WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

Fundamentals of Logic Circuit Design

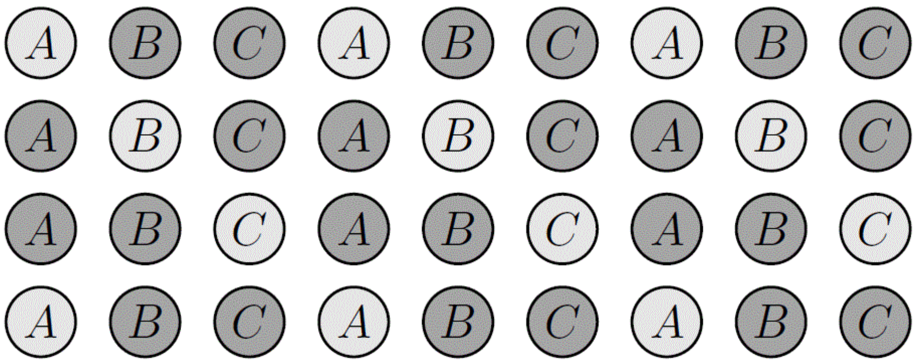
Part 2: Sequential Logic

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!

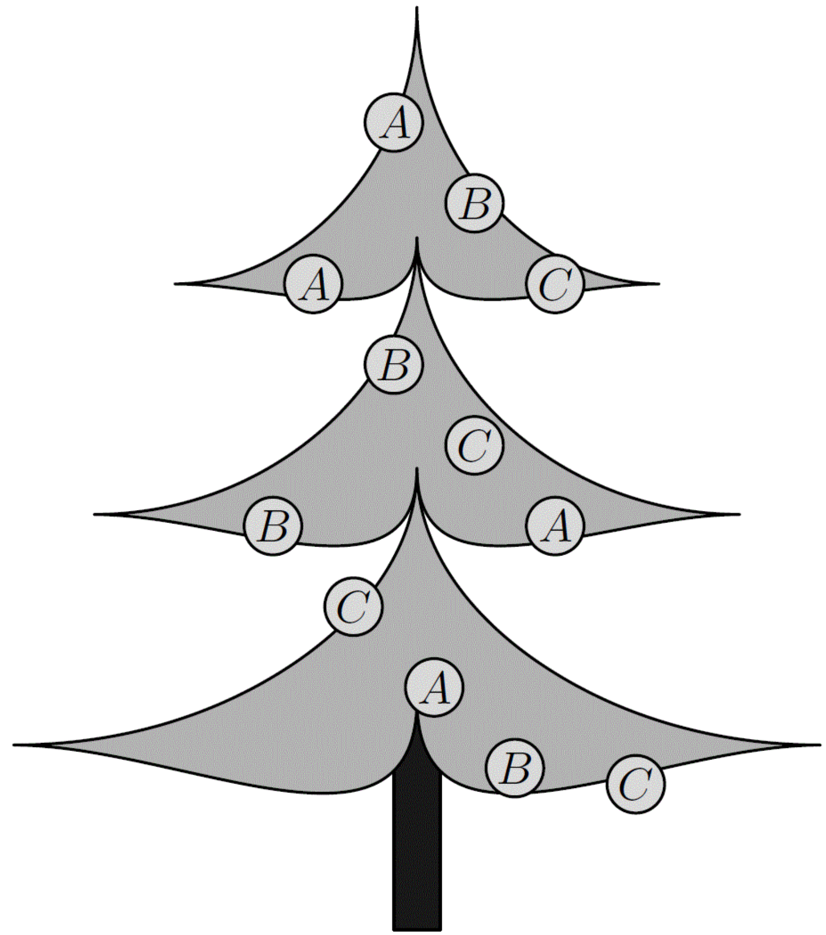
EUROPEAN UNION
EUROPEAN
SOCIAL FUND

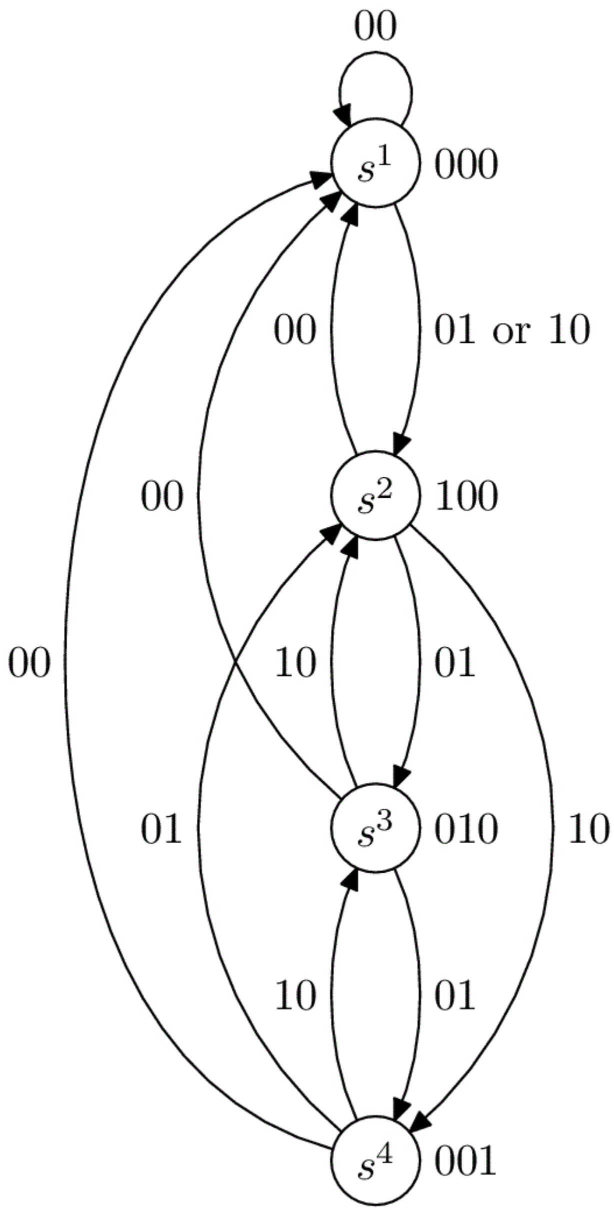
Project is co-financed by European Union within European Social Fund

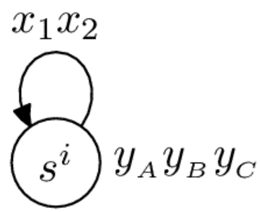
X'mas tree lights



x_1	x_2	action
0	0	all off
0	1	$ABCABC\ldots$
1	0	$CBACBA\ldots$
1	1	not permitted







state \ x_1x_2	00	01	11	10	$y_Ay_By_C$
s^1	s^1	s^2	—	s^2	000
s^2	s^1	s^3	—	s^4	100
s^3	s^1	s^4	—	s^2	010
s^4	s^1	s^2	—	s^3	001

$state'/y$

State transition and output table for the X'mas tree automaton

Coded state transition and output table for the X'mas tree automaton

state	q_1	q_2
s^1	0	0
s^2	0	1
s^3	1	0
s^4	1	1

State assignment: $state \to code$

q_1q_2 \ x_1x_2	00	01	11	10	$y_Ay_By_C$
00	00	01	–	01	000
01	00	10	–	11	100
10	00	11	–	01	010
11	00	01	–	10	001

$q_1'q_2'/y$

NB code

q_1q_2 \ x_1x_2	00	01	11	10	$y_Ay_By_C$
00	00	01	–	01	000
01	00	10	–	11	100
11	00	01	–	10	001
10	00	11	–	01	010

$q_1'q_2'/y$

Gray's code

$q_1'q_2'$

q_1q_2 \ x_1x_2	00	01	11	10
00	00	01	–	01
01	00	10	–	11
11	00	01	–	10
10	00	11	–	01

q_1'

q_1q_2 \ x_1x_2	00	01	11	10
00	0	0	–	0
01	0	1	–	1
11	0	0	–	1
10	0	1	–	0

q_2'

q_1q_2 \ x_1x_2	00	01	11	10
00	0	1	–	1
01	0	0	–	1
11	0	1	–	0
10	0	1	–	1

q_1	q_2	y_A	y_B	y_C
0	0	0	0	0
0	1	1	0	0
1	1	0	0	1
1	0	0	1	0

y_A

q_1 \ q_2	0	1
0	0	1
1	0	0

y_B

q_1 \ q_2	0	1
0	0	0
1	1	0

y_C

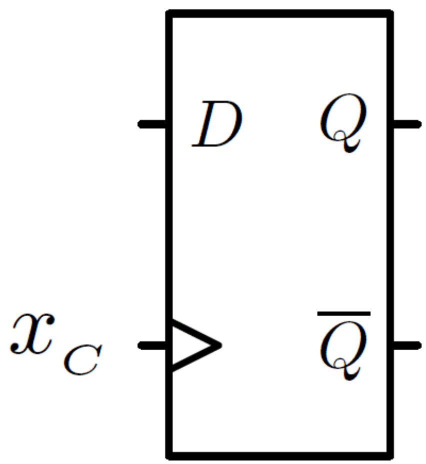
q_1 \ q_2	0	1
0	0	0
1	0	1

D-type flip-flop

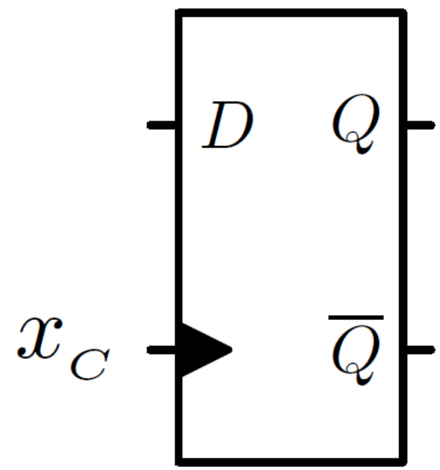
Q	$\to$	Q'	D
0	$\to$	0	0
0	$\to$	1	1
1	$\to$	0	0
1	$\to$	1	1

Q'

$Q \backslash D$	0	1
0	0	1
1	0	1



positive edge triggered



negative edge triggered

$q_1q_2 \backslash x_1x_2$	00	01	11	10
00	0	0	–	0
01	0	1	–	1
11	0	0	–	1
10	0	1	–	0

$D_1 = q_1'$

$q_1q_2 \backslash x_1x_2$	00	01	11	10
00	0	1	–	1
01	0	0	–	1
11	0	1	–	0
10	0	1	–	1

$D_2 = q_2'$

$q_1 \backslash q_2$	0	1
0	0	1
1	0	0

y_A

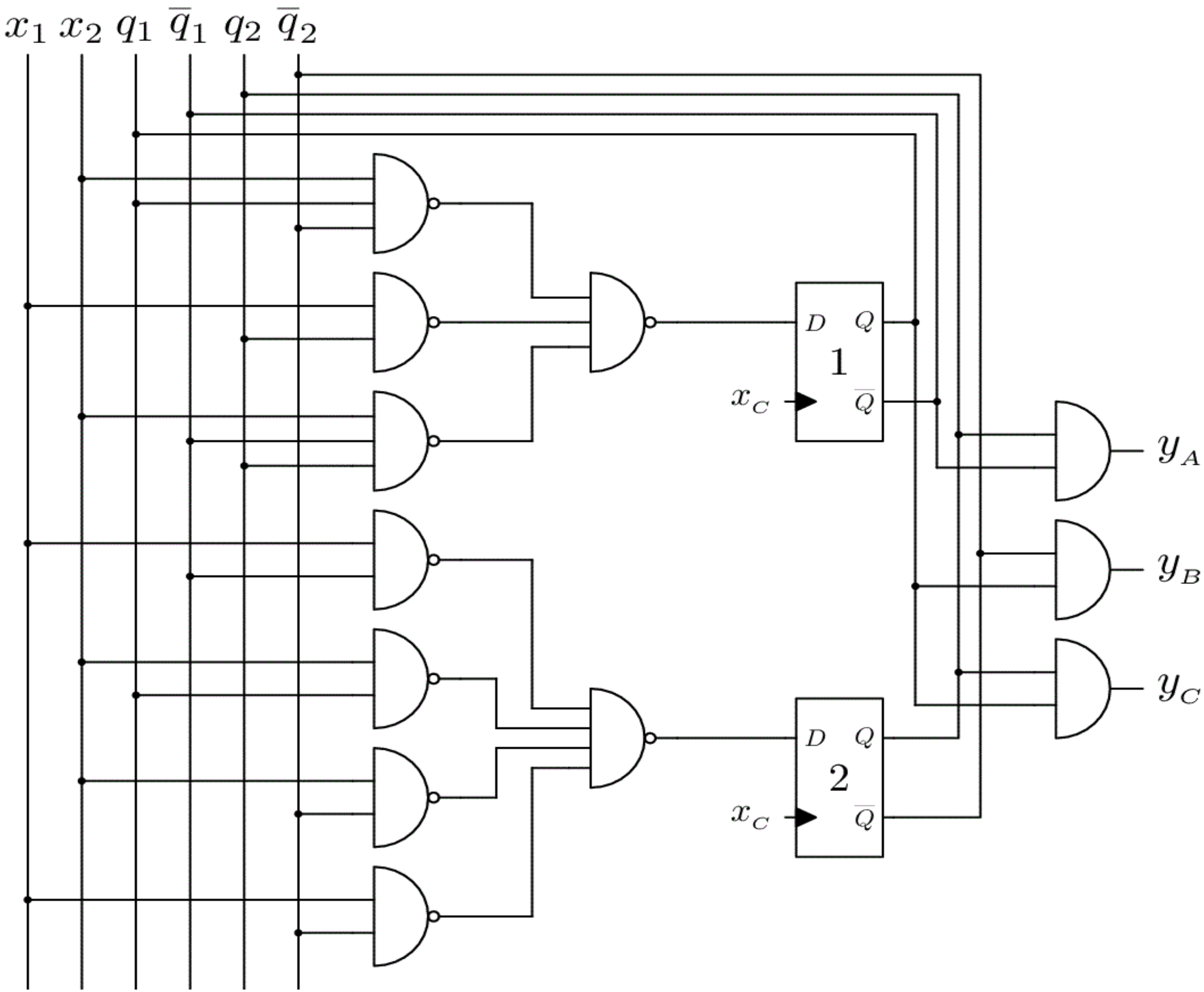
$q_1 \backslash q_2$	0	1
0	0	0
1	1	0

y_B

$q_1 \backslash q_2$	0	1
0	0	0
1	0	1

y_C

$$
\begin{cases}
q_1' &= \overline{q}_1 q_2 x_2 + q_1 \overline{q}_2 x_2 + q_2 x_1 \\
q_2' &= \overline{q}_1 x_1 + q_1 x_2 + \overline{q}_2 x_2 + \overline{q}_2 x_1 \\
y_A &= \overline{q}_1 q_2 \\
y_B &= q_1 \overline{q}_2 \\
y_C &= q_1 q_2
\end{cases}
$$


x_1 x_2 q_1 $\overline{q}_1$ q_2 $\overline{q}_2$
D
Q
1
x_C
$\overline{Q}$
D
Q
2
x_C
$\overline{Q}$
y_A
y_B
y_C

Version 2 of the X'mas tree automaton

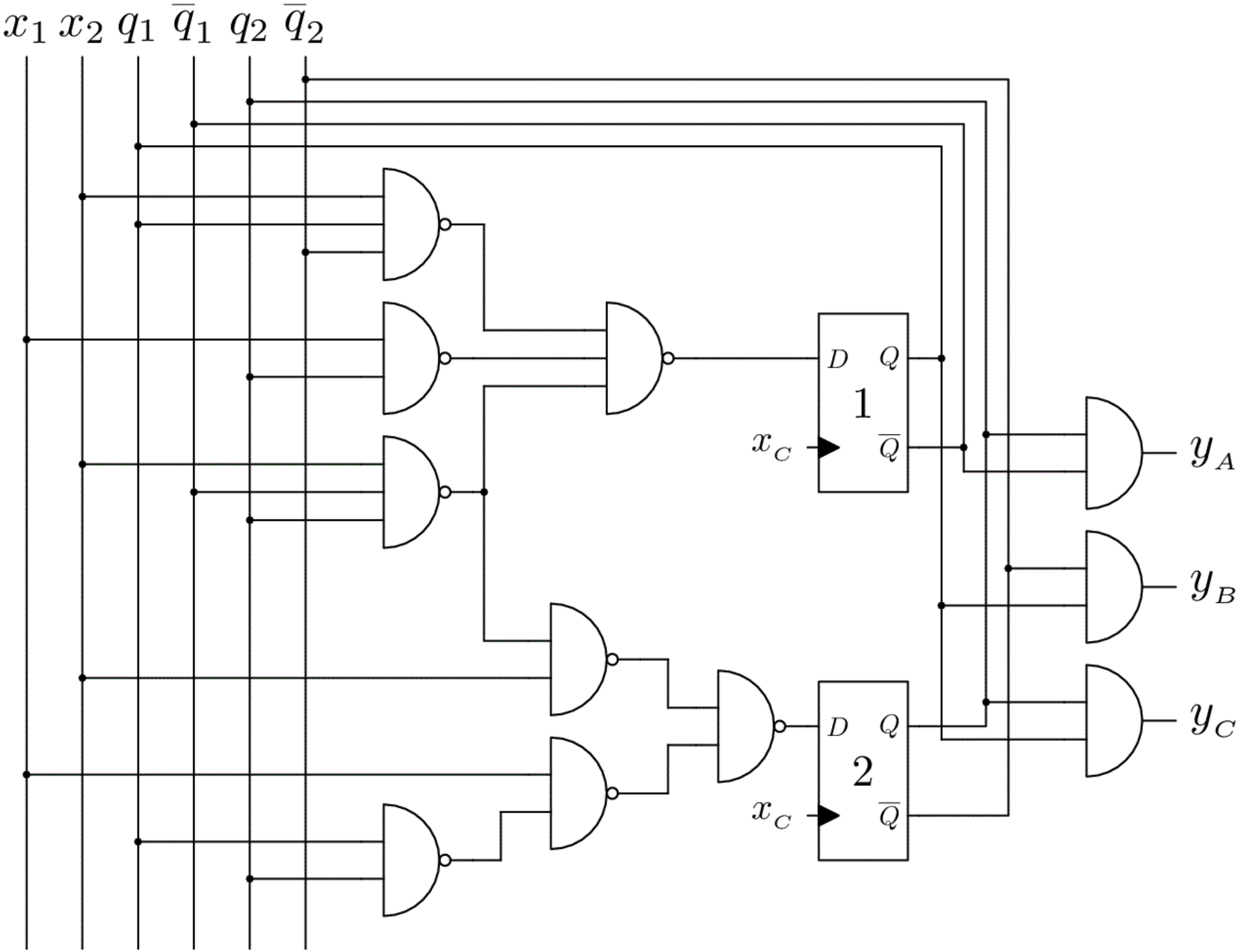
$q_1q_2 \backslash x_1x_2$	00	01	11	10
00	0	0	–	0
01	0	1	–	1
11	0	0	–	1
10	0	1	–	0

$D_1 = q_1'$

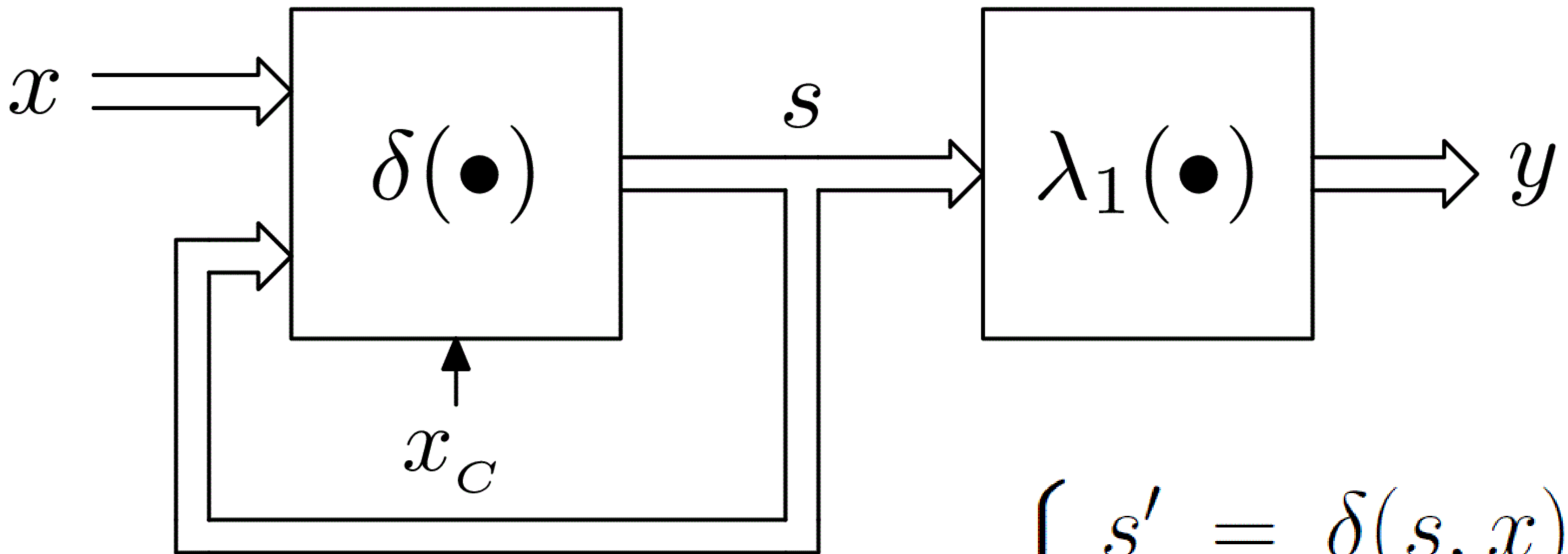
$q_1q_2 \backslash x_1x_2$	00	01	11	10
00	0	1	–	1
01	0	0	–	1
11	0	1	–	0
10	0	1	–	1

$D_2 = q_2'$

$$\begin{cases} q_1' &= \overline{q}_1 q_2 x_2 + q_2 x_1 + q_1 \overline{q}_2 x_2 \\ q_2' &= x_2 \overline{\overline{q}_1 q_2 x_2} + x_1 \overline{q_1 q_2} \\ y_A &= \overline{q}_1 q_2 \\ y_B &= q_1 \overline{q}_2 \\ y_C &= q_1 q_2 \end{cases}$$

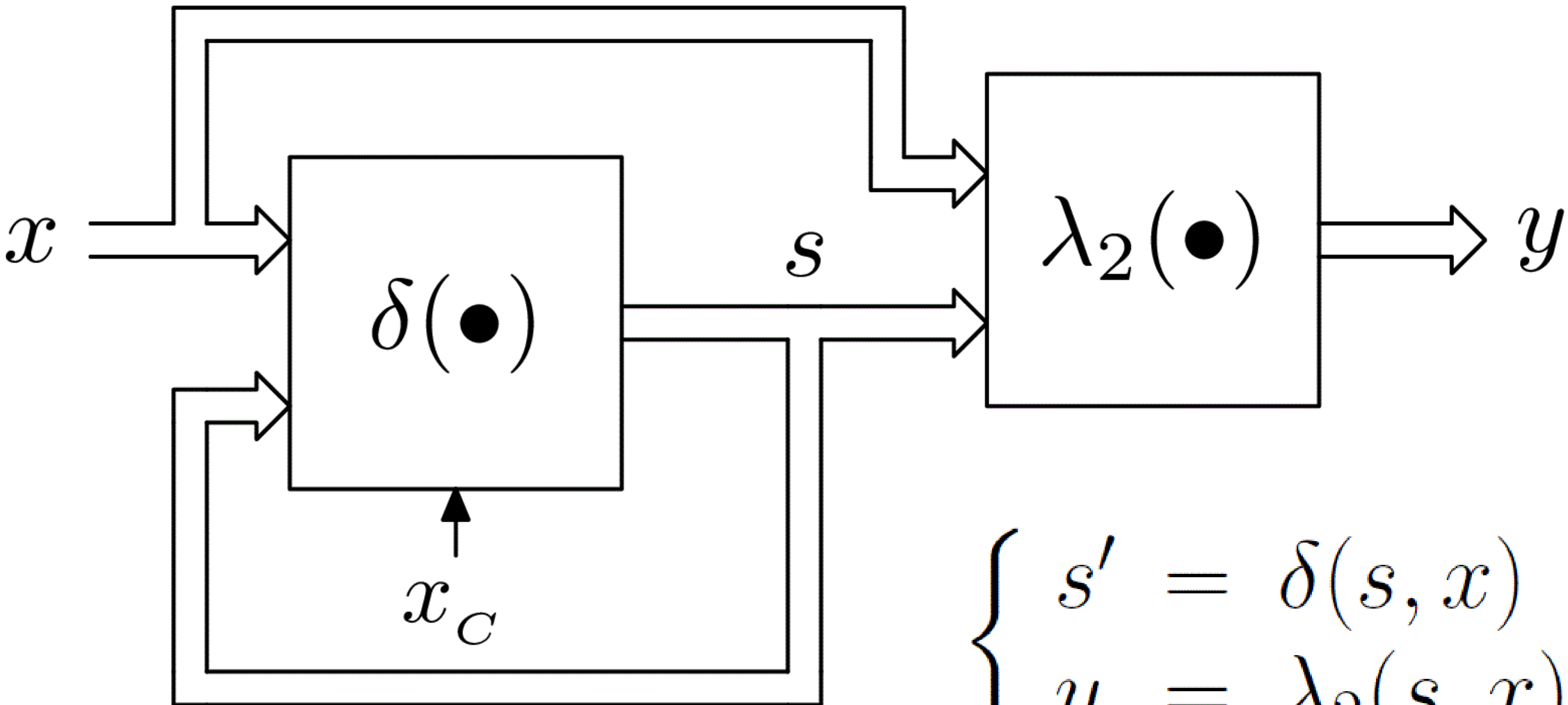

x_1 x_2 q_1 $\overline{q}_1$ q_2 $\overline{q}_2$
D
Q
1
x_C
$\overline{Q}$
D
Q
2
x_C
$\overline{Q}$
y_A
y_B
y_C

Structure of synchronous Moore automaton



$$\begin{cases} s' &= \delta(s, x) \\ y &= \lambda_1(s) \end{cases}$$

Structure of synchronous Mealy automaton



$$
\begin{cases} s' = \delta(s, x) \\ y = \lambda_2(s, x) \end{cases}
$$

Basic definitions

input:	x	input set:	$\boldsymbol{X}$	$x \in \boldsymbol{X}$
output:	y	output set:	$\boldsymbol{Y}$	$y \in \boldsymbol{Y}$
state:	s	state set:	$\boldsymbol{S}$	$s \in \boldsymbol{S}$

Def: automaton

$$A = \left(\boldsymbol{S}, \boldsymbol{X}, \boldsymbol{Y}, \delta, \lambda\right)$$

Def: completely specified automaton

If functions δ and λ for an automaton A are defined for each pair (s, x) from the set $\boldsymbol{S} \times \boldsymbol{X}$ the automaton is called complete (fully defined). Otherwise the automaton is considered incompletely specified (not fully defined).

Mathematical formulation for a Moore automaton:

Automaton $A = (\boldsymbol{S}, \boldsymbol{X}, \boldsymbol{Y}, \delta, \lambda_1)$ is completely specified if

$$\forall (s, x) \in \boldsymbol{S} \times \boldsymbol{X} \quad \exists \delta(s, x) \ \bigwedge \ \exists \lambda_1(s)$$

otherwise it is incompletely specified.

