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

IIC2223 – Teoría de Autómatas y Lenguajes Formales – 2' 2024
IIC2224 – Autómatas y Compiladores

Tarea 4

Publicación: Viernes 18 de octubre.
Entrega: **Jueves 24 de octubre hasta las 23:59 horas**.

Indicaciones

- Debe entregar una solución para cada pregunta (sin importar si está en blanco).
- Cada solución debe estar escrita en LaTeX. No se aceptarán tareas escritas a mano ni en otro sistema de composición de texto.
- Responda cada pregunta en una hoja separada y ponga su nombre en cada hoja de respuesta.
- Debe entregar una copia digital por el buzón del curso, antes de la fecha/hora de entrega.
- **Se penalizará con 1 punto en la nota final de la tarea por cada regla que no se cumpla.**
- La tarea es individual.

Pregunta 1

Un transductor $\mathcal{T} = (Q, \Sigma, \Omega, \Delta, I, F)$ se dice *síncrono* si $\Delta \subseteq Q \times \Sigma \times \Omega \times Q$. En otras palabras, no tiene ϵ-transiciones, ni transiciones que producen ϵ al leer una letra. En particular, es fácil ver que si $\mathcal{T}$ es síncrono y $(u, v) \in [\![\mathcal{T}]\!]$, entonces $|u| = |v|$. Por último, para todo $u = a_1 \dots a_n \in \Sigma^*$ y $v = b_1 \dots b_n \in \Omega^*$ se define la palabra $u \times v = (a_1, b_1) \dots (a_n, b_n)$ sobre el alfabeto $\Sigma \times \Omega$.

1. Para una relación $R \subseteq \Sigma^* \times \Omega^*$ tal que $|u| = |v|$ para todo $(u, v) \in R$, se define el lenguaje:

   $$R^{\times} = \{u \times v \mid (u, v) \in R\}.$$

   Demuestre que para todo transductor síncrono $\mathcal{T}$ se tiene que $[\![\mathcal{T}]\!]^{\times}$ es un lenguaje regular.

2. Para una palabra $w = a_1 \dots a_{n-1} a_n$, se define la palabra reversa de w como $w^{\text{rev}} = a_n a_{n-1} \dots a_1$. Considere la relación $\text{Rev} = \{(w, w^{\text{rev}}) \mid w \in \Sigma^*\}$. Demuestre que NO existe un transductor síncrono $\mathcal{T}$ tal que $\text{Rev} = [\![\mathcal{T}]\!]$.

Solución

Problema 1.1. Queremos demostrar que si $\mathcal{T}$ es síncronno, $[\![\mathcal{T}]\!]$ es regular. Sea $\mathcal{T} = (Q, \Sigma, \Omega, \Delta, I, F)$ síncrono, con $\Delta \subseteq Q \times \Sigma \times \Omega \times Q$. Sea un NFA A_T tal que —$A_T = (Q, \Sigma \times \Omega, \Delta_T, I, F)$ con $\Delta_T = \{(p, (a, b), q) \mid (p, a, b, q) \in \Delta\}$.

P.D. $L(A_T) = [\![\mathcal{T}]\!]^X$.

Demostración: $\supseteq$

Sea $(a_1, b_1)...(a_n, b_n) \in [\![\mathcal{T}]\!]^X$. Por definición, $(a_1...a_n, b_1...b_n) \in [\![\mathcal{T}]\!]$. Como $\mathcal{T}$ es síncrono, existe una ejecución de $\mathcal{T}$ que acepta $(a_1...a_n, b_1...b_n)$. Sea la ejecicion de aceptación:

$$\rho: \; p_0 \xrightarrow{a_1/b_1} \dots \xrightarrow{a_n/b_n} p_n$$

Por construcción, podemos armar una ejecución de A_T sobre $(a_1, b_1)...(a_n, b_n)$ de la siguiente forma:

$$\rho_2: \; p_0 \xrightarrow{(a_1,b_1)} \dots \xrightarrow{(a_n,b_n)} p_n$$

Por lo tanto, $(a_1, b_1)...(a_n, b_n) \in L(A_T)$.

