Minimization

EECS 20

Lecture 13 (February 14, 2001)

Tom Henzinger

Equivalence between state machines M1 and M2:

for every input signal, M1 and M2 produce the same output signal.

Bisimulation between M1 and M2:

the initial states of M1 and M2 are related, and for all related states p of M1 and q of M2,

for every input value, p and q produce the same output value, and the next states are again related.

Theorem: two state machines M1 and M2 are equivalent iff there is a bisimulation between M1 and M2.

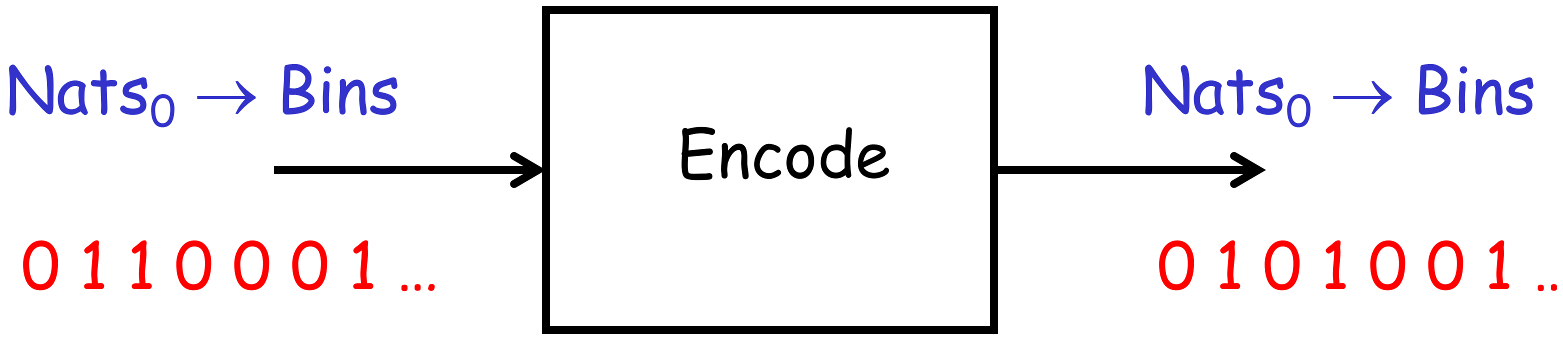
How do we find a bisimulation ?

Edge Encoder

$$\text{Encode} : [\ \text{Nats}_0 \to \text{Bins}\ ] \to [\ \text{Nats}_0 \to \text{Bins}\ ]$$

such that $\forall\, x \in [\ \text{Nats}_0 \to \text{Bins}\ ],\ \forall\, y \in \text{Nats}_0,$

$$(\text{Encode}(x))(y) = \begin{cases} x(y) & \text{if } y = 0 \\ 0 & \text{if } y > 0 \text{ and } x(y) = x(y-1) \\ 1 & \text{if } y > 0 \text{ and } x(y) \neq x(y-1) \end{cases}$$



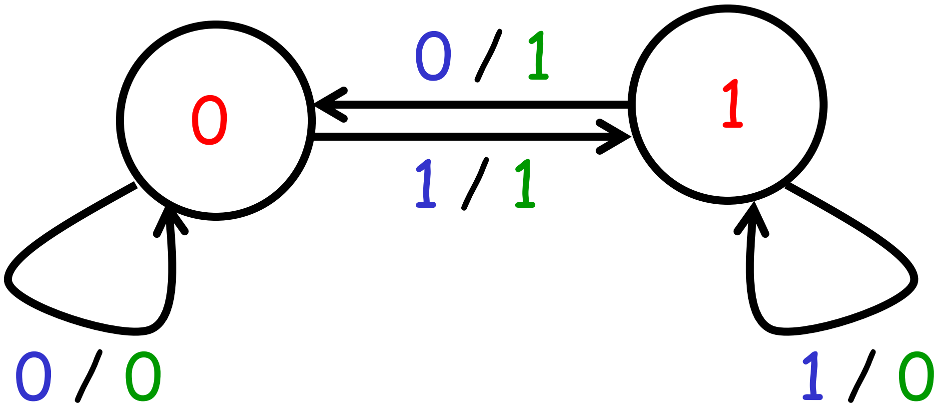
Nats0 → Bins
0 1 1 0 0 0 1 …
Encode
Nats0 → Bins
0 1 0 1 0 0 1 …

Edge Encoder

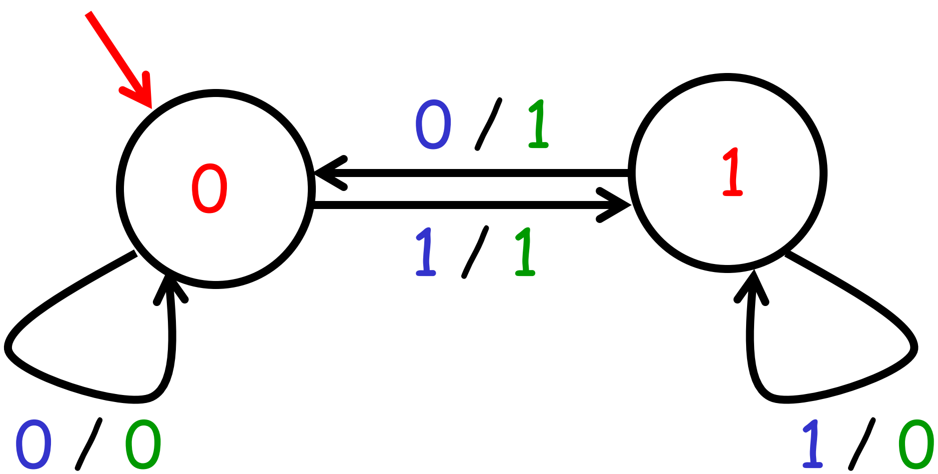
State between time t-1 and t:

0	if $t > 0$ and input at time $t-1$ was 0
1	if $t > 0$ and input at time $t-1$ was 1

Edge Encoder



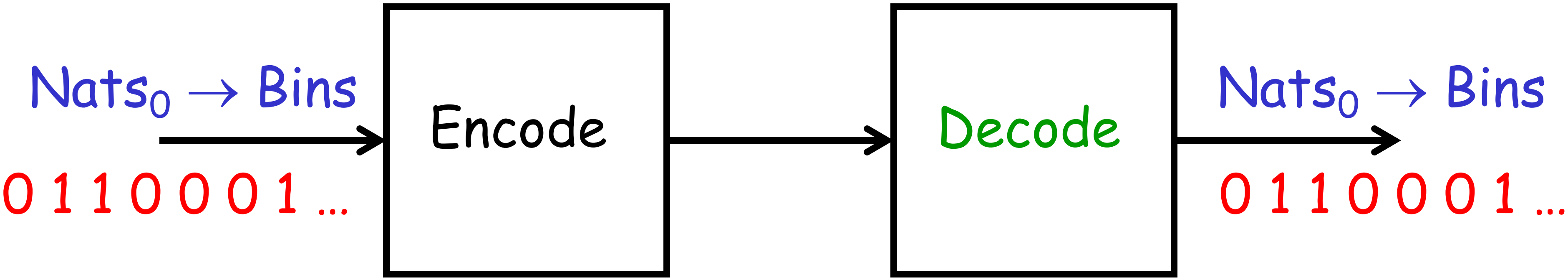
Edge Encoder



Edge Encoder

State between time t-1 and t:

0	if $t > 0$ and input at time $t-1$ was 0, or $t = 0$
1	if $t > 0$ and input at time $t-1$ was 1



$Nats_0 \to Bins$
0 1 1 0 0 0 1 ...
Encode
Decode
$Nats_0 \to Bins$
0 1 1 0 0 0 1 ...

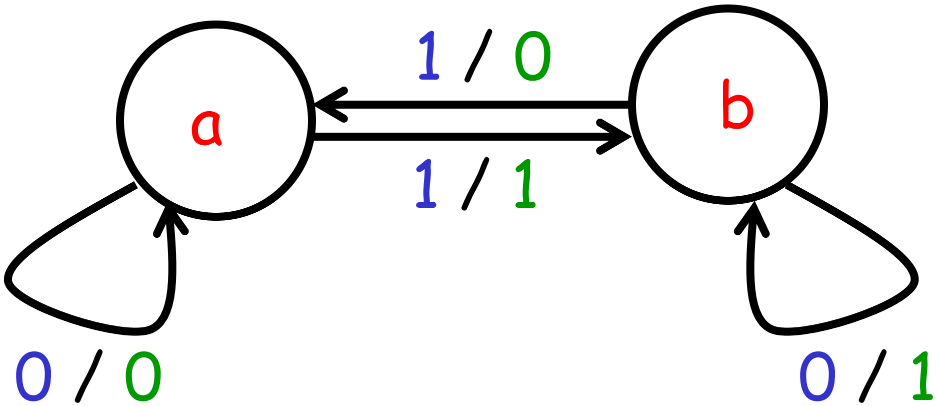
Decoder

State between time t-1 and t:

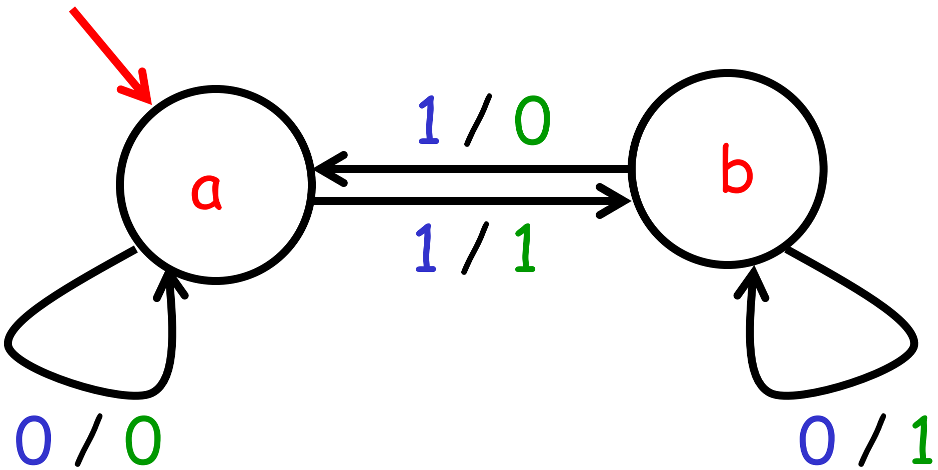
a if $t > 0$ and output at time t-1 was 0

b if $t > 0$ and output at time t-1 was 1

Decoder



Decoder



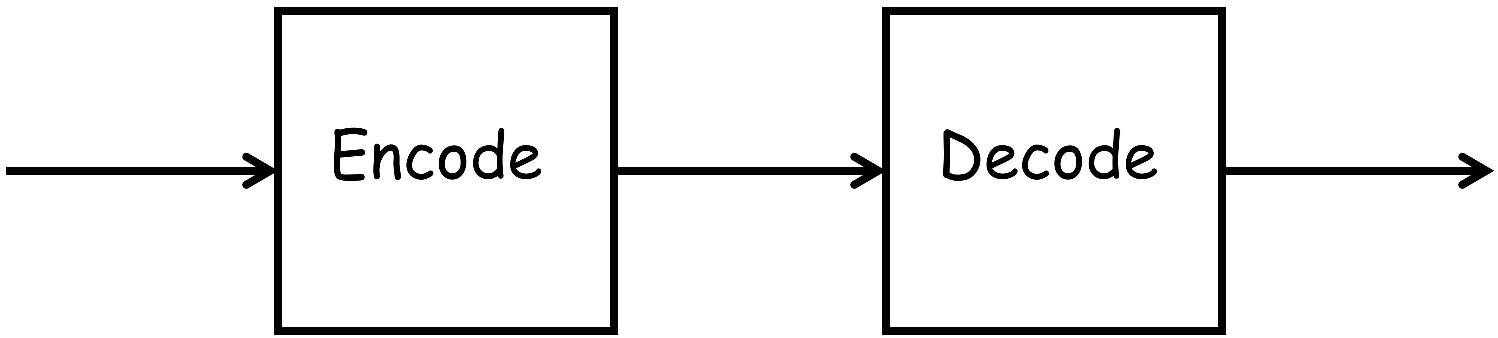
Decoder

State between time t-1 and t:

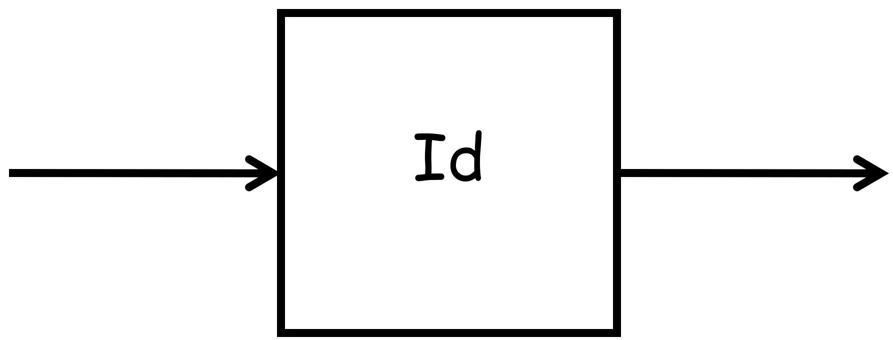
a if $t > 0$ and output at time t-1 was 0, or $t = 0$

