Gramáticas libres de contexto

Clase 19

IIC2223 / IIC2224

Prof. Cristian Riveros

Outline

Algoritmo de KMP (clase anterior)

Definición de grámaticas

Outline

Algoritmo de KMP (clase anterior)

Definición de grámaticas

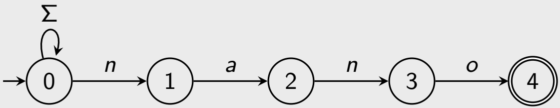
Autómata de un patrón (recordatorio)

Definición

Dado un palabra $w = w_1 \dots w_m$, sea el NFA $\mathcal{A}_w = (Q, \Sigma, \Delta, I, F)$ tal que:

- $Q = \{0, 1, \dots, m\}$
- $\Delta = \{(0, a, 0) \mid a \in \Sigma\} \;\cup\; \{(i, w_{i+1}, i+1) \mid i < m\}$
- $I = \{0\}$ y $F = \{m\}$.

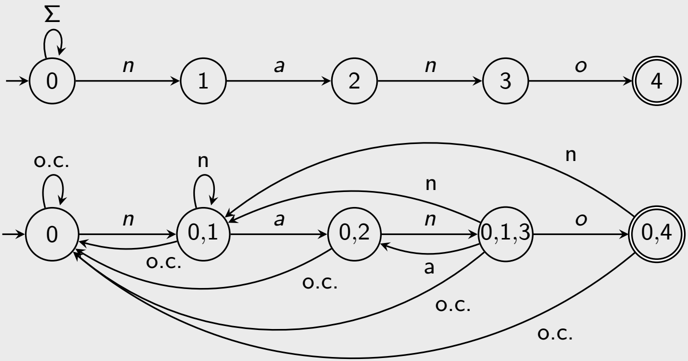
Ejemplo: palabra w = nano



Determinización de $\mathcal{A}_w$ (recordatorio)

Sea $\mathcal{A}_w^{\text{det}} = (Q^{\text{det}}, \Sigma, \delta^{\text{det}}, \{0\}, F^{\text{det}})$ la determinización de $\mathcal{A}_w$ tal que Q^{det} contiene **solo los estados alcanzables** desde $\{0\}$.

Ejemplo: palabra w = nano



¿cuál es el tamaño de $\mathcal{A}_w^{\text{det}}$? (recordatorio)

Sea $w = w_1 \dots w_m$ y $\mathcal{A}_w^{\text{det}} = (Q^{\text{det}}, \Sigma, \delta^{\text{det}}, \{0\}, F^{\text{det}})$ la determ. de $\mathcal{A}_w$.

Teorema

Para todo $S \in Q^{\text{det}}$ y $i \in \{0, 1, \dots, m\}$ se cumple que:

$$i \in S \quad \text{si, y solo si,} \quad w_1 \dots w_i \text{ es un sufijo de } w_1 \dots w_{\max(S)}.$$

Corolarios

- Para todo $S_1, S_2 \in Q^{\text{det}}$, si $\max(S_1) = \max(S_2)$, entonces $S_1 = S_2$.
- $\mathcal{A}_w^{\text{det}}$ tiene $|w| + 1$ estados y a lo más $\mathcal{O}(|w|^2)$ transiciones.

Por lo tanto, encontrar todos los substrings de w en d toma tiempo $\mathcal{O}(|d| + |w|^2)$

Autómata finito con k-lookahead (recordatorio)

Sea Σ un alfabeto finito.

Definiciones

Se definen los siguientes conjuntos de palabra:

- $\Sigma_\bullet = \Sigma^* \times \Sigma^*$
- $\Sigma_\bullet^k = \left\{ (u, v) \in \Sigma_\bullet \mid |uv| = k \right\}$

Notación

En vez de $(u, v) \in \Sigma_\bullet$, escribiremos $u.v \in \Sigma_\bullet$.

