Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2223 — Teoría de Autómatas y Lenguajes Formales — 2' 2022

INTERROGACION 2

Preguntas en blanco: Preguntas o items entregados en blanco se evaluarán con 0.5 de 6 puntos.

Pregunta 1

Demuestre que existen transductores $\mathcal{T}_1$ y $\mathcal{T}_2$ sobre alfabetos Σ y Ω tal que la relación $[\![\mathcal{T}_1]\!] \cap [\![\mathcal{T}_2]\!] = \{(u,v) \in \Sigma^* \times \Omega^* \mid (u,v) \in [\![\mathcal{T}_1]\!] \wedge (u,v) \in [\![\mathcal{T}_2]\!]\}$ no es una relación racional.

Pregunta 2

Considere los siguientes lenguajes sobre el alfabeto $\{a,b,c,d\}$:

$$L_1 = \{a^i b^j c^k d^\ell \mid 2i = \ell \wedge 3j = k\}$$
$$L_2 = \{a^i b^j c^k d^\ell \mid 2i = k \wedge 3j = \ell\}$$

Uno de los lenguajes es libre de contexto y el otro no. ¿Cuál es cuál? Diga cuál lenguaje es libre de contexto y cuál no, y demuestre ambas afirmaciones.

Pregunta 3

Sean Σ y Ω alfabetos finitos. Una función $h : \Sigma^* \to \Omega^*$ *respeta la concatenación* si para todo $u, v \in \Sigma^*$ se cumple que $h(u \cdot v) = h(u) \cdot h(v)$. Para $L \subseteq \Sigma^*$, se define el lenguaje $h(L) = \{h(w) \mid w \in L\}$.

Demuestre que, si $L \subseteq \Sigma^*$ es un lenguaje libre de contexto y $h : \Sigma^* \to \Omega^*$ es una función que respeta la concatenación, entonces $h(L)$ es un lenguaje libre de contexto.

Pregunta 4

Considere el siguiente problema:

Problema:	#SUFFIX-DFA
Input:	Un DFA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ y $w = a_1 \dots a_n \in \Sigma^*$.
Output:	$\lvert\{i \in \{1,\dots,n\} \mid a_i \dots a_n \in \mathcal{L}(\mathcal{A})\}\rvert$.

Esto es, el problema #SUFFIX-DFA consiste en, dado un autómata finito determinista $\mathcal{A}$ y dado una palabra w, contar todos los sufijos de w que son aceptados por $\mathcal{A}$.

Escriba una algoritmo que resuelva #SUFFIX-DFA en tiempo $\mathcal{O}(|\mathcal{A}| \cdot |w|)$ donde $|\mathcal{A}|$ es el número de estados y transiciones de $\mathcal{A}$. Demuestre la correctitud de su algoritmo.

Formulario

Autómata finito determinista (DFA)

$\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ es un *DFA* tal que:

- Q es un conjunto finito de estados.
- Σ es el alfabeto.
- $\delta : Q \times \Sigma \to Q$ es la función de transición.
- $q_0 \in Q$ es el estado inicial.
- $F \subseteq Q$ son los estados finales.

La *ejecución* de $\mathcal{A}$ sobre $w = a_1 \dots a_n$ es una secuencia:

$$\rho : \; p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \xrightarrow{a_3} \dots \xrightarrow{a_n} p_n$$

tal que $p_0 = q_0$ y $\delta(p_i, a_{i+1}) = p_{i+1}$ para todo $i < n$. La ejecución ρ es de *aceptación* si adicionalmente $p_n \in F$.

$\mathcal{A}$ *acepta* w si la ejecución de $\mathcal{A}$ sobre w es de aceptación.

El *lenguaje aceptado por* $\mathcal{A}$ se define como:

$$\mathcal{L}(\mathcal{A}) \;=\; \{w \in \Sigma^* \mid \mathcal{A} \text{ acepta } w\}.$$

Autómata finito no-determinista (NFA)

Un *NFA* $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ es igual a un DFA pero:

- $\Delta \subseteq Q \times \Sigma \times Q$ es la relación de transición.
- $I \subseteq Q$ son los estados iniciales.

Una *ejecución* de $\mathcal{A}$ sobre $w = a_1 \dots a_n$ es una secuencia:

$$\rho : \; p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \xrightarrow{a_3} \dots \xrightarrow{a_n} p_n$$

tal que $p_0 = q_0$ y $(p_i, a_{i+1}, p_{i+1}) \in \Delta$ para todo $i < n$. La ejecución ρ es de *aceptación* si adicionalmente $p_n \in F$.

$\mathcal{A}$ *acepta* w si existe una ejecución de $\mathcal{A}$ sobre w que es de aceptación.

