

Avtomati

Digitalna vezja

Miha Moškon

miha.moskon@fri.uni-lj.si

<https://fri.uni-lj.si/en/about-faculty/employees/miha-moskon>

Končni avtomat (finite-state machine)

Končno število možnih stanj

Regularni izrazi (končno stanje)

Množica stanj in prehodov

Brez pomnilnika (za razliko od Turingovega stroja) – razen pomnjenja trenutnega stanja

Turingov stranj: končni avtomat + trak

Sekvenčna vezja

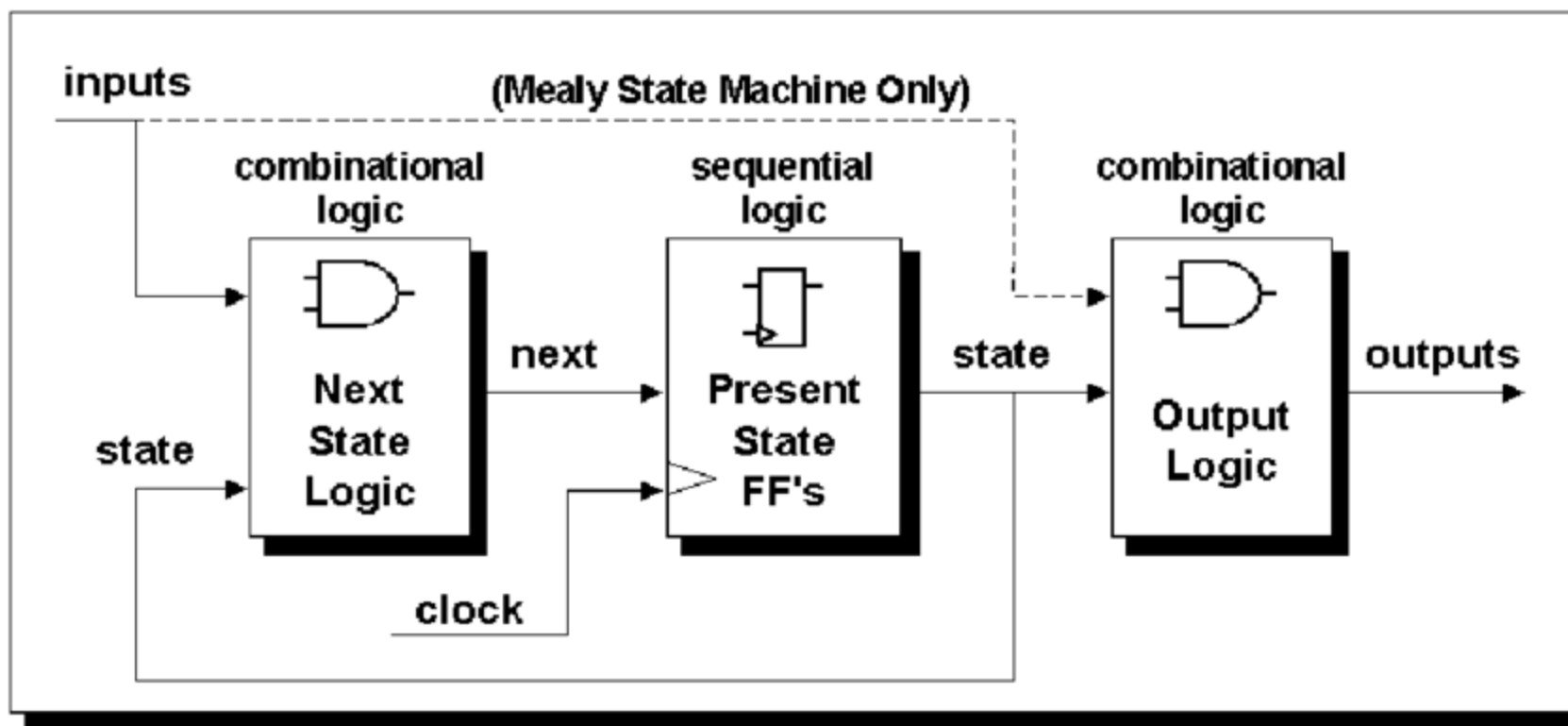
Končni avtomat (finite-state machine)

Način predstavitve sekvenčnih vezij

Višji nivo abstrakcije

Lažje načrtovanje kompleksnejših vezij

Končni avtomati



Končni avtomat

- *Finite state machine (FSM)*

- **Definicija:**

$$A = \{X, S, Z, \delta, \lambda\},$$

X ... neprazna, končna množica vhodnih črk,

S ... neprazna končna množica notranjih črk (stanj),

Z ... neprazna končna množica izhodnih črk,

δ ... funkcija naslednjega notranjega stanja (funkcija prehajanja stanj),

λ ... funkcija naslednje izhodne črke (izhodna funkcija).