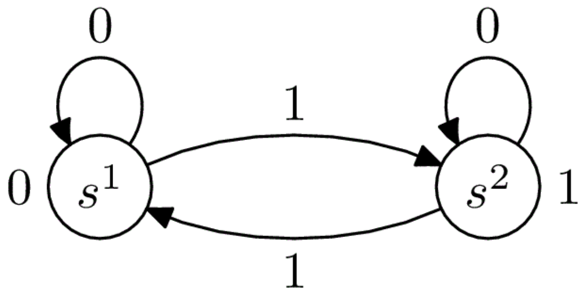
Mathematical formulation for a Mealy automaton:

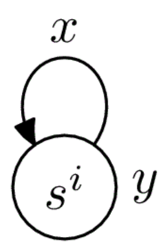
Automaton $A = (\boldsymbol{S}, \boldsymbol{X}, \boldsymbol{Y}, \delta, \lambda_2)$ is completely specified if

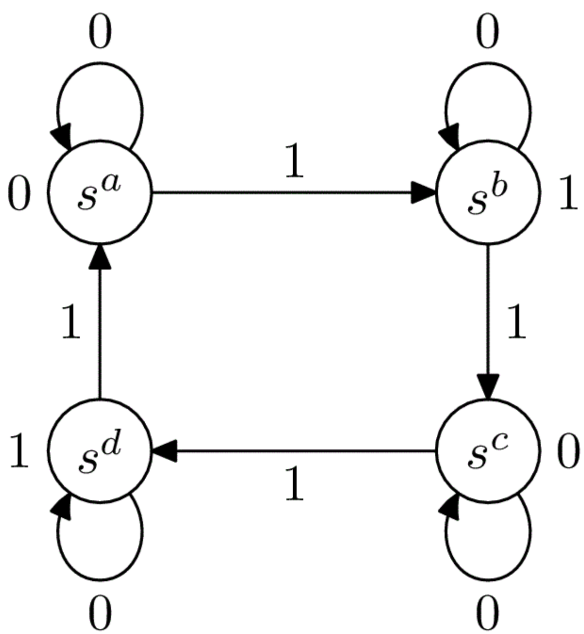
$$\forall (s, x) \in \boldsymbol{S} \times \boldsymbol{X} \quad \exists \delta(s, x) \ \bigwedge \ \exists \lambda_2(s, x)$$

otherwise it is incompletely specified.

Equivalence of states of automatons







$$s^i/y \xrightarrow{x}$$

$$s^1/0 \xrightarrow{0} s^1/0 \xrightarrow{1} s^2/1 \xrightarrow{0} s^2/1 \xrightarrow{1} s^1/0 \xrightarrow{1} s^2/1 \xrightarrow{1} s^1/0$$

$$s^a/0 \xrightarrow{0} s^a/0 \xrightarrow{1} s^b/1 \xrightarrow{0} s^b/1 \xrightarrow{1} s^c/0 \xrightarrow{1} s^d/1 \xrightarrow{1} s^a/0$$

$$\Rightarrow \quad s^1 \quad \equiv \quad s^a$$

Equivalence of states of automatons

Input sequence: $x^* = {}^1x \; {}^2x \; {}^3x \ldots; \quad {}^kx \in \boldsymbol{X}$

Output sequence: $y^* = {}^1y \; {}^2y \; {}^3y \ldots; \quad {}^ky \in \boldsymbol{Y}$

State sequence: $s^* = {}^1s \; {}^2s \; {}^3s \ldots; \quad {}^ks \in \boldsymbol{S}$

Def.1: Two states s^i and s^j of a completely specified automaton are equivalent, if for any input sequence x^* applied to each of them the same output sequence y^* will result in both cases.

Mathematical formulation for a completely specified Moore automaton:

$$s^i \equiv s^j \iff \forall x^* \;\; \lambda_1^*\big(\delta^*(s^i, x^*)\big) = \lambda_1^*\big(\delta^*(s^j, x^*)\big)$$

where:

$$\begin{aligned} \delta^*(s, x^*) \; &= \; s \;\; \delta(s, {}^1x) \;\; \delta\big(\delta(s, {}^1x), {}^2x\big) \ldots &= \;\; s \;\; {}^1s \;\; {}^2s \ldots \\ \lambda_1^*(s^*) \; &= \; \lambda_1(s) \;\; \lambda_1({}^1s) \;\; \lambda_1({}^2s) \ldots &= {}^0y \;\; {}^1y \;\; {}^2y \ldots \end{aligned}$$

Mathematical formulation for a completely specified Mealy automaton:

$$s^i \equiv s^j \iff \forall x^* \;\; \lambda_2^{**}\big(\delta^*(s^i, x^*), x^*\big) = \lambda_2^{**}\big(\delta^*(s^j, x^*), x^*\big)$$

where:

$$\begin{array}{lcll}
\delta^*(s, x^*) & = & s \;\; \delta(s, {}^1x) \;\; \delta\big(\delta(s, {}^1x), {}^2x\big) \ldots & = s \;\; {}^1s \;\; {}^2s \ldots \\
\lambda_2^{**}(s^*, x^*) & = & \lambda_2^*(s, x^{*1}) \;\; \lambda_2^*({}^1s, x^{*2}) \ldots & = y^{*0} \; y^{*1} \ldots \\
y^{*k} & = & {}^{k_1}y \ldots {}^{k_m}y & \\
\lambda_2^*({}^ks, x^{*k}) & = & \lambda_2({}^ks, {}^{k_1}x) \ldots \lambda_2({}^ks, {}^{k_m}x) & = {}^{k_1}y \ldots {}^{k_m}y \\
x^* & = & x^{*1} \;\; x^{*2} \;\; x^{*3} \ldots & \\
x^{*k} & = & {}^{k_1}x \ldots {}^{k_n}x & \\
{}^kx & = & {}^{k_n}x &
\end{array}$$

Equivalence of states of automatons

Def.2: Two states s^i and s^j of completely specified automatons are equivalent, if for any input $x \in \boldsymbol{X}$:

- outputs in both states are equal
- next states are the same or equivalent

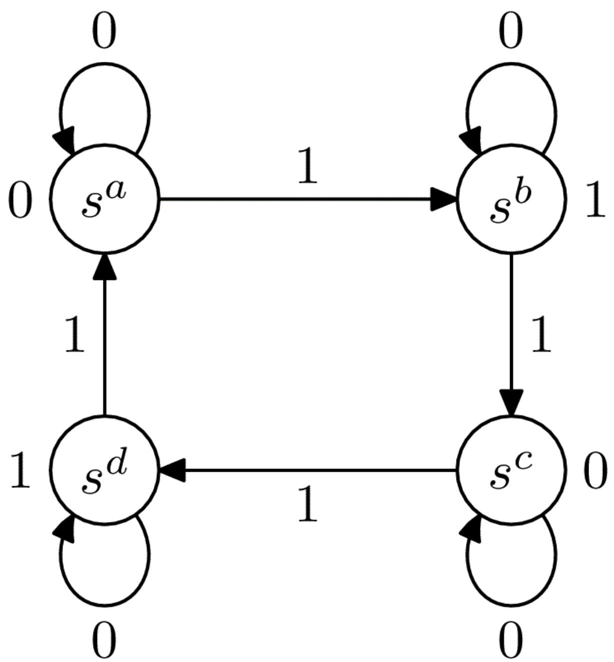
Mathematical formulation for completely specified Moore automatons:

$$s^i \equiv s^j \iff \forall x \; \Big(\lambda_1(s^i) = \lambda_1(s^j)\Big) \bigwedge \Big(\big(\delta(s^i,x) = \delta(s^j,x)\big) \bigvee \big(\delta(s^i,x) \equiv \delta(s^j,x)\big)\Big)$$

Mathematical formulation for completely specified Mealy automatons:

$$s^i \equiv s^j \iff \forall x \; \Big(\lambda_2(s^i,x) = \lambda_2(s^j,x)\Big) \bigwedge \Big(\big(\delta(s^i,x) = \delta(s^j,x)\big) \bigvee \big(\delta(s^i,x) \equiv \delta(s^j,x)\big)\Big)$$

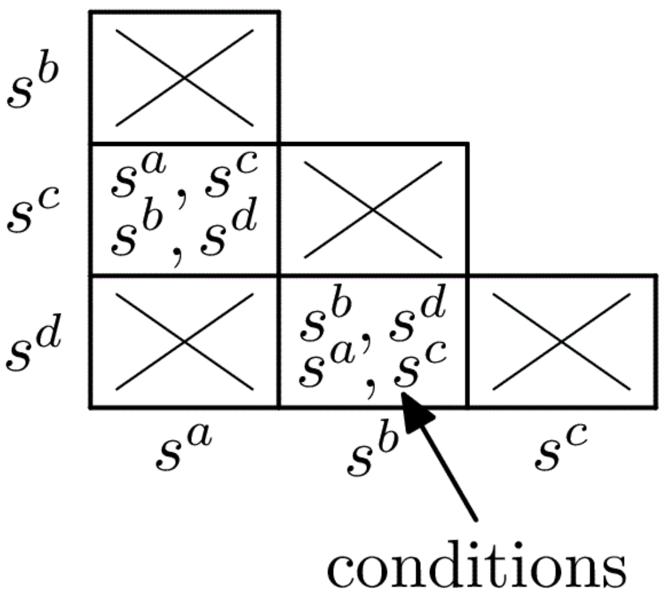
Initial graph and state transition table



$s \backslash x$	0	1	y
s^a	s^a	s^b	0
s^b	s^b	s^c	1
s^c	s^c	s^d	0
s^d	s^d	s^a	1

s'/y

Implication matrix



$\Rightarrow$ looped conditions are fulfilled

$$y = \lambda_1(s^a) = \lambda_1(s^c) = 0$$

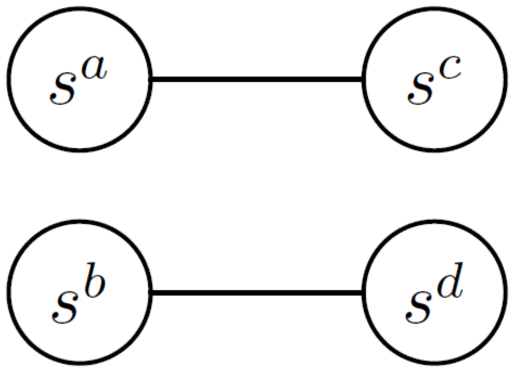
$$\begin{cases} \delta(s^a, 0) = s^a, & y = \lambda_1(s^a) = 0 \\ \delta(s^c, 0) = s^c, & y = \lambda_1(s^c) = 0 \end{cases}$$

$$\begin{cases} \delta(s^a, 1) = s^b, & y = \lambda_1(s^b) = 1 \\ \delta(s^c, 1) = s^d, & y = \lambda_1(s^d) = 1 \end{cases}$$

$$\begin{cases} \delta(s^b, 0) = s^b, & y = \lambda_1(s^b) = 1 \\ \delta(s^d, 0) = s^d, & y = \lambda_1(s^d) = 1 \end{cases}$$

$$\begin{cases} \delta(s^b, 1) = s^c, & y = \lambda_1(s^c) = 0 \\ \delta(s^d, 1) = s^a, & y = \lambda_1(s^a) = 0 \end{cases}$$

Equivalence graph

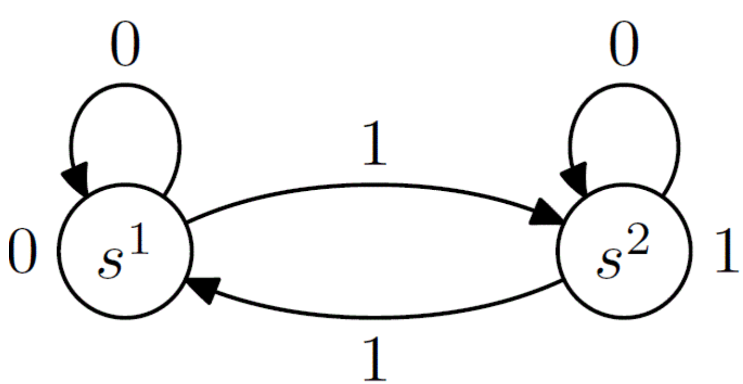


State equivalence

$$\{s^a, s^c\} \rightarrow s^1$$

$$\{s^b, s^d\} \rightarrow s^2$$

Graph of the minimized automaton

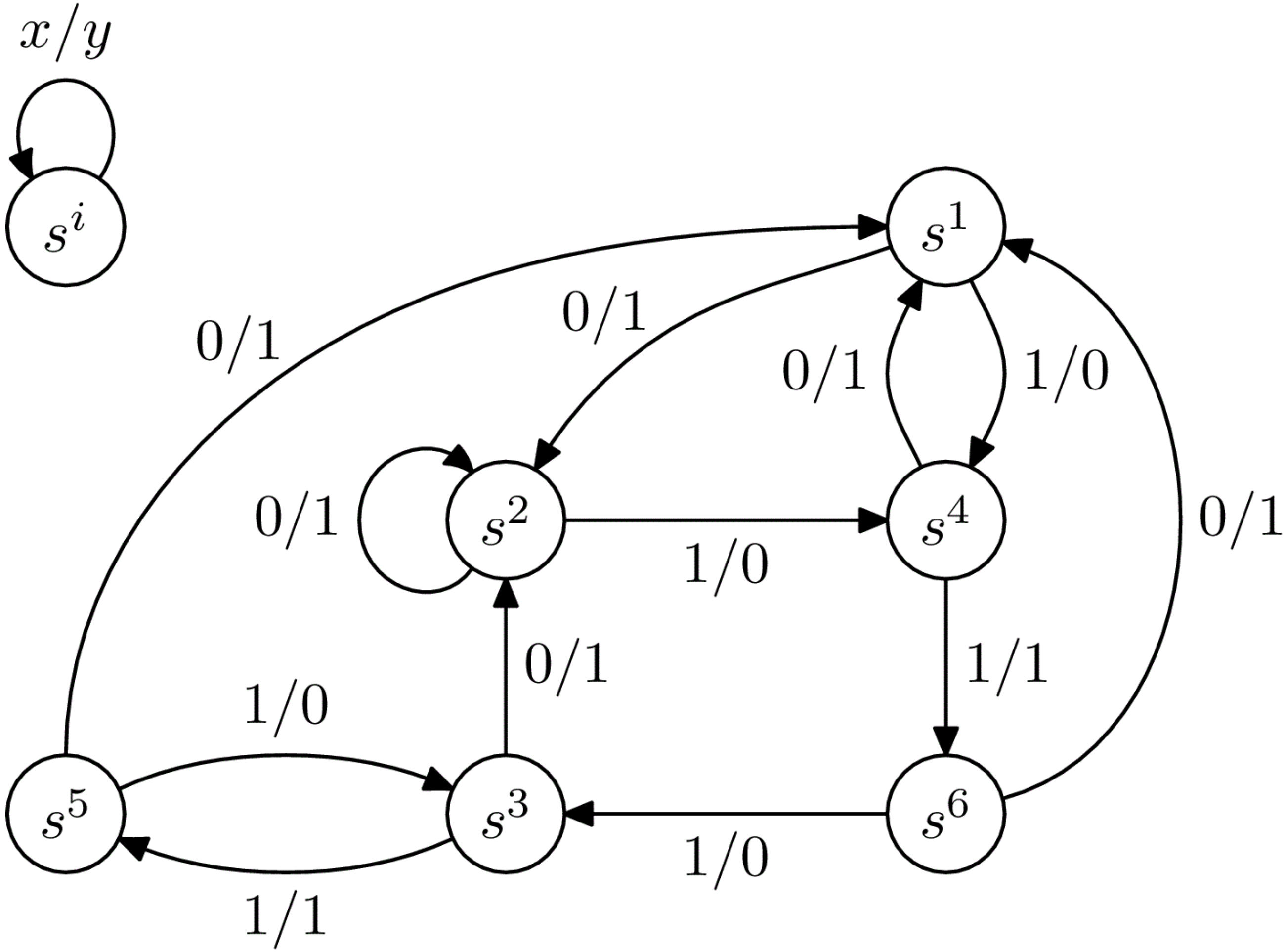


Its state transition table

$s \backslash x$	0	1	y
s^1	s^1	s^2	0
s^2	s^2	s^1	1

s'/y

Mealy automaton



State transition and output table

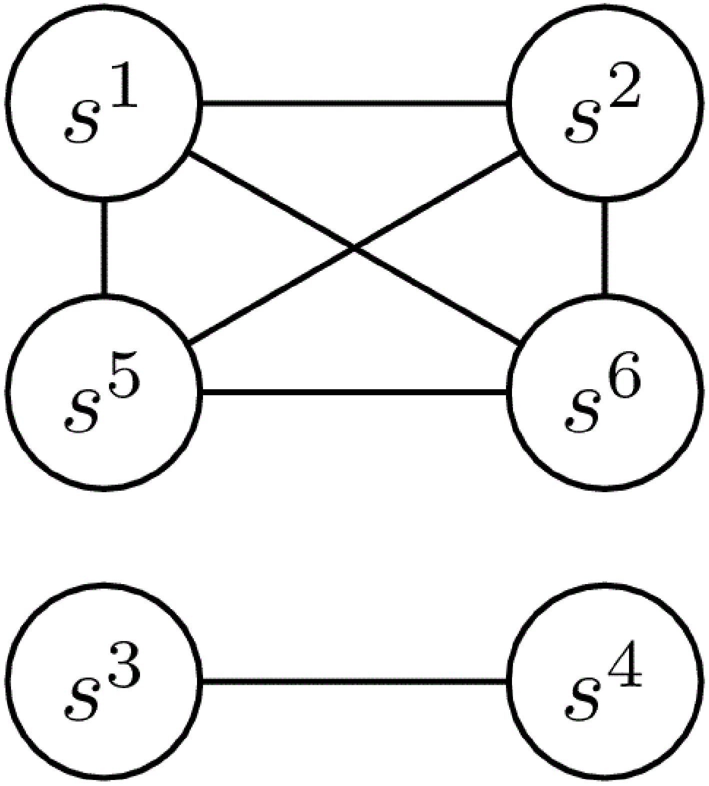
$s \backslash x$	0	1
s^1	$s^2/1$	$s^4/0$
s^2	$s^2/1$	$s^4/0$
s^3	$s^2/1$	$s^5/1$
s^4	$s^1/1$	$s^6/1$
s^5	$s^1/1$	$s^3/0$
s^6	$s^1/1$	$s^3/0$

s'/y

Implication matrix

s^2					
s^3	X	X			
s^4	X	X	s^1, s^2 s^5, s^6		
s^5	s^1, s^2 s^3, s^4	s^1, s^2 s^3, s^4	X	X	
s^6	s^1, s^2 s^3, s^4	s^1, s^2 s^3, s^4	X	X	
	s^1	s^2	s^3	s^4	s^5

Equivalence graph

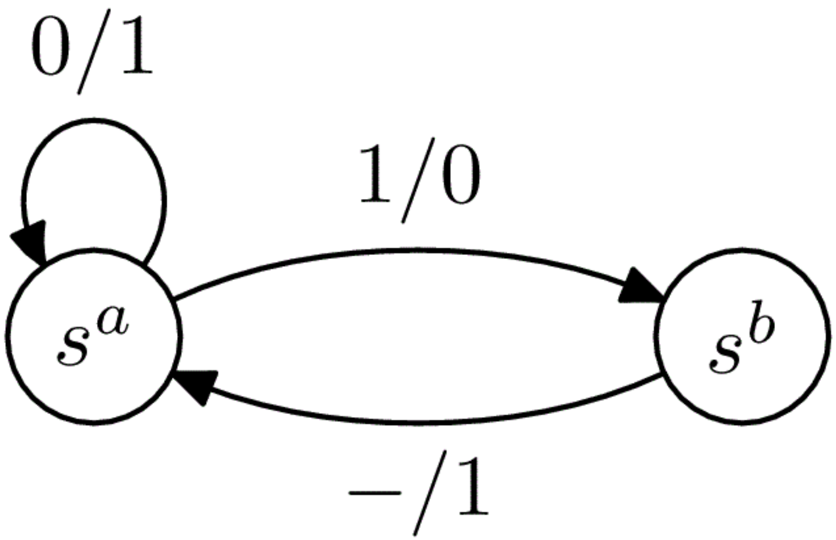


State equivalence

$$\{s^1, s^2, s^5, s^6\} \rightarrow s^a$$

$$\{s^3, s^4\} \rightarrow s^b$$

Graph of minimized automation

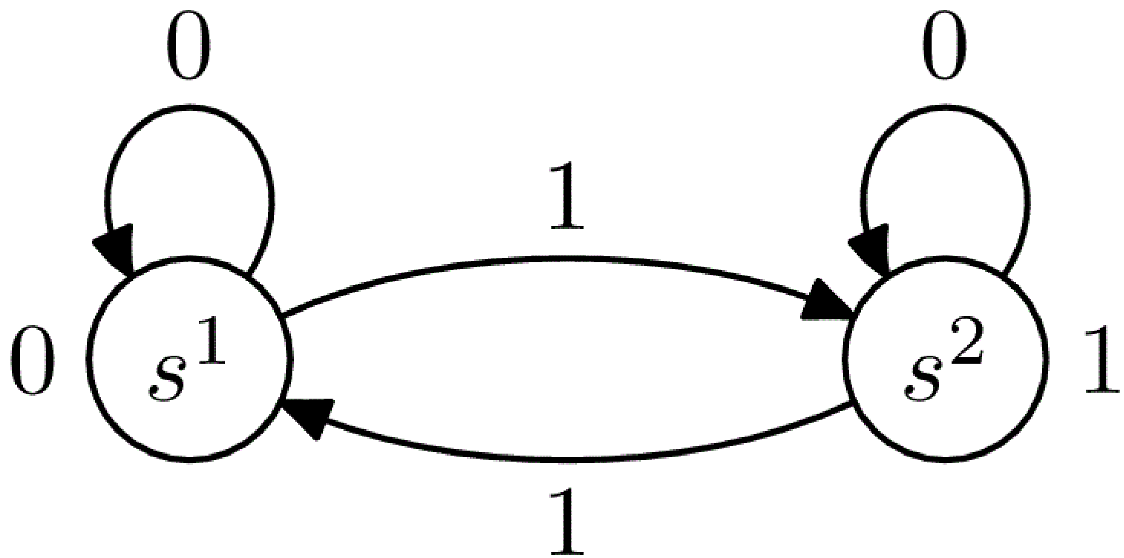


Its state transition table

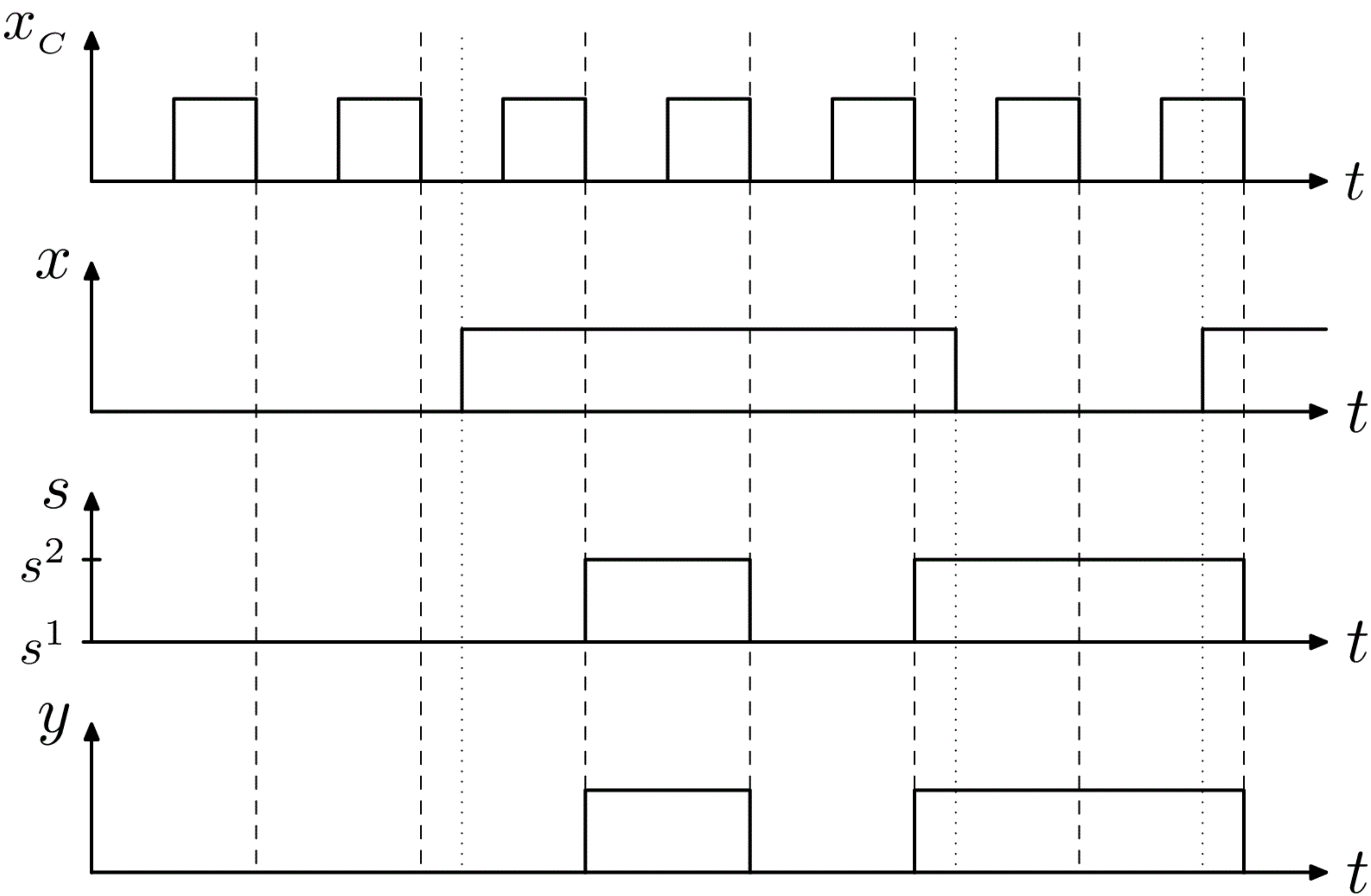
$s \backslash x$	0	1
s^a	$s^a/1$	$s^b/0$
s^b	$s^a/1$	$s^a/1$

