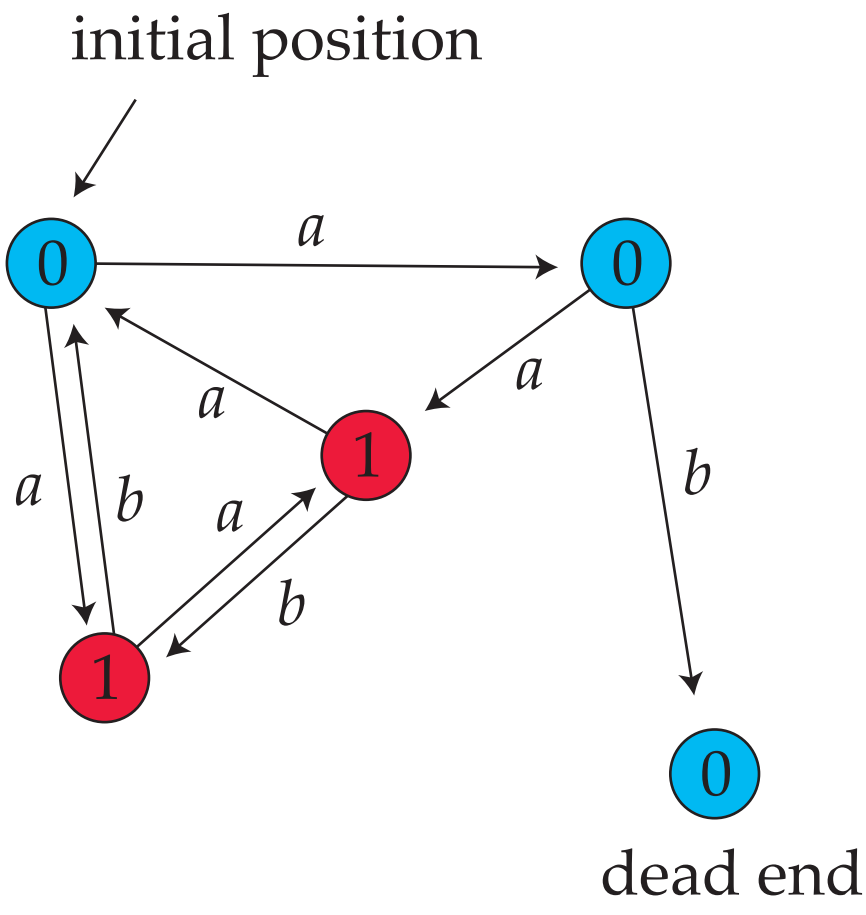
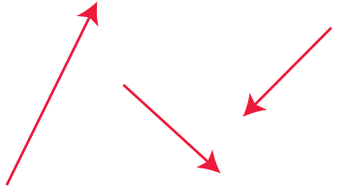


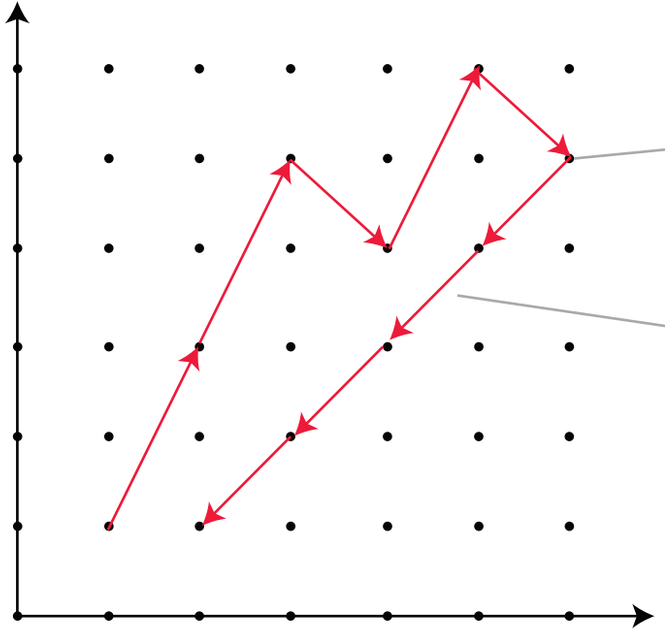
0 position controlled by player 0

1 position controlled by player 1

Winning condition for infinite plays:
player 0 wins if label *a* appears
infinitely often, otherwise 1 wins



δ



run

configuration

transition, i.e. a pair

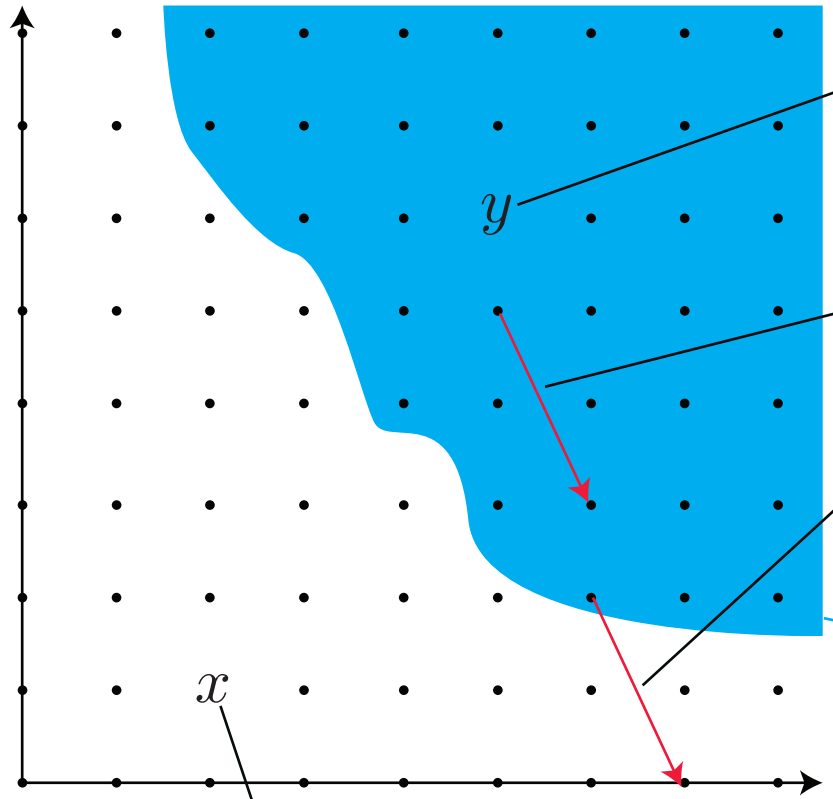
$$x \to y$$

of configurations satisfying

$$y - x \in \delta$$

(3) is upward closed

for the coordinate-wise ordering on tuples of natural numbers

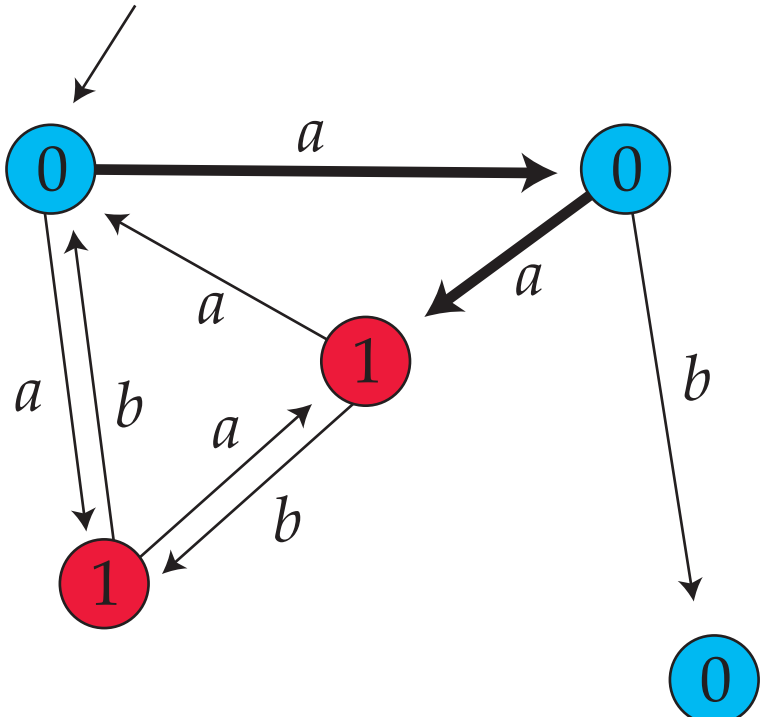


(2) contains y

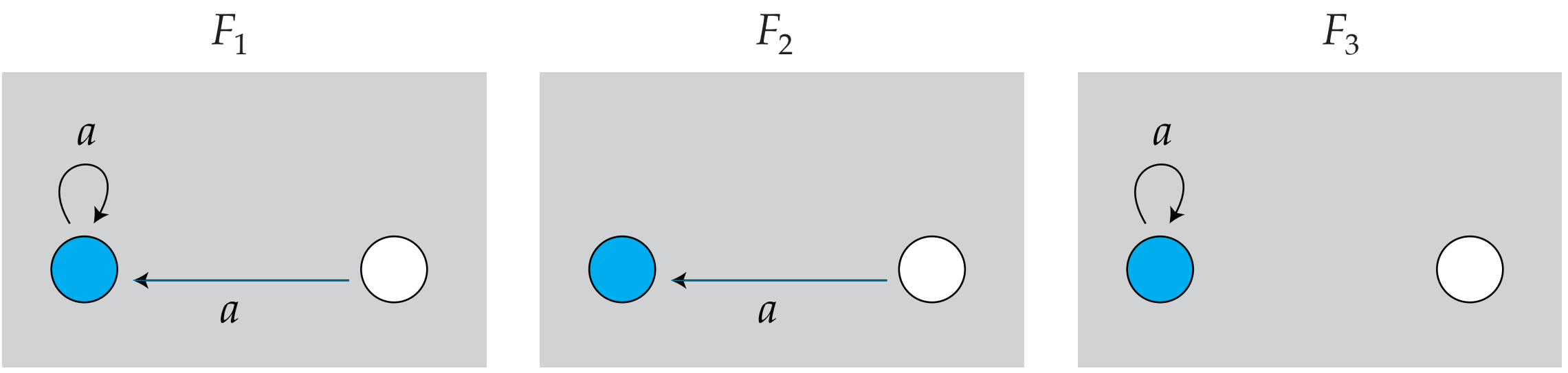
(4) is backward closed under transitions: if it contains the target of a transition, then it also contains the source.

