Indice

Dimostrazione. Cominciamo dal disegnare gli automi per gli stati base: E

ora occupiamoci delle operazioni:

- unione: dati due automi $\mathcal{A}'$ con n' stati e $\mathcal{A}''$ con n'' stati che riconoscano rispettivamente L' e L'' posso costruire l'automa $\mathcal{A}$ che riconosce $L' \cup L''$. L'unione $\mathcal{A}$ allora è un automa non determinisico con $1+n'+n''$. Quindi l'automa determinisitco corrispondente ha al massimo $2^{1+n'+n''}$ stati.

  Alternativamente si può pensare che data una stringa s, se prendiamo i due automi $\mathcal{A}'$ e $\mathcal{A}''$, possiamo far andare entrambi gli automi su due computer diversi per la stessa stringa, e se almeno uno risponde positivamente, allora la stringa fa parte dell'unione. Definiamo

  $$\mathcal{A}' = \langle Q', \Sigma, \delta', q'_0, F' \rangle$$
  $$\mathcal{A}'' = \langle q'', \Sigma, \delta'', q''_0, F'' \rangle$$

  Definisco

  $$\mathcal{A} = \langle Q = Q' \times Q'', \Sigma, \delta, (q'_0, q''_0), F = \{(q', q'') \mid q' \in F' \vee q'' \in F''\} \rangle$$

  con δ definita come

  $$\forall (q', q'') \in Q \forall a \in \Sigma \delta((q', q''), a) = (\delta'(q^prime, a), \delta''(q'', a))$$

  Con questa costruzione se entrambi gli automi $\mathcal{A}'$ con n' stati e $\mathcal{A}''$ con n'' stati sono entrambi deterministici, allora $\mathcal{A}$ ha al massimo $n' \cdot n''$ stati ed è deterministico.

  Questo tipo di automa è detto *automa prodotto*.

  Con lo stesso metodo possiamo anche realizzare l'intersezione, definendo

  $$F = F' \times F''$$

  Mentre l'intersezione con le ϵ-mosse non permette anche di realizzare l'intersezione.

  Questa costruzione dovrebbe funzionare anche per automi non deterministici.

- complemento: dato un automa deterministico

  $$\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$$

  accetti L. Costruriamo il complemento come

  $$\mathcal{A}^C = \langle Q, \Sigma, \delta, q_0, F^C \rangle$$

  riconosce il complemento. Quindi se il numero di stati era n, questo non cambia.

  Bisogna stare attenti agli stati trappola omessi, quindi l'automa $\mathcal{A}$ deve essere completo.

  Nel caso di NFA non si può applicare la stessa costruzione. Il modo più semplice per fare il complemento di un NFA è prima costruire il DFA corrispondente e poi complementare.

  $$\mathcal{A} \to \mathcal{A}_d \to \mathcal{A}_d^C$$

  Quindi in questo caso se $\mathcal{A}$ ha n stati, il complemento ne avrà nel caso peggiore 2^n. Questo deriva dal fatto che negli NFA accettazione e non accettazione non è simmetrica.

- concatenazione o prodotto: dati

  $$\mathcal{A}' = \langle Q', \Sigma, \delta', q_0', F' \rangle$$
  $$\mathcal{A}'' = \langle Q'', \Sigma, \delta'', q_0'', F'' \rangle$$

  che riconoscono rispettivamente L' e L'' ed hanno rispettivamente n' ed n'', voglio riconoscrere $L' \cdot L''$ Con questa costruzione ottengo un automa non deterministico (ϵ-mosse) da $n' + n''$ stati.

  Se gli automi iniziali erano DFA, ottengo un NFA da $n' + n''$ stati, quindi al peggio un DFA da $2^{n'+n''}$.

  Si può ancora pensare con il modello dei più computer: una volta che il primo mi dice sì, faccio partice un secondo sul resto, ma il primo continua ad andare, se il primo dice ancora sì faccio partire un terzo e così via. Definiamo

  $$\mathcal{A} = \langle Q = Q' \times 2^{Q''}, \Sigma, \delta, q_0 = \begin{cases} (q_0', \varnothing) & \text{se } q_0' \notin F' \\ (q_0', \{q_0''\}) & \text{altrimenti} \end{cases}, F = \{(q, \alpha) \in Q' \times 2^{Q''} \mid \alpha \cap F'' \neq \varnothing\} \rangle$$

con δ definito come

$$\forall q \in Q', \alpha \subseteq Q'', a \in \Sigma \mid \delta((q, \alpha), a) = \begin{cases} (\delta'(q, a), \{\delta''(p, a) \mid p \in \alpha\}) & \text{se } \delta'(p, a) \notin F' \\ (\delta'(q, a), \{\delta''(p, a) \mid p \in \alpha\} \cup \{q_0''\}) & \text{altrimenti} \end{cases}$$

Se gli stati erano n' e n'' gli stati del prodotto sono $n' \cdot 2''$.