s'/y

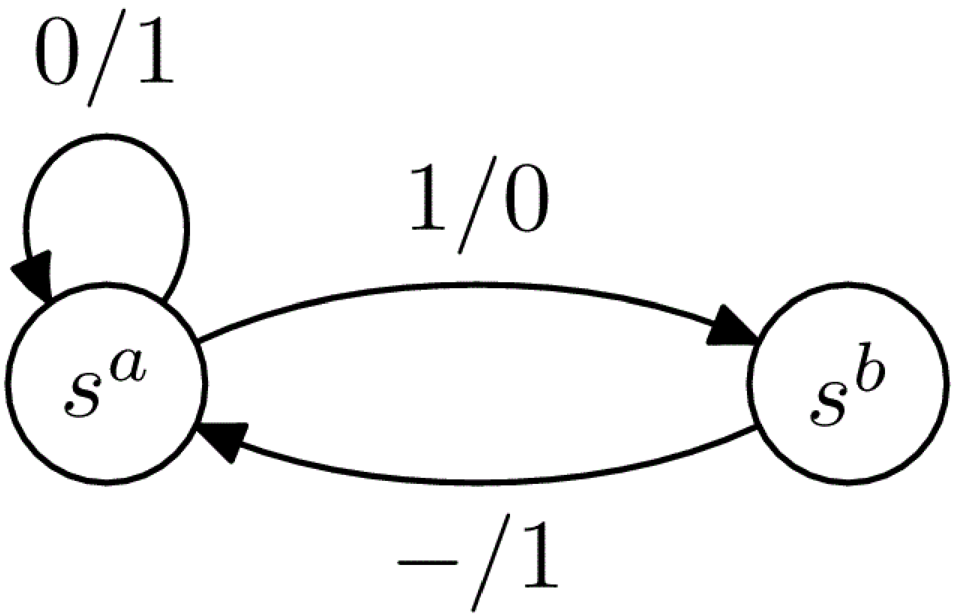
Moore automaton



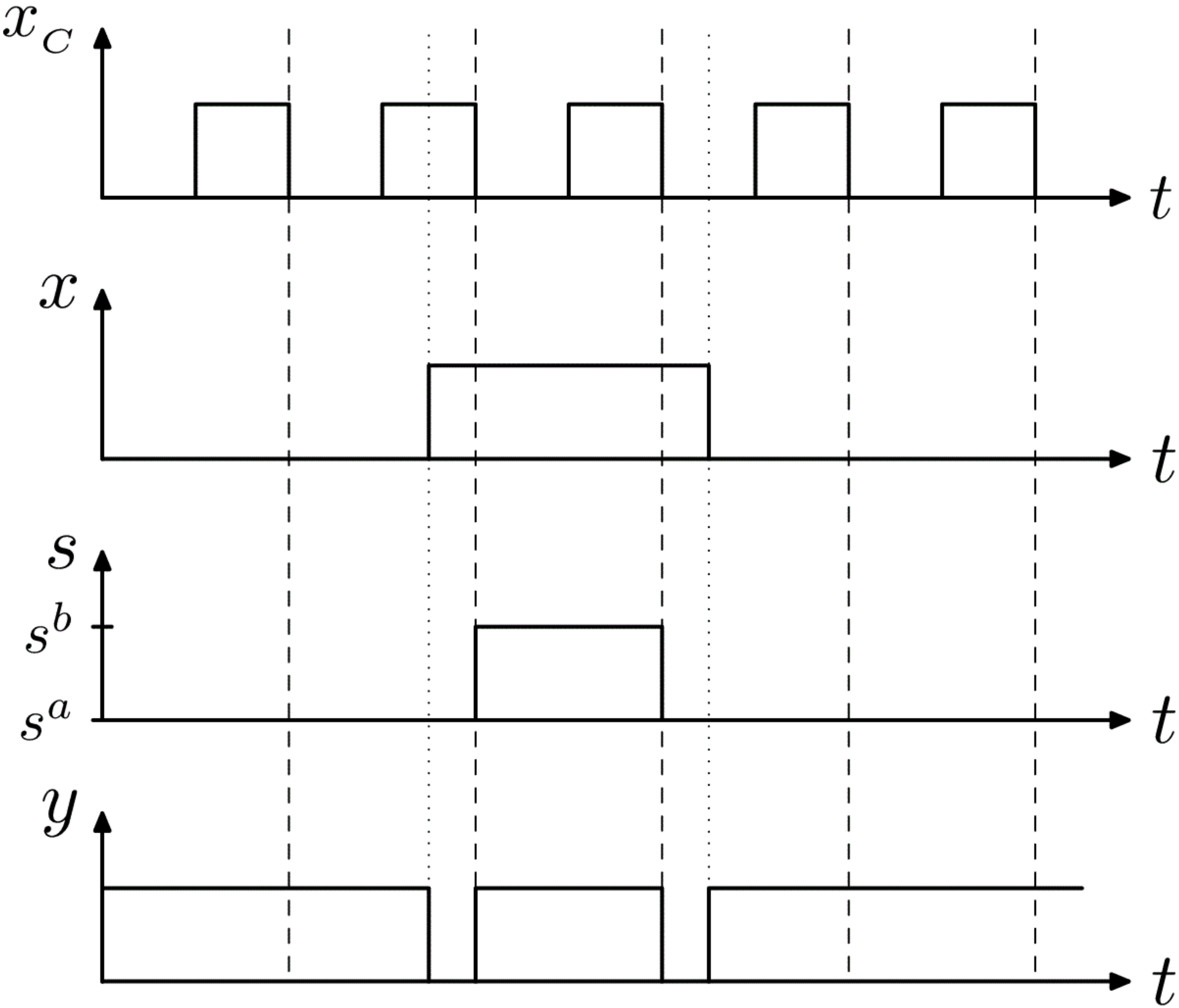
outputs change synchronously with the state changes
(on the active edge of the clock signal)


x_C
t
x
t
s
s^2
s^1
t
y
t

Mealy automaton



outputs do not change synchronously with the state changes, but change between active edges of the clock signal


x_C
t
x
t
s
s^b
s^a
t
y
t

Compatibility of states of automatons

An input x is applicable in state s if there exists $\delta(s, x)$ and $\lambda_1(s)$ (Moore) or $\lambda_2(s, x)$ (Mealy).

$\Rightarrow$ no "$-$" in state transition & output table element (s, x)

Input sequence x^{a*} is said to be applicable if each of the inputs in the sequence is applicable in its respective state.

Def.3 Two states s^i and s^j of an incompletely specified automaton are compatible, if for any applicable input sequence x^{a*} applied to each of them the same output sequence y^* will result in both cases.

Mathematical formulation for an incompletely specified Moore automaton:

$$s^i \approxeq s^j \iff \forall x^{a*} \;\; \lambda_1^*\big(\delta^*(s^i, x^{a*})\big) = \lambda_1^*\big(\delta^*(s^j, x^{a*})\big)$$

where:

$$\begin{aligned}
\delta^*(s, x^{a*}) &= s \;\; \delta(s, {}^1x) \;\; \delta\big(\delta(s, {}^1x), {}^2x\big) \ldots &&= s \; {}^1s \; {}^2s \ldots \\
\lambda_1^*(s^*) &= \lambda_1(s) \;\; \lambda_1({}^1s) \;\; \lambda_1({}^2s) \ldots &&= {}^0y \; {}^1y \; {}^2y \ldots \\
x^{a*} &= {}^1x \;\; {}^2x \;\; {}^3x \ldots
\end{aligned}$$

Mathematical formulation for an incompletely specified Mealy automaton:

$$s^i \approxeq s^j \iff \forall x^{a*} \;\; \lambda_2^{**}\Big(\delta^*(s^i, x^{a*}), x^{a*}\Big) = \lambda_2^{**}\Big(\delta^*(s^j, x^{a*}), x^{a*}\Big)$$

where:

$$\begin{array}{lcllcl}
\delta^*(s, x^{a*}) & = & s \;\; \delta(s, {}^1x) \;\; \delta\big(\delta(s, {}^1x), {}^2x\big) \ldots & = & s \;\; {}^1s \;\; {}^2s \ldots \\
\lambda_2^{**}(s^*, x^{a*}) & = & \lambda_2^*(s, x^{a*1}) \;\; \lambda_2^*({}^1s, x^{a*2}) \ldots & = & y^{*0} \;\; y^{*1} \ldots \\
y^{a*k} & = & {}^{k_1}y \ldots {}^{k_m}y & & \\
\lambda_2^*(s^k, x^{a*k}) & = & \lambda_2({}^ks, {}^{k_1}x) \ldots \lambda_2({}^ks, {}^{k_m}x) & = & {}^{k_1}y \ldots {}^{k_m}y \\
x^{a*} & = & x^{a*1} \;\; x^{a*2} \;\; x^{a*3} \ldots & & \\
x^{a*k} & = & {}^{k_1}x \ldots {}^{k_n}x & & \\
{}^kx & = & {}^{k_n}x & &
\end{array}$$

Outputs $y^i = y_1^i \dots y_r^i$ and $y^j = y_1^j \dots y_r^j$ are non-contradictory (denoted as: $y^i \doteq y^j$), if for each pair $y_d^i, y_d^j,\ d = 1, \dots, r,\ \ y_d^i = y_d^j$ or $y_d^i = -$ or $y_d^j = -$

Non-contradictory: $0 = 0,\ \ 1 = 1,\ \ 0 = -,\ \ 1 = -$

Next states produced by input x from states s^i and s^j are non-contradictory (denoted as: $\delta(s^i, x) \doteq \delta(s^j, x)$), if $\delta(s^i, x) = \delta(s^j, x)$ or $\delta(s^i, x) = -$ or $\delta(s^j, x) = -$

Def.4 Two states s^i and s^j of incompletely specified automatons are compatible (denoted as: $s^i \approxeq s^j$), if for any applicable input $x \in \boldsymbol{X}$:

- outputs in both states are non-contradictory
- next states are non-contradictory or compatible

Mathematical formulation for incompletely specified Moore automatons:

$$s^i \approxeq s^j \iff \forall x \; \Big(\lambda_1(s^i) \doteq \lambda_1(s^j)\Big) \wedge \Big(\big(\delta(s^i,x) \doteq \delta(s^j,x)\big) \vee \big(\delta(s^i,x) \approxeq \delta(s^j,x)\big)\Big)$$

Mathematical formulation for incompletely specified Mealy automatons:

$$s^i \approxeq s^j \iff \forall x \; \Big(\lambda_2(s^i,x) \doteq \lambda_2(s^j,x)\Big) \wedge \Big(\big(\delta(s^i,x) \doteq \delta(s^j,x)\big) \vee \big(\delta(s^i,x) \approxeq \delta(s^j,x)\big)\Big)$$

State transition and output table of an incompletely specified Mealy automaton

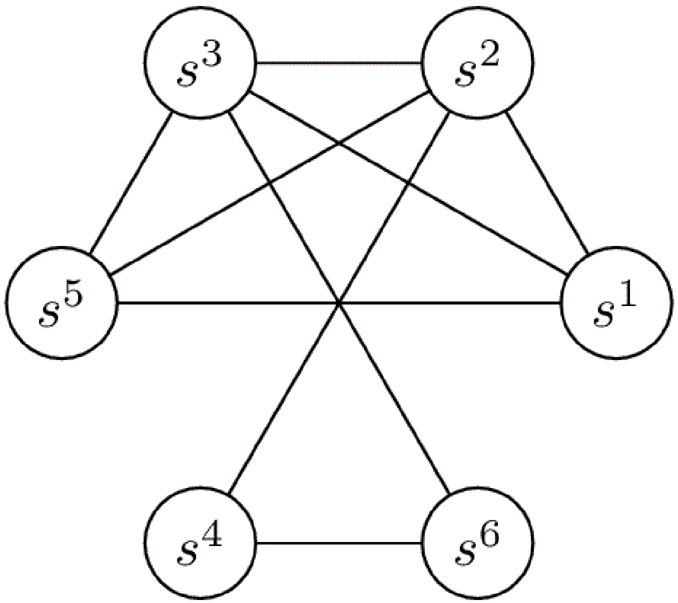
s \ x_1x_2	00	01	11	10
s^1	$s^2/1$	$s^6/1$	$s^4/0$	–
s^2	–	–	–	$s^5/1$
s^3	$s^2/1$	$s^3/1$	–	$s^3/1$
s^4	$s^5/1$	$s^3/0$	$s^1/1$	–
s^5	–	–	$s^2/0$	–
s^6	$s^3/1$	–	$s^2/1$	$s^6/1$

s'/y

Implication matrix

s^2					
s^3	s^3, s^6	s^3, s^5			
s^4	X		X		
s^5	s^2, s^4			X	
s^6	X	s^5, s^6	s^2, s^3	s^3, s^5; s^1, s^2	X
	s^1	s^2	s^3	s^4	s^5

Compatibility graph



Implication matrix

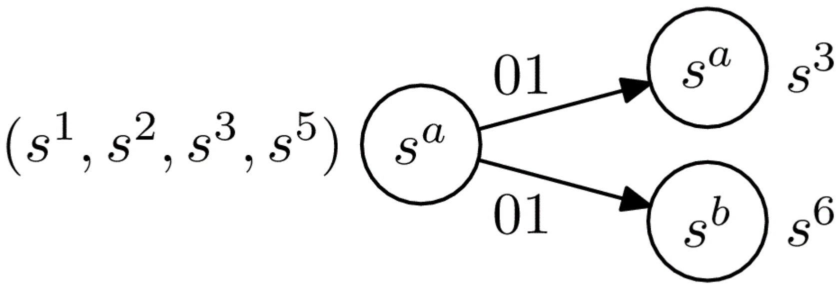
	s^1	s^2	s^3	s^4	s^5
s^2					
s^3	s^3, s^6	s^3, s^5			
s^4	X		X		
s^5	s^2, s^4			X	
s^6	X	$\cancel{s^5, s^6}$	s^2, s^3	s^3, s^5 s^1, s^2	X

State substitution

$$\{s^1, s^2, s^3, s^5\} \rightarrow s^a$$

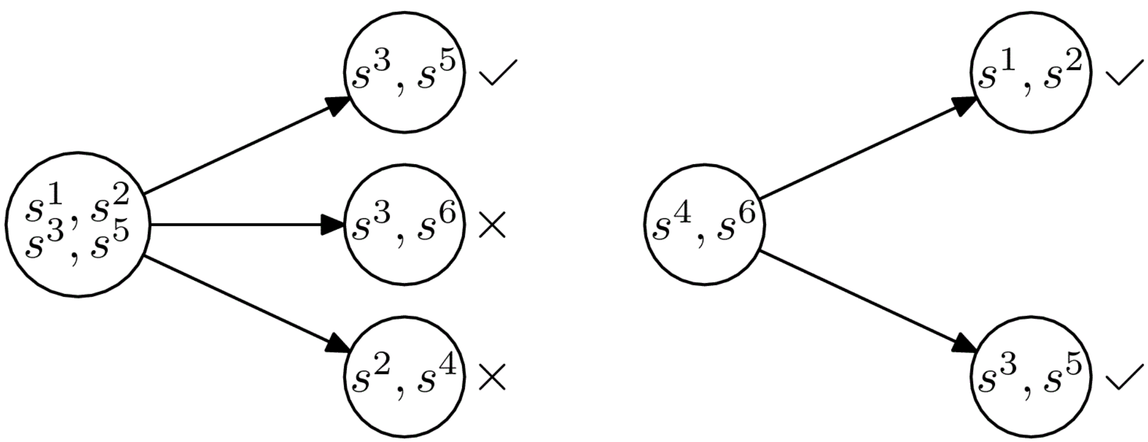
$$\{s^4, s^6\} \rightarrow s^b$$

Problem with state substitution



$s \backslash x_1x_2$	00	01	11	10
s^a	$s^a/1$	$?/1$		
s^b	$s^a/1$	$s^a/0$	$s^a/1$	$s^b/1$

s'/y



Conclusion

States of the minimized incompletely specified automaton must fulfill the following conditions:

- include all the states of the initial automaton (cover condition)
- for all the states of the initial automaton forming a single state of a minimized automaton and for each input all the next states of the initial automaton must be covered by a single state of the minimized automaton (closure condition)

States of the minimized completely specified automaton must fulfill the following condition:

- include all the states of the initial automaton (cover condition)

Minimisation of the number of states of a completely specified automaton – mathematical formulation

${}_{A_1}\boldsymbol{S}$ – set of states of the initial automaton A_1

${}_{A_1}\boldsymbol{S}^k$ – kth set of equivalent states of automaton A_1,

${}_{A_1}\boldsymbol{S}^k \subset {}_{A_1}\boldsymbol{S}$

${}_{A_1}\boldsymbol{S}^{k'}$ – k'th set of equivalent states of automaton A_1,

${}_{A_1}\boldsymbol{S}^{k'} \subset {}_{A_1}\boldsymbol{S}$

${}_{A_2}\boldsymbol{S}$ – set of states of minimal automaton A_2

$\mathbf{2}^{{}_{A_1}\boldsymbol{S}}$ – set (family) of all the subsets of states of automaton A_1

${}_{A_1}\hat{\boldsymbol{S}}$ – family of subsets of states of automaton A_1,

${}_{A_1}\hat{\boldsymbol{S}} \subset \mathbf{2}^{{}_{A_1}\boldsymbol{S}}$

n – minimal number of equivalent state subsets of automaton A_1.

Moreover $k, k' \in \{1, \ldots, n\}$.

We are looking for a minimal family ${}_{A_1}\hat{S}$ of subsets of the set of states of automaton A_1, i.e.:

$$n = \min_{card({}_{A_1}\hat{\boldsymbol{S}})} \left({}_{A_1}\hat{\boldsymbol{S}} \subset \boldsymbol{2}^{{}_{A_1}\boldsymbol{S}}\right)$$

of which the sets ${}_{A_1}\boldsymbol{S}^k, {}_{A_1}\boldsymbol{S}^{k'} \in {}_{A_1}\hat{\boldsymbol{S}}$ fulfil the conditions:

$$\left({}_{A_1}\boldsymbol{S}^k \subset {}_{A_1}\boldsymbol{S}\right) \ : \ \left(\forall\left({}_{A_1}s^{k_i}, {}_{A_1}s^{k_j} \in {}_{A_1}\boldsymbol{S}^k\right) \ \left({}_{A_1}s^{k_i} \equiv {}_{A_1}s^{k_j}\right)\right)$$

$$(k \neq k') \ \Rightarrow \ \left({}_{A_1}\boldsymbol{S}^k \bigcap {}_{A_1}\boldsymbol{S}^{k'} = \emptyset\right)$$

$$\left(\bigcup_{k=1}^{n} {}_{A_1}\boldsymbol{S}^k\right) =_{A_1} \boldsymbol{S}$$

Then with each of those subsets a state of the minimal automaton A_2 can be associated in the following manner:

$${}_{A_1}\boldsymbol{S}^k \rightarrow {}_{A_2}s^k, \quad {}_{A_2}s^k \in {}_{A_2}\boldsymbol{S}$$

In short, we are looking for a least numerous family of subsets of the set of states of automaton A_1. The elements of this family must fulfil the following conditions:

1. This are the sets of equivalent states.
2. None of the states of the initial automaton belongs to two different sets.
3. Those sets fulfil the cover condition.

Minimisation of the number of states of an incompletely specified automaton – mathematical formulation

$_{A_1}\boldsymbol{S}$ – set of states of the initial automaton A_1

$_{A_1}\boldsymbol{S}^k$ – kth set of compatible states of automaton A_1,
$_{A_1}\boldsymbol{S}^k \subset {_{A_1}\boldsymbol{S}}$

$_{A_1}\boldsymbol{S}^{k'}$ – k'th set of compatible states of automaton A_1,
$_{A_1}\boldsymbol{S}^{k'} \subset {_{A_1}\boldsymbol{S}}$

$_{A_2}\boldsymbol{S}$ – set of states of the minimal automaton A_2

$2^{_{A_1}\boldsymbol{S}}$ – set (family) of all the subsets of states of automaton A_1

$_{A_1}\hat{\boldsymbol{S}}$ – family of subsets of states of automaton A_1,
$_{A_1}\hat{\boldsymbol{S}} \subset 2^{_{A_1}\boldsymbol{S}}$

n – minimal number of compatible state subsets of automaton A_1.

Moreover $k, k' \in \{1, \ldots, n\}$.

We are looking for a minimal family ${}_{A_1}\hat{\hat{S}}$ of subsets of the set of states of automaton A_1, i.e.:

$$n = \min_{card({}_{A_1}\hat{\boldsymbol{S}})} \left({}_{A_1}\hat{\boldsymbol{S}} \subset \boldsymbol{2}^{{}_{A_1}\boldsymbol{S}} \right)$$

of which the sets ${}_{A_1}\boldsymbol{S}^k, {}_{A_1}\boldsymbol{S}^{k'} \in {}_{A_1}\hat{\boldsymbol{S}}$ fulfil the following conditions:

$$\left({}_{A_1}\boldsymbol{S}^k \subset {}_{A_1}\boldsymbol{S} \right) \; : \; \left(\forall \left({}_{A_1}s^{k_i}, {}_{A_1}s^{k_j} \in {}_{A_1}\boldsymbol{S}^k \right) \; \left({}_{A_1}s^{k_i} \approxeq {}_{A_1}s^{k_j} \right) \right)$$

$$(k \neq k') \;\Rightarrow\; \left({}_{A_1}\boldsymbol{S}^k \bigcap {}_{A_1}\boldsymbol{S}^{k'} = \emptyset \right)$$

$$\left(\bigcup_{k=1}^{n} {}_{A_1}\boldsymbol{S}^k \right) = {}_{A_1}\boldsymbol{S}$$

$$\left(\forall \left({}_{A_1}s^{k_i}, {}_{A_1}s^{k_j}, x \right) : \left({}_{A_1}s^{k_i}, {}_{A_1}s^{k_j} \in {}_{A_1}\boldsymbol{S}^k \right) \right)$$

$$\Rightarrow \left(\exists_{A_1}\boldsymbol{S}^{k'} : \left(\delta({}_{A_1}s^{k_i}, x), \delta({}_{A_1}s^{k_j}, x) \in {}_{A_1}\boldsymbol{S}^{k'} \right) \right)$$

Then with each of those subsets a state of the minimal automaton A_2 can be associated in the following manner:

$${}_{A_1}\boldsymbol{S}^k \rightarrow {}_{A_2}s^k, \qquad {}_{A_2}s^k \in {}_{A_2}\boldsymbol{S}$$

In short, we are looking for a least numerous family of subsets of the set of states of automaton A_1. The elements of this family must fulfil the following conditions:

1. This are the sets of equivalent states.
2. None of the states of the initial automaton belongs to two different sets.
3. Those sets must fulfil the cover condition.
4. Those sets must fulfil the closure condition.

$s \backslash x_1x_2$	00	01	11	10
s^1	$s^2/1$	$s^6/1$	$s^4/0$	$-$
s^2	$-$	$-$	$-$	$s^5/1$
s^3	$s^2/1$	$s^3/1$	$-$	$s^3/1$
s^4	$s^5/1$	$s^3/0$	$s^1/1$	$-$
s^5	$-$	$-$	$s^2/0$	$-$
s^6	$s^3/1$	$-$	$s^2/1$	$s^6/1$

s'/y

$$\{s^1, s^2\} \to s^\alpha$$
$$\{s^3, s^5\} \to s^\beta$$
$$\{s^4, s^6\} \to s^\gamma$$

$s \backslash x_1x_2$	00	01	11	10
s^α	$s^\alpha/1$	$s^\gamma/1$	$s^\gamma/0$	$s^\beta/1$
s^β	$s^\alpha/1$	$s^\beta/1$	$s^\alpha/0$	$s^\beta/1$
s^γ	$s^\beta/1$	$s^\beta/0$	$s^\alpha/1$	$s^\gamma/1$

s'/y

s	$\to$	q_1	q_2
s^α	$\to$	0	0
s^β	$\to$	0	1
s^γ	$\to$	1	1

q_1q_2 \ x_1x_2	00	01	11	10
00	00/1	11/1	11/0	01/1
01	00/1	01/1	00/0	01/1
11	01/1	01/0	00/1	11/1
10	–	–	–	–

$q_1'q_2'/y$

q_1q_2 \ x_1x_2	00	01	11	10
00	1	1	0	1
01	1	1	0	1
11	1	0	1	1
10	–	–	–	–

y

q_1q_2 \ x_1x_2	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	0	0	0	1
10	–	–	–	–

q_1'

q_1q_2 \ x_1x_2	00	01	11	10
00	0	1	1	1
01	0	1	0	1
11	1	1	0	1
10	–	–	–	–

q_2'

D type flip-flop realisation

$q_1q_2 \backslash x_1x_2$	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	0	0	0	1
10	–	–	–	–

$D_1 = q_1'$

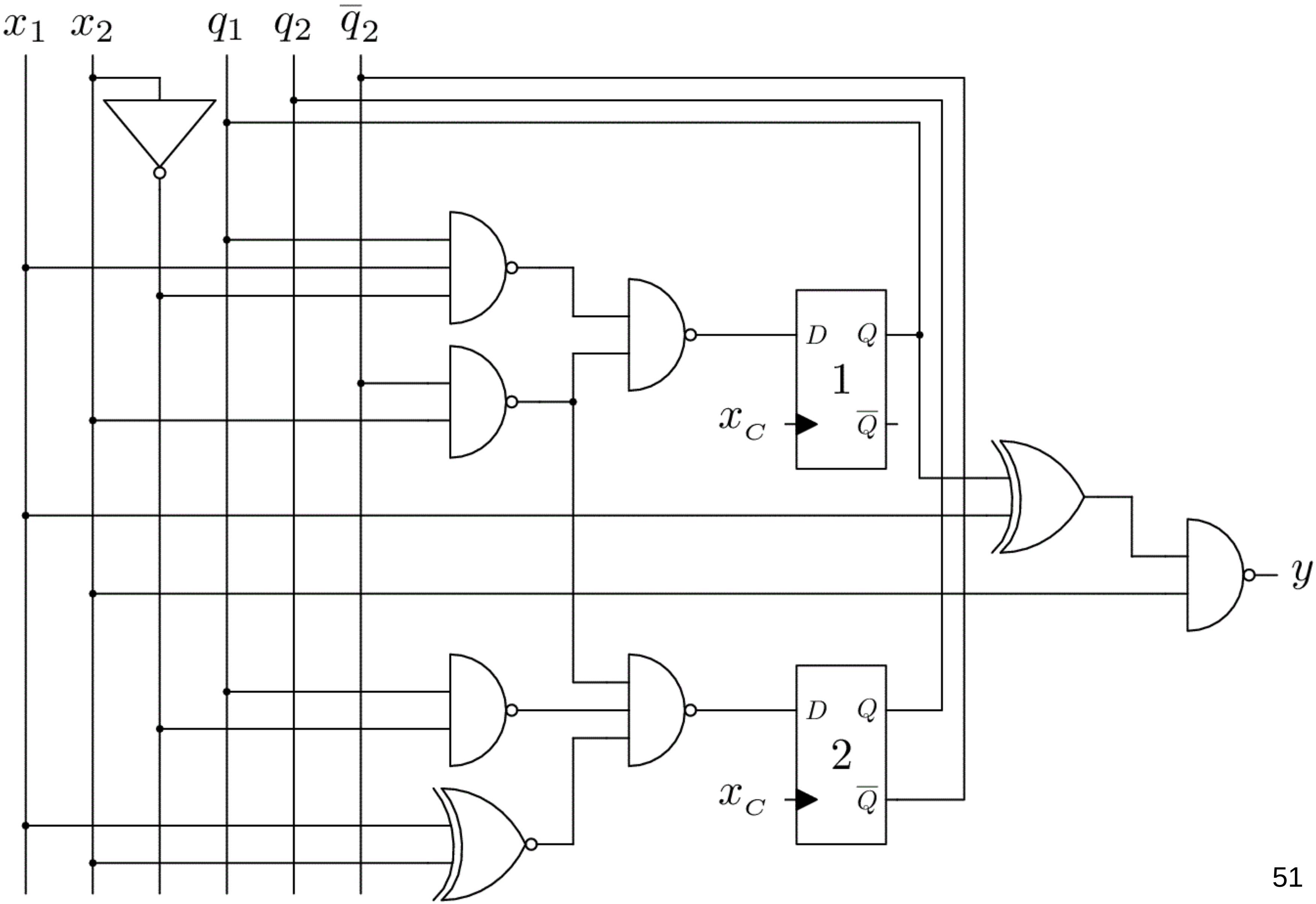
$q_1q_2 \backslash x_1x_2$	00	01	11	10
00	0	1	1	1
01	0	1	0	1
11	1	1	0	1
10	–	–	–	–

$D_2 = q_2'$

$q_1q_2 \backslash x_1x_2$	00	01	11	10
00	1	1	0	1
01	1	1	0	1
11	1	0	1	1
10	–	–	–	–

y

$$\begin{cases} D_1 = q_1x_1\overline{x}_2 + \overline{q}_2x_2 \\ D_2 = \overline{x}_1x_2 + x_1\overline{x}_2 + q_1\overline{x}_2 + \overline{q}_2x_2 = x_1 \oplus x_2 + q_1\overline{x}_2 + \overline{q}_2x_2 \\ y = \overline{x}_2 + \overline{q}_1\overline{x}_1 + q_1x_1 = \overline{x}_2 + \overline{q_1 \oplus x_1} \\ \quad = \overline{x_2(q_1 \oplus x_1)} \end{cases}$$


x_1
x_2
q_1
q_2
$\overline{q}_2$
D
Q
1
x_C
$\overline{Q}$
D
Q
2
x_C
$\overline{Q}$
y