the separator

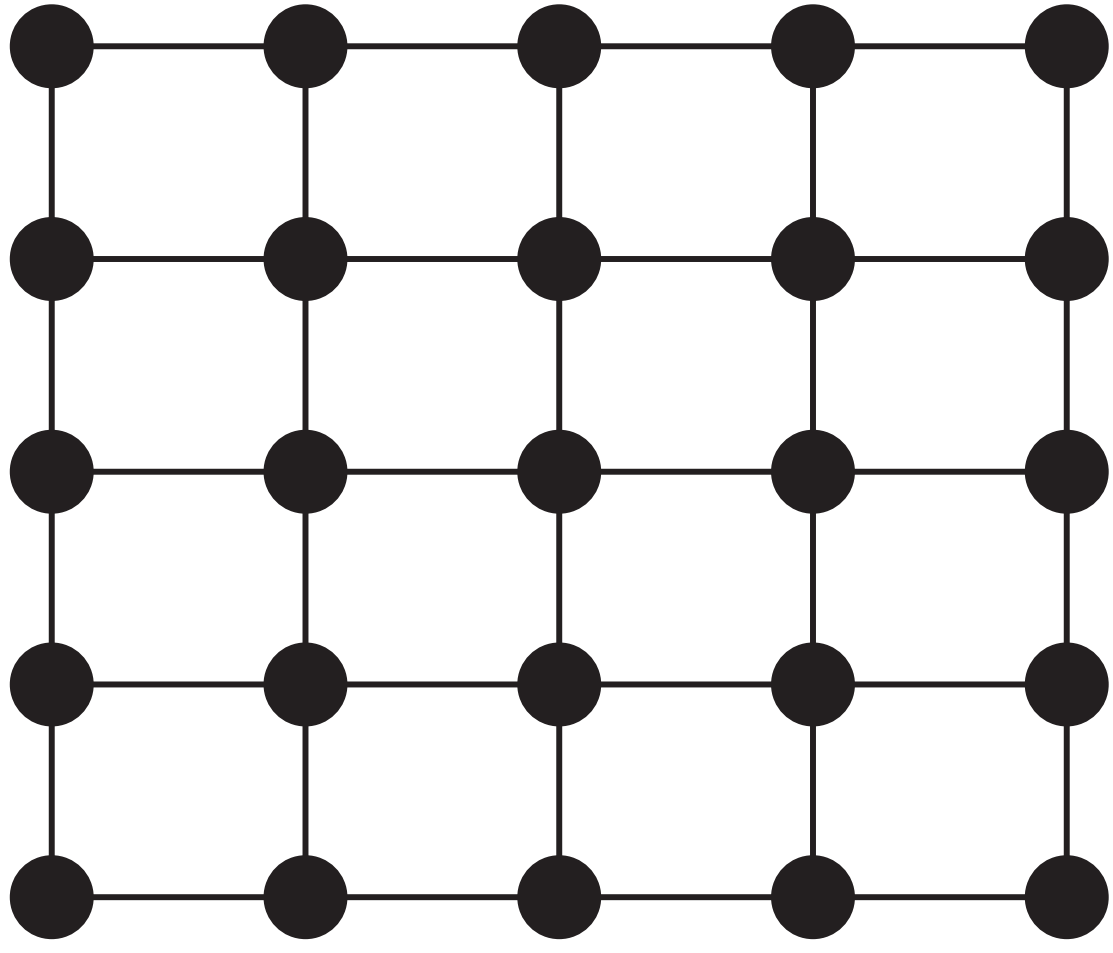
(1) does not contain x



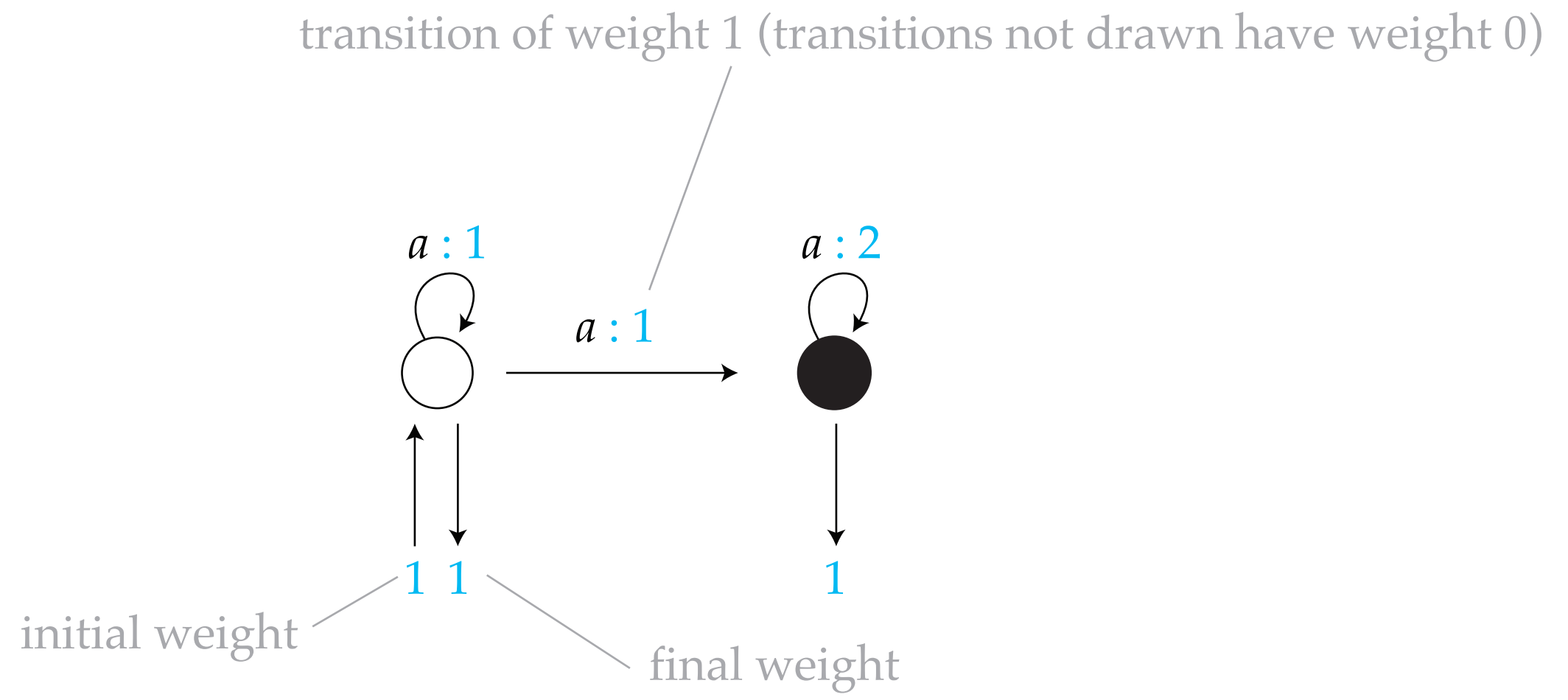
moves chosen by player 0



a set of transitions is pictured by showing the automaton with only transitions from the set



square grid of dimension 5

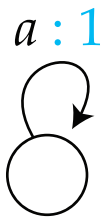

transition of weight 1 (transitions not drawn have weight 0)
a : 1
a : 1
a : 2
1 1
1
initial weight
final weight

1

initial weight of

1^i

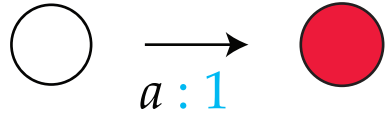
number of times the transition



is used

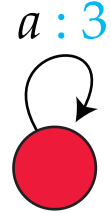
1

weight of transition



2^j

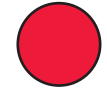
number of times the transition



is used

2

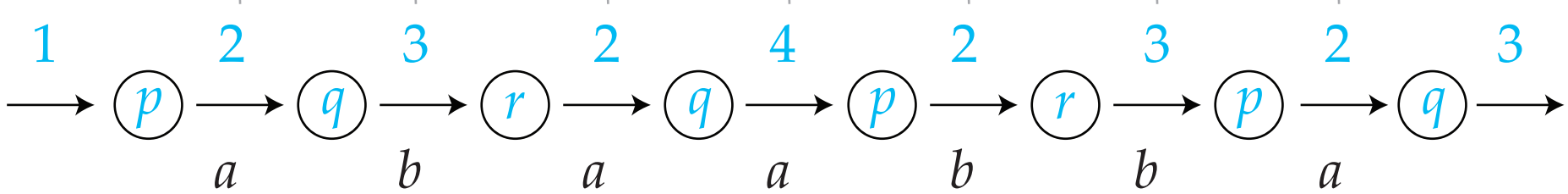
final weight of



initial weight of first state

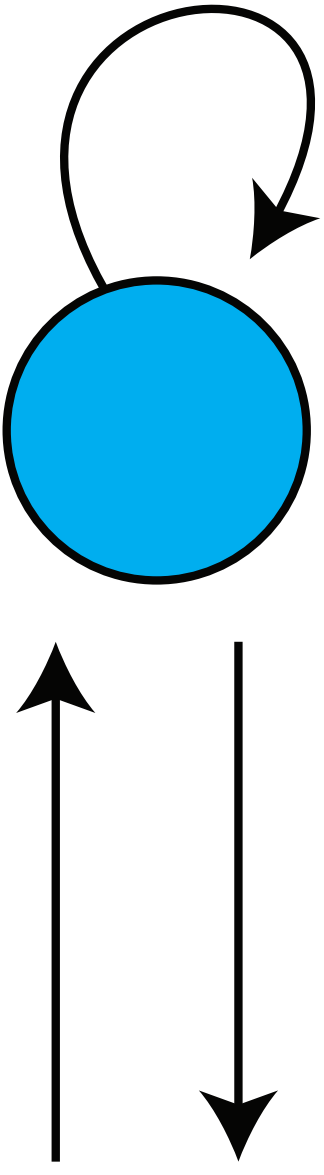
final weight of last state

weights of transitions connecting consecutive states

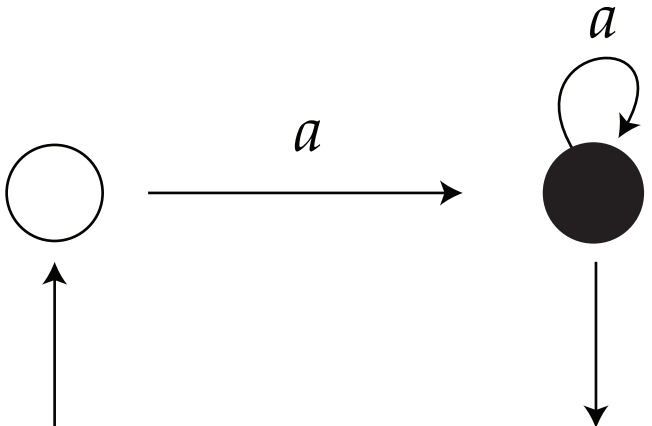
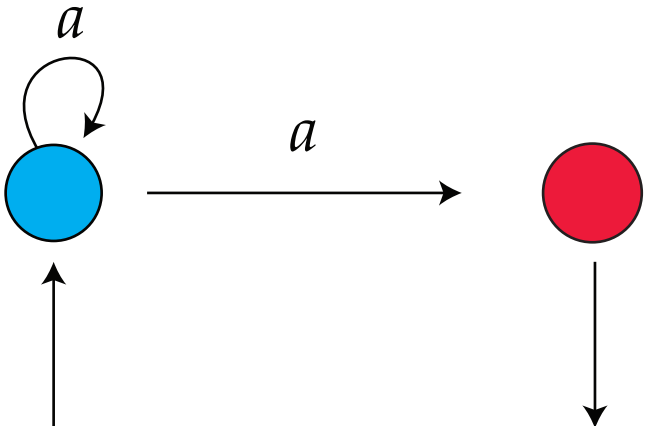


