NOMBRE: Benjamín Farías Valdés

N.ALUMNO: 17642531

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

IIC2223 — Teoría de Autómatas y Lenguajes Formales — 2' 2021

Tarea 3

Pregunta 1

El algoritmo propuesto es el siguiente:

Algorithm 1 $SEP(\mathcal{A}, w)$

$S = I$
$n = |w|$
for $i = 1$ to n **do**
  $S_{old} = S$
  $S = \varnothing$
  for each $p \in S_{old}$ **do**
    $S = S \cup \{q | (p, a_i, q) \in \Delta\}$
  end for
  if check($S \cap F \neq \varnothing$) **and** $i < n$ **then**
    $S = I$
  end if
end for
return check($S \cap F \neq \varnothing$)

Para entender el algoritmo, primero debemos notar ciertas propiedades de la separabilidad de palabras:

- Se tienen 2 condiciones de separabilidad, una fuerte y una débil. La fuerte ocurre cuando la palabra $w \in L$ al mismo tiempo que **todos sus prefijos** u satisfacen $u \notin L$. La débil consiste en que la palabra w se pueda dividir en 2 subpalabras u y v tal que **ambas sean separables.**
- Si una subpalabra u es separable bajo la condición fuerte, entonces cualquier palabra que la tenga como prefijo **no podrá ser separable bajo la condición fuerte,** ya que existirá un prefijo (la palabra u) que pertenece al lenguaje. Esto significa que **toda palabra w que tenga de prefijo a u solo podrá ser separable bajo la condición débil.**
- Si una palabra w se divide una cantidad arbitraria de veces, eventualmente las subpalabras van a satisfacer la condición fuerte, o bien, no podrán ser divididas (tendrán largo 1), y por lo tanto deberán satisfacer de igual manera la condición fuerte para que las palabras compuestas por ellas puedan satisfacer la condición débil. **A todas estas subpalabras les llamaremos mínimas.**

- De lo anterior, podemos notar que **toda palabra que satisface la condición débil debe poder dividirse de forma recursiva en subpalabras z, tal que eventualmente todas estas subpalabras z satisfagan la condición fuerte.** De otro modo no será posible que la palabra padre de algún z cumpla con que ambas subpalabras hijas sean separables (ya que en ese caso z no puede ser dividida y no cumple con la condición fuerte).

Considerando toda la información anterior, se llega a que para resolver este problema de forma eficiente se deben seguir estos pasos:

1. Se lee la palabra de input w y se revisa letra por letra si el prefijo p acumulado pertenece al lenguaje. Si se llega a un prefijo p que pertenece, entonces sabemos que este prefijo **es una subpalabra que cumple con ser separable** (ya que antes no había ningún prefijo que perteneciera al lenguaje) y además la palabra w deberá ser separable bajo la condición débil (ya que se encontró un prefijo que evita que se cumpla la condición fuerte). En el caso de que no hayan prefijos que pertenecen al lenguaje, la palabra w deberá satisfacer la condición fuerte (ya que si se divide nunca será posible que la subpalabra izquierda pertenezca al lenguaje).

2. Todas las letras que vengan a continuación del prefijo p **deberán eventualmente ser parte de una subpalabra mínima distinta a** p, ya que de otro modo no podrían satisfacer la condición fuerte al tener a p como prefijo. Entonces, a partir de la letra siguiente a la última de p, se comienza a revisar la pertenencia al lenguaje desde 0 (como si fuera una palabra nueva).

3. Si se repiten estos pasos hasta leer la palabra w completa, entonces **la última subpalabra revisada indicará si se logró la separabilidad. Si esta subpalabra es mínima** (satisface la condición fuerte de separabilidad), entonces se tiene que la palabra w se puede particionar en subpalabras mínimas, de forma que todas sean separables bajo la condición fuerte y al combinarlas se formen palabras separables bajo la condición débil hasta llegar a w. En cambio, **si la última subpalabra no es mínima,** significa que sin importar las divisiones realizadas, la subpalabra derecha nunca podrá ser separable (ya que o tendrá prefijos o no pertenecerá al lenguaje).

