Clase 3. Autómatas Finitos.

Minimización (Repaso).

Autómatas Finitos No Deterministas (AFND).

Autómatas Finitos No Deterministas Lambda (AFND-λ).

Equivalencias.

Construcción del conjunto cociente.

Entrada: Q

Salida: Conjunto cociente de Q por la relación de indistinguibilidad. $\frac{Q}{E}$

1. $\frac{Q}{E_0} = \{F, Q - F\}$
2. Generar $\frac{Q}{E_{i+1}}$ a partir de $\frac{Q}{E_i}$ de la siguiente manera:

Los estados p y q pertenecen a la misma clase en $\frac{Q}{E_{i+1}} \Leftrightarrow$

- p y q pertenecen a la misma clase en $\frac{Q}{E_i}$ y
- $\forall a \in \Sigma, \delta(p, a) y\ \delta(q, a)$ pertenecen a la misma clase en $\frac{Q}{E_i}$

3. Si $\frac{Q}{E_{i+1}} = \frac{Q}{E_i}$, entonces $\frac{Q}{E_{i+1}} = \frac{Q}{E}$. Sino, volver al paso 2.

Minimización de AFD.

Entrada:$A =< Q, \Sigma, \delta, q_0, F >$

Salida:$A' =< Q', \Sigma, \delta', {q_0}', F' >$equivalente a A con mínima cantidad de estados.

1. Eliminar estados inaccesibles desde q_0
2. Construir el conjunto cociente $\frac{Q}{E}$.
3. $A' =< Q', \Sigma, \delta', {q_0}', F' >$ donde:
   - $Q' = \frac{Q}{E}$
   - q_0' es el elemento de $\frac{Q}{E}$ tal que $q_0 \in q_0'$ (la clase donde está q_0)
   - $F' = \{s \in \frac{Q}{E} | S \cap F \neq \emptyset\}$
   - $\delta'(s_i, a) = s_j \Leftrightarrow \exists p \in s_i \wedge \exists q \in s_j | \delta(p, a) = q$

Teorema

El autómata A’ obtenido con el algoritmo anterior es equivalente al autómata A y es mínimo (el número de estados de A’ es menor o igual que el de cualquier otro AFD equivalente a A).

Demostración

1. Demostrar equivalencia: L(A)=L(A’). Para ello, primero ver que $\hat{\delta}'(q_0', \omega) = s_i \Leftrightarrow \exists p \in s_i \middle| \hat{\delta}(q_0, \omega) = p$.
2. Suponer que existe un autómata $A" =< Q", \Sigma, \delta", q_0", F" >$ equivalente a A’ con menos estados.

Sean α y β tales que $\hat{\delta}(q_0', \alpha) = p$ y $\hat{\delta}(q_0', \beta) = q$ en A’,

pero $\hat{\delta}(q_0", \alpha) = \hat{\delta}(q_0", \beta) = r$ en A”.

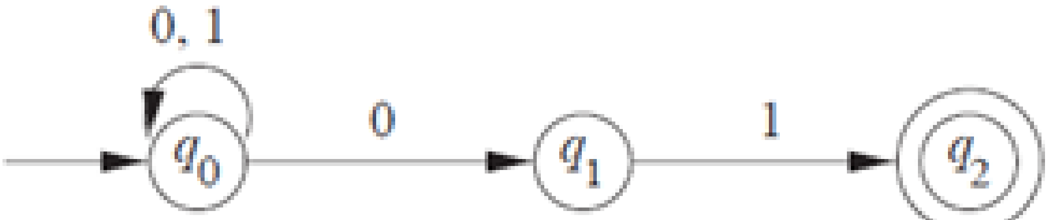
Como A’ fue obtenido por el algoritmo, se sabe que p y q son distinguible tanto A’ aceptaría $\alpha\omega$ pero no $\beta\omega$.

ABSURDO

Sin embargo, en A”, $\hat{\delta}(q_0", \alpha\omega) = \hat{\delta}(q_0", \beta\omega)$,aceptando ambas o rechazand ambas.

Autómatas Finitos No Determinísticos (AFND)

- Cinco componentes: $A = < Q, \Sigma, q_0, F, \delta >$
  - δ = función de transición.
    - Se define de Qx ∑→P(Q)



$$M = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_2\})$$

Autómatas Finitos No Determinísticos

- Función de transición: δ

δ se define de Qx $\Sigma \rightarrow P(Q)$

- Función de transición extendida: $\hat{\delta}$

Describe lo que ocurre cuando se parte de cualquier estado y se sigue una secuencia de entradas.