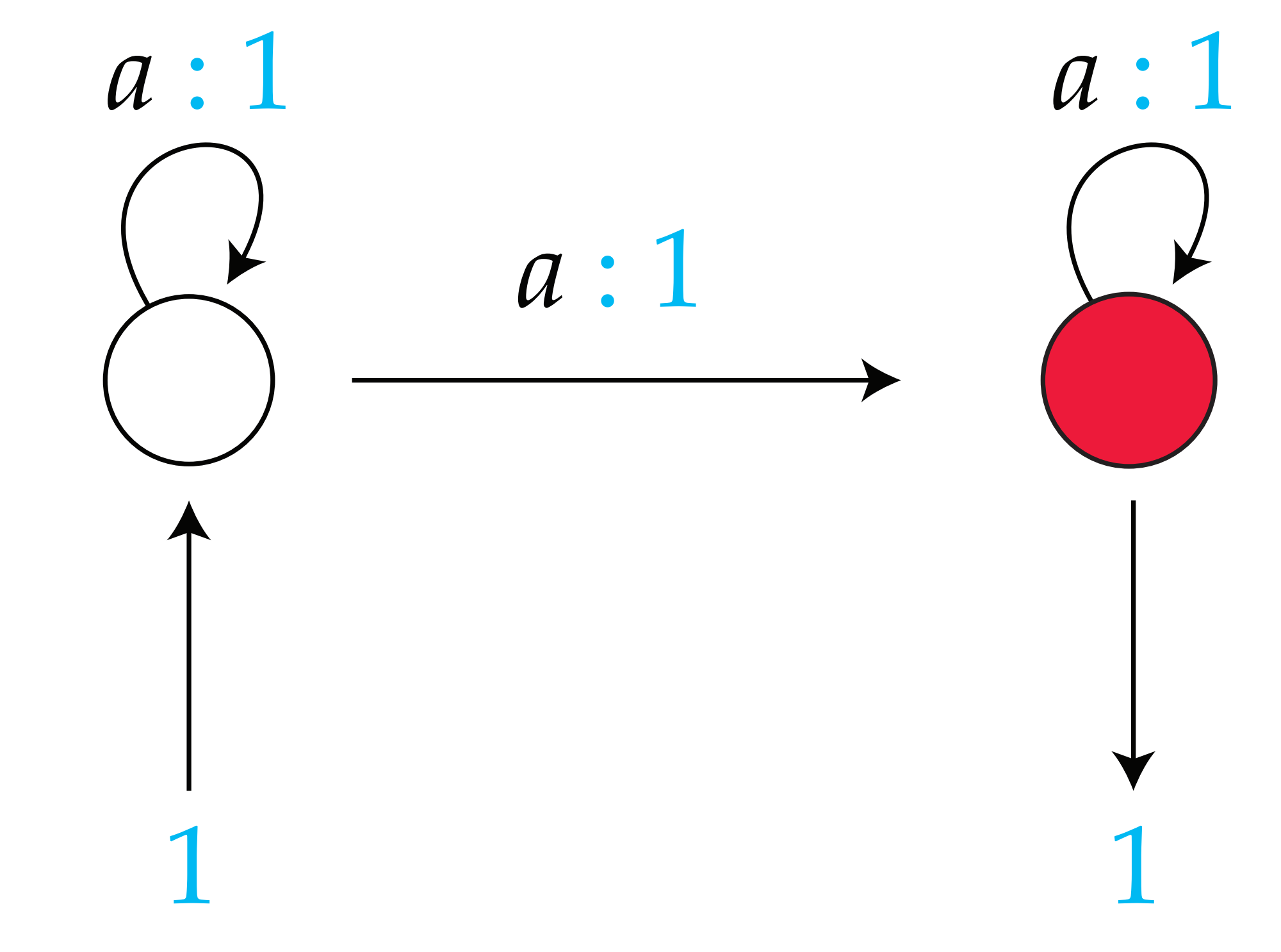
weight of the run $1 \cdot 2 \cdot 3 \cdot 2 \cdot 4 \cdot 2 \cdot 3 \cdot 2 \cdot 3 = 1728$

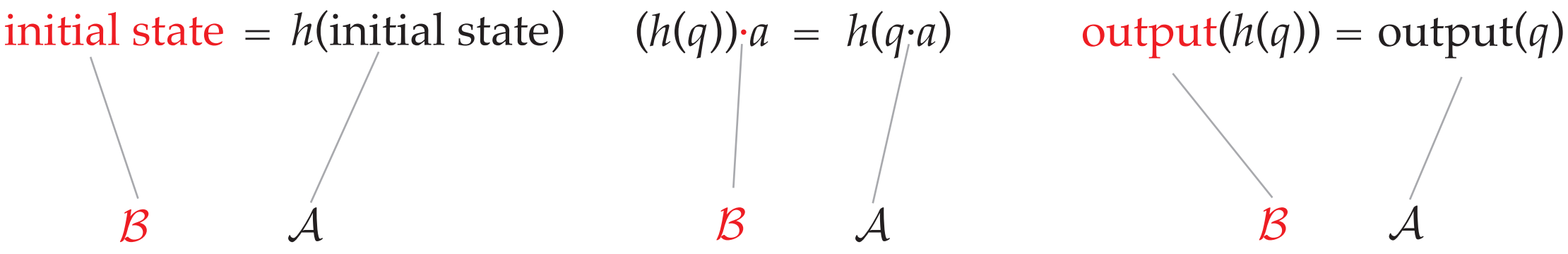
$a : 2$



1 1

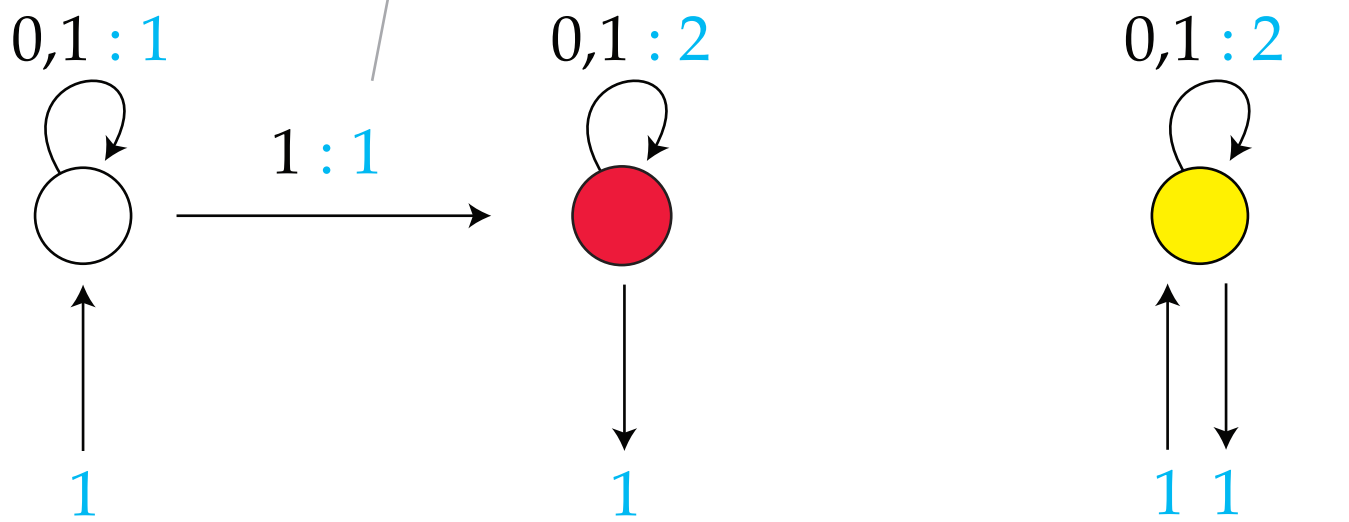

a
a

a
a


a : 1
a : 1
a : 1
1
1


initial state = h(initial state)
(h(q))·a = h(q·a)
output(h(q)) = output(q)
B
A
B
A
B
A

if a position in the input word has bit 1 and is followed by i bits, then the position contributes 2^i to the output

if the input word has length n, then the loop around this state will contribute 2^n