Končni avtomat

Osnovna tipa avtomatov:

- Moorov avtomat,
- Mealyjev avtomat.

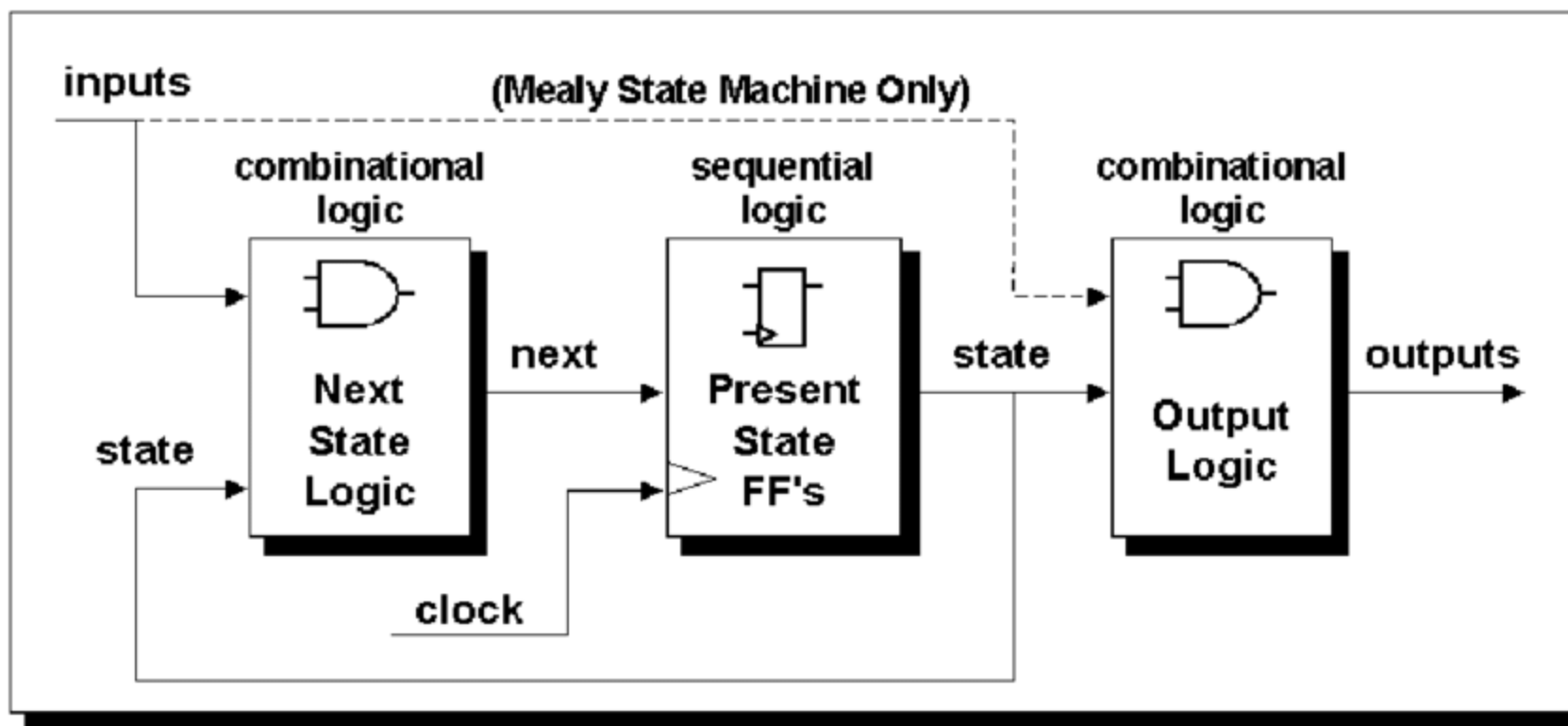
Predstavitev:

- Diagram prehajanja stanj,
- Tabela prehajanja stanj.