La implementación se realizó de la siguiente forma:

1. Se utiliza la idea de NFA *on-the-fly* para **evaluar letra por letra la ejecución del autómata no determinista.** En cada letra se revisa si es que el conjunto de estados obtenidos contiene alguno final, en cuyo caso sabemos que **apareció una subpalabra mínima y por lo tanto se devuelve el autómata a sus estados iniciales** para poder buscar la siguiente subpalabra mínima desde el principio.

2. Al pasar por todas las letras del input, se revisa al final si es que la última subpalabra revisada es mínima o no. **En caso de que no lo sea se retorna FALSE,** ya que fue imposible separar al input en subpalabras mínimas tales que todas sean separables bajo la condición fuerte. **En caso de que sea mínima se retorna TRUE,** ya que se logró particionar a la palabra de input en distintas subpalabras mínimas independientes.

En términos de la complejidad, el algoritmo es prácticamente equivalente a NFA *on-the-fly*, añadiendo solamente la condición para chequear que se haya encontrado una subpalabra mínima. Este chequeo corresponde a una intersección de conjuntos, la que en el peor caso debe pasar por todos los estados finales de $\mathcal{A}$, es decir, ser $O(|F|)$ (lo que es a su vez $O(|A|)$). En resumen, se itera sobre cada letra de la palabra w, y en cada iteración se generan los posibles estados siguientes (el costo de esto es $|\mathcal{A}|$ en el peor caso) más las intersecciones, por lo que **la complejidad final de este algoritmo es:**

$$O(|w| \cdot (|\mathcal{A}| + |F|)) = O(|w| \cdot (c \cdot |\mathcal{A}|)) = O(|\mathcal{A}| \cdot |w|)$$

NOMBRE: Benjamín Farías Valdés

N.ALUMNO: 17642531

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

IIC2223 — Teoría de Autómatas y Lenguajes Formales — 2' 2021

Tarea 3

Pregunta 2

1)

VERDADERO

Como R es una relación racional, tenemos que existe un transductor $\mathcal{T}_r = (Q, \Sigma, \Omega, \Delta, I, F)$ que la define. Se construye un nuevo transductor $\mathcal{T}_r^{-1}$, definido a continuación:

$$\mathcal{T}_r^{-1} := (Q, \Omega, \Sigma, \Delta^{-1}, I, F)$$
$$\Delta^{-1} := \{(p, b, a, q) | (p, a, b, q) \in \Delta\}$$

Este transductor es similar a $\mathcal{T}_r$, pero tiene las siguientes **diferencias:**

- Los alfabetos de input y output están intercambiados, puesto que por definición los pares de la relación R^{-1} pertenecen a $\Omega^* \times \Sigma^*$.
- Las transiciones son al revés, es decir, si en $\mathcal{T}_r$ al leer una a se podía escribir una b en el output, entonces en $\mathcal{T}_r^{-1}$ al leer una b se puede escribir una a en el output.

La intuición indica que este transductor es el inverso de $\mathcal{T}_r$, ya que dada una palabra de output permite reconstruir los posibles inputs ingresados. Se requiere demostrar que $\mathcal{T}_r^{-1}$ define a la relación R^{-1}, es decir, $(u, v) \in R^{-1} \longleftrightarrow \mathcal{T}_r^{-1}$ entrega v con input u.

