

Задача 1. (10 т.) Да се провери кои от следващите езици над азбуката $\Sigma = \{0, 1\}$ са едни и същи и кои са различни:

1. L_1 се представя чрез регулярния израз $1 \cdot (0 \cup 1)^* \cdot 0$
2. L_2 се разпознава от крайния недетерминиран автомат

$$\mathcal{A} = \langle \{q_0, q_1, q_2, q_3\}, \{0, 1\}, q_0, \delta, \{q_3\} \rangle$$

с функция на преходите δ , представена чрез таблицата:

q	0	1
q_0	$\{q_1\}$	$\{q_2\}$
q_1	$\{q_1, q_3\}$	$\{q_1\}$
q_2	$\{q_2\}$	$\{q_2, q_3\}$
q_3	\emptyset	\emptyset

3. L_3 се разпознава от крайния детерминиран автомат

$$\mathcal{C} = \langle \{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, q_0, \delta, \{q_2, q_4\} \rangle$$

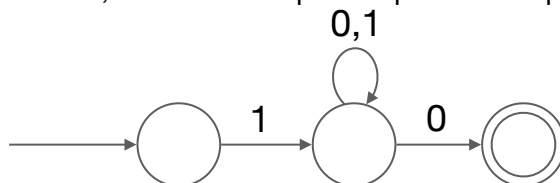
с функция на преходите δ , представена чрез таблицата:

q	0	1
q_0	q_1	q_3
q_1	q_2	q_1
q_2	q_2	q_1
q_3	q_3	q_4
q_4	q_3	q_4

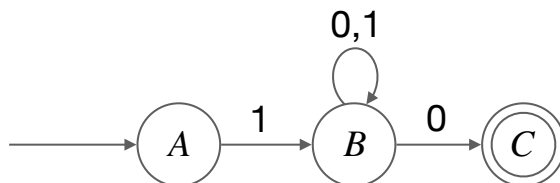
За да покажете, че два езика са различни посочете дума, която е от единия език, но не е от другия, а за да покажете, че два езика съвпадат сравнете крайните детерминирани автомати, които ги разпознават.

Решение.

Директно построяваме автомата, който отговаря на простичкия регулярен израз:

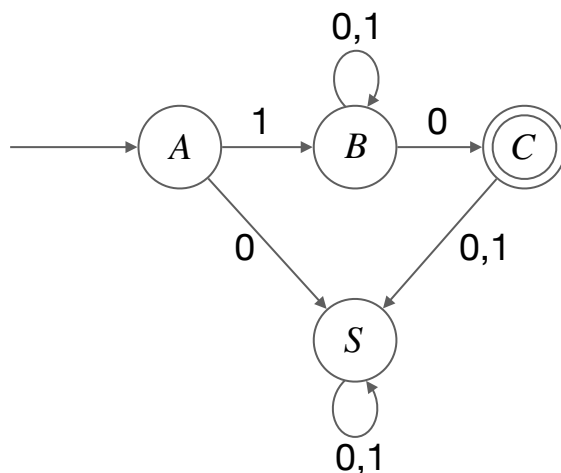


Именуваме състоянията на автомата за улеснение.



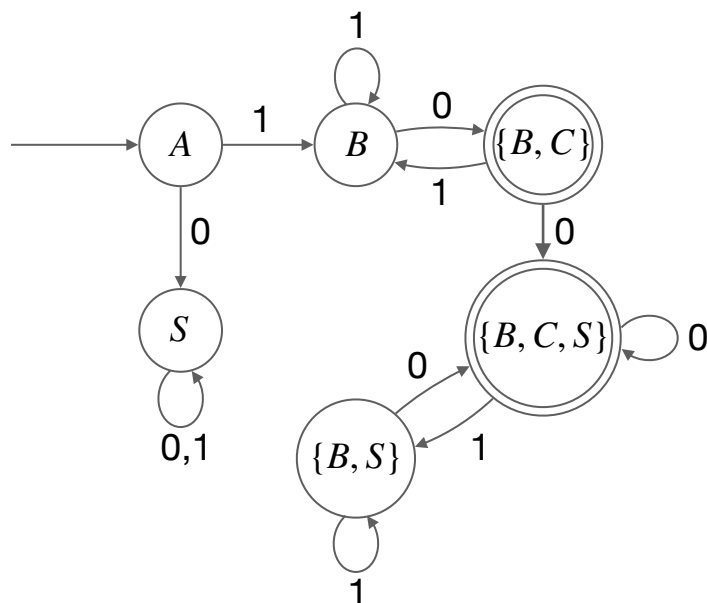
Очевидно от състояние A няма преход с 0, а от състояние C няма преход с 0 и 1. Следователно автомата не е тотален и го тотализираме.