JK-type flip-flop

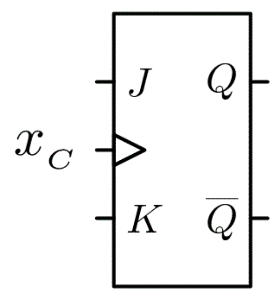
$Q \backslash JK$	00	01	11	10
0	0	0	1	1
1	1	0	0	1

Q'

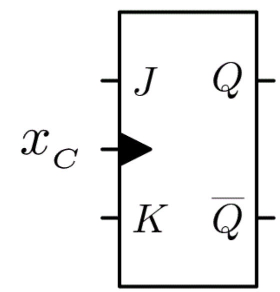
$Q \backslash JK$	00	01	11	10
0	0	0	1	1
1	1	0	0	1

Q'

Q	$\rightarrow$	Q'	J	K
0	$\rightarrow$	0	0	–
0	$\rightarrow$	1	1	–
1	$\rightarrow$	0	–	1
1	$\rightarrow$	1	–	0



positive edge triggered



negative edge triggered

Coded state transition tables for the JK-type flip-flop

q'_1

q_1q_2 \ x_1x_2	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	0	0	0	1
10	—	—	—	—

K_1

q_1q_2 \ x_1x_2	00	01	11	10
00	—	—	—	—
01	—	—	—	—
11	1	1	1	0
10	—	—	—	—

J_1

q_1q_2 \ x_1x_2	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	—	—	—	—
10	—	—	—	—

q'_2

q_1q_2 \ x_1x_2	00	01	11	10
00	0	1	1	1
01	0	1	0	1
11	1	1	0	1
10	—	—	—	—

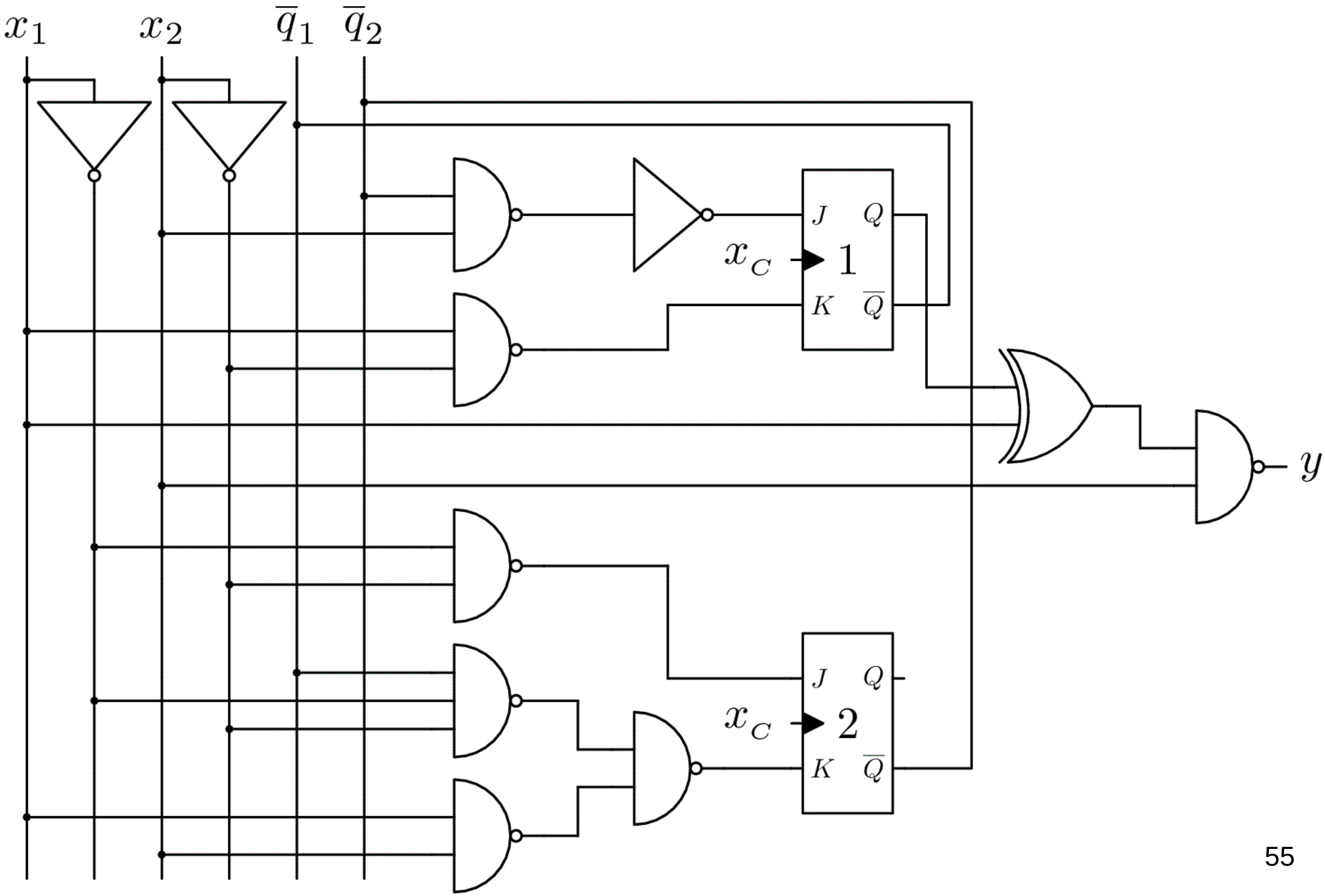
K_2

q_1q_2 \ x_1x_2	00	01	11	10
00	—	—	—	—
01	1	0	1	0
11	0	0	1	0
10	—	—	—	—

J_2

q_1q_2 \ x_1x_2	00	01	11	10
00	0	1	1	1
01	—	—	—	—
11	—	—	—	—
10	—	—	—	—

$$\begin{cases} J_1 &= \overline{q}_2 x_2 \\ K_1 &= \overline{x}_1 + x_2 = \overline{x_1 \overline{x}_2} \\ J_2 &= x_1 + x_2 = \overline{\overline{x}_1 \overline{x}_2} \\ K_2 &= x_1 x_2 + \overline{q}_1 \overline{x}_1 x_2 \end{cases}$$


x_1
x_2
$\overline{q}_1$
$\overline{q}_2$
J
Q
x_C
1
K
$\overline{Q}$
y
J
Q
x_C
2
K
$\overline{Q}$

Transformation of a Moore automaton into a partially equivalent Mealy automaton

Moore automaton state transition and output table

$s \backslash x$	0	1	y
s^1	s^1	s^2	0
s^2	s^2	s^1	1

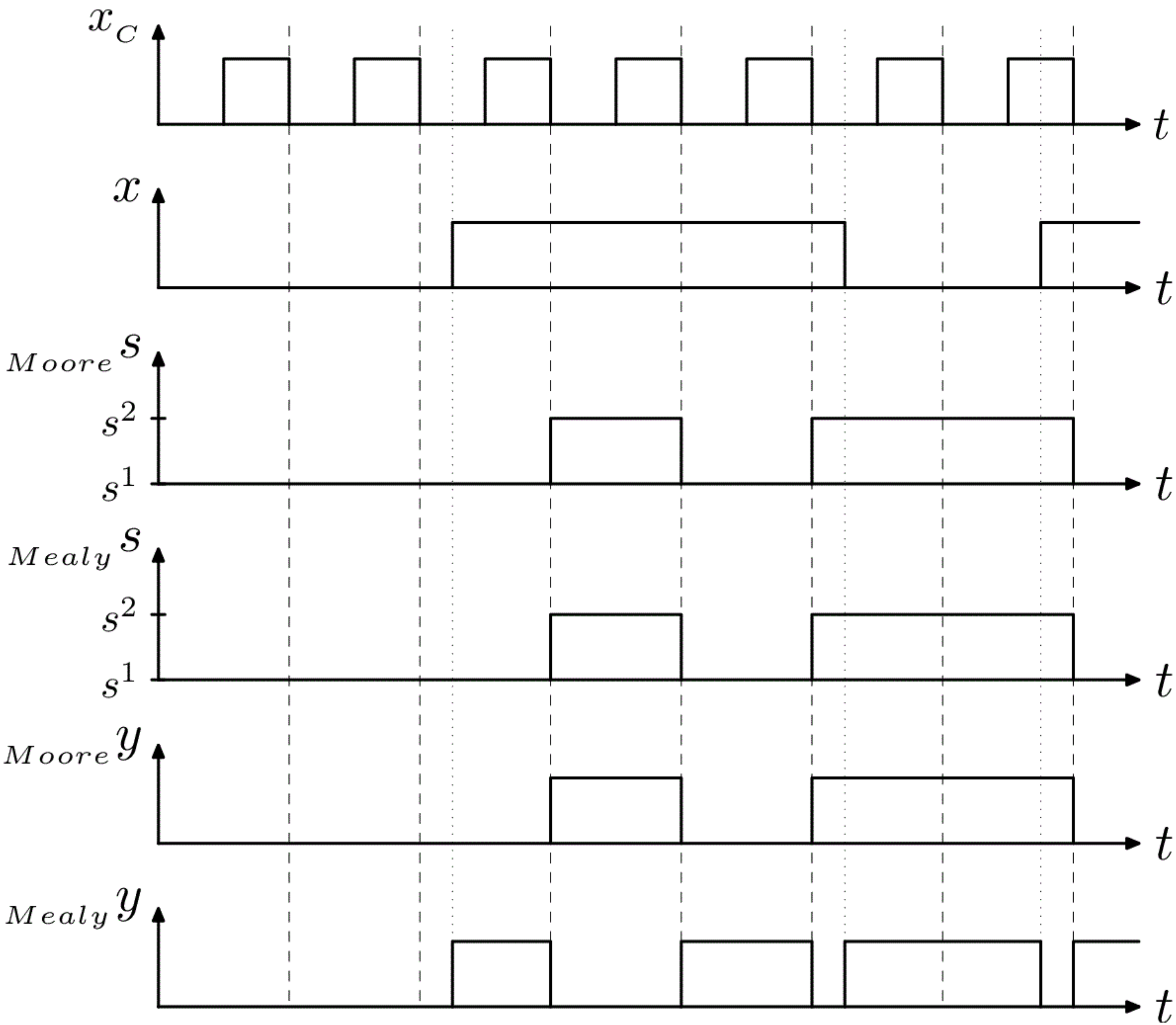
s'/y

$\Rightarrow$

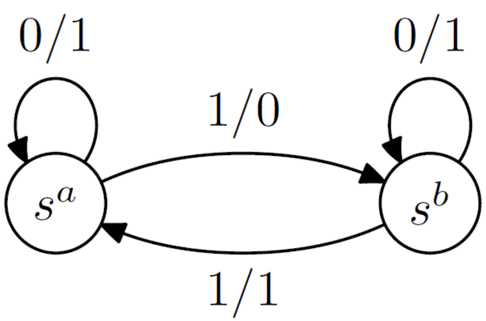
Mealy automaton state transition and output table

$s \backslash x$	0	1
s^1	$s^1/0$	$s^2/1$
s^2	$s^2/1$	$s^1/0$

s'/y


x_C
t
x
t
$_{Moore}s$
s^2
s^1
t
$_{Mealy}s$
s^2
s^1
t
$_{Moore}y$
t
$_{Mealy}y$
t

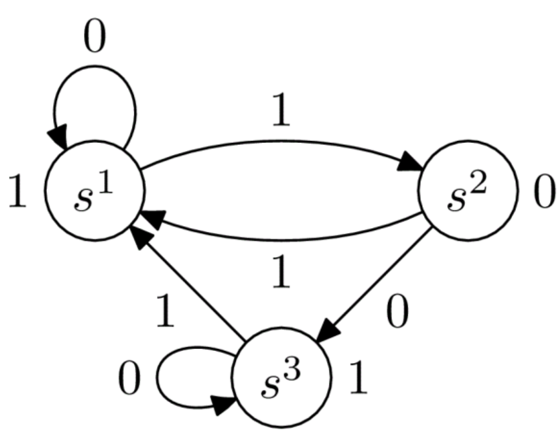
Transformation of a Mealy automaton into a partially equivalent Moore automaton



$s \backslash x$	0	1
s^a	$s^a/1$	$s^b/0$
s^b	$s^b/1$	$s^a/1$

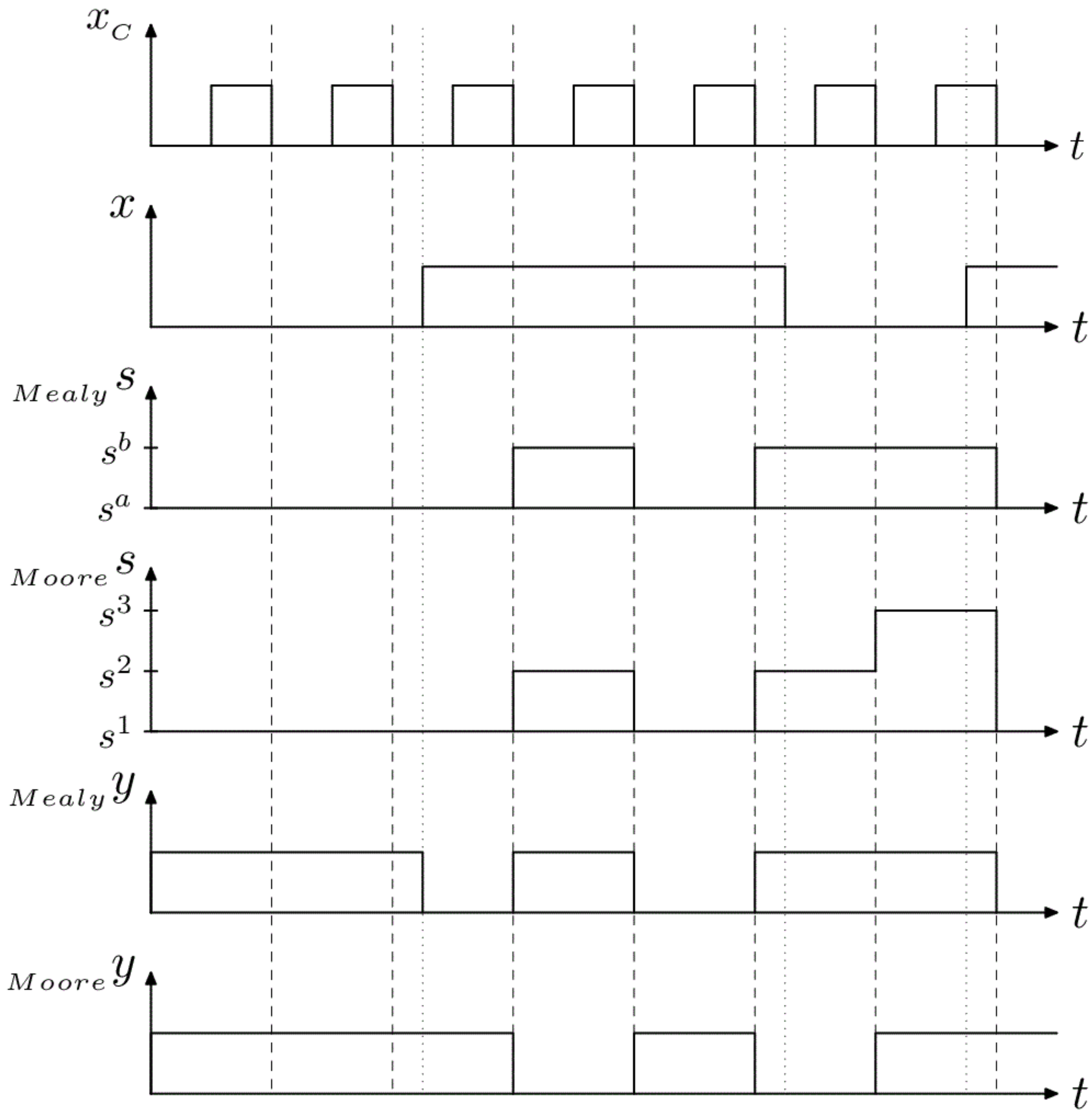
s'/y

$$s^a/1 \rightarrow s^1$$
$$s^b/0 \rightarrow s^2$$
$$s^b/1 \rightarrow s^3$$

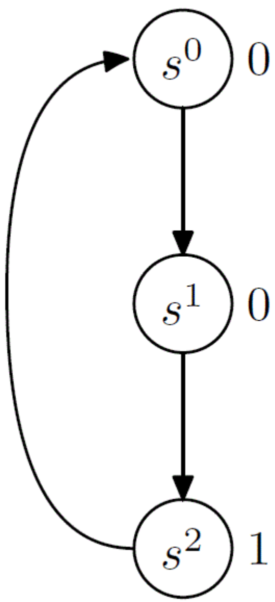


$s \backslash x$	0	1	y
s^1	s^1	s^2	1
s^2	s^3	s^1	0
s^3	s^3	s^1	1

s'/y


x_C
t
x
t
$_{Mealy}s$
s^b
s^a
t
$_{Moore}s$
s^3
s^2
s^1
t
$_{Mealy}y$
t
$_{Moore}y$
t

Frequency divider by 3 (inputless circuit)



s	s'	y
s^0	s^1	0
s^1	s^2	0
s^2	s^0	1

s	$\rightarrow$	q_1	q_0
s^0	$\rightarrow$	0	0
s^1	$\rightarrow$	0	1
s^2	$\rightarrow$	1	0

$q_1 q_0$	$q_1' q_0'$	y
00	01	0
01	10	0
11	–	–
10	00	1

q_1':

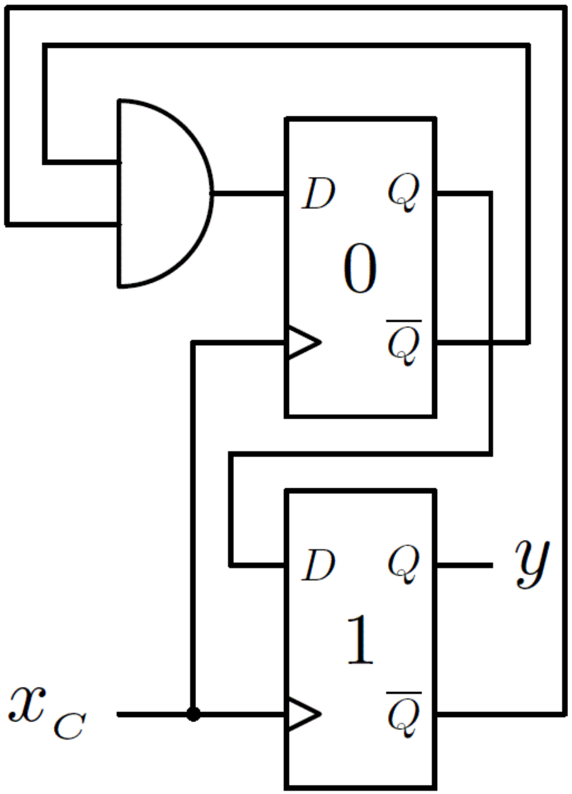
$q_1 \backslash q_0$	0	1
0	0	1
1	0	–

q_0':

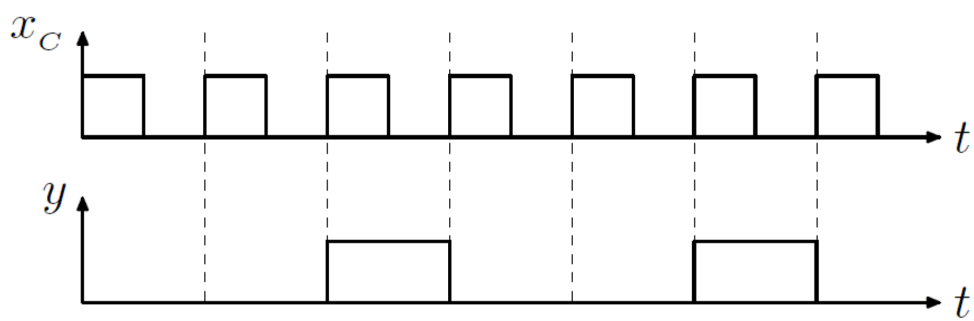
$q_1 \backslash q_0$	0	1
0	1	0
1	0	–

y:

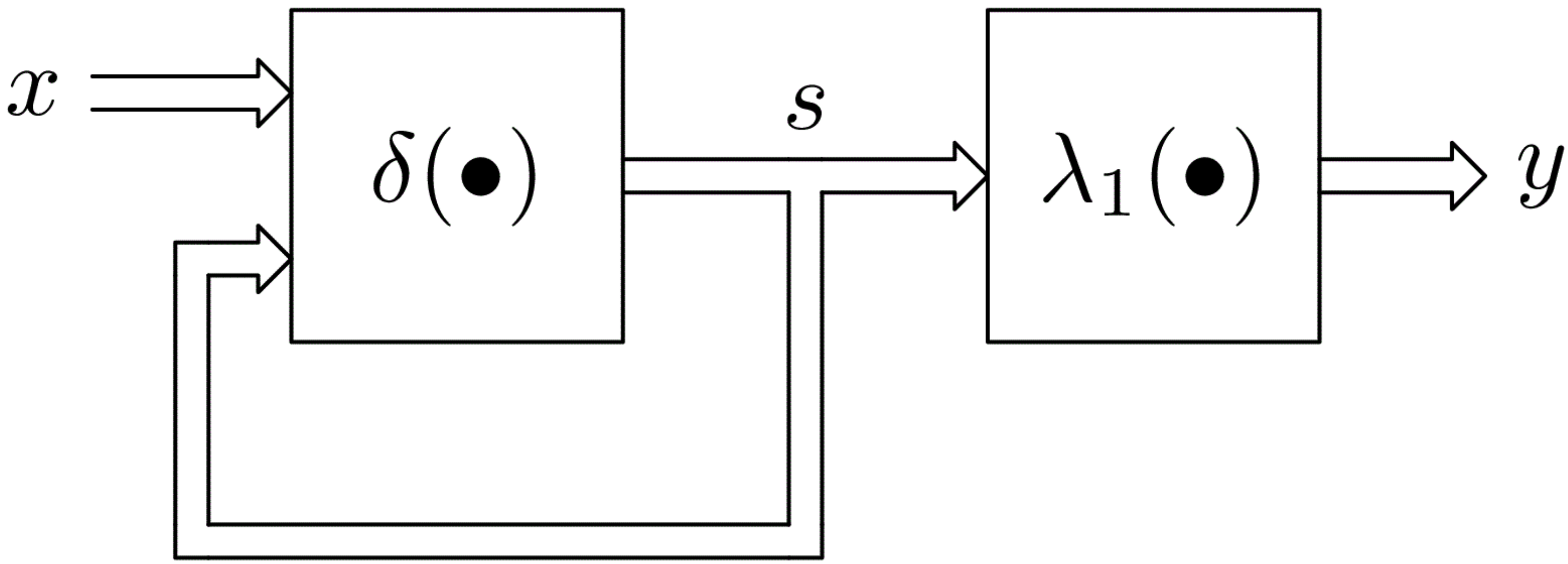
$q_1 \backslash q_0$	0	1
0	0	0
1	1	–



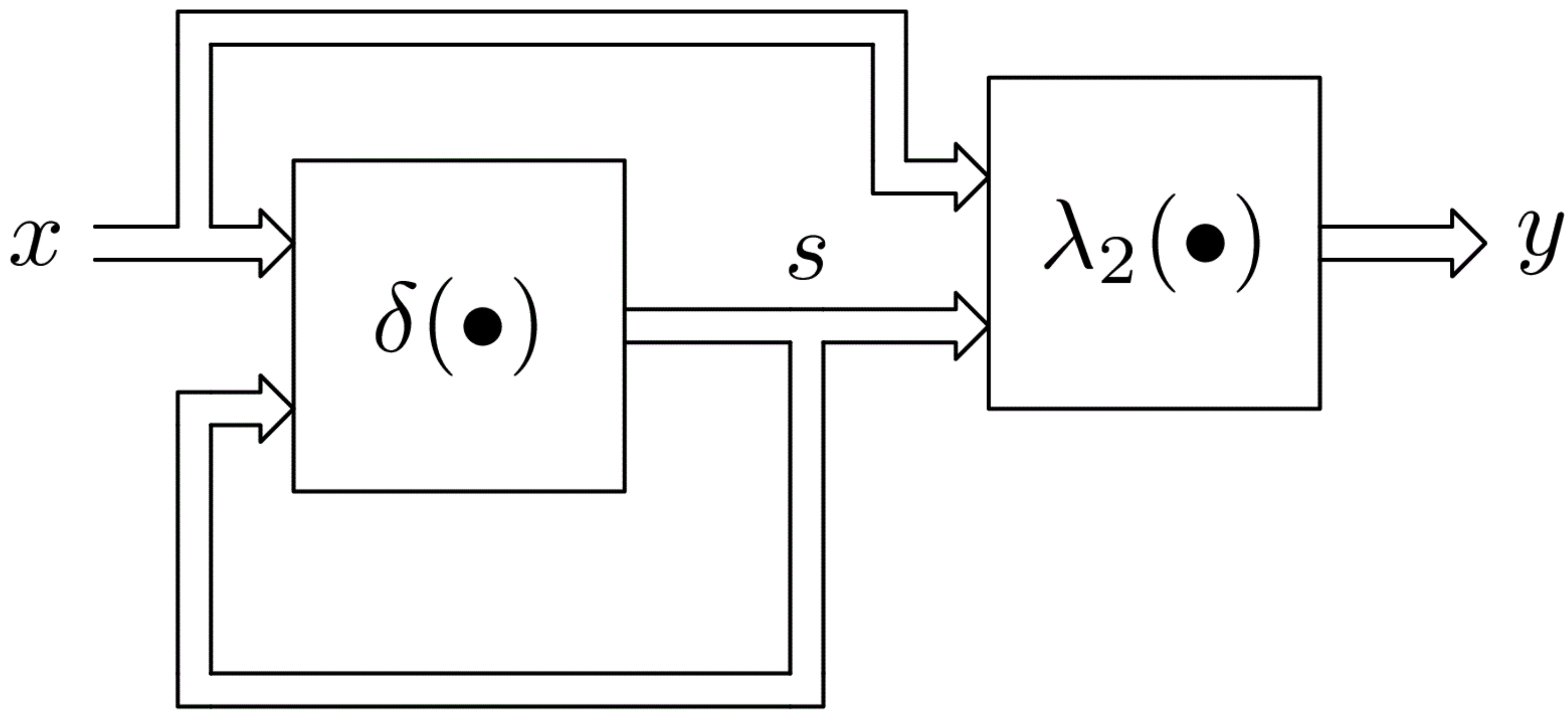
$$\begin{cases} q_0' = \overline{q}_1\overline{q}_0 \\ q_1' = q_0 \\ y = q_1 \end{cases}$$



Structure of asynchronous Moore automaton



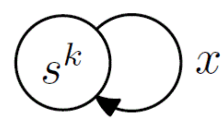
Structure of asynchronous Mealy automaton



