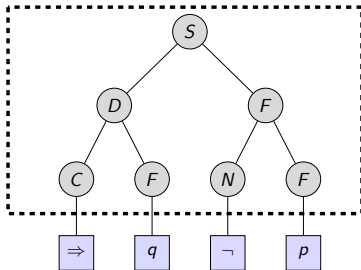
Conteo de reglas

En total: $n + (n - 1) = 2n - 1$ aplicaciones de reglas.



Top-down a ciegas

¿Cuántos recursos consume una búsqueda top-down a ciegas?



Conteo de reglas

Si la gramática tiene m reglas, hay m^{2^n-1} posibles combinaciones.



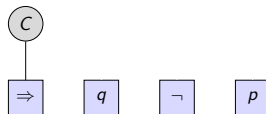
Bottom-up parser

$$\begin{aligned}G = \{ & S \rightarrow NF \mid DF \mid p \mid q \\ & F \rightarrow NF \mid DF \mid p \mid q \\ & D \rightarrow CF \\ & N \rightarrow \neg \\ & C \rightarrow \wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow \}\end{aligned}$$



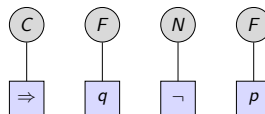
Bottom-up parser

$$\begin{aligned}G = \{ & S \rightarrow NF \mid DF \mid p \mid q \\ & F \rightarrow NF \mid DF \mid p \mid q \\ & D \rightarrow CF \\ & N \rightarrow \neg \\ & C \rightarrow \wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow \}\end{aligned}$$



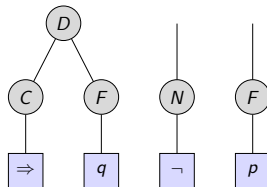
Bottom-up parser

$$\begin{aligned}G = \{ & S \rightarrow NF \mid DF \mid p \mid q \\ & F \rightarrow NF \mid DF \mid p \mid q \\ & D \rightarrow CF \\ & N \rightarrow \neg \\ & C \rightarrow \wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow \}\end{aligned}$$



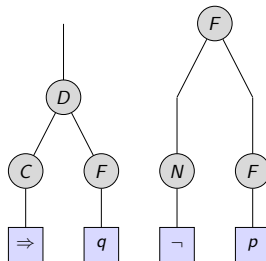
Bottom-up parser

$G = \{ S \rightarrow NF \mid DF \mid p \mid q$
 $F \rightarrow NF \mid DF \mid p \mid q$
 $D \rightarrow CF$
 $N \rightarrow \neg$
 $C \rightarrow \wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow \}$



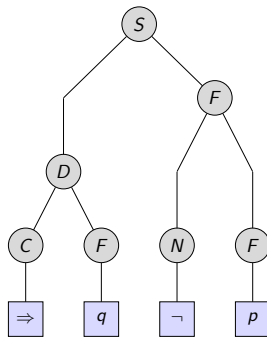
Bottom-up parser

$G = \{S \rightarrow NF \mid DF \mid p \mid q$
 $\quad F \rightarrow NF \mid DF \mid p \mid q$
 $\quad D \rightarrow CF$
 $\quad N \rightarrow \neg$
 $\quad C \rightarrow \wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow\}$



Bottom-up parser

$G = \{ S \rightarrow NF \mid DF \mid p \mid q$
 $F \rightarrow NF \mid DF \mid p \mid q$
 $D \rightarrow CF$
 $N \rightarrow \neg$
 $C \rightarrow \wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow \}$



Contenido

Scanners y Parsers

Métodos de parsing

Autómatas de Knuth

Autómata DK



Definiciones

Manijas

Una **manija** para una cadena $w = xhy$ es una regla de la forma $A \rightarrow h$.



Definiciones

Manijas

Una **manija** para una cadena $w = xhy$ es una regla de la forma $A \rightarrow h$.

Reducción

La regla $A \rightarrow h$ permite reducir la expresión xhy a la expresión xAy ($xhy \rightarrow xAy$).



Definiciones

Manijas

Una **manija** para una cadena $w = xhy$ es una regla de la forma $A \rightarrow h$.