Ejemplos

Si $\Sigma = \{a, b\}$ entonces:

- $ab.ba \in \Sigma_\bullet$ y $.aba \in \Sigma_\bullet$
- $ab.ba \in \Sigma_\bullet^4$ y $.aba \in \Sigma_\bullet^3$

Autómata finito con k-lookahead (recordatorio)

Definición

Un autómata finito determinista con k-lookahead es:

$$\mathcal{A} \quad = \quad (Q, \Sigma, \delta, q_0, F)$$

- Q es un conjunto finito de estados.
- Σ es el alfabeto de input.
- q_0 es el estado inicial
- $F \subseteq Q$ es el conjunto de estados finales.

$$+$$

- $\delta : Q \times (\Sigma \cup \{\$\})^k_\bullet \rightharpoonup Q$ es una función parcial, tal que:

  para todo $p \in Q$ y $w \in (\Sigma \cup \{\$\})^k$: $|\{\, u.v \mid \delta(p, u.v) = q \text{ y } uv = w \,\}| \leq 1$.

k-lookahead y lenguajes regulares (recordatorio)

Teorema

Para todo DFA con k-lookahead $\mathcal{A}$
se tiene que $\mathcal{L}(\mathcal{A})$ es un **lenguaje regular**.

Demostración: ejercicio.

Definición

Llamaremos un **lazy automata** a un DFA con 1-lookahead.

¿cuál es la ventaja de un lazy autómata?

Construcción de un lazy autómata a partir de $\mathcal{A}_w^{\text{det}}$

Sea $w = w_1 \dots w_m$ y $\mathcal{A}_w^{\text{det}} = (Q^{\text{det}}, \Sigma, \delta^{\text{det}}, \{0\}, F^{\text{det}})$ la determ. de $\mathcal{A}_w$.

Teorema

Para todo $S \in Q^{\text{det}}$ y $i \in \{0, 1, \dots, m\}$ se cumple que:

$$i \in S \quad \text{si, y solo si,} \quad w_1 \dots w_i \text{ es un sufijo de } w_1 \dots w_{\max(S)}.$$

Para $i \in [0, m]$, sea S_i el **único estado** en Q^{det} tal que $i = \max(S_i)$.

(¿por qué S_i es único?)

Construcción de un lazy autómata a partir de $\mathcal{A}_w^{\text{det}}$

Sea $w = w_1 \dots w_m$ y $\mathcal{A}_w^{\text{det}} = (Q^{\text{det}}, \Sigma, \delta^{\text{det}}, \{0\}, F^{\text{det}})$ la determ. de $\mathcal{A}_w$.

Teorema

Para todo $S \in Q^{\text{det}}$ y $i \in \{0, 1, \dots, m\}$ se cumple que:

$$i \in S \quad \text{si, y solo si,} \quad w_1 \dots w_i \text{ es un sufijo de } w_1 \dots w_{\max(S)}.$$

Para $i \in [0, m]$, sea S_i el **único estado** en Q^{det} tal que $i = \max(S_i)$.

Propiedad 2

Para todo $a \in \{w_1, \dots, w_m\}$ y $i \in [0, m-1]$:

1. $S_i \setminus \{i\} \in Q^{\text{det}}$.
2. $a = w_{i+1}$, entonces $\delta^{\text{det}}(S_i, a) = S_{i+1}$.
3. $a \neq w_{i+1}$, entonces $\delta^{\text{det}}(S_i, a) = \delta^{\text{det}}(S_i \setminus \{i\}, a)$.