b if $t > 0$ and output at time t-1 was 1

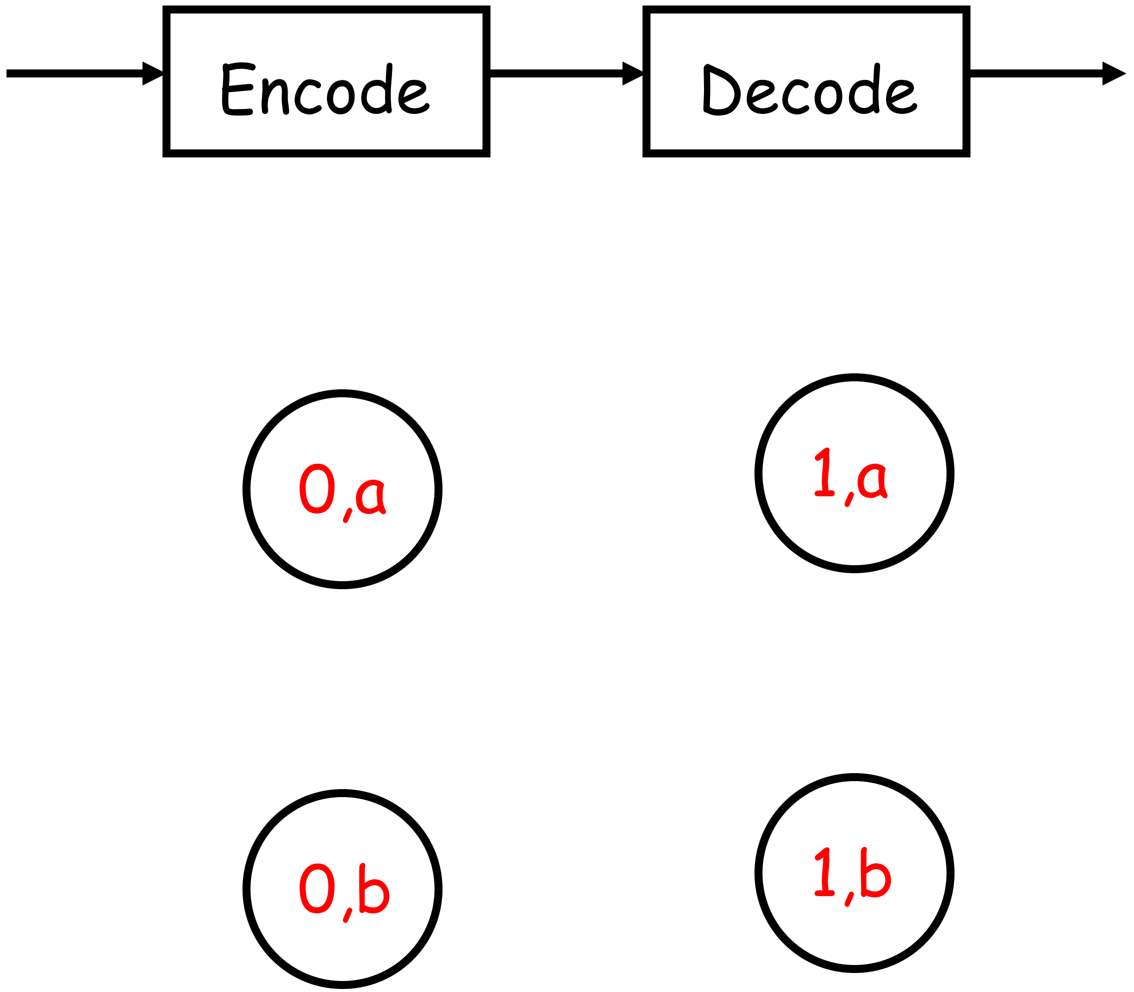
4 states

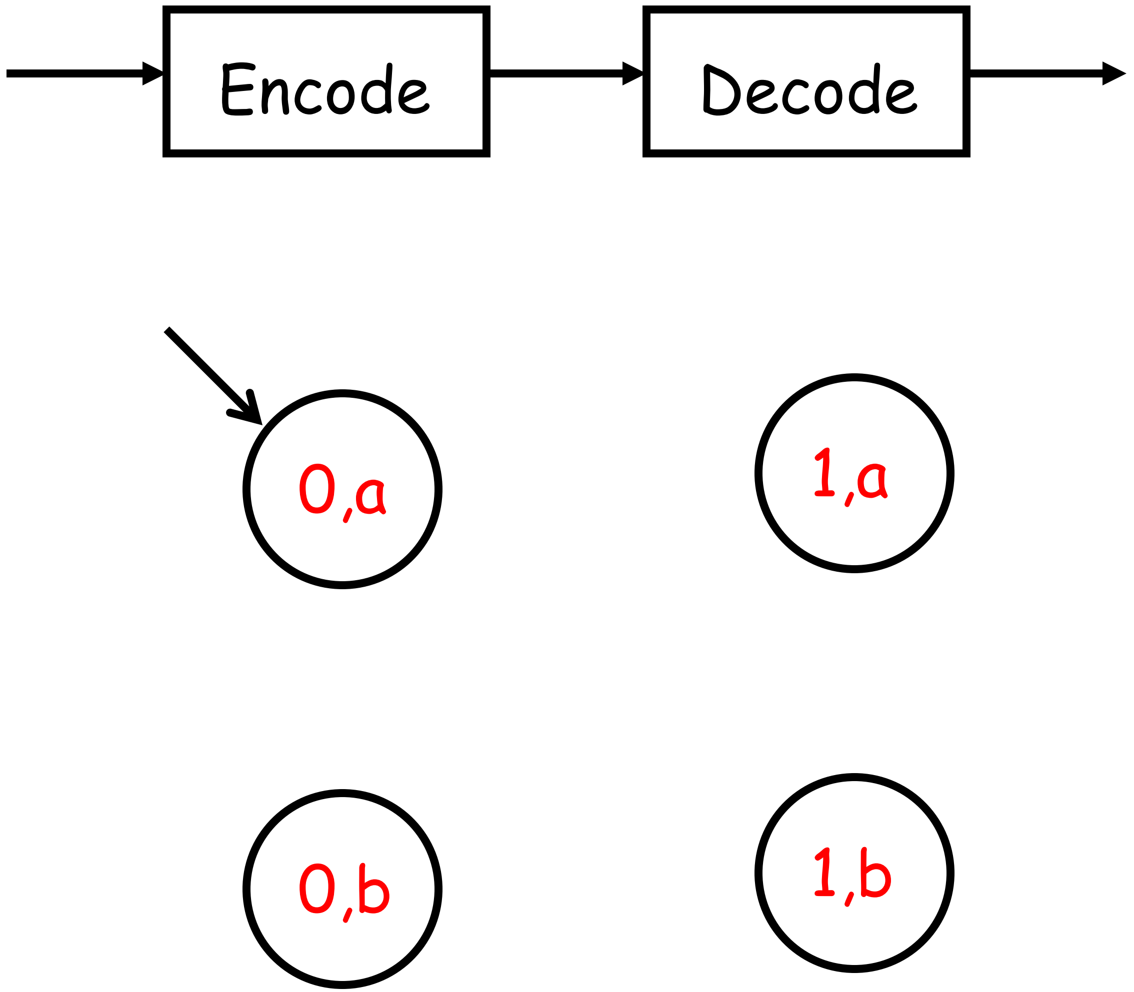


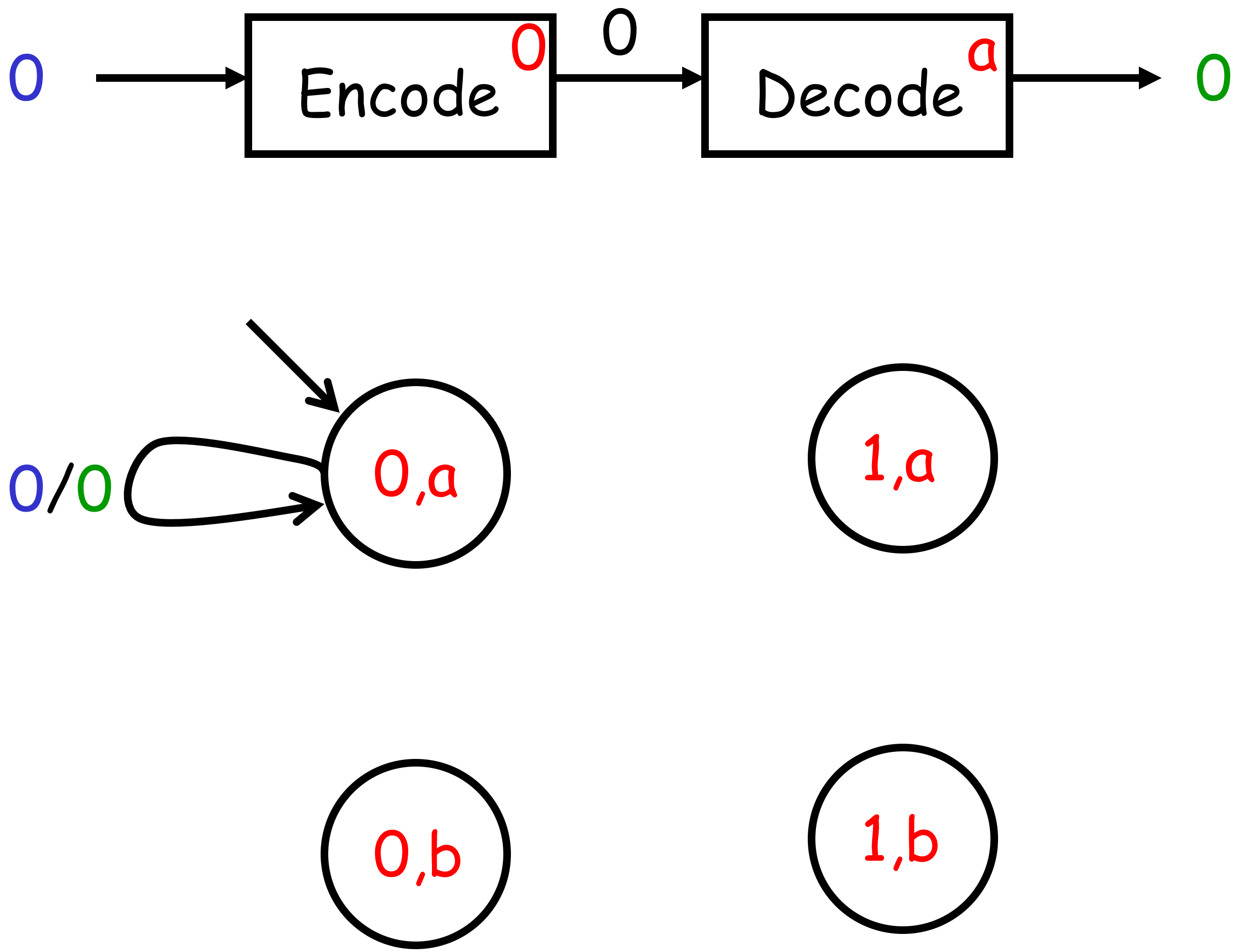
should be equivalent to

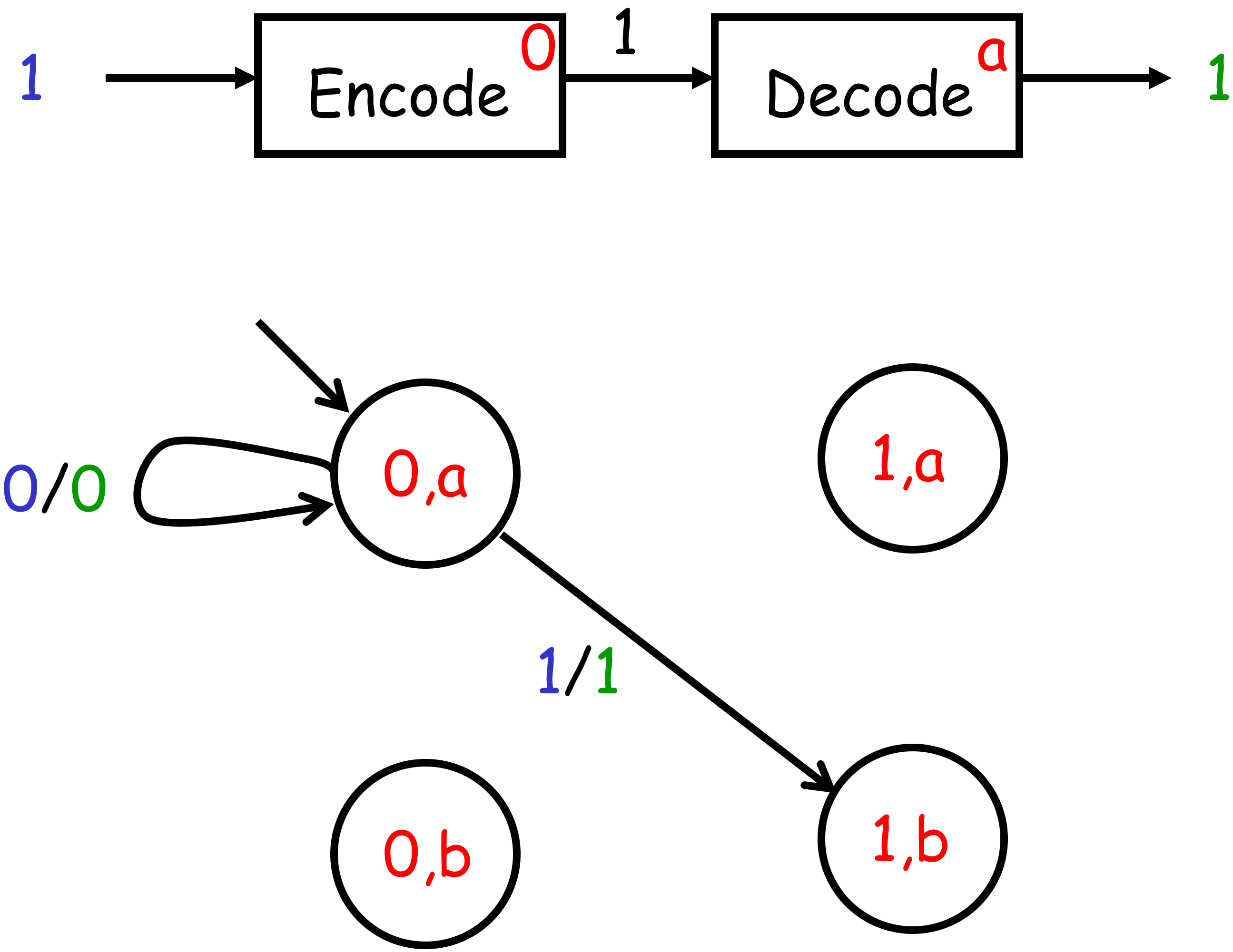


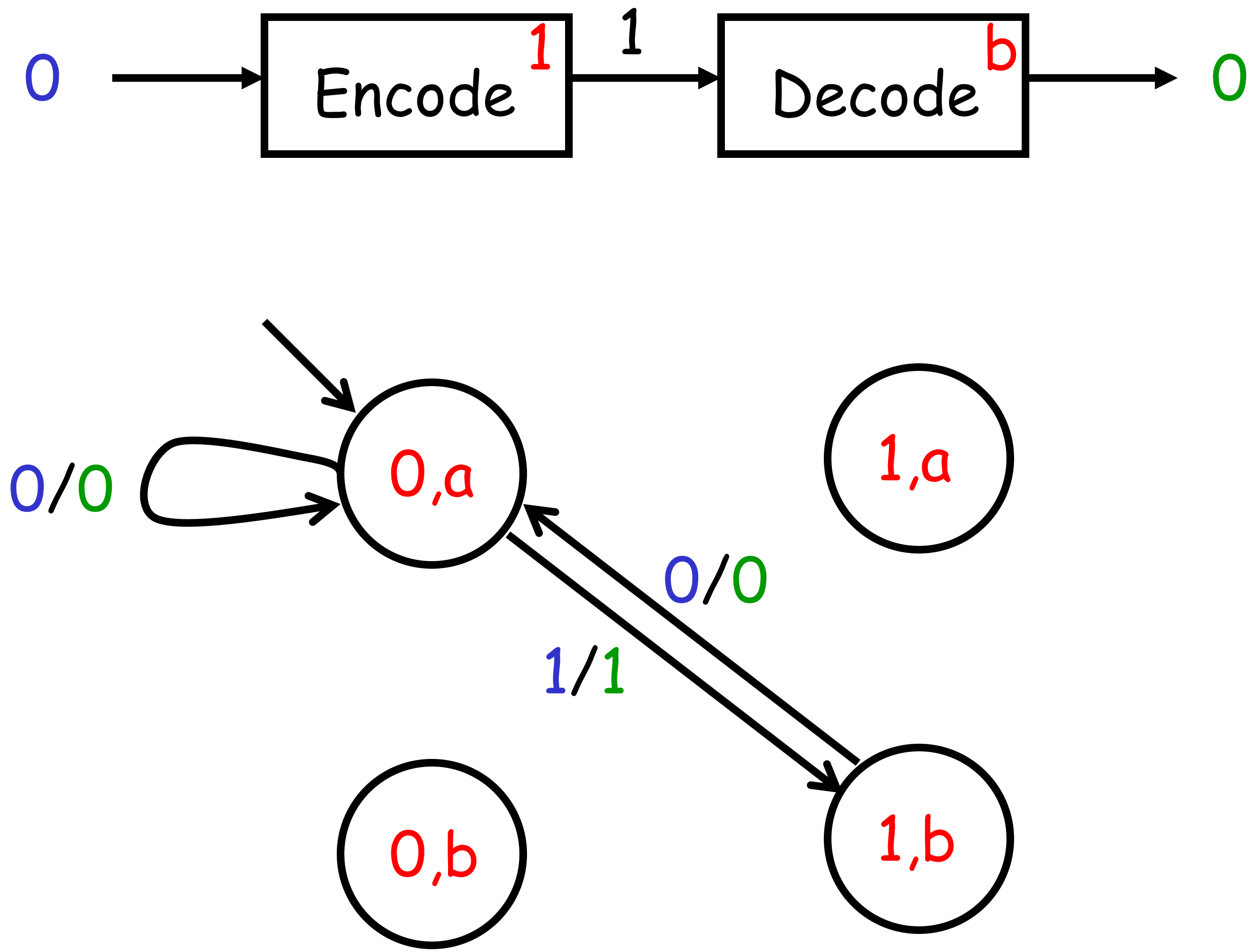
1 state (memory-free)