Reducción

La regla $A \rightarrow h$ permite reducir la expresión xhy a la expresión xAy ($xhy \rightarrow xAy$).

Ejemplo

$$\begin{aligned} \Rightarrow q \neg p &\rightarrow Cq \neg p \rightarrow CF \neg p \rightarrow CFN \underline{p} \\ &\rightarrow \underline{CFNF} \rightarrow \underline{DNF} \rightarrow \underline{DF} \rightarrow S \end{aligned}$$



Búsqueda de manijas

Definición

Decimos que una CFG G es **determinista** sii cada cadena en el procesamiento bottom-up de w (con $w \in L(G)$) tiene sólo una manija.



Búsqueda de manijas

Definición

Decimos que una CFG G es **determinista** sii cada cadena en el procesamiento bottom-up de w (con $w \in L(G)$) tiene sólo una manija.

Ejemplo

$$G = \{F \rightarrow NF \mid DF \\ D \rightarrow CF\}$$

Determinista



Búsqueda de manijas

Definición

Decimos que una CFG G es **determinista** sii cada cadena en el procesamiento bottom-up de w (con $w \in L(G)$) tiene sólo una manija.

Ejemplo

$$G = \{F \rightarrow NF \mid DF \\ D \rightarrow CF\}$$

Determinista

$$G' = \{S \rightarrow NF \mid DF \\ F \rightarrow NF \mid DF \\ D \rightarrow CF\}$$

No determinista



Búsqueda de manijas

Definición

Decimos que una CFG G es **determinista** sii cada cadena en el procesamiento bottom-up de w (con $w \in L(G)$) tiene sólo una manija.

Ejemplo

$$G = \{F \rightarrow NF \mid DF \\ D \rightarrow CF\}$$

Determinista

$$G' = \{S \rightarrow NF \mid DF \\ F \rightarrow NF \mid DF \\ D \rightarrow CF\}$$

No determinista

☞ Construiremos un autómata mediante el cual buscar cada manija en el procesamiento bottom-up de w .



El autómata K para encontrar manijas:

Reglas punteadas

$$F \rightarrow .NF$$

$$F \rightarrow N.F$$

$$F \rightarrow NF.$$



El autómatas K para encontrar manijas:

Reglas punteadas

$$F \rightarrow .NF$$

$$F \rightarrow N.F$$

$$F \rightarrow NF.$$

Estados del autómatas K

q_0 más todos los estados correspondientes a reglas punteadas:

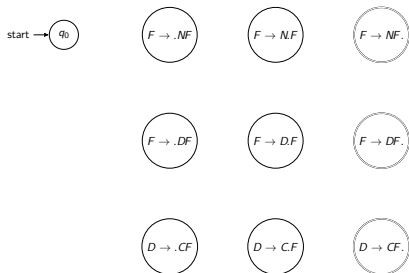
$$F \rightarrow .NF$$

$$F \rightarrow N.F$$

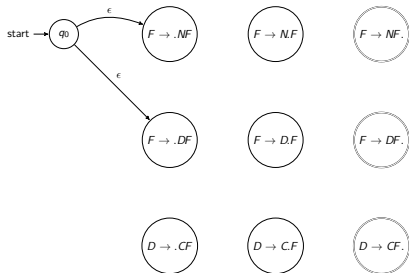
$$F \rightarrow NF.$$



El autómatas K para encontrar manijas:



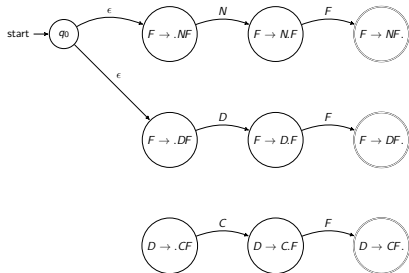
El autómata K para encontrar manijas:



👉 Transición ϵ de q_0 a las reglas con el símbolo inicial.