Sinhroni avtomati:

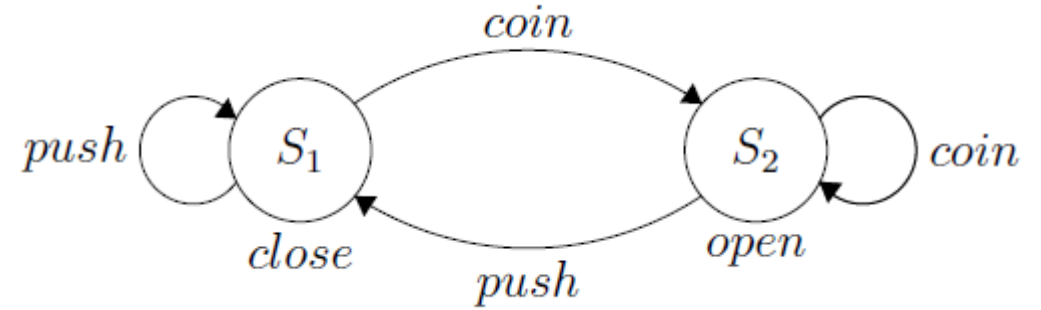
- prehode med stanji proži urin signal,
- najpogostejša uporaba.

Končni avtomati



Primer

Avtomat, ki kontrolira plačljiva krožna vrata (npr. podzemna železnica).



	<i>close</i>	<i>open</i>
<i>coin</i>	S_2	S_2
<i>push</i>	S_1	S_1

Moorov avtomat

Naslednje stanje je določeno z vhodno črko in stanjem avtomata:

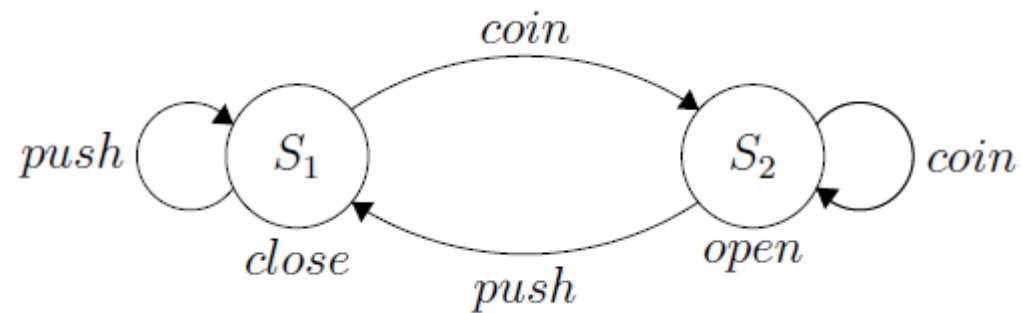
$$\delta: X \times S \rightarrow S$$

Izhodna črka je določena s stanjem avtomata:

$$\lambda: S \rightarrow Z$$

Primer

Mooreov avtomat, ki kontrolira plačljiva krožna vrata (npr. podzemna železnica).



	<i>close</i>	<i>open</i>
<i>coin</i>	S_2	S_2
<i>push</i>	S_1	S_1

Mealyjev avtomat

Naslednje stanje je določeno z vhodno črko in stanjem avtomata:

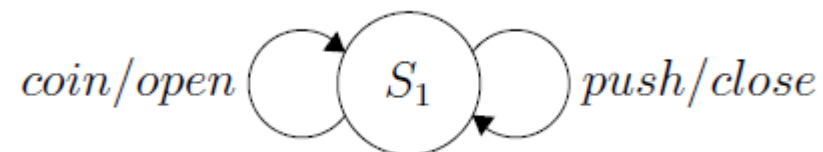
$$\delta : X \times S \rightarrow S$$

Izhodna črka je določena s stanjem in z vhodno črko (s prehodom med stanji):

$$\lambda : X \times S \rightarrow Z$$

Primer

Mealyjev avtomat, ki kontrolira plačljiva krožna vrata (npr. podzemna železnica).