- $(u, v) \in R^{-1} \longrightarrow \mathcal{T}_r^{-1}$ entrega v con input u:
  Por construcción de R^{-1}, sabemos que $(v, u) \in R$, por lo que $\mathcal{T}_r$ entrega u con input v. Esto significa que existe una configuración **inicial** (q_1, v, ϵ) y una configuración **final** (q_f, ϵ, u) tal que se cumple lo siguiente:

$$(q_1, v, \epsilon) \vdash^*_{\mathcal{T}_r} (q_f, \epsilon, u)$$

  Expandiendo la clausura:

$$(q_1, v, \epsilon) \vdash_{\mathcal{T}_r} ... \vdash_{\mathcal{T}_r} (q_f, \epsilon, u)$$

Ahora, por definición de la relación de **siguiente paso** para $\mathcal{T}_r$, podemos observar que se tiene que cumplir lo siguiente entre cada par de configuraciones $(i, i+1)$:

$$(q_i, v_i, u_i) \vdash_{\mathcal{T}_r} (q_{i+1}, v_{i+1}, u_{i+1})$$
$$v_i = a \cdot v_{i+1}$$
$$u_{i+1} = u_i \cdot b$$
$$(q_i, a, b, q_{i+1}) \in \Delta$$

Ahora ejecutaremos $\mathcal{T}_r^{-1}$ con input u. Como $\mathcal{T}_r^{-1}$ tiene el mismo conjunto de estados iniciales que $\mathcal{T}_r$, existe una ejecución donde parte desde el mismo estado q_1 (asumimos que $|u| \geq n$):

$$(q_1, u, \epsilon) \vdash_{\mathcal{T}_r^{-1}} \dots \vdash_{\mathcal{T}_r^{-1}} (q_n, \epsilon, w)$$

En cada paso de esta ejecución se satisface lo siguiente:

$$(q_i, u_i, w_i) \vdash_{\mathcal{T}_r^{-1}} (q_{i+1}, u_{i+1}, w_{i+1})$$
$$u_i = c \cdot u_{i+1}$$
$$w_{i+1} = w_i \cdot d$$
$$(q_i, c, d, q_{i+1}) \in \Delta^{-1}$$

Por construcción de la relación de transición de $\mathcal{T}_r^{-1}$, se desprende que $(q_i, d, c, q_{i+1}) \in \Delta$. De esto se pueden concluir 2 cosas:

1. Como $w_{i+1} = w_i \cdot d$ (por $\mathcal{T}_r^{-1}$) y $v_i = d \cdot v_{i+1}$ (por $\mathcal{T}_r$), se tiene finalmente que $w = v$, ya que en cada paso se agrega al output la misma letra que se lee de input **en la ejecución de $\mathcal{T}_r$ anteriormente mostrada.**
2. Como los estados de cada transición de estas ejecuciones **son los mismos para ambos transductores** (ya que por construcción sabemos que existe ese camino de configuraciones posibles), entonces $q_n = q_f$ y se tiene que q_n es un estado final.

De lo anterior, al contraer la ejecución:

$$(q_1, u, \epsilon) \vdash^*_{\mathcal{T}_r^{-1}} (q_f, \epsilon, v)$$

Por lo tanto, llegamos a que $\mathcal{T}_r^{-1}$ entrega v con input u.

- $\mathcal{T}_r^{-1}$ entrega v con input $u \longrightarrow (u, v) \in R^{-1}$:

  Como $\mathcal{T}_r^{-1}$ entrega v con input u, se sabe que existe una configuración **inicial** (q_1, u, ϵ) y una configuración **final** (q_f, ϵ, v) tal que se cumple lo siguiente:

$$(q_1, u, \epsilon) \vdash^*_{\mathcal{T}_r^{-1}} (q_f, \epsilon, v)$$

En cada paso de esta ejecución se satisface lo siguiente:

$$(q_i, u_i, v_i) \vdash_{\mathcal{T}_r^{-1}} (q_{i+1}, u_{i+1}, v_{i+1})$$
$$u_i = a \cdot u_{i+1}$$
$$v_{i+1} = v_i \cdot b$$
$$(q_i, a, b, q_{i+1}) \in \Delta^{-1}$$