El autómata K para encontrar manijas:

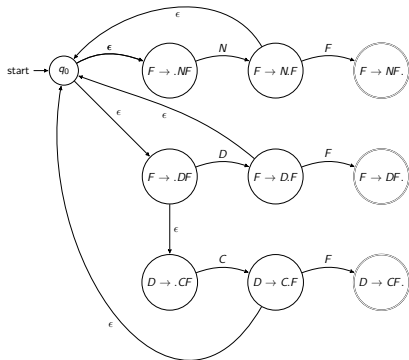


 **Shift-moves** entre las reglas punteadas correspondientes:

$$\boxed{B \rightarrow u.av} \xrightarrow{a} \boxed{B \rightarrow ua.v}$$



El autómata K para encontrar manijas:



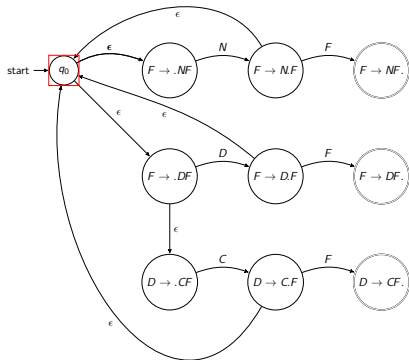
👉 **ϵ -moves** desde $\cdot C$ hasta una regla que comienza con C :

$$B \rightarrow u \cdot C v \xrightarrow{\epsilon} C \rightarrow \cdot r$$



MACC
Matemáticas Aplicadas y
Ciencias de la Computación

El autómata K para encontrar manijas:

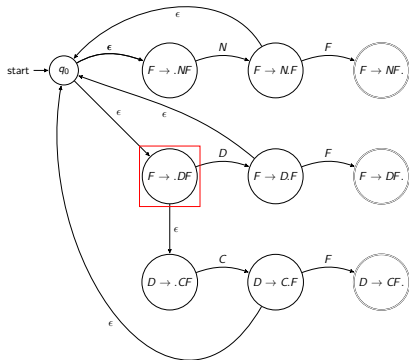


Ejemplo: $CFNF$



MACC
Matemáticas Aplicadas y
Ciencias de la Computación

El autómata K para encontrar manijas:

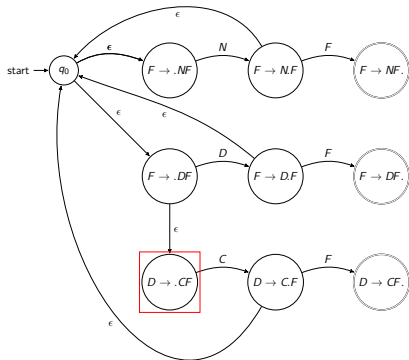


Ejemplo: $CFNF$



MACC
Matemáticas Aplicadas y
Ciencias de la Computación

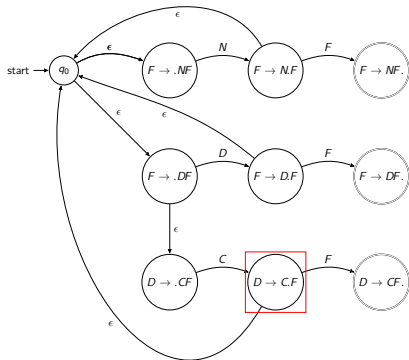
El autómata K para encontrar manijas:



Ejemplo: $CFNF$



El autómata K para encontrar manijas:

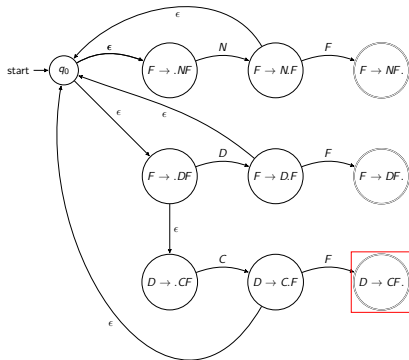


Ejemplo: $CFNF$



MACC
Matemáticas Aplicadas y
Ciencias de la Computación

El autómata K para encontrar manijas:



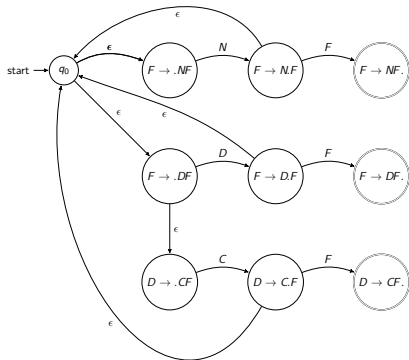
Ejemplo: \underline{CFNF}

¡Se encontró la manija!



