Limbaje Formale și Automate
Tutoriat 5

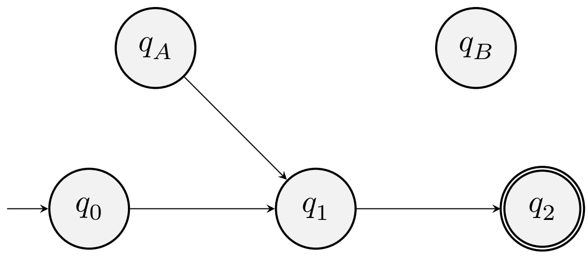
Gabriel Majeri

Minimizarea și echivalența automatelor finite

Ca să minimizăm un DFA, eliminăm stările *inaccesibile* și le grupăm pe cele *redundante*.

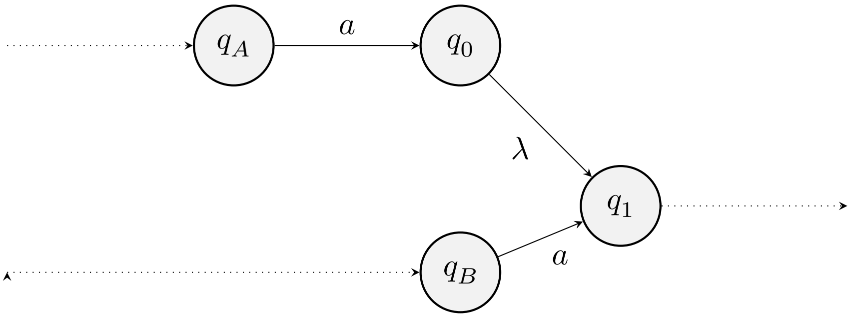
O stare este **inaccesibilă** dacă nu există niciun drum de la starea inițială la ea. Aceste stări pot fi eliminate fără problemă dintr-un automat, nu modifică limbajul acceptat de acesta.

Exemplu de stări inaccesibile (q_A și q_B)



Două stări sunt **echivalente** (nu pot fi distinse) dacă din ele se acceptă exact aceleași limbaje. Aceste stări pot fi unite într-una singură.

Exemplu de stări echivalente (q_A și q_B)



Asta ne permite să determinăm iterativ care sunt stările echivalente.

Forma Normală Chomsky (FNC)

În comparație cu automatele finite, la gramatici **nu** există o unică reprezentare minimă. Putem obține mai multe forme Chomsky pornind de la aceeași gramatică, și nu putem să le comparăm în vreun fel.

O gramatică se află în **formă normală Chomsky** dacă toate producțiile ei sunt de forma

- $A \to a$, unde a este un terminal
- $A \to XY$, unde X și Y sunt neterminale

Algoritm de obținere a FNC

Aceasta este ordinea recomandată pentru că limitează numărul de producții care apar în final (altfel, putem avea un număr *exponențial* de producții).

1. Eliminăm simbolurile cu o buclă care nu se termină. De exemplu, producțiile

$$A \to B$$
$$B \to C$$
$$C \to A$$

   pot fi eliminate.

2. Eliminăm simbolurile care nu pot fi obținute plecând din simbolul de start. De exemplu, din gramatica

$$S \to A$$
$$A \to a$$
$$X \to b$$

   putem elimina $X \to b$.

3. Înlocuim terminalii din producți cu neterminali. Dacă avem, de exemplu:

$$A \to AbCDe$$

   transformăm în

$$A \to ABCDE$$
$$B \to b$$
$$E \to e$$

4. Eliminăm producțiile de lungime > 2. Orice producție lungă poate fi spartă în mai multe producții mai mici. De exemplu:

   $$A \to BCDE$$

   devine:

   $$\begin{aligned} A &\to BX \\ X &\to CY \\ Y &\to DE \end{aligned}$$

   unde toate producțiile au lungime 2.

5. Eliminăm λ-producțiile. Dacă un neterminal are o producție care merge în λ, eliminăm această producție, căutăm toate celelalte producții care îl conțin,

   $$\begin{aligned} A &\to XaYb \\ X &\to x \mid \lambda \\ Y &\to y \mid \lambda \end{aligned}$$

   devine

   $$\begin{aligned} &A \to XaYb \mid aYb \mid Xab \mid ab \\ &X \to x \\ &Y \to y \end{aligned}$$

   Dacă într-o producție avem n neterminali care pot genera λ, o vom înlocui cu 2^n producții.

   Observație. Prin acest proces este posibil să-l pierdem pe λ din limbajul generat de gramatică. Obținem ce se numește o gramatică λ-echivalentă cu cea inițială (aceleași cuvinte, dar lipsește λ).

6. Eliminăm producțiile unitate. Producțiile unitate sunt de forma

   $$A \to B$$

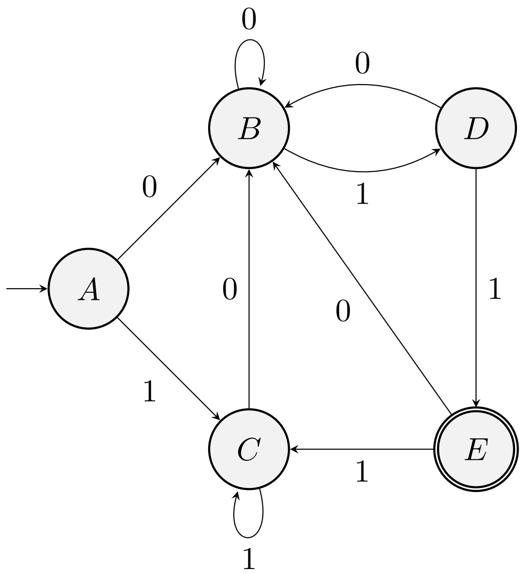
   Eliminăm această producție, și înlocuim peste tot unde era B cu A (sau invers).

