Autómatas Mínimos

Encontrar el autómata mínimo.

Universidad de Cantabria

Introducción

Dado un lenguaje regular sabemos encontrar un autómata finito. Pero, ¿hay autómatas más sencillos que aceptan el mismo lenguaje?

¿Que significa que un autómata sea más sencillo?

Los autómatas minimos son aquellos que tienen el mínimo número de estados.

Introducción

Dado un lenguaje regular sabemos encontrar un autómata finito. Pero, ¿hay autómatas más sencillos que aceptan el mismo lenguaje?

¿Que significa que un autómata sea más sencillo?

Los autómatas minimos son aquellos que tienen el mínimo número de estados.

Introducción

Dado un lenguaje regular sabemos encontrar un autómata finito. Pero, ¿hay autómatas más sencillos que aceptan el mismo lenguaje?

¿Que significa que un autómata sea más sencillo?

Los autómatas minimos son aquellos que tienen el mínimo número de estados.

Introducción

Nos restringiremos a autómatas deterministas. Queremos tener un procedimiento que, dado un autómata determinista, se retorne un autómata mínimo.
Para resolver esta situación se utiliza un proceso de minimización de los autómatas que pasaremos a describir a continuación.

Minimización de Autómatas Deterministas

Definición

Sea $A := (Q, \Sigma, q_0, F, \delta)$ *un autómata. Dos estados* $p, q \in Q$ *se dicen equivalentes si se verifica que* $\forall z \in \Sigma^*$*:*

$$(p, z) \vdash (p', \lambda) \land (q, z) \vdash (q', \lambda) \Longrightarrow (p' \in F) \Leftrightarrow (q' \in F).$$

Minimización de Autómatas Deterministas

En otras palabras, dos estados son equivalentes si para cualquier palabra el efecto de la computación que generan es el mismo (en términos de alcanzar o no un estado final aceptador).

Minimización de Autómatas Deterministas

Denotaremos por $p \sim_A q$ en el caso de que p y q sean equivalentes. Para cada estado $q \in Q$, denotaremos por $[q]_A$ la clase de equivalencia definida por q y denotaremos por $Q/\sim_A$ al conjunto cociente. Definiremos autómata minimal al autómata que tiene el menor número de estados y que acepta un lenguaje.

Minimización de Autómatas Deterministas

Esta relación es equivalente a la relación de equivalencia dada por los prefijos. En el caso anterior trabajamos con las palabras del lenguaje, lo que quiere decir que es un invariante del lenguaje.

Resultado Principal

Teorema

Sea L un lenguaje aceptado por un autómata determinista A. Entonces, existe un autómata mínimo que lo acepta. Dicho autómata $(\tilde{Q}, \Sigma, \widetilde{Q_0}, \tilde{F}, \tilde{\delta})$ *viene dado por las propiedades siguientes:*

- $\tilde{Q} := Q/\sim_A$,
- $\tilde{F} := \{[q]_A \ : \ q \in F\}$.
- $\widetilde{q_0} := [q_0]_A$.
- $\tilde{\delta}([q]_A, z) := [\delta(q, a)]$.

Nota

Este autómata se llama cociente, por la relación de equivalente, ya que agrupa estados equivalentes y solo se preocupa de sus transiciones. Para definir las transiciones van de la clase de equivalencia a otra clase de equivalencia y viene dada por las transiciones de cualquiera de sus miembros.

Ejemplo

Sea el siguiente automata
$A = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, q_0, q_0, q_1, \delta)$, donde la función de transición es:

δ	**a**	**b**
q_0	q_1	q_3
q_1	q_0	q_2
q_2	q_3	q_2
q_3	q_3	q_3

Si supieramos que $\{q_0, q_1\}$ y $\{q_2, q_3\}$ son equivalentes entonces podemos hallar un nuevo autómata que solo tiene dos estados.

Ejemplo

¿Pueden ser q_0, q_2 equivalentes?, ¿sabiendo que $\{q_0, q_1\}$ son equivalentes se podia deducir que $\{q_2, q_3\}$ son equivalentes?

El Algoritmo

El problema es que el cálculo de las clases de equivalencia no puede hacerse de manera simple (porque habríamos de verificar todas las palabras $z \in \Sigma^*$).

El Algoritmo

Sea $A := (Q, \Sigma, q_0, F, \delta)$ un autómata. Vamos a definir diferentes relaciones, cada una más tosca que la siguiente que nos permita hallar las clases de equivalencia.

El Algoritmo

La relación E_0: Dados $p, q \in Q$, diremos que pE_0q (p y q están relacionados al nivel 0) si se verifica:

$$p \in F \Leftrightarrow q \in F.$$

Es claramente una relación de equivalencia. El conjunto cociente está formado por dos clases:

$$Q/E_0 := \{F, Q \setminus F\}.$$

Definamos $e_0 := \sharp(Q/E_0) = 2$.

El Algoritmo

La relación E_1: Dados $p, q \in Q$, diremos que pE_1q (p y q están relacionados al nivel 1) si se verifica:

$$pE_1q \Leftrightarrow \begin{cases} pE_0q, \\ \wedge \\ \delta(p,z)E_0\delta(q,z), \quad \forall z \in \Sigma \cup \{\lambda\} \end{cases}$$

Es, de nuevo, una relación de equivalencia. El conjunto cociente ya no es tan obvio, y definimos:

$$e_1 := \sharp(Q/E_1)\,.$$

El Algoritmo

La relación E_n: Para $n \geq 2$, definimos la relación del modo siguiente: Dados $p, q \in Q$, diremos que pE_nq (p y q están relacionados al nivel n) si se verifica:

$$pE_nq \Leftrightarrow \begin{cases} pE_{n-1}q, \\ \wedge \\ \delta(p,z)E_{n-1}\delta(q,z), \quad \forall z \in \Sigma \cup \{\lambda\} \end{cases}$$

Es, de nuevo, una relación de equivalencia. El conjunto cociente ya no es tan obvio, y definimos:

$$e_n := \sharp\,(Q/E_n)\,.$$

El Algoritmo

Esto separa cada vez más estados y muestra cuales pueden ser equivalentes y cuales no. En algún momento estas relaciones nos daran los mismos conjuntos.
Queremos ver si cuando da los mismos conjuntos tenemos el autómata deseado.

El Algoritmo

Lo que esta claro es que la relación de equivalencia E_n es más débil que pedir que dos estados sean equivalentes, pero si n es lo suficientemente grande, se tiene que son la misma relación de equivalencia.

Teorema

Teorema

Sea $A := (Q, \Sigma, q_0, F, \delta)$ *un autómata sin* $\lambda-$*transiciones y sean* p, q *dos estados. Entonces, tomando* $n = \sharp(Q) - 2$*, se tendrá que*

$$p \sim_A q \Leftrightarrow pE_n q.$$

Resumen del algoritmo

- Hallar el conjunto cociente (Q/E_0) y su cardinal e_0.
- (Siguiendo los E_i's) Mientras el conjunto cociente "nuevo" sea alterado con respecto al anterior, hallar el conjunto cociente siguiente.
- Parar cuando el cardinal del "nuevo" conjunto cociente coincida con el último calculado.