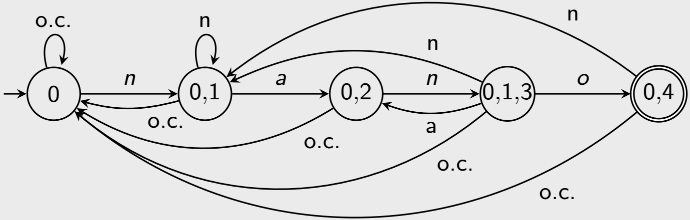
Construcción de un lazy autómata a partir de $\mathcal{A}_w^{\text{det}}$

Propiedad 2

Para todo $a \in \{w_1, \ldots, w_m\}$ y $i \in [0, m-1]$:

1. $S_i \setminus \{i\} \in Q^{\text{det}}$.
2. $a = w_{i+1}$, entonces $\delta^{\text{det}}(S_i, a) = S_{i+1}$.
3. $a \neq w_{i+1}$, entonces $\delta^{\text{det}}(S_i, a) = \delta^{\text{det}}(S_i \setminus \{i\}, a)$.

Ejemplo: palabra w = nano



Construcción de un lazy autómata a partir de $\mathcal{A}_w^{\mathrm{det}}$

Propiedad 2

Para todo $a \in \{w_1, \ldots, w_m\}$ y $i \in [0, m-1]$:

1. $S_i \setminus \{i\} \in Q^{\mathrm{det}}$.
2. $a = w_{i+1}$, entonces $\delta^{\mathrm{det}}(S_i, a) = S_{i+1}$.
3. $a \neq w_{i+1}$, entonces $\delta^{\mathrm{det}}(S_i, a) = \delta^{\mathrm{det}}(S_i \setminus \{i\}, a)$.

Demostración: ejercicio.

¿cómo podemos construir un lazy autómata usando la Propiedad 2?

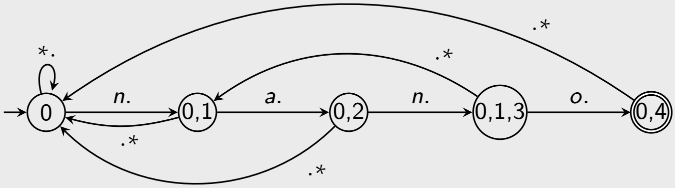
Construcción de un lazy autómata a partir de $\mathcal{A}_w^{\text{det}}$

Construcción

Se define el lazy autómata $\mathcal{A}_w^{\text{lazy}} = (Q^{\text{det}}, \Sigma, \delta^{\text{lazy}}, \{0\}, F^{\text{det}})$ tal que:

- para todo $a \neq w_1$: $\delta^{\text{lazy}}(\{0\}, a.) = \{0\}$.
- para todo $a \in \{w_1, \ldots, w_m\}$ y $i \in [0, m-1]$:
  - si $a = w_{i+1}$, entonces $\delta^{\text{lazy}}(S_i, a.) = S_{i+1}$
  - si $a \neq w_{i+1}$ y $i \neq 0$, entonces $\delta^{\text{lazy}}(S_i, .a) = S_i \setminus \{i\}$.

Ejemplo



Construcción de un lazy autómata a partir de $\mathcal{A}_w^{\text{det}}$

Construcción

Se define el lazy autómata $\mathcal{A}_w^{\text{lazy}} = (Q^{\text{det}}, \Sigma, \delta^{\text{lazy}}, \{0\}, F^{\text{det}})$ tal que:

- para todo $a \neq w_1$: $\delta^{\text{lazy}}(\{0\}, a.) = \{0\}$.
- para todo $a \in \{w_1, \ldots, w_m\}$ y $i \in [0, m-1]$:
  - si $a = w_{i+1}$, entonces $\delta^{\text{lazy}}(S_i, a.) = S_{i+1}$
  - si $a \neq w_{i+1}$ y $i \neq 0$, entonces $\delta^{\text{lazy}}(S_i, .a) = S_i \setminus \{i\}$.

Teorema

Para todo w se cumple que $\mathcal{L}(\mathcal{A}_w^{\text{det}}) = \mathcal{L}(\mathcal{A}_w^{\text{lazy}})$.

Demostración: ejercicio. (usando Propiedad 2)

¿cuántos pasos toma $\mathcal{A}_w^{\text{lazy}}$ sobre un documento d?