binary representation

leading 1

q'_0 q'_1 q'_2 $\cdots$ q'_{n-1} q'_n
p_0 p_1 p_2 p_{n-1} p_n

run of the product automaton

w_1 w_2 $\cdots$ w_n

output of the transducer

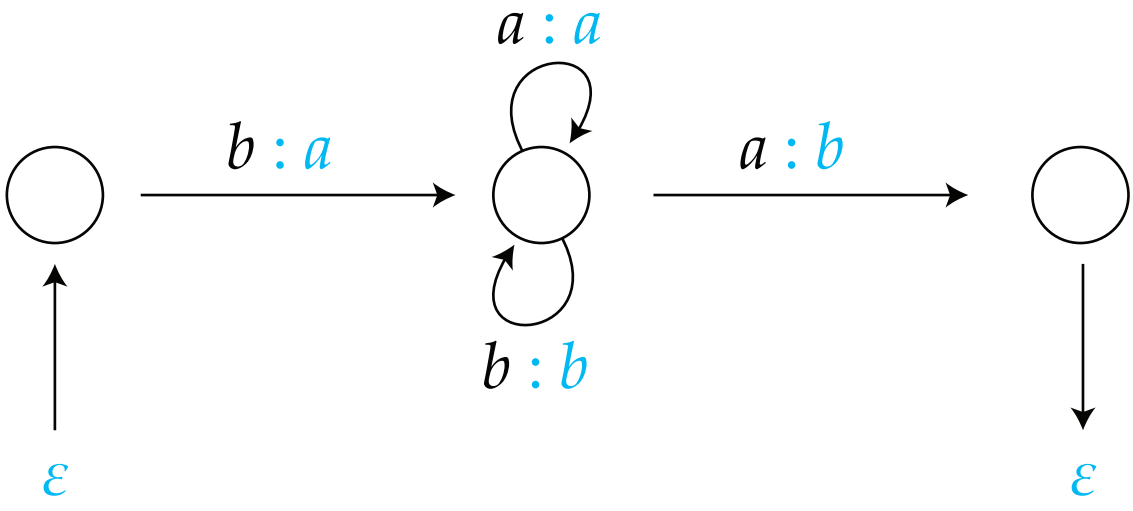
q_0 q_1 q_2 $\cdots$ q_{n-1} q_n

states of the transducer

a_1 a_2 $\cdots$ a_n

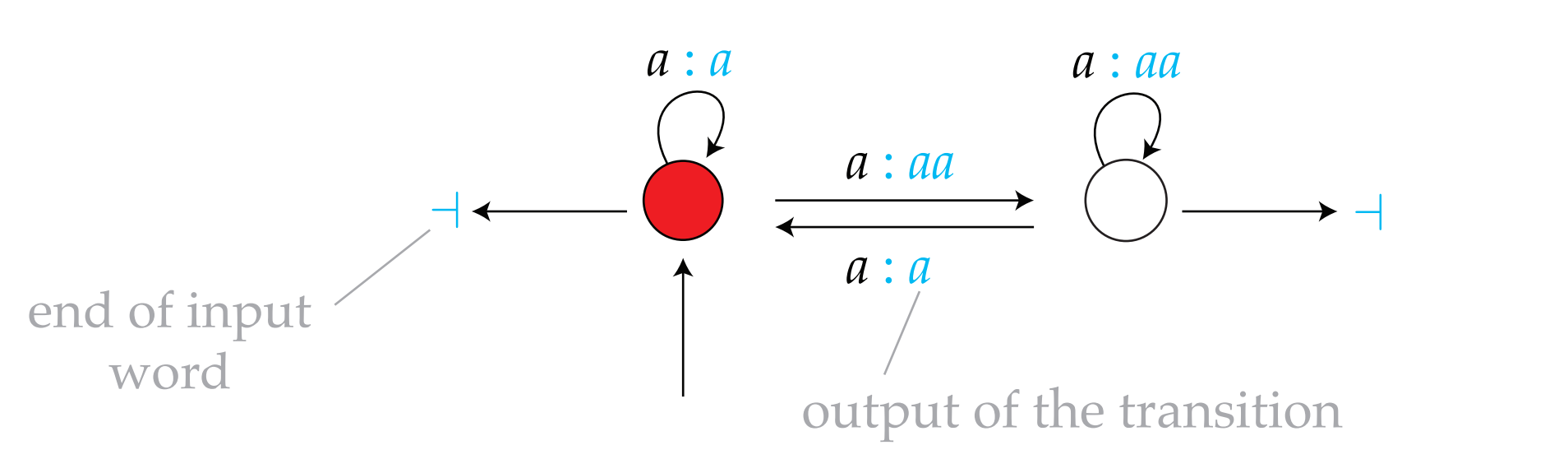
input word

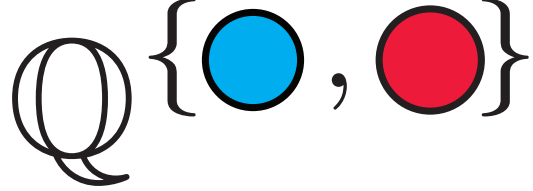
transition that inputs b and outputs a

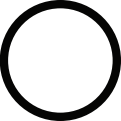


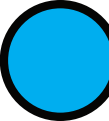