Създаваме ново състояние S (sink), което ще играе ролята на `/dev/null` и ще обира липсващите преходи със съответните букви от входната азбука. Попадането в това ново състояние на автомата ще кара автомата да зацикля в същото състояние с всяка една последваща буква, която прочита и по този начин няма да му позволим да стига до финално състояние.



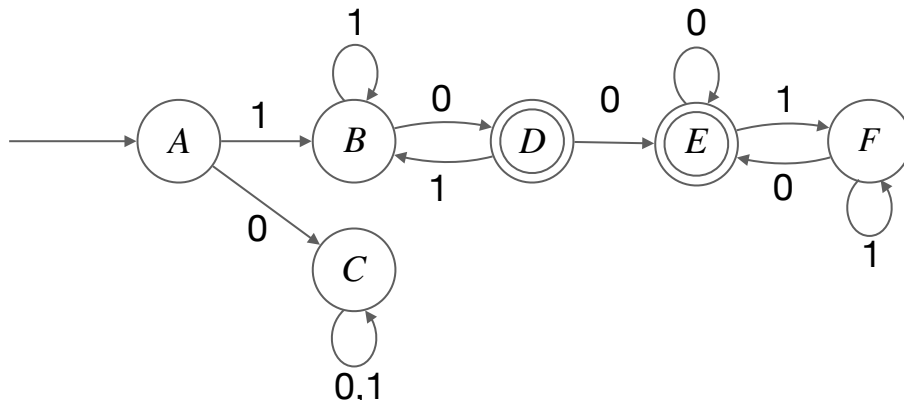
При така получения тотален краен автомат, от състояние B може да траверсираме недетерминирано отново до състояние B или до състояние C с 0 . Следователно автомата е недетерминиран и го детерминираме.

състояние	преход с 0	преход с 1
A	S НОВО	B НОВО
S	S	S
B	$\{B, C\}$ НОВО	B
$\{B, C\}$	$\{B, C, S\}$ НОВО	B
$\{B, C, S\}$	$\{B, C, S\}$	$\{B, S\}$ НОВО
$\{B, S\}$	$\{B, C, S\}$	$\{B, S\}$



Крайни състояния са тези, които имат поне едно крайно състояние в множеството, с което се репрезентират. Тъй като само C е крайно състояние, то крайните състояния в детерминирания краен автомат са $\{B,C\}$ и $\{B,C,S\}$.

След операцията детерминизация не е гарантирано, че ще получим минимален краен детерминиран автомат. За това ще извършим и операцията минимизация. Преди това ще преименуваме състоянията за улеснение.



Минимизация. Разделяме състоянията на два класа на еквивалентност – междинни и терминиращи (нефинални и финални).

$$P_1 = \{A, B, C, F\},$$

$$P_2 = \{D, E\}.$$

състояние	преход с 0	преход с 1
A	P_1	P_1
B	P_2	P_1
C	P_1	P_1
D	P_2	P_1
E	P_2	P_1
F	P_2	P_1

Състоянията от P_2 имат еднакво поведение и тяхното множество няма да се разбие, но състоянията от P_1 имат различно поведение: A и C с 0 отиват в P_1 , а B и F с 0 отиват в P_2 . Следователно разбиваме множеството P_1 на две нови множества.