Definitions

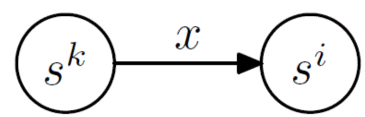
Stable state under the influence of input x

$$\delta(s^k, x) = s^k$$



Unstable state under the influence of input x

$$\delta(s^k, x) = s^j, \quad j \neq k$$

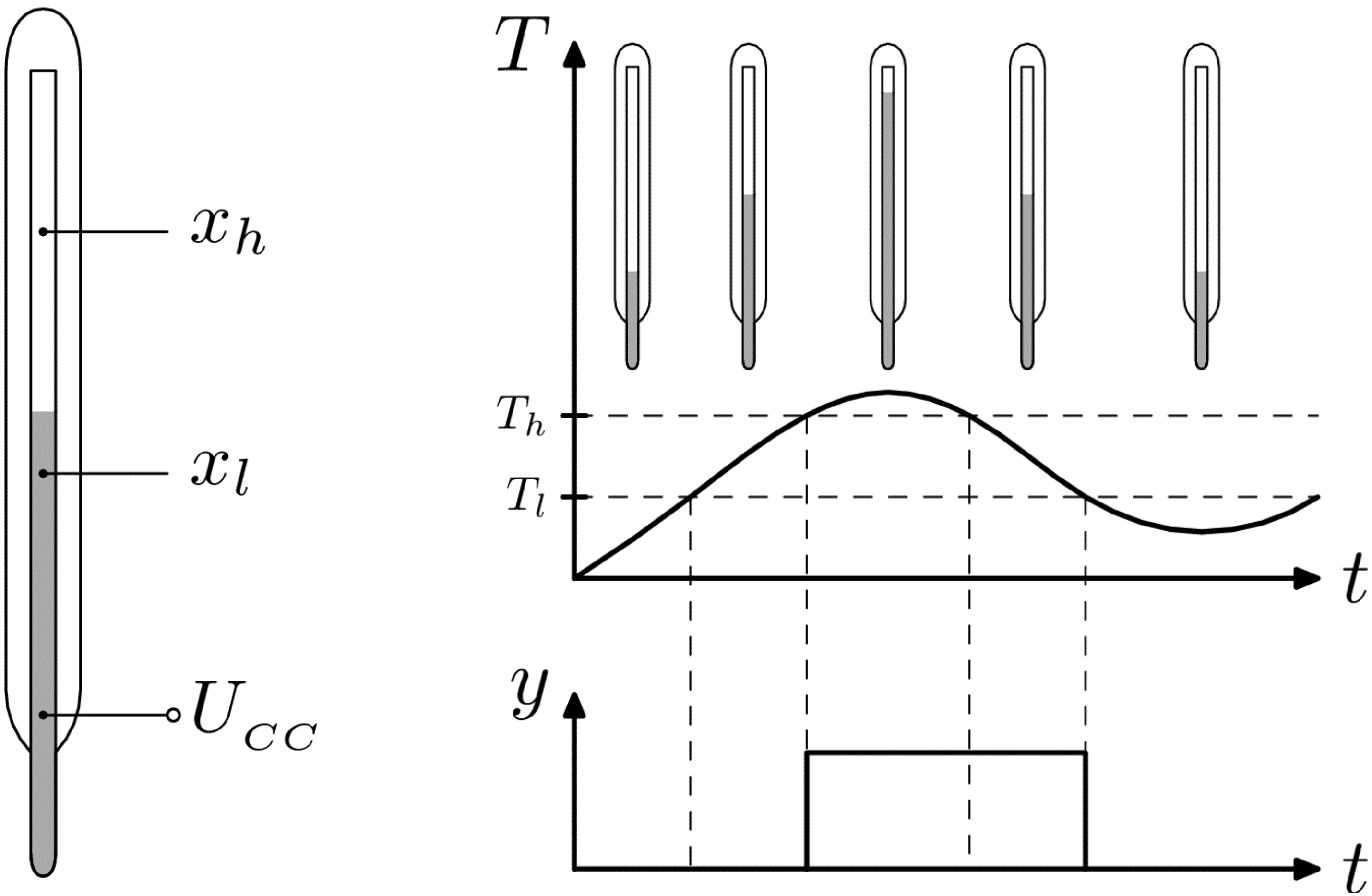


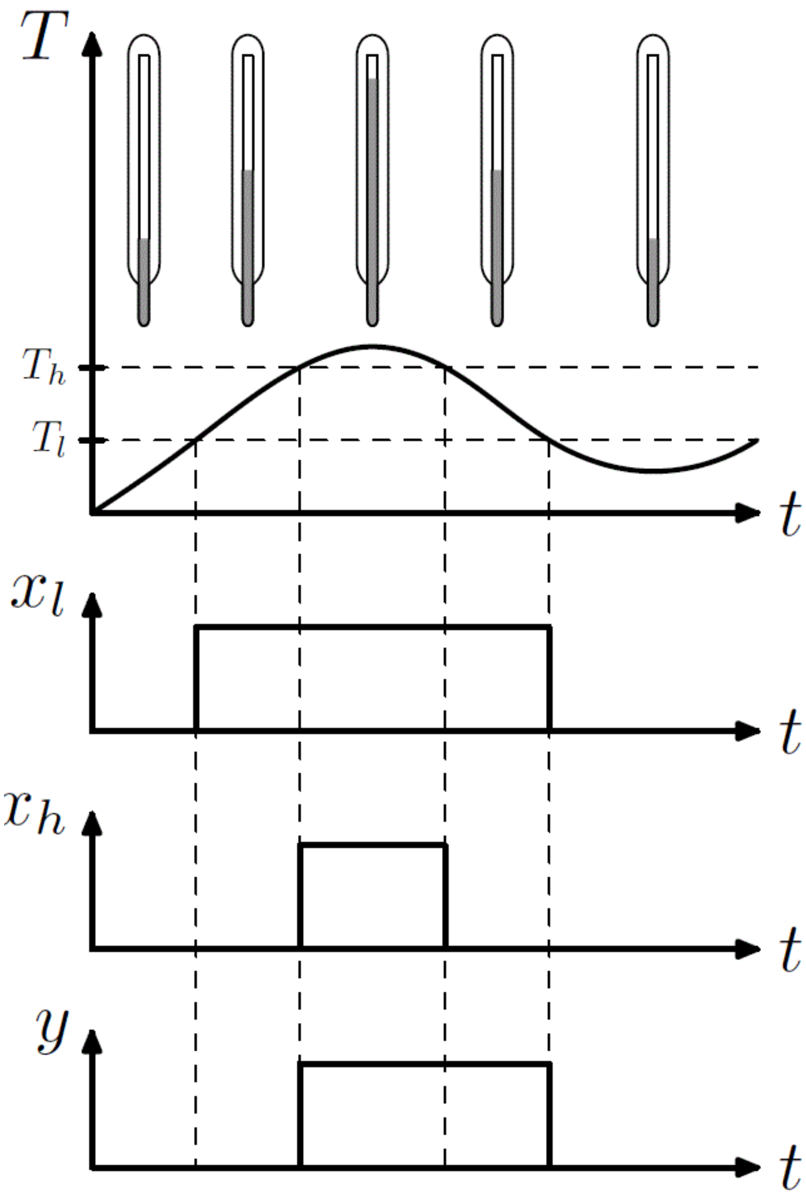
⋆ Each of the above transitions takes τ

Limitations imposed on the design of asynchronous automatons:

- when an input x changes only one signal x_j changes $x = [x_1, \ldots, x_j, \ldots, x_n]$
- the next input change may take place only when the time (τ), necessary for the internal state to become stable, has elapsed

Non-overreacting aircon controller





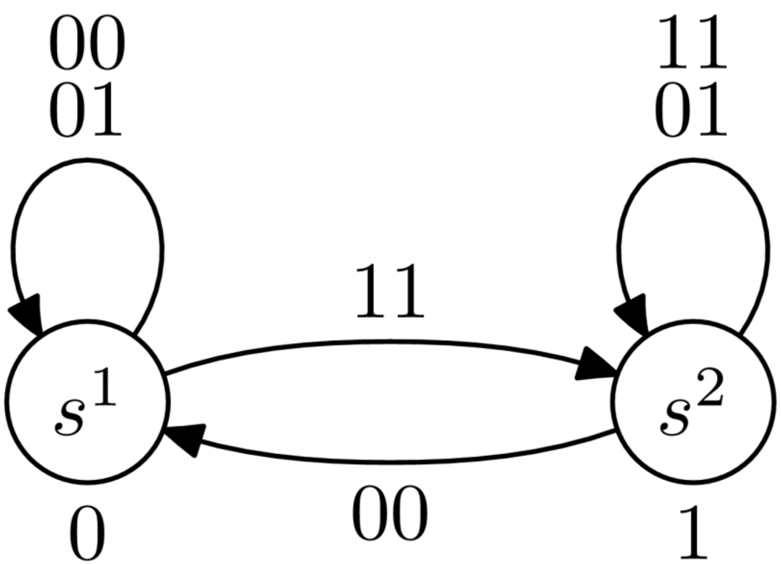
T	x_h	x_l
$T < T_l$	0	0
$T_l \leqslant T < T_h$	0	1
$T \geqslant T_h$	1	1
$-$	1	0

$x_h x_l = 01 \quad \Rightarrow \quad y = 0$ or $y = 1$

$\Rightarrow$ this is not a combinational problem

$\Rightarrow$ memory is needed

Graph of the aircon controller



Its state transition and output table

$s \backslash x_h x_l$	00	01	11	10	y
s^1	s^1	s^1	s^2	—	0
s^2	s^1	s^2	s^2	—	1

s'/y

Coding

s		q	y
s_1	$\rightarrow$	0	0
s_2	$\rightarrow$	1	1

$$\Rightarrow y = q$$

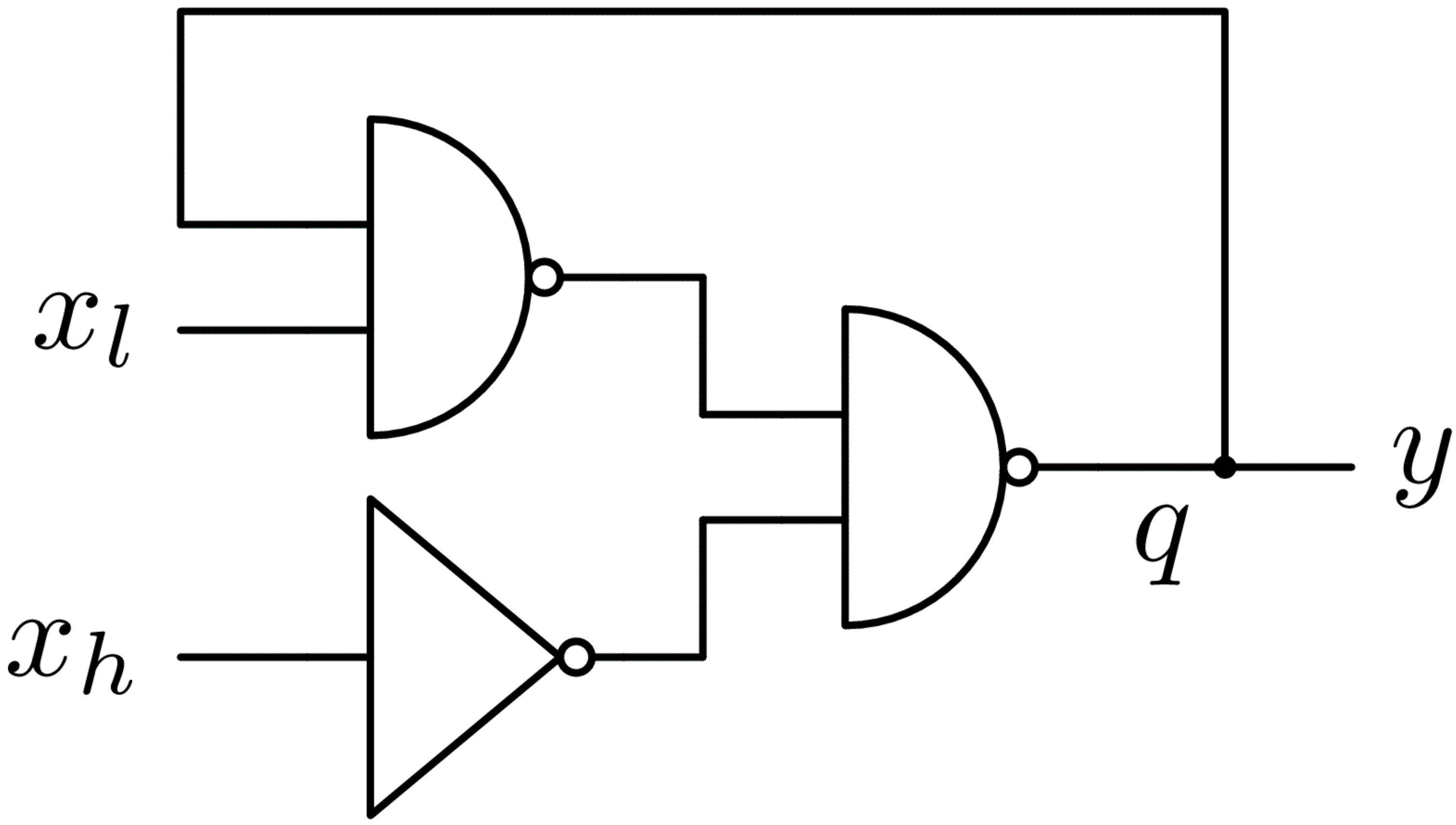
$$q' = x_h + q x_l$$

Coded state transition table

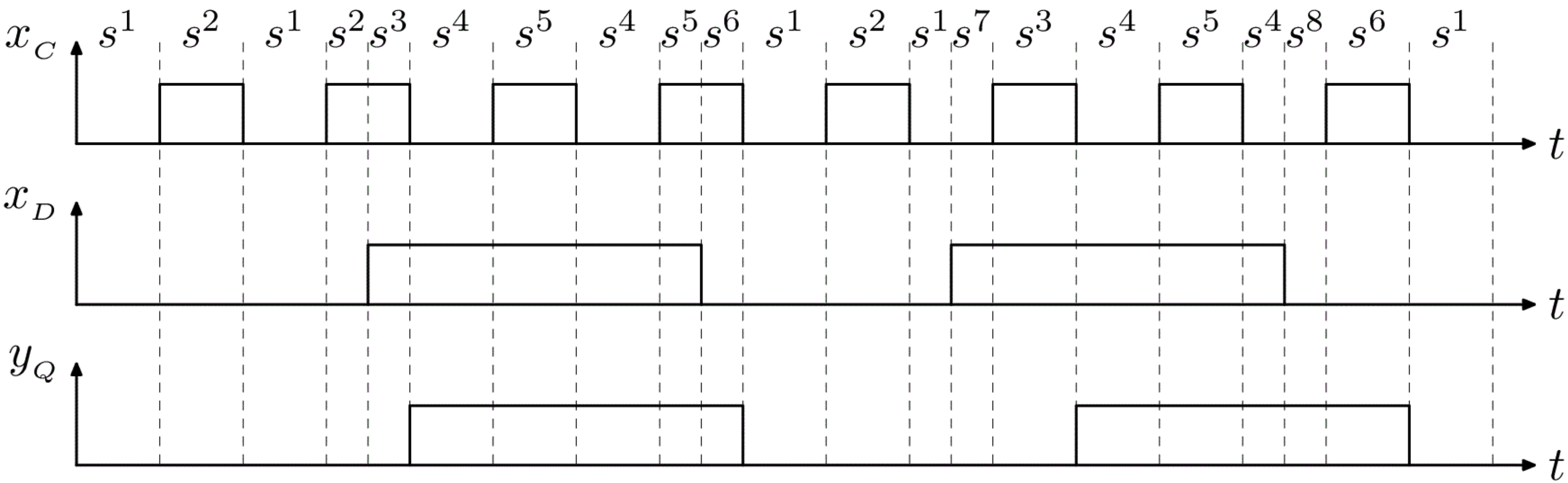
$q \backslash x_h x_l$	00	01	11	10
0	0	0	1	$-$
1	0	1	1	$-$

q'

Circuit diagram

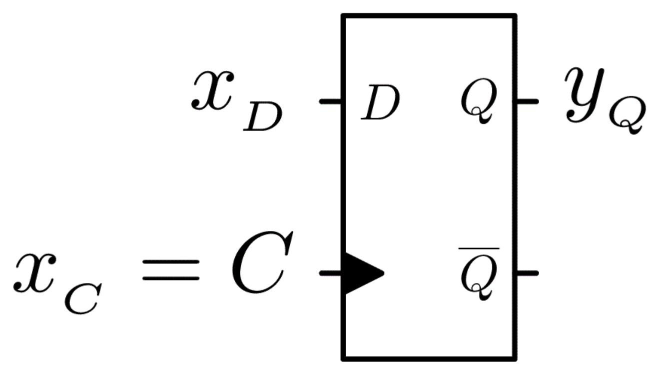


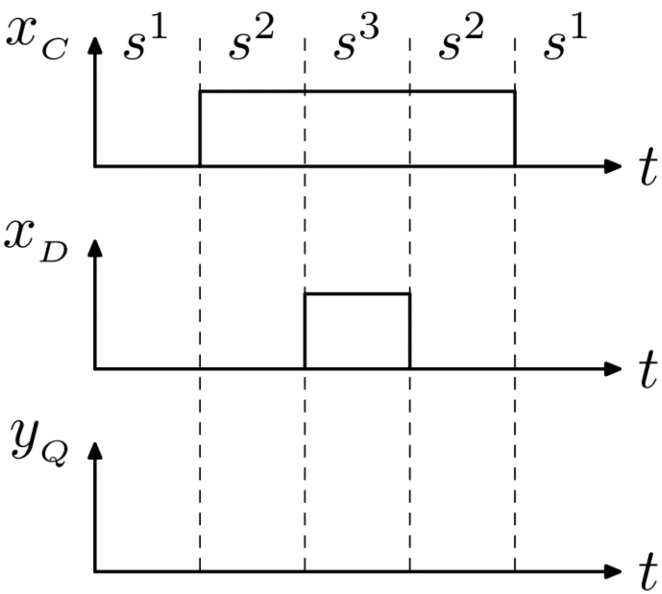
Negative-edge triggered D-type flip-flop

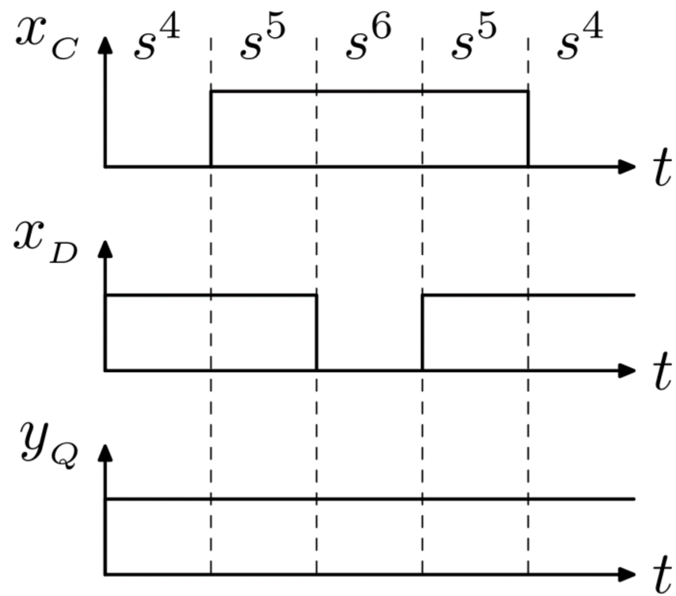


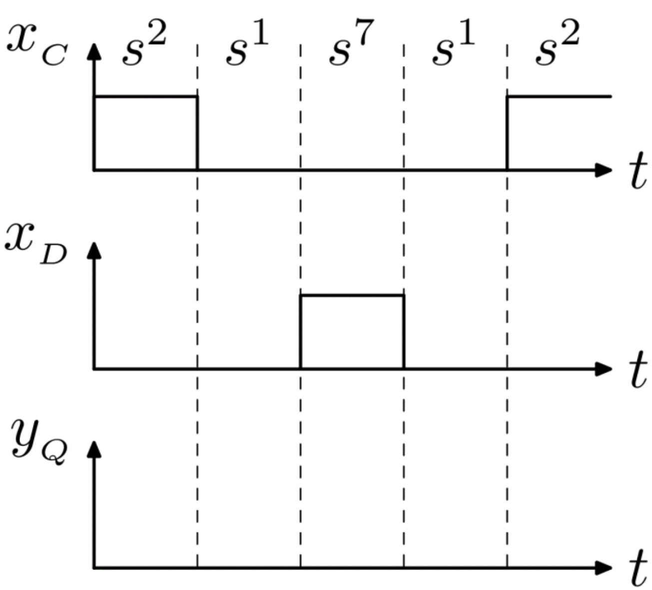
$Q \backslash D$	0	1
0	0	1
1	0	1

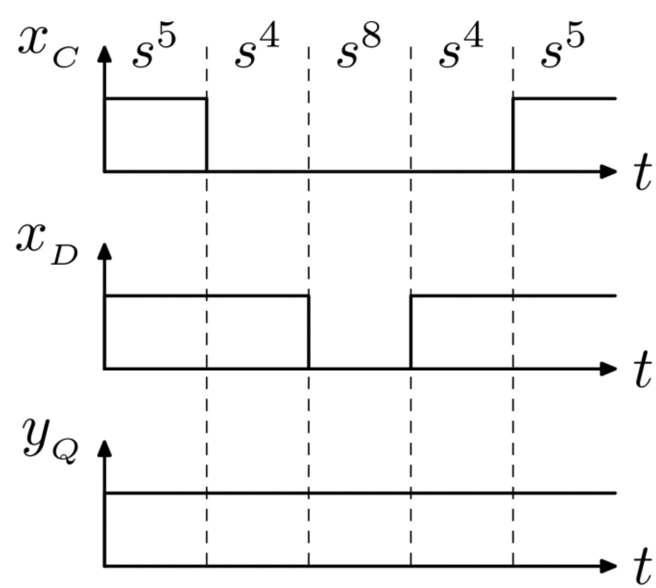
Q'




x_C
s^1 s^2 s^3 s^2 s^1
t
x_D
t
y_Q
t


x_C
s^4 s^5 s^6 s^5 s^4
t
x_D
t
y_Q
t


x_C
s^2 s^1 s^7 s^1 s^2
t
x_D
t
y_Q
t


x_C
s^5 s^4 s^8 s^4 s^5
t
x_D
t
y_Q
t

$s \backslash x_C x_D$	00	01	11	10	y_Q
s^1	s^1	s^7	—	s^2	0
s^2	s^1	—	s^3	s^2	0
s^3	—	s^4	s^3		0
s^4	s^8	s^4	s^5	—	1
s^5	—	s^4	s^5	s^6	1
s^6	s^1	—		s^6	1
s^7		s^7	s^3	—	0
s^8	s^8		—	s^6	1

s'/y_Q

$s \backslash x_C x_D$	00	01	11	10	y_Q
s^1	s^1	s^7	—	s^2	0
s^2	s^1	—	s^3	s^2	0
s^3	—	s^4	s^3	s^2	0
s^4	s^8	s^4	s^5	—	1
s^5	—	s^4	s^5	s^6	1
s^6	s^1	—	s^5	s^6	1
s^7	s^1	s^7	s^3	—	0
s^8	s^8	s^4	—	s^6	1

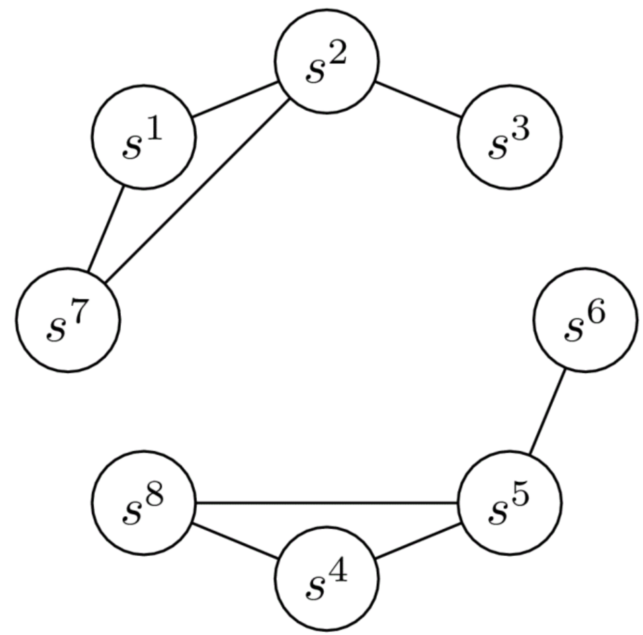
s'/y_Q

Minimization

Triangular table

	s^1	s^2	s^3	s^4	s^5	s^6	s^7
s^2							
s^3	$\cancel{s^4, s^7}$						
s^4	×	×	×				
s^5	×	×	×				
s^6	×	×	×	$\cancel{s^1, s^8}$			
s^7			$\cancel{s^4, s^7}$	×	×	×	
s^8	×	×	×			$\cancel{s^1, s^8}$	×

Compatibility graph



Two reasonable state substitutions:

$$\begin{cases} \{s^1, s^2, s^7\} \rightarrow s^a \\ \{s^4, s^5, s^8\} \rightarrow s^b \\ \{s^3\} \rightarrow s^c \\ \{s^6\} \rightarrow s^d \end{cases}$$

$$\begin{cases} \{s^1, s^7\} \rightarrow s^\alpha \\ \{s^4, s^8\} \rightarrow s^\beta \\ \{s^2, s^3\} \rightarrow s^\gamma \\ \{s^5, s^6\} \rightarrow s^\omega \end{cases}$$

Coding – first try

s	$\rightarrow$	q_1	q_2
s^a	$\rightarrow$	0	0
s^b	$\rightarrow$	1	0
s^c	$\rightarrow$	0	1
s^d	$\rightarrow$	1	1

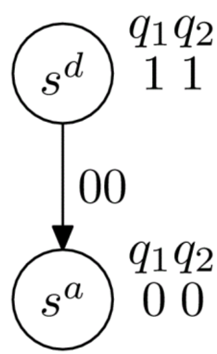
s \ $x_C x_D$	00	01	11	10	y_Q
s^a	s^a	s^a	s^c	s^a	0
s^b	s^b	s^b	s^b	s^d	1
s^c	–	s^b	s^c	s^a	0
s^d	s^a	–	s^b	s^d	1

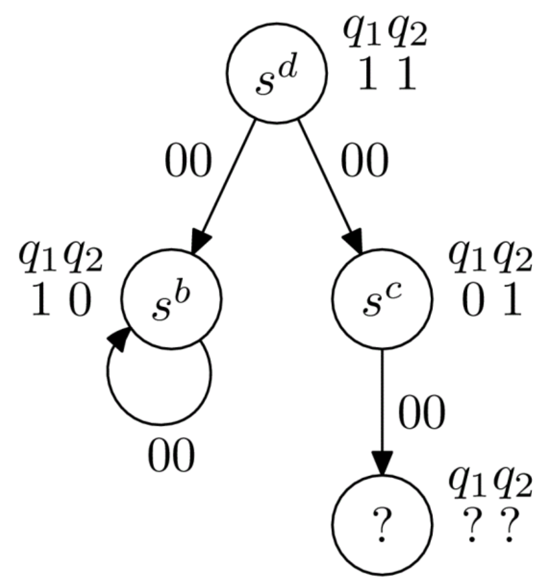
s' / y_Q

Coding causing a critical race

q_1q_2	s \ x_Cx_D	00	01	11	10	y_Q
00 ←	s^a	s^a	s^a	s^c	s^a	0
10 ←	s^b	s^b	s^b	s^b	s^d	1
01 ←	s^c	–	s^b	s^c	s^a	0
11 ←	s^d	s^a	–	s^b	s^d	1

s'/y_Q





Coding causing a critical race

$q_1 q_2$	s \ $x_C x_D$	00	01	11	10	y_Q
00 $\leftarrow$	s^a	s^a	s^a	s^c	s^a	0
10 $\leftarrow$	s^b	s^b	s^b	s^b	s^d	1
01 $\leftarrow$	s^c	–	s^b	s^c	s^a	0
11 $\leftarrow$	s^d	s^a	–	s^b	s^d	1

s'/y_Q

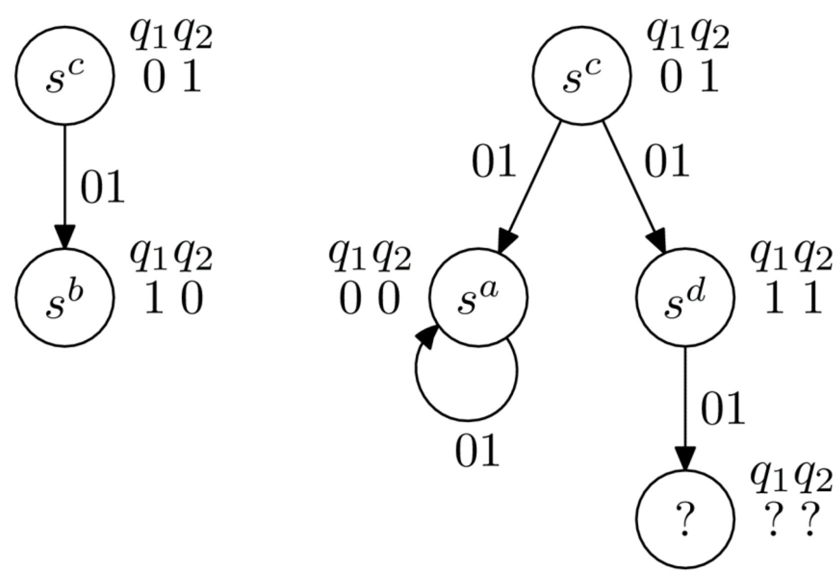
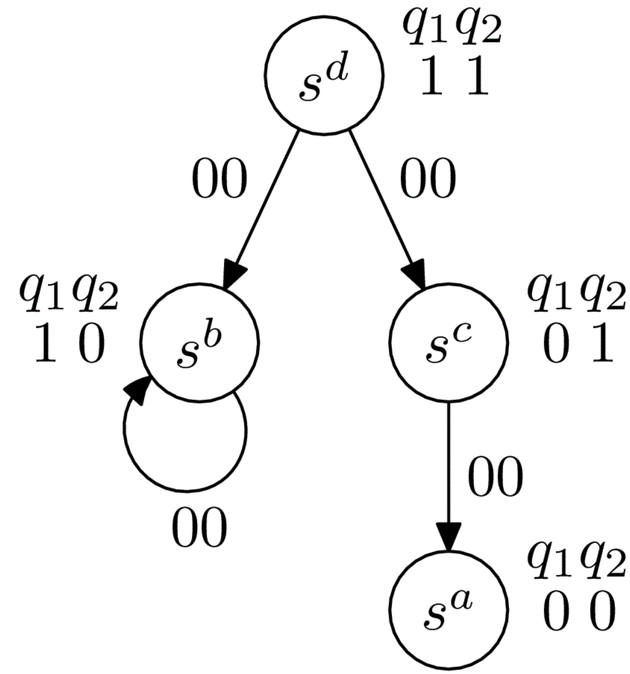