initial state
labelled by ε

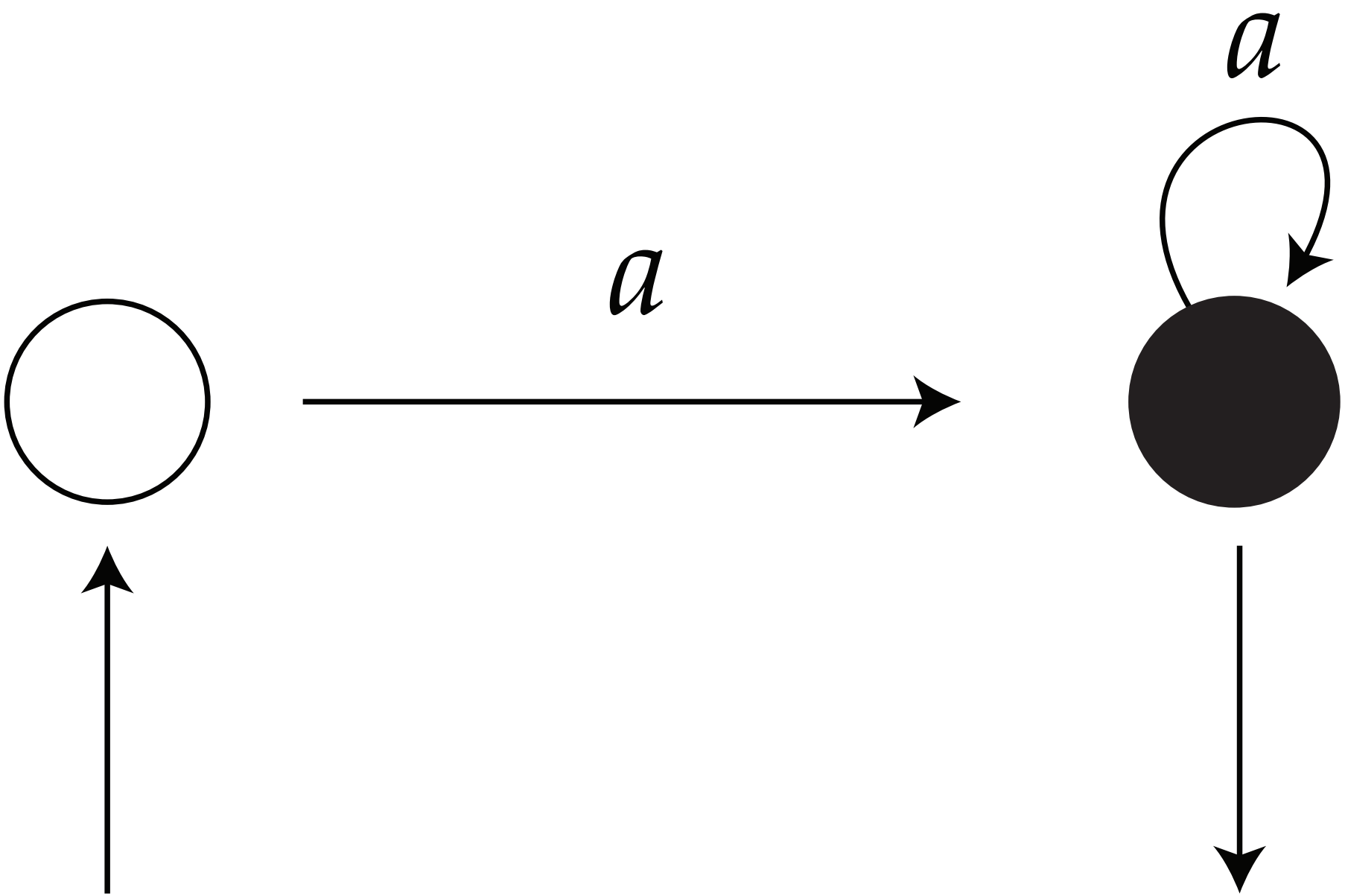
final state
labelled by ε

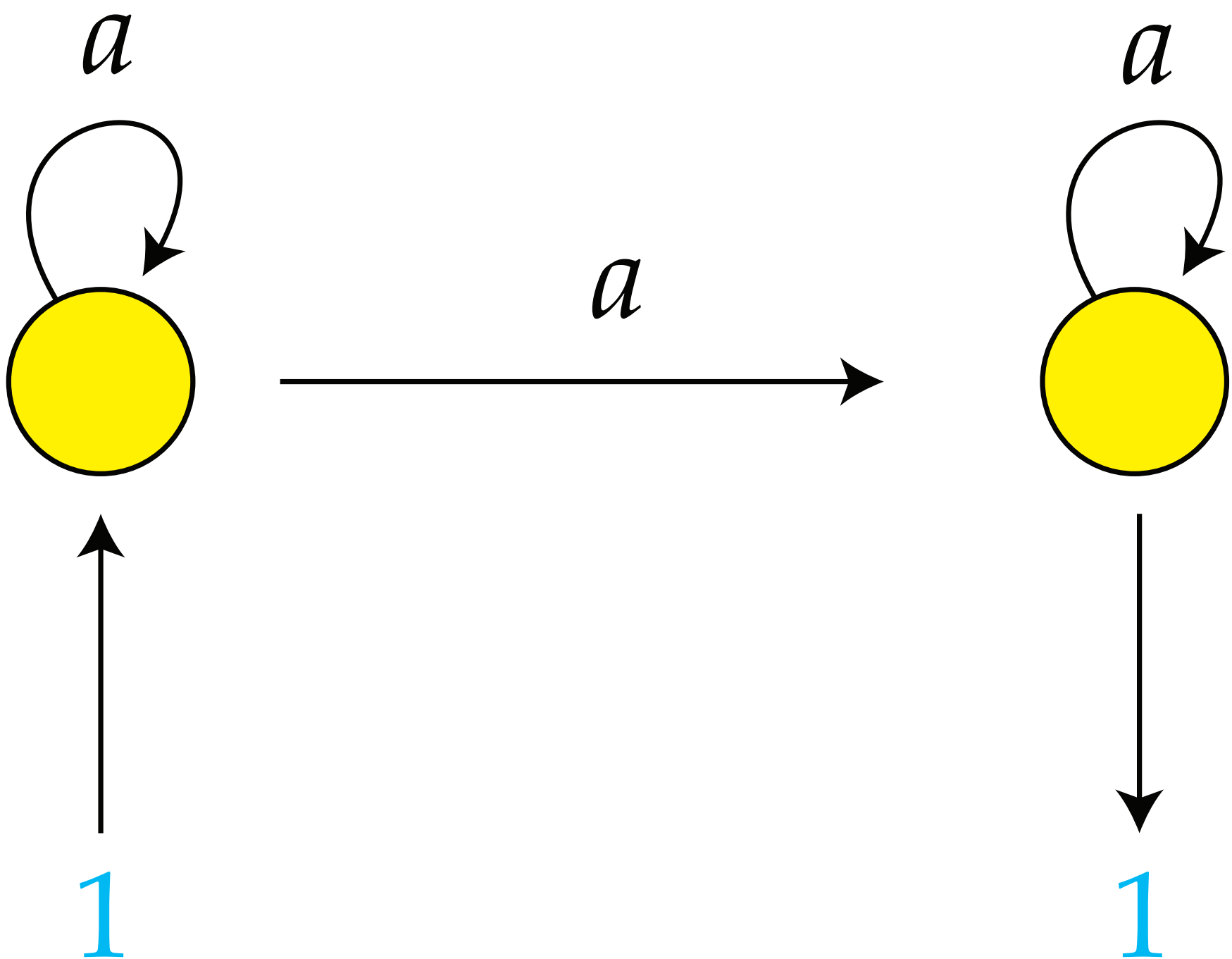

a : a
a : aa
a : aa
a : a
end of input word
output of the transition

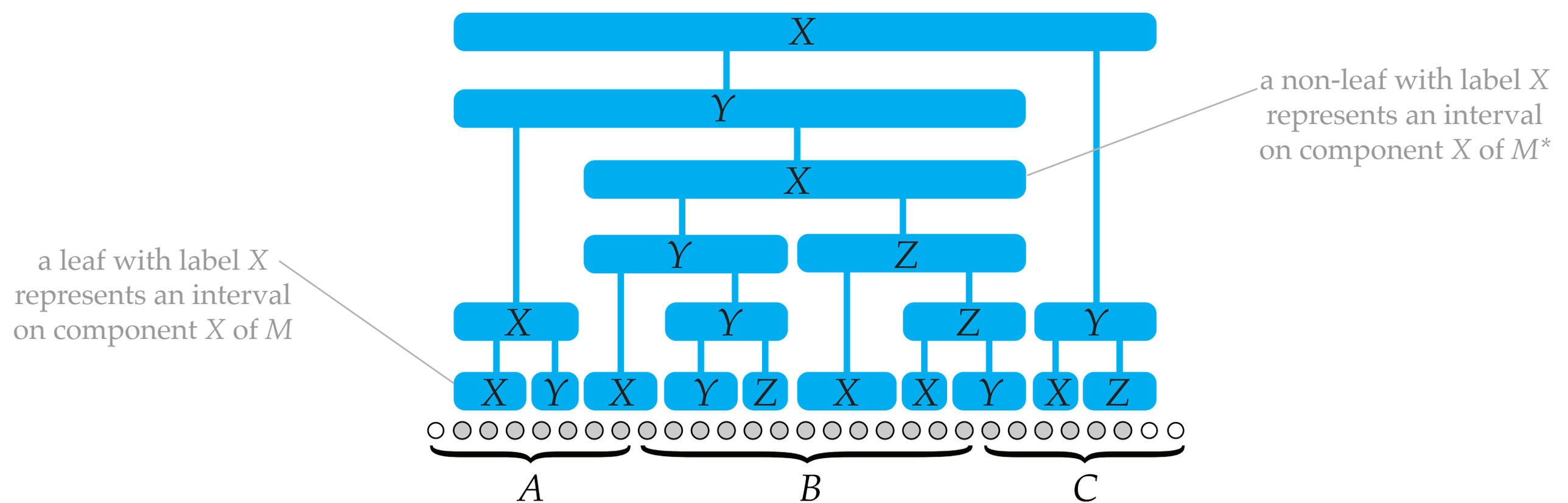


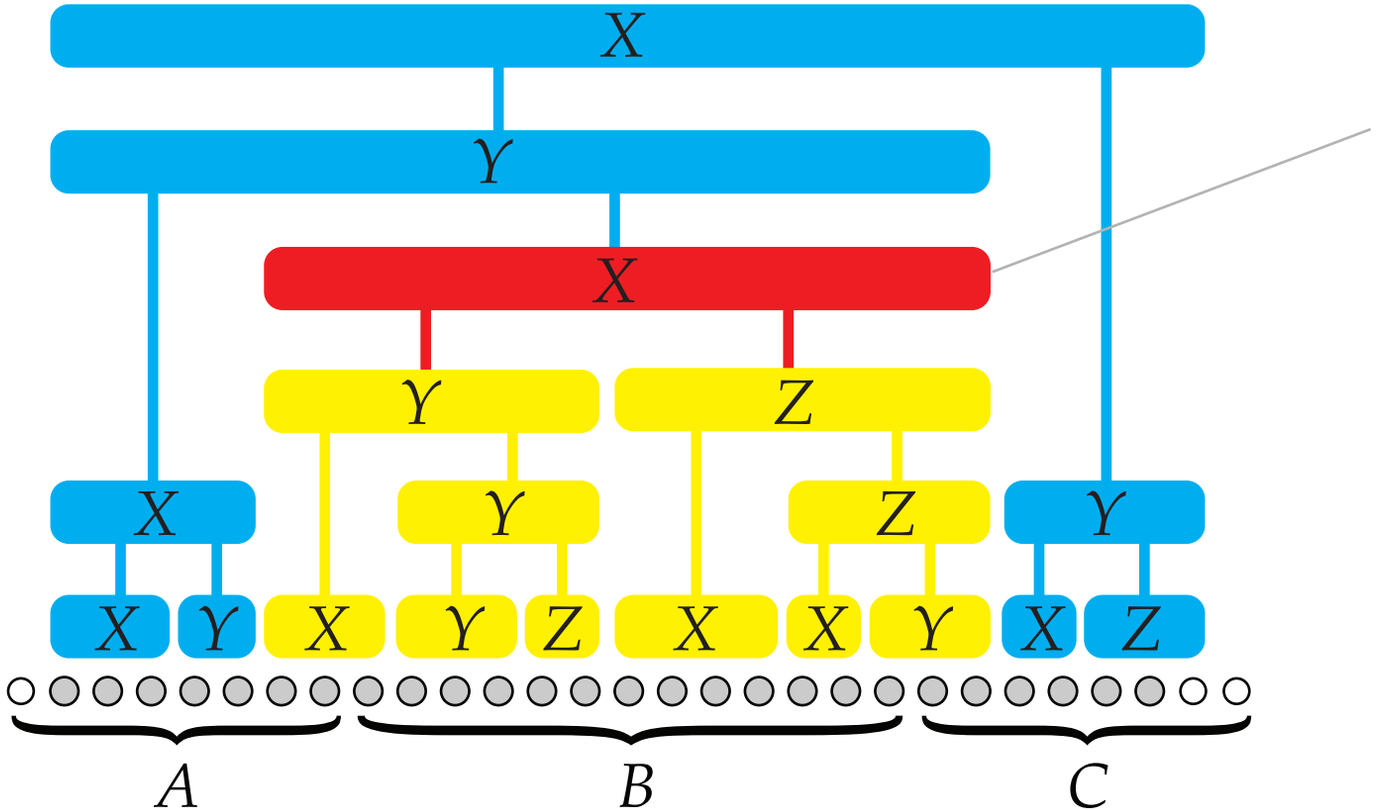




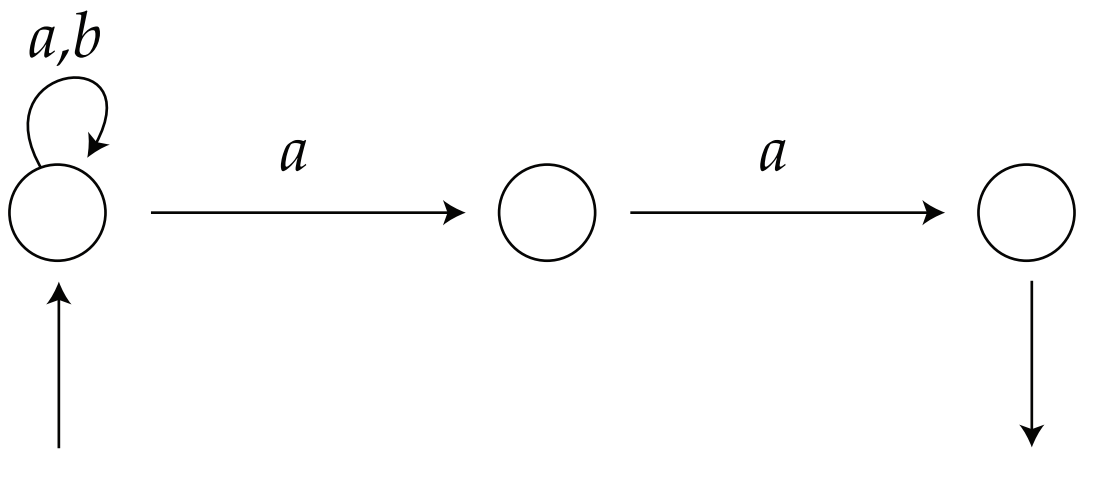

a
a

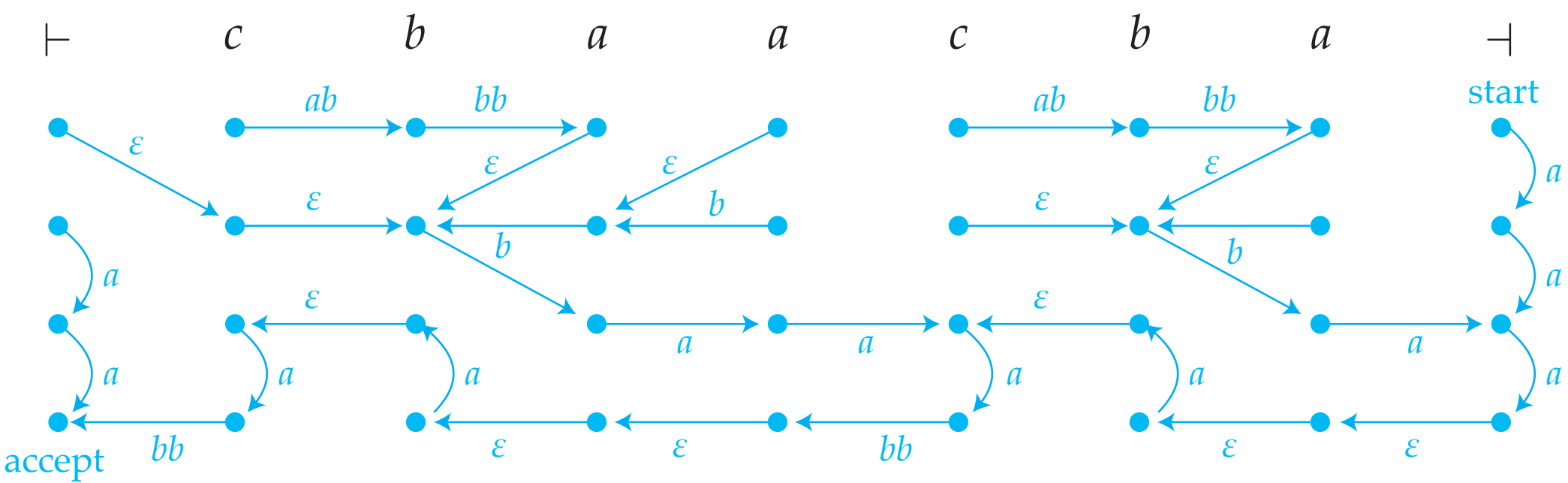

a
a
a
1
1

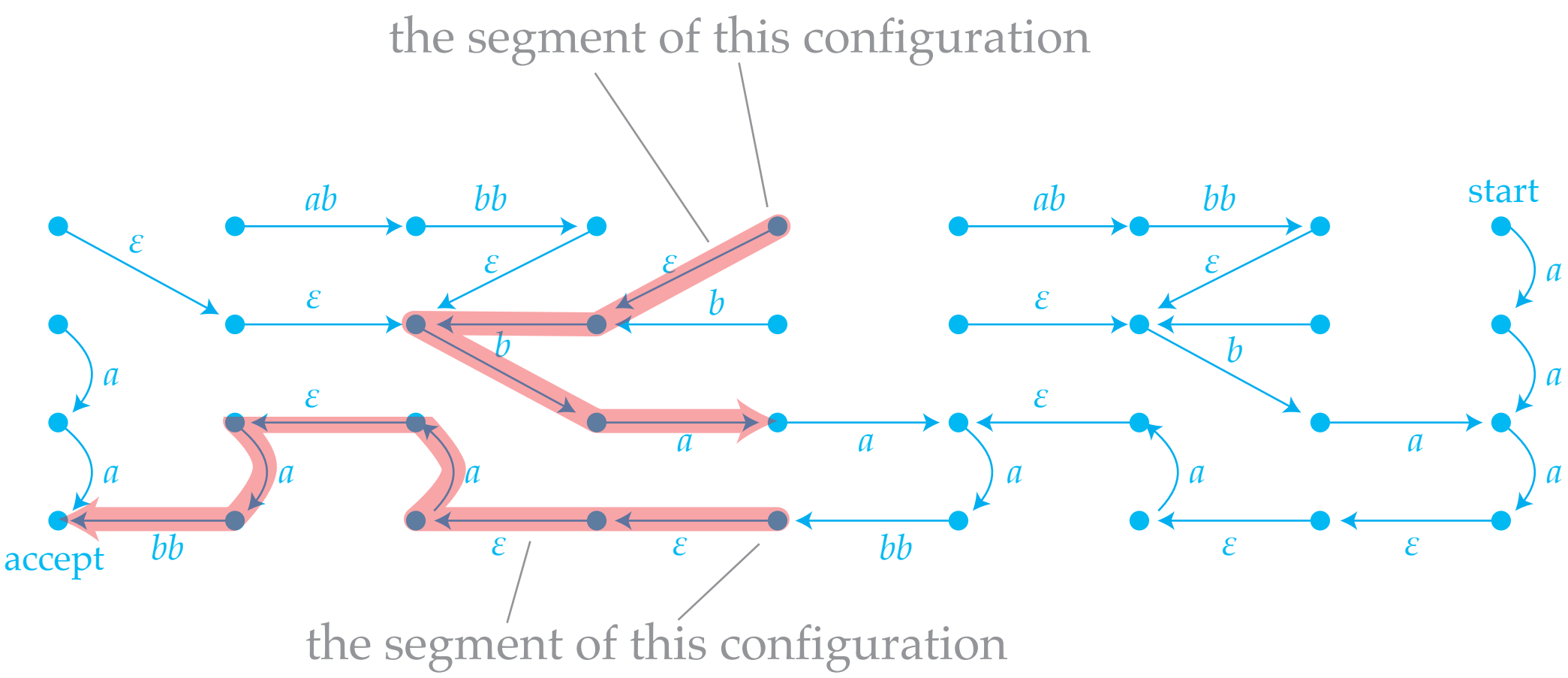

X
Y
X
Y
Z
X
Y
Z
Y
X
Y
Z
X
X
Y
X
Y
Z
X
X
Y
X
Z
A
B
C