	S_1
$coin$	$S_1/open$
$push$	$S_1/close$

Tabelaričen zapis

	izhodna črka (Moore)
	trenutno stanje
vhodna črka	naslednje stanje
	+
	izhodna črka (Mealy)

Diagram prehajanja stanj

Diagram prehajanja stanj je podan z usmerjenim grafom, katerega vozlišča določajo stanja avtomata, povezave med vozlišči pa opisujejo prehode med stanji ob prisotnosti vhodnih črk

V primeru Moorovega avtomata stanjem pripišemo izhodno črko

Ta je v primeru Mealyjevega avtomata vezana na prehode med stanji

Primer (Moore)

Nariši diagram prehajanja stanj za avtomat podan s tabelo

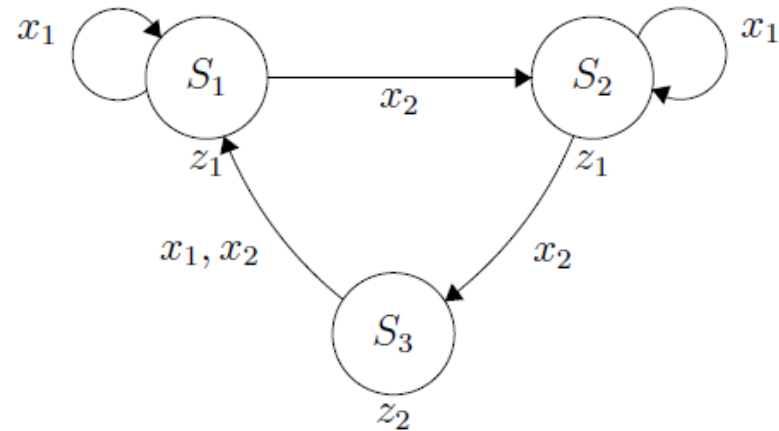
	z_1	z_1	z_2
	S_1	S_2	S_3
x_1	S_1	S_2	S_1
x_2	S_2	S_3	S_1

Primer

Najprej zapišimo vse tri abecede avtomata:

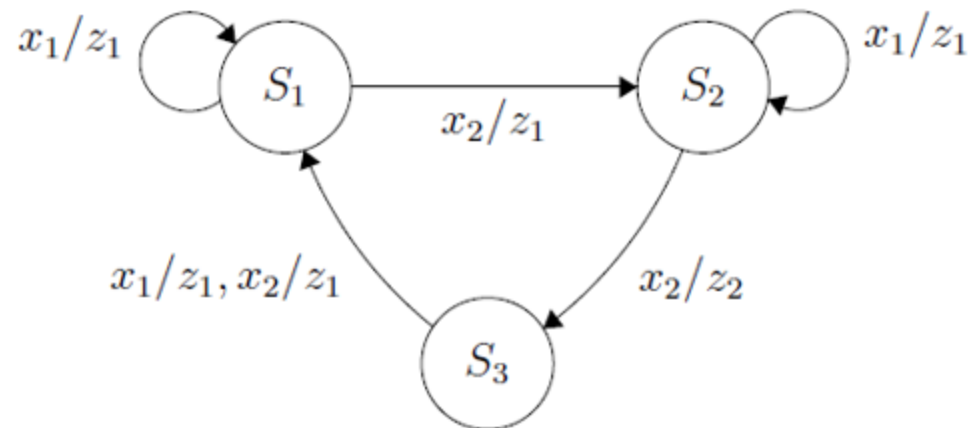
- $X = \{x_1, x_2\}$,
- $S = \{S_1, S_2, S_3\}$,
- $Z = \{z_1, z_2\}$.

Diagram prehajanja stanj ima torej tri vozlišča, iz vsakega vozlišča pa vodita največ dve povezavi (za vsako vhodno črko ena):