- Número de pasos que $\mathcal{A}_w^{\text{lazy}}$ **consume** letras $= |d|$
- Número de pasos que $\mathcal{A}_w^{\text{lazy}}$ **retrocede** $\leq |d|$
- Número de **pasos totales** de $\mathcal{A}_w^{\text{lazy}}$ $\leq 2 \cdot |d|$

Por lo tanto, la cantidad de pasos es **lineal** en $\mathcal{O}(|d|)$.

Algoritmo de Knuth-Morris-Pratt

Algoritmo

Dado una palabra w y un documento d:

- Construimos $\mathcal{A}_w^{\text{lazy}}$ desde $\mathcal{A}_w$. $\mathcal{O}(|w|)$
- Ejecutamos $\mathcal{A}_w^{\text{lazy}}$ sobre d. $\mathcal{O}(|d|)$

Tiempo del algoritmo: $\mathcal{O}(|w| + |d|)$

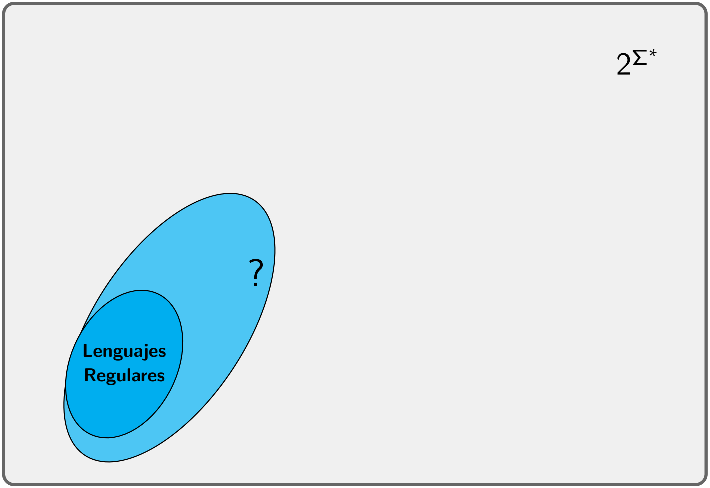
Ejercicio: demuestre como construir $\mathcal{A}_w^{\text{lazy}}$ en tiempo $\mathcal{O}(|w|)$

Outline

Algoritmo de KMP (clase anterior)

Definición de grámaticas

¿dónde estamos?



¿qué le falta a los lenguajes regulares?

Gramáticas libres de contexto

Definición

Una **gramática libre de contexto** (CFG) es una tupla:

$$\mathcal{G} = (V, \Sigma, P, S)$$

- V es un conjunto finito de **variables** o **no-terminales**.
- Σ es un alfabeto finito (o **terminales**) tal que $\Sigma \cap V = \varnothing$.
- $P \subseteq V \times (V \cup \Sigma)^*$ es un subconjunto finito de **reglas** o **producciones**.
- $S \in V$ es la **variable inicial**.

Gramáticas libres de contexto

Ejemplo

Consideré la grámatica $\mathcal{G} = (V, \Sigma, P, S)$ tal que:

- $V = \{ X, Y \}$
- $\Sigma = \{ a, b \}$
- $P = \{ (X, aXb),\ (X, Y),\ (Y, \epsilon) \}$
- $S = X$

$$\begin{array}{llll} \mathcal{G}: & X & \rightarrow & aXb \\ & X & \rightarrow & Y \\ & Y & \rightarrow & \epsilon \end{array}$$

Notación para gramáticas libres de contexto

Notación

- Para las **variables** en una gramática usaremos letras mayúsculas:

$$X, Y, Z, A, B, C, \ldots$$

- Para los **terminales** en una gramática usaremos letras minúsculas:

$$a, b, c, \ldots$$

- Para palabras en $(V \cup \Sigma)^*$ usaremos símbolos:

$$\alpha, \beta, \gamma, \ldots$$

- Para una producción $(A, \alpha) \in P$ la escribimos como:

$$A \to \alpha$$

Notación para gramáticas libres de contexto

Ejemplo anterior

Consideré la grámatica $\mathcal{G} = (V, \Sigma, P, S)$ tal que:

- $V = \{ X, Y \}$ → variables en **mayus.**
- $\Sigma = \{ a, b \}$ → letras en **minus.**
- $P = \{ X \to aXb,\ X \to Y,\ Y \to \epsilon \}$ → **producciones**
- $S = X$

$$
\begin{array}{llll}
\mathcal{G}: & X & \to & aXb \\
 & X & \to & Y \\
 & Y & \to & \epsilon
\end{array}
$$

Simplificación para gramáticas libres de contexto

Simplificación

Si tenemos un conjunto de reglas de la forma:

$$\begin{array}{ccc} X & \to & \alpha_1 \\ X & \to & \alpha_2 \\ & \cdots & \\ X & \to & \alpha_n \end{array}$$

entonces escribimos estas reglas **sucintamente** como:

$$X \quad \to \quad \alpha_1 \mid \alpha_2 \mid \cdots \mid \alpha_n$$

(recordar que: $\alpha_1, \alpha_2, \ldots, \alpha_n \in (\Sigma \cup V)^*$)

Simplificación para gramáticas libres de contexto

Ejemplo anterior

$$
\begin{array}{rccl}
\mathcal{G}: & X & \rightarrow & aXb \\
 & X & \rightarrow & Y \\
 & Y & \rightarrow & \epsilon
\end{array}
$$

Esta grámatica la escribiremos en notación **sucinta** como:

$$
\begin{array}{rccl}
\mathcal{G}: & X & \rightarrow & aXb \mid Y \\
 & Y & \rightarrow & \epsilon
\end{array}
$$

Producciones

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una CFG.

Definición

Definimos la relación $\Rightarrow \subseteq (V \cup \Sigma)^* \times (V \cup \Sigma)^*$ de **producción** tal que:

$$\alpha \cdot X \cdot \beta \Rightarrow \alpha \cdot \gamma \cdot \beta \qquad \text{si, y solo si,} \qquad (X \to \gamma) \in P$$

para todo $X \in V$ y $\alpha, \beta, \gamma \in (V \cup \Sigma)^*$.

Si $\alpha X \beta \Rightarrow \alpha \gamma \beta$ entonces decimos que

- $\alpha X \beta$ **produce** $\alpha \gamma \beta$ o
- $\alpha \gamma \beta$ **es producible** desde $\alpha X \beta$.

$\alpha X \beta \Rightarrow \alpha \gamma \beta$ es **reemplazar** γ en X en la palabra $\alpha X \beta$.

Producciones

¿cuál de las siguientes producciones son correctas?

$$\begin{aligned} \mathcal{G}: \quad X &\rightarrow aXb \mid Y \\ Y &\rightarrow \epsilon \end{aligned}$$

- $X \Rightarrow Y$?
- $aaXbb \Rightarrow aaaXbbb$?
- $aaaYbbb \Rightarrow aaaXbbb$?
- $aXaXbYX \Rightarrow aXaXbYaXb$?

Derivaciones

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una CFG.

Definición

Dada dos palabras $\alpha, \beta \in (V \cup \Sigma)^*$ decimos que α **deriva** β:

$$\alpha \overset{\star}{\Rightarrow} \beta$$

Si existe $\alpha_1, \alpha_2, \ldots, \alpha_n \in (V \cup \Sigma)^*$ tal que:

$$\alpha \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \beta$$

Derivaciones

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una CFG.