MACC
Matemáticas Aplicadas y
Ciencias de la Computación

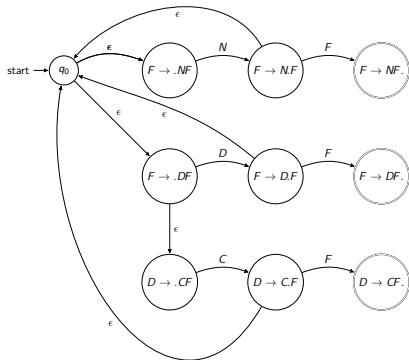
El autómata K para encontrar manijas:



Ejemplo: $\underline{CFNF} \rightarrow DNF$



El autómata K para encontrar manijas:

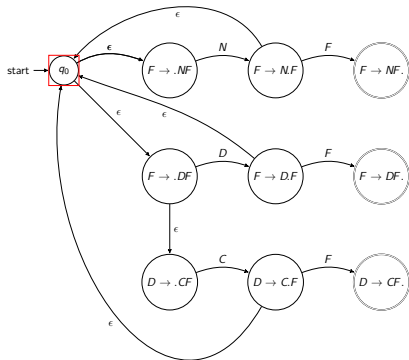


Ejemplo: Ahora se inicializa el autómata para procesar DNF



MACC
Matemáticas Aplicadas y
Ciencias de la Computación

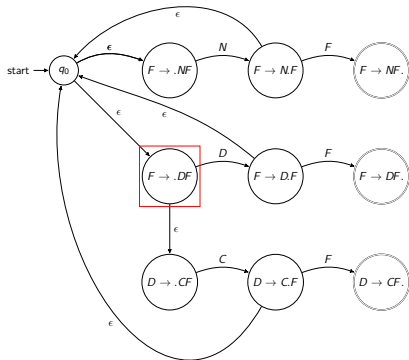
El autómata K para encontrar manijas:



Ejemplo: DNF



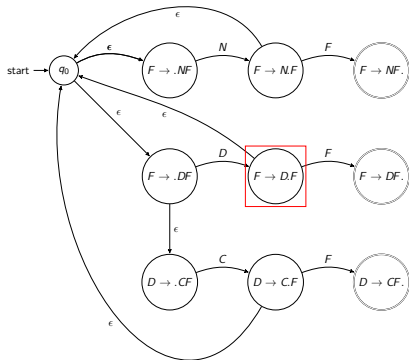
El autómata K para encontrar manijas:



Ejemplo: DNF



El autómata K para encontrar manijas:

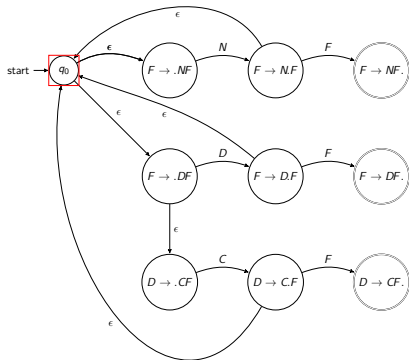


Ejemplo: DNF



MACC
Matemáticas Aplicadas y
Ciencias de la Computación

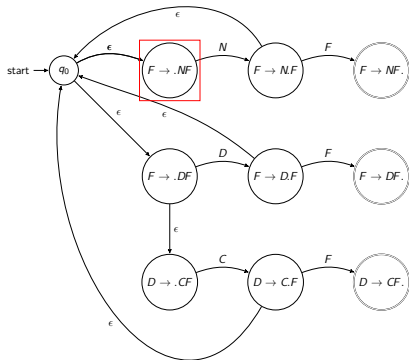
El autómata K para encontrar manijas:



Ejemplo: DNF



El autómata K para encontrar manijas:

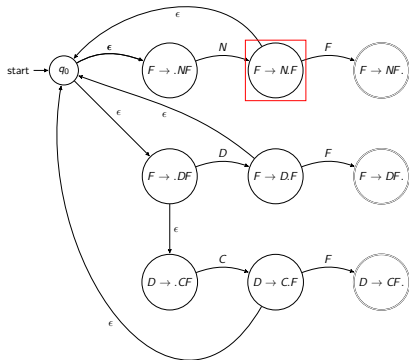


Ejemplo: DNF



MACC
Matemáticas Aplicadas y
Ciencias de la Computación

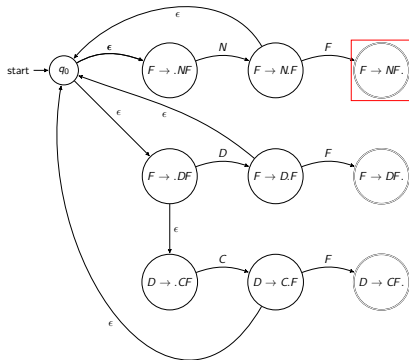
El autómata K para encontrar manijas:



Ejemplo: DNF



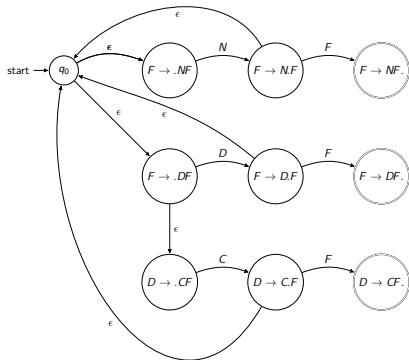
El autómatas K para encontrar manijas:



Ejemplo: $\underline{DNF} \rightarrow DF$



El autómata K para encontrar manijas:

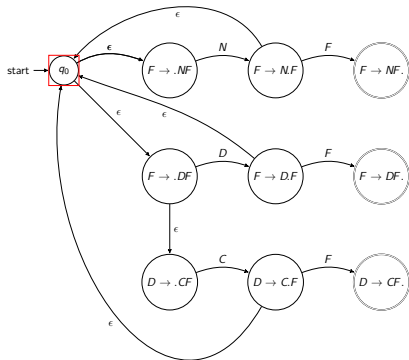


Ejemplo: Ahora se inicializa el autómata para procesar DF



MACC
Matemáticas Aplicadas y
Ciencias de la Computación

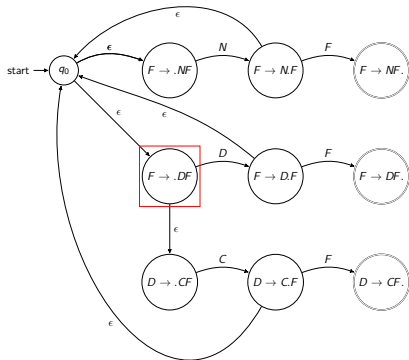
El autómata K para encontrar manijas:



Ejemplo: DF



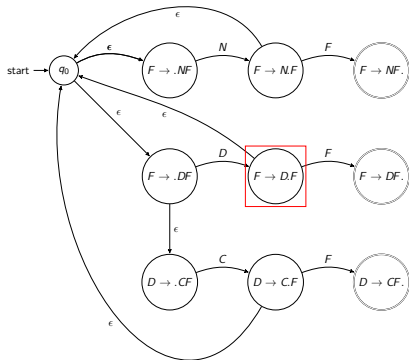
El autómata K para encontrar manijas:



Ejemplo: DF



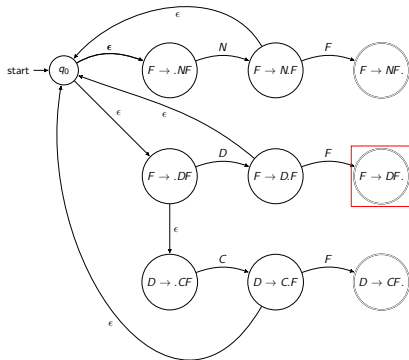
El autómata K para encontrar manijas:



Ejemplo: DF



El autómata K para encontrar manijas:



Ejemplo: $\underline{DF} \rightarrow F$



Contenido

Scanners y Parsers

Métodos de parsing

Autómatas de Knuth

Autómata DK



Versión determinista

👉 Todo autómata no determinista tiene su equivalente determinista.



Versión determinista

👉 Todo autómata no determinista tiene su equivalente determinista.