a leaf with label X represents an interval on component X of M
a non-leaf with label X represents an interval on component X of M*

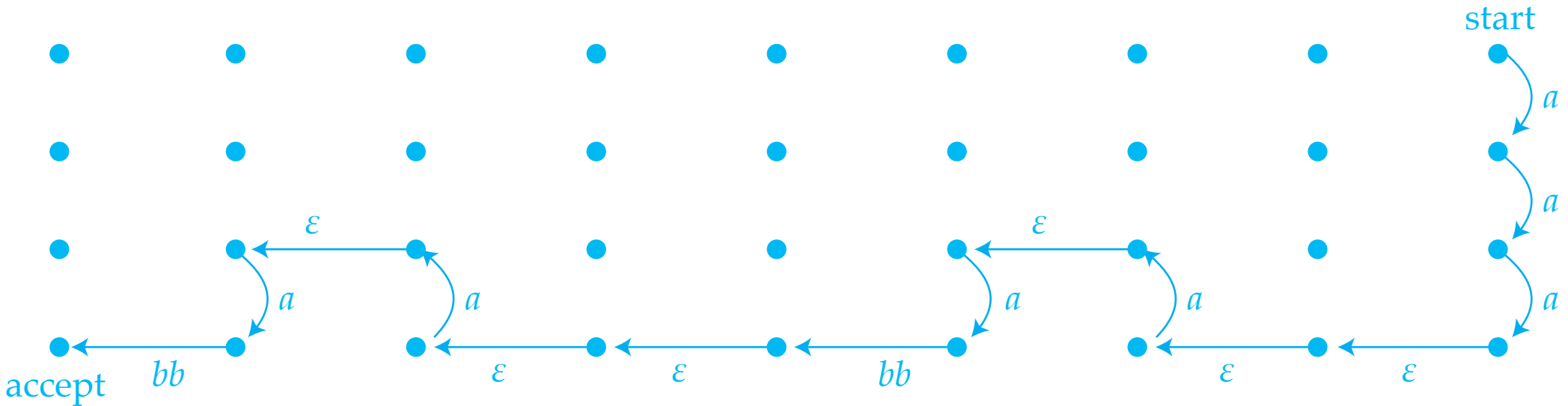


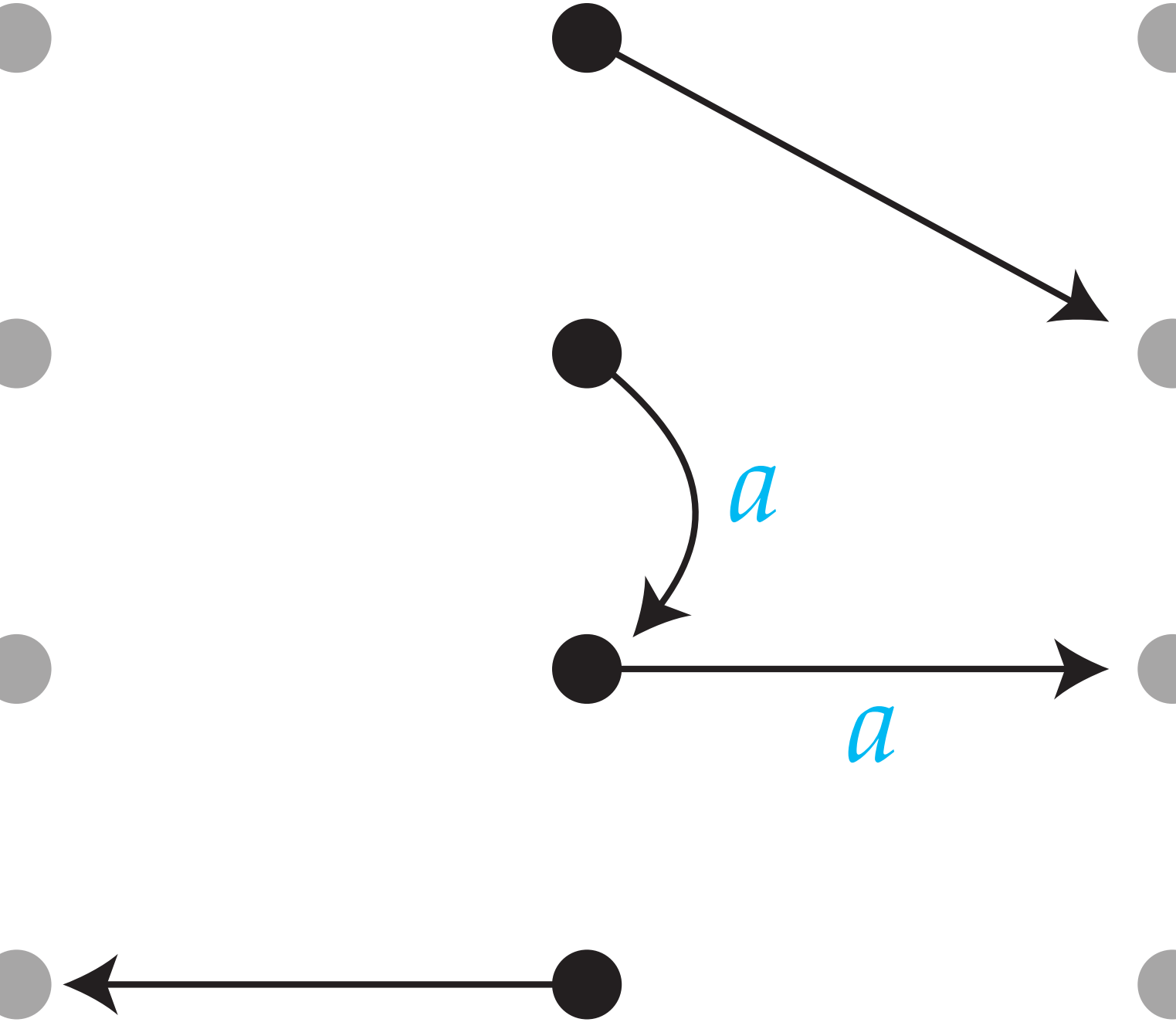
smallest interval
which contains B

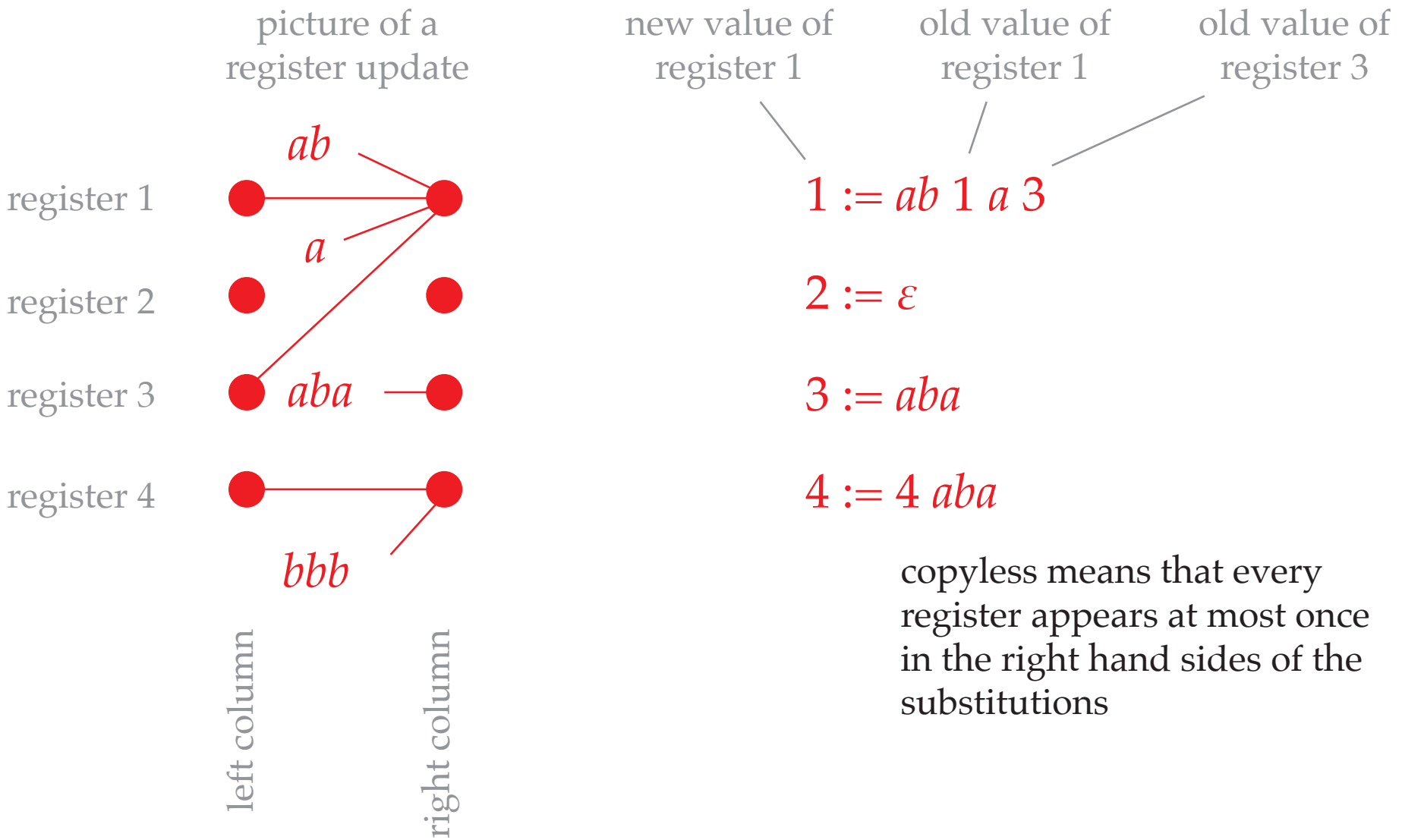

a,b
a
a


⊢
c
b
a
a
c
b
a
⊣
start
ab
bb
ε
ε
ε
ε
b
b
a
a
a
a
a
a
a
a
a
a
a
a
accept
bb
ε
ε
bb
ε
ε


the segment of this configuration
ab
bb
ε
ε
ε
ε
b
b
a
ε
a
a
a
a
a
accept
bb
ε
ε
bb
ab
bb
start
ε
ε
b
a
a
ε
a
a
a
a
ε
ε
the segment of this configuration