BASE: $\hat{\delta}(q, \lambda) = \{q\}$

PASO INDUCTIVO: $\mathbf{Si}\ \boldsymbol{\omega} = \boldsymbol{\omega' a},\ \hat{\delta}(q, \boldsymbol{\omega'}) = \{\boldsymbol{p_1}, \boldsymbol{p_2}, \dots, \boldsymbol{p_k}\}, y\ \bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \dots, r_m\}$

$$\Rightarrow \hat{\delta}(q, \omega) = \hat{\delta}(q, \omega' a) = \{r_1, r_2, \dots, r_m\}$$

Lenguaje aceptado por un AFND.

Mediante secuencia de configuraciones:

$$L = \{\omega \in \Sigma * / [q_0, \omega] \mapsto * [p, \lambda], p \in F\}$$

Mediante función de transición extendida:

$$L = \left\{\omega \in \Sigma * \,|\, \hat{\delta}(q_0, \omega) \cap F \neq \emptyset\right\}$$

Equivalencia entre AFD y AFND.

Todo lenguaje reconocido por un AFND puede ser reconocido por un AFD y viceversa.

Construcción de subconjuntos.

Sea $\mathrm{N} = < \mathrm{Q_N}, \Sigma, \delta_N, q_0, F_N >$ un AFND y sea $\mathrm{D} = < \mathrm{Q_D}, \Sigma, \delta_D, \{q_0\}, F_D >$ donde:

- $\mathrm{Q_D} = P(Q_N)$
- $\mathrm{F_D} = \{S \in Q_D | S \cap F_N \neq \emptyset\}$
- $\forall a \in \Sigma, \forall S \subseteq Q_N: \delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a)$

Se puede hacer una "construcción lazy" para dejar en Q_D sólo los estados accesibles.

1. $\{q_0\} \in Q_D$
2. $\forall a \in \Sigma, \forall S \in Q_D, si\ \delta_D(S, a) \notin Q_D,$ $agregar\ \delta_D(S, a)\ a\ Q_D$

Equivalencia entre AFD y AFND.

<u>1ª parte:</u>

Si $\mathrm{D} = < \mathrm{Q_D}, \Sigma, \delta_D, \{q_0\}, F_D >$ se construyó a partir de $\mathrm{N} = < \mathrm{Q_N}, \Sigma, \delta_N, q_0, F_N >$ con el algoritmo anterior, L(D)=L(N).

DEMOSTRACIÓN.

Por inducción en la longitud de ω, se demuestra primero que:

$$\forall \omega \in \Sigma^*: \widehat{\delta_D}(\{q_0\}, \omega) = \hat{\delta}_N(q_0, \omega)$$

Luego se demuestra que:

$$\forall \omega \in \Sigma^*: \omega \in \mathrm{L}(N) \Leftrightarrow \omega \in L(D)$$

Equivalencia entre AFD y AFND.

<u>2ª parte:</u>

Existe $\mathrm{N} = < \mathrm{Q_N}, \Sigma, \delta_N, \{q_0\}, F_N >$ construido a partir de $\mathrm{D} = < \mathrm{Q_D}, \Sigma, \delta_D, q_0, F_D >$
Tal que: L(D)=L(N).

CONSTRUCCIÓN:

- $\mathrm{Q_N} = \{\{q\} | q \in Q_D\}$
- $\mathrm{F_N} = \{\{q\} | q \in F_D\}$
- $\forall \{q\} \in Q_N \forall a \in \Sigma\text{:}\ \delta_N(\{q\}, a) = \{p\} \Leftrightarrow \delta_D(q, a) = p$

Por inducción en la longitud de ω, se demuestra primero que:

$$\forall \omega \in \Sigma^*\text{:}\ \widehat{\delta_N}(\{q_0\}, \omega) = \hat{\delta}_D(q_0, \omega)$$

Luego se demuestra que:

$$\forall \omega \in \Sigma^*\text{:}\ \omega \in \mathrm{L}(N) \Leftrightarrow \omega \in L(D)$$

Autómatas Finitos No Determinísticos con transiciones lambda. (AFND-λ)

- Cinco componentes: $A =< Q, \Sigma, q_0, F, \delta >$
  - δ = función de transición.
    - Se define de Qx ∑∪ $\{\lambda\}$→P(Q)

$$M = (\{p, q, r, s\}, \{a, b\}, \delta, p, \{p, s\})$$

δ	a	b	λ
$*p$	$\{q\}$		
q	$\{q, r, s\}$	$\{p, r\}$	$\{s\}$
r		$\{p, s\}$	$\{r, s\}$
$*s$			$\{r\}$

Autómatas Finitos No Determinísticos Lambda

- Función de transición extendida: $\hat{\delta}$

Se redefine teniendo en cuenta los conjuntos de estados a los que se puede llegar con las transiciones λ

CONSTRUCCIÓN DE CLAUSURAS λ PARA CADA ESTADO.