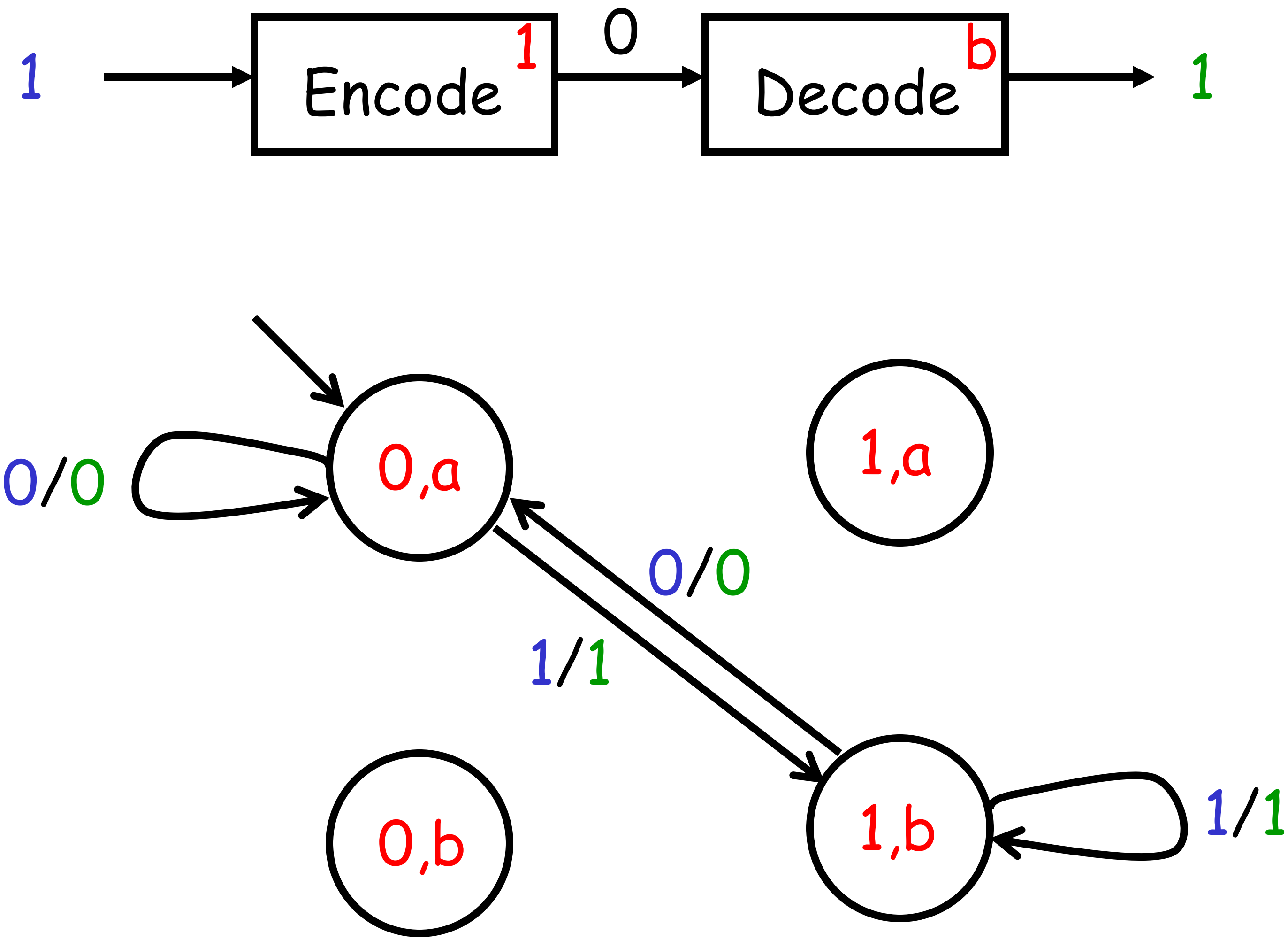

Encode
Decode
0,a
1,a
0,b
1,b


Encode
Decode
0,a
1,a
0,b
1,b

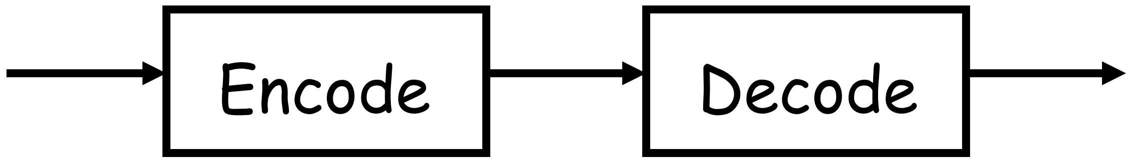

0
Encode
0
0
Decode
a
0
0/0
0,a
1,a
0,b
1,b

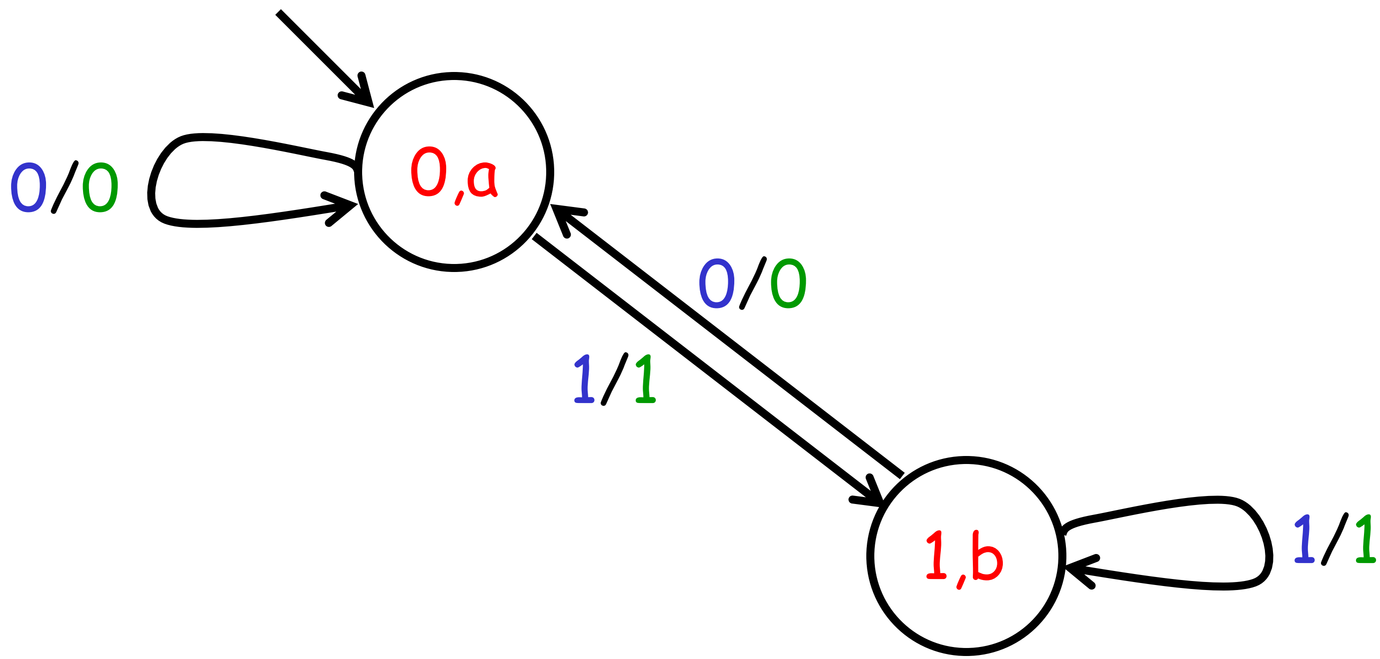

1
Encode
0
1
Decode
a
1
0/0
0,a
1,a
1/1
0,b
1,b


0
Encode
1
1
Decode
b
0
0/0
0,a
1,a
0/0
1/1
0,b
1,b

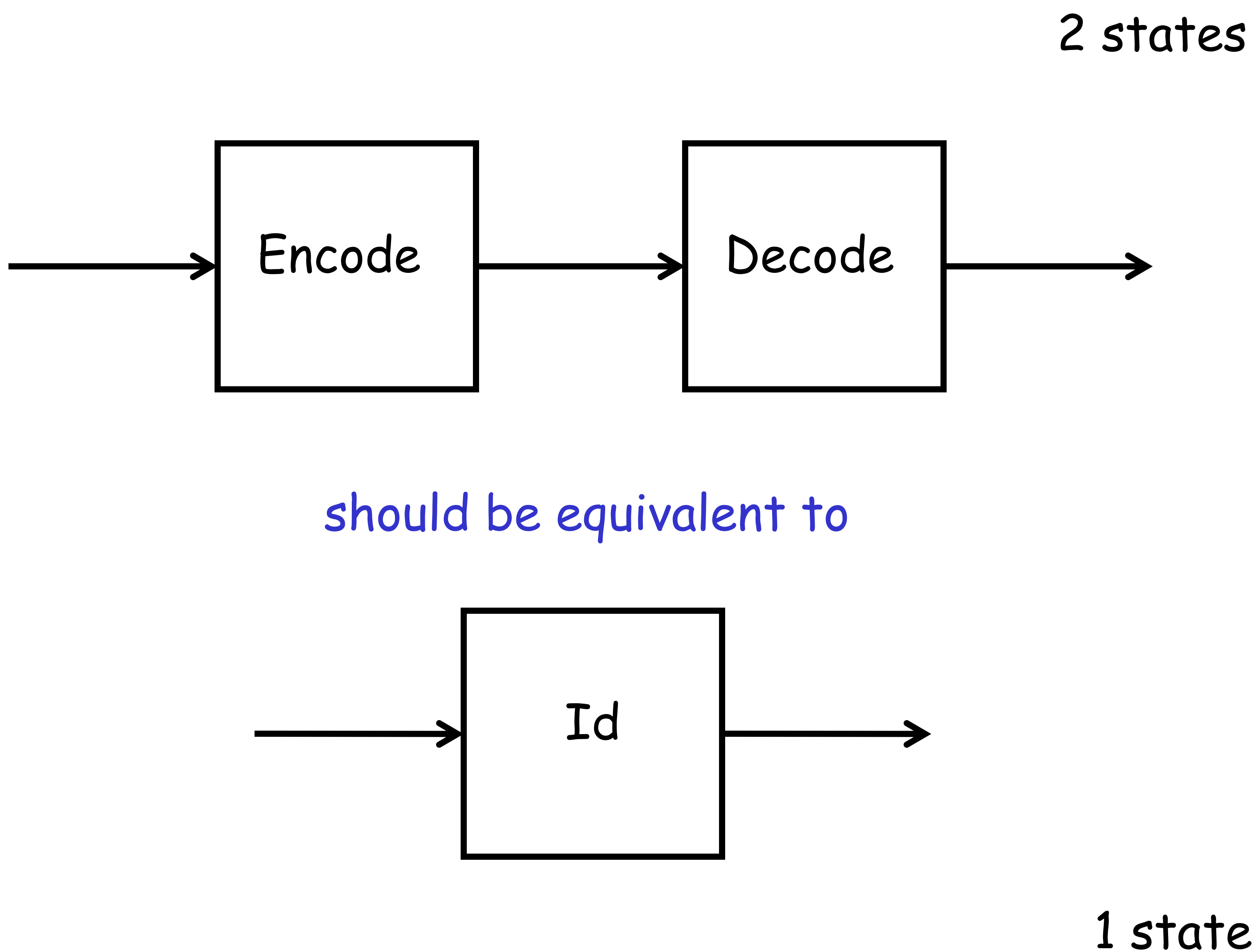


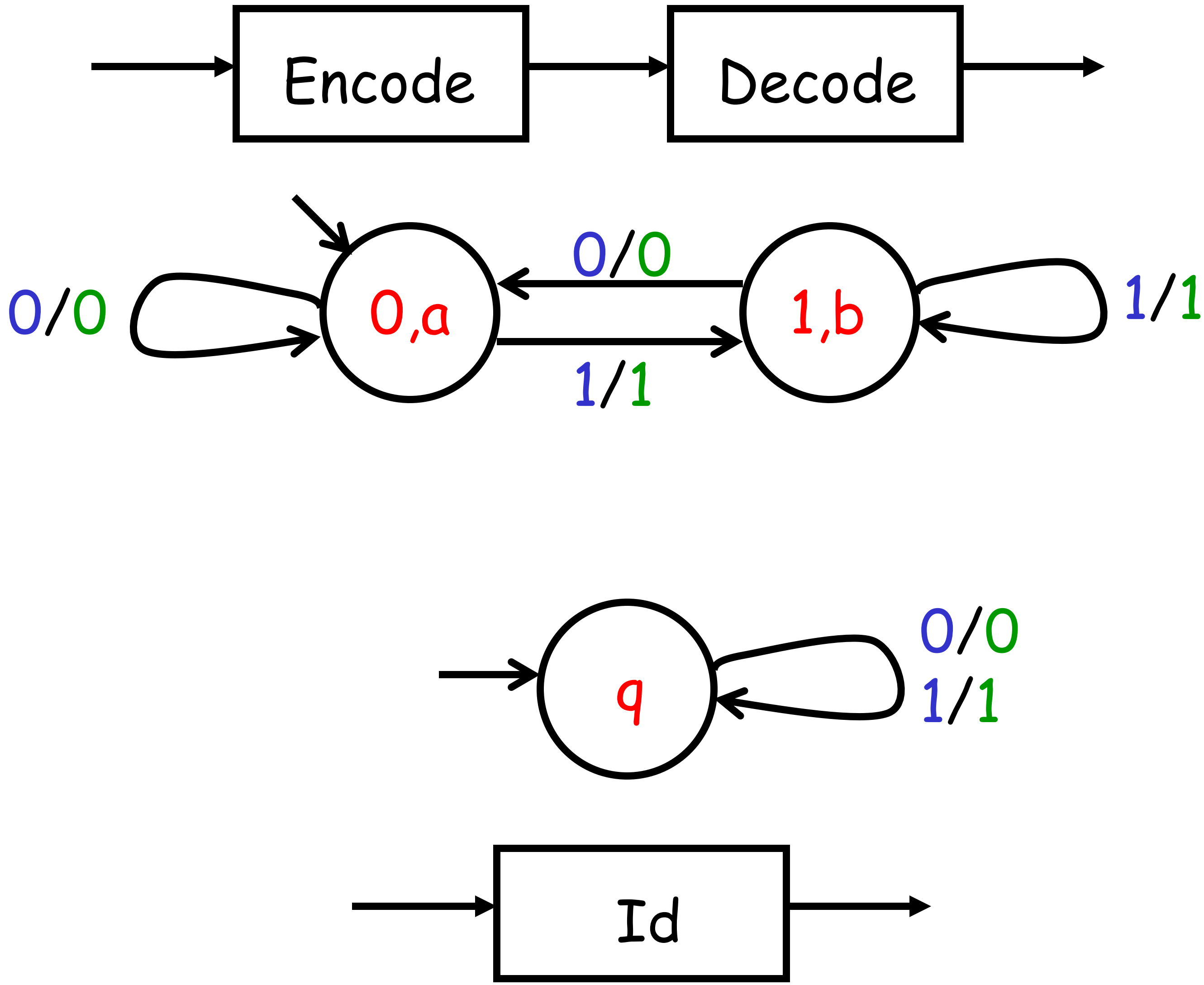
1
Encode
1
0
Decode
b
1
0/0
0,a
1,a
0/0
1/1
0,b
1,b
1/1