Illustration of a critical and a non-critical race

q_1q_2 \ s \ x_Cx_D	00	01	11	10	y_Q
$00 \leftarrow s^a$	s^a	s^a	s^c	s^a	0
$10 \leftarrow s^b$	s^b	s^b	s^b	s^d	1
$01 \leftarrow s^c$	$\times / s^a$	s^b	s^c	s^a	0
$11 \leftarrow s^d$	s^a	—	s^b	s^d	1

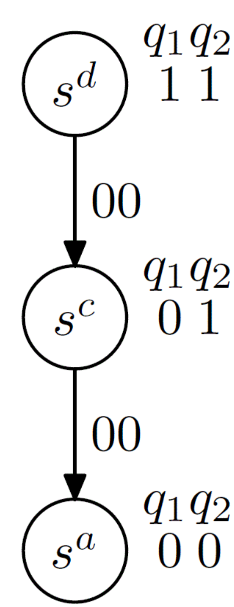
s'/y_Q



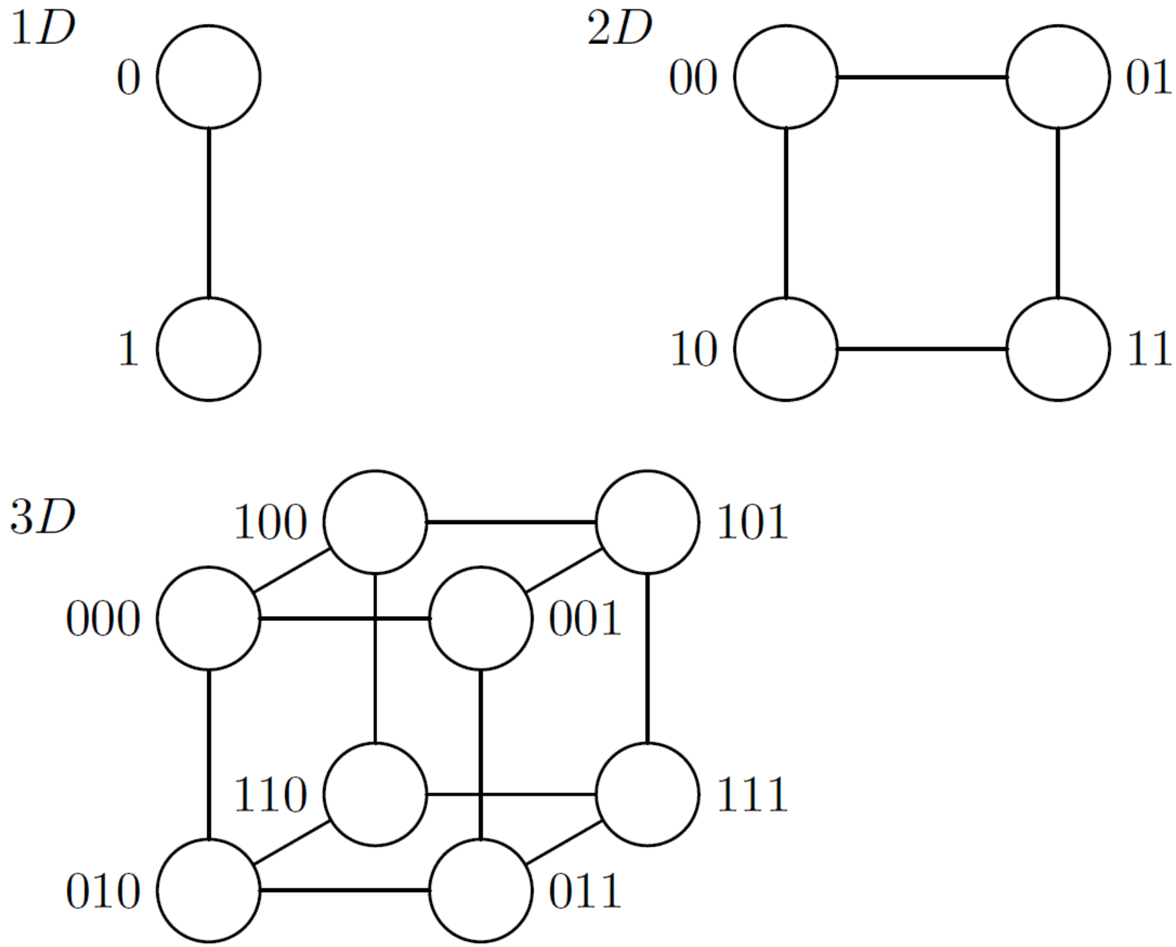
Partial solution by introducing a cyclic transition

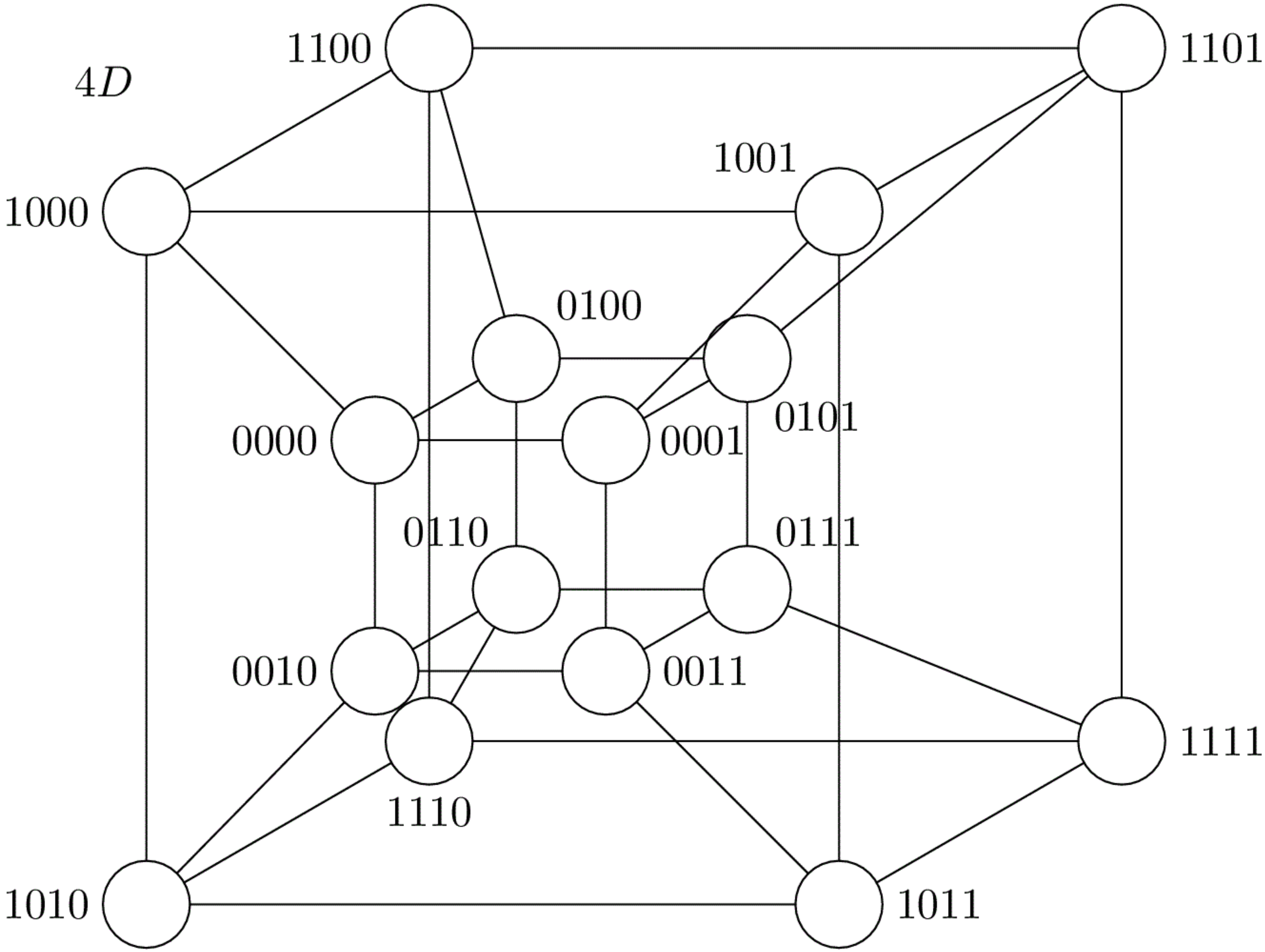
q_1q_2 \ s \ $x_C x_D$	00	01	11	10	y_Q
$00 \leftarrow s^a$	s^a	s^a	s^c	s^a	0
$10 \leftarrow s^b$	s^b	s^b	s^b	s^d	1
$01 \leftarrow s^c$	$\cancel{\times}/s^a$	s^b	s^c	s^a	0
$11 \leftarrow s^d$	$\cancel{s^a}/s^c$	$-$	s^b	s^d	1

s'/y_Q

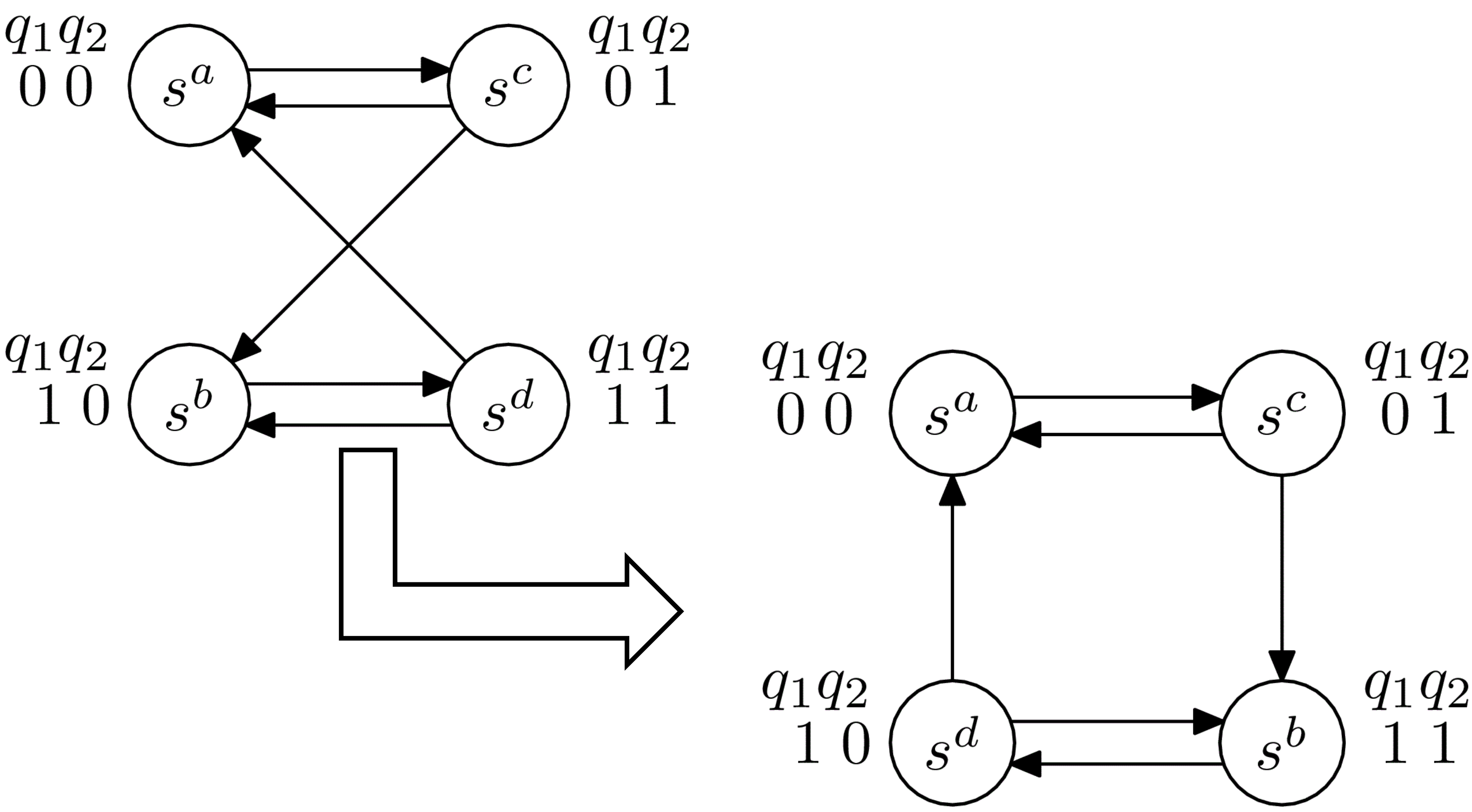


Hypercube




4D
1100
1101
1001
1000
0100
0000
0001
0101
0110
0111
0010
0011
1111
1110
1010
1011

State transition graph transformation - coding method



Final coding

q_1q_2	s \ x_Cx_D	00	01	11	10	y_Q
00 $\leftarrow$	s^a	s^a	s^a	s^c	s^a	0
01 $\leftarrow$	s^c	—	s^b	s^c	s^a	0
11 $\leftarrow$	s^b	s^b	s^b	s^b	s^d	1
10 $\leftarrow$	s^d	s^a	—	s^b	s^d	1

s'/y_Q

Coded state transition and output table

q_1q_2 \ x_Cx_D	00	01	11	10	y_Q
00	00	00	01	00	0
01	–	11	01	00	0
11	11	11	11	10	1
10	00	–	11	10	1

$q_1'q_2'/y_Q$

$$\Rightarrow \quad y_Q = q_1$$

$q_1q_2 \backslash x_Cx_D$	00	01	11	10
00	0	0	0	0
01	–	1	0	0
11	1	1	1	1
10	0	–	1	1

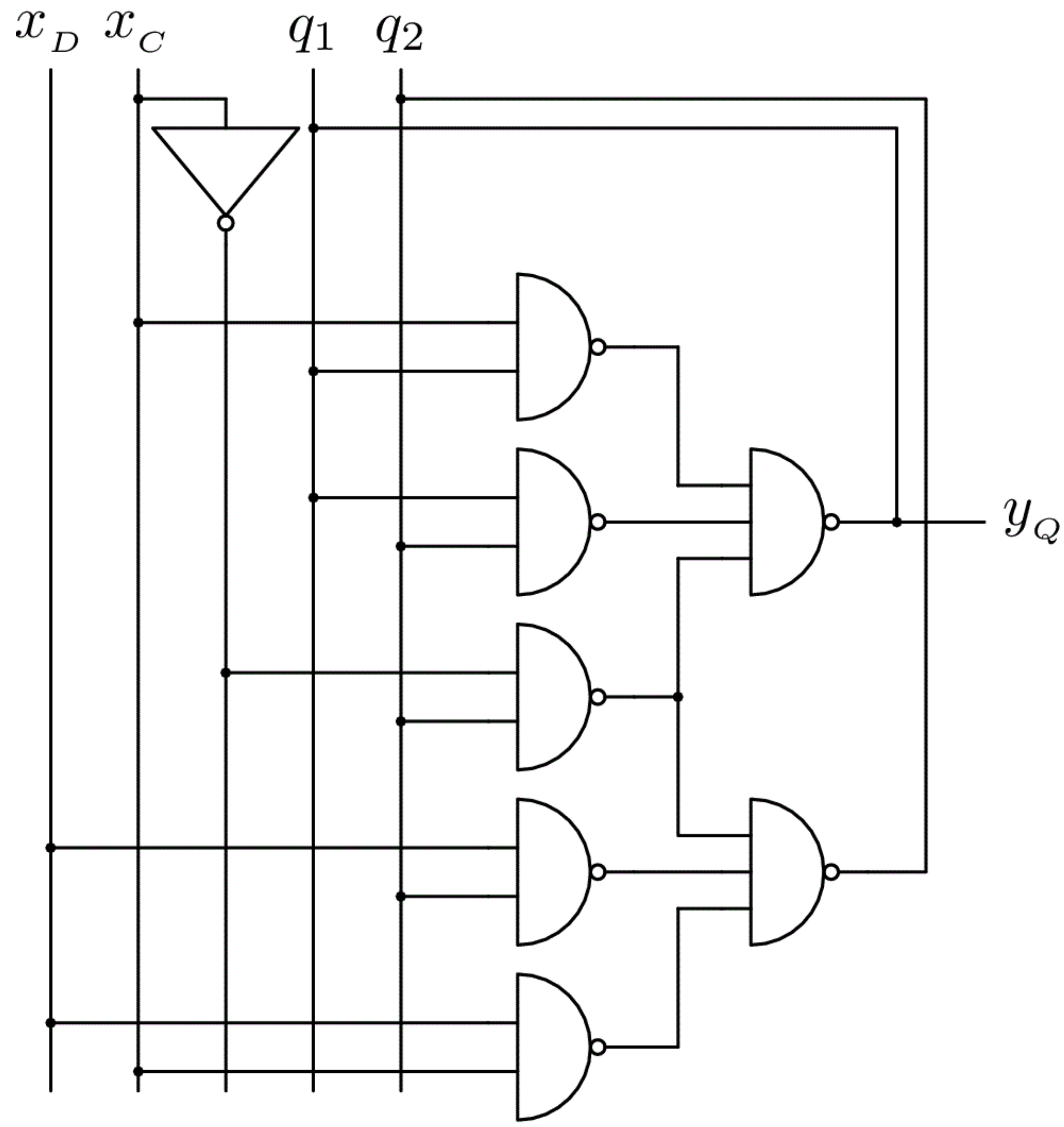
q_1'

$q_1q_2 \backslash x_Cx_D$	00	01	11	10
00	0	0	1	0
01	–	1	1	0
11	1	1	1	0
10	0	–	1	0

q_2'

$$\begin{cases} q_1' = q_1x_C + q_1q_2 + q_2\overline{x}_C \\ q_2' = q_2x_D + x_Cx_D + q_2\overline{x}_C \\ y_Q = q_1 \end{cases}$$

Circuit diagram of the negative-edge triggered D-type flip-flop



- D-type flip-flop is an asynchronous automaton
- Synchronous automatons are constructed out of D-type flip-flops and combinational logic

  $\Rightarrow$ Every synchronous automaton is a special case of an asynchronous automaton

Alternative method of minimizing asynchronous automatons

1. It is a Moore automaton table.
2. Each row contains a single stable state.
3. All unstable next states s^k appear only in the column in which the state s^k is stable.

- Property (3) $\Rightarrow$ conditions for compatibility can appear only between states that are stable in the same column

The process of minimization can be divided into two phases:

1. Finding pseudoequivalent states
   (at this stage conditional compatibility is taken into account)

2. Finding pseudocompatible states
   (at this stage the states are pseudocompatible unconditionally)

Pseudoequivalent states

States that are being stable in the same column of the initial table and at the same time are being compatible.

$$\left(s^i \cong s^j\right) \iff \exists\Big(x : \left(\delta(s^i, x) = s^i\right) \bigwedge \left(\delta(s^j, x) = s^j\right) \bigwedge \left(s^i \approxeq s^j\right)$$

Moore pseudocompatible states

States for which the next states are non-contradictory and the outputs are also non-contradictory

$$\left(s^i \approx s^j\right) \iff \left(\forall x \;\; \left(\delta(s^i, x) \doteq \delta(s^j, x)\right)\right) \wedge \left(\lambda_1(s^i) \doteq \lambda_1(s^j)\right)$$

Mealy pseudocompatible states

States for which the next states are non-contradictory

$$\left(s^i \sim s^j\right) \iff \forall x \left(\delta(s^i, x) \doteq \delta(s^j, x)\right)$$

Minimisation of the number of states of an asynchronous automaton – mathematical formulation of phase 1

$$\forall\Big(\big({}_{A_1}\boldsymbol{S}^k \subset {}_{A_1}\boldsymbol{S}\big) \;:\; \Big(\forall\big({}_{A_1}s^{k_i}, {}_{A_1}s^{k_j} \in {}_{A_1}\boldsymbol{S}^k\big) \;\; \big({}_{A_1}s^{k_i} \cong {}_{A_1}s^{k_j}\big)\Big)\Big)$$

and $k, k' = 1, \ldots, n$

the following conditions must be fulfilled:

$$\big({}_{A_1}s^{k_i} \in {}_{A_1}\boldsymbol{S}^k\big) \bigwedge \big({}_{A_1}s^{k_j} \in {}_{A_1}\boldsymbol{S}^{k'}\big) \bigwedge (k \neq k') \;\Rightarrow\; (k_i \neq k_j)$$

$$\Big(\bigcup_{k=1}^{n} {}_{A_1}\boldsymbol{S}^k\Big) = {}_{A_1}\boldsymbol{S} \;\longrightarrow\; \text{cover condition}$$

$$\Big(\forall\big({}_{A_1}s^{k_i}, {}_{A_1}s^{k_j}, x\big) : \big({}_{A_1}s^{k_i}, {}_{A_1}s^{k_j} \in {}_{A_1}\boldsymbol{S}^k\big)\Big)$$

$$\Rightarrow \;\Big(\exists_{A_1}\boldsymbol{S}^{k'} : \big(\delta({}_{A_1}s^{k_i}, x), \delta({}_{A_1}s^{k_j}, x) \in {}_{A_1}\boldsymbol{S}^{k'}\big)\Big)$$

$\longrightarrow$ closure condition

$$ {}_{A_1}\boldsymbol{S}^k \rightarrow {}_{A_2}s^k, \quad {}_{A_2}s^k \in {}_{A_2}\boldsymbol{S} $$

where:

${}_{A_1}\boldsymbol{S}$ – set of states of initial automaton A_1

${}_{A_1}\boldsymbol{S}^k$ – kth set of pseudoequivalent states of automaton A_1

${}_{A_1}\boldsymbol{S}^{k'}$ – k'th set of pseudoequivalent states of automaton A_1

${}_{A_2}\boldsymbol{S}$ – set of states of minimized automaton A_2

n – number of sets of pseudoequivalent states

Minimisation of the number of states of an asynchronous automaton - mathematical formulation of phase 2

$$\forall\Bigg(\left({}_{A_1}\boldsymbol{S}^k \subset {}_{A_1}\boldsymbol{S}\right) \;:\; \left(\forall\left({}_{A_1}s^{k_i}, {}_{A_1}s^{k_j} \in {}_{A_1}\boldsymbol{S}^k\right) \;\left({}_{A_1}s^{k_i} \approx {}_{A_1}s^{k_j}\right)\right)\Bigg)$$

and $k, k' = 1, \ldots, n$

the following conditions must be fulfilled:

$$\left({}_{A_1}s^{k_i} \in {}_{A_1}\boldsymbol{S}^k\right) \bigwedge \left({}_{A_1}s^{k_j} \in {}_{A_1}\boldsymbol{S}^{k'}\right) \bigwedge (k \neq k') \;\Rightarrow\; (k_i \neq k_j)$$

$$\left(\bigcup_{k=1}^{n} {}_{A_1}\boldsymbol{S}^k\right) = {}_{A_1}\boldsymbol{S} \;\longrightarrow\; \text{cover condition}$$

$${}_{A_1}\boldsymbol{S}^k \rightarrow {}_{A_2}s^k, \qquad {}_{A_2}s^k \in {}_{A_2}\boldsymbol{S}$$

where:

${}_{A_1}\boldsymbol{S}$ – set of states of initial automaton A_1

${}_{A_1}\boldsymbol{S}^{k}$ – kth set of pseudocompatible states of automaton A_1

${}_{A_1}\boldsymbol{S}^{k'}$ – k'th set of pseudocompatible states of automaton A_1

${}_{A_2}\boldsymbol{S}$ – set of states of minimized automaton A_2

n – number of sets of pseudocompatible states

- If Moore pseudocompatibility ($\approx$) is substituted by Mealy pseudocompatibility ($\sim$) the above minimization procedure will lead to a Mealy automaton instead of a Moore automaton.