BASE: $\boldsymbol{claus_{\lambda}}(q) = \{q\}$

PASO INDUCTIVO: $\mathbf{Si}\ \boldsymbol{p} \in \boldsymbol{claus_{\lambda}}(\boldsymbol{q}) \wedge \boldsymbol{r} \in \boldsymbol{\delta}(\boldsymbol{p}, \boldsymbol{\lambda}) \Rightarrow \boldsymbol{r} \in \boldsymbol{claus_{\lambda}}(q)$

DEFINICIÓN DE LA FUNCIÓN DE TRANSICIÓN EXTENDIDA.

BASE: $\hat{\boldsymbol{\delta}}(q, \lambda) = claus_{\lambda}\{q\}$

PASO INDUCTIVO: $\mathbf{Si}\ \boldsymbol{\omega} = \boldsymbol{\omega}'\boldsymbol{a},\ \hat{\boldsymbol{\delta}}(q, \boldsymbol{\omega}') = \{\boldsymbol{p_1}, \boldsymbol{p_2}, \ldots, \boldsymbol{p_k}\}, y\ \bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, .., r_m\}$

$$\Rightarrow \hat{\boldsymbol{\delta}}(q, \omega) = \hat{\boldsymbol{\delta}}(q, \omega' a) = \bigcup_{i=1}^{m} claus_{\lambda}(r_i)$$

Lenguaje aceptado por un AFND-λ .

Mediante secuencia de configuraciones:

$$L = \{\omega \in \Sigma * / [q_0, \omega] \mapsto * \, [p, \lambda], p \in F\}$$

Mediante función de transición extendida:

$$L = \left\{\omega \in \Sigma * \,|\hat{\delta}(q_0, \omega) \cap F \neq \emptyset\right\}$$

Equivalencia entre AFD y AFND-λ.

Todo lenguaje reconocido por un AFND-λ puede ser reconocido por un AFD y viceversa.

Eliminación de transiciones λ.

Sea $\mathrm{E} = < \mathrm{Q_E}, \Sigma, \delta_E, q_0, F_E >$ un AFND-λ.

El AFD equivalente $\mathrm{D} = < \mathrm{Q_D}, \Sigma, \delta_D, claus_\lambda\{q_0\}, F_D >$ se define así:

- $\mathrm{Q_D}$ es el conjunto de subconjuntos de $\mathrm{Q_E}$, $S \subseteq Q_E$ tales que $S = claus_\lambda(S)$
- $\mathrm{F_D} = \{S \in Q_D | S \cap F_E \neq \emptyset\}$
- Se calcula $\delta_D(S, a)$ para todo $a \in \Sigma$ y para todos los subconjuntos S pertenecientes a $\mathrm{Q_D}$ así:
  - $\forall S = \{p_1, p_2, \dots, p_k\}$, $\bigcup_{i=1}^{k} \delta_E(p_i, a) = \{r_1, r_2, \dots, r_m\}$
  - Luego, $\delta_D(S, a) = \bigcup_{j=1}^{m} claus_\lambda(r_j)$

Se puede hacer una “construcción lazy” para dejar en Q_D sólo los estados accesibles.

Equivalencia entre AFD y AFND-λ.

L es aceptado por algún AFND-λ si y sólo si L es aceptado por algún AFD.

DEMOSTRACIÓN.

<u>1ª parte:</u>

Si L es aceptado por algún AFND-λ. $E =< Q_E, \Sigma, \delta_E, \{q_D\}, F_E >$

entonces es aceptado por un AFD: $D =< \mathrm{Q_D}, \Sigma, q_D, \delta_D, F_D >$, construido por eliminación de transiciones lambda.

Se demuestra primero por inducción que $\hat{\delta}_D(claus_\lambda(q_0), \omega) = \hat{\delta}_E(q_0, \omega)$.

Luego se demuestra que $\forall\omega: \omega \in L(D) \Leftrightarrow \omega \in L(E)$

Equivalencia entre AFD y AFND-λ

<u>2ª parte.</u>

Si L es aceptado por algún AFD $D =< \mathrm{Q_D}, \Sigma, q_D, \delta_D, F_D >$,
entonces L es aceptado por un AFND-λ. $E =< Q_E, \Sigma, \delta_E, \{q_D\}, F_E >$ construido por:

- $Q_E = \{\{q\} | q \in Q_D\} \cup \emptyset$
- $\forall \mathrm{q} \in Q_D : \delta_E(q, \lambda) = \emptyset$ y $\forall \mathrm{q} \in Q_D, \forall a \in \Sigma : \delta_D(q, a) = p \Rightarrow \delta_E(\{q\}, a) = \{p\}$
- $F_E = \{\{q\} | q \in F_D\}$