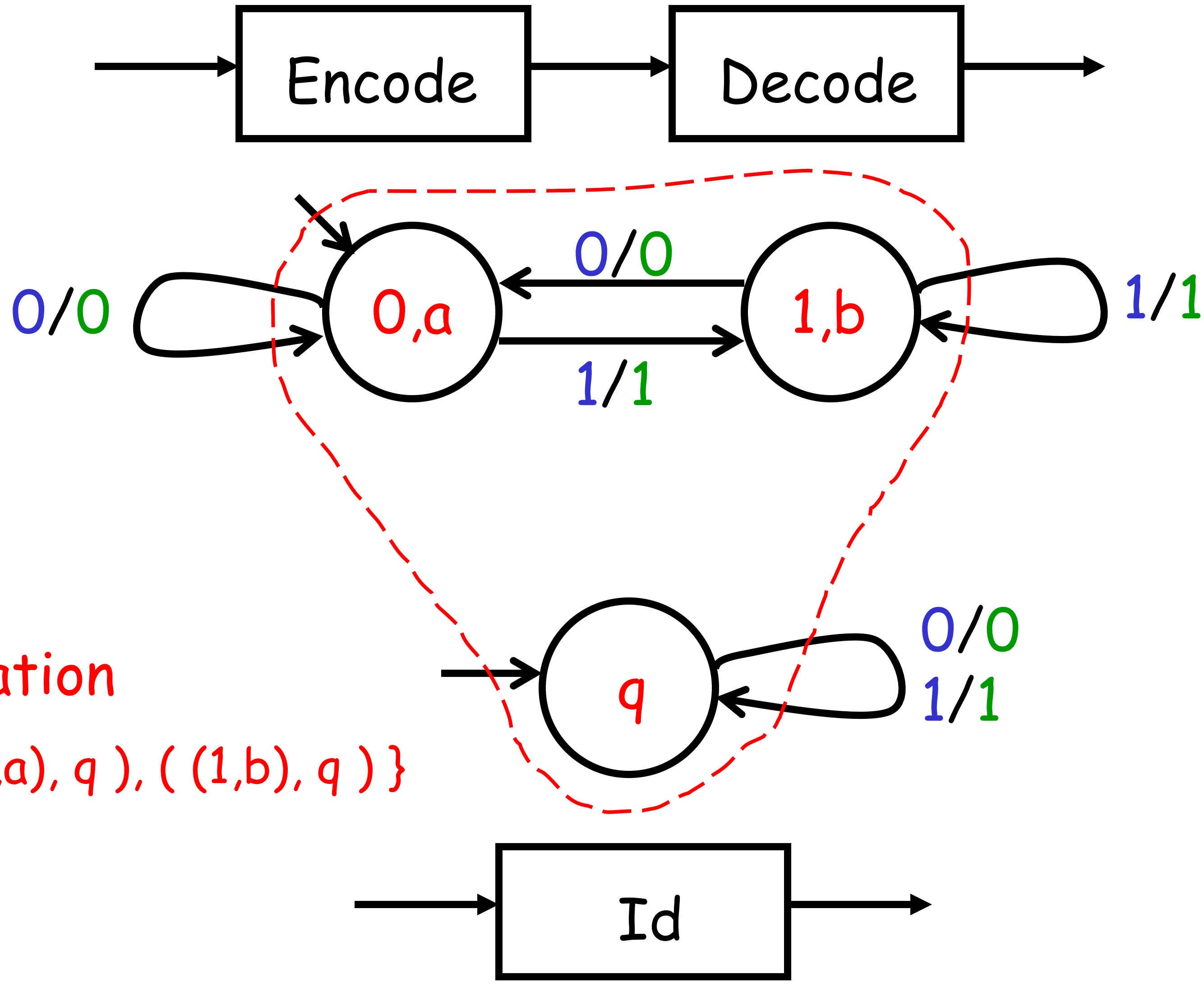


Remove unreachable states


2 states
Encode
Decode
should be equivalent to
Id
1 state



Encode
Decode
0/0
0,a
0/0
1,b
1/1
1/1
q
0/0
1/1
Id



Bisimulation

$B = \{ \, (\, (0,a), \, q \,), \, (\, (1,b), \, q \,) \, \}$

The Minimization Algorithm

Input : state machine M

Output : minimize (M), the state machine with the fewest states that is bisimilar to M

(the result is unique up to renaming of states)

If minimize (M) = N , then:

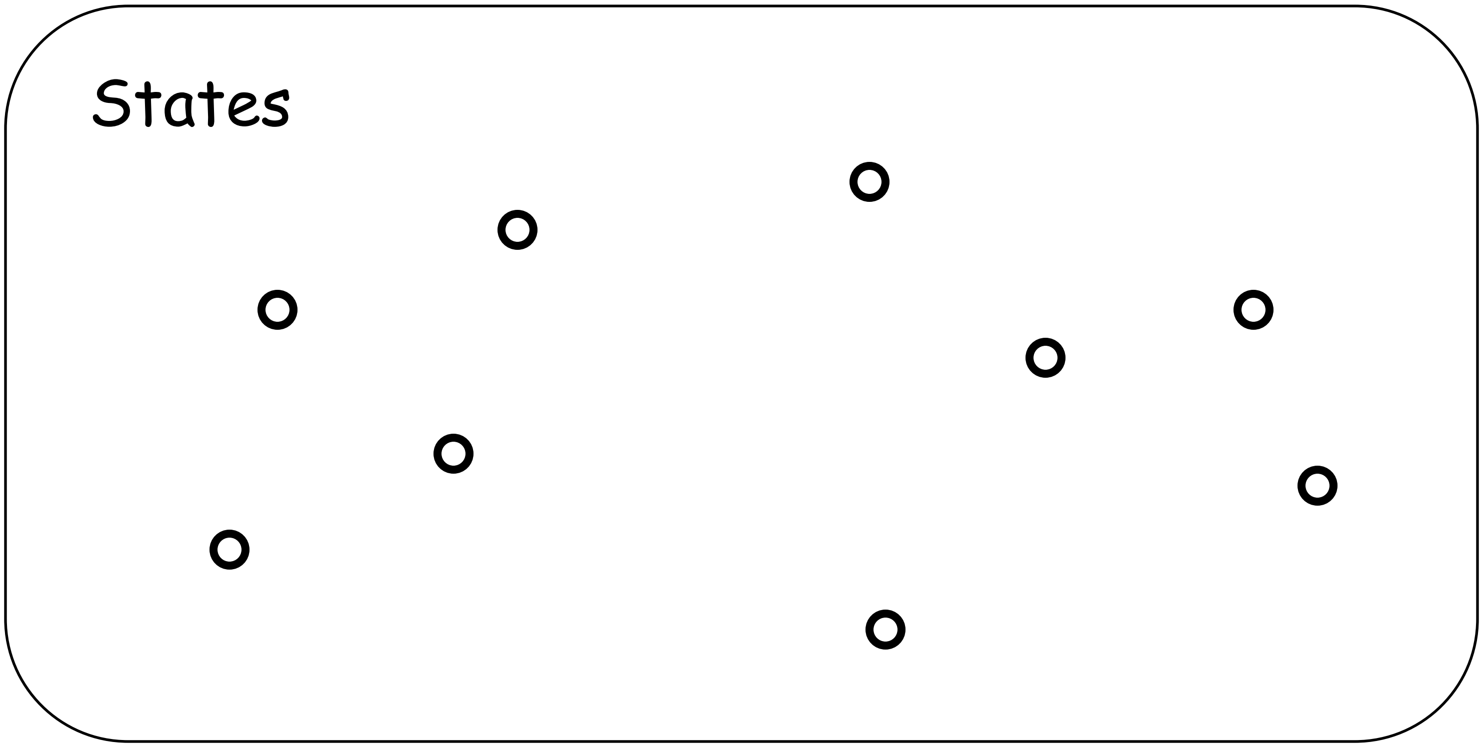
1. M and N are bisimilar
   (i.e., there is a bisimulation between M and N).

2. For every state machine N' that is bisimilar to M:

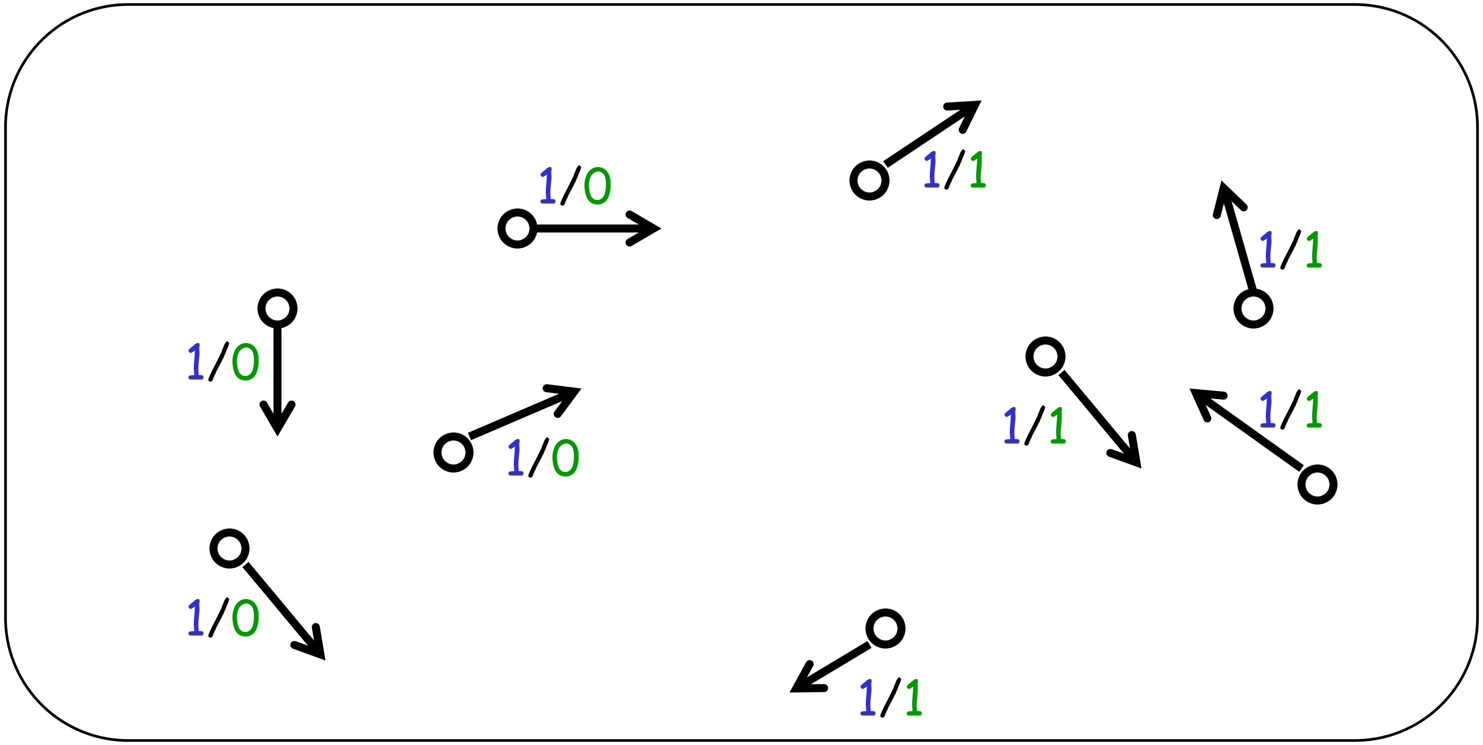
   2a. N' has at least as many states as N.

   2b. If N' has the same number of states as N,
   then N' and N differ only in the names of states.