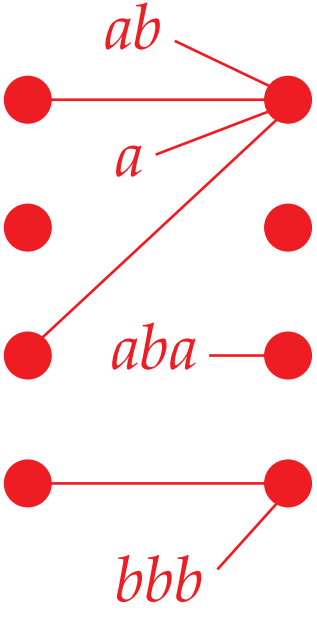

start
a
a
a
ε
a
ε
a
a
accept
bb
ε
ε
bb
ε
ε

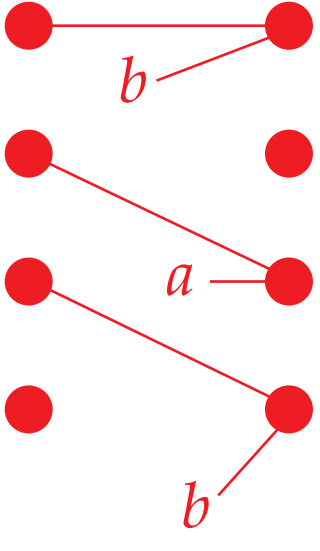

a
a



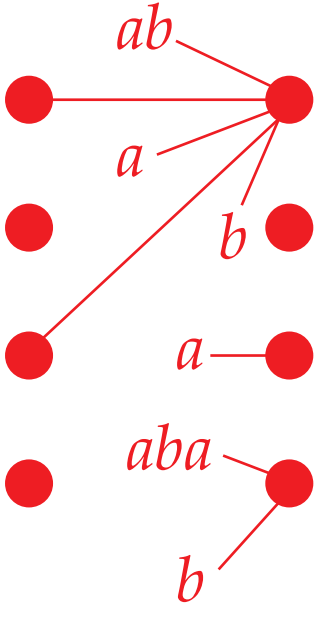
copyless means that every register appears at most once in the right hand sides of the substitutions