Ahora ejecutaremos $\mathcal{T}_r$ con input v. Como $\mathcal{T}_r$ tiene el mismo conjunto de estados iniciales que $\mathcal{T}_r^{-1}$, existe una ejecución donde parte desde el mismo estado q_1 (asumimos que $|v| \geq n$):

$$(q_1, v, \epsilon) \vdash_{\mathcal{T}_r} \ldots \vdash_{\mathcal{T}_r} (q_n, \epsilon, w)$$

En cada paso de esta ejecución se satisface lo siguiente:

$$(q_i, v_i, w_i) \vdash_{\mathcal{T}_r} (q_{i+1}, v_{i+1}, w_{i+1})$$
$$v_i = c \cdot v_{i+1}$$
$$w_{i+1} = w_i \cdot d$$
$$(q_i, c, d, q_{i+1}) \in \Delta$$

Por construcción de la relación de transición de $\mathcal{T}_r^{-1}$, se desprende que $(q_i, d, c, q_{i+1}) \in \Delta^{-1}$. De esto se pueden concluir 2 cosas:

1. Como $w_{i+1} = w_i \cdot d$ (por $\mathcal{T}_r$) y $u_i = d \cdot u_{i+1}$ (por $\mathcal{T}_r^{-1}$), se tiene finalmente que $w = u$, ya que en cada paso se agrega al output la misma letra que se lee de input **en la ejecución de $\mathcal{T}_r^{-1}$ anteriormente mostrada.**
2. Como los estados de cada transición de estas ejecuciones **son los mismos para ambos transductores** (ya que por construcción sabemos que existe ese camino de configuraciones posibles), entonces $q_n = q_f$ y se tiene que q_n es un estado final.

De lo anterior, al contraer la ejecución:

$$(q_1, v, \epsilon) \vdash^*_{\mathcal{T}_r} (q_f, \epsilon, u)$$

Por lo tanto, llegamos a que $\mathcal{T}_r$ entrega u con input v y $(v, u) \in R$ por definición de relación racional.

Finalmente, como $(v, u) \in R$ entonces $(u, v) \in R^{-1}$ por enunciado.

Queda demostrado que $(u, v) \in R^{-1} \longleftrightarrow \mathcal{T}_r^{-1}$ entrega v con input u, por lo que R^{-1} **es una relación racional.**

2)

VERDADERO

Como R y S son relaciones racionales, tenemos que existen los siguientes 2 transductores que las definen:

$$\mathcal{T}_r = (Q_r, \Sigma, \Omega, \Delta_r, I_r, F_r)$$
$$\mathcal{T}_s = (Q_s, \Omega, \Gamma, \Delta_s, I_s, F_s)$$

Se construye un nuevo transductor $\mathcal{T}_{rs}$, definido a continuación:

$$\mathcal{T}_{rs} := (Q_r \times Q_s, \Sigma, \Gamma, \Delta_{rs}, I_r \times I_s, F_r \times F_s)$$
$$\Delta_{rs} := \{((p_r, p_s), a, c, (q_r, q_s)) | \exists b \in \Omega.(p_r, a, b, q_r) \in \Delta_r \wedge (p_s, b, c, q_s) \in \Delta_s\}$$

Este transductor se encarga de ejecutar los transductores $\mathcal{T}_r$ y $\mathcal{T}_s$ en paralelo y entregar el output de la composición de ambos. Para lograr esto, por cada letra b que $\mathcal{T}_r$ escribe en su output, este nuevo transductor escribirá la letra c que $\mathcal{T}_s$ entrega al leer b. Además, para que efectivamente se cumpla con la relación de composición, debemos asegurar que tanto $\mathcal{T}_r$ como $\mathcal{T}_s$ terminen en un estado final, por lo que el conjunto de estados finales de $\mathcal{T}_{rs}$ corresponde a todos los pares de estados finales de $\mathcal{T}_r$ y $\mathcal{T}_s$.

Debemos demostrar que el transductor $\mathcal{T}_{rs}$ define a la relación $R \circ S$, es decir, $(u, v) \in R \circ S \longleftrightarrow \mathcal{T}_{rs}$ entrega v con input u.