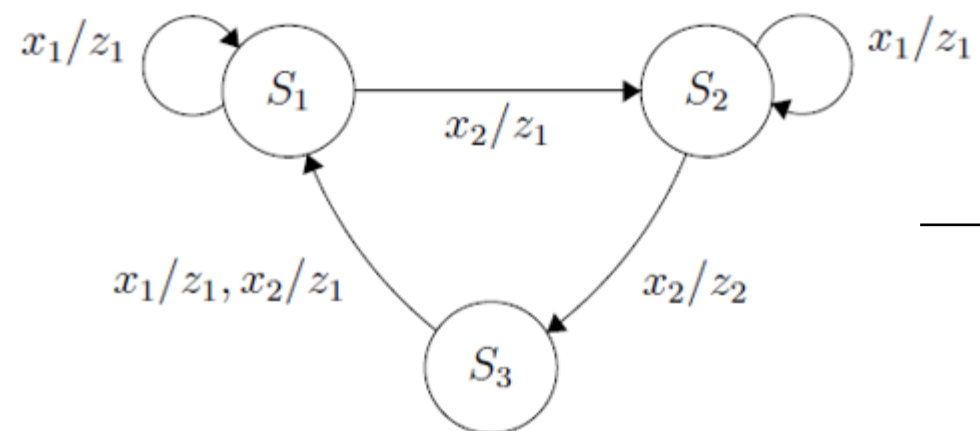


Primer (Mealy)

Zapiši tabelo prehajanja stanj za avtomat podan z diagramom



Primer (Mealy)



	S_1	S_2	S_3
x_1	S_1/z_1	S_2/z_1	S_1/z_1
x_2	S_2/z_1	S_3/z_2	S_1/z_1

Pretvorbe med avtomati

Za vsak Mooreov avtomat obstaja Mealyjev ekvivalent in obratno

Mealy \rightarrow Moore

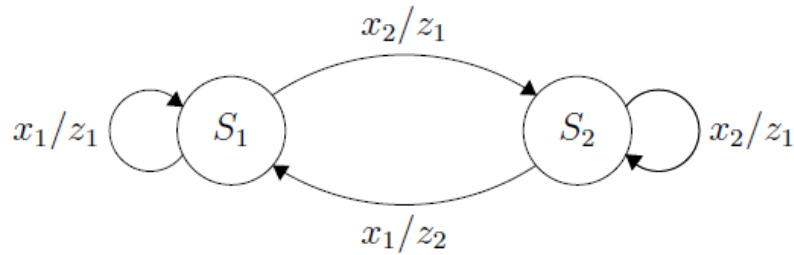
Zapišemo vse tri abecede Moorovega avtomata, pri čemer se vhodna in izhodna abecedi ohranjata, notranjo abeceda pa tvorimo s pari notranje stanje/izhodna crka, ki se pojavijo znotraj tabele Mealyjevega avtomata.

Zapišemo tabelo Moorovega avtomata, v kateri nastopajo vsa stanja, ki smo jih definirali v prejšnjem koraku.

Izhodna črka za stanje je določena z izhodno črko, katere oznaka nastopa pri posameznem stanju.

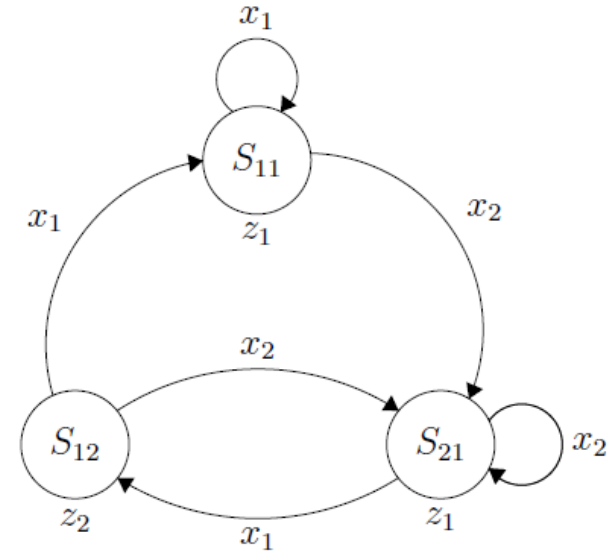
Prehodi med stanji so določeni s prehodi med ekvivalentimi stanji v Mealyjevem avtomatu.

Mealy \rightarrow Moore (primer)



	S_1	S_2
x_1	S_1/z_1	S_1/z_2
x_2	S_2/z_1	S_2/z_1

	z_1	z_2	z_1
	S_{11}	S_{12}	S_{21}
x_1	S_{11}	S_{11}	S_{12}
x_2	S_{21}	S_{21}	S_{21}



Moore \rightarrow Mealy

Moorov avtomat zapišemo tabelarično.

Zapišemo vse tri abecede Mealyjevega avtomata, ki so kar enake abecedam Moorovega avtomata.