Negative-edge triggered D-type flip-flop (cntd)

s \ $x_C x_D$	00	01	11	10	y_Q
s^1	s^1	s^7	—	s^2	0
s^2	s^1	—	s^3	s^2	0
s^3	—	s^4	s^3	s^2	0
s^4	s^8	s^4	s^5	—	1
s^5	—	s^4	s^5	s^6	1
s^6	s^1	—	s^5	s^6	1
s^7	s^1	s^7	s^3	—	0
s^8	s^8	s^4	—	s^6	1

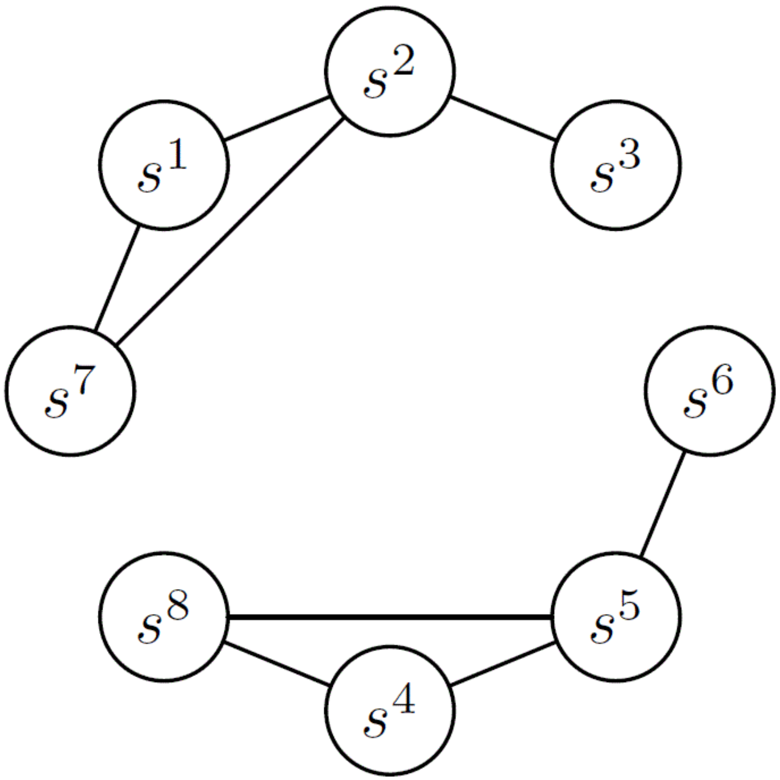
s'/y_Q

$s^1 \not\cong s^8$ because $\lambda_1(s^1) = 0, \ \lambda_1(s^8) = 1$
$s^3 \not\cong s^5$ because $\lambda_1(s^3) = 0, \ \lambda_1(s^5) = 1$
$s^2 \not\cong s^6$ because $\lambda_1(s^2) = 0, \ \lambda_1(s^6) = 1$
$s^4 \not\cong s^7$ because $\lambda_1(s^4) = 1, \ \lambda_1(s^7) = 0$

$$\Rightarrow \lambda_1(s^1) \neq \lambda_1(s^8)$$
$$\Rightarrow \lambda_1(s^3) \neq \lambda_1(s^5)$$
$$\Rightarrow \lambda_1(s^2) \neq \lambda_1(s^6)$$
$$\Rightarrow \lambda_1(s^4) \neq \lambda_1(s^7)$$

$\Rightarrow$ there are no pseudoeqivalent states

Graph of Moore pseudocompatible states



$$\begin{cases} \{s^1, s^2, s^7\} \longrightarrow s^a \\ \{s^4, s^5, s^8\} \longrightarrow s^b \\ \{s^3\} \longrightarrow s^c \\ \{s^6\} \longrightarrow s^d \end{cases}$$

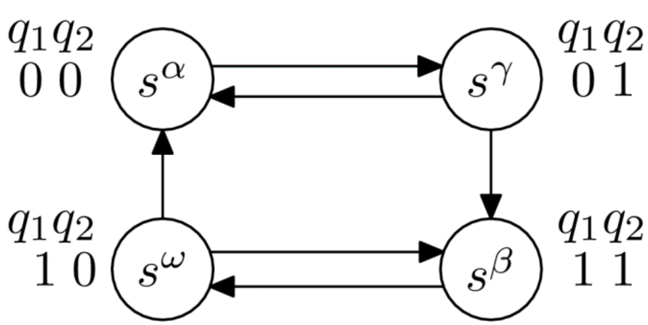
$$\begin{cases} \{s^1, s^7\} \longrightarrow s^\alpha \\ \{s^4, s^8\} \longrightarrow s^\beta \\ \{s^2, s^3\} \longrightarrow s^\gamma \\ \{s^5, s^6\} \longrightarrow s^\omega \end{cases}$$

Pseudocompatible states are <u>unconditionally pseudocompatible</u>, so in the process of minimization only <u>the cover condition</u> must be fullled.

State transition and output table for the second state substitution

s \ $x_C x_D$	00	01	11	10	y_Q
s^α	s^α	s^α	s^γ	s^γ	0
s^β	s^β	s^β	s^ω	s^ω	1
s^γ	s^α	s^β	s^γ	s^γ	0
s^ω	s^α	s^β	s^ω	s^ω	1

s' / y_Q



$q_1 q_2$	s \ $x_C x_D$	00	01	11	10	y_Q
$00 \leftarrow$	s^α	s^α	s^α	s^γ	s^γ	0
$01 \leftarrow$	s^γ	s^α	s^β	s^γ	s^γ	0
$11 \leftarrow$	s^β	s^β	s^β	s^ω	s^ω	1
$10 \leftarrow$	s^ω	s^α	s^β	s^ω	s^ω	1

Coding

s' / y_Q

$q_1q_2 \backslash x_Cx_D$	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	1	1	1	1
10	0	1	1	1

q_1'

$q_1q_2 \backslash x_Cx_D$	00	01	11	10	y_Q
00	00	00	01	01	0
01	00	11	01	01	0
11	11	11	10	10	1
10	00	11	10	10	1

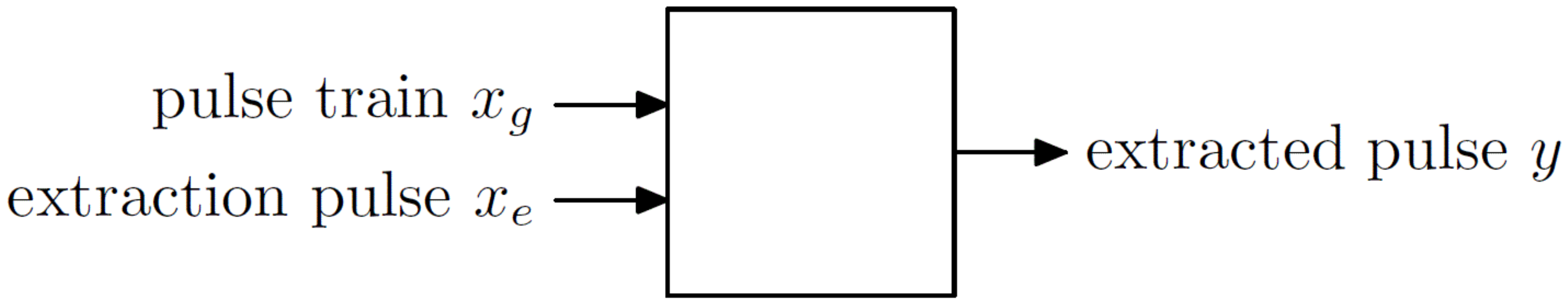
$q_1'q_2'/y_Q$

$q_1q_2 \backslash x_Cx_D$	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	1	1	0	0
10	0	1	0	0

q_2'

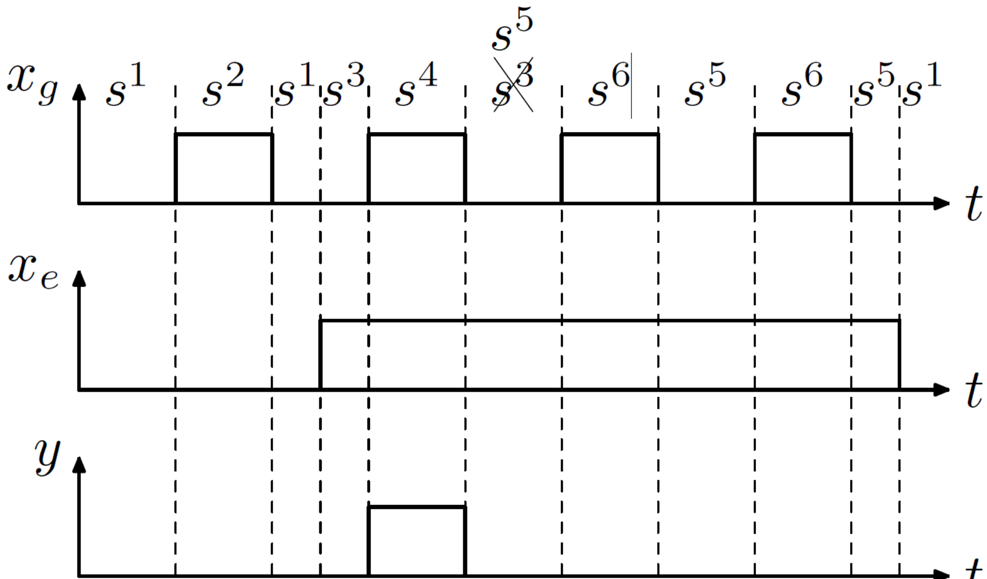
$$\begin{cases} q_1' = q_1x_C + q_1x_D + q_1q_2 + q_1\overline{x}_Cx_D \\ q_2' = \overline{q}_1x_C + q_1q_2\overline{x}_C + q_2\overline{x}_Cx_D + q_1\overline{x}_Cx_D + \overline{q}_1q_2x_D \\ y_Q = q_1 \end{cases}$$

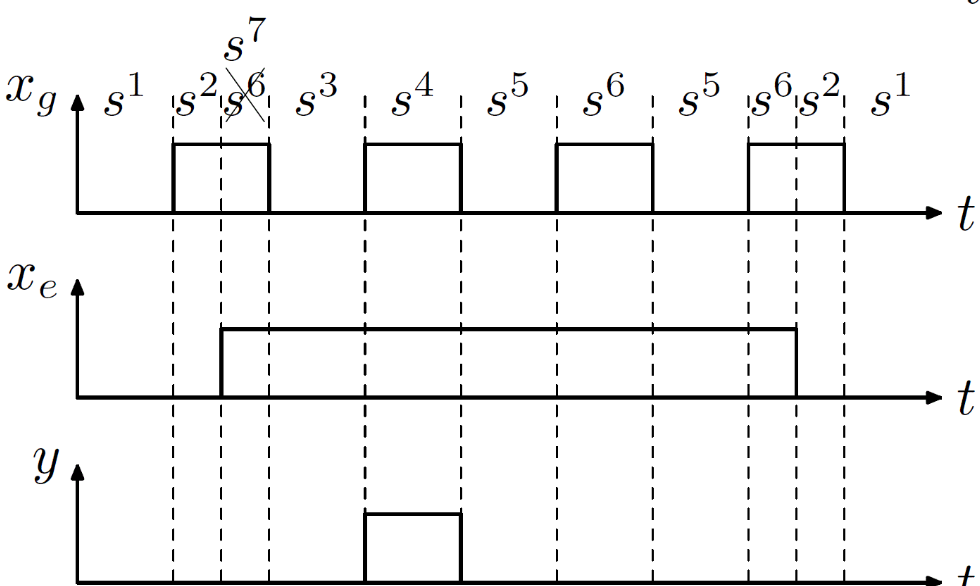
Single-pulse extraction circuit



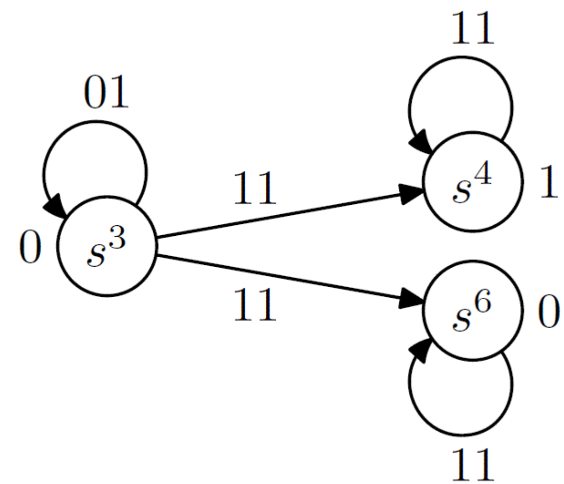
$$T|_{x_e=0} \gg T|_{x_g=0}$$
$$T|_{x_e=1} \gg T|_{x_g=1}$$

- Excel generated solutions will not be accepted
- Please provide hand-made truth tables with islands marked

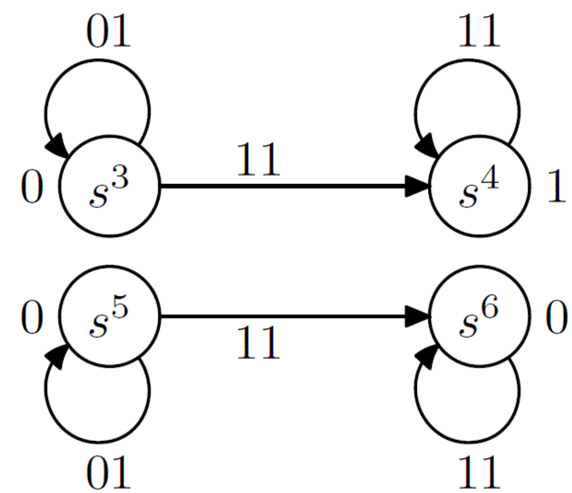




Cause of the problem



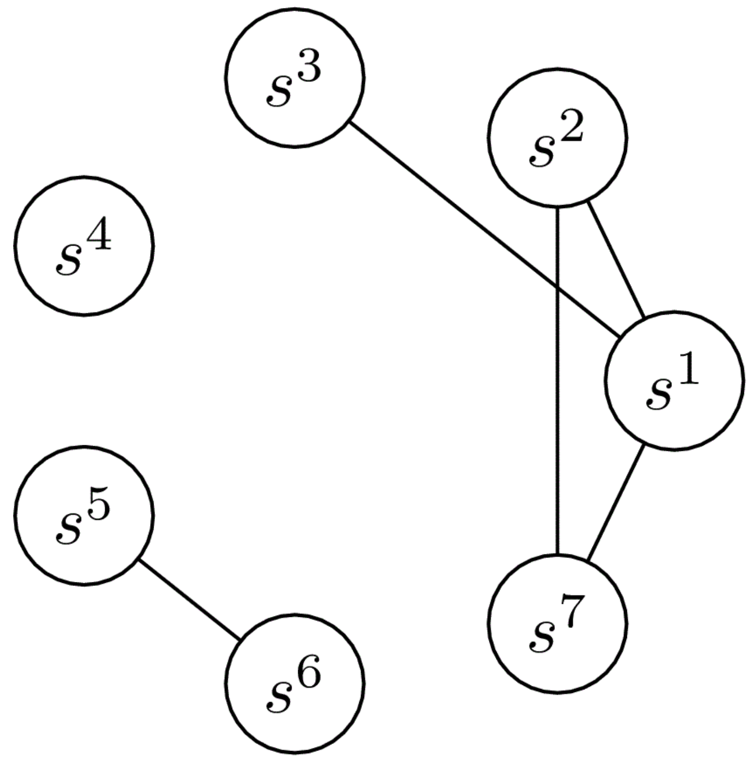
Solution to the problem



$s \backslash x_g x_e$	00	01	11	10	y
s^1	s^1	s^3	—	s^2	0
s^2	s^1	—	s^7	s^2	0
s^3	—	s^3	s^4	—	0
s^4	—	s^5	s^4	—	1
s^5	s^1	s^5	s^6	—	0
s^6	—	s^5	s^6	s^2	0
s^7	—	s^3	s^7	—	0

s'/y

Graph of Moore pseudocompatible states



<u>there are no pseudoeqivalent states</u>

Two reasonable state substitutions:

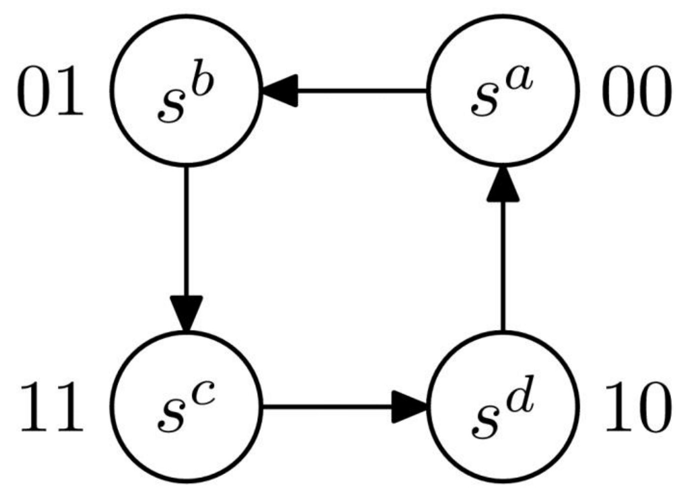
$$\left\{\begin{array}{r} \{s^1, s^2, s^7\} \rightarrow s^a \\ \{s^3\} \rightarrow s^b \\ \{s^4\} \rightarrow s^c \\ \{s^5, s^6\} \rightarrow s^d \end{array}\right. \qquad \left\{\begin{array}{r} \{s^1, s^3\} \rightarrow s^\alpha \\ \{s^2, s^7\} \rightarrow s^\beta \\ \{s^4\} \rightarrow s^\gamma \\ \{s^5, s^6\} \rightarrow s^\omega \end{array}\right.$$

State transition and output table for the first state substitution

$s \backslash x_g x_e$	00	01	11	10	y
s^a	s^a	s^b	s^a	s^a	0
s^b	—	s^b	s^c	—	0
s^c	—	s^d	s^c	—	1
s^d	s^a	s^d	s^d	s^a	0

s'/y

Coding



$q_1q_2 \backslash x_gx_e$	00	01	11	10	y
00	00	01	00	00	0
01	–	01	11	–	0
11	–	10	11	–	1
10	00	10	10	00	0

$q_1'q_2'/y$

$q_1q_2 \backslash x_gx_e$	00	01	11	10
00	0	0	0	0
01	–	0	1	–
11	–	1	1	–
10	0	1	1	0

q_1'

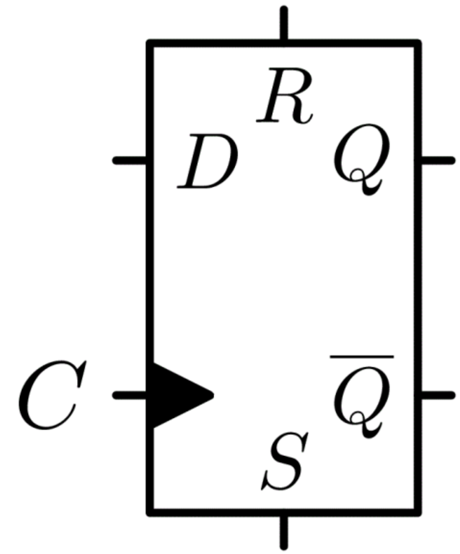
$q_1q_2 \backslash x_gx_e$	00	01	11	10
00	0	1	0	0
01	–	1	1	–
11	–	0	1	–
10	0	0	0	0

q_2'

$$
\begin{cases}
q_1' = q_2x_g + q_1x_e \\
q_2' = q_2x_g + \overline{q}_1q_2 + \overline{q}_1\overline{x}_gx_e \\
y = q_1q_2
\end{cases}
$$

SR-type static flip-flop

Q	$\rightarrow$	Q'	S	R
0	$\rightarrow$	0	0	–
0	$\rightarrow$	1	1	0
1	$\rightarrow$	0	0	1
1	$\rightarrow$	1	–	0



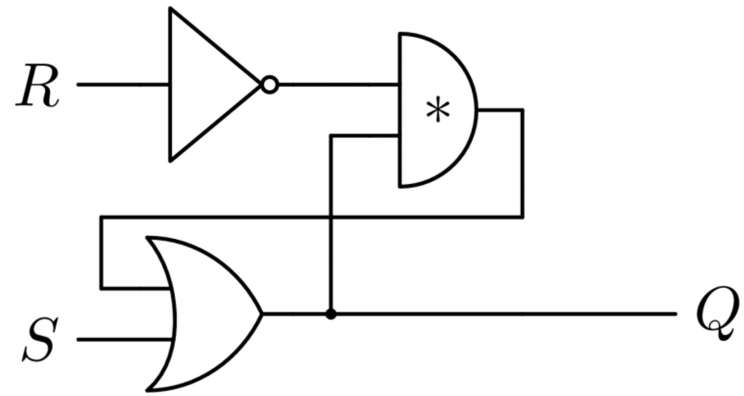
Q'

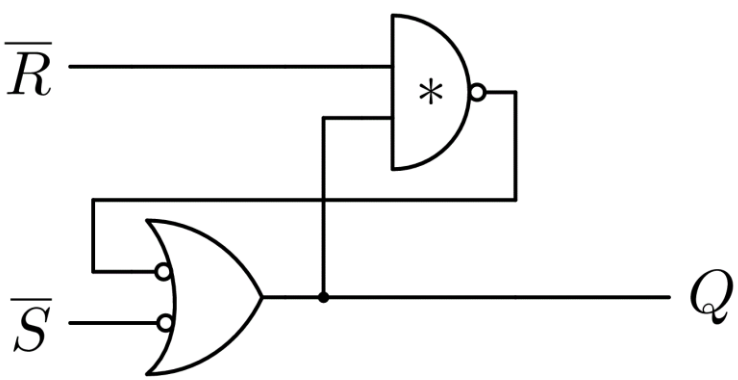
$Q \backslash SR$	00	01	11	10
0	0	0	–	1
1	1	0	–	1

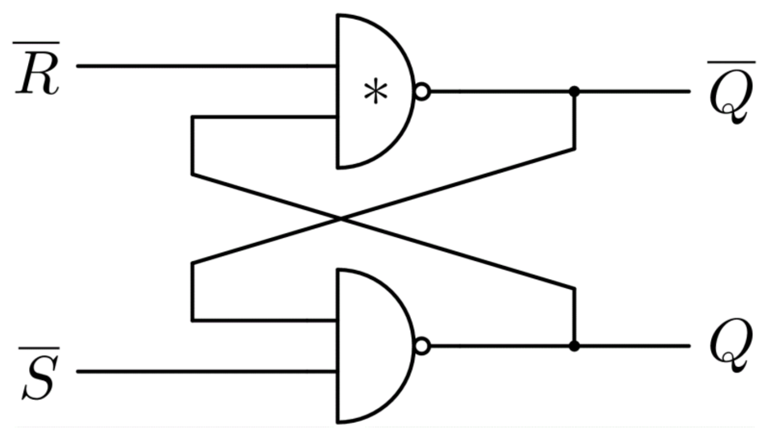
Q'

$Q \backslash SR$	00	01	11	10
0	0	0	–	1
1	1	0	–	1

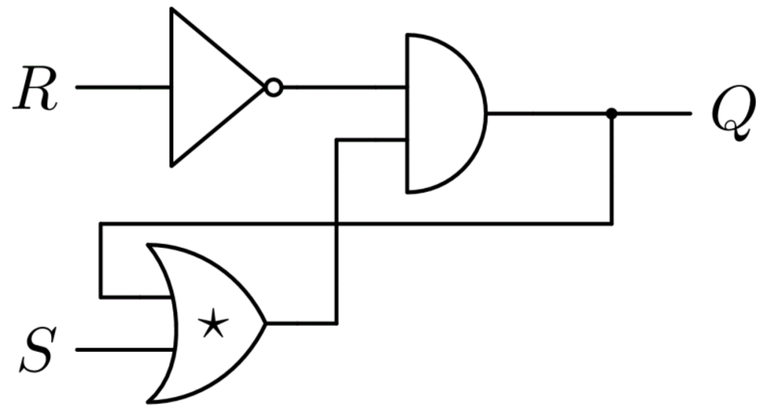
$$Q' = S + \underbrace{Q\overline{R}}_{*}$$

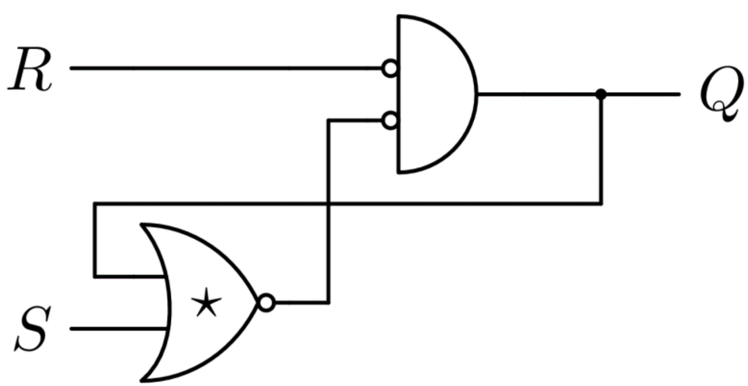


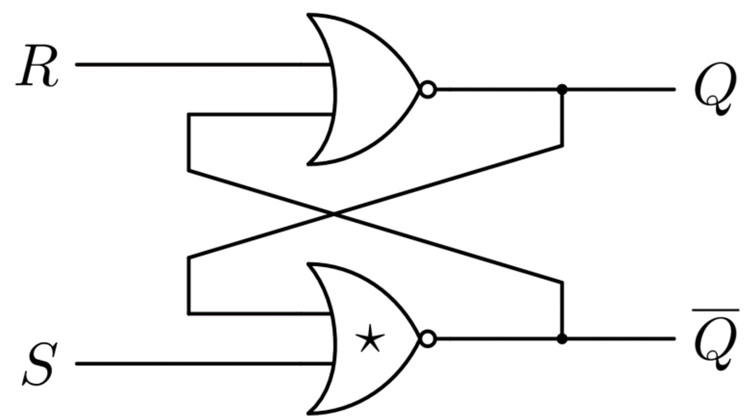




$$Q' = \overline{R}\,\underbrace{(Q+S)}_{\star}$$







$q_1q_2 \backslash x_gx_e$	00	01	11	10
00	0	0	0	0
01	–	0	1	–
11	–	1	1	–
10	0	1	1	0

q_1'

$$R_1 = \overline{x}_e$$

$$S_1 = q_2 x_g$$

$q_1q_2 \backslash x_gx_e$	00	01	11	10
00	–	–	–	–
01	–	–	0	–
11	–	0	0	–
10	1	0	0	1

R_1

$q_1q_2 \backslash x_gx_e$	00	01	11	10
00	0	0	0	0
01	–	0	1	–
11	–	–	–	–
10	0	–	–	0

S_1

q_1q_2 \ x_gx_e	00	01	11	10
00	0	1	0	0
01	–	1	1	–
11	–	0	1	–
10	0	0	0	0

q_2'

q_1q_2 \ x_gx_e	00	01	11	10
00	–	0	–	–
01	–	0	0	–
11	–	1	0	–
10	–	–	–	–

R_2

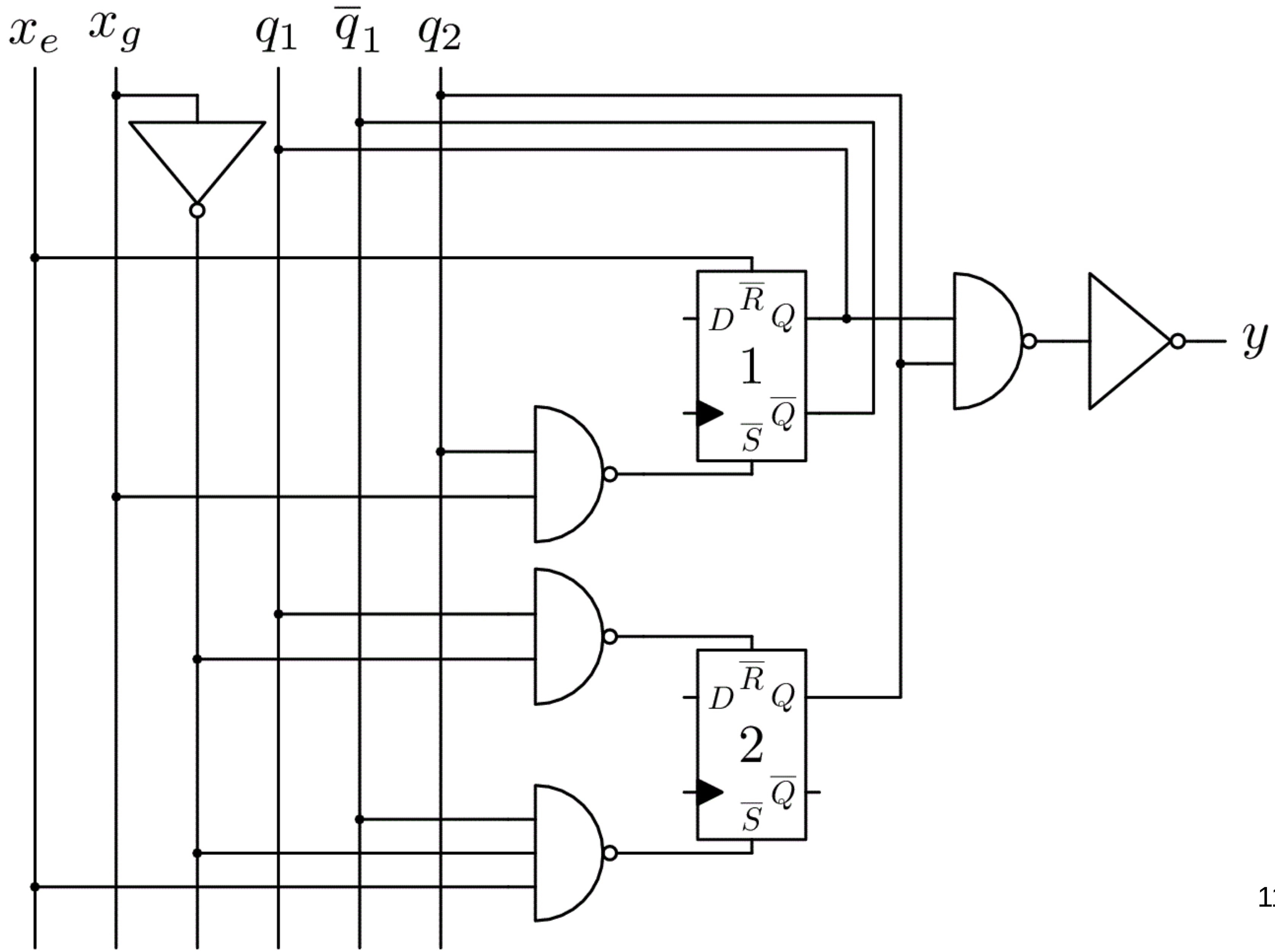
q_1q_2 \ x_gx_e	00	01	11	10
00	0	1	0	0
01	–	–	–	–
11	–	0	–	–
10	0	0	0	0