- $(u, v) \in R \circ S \longrightarrow \mathcal{T}_{rs}$ entrega v con input u:

  Por construcción de $R \circ S$, sabemos que existe una palabra $w \in \Omega^*$ tal que $(u, w) \in R \wedge (w, v) \in S$. Esto significa que $\mathcal{T}_r$ entrega w con input u y $\mathcal{T}_s$ entrega v con input w. Estas ejecuciones se pueden ver de la forma siguiente (análogo al ejercicio anterior):

  $$\rho_r : (q_1, u, \epsilon) \vdash_{\mathcal{T}_r} ... \vdash_{\mathcal{T}_r} (q_f, \epsilon, w)$$

  $$\rho_s : (p_1, w, \epsilon) \vdash_{\mathcal{T}_s} ... \vdash_{\mathcal{T}_s} (p_f, \epsilon, v)$$

  En cada paso de estas ejecuciones se satisface lo siguiente:

  $$(q_i, u_i, w_i) \vdash_{\mathcal{T}_r} (q_{i+1}, u_{i+1}, w_{i+1})$$
  $$u_i = a \cdot u_{i+1}$$
  $$w_{i+1} = w_i \cdot b$$
  $$(q_i, a, b, q_{i+1}) \in \Delta_r$$

  $$(p_i, w_i, v_i) \vdash_{\mathcal{T}_s} (p_{i+1}, w_{i+1}, v_{i+1})$$
  $$w_i = c \cdot w_{i+1}$$
  $$v_{i+1} = v_i \cdot d$$
  $$(p_i, c, d, p_{i+1}) \in \Delta_s$$

  Como $\mathcal{T}_{rs}$ ejecuta ambos en paralelo, existe la siguiente ejecución en la que comienza en el par de estados iniciales (q_1, p_1) y con input u:

  $$\rho_{rs} : ((q_1, p_1), u, \epsilon) \vdash_{\mathcal{T}_{rs}} ... \vdash_{\mathcal{T}_{rs}} ((q_n, p_n), \epsilon, x)$$

En cada paso de esta ejecución se satisface lo siguiente:

$$((q_i, p_i), u_i, x_i) \vdash_{\mathcal{T}_{rs}} ((q_{i+1}, p_{i+1}), u_{i+1}, x_{i+1})$$
$$u_i = a \cdot u_{i+1}$$
$$x_{i+1} = x_i \cdot c$$
$$((q_i, p_i), a, c, (q_{i+1}, p_{i+1})) \in \Delta_{rs}$$

Por construcción de la relación de transición de $\mathcal{T}_{rs}$, se desprende que $\exists b \in \Omega$ tal que $(q_i, a, b, q_{i+1}) \in \Delta_r$ y $(p_i, b, c, p_{i+1}) \in \Delta_s$. De esto se pueden concluir 2 cosas:

1. Como $x_{i+1} = x_i \cdot c$ (por $\mathcal{T}_{rs}$) y $v_{i+1} = v_i \cdot c$ (por $\mathcal{T}_s$), se tiene finalmente que $x = v$, ya que en cada paso se agrega al output la misma letra que **en la ejecución de $\mathcal{T}_s$ anteriormente mostrada.**
2. Como los pares de estados de cada transición de esta ejecución **son los mismos de los transductores $\mathcal{T}_r$ y $\mathcal{T}_s$** en las ejecuciones de más arriba, entonces $(q_n, p_n) = (q_f, p_f)$ y se tiene que (q_n, p_n) es un par de estados finales, y por lo tanto es final para $\mathcal{T}_{rs}$.

De lo anterior, al contraer la ejecución:

$$((q_1, p_1), u, \epsilon) \vdash^*_{\mathcal{T}_{rs}} ((q_f, p_f), \epsilon, v)$$

Por lo tanto, llegamos a que $\mathcal{T}_{rs}$ entrega v con input u.