Demostración: $\subseteq$

El mismo procedimiento puede ser seguido de manera reversa gracias a la manera en la que se construyó el autómata.

Distribución de puntaje

- 1 punto por plantear y definir el lenguaje del autómata.
- 1 punto por plantear y definir el conjunto Δ del autómata.
- 1 punto por cada lado de la demostración (2 en total por la demostración de la igualdad).

Problema 1.2. Por contradicción. Supongamos existe un transductor síncrono tal que $[\![\mathcal{T}]\!]$ = Rev. Por el problema 1.1, Rev^X es regular.

P.D. Rev^X no es regular.

Notar que Rev^X es de la forma:

$$\text{Rev}^X = \{(a_1, a_n)(a_2, a_{n-1})...(a_n, a_1) \mid a_i \in \Sigma\}.$$

Por bombeo, sea $N > 0$ cualquiera. Consideremos $(a^N b^N, b^N a^N) \in$ Rev. De lo anterior, se tiene que $(a,b)^N(b,a)^N \in \text{Rev}^X$ y podemos elegir $x = \varepsilon$, $y = (a,b)^N$, $z = (b,a)^N$. Suponemos y tal que $y = (a,b)^N = (a,b)^k(a,b)^l(a,b)^m$ con $k + l + m = N$ y $l > 0$. Elegimos $i = 2$ y tenemos $(a,b)^k(a,b)^{2l}(a,b)^m(b,a)^N = (a,b)^{N+l}(b,a)^N$. Pero vemos que $(a,b)^{N+l}(b,a)^N \notin \text{Rev}^X$, ya que $(a^{N+l}b^N, b^{N+l}a^N) \notin$ Rev.

Distribución de puntaje

- 1 punto por plantear la contradicción (o el tipo de demostración que se vaya a utilizar).
- 1 punto por utilizar el ejercicio 1.1 para comenzar la demostración.
- 1 punto por plantear la palabra bombeable.
- 1 punto por realizar bien el bombeo y llegar a la contradicción.

Pregunta 2

Para cada uno de los siguientes lenguajes, muestre una gramática libre de contexto que lo defina y explique su correctitud. No es necesario demostrar su correctitud, pero si explicar de manera precisa porque la gramática propuesta cumple con lo solicitado.

1. Todas las formulas proposicionales con una variable p y constantes 0 (false) y 1 (true).
2. Todas las formulas proposicionales con una variable p y constantes 0 y 1 que son *tautologías*.

Notar que el alfabeto de ambos lenguajes es $\{p, 0, 1, \neg, \wedge, \vee, (,)\}$. Por ejemplo, $(\neg(p) \wedge 1)$ y $((1 \wedge \neg(p)) \vee p)$ son palabras en el primer lenguaje. En cambio, $(\neg p \wedge 1)$ o $0 \wedge \neg p \vee p$ no lo son. Notar que cada operación $\neg$, $\wedge$, o $\vee$ de la formula tiene que estar entre paréntesis.

Solución

Problema 2.1. Para este lenguaje podemos definir la siguiente gramática libre de contexto $\mathcal{G}_1$:

$$\begin{array}{lrl} \mathcal{G}_1: & S \to & p \mid 0 \mid 1 \qquad \text{casos base} \\ & \mid & (S \wedge S) \mid (S \vee S) \qquad \text{casos and y or} \\ & \mid & \neg(S) \qquad \text{caso not} \end{array}$$

La gramática anterior considera los casos base $p, 0, 1$ que nos permiten formar formulas proposicionales utilizando la variable p y las constantes 0 y 1 como se solicita en la pregunta. Luego los casos *and, or y not* nos permiten crear formulas proposicionales más complejas utilizando los símbolos $\wedge, \vee, \neg$ respectivamente.

Distribución de puntaje

- 1 punto por casos bases
- 1 punto por casos and y or
- 1 punto por caso not
- 1 punto por incluir correctamente los paréntesis

Problema 2.2. Para definir la gramática libre de contexto $\mathcal{G}_2$ para este problema, definiremos primero una variable X_{b_0,b_1} para cada $b_0, b_1 \in 0, 1$. Cada variable X_{b_0,b_1} representa a una fórmula φ que al ser evaluada p con el valor 0, φ toma el valor b_0 y al ser evaluada p con el valor 1, φ toma el valor b_1.