S_2

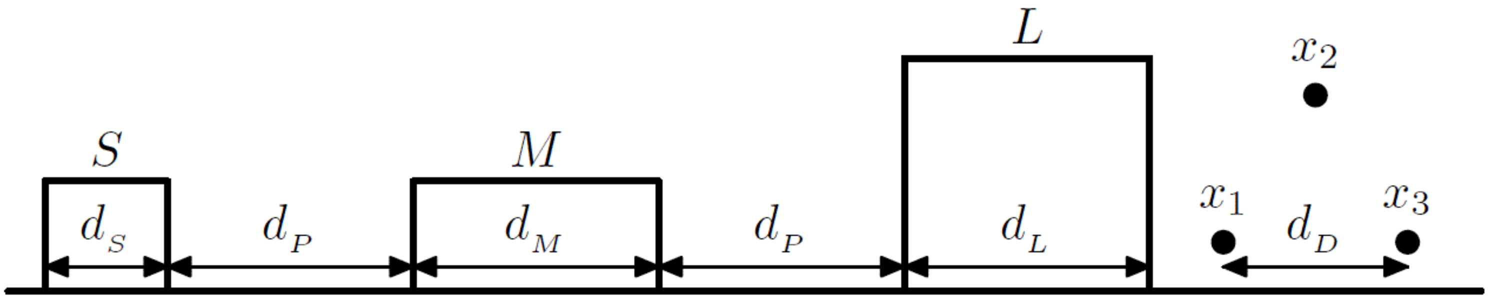
$$R_2 = q_1\overline{x}_g$$

$$S_2 = \overline{q}_1\overline{x}_g x_e$$

$$y = q_1q_2$$

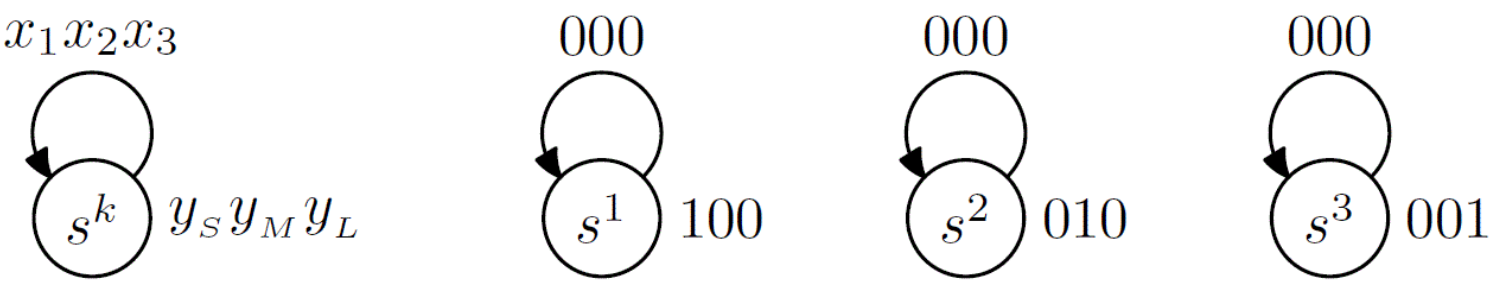

x_e
x_g
q_1
$\overline{q}_1$
q_2
D
$\overline{R}$
Q
1
$\overline{Q}$
$\overline{S}$
D
$\overline{R}$
Q
2
$\overline{Q}$
$\overline{S}$
y

Parcel sorter

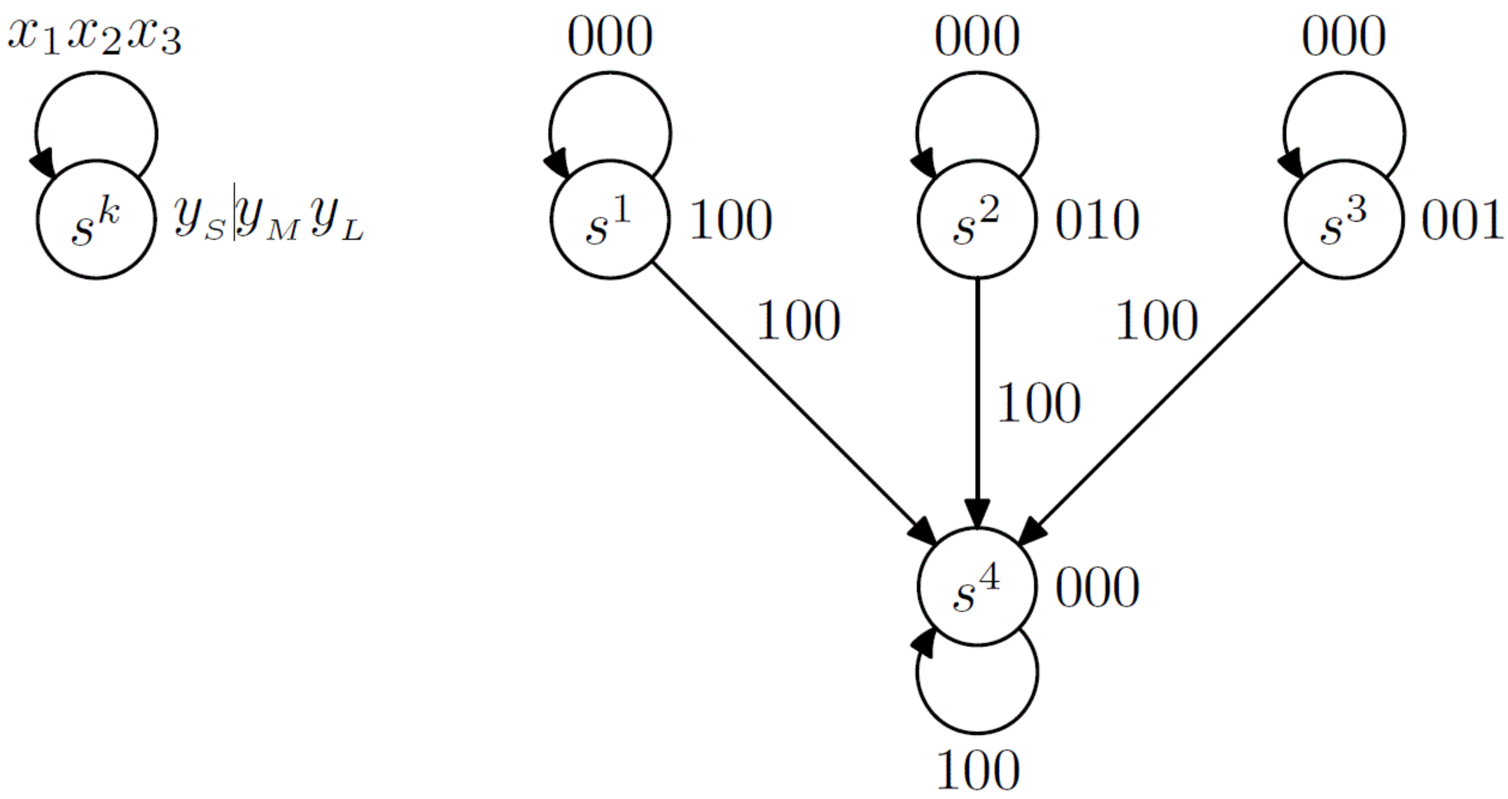


$$d_S < d_D < d_M < d_L$$
$$d_D < d_P$$

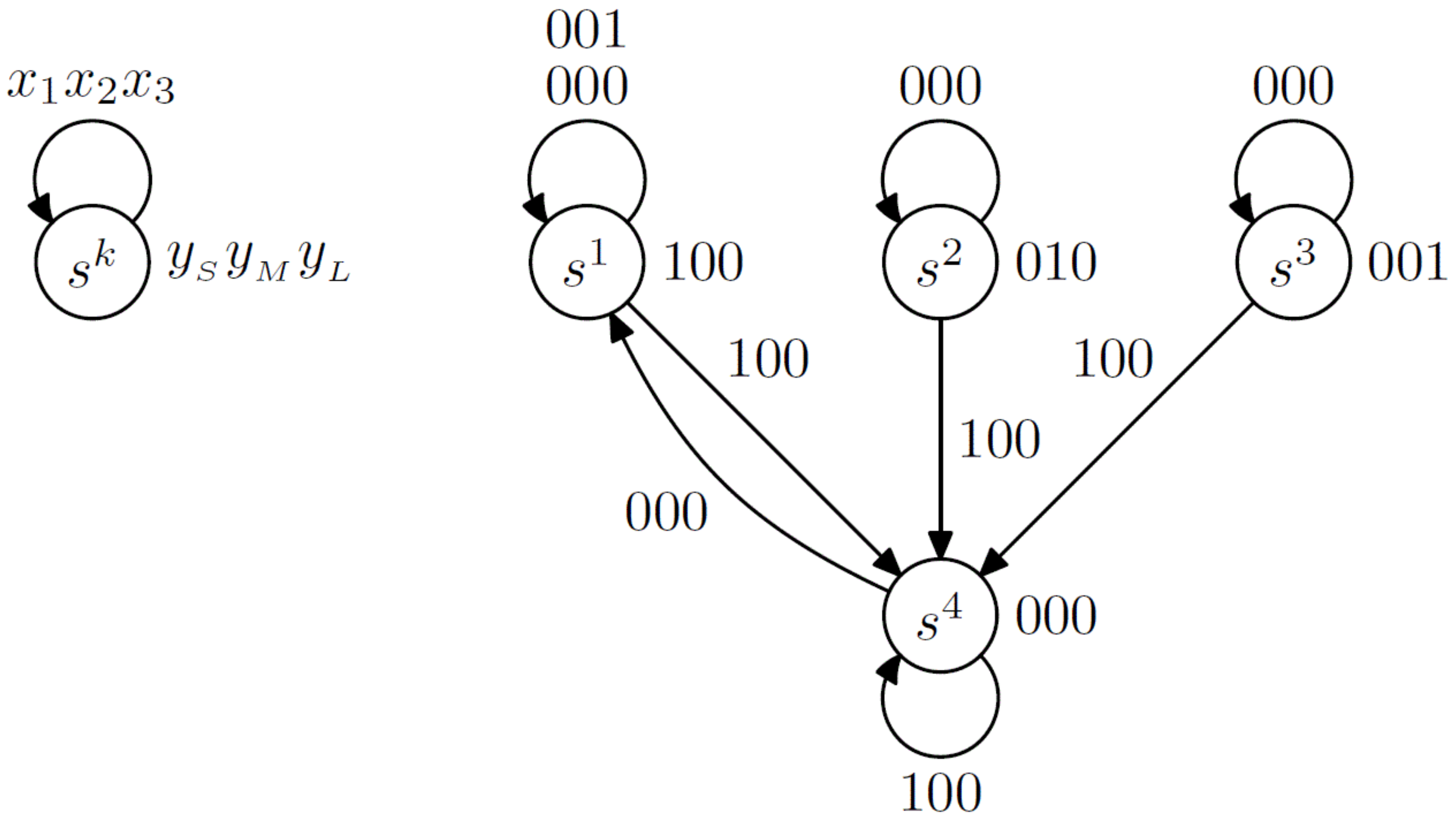
Initial situation - no parcel at the detecting station



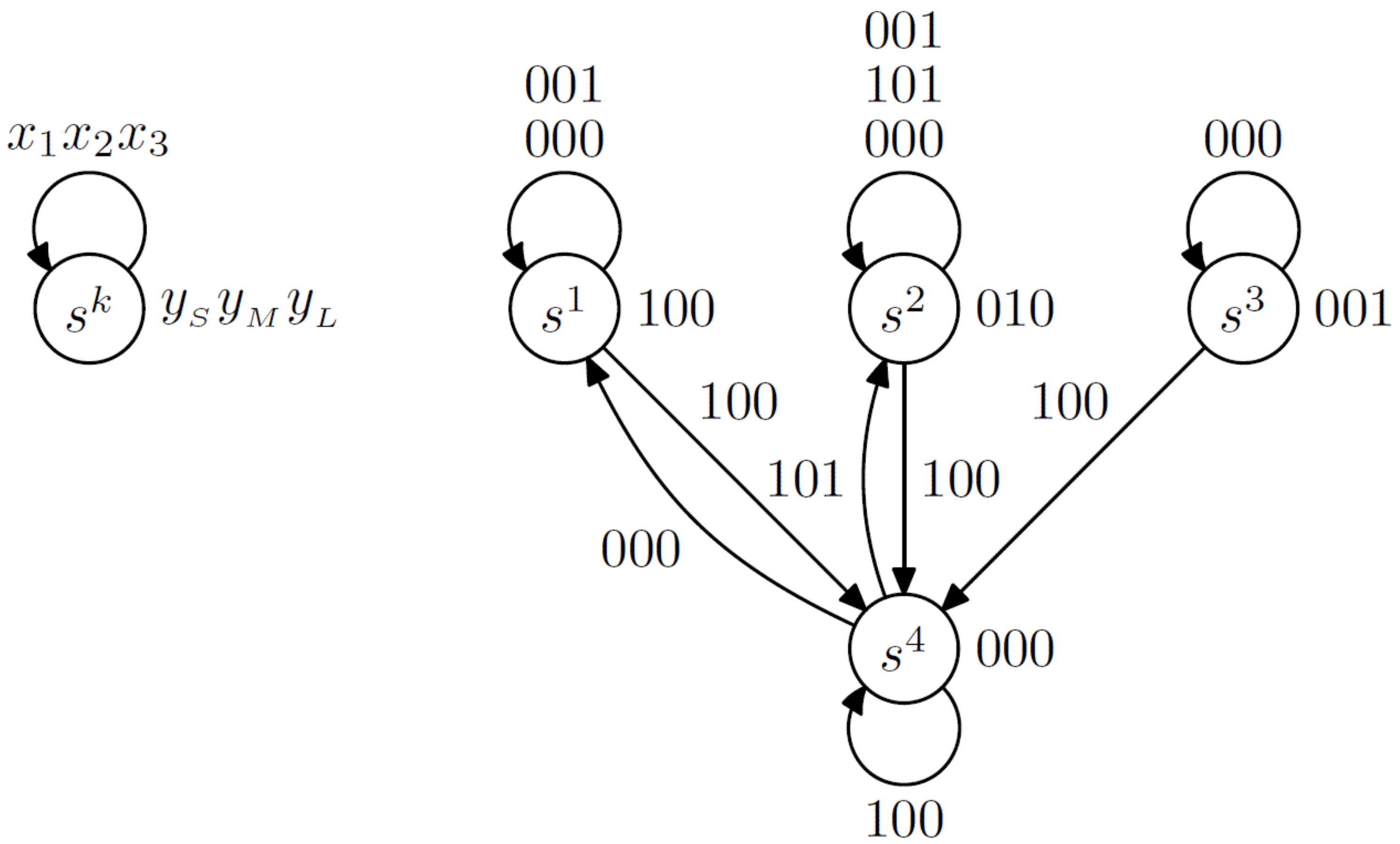
A parcel crossing the first detector



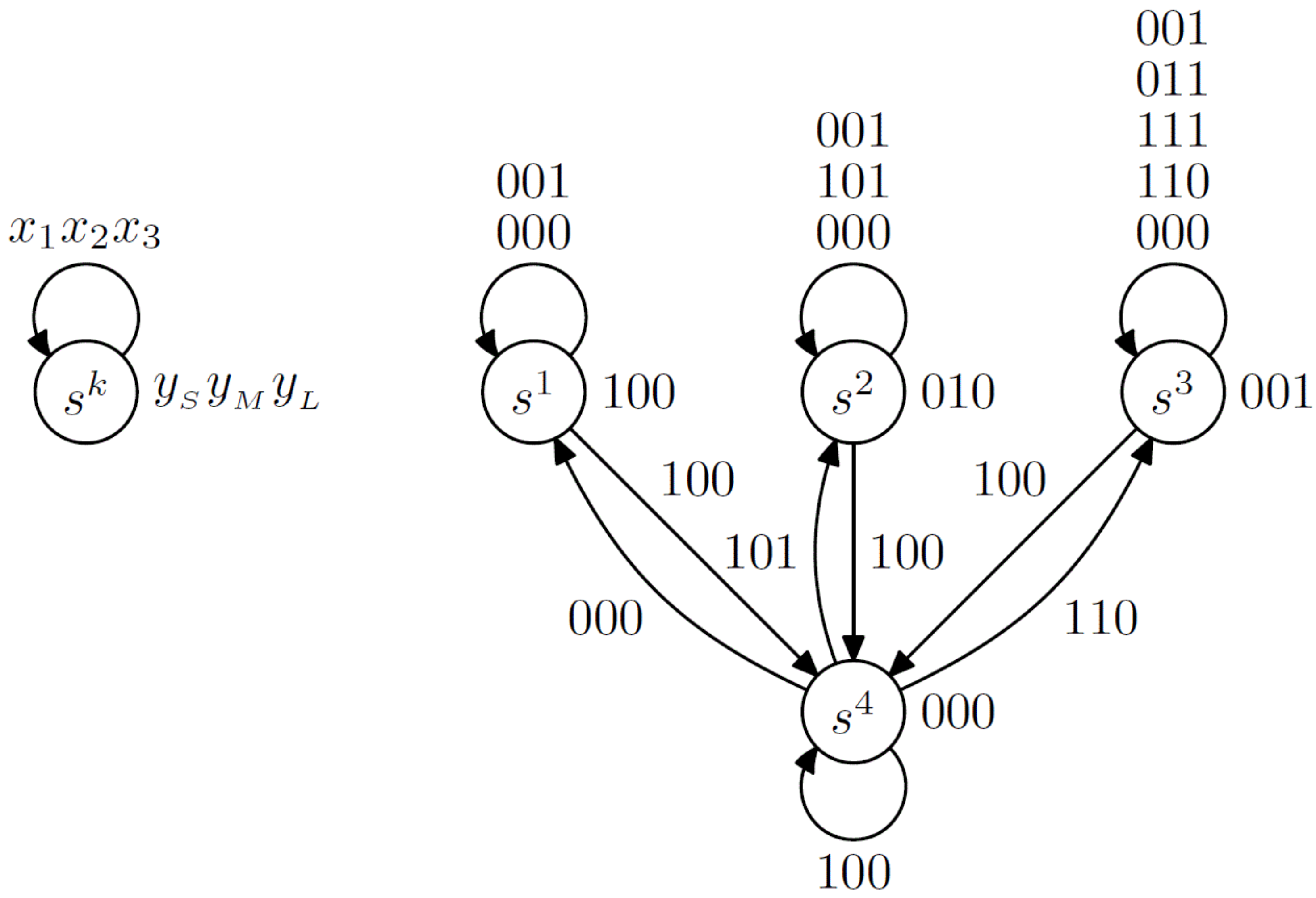
The case of a small parcell *S*



The case of a medium sized parcel *M*



The case of a large parcel *L*

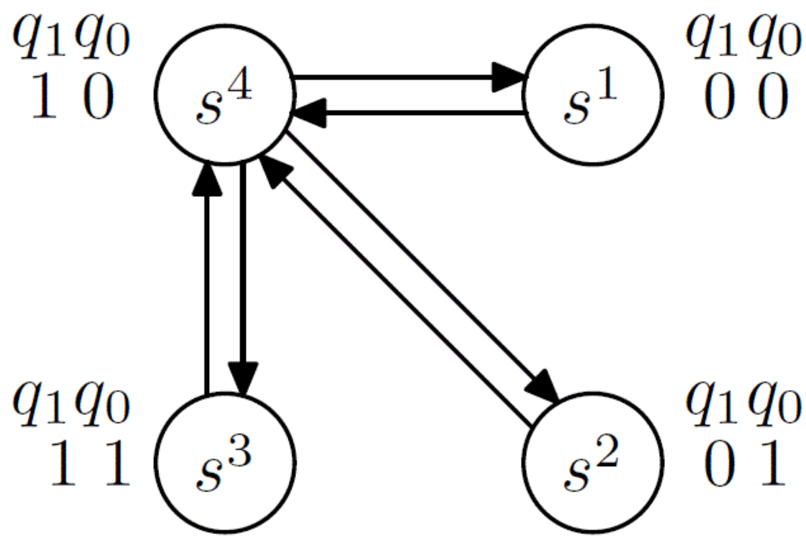


s \ $x_1x_2x_3$	000	001	011	010	110	111	101	100	y_S	y_M	y_L
s^1	s^1	s^1	–	–	–	–	–	s^4	1	0	0
s^2	s^2	s^2	–	–	–	–	s^2	s^4	0	1	0
s^3	s^3	s^3	s^3	–	s^3	s^3	–	s^4	0	0	1
s^4	s^1	–	–	–	s^3	–	s^2	s^4	0	0	0

s'/y

s \ $x_1x_2x_3$	000	001	011	010	110	111	101	100	y_S	y_M	y_L
s^1	s^1	s^1	–	–	–	–	s^2	s^4	1	0	0
s^2	s^2	s^2	–	–	–	–	s^2	s^4	0	1	0
s^3	s^3	s^3	s^3	–	s^3	s^3	s^2	s^4	0	0	1
s^4	s^1	–	–	–	s^3	–	s^2	s^4	0	0	0

s'/y



q_1q_0 \ $x_1x_2x_3$	000	001	011	010	110	111	101	100	y_S	y_M	y_L
00	00	00	–	–	–	–	01	10	1	0	0
01	01	01	–	–	–	–	01	10	0	1	0
11	11	11	11	–	11	11	01	10	0	0	1
10	00	–	–	–	11	–	01	10	0	0	0

$q_1'q_0'/y$

q_1q_0 \ $x_1x_2x_3$	000	001	011	010	110	111	101	100
00	0	0	–	–	–	–	0	1
01	0	0	–	–	–	–	0	1
11	1	1	1	–	1	1	0	1
10	0	–	–	–	1	–	0	1

q_1'

q_1q_0 \ $x_1x_2x_3$	000	001	011	010	110	111	101	100
00	0	0	–	–	–	–	1	0
01	1	1	–	–	–	–	1	0
11	1	1	1	–	1	1	1	0
10	0	–	–	–	1	–	1	0

q_0'

$q_1 \backslash q_0$	0	1
0	1	0
1	0	0

y_S

$q_1 \backslash q_0$	0	1
0	0	0
1	0	1

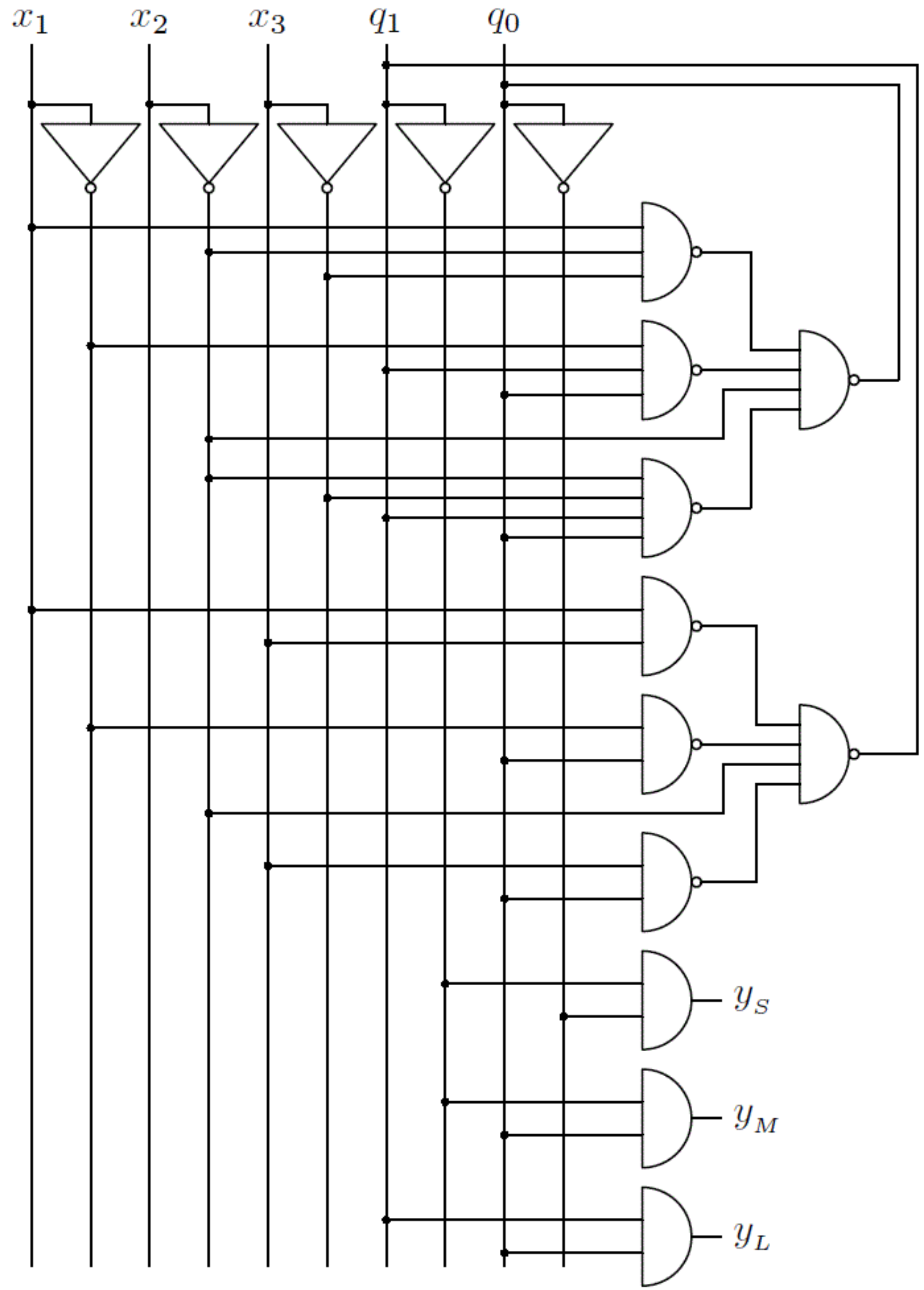
y_L

$q_1 \backslash q_0$	0	1
0	0	1
1	0	0

y_M

$$
\begin{cases}
q_1' &= x_2 + x_1\overline{x}_2\overline{x}_3 + \overline{x}_1 q_1 q_0 + \overline{x}_2\overline{x}_3 q_1 q_0 \\
q_0' &= x_2 + x_1 x_3 + \overline{x}_1 q_0 + x_3 q_0 \\
y_S &= \overline{q}_1\overline{q}_0 \\
y_M &= \overline{q}_1 q_0 \\
y_L &= q_1 q_0
\end{cases}
$$

WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME



WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

HUMAN CAPITAL
HUMAN – BEST INVESTMENT!

EUROPEAN UNION
EUROPEAN
SOCIAL FUND

Project is co-financed by European Union within European Social Fund