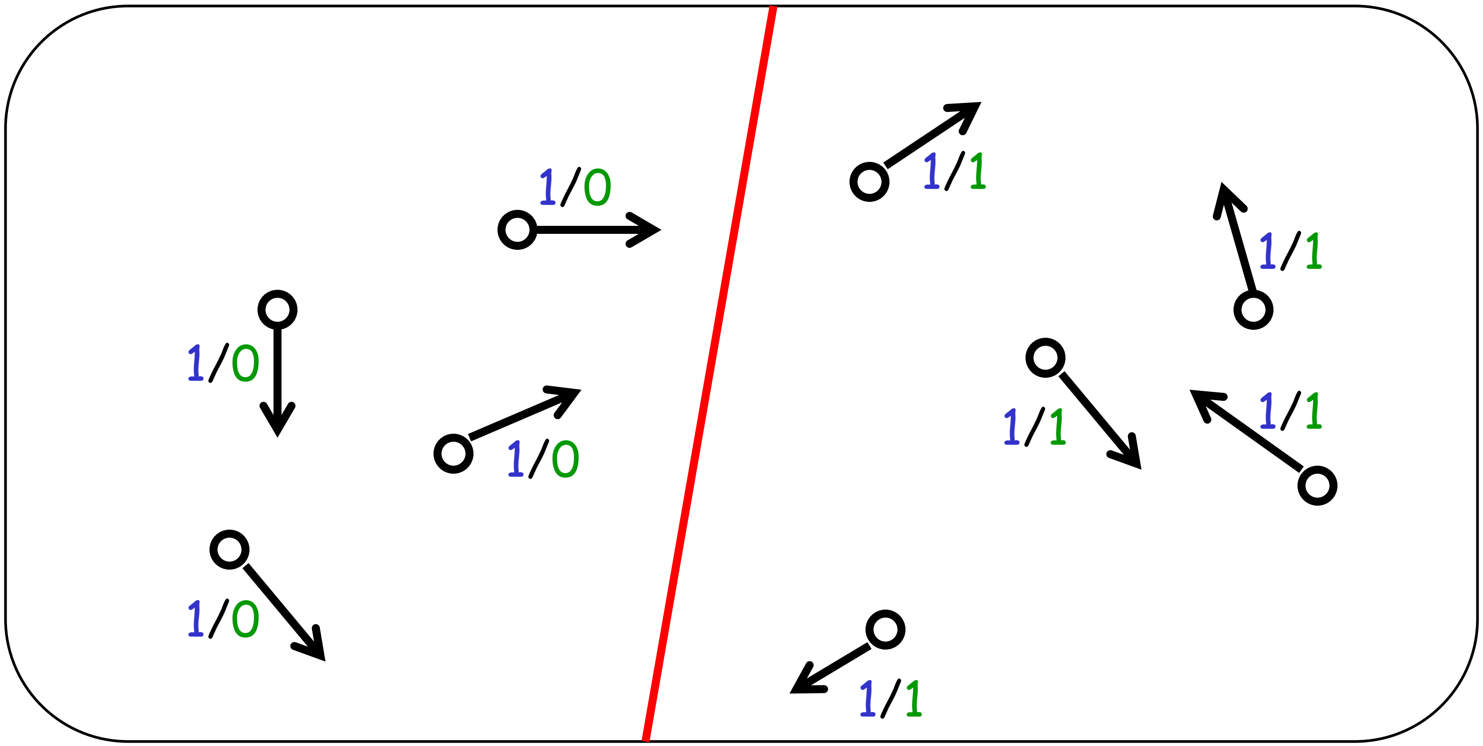
The Minimization Algorithm



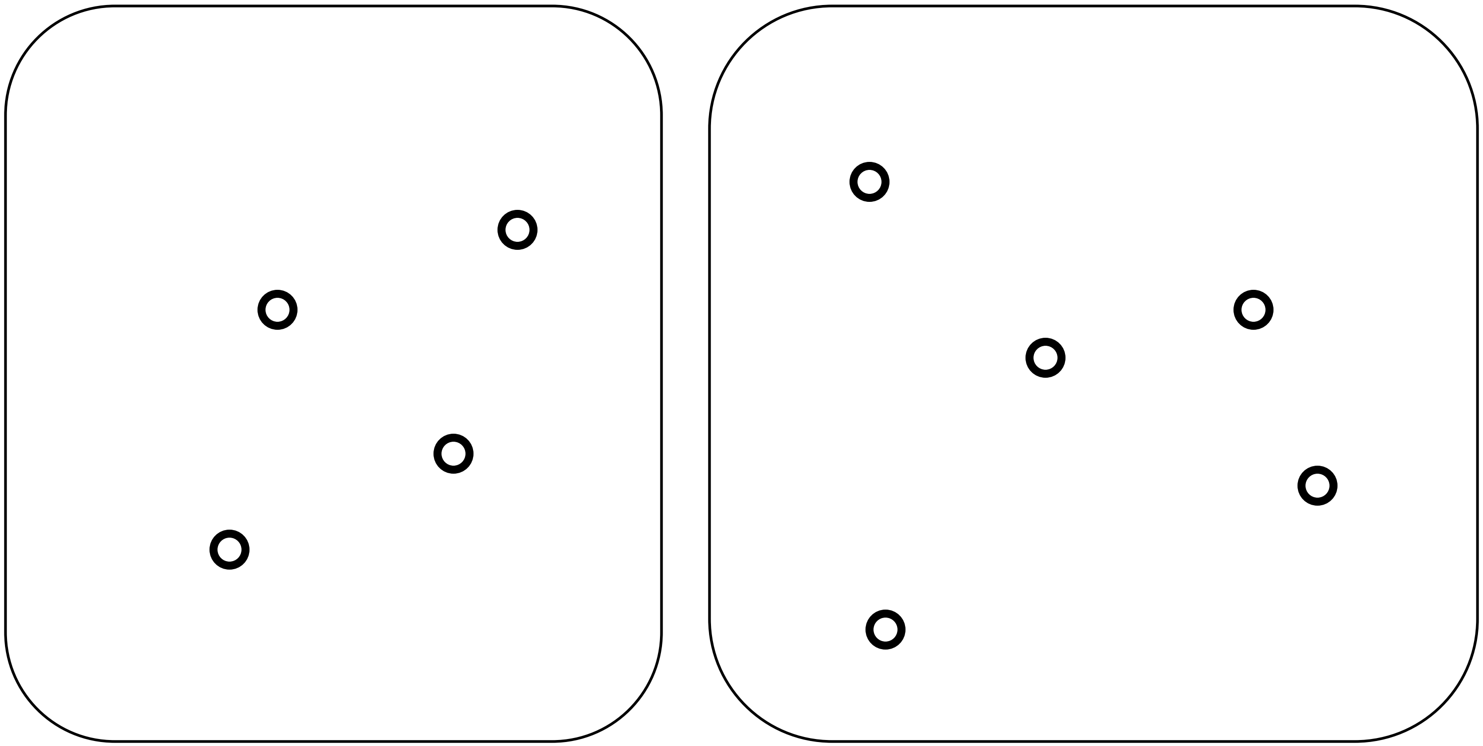
The Minimization Algorithm



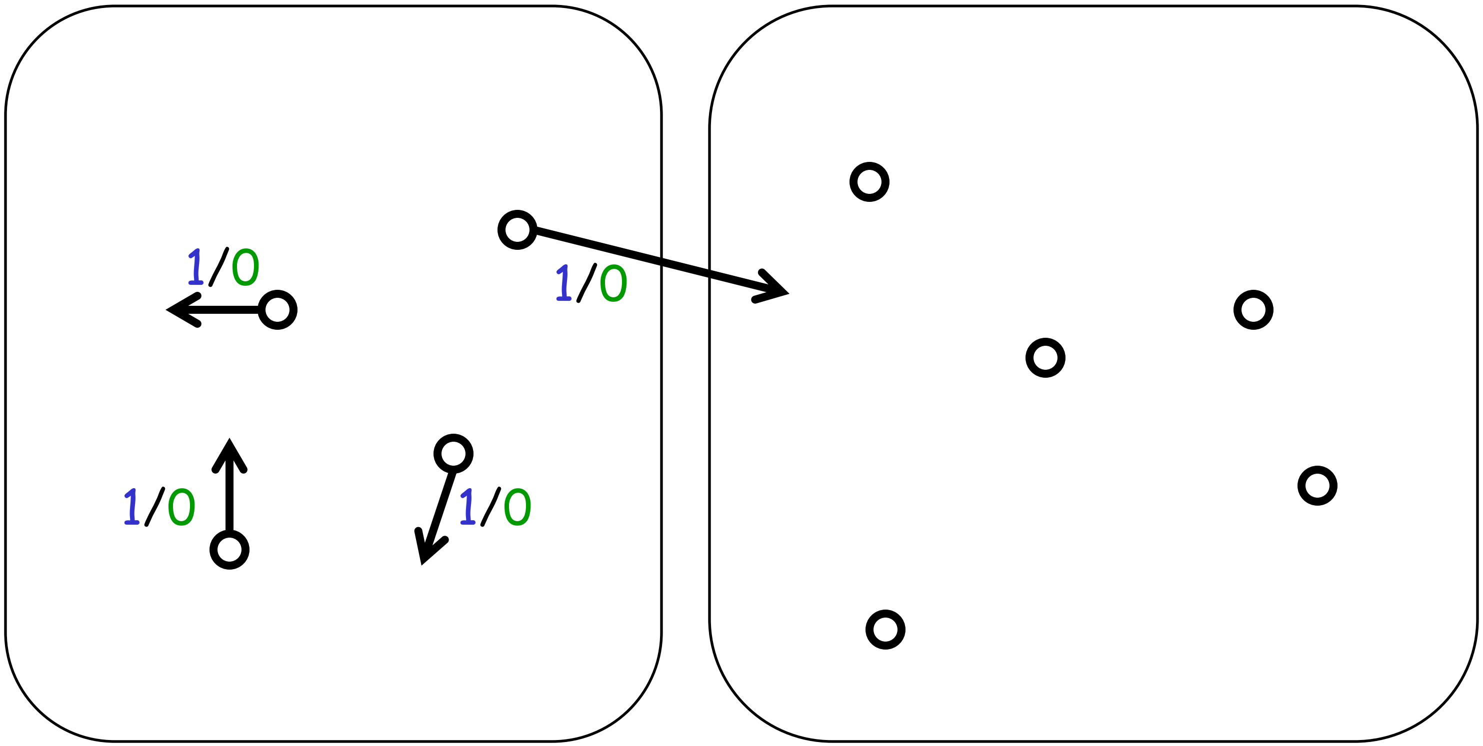
The Minimization Algorithm



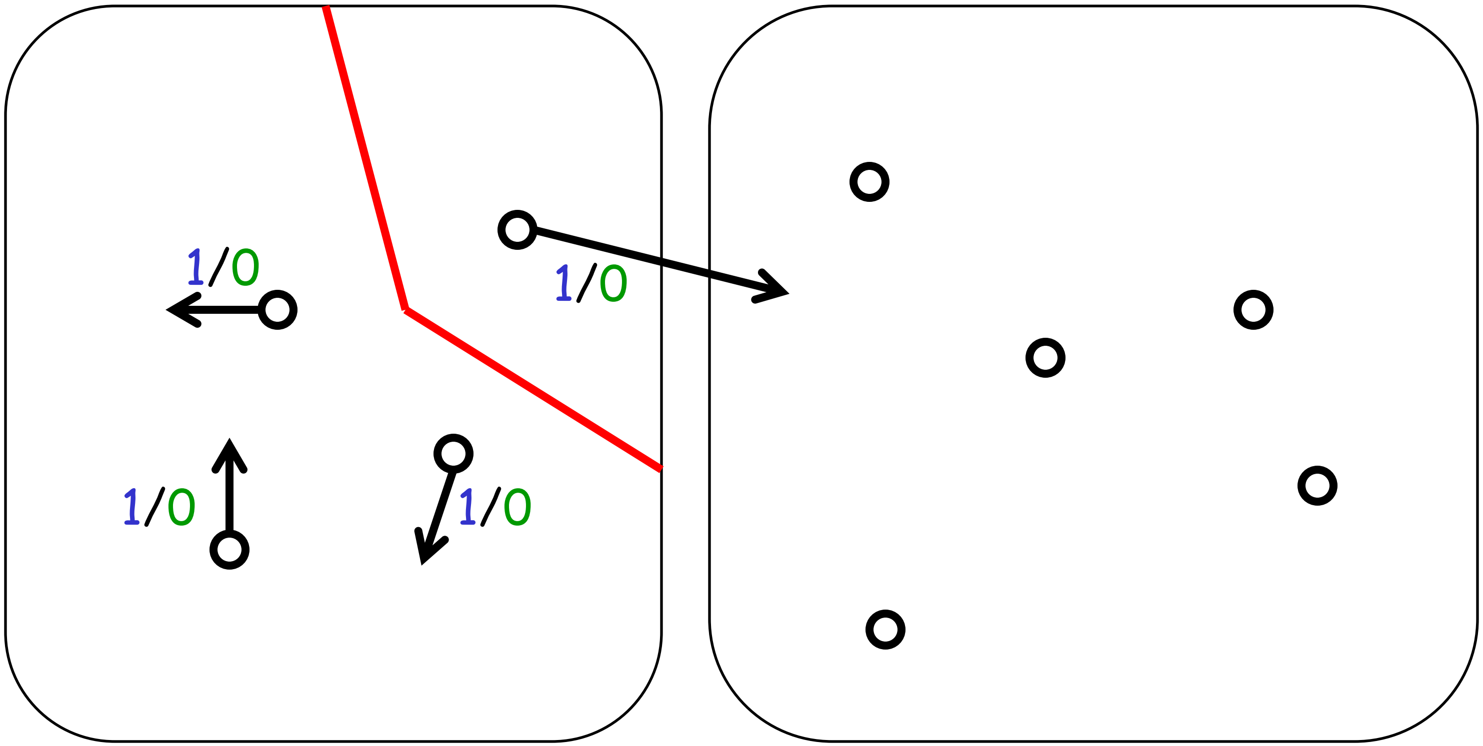
The Minimization Algorithm



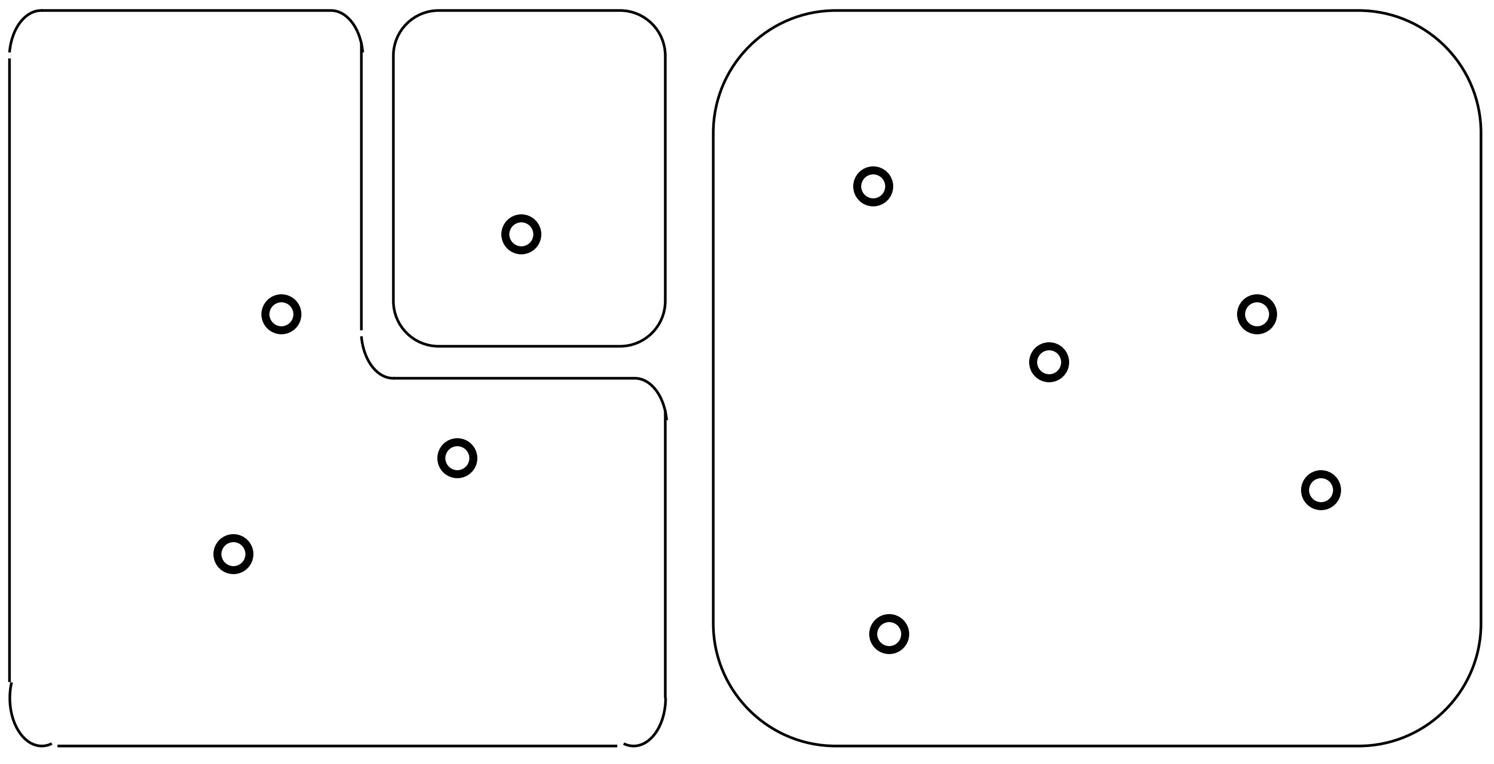
The Minimization Algorithm



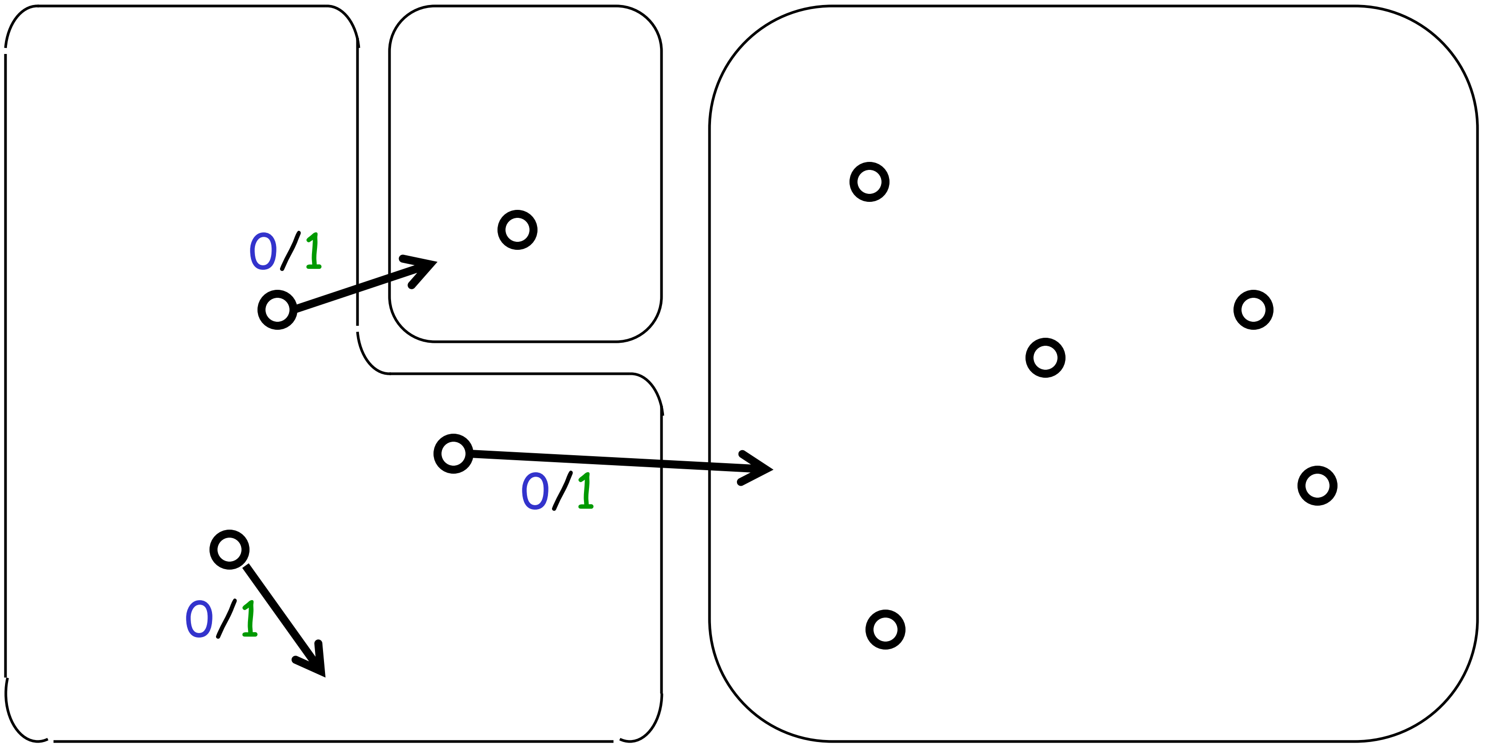
The Minimization Algorithm



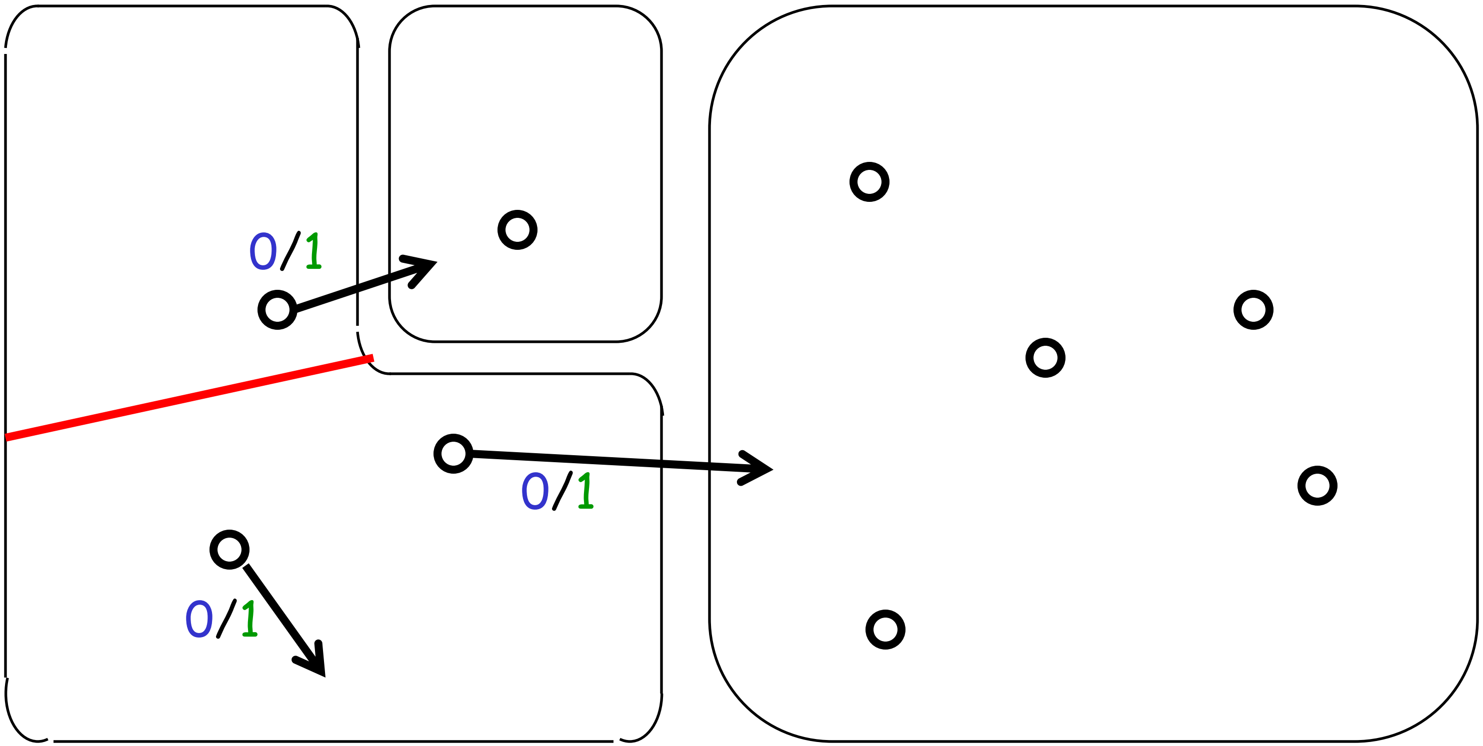
The Minimization Algorithm



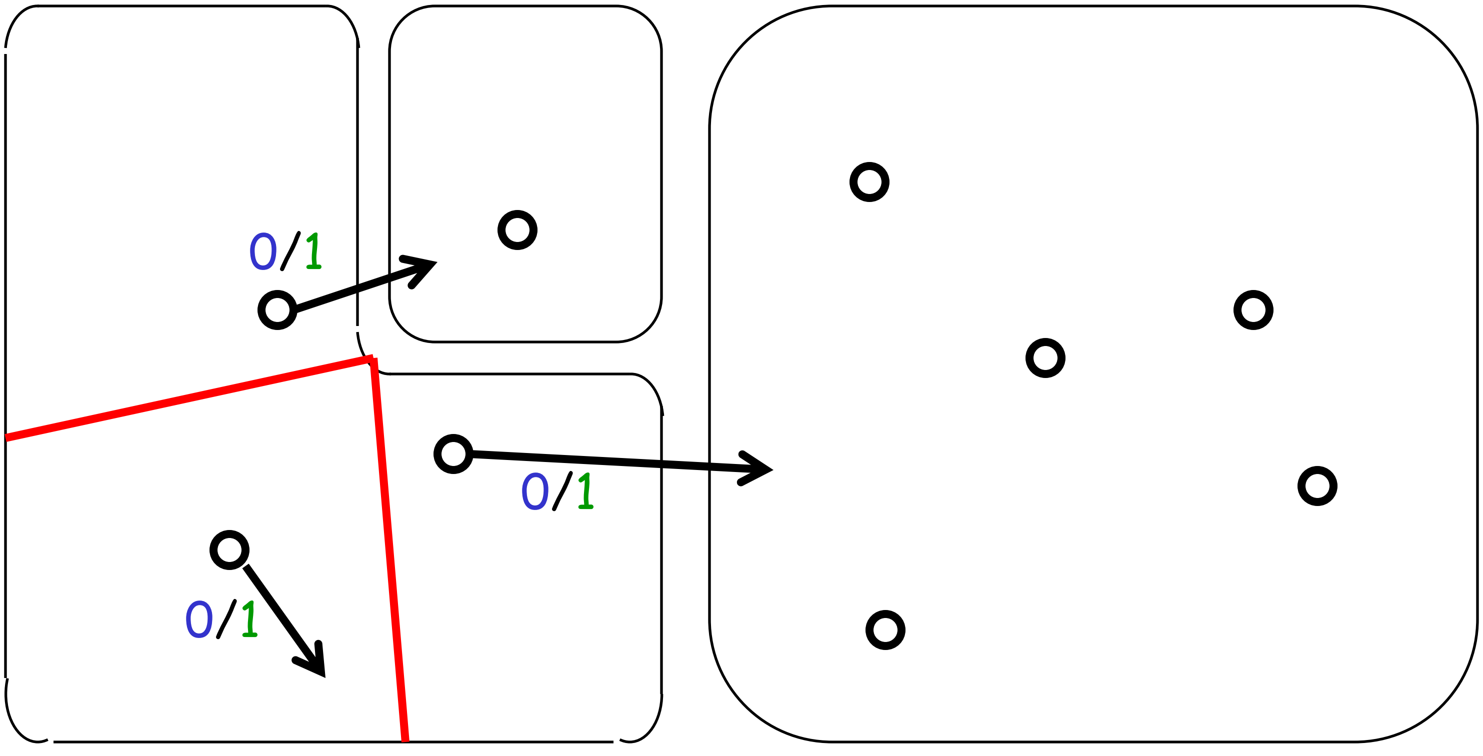
The Minimization Algorithm



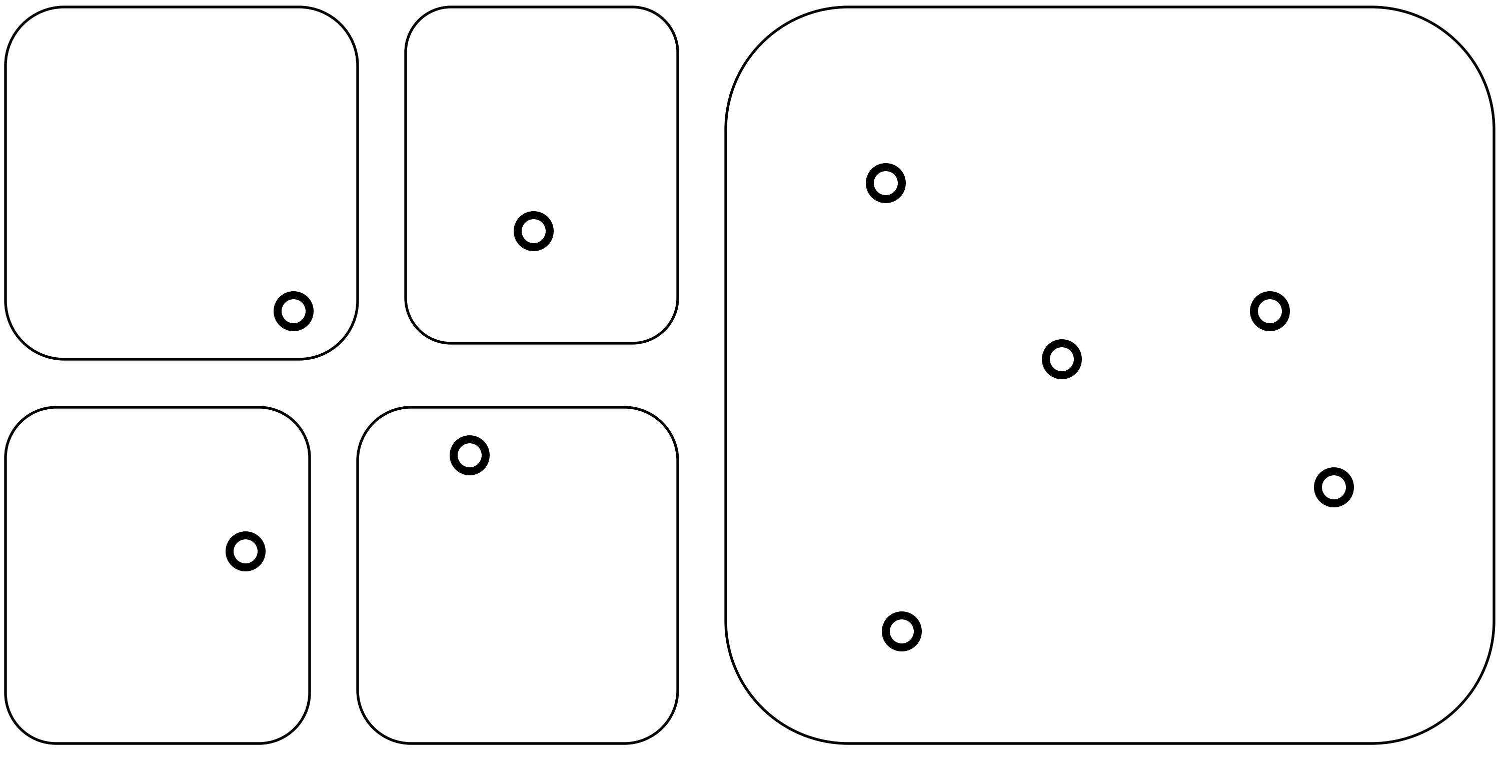
The Minimization Algorithm



The Minimization Algorithm



The Minimization Algorithm



The Minimization Algorithm

1. Let Q be set of all reachable states of M.