- chiusura di Kleene: data una stringa in input w devo trovare un numero arbitrario di volte in cui scomporla. In questo caso partendo da n stati generiamo un NFA da $n + 1$ stati. Utilizzando una costruzione simile a quella di prima possiamo arrivare al meno ad un DFA da $2^n + 1$ stati invece che 2^{n+1}. Se l'automa è non deterministico possiamo definire

$$\mathcal{A}' = \langle Q' = 2^Q \cup \{q_0'\}, \Sigma, \delta', F' = \{\alpha \subseteq Q \mid \alpha \cap F \neq \varnothing\}\rangle$$

con q_0' nuovo, e con δ' definito come

$$\forall a \in \Sigma \mid \delta'(q_0', a) = \begin{cases} \{\delta(q_0, a)\} & \text{se } \delta(q_0, a) \notin F \\ \{\delta(q_0, a), q_0\} & \text{altrimenti} \end{cases}$$

e

$$\forall \alpha \subseteq Q, a \in \Sigma \mid \delta(\alpha, a) = \begin{cases} \beta & \text{se } \beta \cap F = \varnothing \\ \beta \cup \{q_0\} & \text{altrimenti} \end{cases}$$

con

$$\beta = \{\delta(p, a) \mid p \in \alpha, a \subseteq Q\}$$

E si può vedere che questo da n stati passa a $2^n + 1$.

■

Per esempio definiamo l'unione dei due automi
Scegliendo 00 come unico stato finale si realizza invece l'intersezione dei due automi.

Costruiiamo il complemento di $\mathcal{A}$
Supponiamo ora di applicare la stessa costruzione ad un NFA.

Mostriamo che il complemento di un NFA di m stati ha al peggio 2^m stati. Prendiamo

$$K_n = \{w \in \{a,b\}^* \mid \text{i simboli a distanza } n \text{ in } w \text{ son ouguali}\}$$

utilizzando la tecnica del fooling set abbiamo mostrato che ogni automa non deterministico ha al peggio 2^n stati.
Ora definiamo

$$K_n^C = \{w \in \{a,b\}^* \mid \text{esistono due simboli a distanza } n \text{ in } w \text{ diversi tra loro}\}$$

questo ha $2n+2$ stati.
Un NFA come quello che riconosce il linguaggio il cui terzultimo simbolo è una a, il suo complemento ha lo stesso numero di stati (con stati iniziali multipli).

Il nondeterminismo è necessaria nel caso del prodotto, infatti supponiamo di avere i linguaggi $L' = abaa*$ e $L'' = aab*$, e la parola $w = abaaaaaaaab$, la suddifivisione corretta va scelta

$$abaaaaaa \mid aab$$

Sia $L' = a + b*$ e sia $L'' = a(a+b)^{n-1}$. Il primo è riconosciuto in uno stato, il secondo in $n+1$, ma il prodotto è l'automa che riconosce che l'ennesimo simbolo dalla fine è una a, quindi sono 2^n stati.

Non possiamo mettere come stato finale il primo stato dell'automa con il perché c'è il rischio di riconoscere qualcosa in più.

Diciamo che $X \subseteq \Sigma^*$ è un codice sse

$$\forall w \in X^* \exists! \text{ decomposizione di } w \text{ in stringhe di } X$$

I codici hanno quindi la caratteristica di essere scomponibili velocemente.

Ad esempio $X = \{ab, aab\}$ è un codice, infatti basta scomporre dopo ogni b:

$$ab \mid aab \mid aab \mid ab$$

Ancora $X = aba + ab$ è un codice, infatti

$$aba \mid ab \mid ab \mid aba$$

Infine $X = \{abaa, ab, a, aa\}$ non è un codice, infatti $abaa$ è sia una parola di X che può essere scomposto in

$$ab \mid aa$$

In particolare parliamo di codice prefisso se questo non contiene stringhe che sono prefisse di altre. Se il nostro linguaggio è un codice prefisso, allora la costruzione dell'automa della chiusura di Kleene è più semplice.