El *lenguaje aceptado por* $\mathcal{A}$ se define como:

$$\mathcal{L}(\mathcal{A}) \;=\; \{w \in \Sigma^* \mid \mathcal{A} \text{ acepta } w\}.$$

Evaluación NFA on-the-fly

Para un NFA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ y $w = a_1 \dots a_n$.

Function `eval-NFAonthefly` *($\mathcal{A}$, w)*
- $S \leftarrow I$
- **for** $i = 1$ **to** n **do**
  - $S_{\text{old}} \leftarrow S$
  - $S \leftarrow \varnothing$
  - **foreach** $p \in S_{old}$ **do**
    - $S \leftarrow S \cup \{q \mid (p, a_i, q) \in \Delta\}$
- **return check** $(S \cap F \neq \varnothing)$

Transductores

$\mathcal{T} = (Q, \Sigma, \Omega, \Delta, I, F)$ es un *transductor* tal que:

- Q es un conjunto finito de estados.
- Σ es el alfabeto de input.
- Ω es el alfabeto de output.
- $\Delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times (\Omega \cup \{\epsilon\}) \times Q$ es la relación de transición.
- $I \subseteq Q$ es un conjunto de estados iniciales.
- $F \subseteq Q$ es el conjunto de estados finales.

Un par $(q, u, v) \in Q \times \Sigma^* \times \Omega^*$ es una configuración de $\mathcal{T}$. Definimos $\vdash_{\mathcal{T}}$ como la relación de *siguiente-paso* entre configuraciones de $\mathcal{T}$:

$$(p, u_1, v_1) \;\vdash_{\mathcal{T}}\; (q, u_2, v_2)$$

si, y solo si, existe $(p, a, b, q) \in \Delta$ tal que $u_1 = a \cdot u_2$ y $v_2 = v_1 \cdot b$. Se define $\vdash_{\mathcal{T}}^*$ como la clausura *refleja* y *transitiva* de $\vdash_{\mathcal{T}}$. Se dice que $\mathcal{T}$ *entrega* v con input u si existe una configuración (q_0, u, ϵ) con $q_0 \in I$ y una configuración (q_f, ϵ, v) con $q_f \in F$ tal que $(q_0, u, \epsilon) \vdash_{\mathcal{T}}^* (q_f, \epsilon, v)$. Un transductor $\mathcal{T}$ define la *relación* $[\![\mathcal{T}]\!] \subseteq \Sigma^* \times \Omega^*$:

$$(u, v) \in [\![\mathcal{T}]\!] \quad \text{si, y solo si,} \quad \mathcal{T} \text{ entrega } v \text{ con input } u.$$

Se dice que $R \subseteq \Sigma^* \times \Omega^*$ es una *relación racional* si existe un transductor $\mathcal{T}$ tal que $[\![\mathcal{T}]\!] = R$.

Gramáticas libres de contexto

$\mathcal{G} = (V, \Sigma, P, S)$ es una *gramática libre de contexto* tal que:

- V es un conjunto variables.
- Σ es un alfabeto finito con $\Sigma \cap V = \varnothing$.
- $P \subseteq V \times (V \cup \Sigma)^*$ es un subconjunto finito de reglas.
- $S \in V$ es la variable inicial.

Definimos la relación $\Rightarrow \subseteq (V \cup \Sigma)^* \times (V \cup \Sigma)^*$ tal que $\alpha X \beta \Rightarrow \alpha\gamma\beta$ si, y solo si, $(X \to \gamma) \in P$ para todo $X \in V$ y $\alpha, \beta, \gamma \in (V \cup \Sigma)^*$. Se define $\overset{\star}{\Rightarrow}$ como la clausura *refleja* y *transitiva* de $\Rightarrow$. El *lenguaje aceptado* por $\mathcal{G}$ se define como:

$$\mathcal{L}(\mathcal{G}) \;=\; \left\{\, w \in \Sigma^* \;\middle|\; S \overset{\star}{\Rightarrow} w \,\right\}.$$

Se dice que $L \subseteq \Sigma^*$ es un *lenguaje libre de contexto* si existe una gramática $\mathcal{G}$ tal que $\mathcal{L}(\mathcal{G}) = L$.

Forma normal de Chomsky

Una gramática $\mathcal{G}$ esta en *forma normal de Chomsky* (CNF) si todas sus reglas son de la forma $X \to YZ$ o $X \to a$.

Lema de bombeo para lenguajes libres de contexto´

Sea $L \subseteq \Sigma^*$. Si L es libre de contexto entonces existe un $N > 0$ tal que para toda palabra $z \in L$ con $|z| \geq N$ existe una descomposición $z = uvwxy$ con $vx \neq \epsilon$ y $|vwx| \leq N$ tal que $u \cdot v^i \cdot w \cdot x^i \cdot y \in L$ para todo $i \geq 0$.