2. Maintain a set P of state sets:

   Initially let P = { Q }.

   2a. Repeat until no longer possible: output split P.

   2b. Repeat until no longer possible: next-state split P.
3. When done, every state set in P represents a single state of the smallest state machine bisimilar to M.

Output split P

If there exist

$\quad$ a state set $R \in P$

$\quad$ two states $r1 \in R$ and $r2 \in R$

$\quad$ an input $x \in Inputs$

such that

$$output\ (r1, x) \neq output\ (r2, x)$$

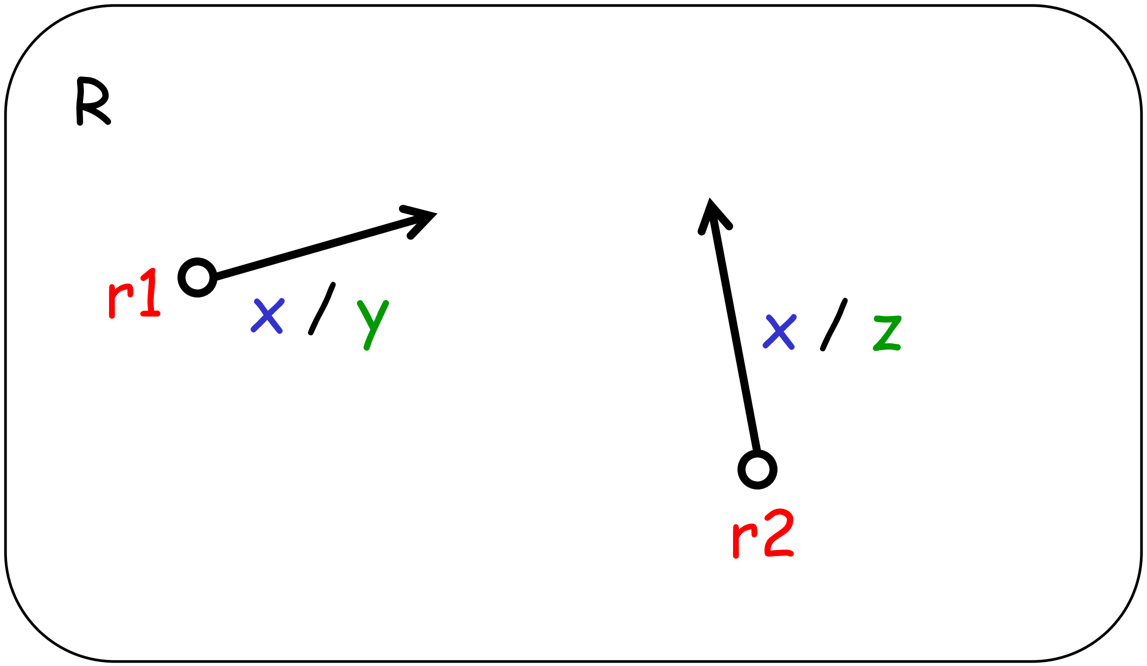
then

$\quad$ let $R1 = \{ r \in R \mid output\ (r,x) = output\ (r1,x) \}$;

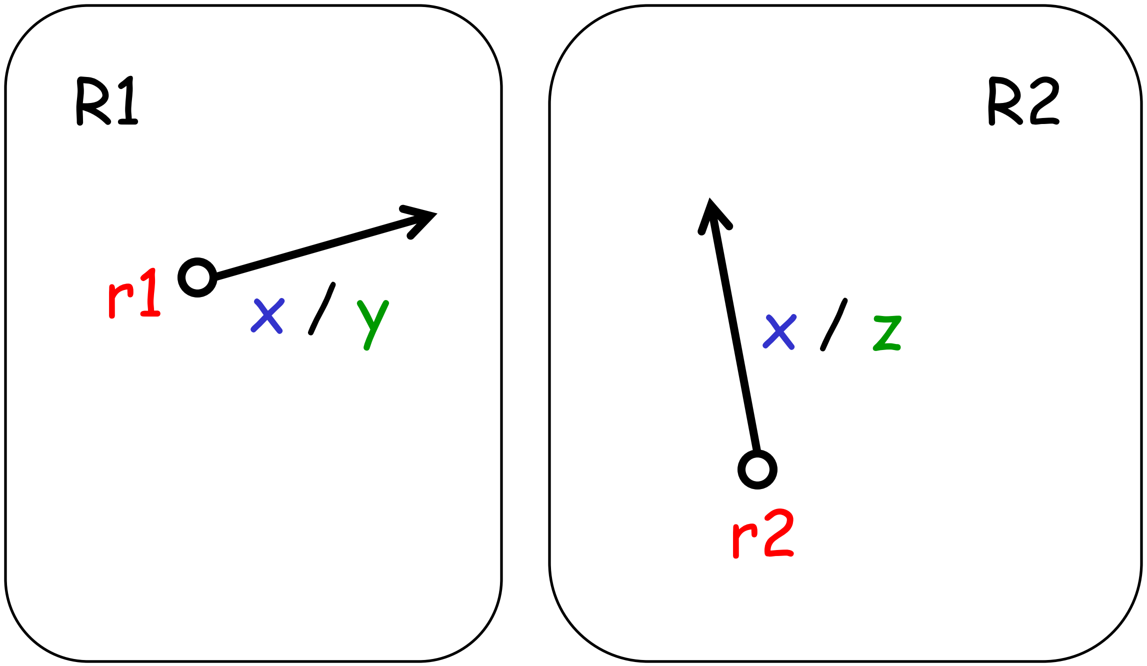
$\quad$ let $R2 = R \setminus R1$;

$\quad$ let $P = (P \setminus \{ R \}) \cup \{ R1, R2 \}$.