- $\mathcal{T}_{rs}$ entrega v con input $u \longrightarrow (u, v) \in R \circ S$:

  Como $\mathcal{T}_{rs}$ entrega v con input u, se sabe que existe una configuración **inicial** $((q_1, p_1), u, \epsilon)$ y una configuración **final** $((q_f, p_f), \epsilon, v)$ tal que se cumple lo siguiente:

$$((q_1, p_1), u, \epsilon) \vdash^*_{\mathcal{T}_{rs}} ((q_f, p_f), \epsilon, v)$$

En cada paso de esta ejecución se satisface lo siguiente:

$$((q_i, p_i), u_i, v_i) \vdash_{\mathcal{T}_{rs}} ((q_{i+1}, p_{i+1}), u_{i+1}, v_{i+1})$$
$$u_i = a \cdot u_{i+1}$$
$$v_{i+1} = v_i \cdot c$$
$$((q_i, p_i), a, c, (q_{i+1}, p_{i+1})) \in \Delta_{rs}$$

Ahora ejecutaremos $\mathcal{T}_r$ con input u, partiendo desde el estado inicial q_1:

$$\rho_r : (q_1, u, \epsilon) \vdash_{\mathcal{T}_r} \ldots \vdash_{\mathcal{T}_r} (q_n, \epsilon, w)$$

Por construcción de la relación de transición de $\mathcal{T}_{rs}$ se tiene que $q_n = q_f$, ya que $\mathcal{T}_{rs}$ lee el **mismo input u que en la ejecución de $\mathcal{T}_r$ y el primer estado de cada par es justamente el mismo al que transiciona $\mathcal{T}_r$ al leer cada letra** (por construcción). Entonces, $q_n \in F_r$ y $\mathcal{T}_r$ entrega w con input u.

Ahora ejecutaremos $\mathcal{T}_s$ con input la palabra w recién obtenida, partiendo desde el estado inicial p_1:

$$\rho_s : (p_1, w, \epsilon) \vdash_{\mathcal{T}_s} ... \vdash_{\mathcal{T}_s} (p_n, \epsilon, x)$$

Por construcción de la relación de transición de $\mathcal{T}_{rs}$, se desprende que $\exists b \in \Omega$ tal que $(q_i, a, b, q_{i+1}) \in \Delta_r$ y $(p_i, b, c, p_{i+1}) \in \Delta_s$. En particular, gracias a que ejecutamos a $\mathcal{T}_r$ y $\mathcal{T}_{rs}$ sobre el mismo input u, cada letra b que satisface estas transiciones **corresponde justamente a las letras que escribe $\mathcal{T}_r$ en su output** (por construcción de Δ_{rs}), es decir, que cada letra que forma a la palabra w cumple con estas propiedades.

Entonces, ahora que sabemos que cada letra $w_i \in w$ satisface $(p_i, w_i, c, p_{i+1}) \in \Delta_s$, y que $v_{i+1} = v_i \cdot c$, se obtiene finalmente que $x = v$, ya que están formadas por exactamente las mismas letras:

$$\rho_s : (p_1, w, \epsilon) \vdash_{\mathcal{T}_s} ... \vdash_{\mathcal{T}_s} (p_n, \epsilon, v)$$

Además, por el mismo argumento que con la ejecución ρ_r, se tiene que $p_n \in F_s$ y entonces $\mathcal{T}_s$ entrega v con input w.

Tenemos que como $\mathcal{T}_r$ entrega w con input u, entonces $(u, w) \in R$ por definición de relación racional. Lo mismo ocurre para $\mathcal{T}_s$ que entrega v con input w, satisfaciendo $(w, v) \in S$. Además $w \in \Omega^*$, ya que es el output del transductor $\mathcal{T}_r$.

Finalmente, llegamos a que $(u, v) \in R \circ S$ por enunciado.

Queda demostrado que $(u, v) \in R \circ S \longleftrightarrow \mathcal{T}_{rs}$ entrega v con input u, por lo que $R \circ S$ **es una relación racional.**