Zapišemo tabelo Mealyjevega avtomata

Pri tem je naslednje stanje enako kot pri Moorovem avtomatu

Izhodna črka je določena z izhodno črko stanja Moorovega avtomata, v katerega bo avtomat z določeno kombinacijo stanje/vhodna črka prišel.

Moore \rightarrow Mealy (primer)

	z_1	z_2	z_1
	S_{11}	S_{12}	S_{21}
x_1	S_{11}	S_{11}	S_{12}
x_2	S_{21}	S_{21}	S_{21}



	S_{11}	S_{12}	S_{21}
x_1	S_{11}/z_1	S_{11}/z_1	S_{12}/z_2
x_2	S_{21}/z_1	S_{21}/z_1	S_{21}/z_1



	S_1	S_2
x_1	S_1/z_1	S_1/z_2
x_2	S_2/z_1	S_2/z_1

Izvedba avtomatov v digitalnih vezjih

Izvedeni kot sinhrona sekvenčna vezja:

- prehodi med stanji ob pozitivni (negativni) fronti ure
- kodiranje stanj avtomata in implementacija s pomnilnimi celicami

Mooreov avtomat: ker je izhodna črka vezana le na stanje, ki se spreminja sinhrono, se tudi izhodna črka spreminja sinhrono.

Mealyjev avtomat: izhodna črka je odvisna tudi od vhodne, ki se lahko spreminja asinhronsko. Pravilno načrtovanje: vhodne črke sinhroniziramo z urinim signalom (metastabilna stanja).

Osnovna izvedba avtomatov

Določimo kodiranje stanj avtomata (kudirne tabele).

Po potrebi določimo kudirne tabele vhodnih in izhodnih črk avtomata.

Na podlagi kudirnih tabel in diagrama/tabele prehajanja stanj avtomata določimo pravilnostno tabelo avtomata.

Izbira tipa pomnilnih celic in izvedba kot običajnih sekvenčnih vezij (uporaba vzbujevalne tabele, določanje kombinatornega dela vezja – preklopne funkcije,...)

Postopek

Zapis kodirnih tabel: vhodna abeceda, notranja abeceda in izhodna abeceda so predstavljene z abstraktnim zapisom. Za realizacijo s preklopnimi funkcijami moramo le-te predstaviti s preklopnimi spremenljivkami. Pri tem upoštevamo dejstvo, da lahko z i vhodnimi spremenljivkami zakodiramo 2^i vhodnih črk, z j izhodnimi spremenljivkami 2^j izhodnih črk, s k pomnilnimi celicami pa 2^k notranjih stanj. S kodiranimi tabelami povežemo posamezne spremenljivke oziroma notranja stanja pomnilnih celic s posameznimi črkami oziroma stanji avtomata.

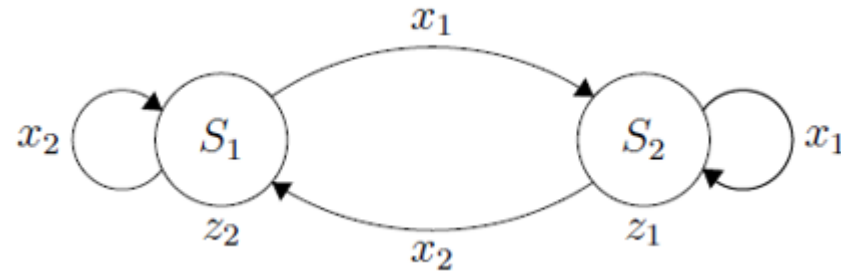
Zapis pravilnostne tabele avtomata: na podlagi diagrama prehajanja stanj oziroma tabele prehajanja stanj in kodirnih tabel, lahko zapišemo pravilnostno tabelo avtomata. Pri tem na levi strani tabele nastopajo spremenljivke, ki določajo vhodne črke in trenutna notranja stanja avtomata (neodvisne spremenljivke), na desni strani pa spremenljivke, ki določajo notranje stanje avtomata v naslednjem časovnem koraku in spremenljivke, ki določajo izhodne črke avtomata (odvisne spremenljivke).

Določitev vhodov v pomnilne celice: na podlagi prehodov med spremenljivkami, ki določajo trenutno stanje avtomata in stanje avtomata v naslednjem časovnem koraku ter vzbujevalnih tabel pomnilnih celic, ki jih imamo na razpolago, lahko določimo potrebne vhode v pomnilne celice v posamezni vrstici (tabelo dopolnimo na podoben način kot smo jo pri realizaciji sekvenčnih vezij s pomnilnimi celicami).