Definición

Dada dos palabras $\alpha, \beta \in (V \cup \Sigma)^*$ decimos que α **deriva** β:

$$\alpha \overset{\star}{\Rightarrow} \beta$$

con $\overset{\star}{\Rightarrow}$ es la **clausura refleja y transitiva** de $\Rightarrow$, esto es:

1. $\alpha \overset{\star}{\Rightarrow} \alpha$
2. $\alpha \overset{\star}{\Rightarrow} \beta$ si, y solo si, existe γ tal que $\alpha \overset{\star}{\Rightarrow} \gamma$ y $\gamma \Rightarrow \beta$

para todo $\alpha, \beta \in (V \cup \Sigma)^*$.

Notar que $\Rightarrow$ y $\overset{\star}{\Rightarrow}$ son relaciones entre palabras en $(V \cup \Sigma)^*$

Derivaciones

¿cuál de las siguientes derivaciones son correctas?

$$\mathcal{G}: \quad \begin{array}{lcl} X & \rightarrow & aXb \mid Y \\ Y & \rightarrow & \epsilon \end{array}$$

- $X \stackrel{\star}{\Rightarrow} aaaXbbb$?
- $aaXbb \stackrel{\star}{\Rightarrow} aaaYbb$?
- $aaXbb \stackrel{\star}{\Rightarrow} aaabbb$?

Lenguaje definido por una gramática

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una CFG.

Definición

El **lenguaje** de una grámatica $\mathcal{G}$ se define como:

$$\mathcal{L}(\mathcal{G}) = \left\{ w \in \Sigma^* \mid S \overset{\star}{\Rightarrow} w \right\}$$

$\mathcal{L}(\mathcal{G})$ son todas las palabras en Σ^* que se pueden derivar desde S.

Lenguaje definido por una gramática

¿qué palabras están en $\mathcal{L}(\mathcal{G})$?

$$\mathcal{G}: \quad \begin{array}{rcl} X & \rightarrow & aXb \mid Y \\ Y & \rightarrow & \epsilon \end{array}$$

- Como $X \overset{\star}{\Rightarrow} aaabbb$, entonces $aaabbb \in \mathcal{L}(\mathcal{G})$.
- En general, uno puede demostrar **por inducción** que:

$$\mathcal{L}(\mathcal{G}) = \left\{a^n b^n \mid n \geq 0\right\}$$

Lenguaje definido por una gramática

¿qué lenguaje define cada grámatica libre de contexto?

1. $$\mathcal{G}: \begin{array}{rcl} S & \to & XS \mid \epsilon \\ X & \to & aa \mid ab \mid ba \mid bb \end{array}$$

2. $$\mathcal{G}: \begin{array}{rcl} S & \to & S + S \mid S \times S \mid (S) \mid X \\ X & \to & 0 \mid 1 \mid \ldots \mid 9 \end{array}$$

3. $$\mathcal{G}: \quad S \to aSb \mid SS \mid \epsilon$$

Lenguaje definido por una gramática

¿cuál es una gramática para cada lenguaje?

1. $L_1 = \left\{ a^n b^n \mid n \geq 0 \right\} \cup \left\{ b^n a^n \mid n \geq 0 \right\}$

2. $L_2 = \left\{ w \in \{a, b\}^* \mid w = w^{\text{rev}} \right\}$

Lenguajes libres de contexto

Definición

Diremos que $L \subseteq \Sigma^*$ es un **lenguaje libre de contexto** ssi existe una gramática libre de contexto $\mathcal{G}$ tal que:

$$L = \mathcal{L}(\mathcal{G})$$

Ejemplos

Los siguientes son lenguajes libres de contexto:

- $L = \{a^n b^n \mid n \geq 0\}$
- $\text{Par} = \{ w \in \{a, b\}^* \mid w \text{ tiene largo par } \}$
- $\text{Pal} = \{ w \in \{a, b\}^* \mid w = w^{\text{rev}} \}$

Cierre de clase

En esta clase vimos:

- Algoritmo de Knuth-Morris-Pratt.
- Definición de gramáticas libres de contexto.

Próxima clase: Propiedades de gramáticas.