👉 El procedimiento estándar para obtener la versión determinista del autómata K, llamado DK, es muy engorroso.



Versión determinista

- 👉 Todo autómata no determinista tiene su equivalente determinista.
- 👉 El procedimiento estándar para obtener la versión determinista del autómata K , llamado DK , es muy engorroso.
- 👉 ¡Hay un procedimiento más elegante!



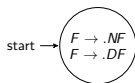
Construcción del autómata DK (1/2)

1. Cree un estado inicial q_0 con todas las reglas punteadas que arrancan con el símbolo inicial y el punto al inicio.
2. Si alguna regla en el estado en consideración, digamos q_i , tiene un punto antes de un símbolo no terminal, digamos C , entonces debemos añadir a q_i todas las reglas punteadas que arrancan con C y el punto al inicio. Continúe este proceso hasta que no pueda repetirse más.



Ejemplo

Paso 1:

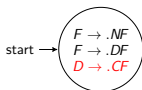


$$G = \{ F \rightarrow NF \mid DF \\ D \rightarrow CF \}$$



Ejemplo

Paso 2 aplicado al estado inicial:



$$G = \{F \rightarrow NF \mid DF \\ D \rightarrow CF\}$$



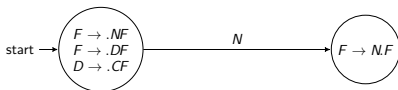
Construcción del autómata DK (2/2)

3. Para cada regla punteada de q_i y para cada símbolo c que sea precedido por el punto (p.ej., $B \rightarrow u.ca$), se debe enviar una transición etiquetada con c al estado q_j , el cual contiene todas las reglas $B \rightarrow uc.a$ de q_i (hay que crear q_j si no existiera).
4. Los estados finales son los que contienen por lo menos una regla punteada completa (p.ej., $B \rightarrow x.$).



Ejemplo

Paso 3 creando el estado:

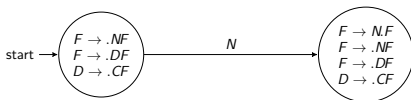


$$G = \{F \rightarrow NF \mid DF \\ D \rightarrow CF\}$$



Ejemplo

Paso 2 sobre el estado recién creado:

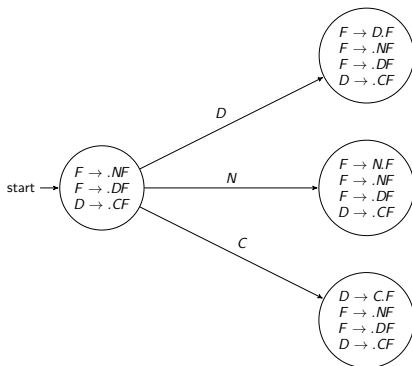


$$G = \{F \rightarrow NF \mid DF \\ D \rightarrow CF\}$$



Ejemplo

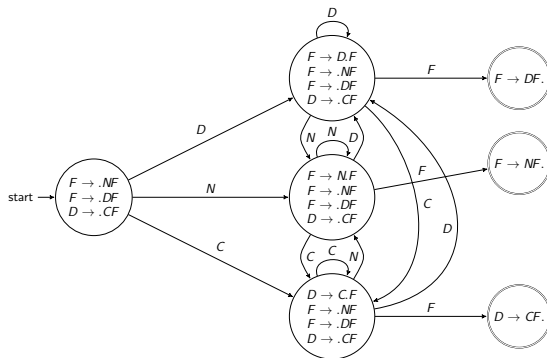
Iterando los pasos 3 y 2:



$$G = \{ F \rightarrow NF \mid DF \\ D \rightarrow CF \}$$



Después de varias iteraciones...



👉 Intente realizar el parsing de $CFNF$ encontrando las manijas mediante este autómata.



En esta sesión usted aprendió

- Reconocer el papel del scanner y del parser en el procesamiento de textos.
- Distinguir entre los métodos de parsing top-down y bottom-up.
- Calcular la complejidad del método top-down para el caso de las gramáticas en Forma Normal de Chomsky.
- Encontrar los autómatas de Knuth de una gramática determinista.
- Realizar el método bottom-up buscando las manijas mediante los autómatas de Knuth.