Izpis in minimizacija izhodne funkcije in funkcije prehajanja stanj: na podlagi pravilnostne tabele lahko s pomočjo minimizacije izpišemo preklopne funkcije, ki določajo izhodne črke avtomata (izhodna funkcija) in preklopne funkcije, ki nastopajo na vhodih pomnilnih celic in tako določajo prehode med stanji avtomata (funkcija prehajanja stanj).

Izvedba (primer 1)

Realiziraj avtomat z uporabo T pomnilne celice in poljubne kombinatorne logike



Izvedba (primer 1)

Zapis kodirnih tabel:

	x		q		y
x_1	0	S_1	0	z_1	0
x_2	1	S_2	1	z_2	1

Izvedba (primer 1)

Zapis pravilnostne tabele in izvedba sekvenčnega vezja:

q	x	D^1q	y	t
0	0	1	1	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

$$t = \bar{x} \bar{q} \vee xq = x \equiv q \qquad y = \bar{q}$$

Izvedba (primer 2)

Realiziraj avtomat z uporabo D pomnilnih celic in poljubne kombinatorne logike

	z_1	z_2	z_3
	S_1	S_2	S_3
x_1	S_1	S_1	S_1
x_2	S_2	S_3	S_2


Izvedba (primer 2)

Kodirne tabele:

	x		q_1	q_2		y_1	y_2
x_1	0	S_1	0	0	z_1	0	0
x_2	1	S_2	0	1	z_2	0	1
		S_3	1	0	z_3	1	0

Pravilnostna tabela:

q_1	q_2	x	D^1g_1	D^1g_2	y_1	y_2				
0	0	0	0	0	0	0		0	0	
0	0	1	0	1	0	0		0	1	
0	1	0	0	0	0	1		0	0	
0	1	1	1	0	0	1		1	0	
1	0	0	0	0	1	0		0	0	
1	0	1	0	1	1	0		0	1	
1	1	0	?	?	?	?		?	?	
1	1	1	?	?	?	?		?	?	



q_1	q_2	x	D^1g_1	D^1g_2	y_1	y_2	d_1	d_2
0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	1	0	0	1	1	0
1	0	0	0	0	1	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	?	?	?	?	?	?
1	1	1	?	?	?	?	?	?

Izvedba (primer 2)

Kodirne tabele:

	x		q_1	q_2		y_1	y_2
x_1	0	S_1	0	0	z_1	0	0
x_2	1	S_2	0	1	z_2	0	1
		S_3	1	0	z_3	1	0

Pravilnostna tabela:

q_1	q_2	x	D^1g_1	D^1g_2	y_1	y_2
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	1	0	1	1	0
1	1	0	?	?	?	?
1	1	1	?	?	?	?

Funkcije:

$$d_1 = q_2x \qquad d_2 = \bar{q}_2x$$

$$y_1 = q_1 \qquad y_2 = q_2$$

q_1	q_2	x	D^1g_1	D^1g_2	y_1	y_2	d_1	d_2
0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	1	0	0	1	1	0
1	0	0	0	0	1	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	?	?	?	?	?	?
1	1	1	?	?	?	?	?	?

Izvedba (primer 3)

Z uporabo logičnih vrat in pomnilnih celic D realiziraj avtomat.

Primer: Mooreov avtomat, ki opisuje ključavnico, ki sprejme sekvenco 1101

	0	0	0	0	1
	S_0	S_1	S_2	S_3	S_4
0	S_0	S_0	S_3	S_0	S_0
1	S_1	S_2	S_2	S_4	S_0

Osnovno kodiranje, uporaba D pomnilnih celic.

Izvedba (primer 4)

Z uporabo logičnih vrat in D pomnilnih celic realiziraj avtomat.

Primer: Mealyjev avtomat, ki opisuje ključavnico, ki sprejme sekvenco 1101

	S_0	S_1	S_2	S_3
0	$S_0/0$	$S_0/0$	$S_3/0$	$S_0/0$
1	$S_1/0$	$S_2/0$	$S_2/0$	$S_0/1$

Osnovno kodiranje, uporaba D pomnilnih celic.