Por simplicidad y para hacer más sucinta la gramática, consideramos que X_{*,b_1} o $X_{b_0,*}$ es equivalente a decir que no nos importa que valor hay en la posición *. Por ejemplo $X_{0,*}$ representa a las variables $X_{0,0}$ y $X_{0,1}$. Si tenemos una producción $X_{b_0,b_1} \to X_{*,b_1}$, esta representa a las dos producciones $X_{b_0,b_1} \to X_{0,b_1}$ y $X_{b_0,b_1} \to X_{0,b_1}$.

Con esta definición, comenzamos formando las siguientes producciones, que serán los casos base de nuestra gramática $\mathcal{G}_2$:

$$\begin{aligned} X_{0,0} &\to 0 \\ X_{0,1} &\to p \\ X_{1,1} &\to 1 \end{aligned}$$

Luego, podemos definir los casos base recursivos de la siguiente manera:

$X_{0,0}$	$\to$	$\neg(X_{1,1})$	representa a la constante 0
$X_{0,1}$	$\to$	$\neg(X_{1,0})$	representa a la variable p
$X_{1,0}$	$\to$	$\neg(X_{0,1})$	representa a la "variable"$\neg p$
$X_{1,1}$	$\to$	$\neg(X_{0,0})$	representa a la consante 1

Juntando todos los casos base, obtenemos lo siguiente:

$$\begin{aligned} X_{0,0} &\to 0 \mid \neg(X_{1,1}) \\ X_{0,1} &\to p \mid \neg(X_{1,0}) \\ X_{1,0} &\to \neg(X_{0,1}) \\ X_{1,1} &\to 1 \mid \neg(X_{0,0}) \end{aligned}$$

Con estos casos base, generalizaremos las operaciones *and y or* de la siguiente manera:

- **AND**

$$\begin{aligned} X_{0,0} &\to (X_{0,0} \wedge X_{*,*}) \mid (X_{*,*} \wedge X_{0,0}) \mid (X_{0,*} \wedge X_{*,0}) \mid (X_{*,0} \wedge X_{0,*}) \\ X_{0,1} &\to (X_{0,1} \wedge X_{*,1}) \mid (X_{*,1} \wedge X_{0,1}) \\ X_{1,0} &\to (X_{1,0} \wedge X_{1,*}) \mid (X_{1,*} \wedge X_{1,0}) \\ X_{1,1} &\to (X_{1,1} \wedge X_{1,1}) \end{aligned}$$

- **OR**

$$\begin{aligned} X_{0,0} &\to (X_{0,0} \vee X_{0,0}) \\ X_{0,1} &\to (X_{0,1} \vee X_{0,*}) \mid (X_{0,*} \vee X_{0,1}) \\ X_{1,0} &\to (X_{1,0} \vee X_{*,0}) \mid (X_{*,0} \vee X_{1,0}) \\ X_{1,1} &\to (X_{1,1} \vee X_{*,*}) \mid (X_{*,*} \vee X_{1,1}) \mid (X_{1,*} \vee X_{*,1}) \mid (X_{*,1} \vee X_{1,*}) \end{aligned}$$

Con las variables X_{b_0,b_1} definidas, para obtener todas las tautologías basta con definir que la variable inicial de la gramática $\mathcal{G}_2$ es $X_{1,1}$. Debido a nuestra construcción $X_{1,1}$ solo producirá fórmulas que sea tautologías y considerará todas las operaciones, es decir, AND, OR y NOT.

Distribución de puntaje

- 1 punto por casos bases
- 1 punto por casos base recursivos
- 1 punto por caso and
- 1 punto por caso or

Evaluación y puntajes de la tarea

Cada item de cada pregunta se evaluará con un puntaje de 0, 1, 2, 3 o 4 puntos. Todas las preguntas tienen la misma ponderación en la nota final y cada item tiene la misma ponderación en cada pregunta.