De ce este utilă această formă?

Un *parser* este un program care poate determina un cuvânt **aparține sau nu unei gramatici**, și dacă aparține, **cum se poate deriva** acesta. Un exemplu de algoritm care folosește FNC pentru a determina arborele de derivare este algoritmul Cocke-Younger-Kasami (CYK).

Exemple

Exercițiul 1. Să se minimizeze următorul DFA:



Rezolvare. O rezolvare pas cu pas se poate găsi în acest video. □

Exercițiul 2. Să se reducă la Formă Normală Chomsky următoarea gramatică:

$$\begin{aligned}
S &\to T \mid U \mid X \\
T &\to VaT \mid VaV \mid TaV \\
U &\to VbU \mid VbV \mid UbV \\
V &\to aVbV \mid bVaV \mid \lambda \\
X &\to Y \\
Y &\to X
\end{aligned}$$

Aceasta este inspirată de cea care generează cuvintele cu un număr diferit de a-uri și b-uri.

Rezolvare. Aplicăm pașii de mai sus:

1. Eliminăm ultimele două producții, care ciclează:

$$
\begin{aligned}
S &\to T \mid U \mid X \\
T &\to VaT \mid VaV \mid TaV \\
U &\to VbU \mid VbV \mid UbV \\
V &\to aVbV \mid bVaV \mid \lambda
\end{aligned}
$$

2. Eliminăm simbolul nefolosit X:

$$
\begin{aligned}
S &\to T \mid U \\
T &\to VaT \mid VaV \mid TaV \\
U &\to VbU \mid VbV \mid UbV \\
V &\to aVbV \mid bVaV \mid \lambda
\end{aligned}
$$

3. Extragem neterminalii a și b:

$$
\begin{aligned}
S &\to T \mid U \\
T &\to VAT \mid VAV \mid TAV \\
U &\to VBU \mid VBV \mid UBV \\
V &\to AVBV \mid BVAV \mid \lambda \\
A &\to a \\
B &\to b
\end{aligned}
$$

4. Spargem producțiile lungi în mai multe producții scurte, introducând noi neterminali:

$$
\begin{aligned}
S &\to T \mid U \\
T &\to VT_1 \mid VT_2 \mid TT_3 \\
T_1 &\to AT \\
T_2 &\to AV \\
T_3 &\to AV \\
U &\to VU_1 \mid VU_2 \mid UU_3 \\
U_1 &\to BU \\
U_2 &\to BV \\
U_3 &\to BV \\
V &\to AV_1 \mid BV_3 \mid \lambda \\
V_1 &\to VV_2 \\
V_2 &\to BV \\
V_3 &\to VV_4 \\
V_4 &\to AV \\
A &\to a \\
B &\to b
\end{aligned}
$$

5. Eliminăm producțiile cu λ:

$$
\begin{aligned}
S &\to T \mid U \\
T &\to VT_1 \mid T_1 \mid VT_2 \mid T_2 \mid TT_3 \\
T_1 &\to AT \\
T_2 &\to AV \mid A \\
T_3 &\to AV \mid A \\
U &\to VU_1 \mid U_1 \mid VU_2 \mid U_2 \mid UU_3 \\
U_1 &\to BU \\
U_2 &\to BV \mid B \\
U_3 &\to BV \mid B \\
V &\to AV_1 \mid BV_3 \\
V_1 &\to VV_2 \mid V_2 \\
V_2 &\to BV \mid B \\
V_3 &\to VV_4 \mid V_4 \\
V_4 &\to AV \mid A \\
A &\to a \\
B &\to b
\end{aligned}
$$

6. Eliminăm producțiile unitate:

$$
\begin{aligned}
S &\to VT_1 \mid AT \mid VT_2 \mid AV \mid a \mid TT_3 \\
S &\to VU_1 \mid BU \mid VU_2 \mid BV \mid b \mid UU_3 \\
T &\to VT_1 \mid AT \mid VT_2 \mid AV \mid a \mid TT_3 \\
T_1 &\to AT \\
T_2 &\to AV \mid a \\
T_3 &\to AV \mid a \\
U &\to VU_1 \mid BU \mid VU_2 \mid BV \mid b \mid UU_3 \\
U_1 &\to BU \\
U_2 &\to BV \mid b \\
U_3 &\to BV \mid b \\
V &\to AV_1 \mid BV_3 \\
V_1 &\to VV_2 \mid BV \mid b \\
V_2 &\to BV \mid b \\
V_3 &\to VV_4 \mid AV \mid a \\
V_4 &\to AV \mid a \\
A &\to a \\
B &\to b
\end{aligned}
$$

Toate producțiile au fie doi neterminali în dreapta fie un singur terminal.

□