Output split



Output split



Next-state split P

If there exist

two state sets $R \in P$ and $R' \in P$

two states $r1 \in R$ and $r2 \in R$

an input $x \in$ Inputs

such that

$\text{nextState}(r1, x) \in R'$ and $\text{nextState}(r2, x) \notin R'$

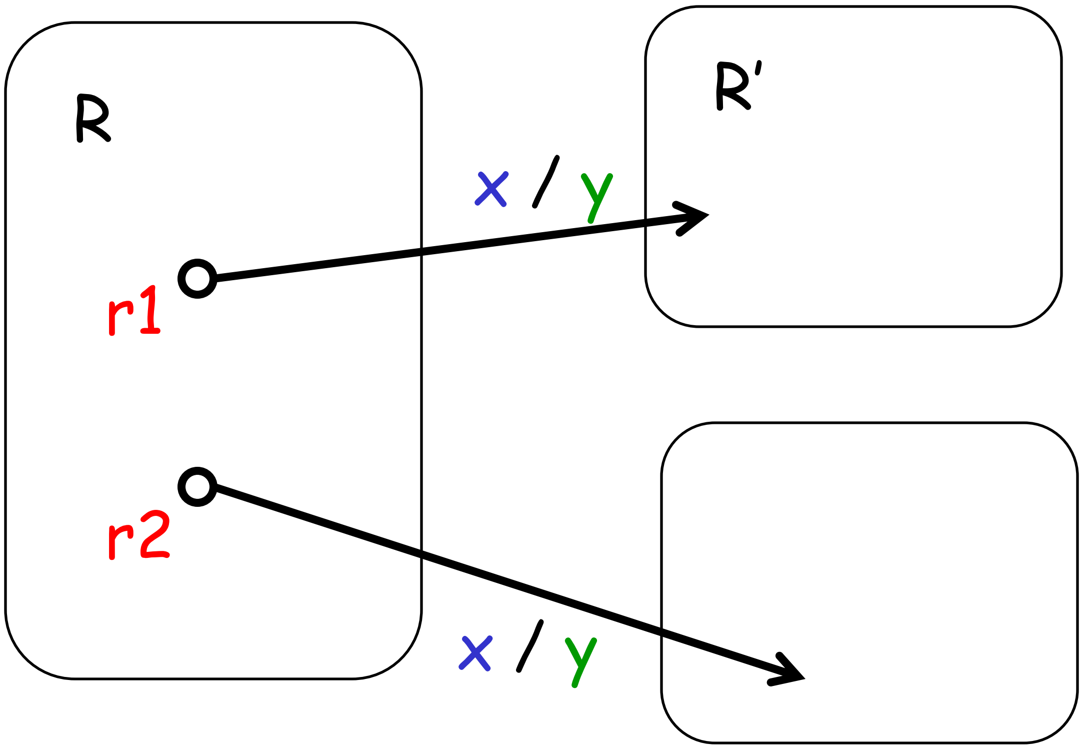
then

let $R1 = \{ r \in R \mid \text{nextState}(r,x) \in R' \}$;

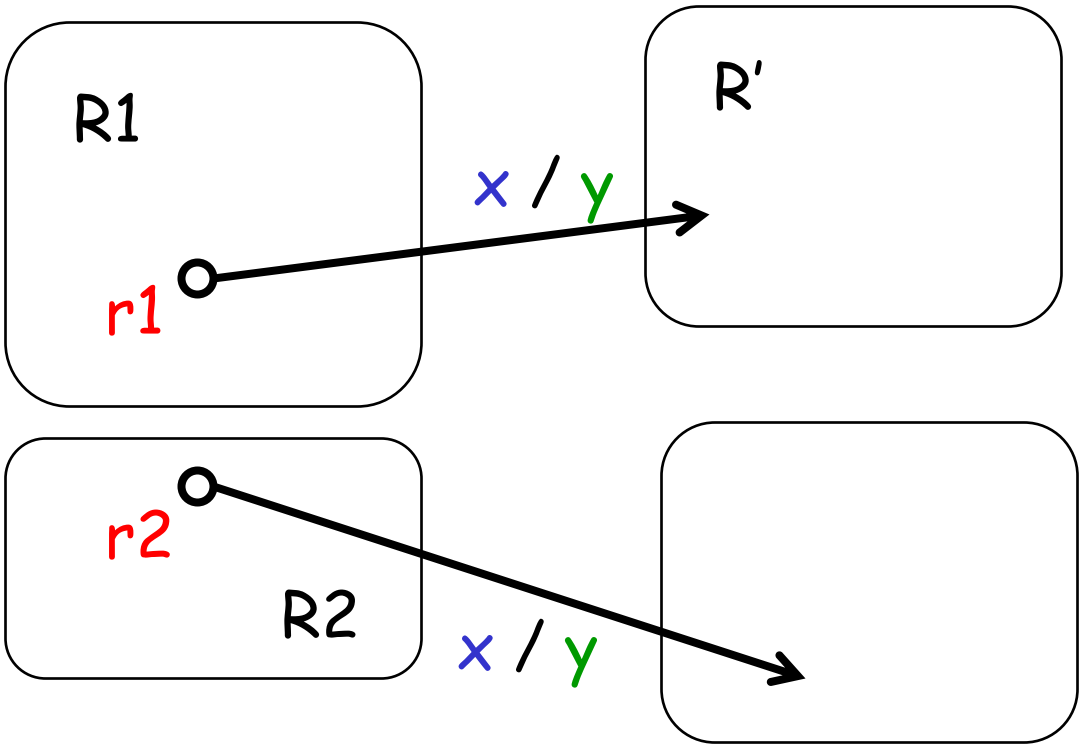
let $R2 = R \setminus R1$;

let $P = (P \setminus \{R\}) \cup \{R1, R2\}$.