$$P_2 = \{D, E\},$$

$$P_3 = \{A, C\},$$

$$P_4 = \{B, F\}$$

състояние	преход с 0	преход с 1
A	P_4	P_3
B	P_2	P_4
C	P_3	P_3
D	P_2	P_4
E	P_2	P_4
F	P_2	P_4

Състоянията от P_2 и състоянията от P_4 имат еднакво поведение и множествата им няма да се разбият, но състоянията от P_3 имат различно поведение: A с 0 отива в P_4 , B с 0 отива в P_3 . Следователно разбиваме множеството P_3 на две нови множества.

$$P_2 = \{D, E\},$$

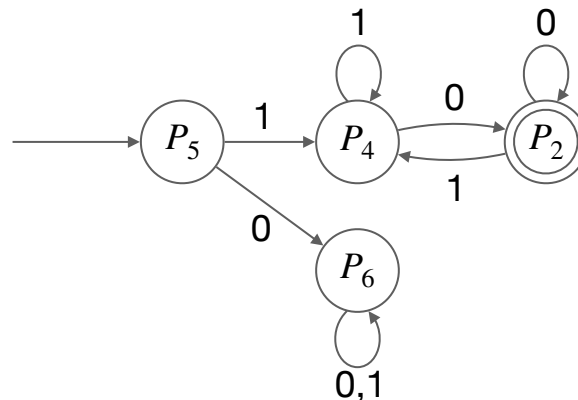
$$P_4 = \{B, F\},$$

$$P_5 = \{A\},$$

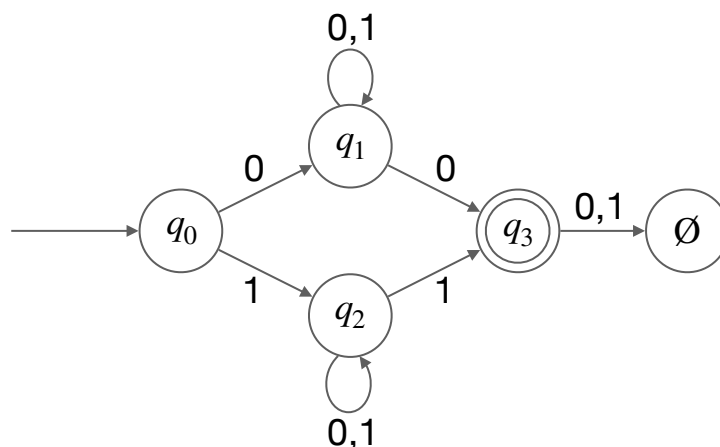
$$P_6 = \{C\}$$

състояние	преход с 0	преход с 1
A	P_6	P_4
B	P_2	P_4
C	P_6	P_6
D	P_2	P_4
E	P_2	P_4
F	P_2	P_4

Всички състояния от всички множества имат еднакво поведение. Алгоритъмът спира. Финалните състояния са тези, в които има поне едно финално състояние от оригиналния автомат, който минимизирахме – това е само състоянието P_2 . Окончателно за езика L_1 зададен с регулярния израз $1 \cdot (0 \cup 1)^* \cdot 0$ получаваме крайния детерминиран и минимизиран автомат:

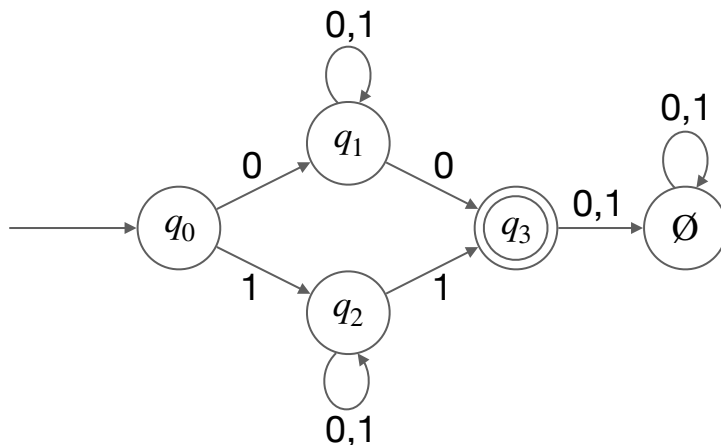


За езика L_2 имаме следния краен недетерминиран автомат:



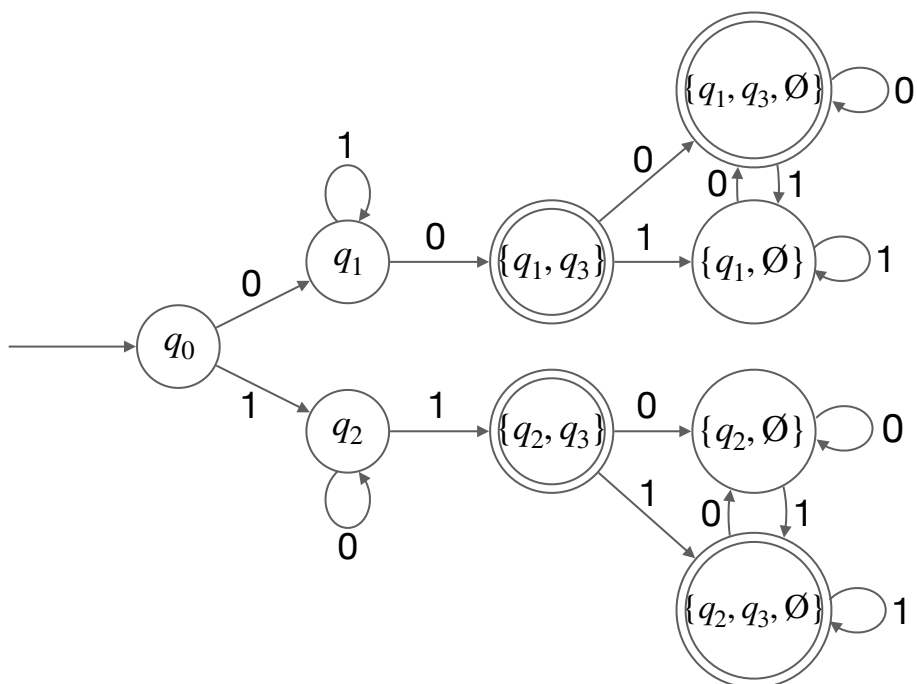
Очевидно е, че това е автомата, който разпознава всички думи, които започват и завършват с еднаква буква от входната азбука. Нека го тотализираме и след това детерминираме и минимизираме.

След тотализация автомата ще има следния вид:

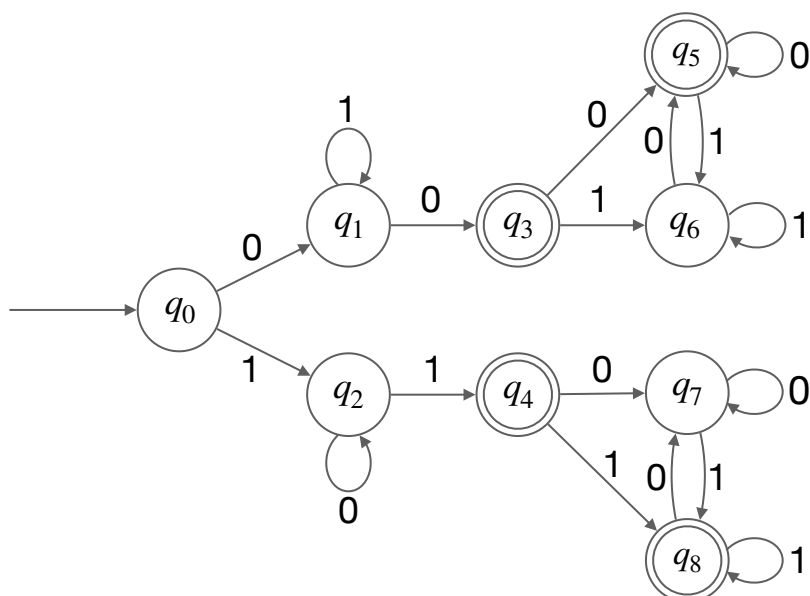


Детерминизация:

състояние	преход с 0	преход с 1
q_0	q_1 НОВО	q_2 НОВО
q_1	$\{q_1, q_3\}$ НОВО	q_1
q_2	q_2	$\{q_2, q_3\}$ НОВО
$\{q_1, q_3\}$	$\{q_1, q_3, \emptyset\}$ НОВО	$\{q_1, \emptyset\}$ НОВО
$\{q_2, q_3\}$	$\{q_2, \emptyset\}$ НОВО	$\{q_2, q_3, \emptyset\}$ НОВО
$\{q_1, q_3, \emptyset\}$	$\{q_1, q_3, \emptyset\}$	$\{q_1, \emptyset\}$
$\{q_1, \emptyset\}$	$\{q_1, q_3, \emptyset\}$	$\{q_1, \emptyset\}$
$\{q_2, q_3, \emptyset\}$	$\{q_2, \emptyset\}$	$\{q_2, q_3, \emptyset\}$
$\{q_2, \emptyset\}$	$\{q_2, \emptyset\}$	$\{q_2, q_3, \emptyset\}$



Преименуване:



Минимизация:

$$P_1 = \{q_0, q_1, q_2, q_6, q_7\},$$

$$P_2 = \{q_3, q_4, q_5, q_8\}$$

състояние	преход с 0	преход с 1
q_0	P_1	P_1
q_1	P_2	P_1
q_2	P_1	P_2
q_3	P_2	P_1
q_4	P_1	P_2
q_5	P_2	P_1
q_6	P_2	P_1
q_7	P_1	P_2
q_8	P_1	P_2

Разбиваме P_1 на

$$P_3 = \{q_0\},$$

$$P_4 = \{q_1, q_6\},$$

$$P_5 = \{q_2, q_7\}$$

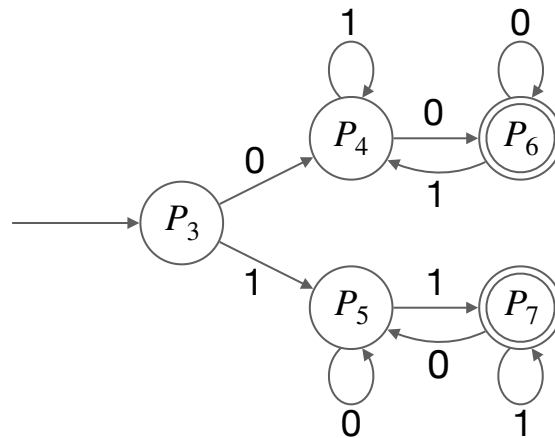
и P_2 на

$$P_6 = \{q_3, q_5\},$$

$$P_7 = \{q_4, q_8\}.$$

състояние	преход с 0	преход с 1
q_0	P_4	P_5
q_1	P_6	P_4
q_2	P_5	P_7
q_3	P_6	P_4
q_4	P_5	P_7
q_5	P_6	P_4
q_6	P_6	P_4
q_7	P_5	P_7
q_8	P_5	P_7

Алгоритъмът за минимизация терминара, тъй като състоянията от всяко множество имат еднакво поведение и не може да разбиваме повече. Финални състояния са тези, които в множеството си имат поне едно от състоянията, които са били финални в автомата, който минимизираме, а именно q_3 , q_4 , q_5 или q_8 . Това са съответно новите състояния P_6 и P_7 .



За езика L_3 имаме същия краен детерминиран автомат (освен това знаем, че той е и минимален), като този за езика L_2 (с точност до изоморфизъм/преименуване на състоянията). Следователно езиците L_2 и L_3 са едни и същи (и разпознават думите от азбуката $\Sigma = \{0, 1\}$, които започват и завършват с еднакви букви).

За да докажем, че езика L_1 е различен е достатъчно да вземем дума от него, която започва с 1 и завършва на 0 (тъй като от L_1 няма думи започващи с 0). Това е например думата 10.