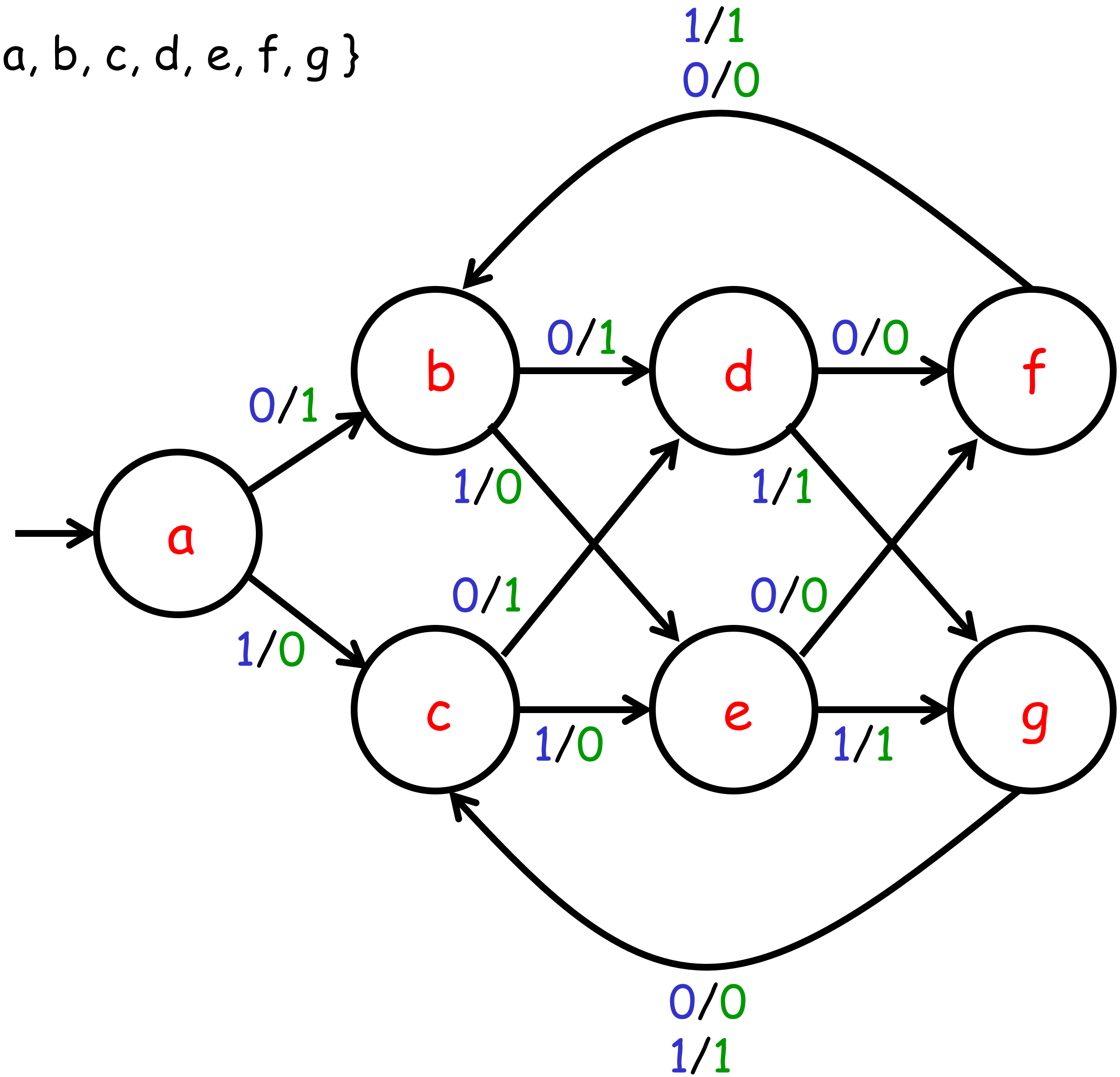
Next-state split



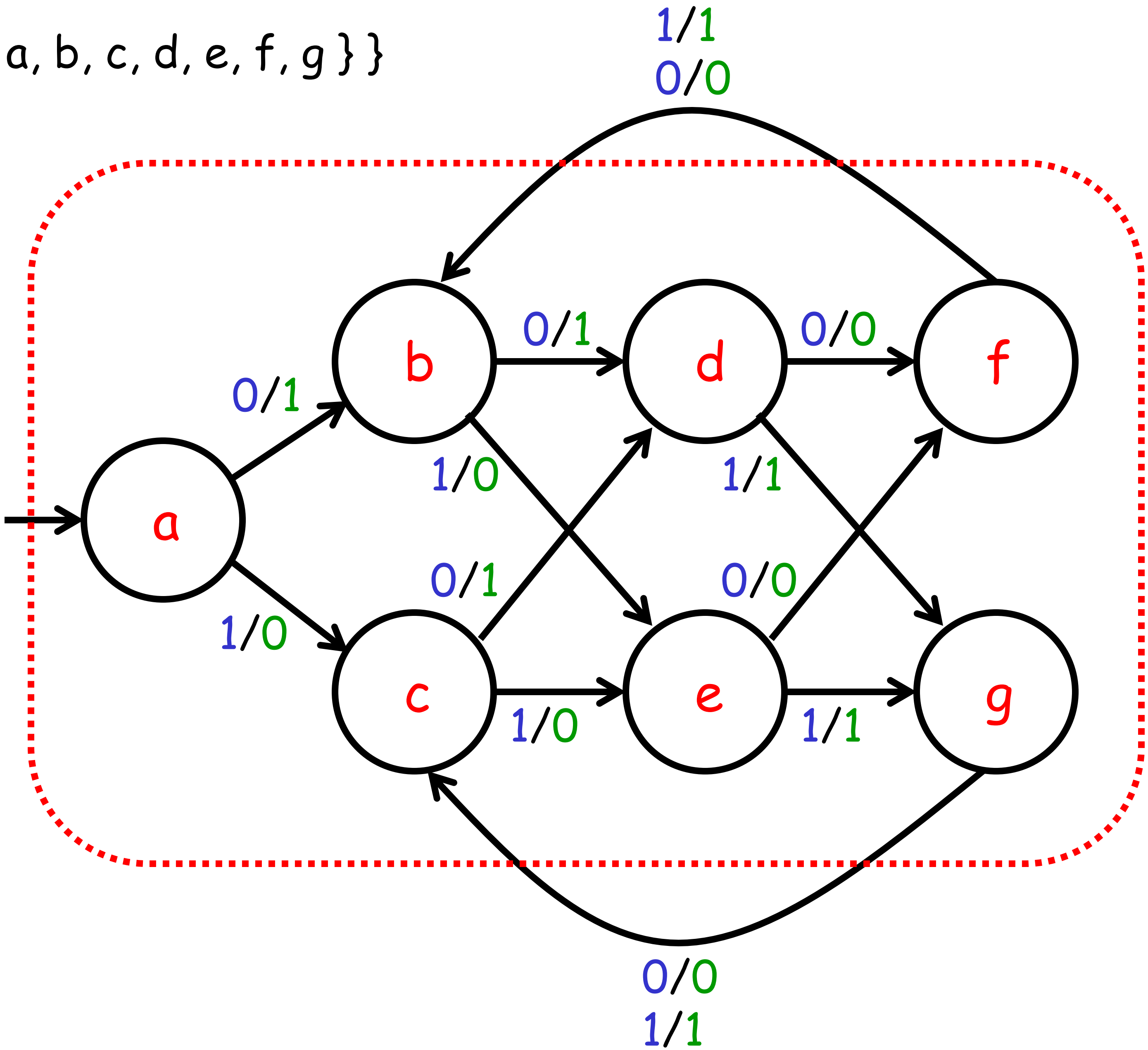
Next-state split



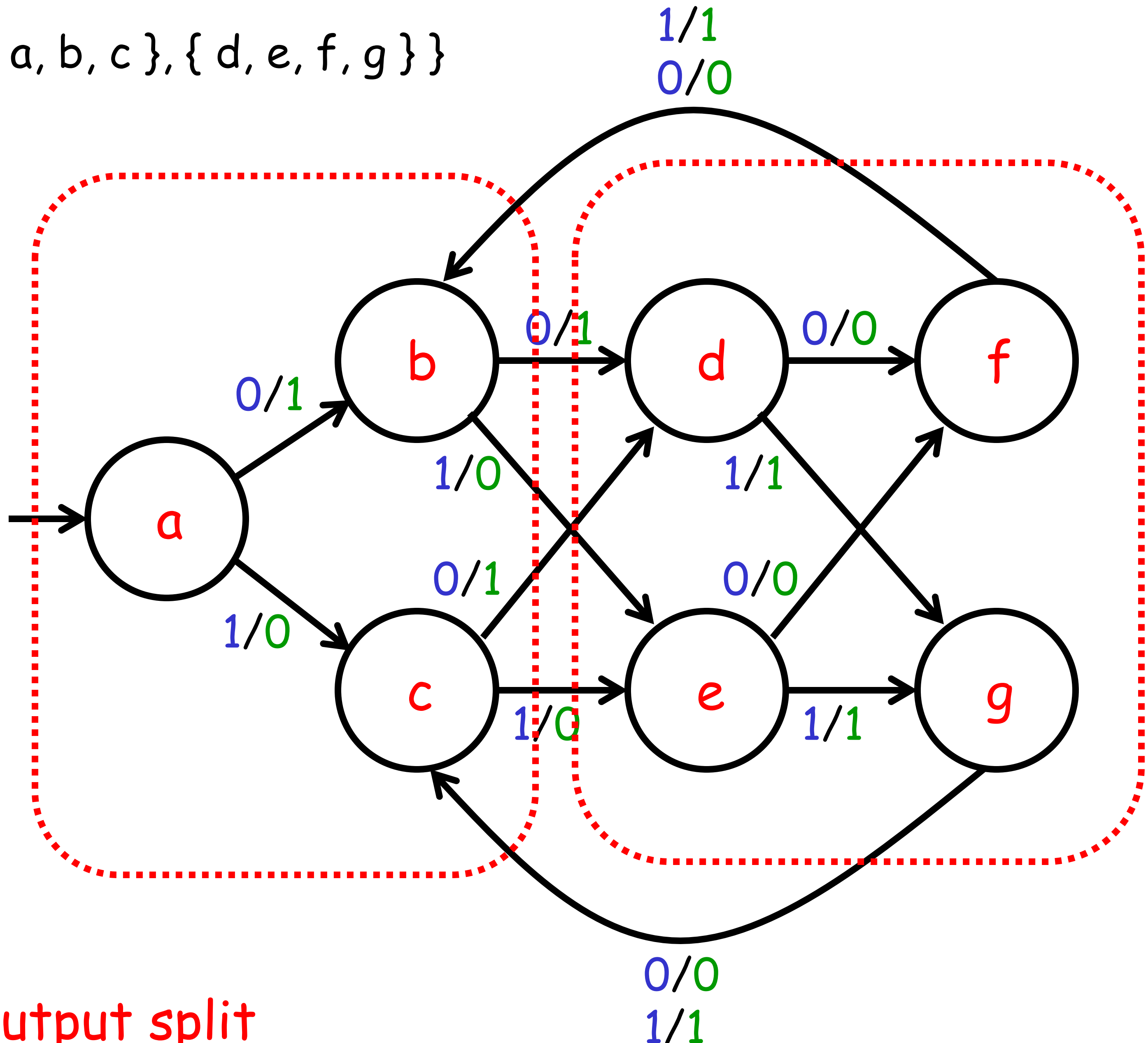
$Q = \{ a, b, c, d, e, f, g \}$



$P = \{ \{ a, b, c, d, e, f, g \} \}$

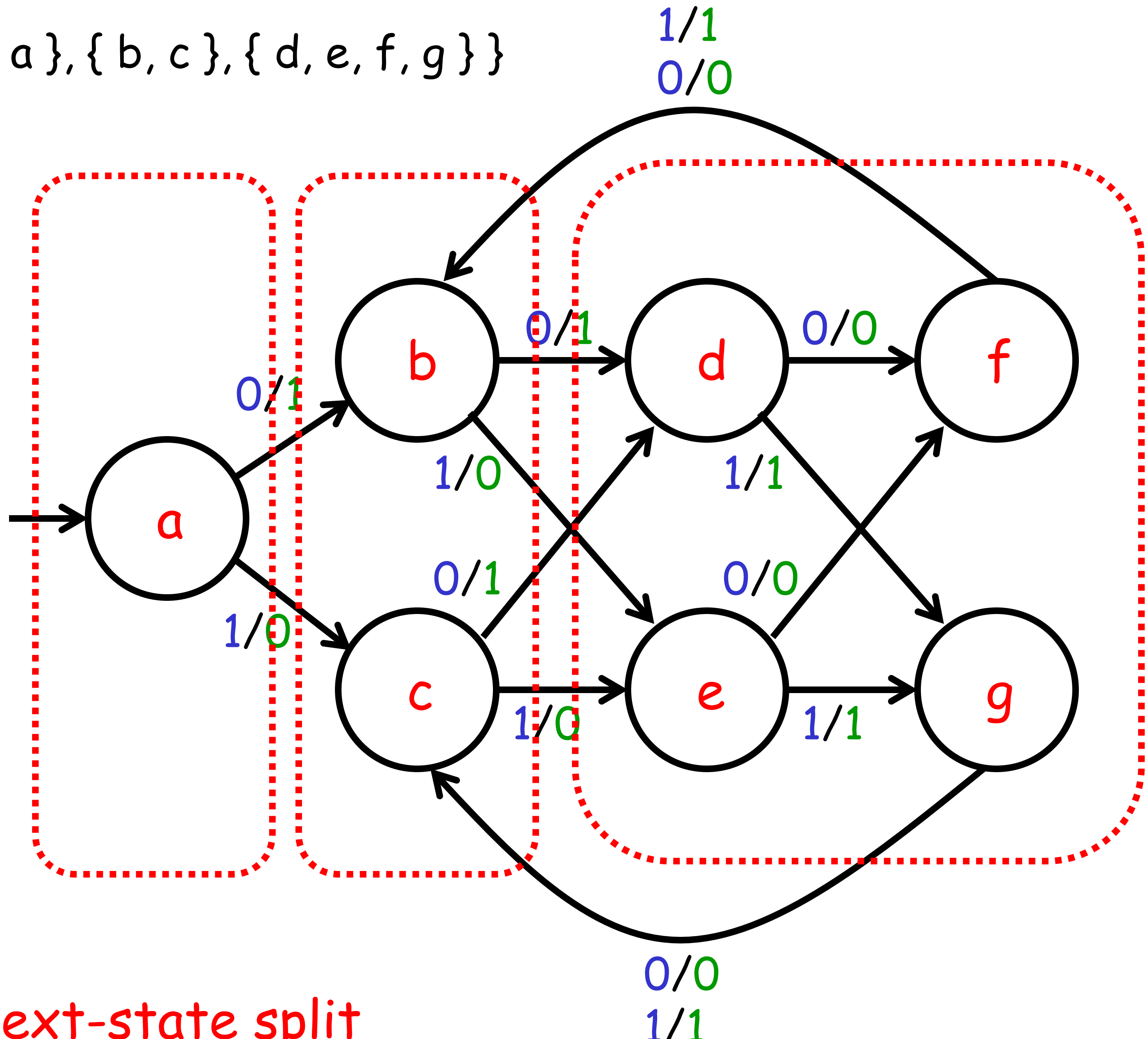


P = { { a, b, c }, { d, e, f, g } }



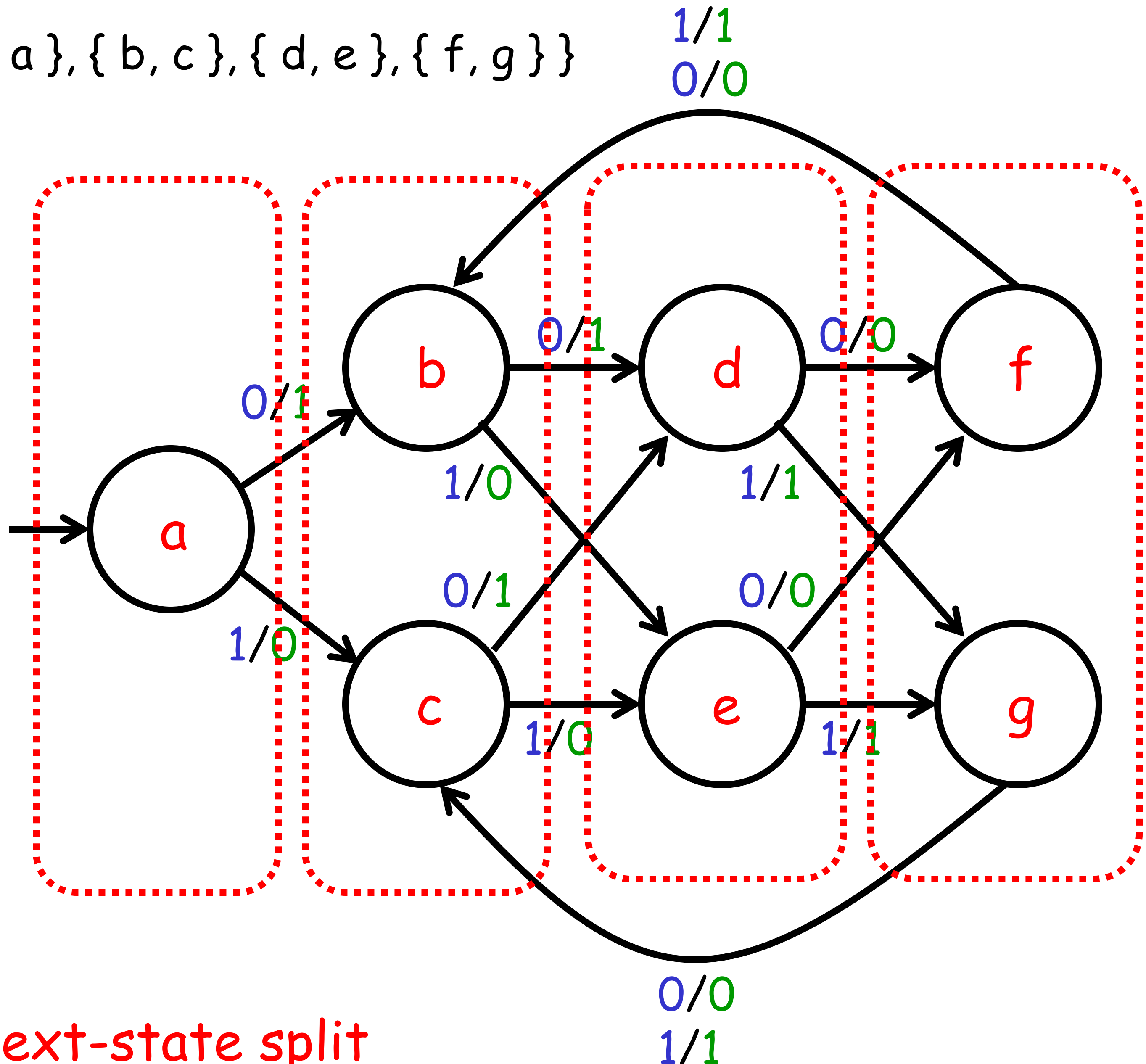
Output split

P = { { a }, { b, c }, { d, e, f, g } }

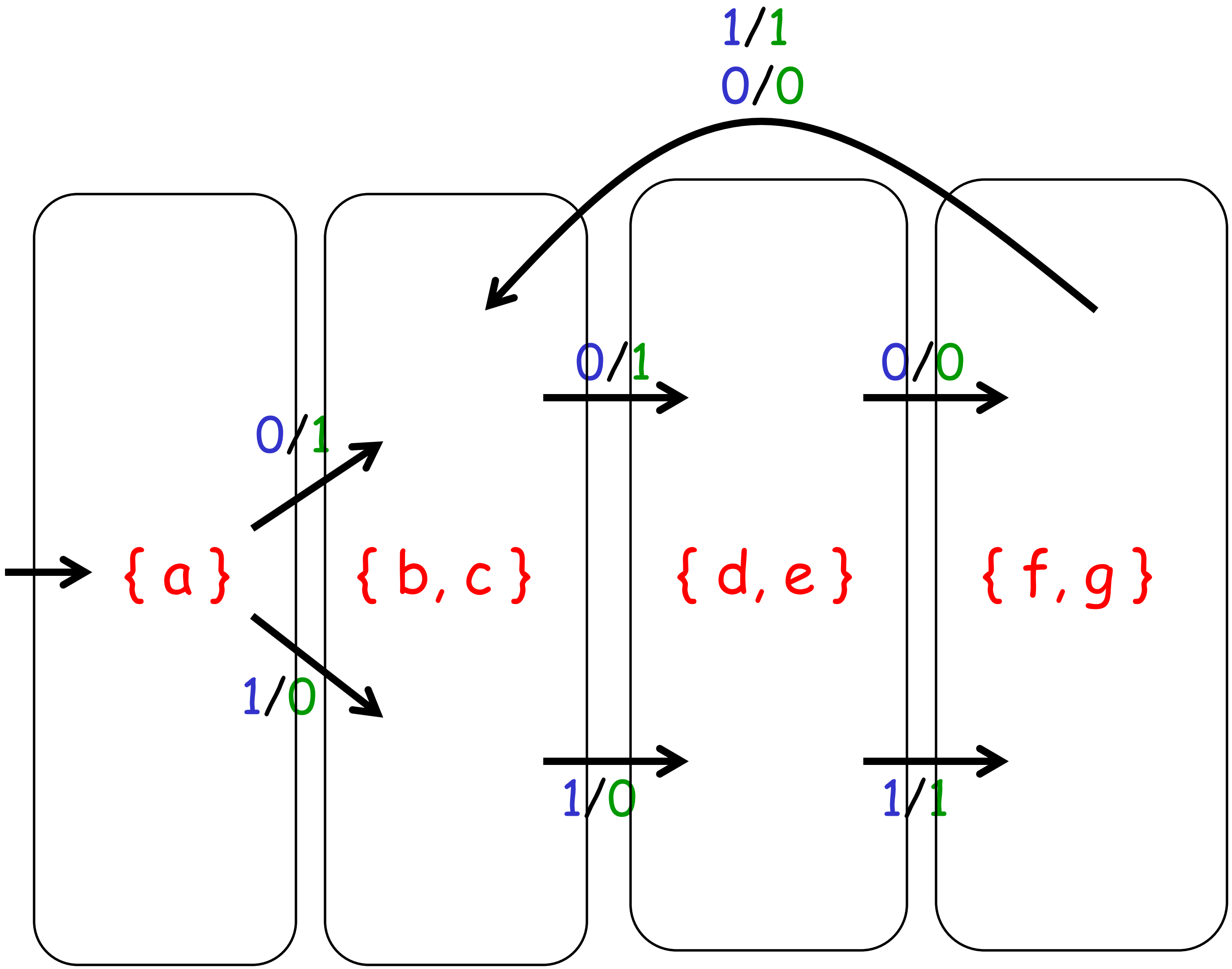


Next-state split

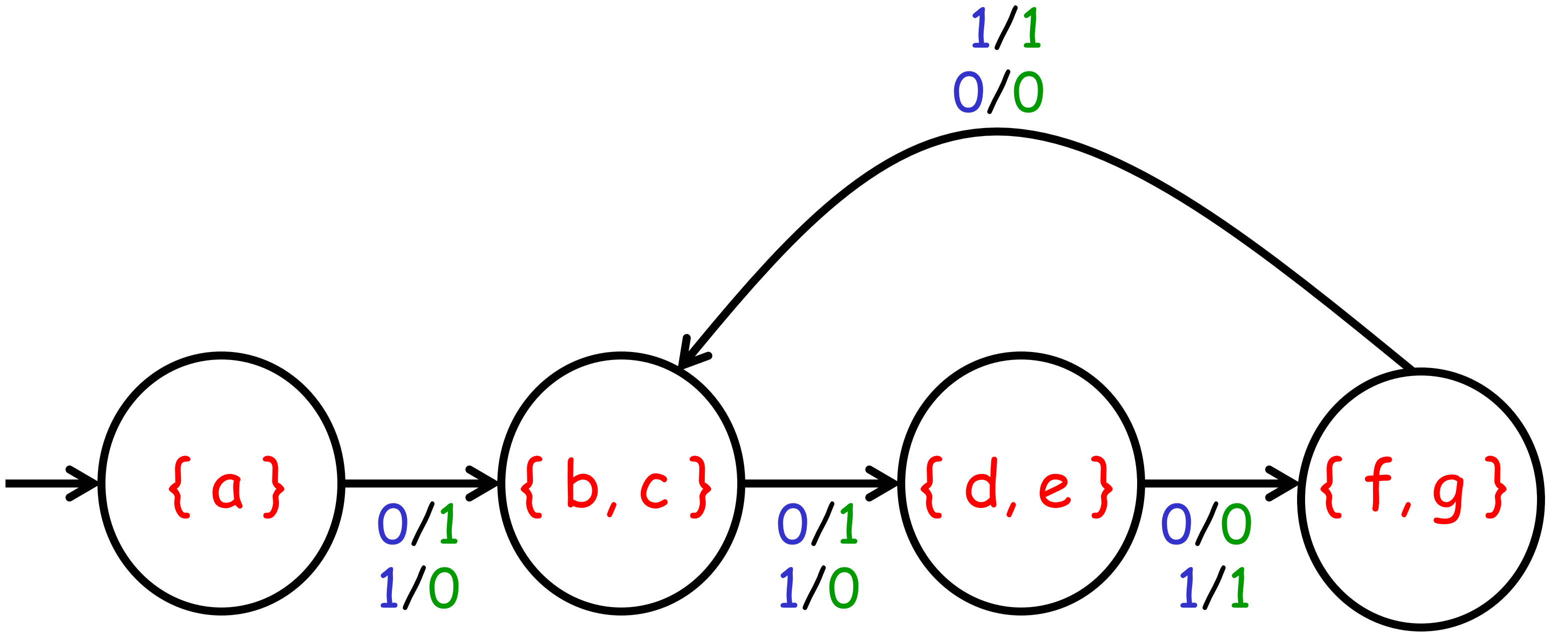
$P = \{\{a\}, \{b, c\}, \{d, e\}, \{f, g\}\}$



Next-state split



Minimal bisimilar state machine



4 instead of 7 states

Theorem:

There is a bisimulation between two state machines M1 and M2

iff

there is an isomorphism between minimize (M1) and minimize (M2)

(i.e., a bisimulation that is a one-to-one and onto function).

a renaming of the states

How to check if M1 and M2 are equivalent :

1. Minimize M1 and call the result N1
2. Minimize M2 and call the result N2
3. Check if the states of N1 can be renamed so that N1 and N2 are identical