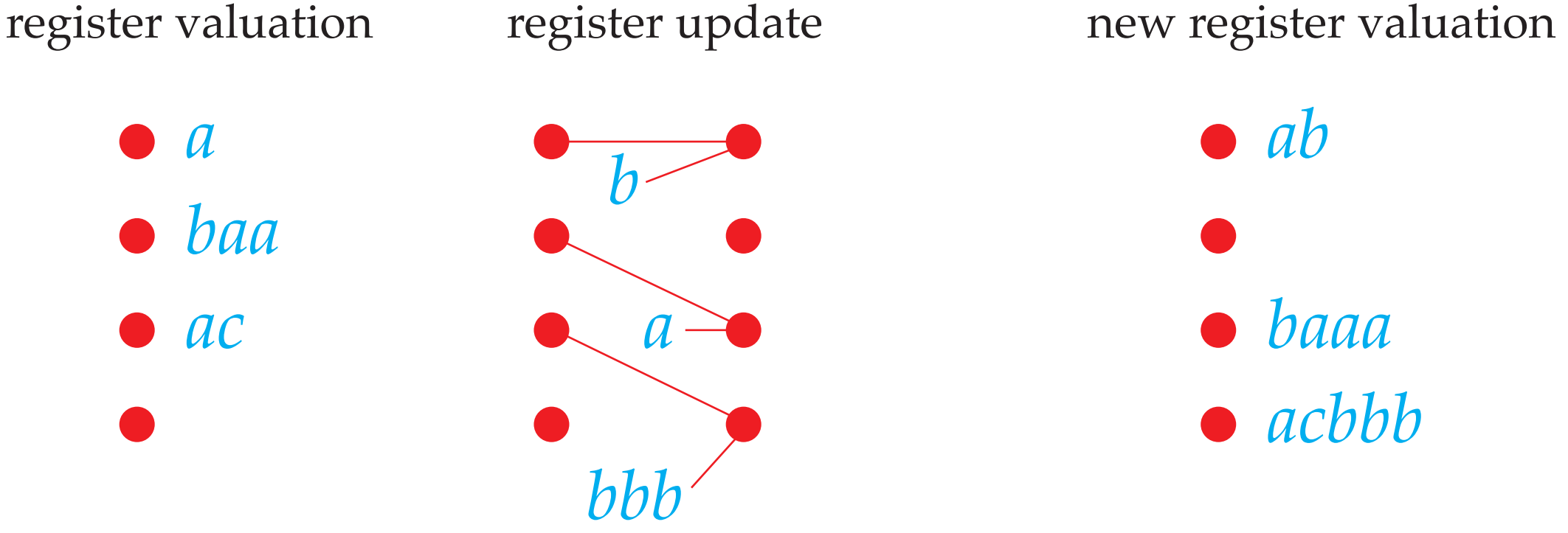
two register updates τ, σ

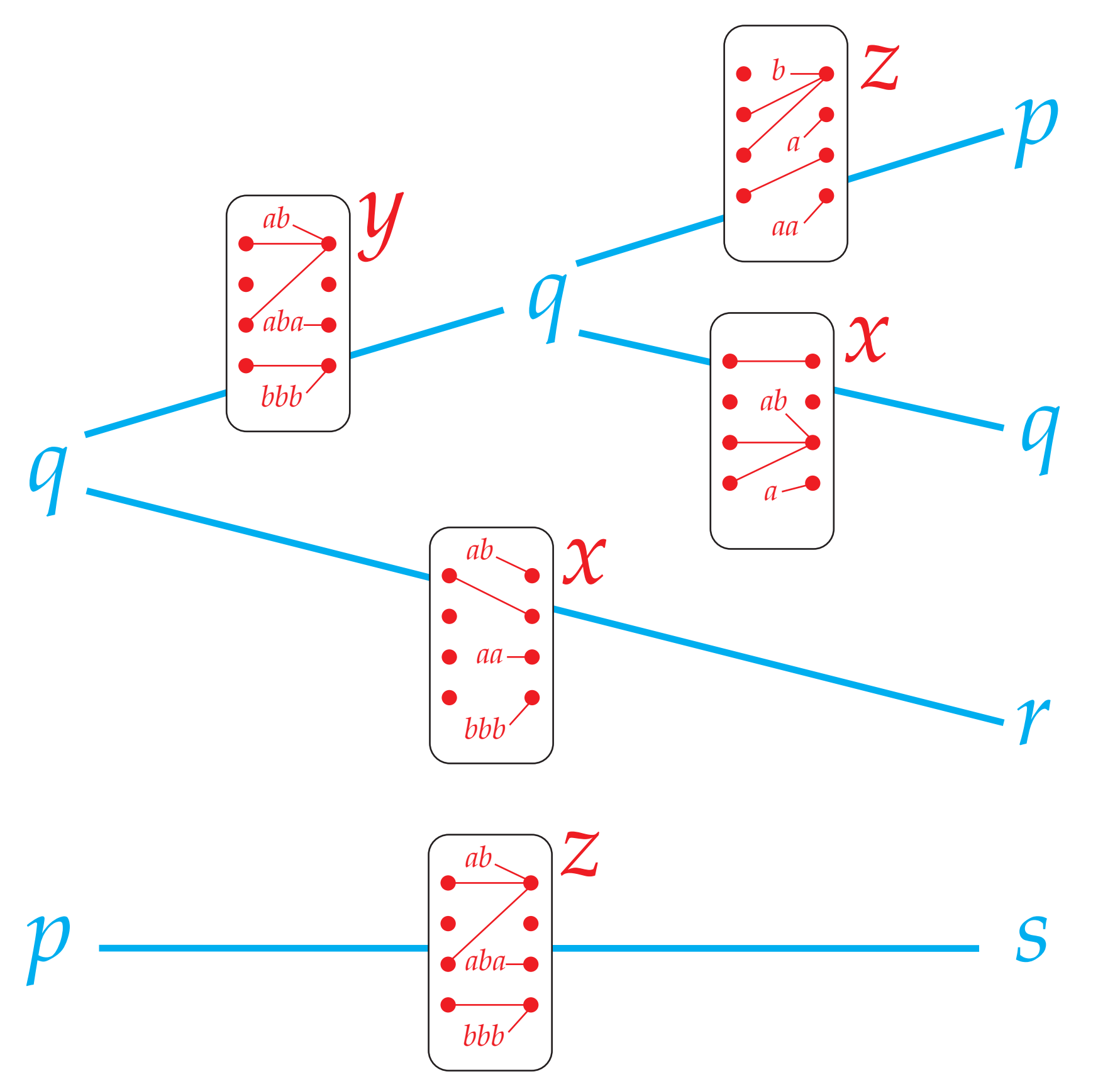


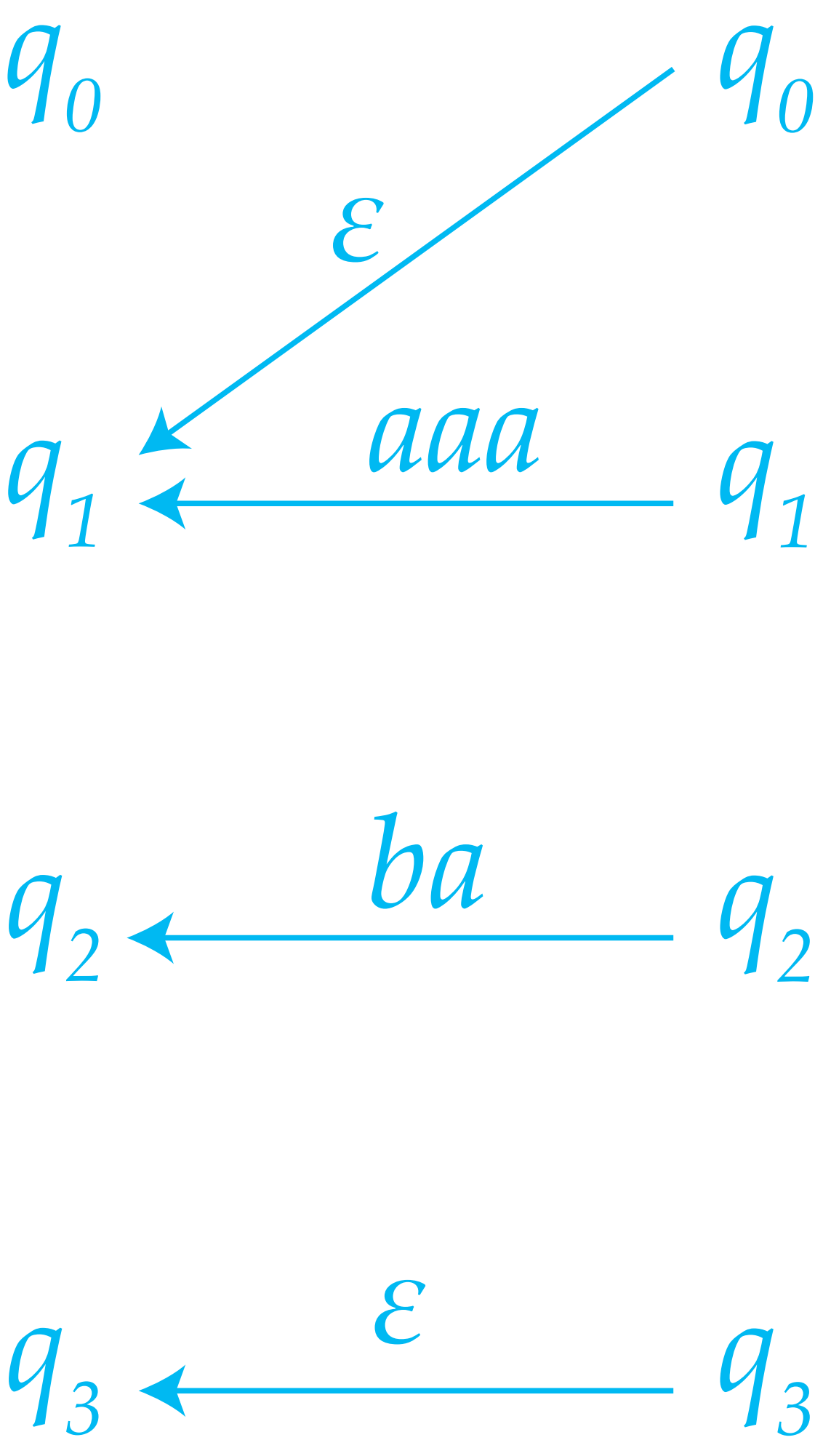


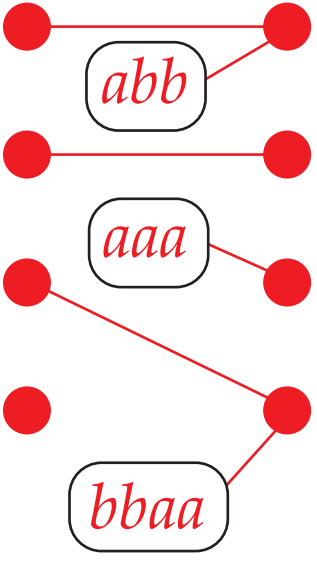
their composition τ·σ




register valuation
register update
new register valuation
a
baa
ac
b
a
bbb
ab
baaa
acbbb

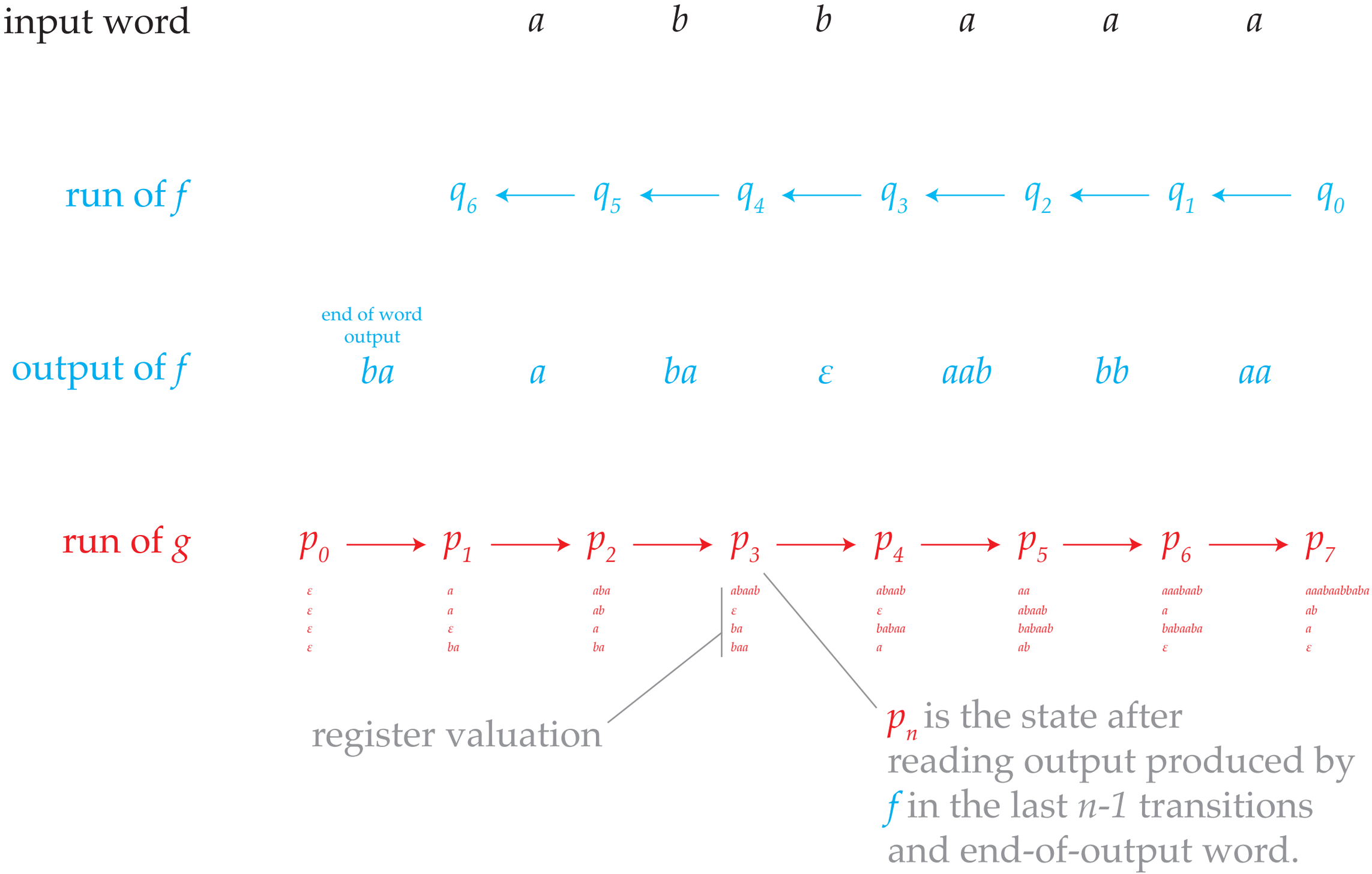

b
z
a
aa
p
ab
y
aba
q
bbb
x
ab
q
a
q
ab
x
aa
bbb
r
ab
z
p
aba
s
bbb

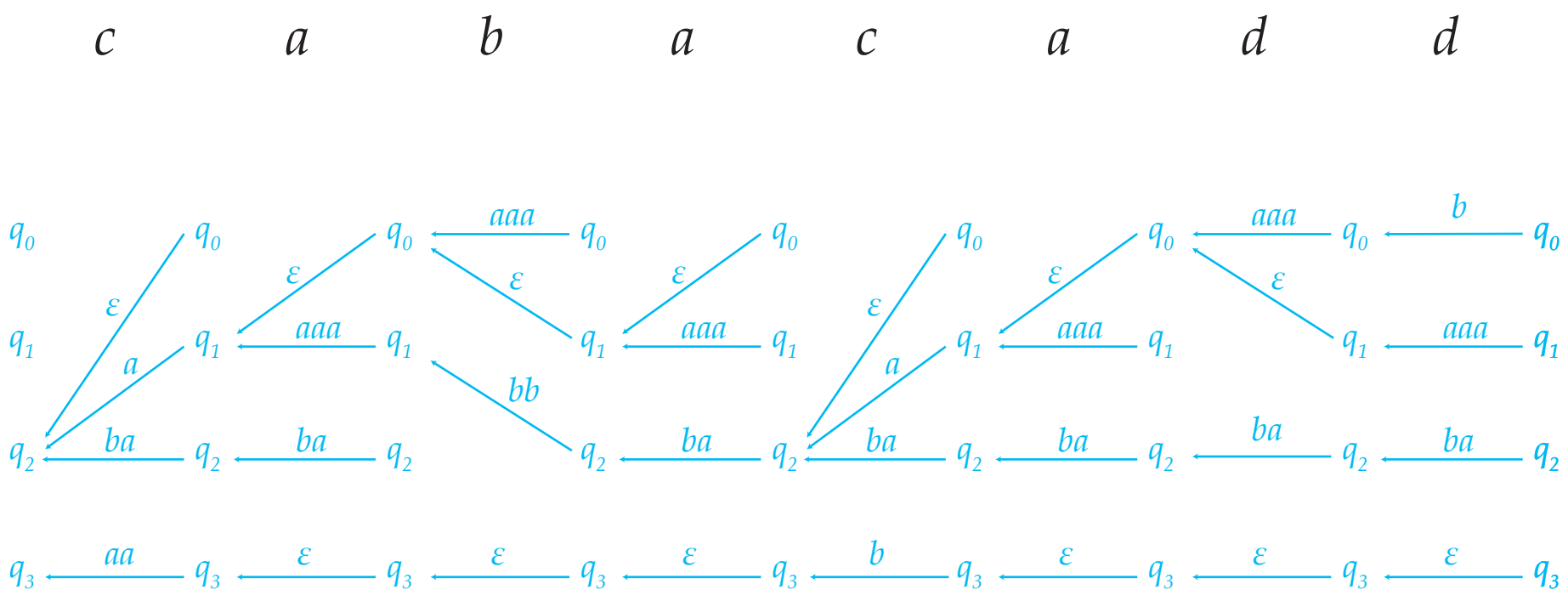

q_0
q_0
ε
aaa
q_1
q_1
ba
q_2
q_2
ε
q_3
q_3

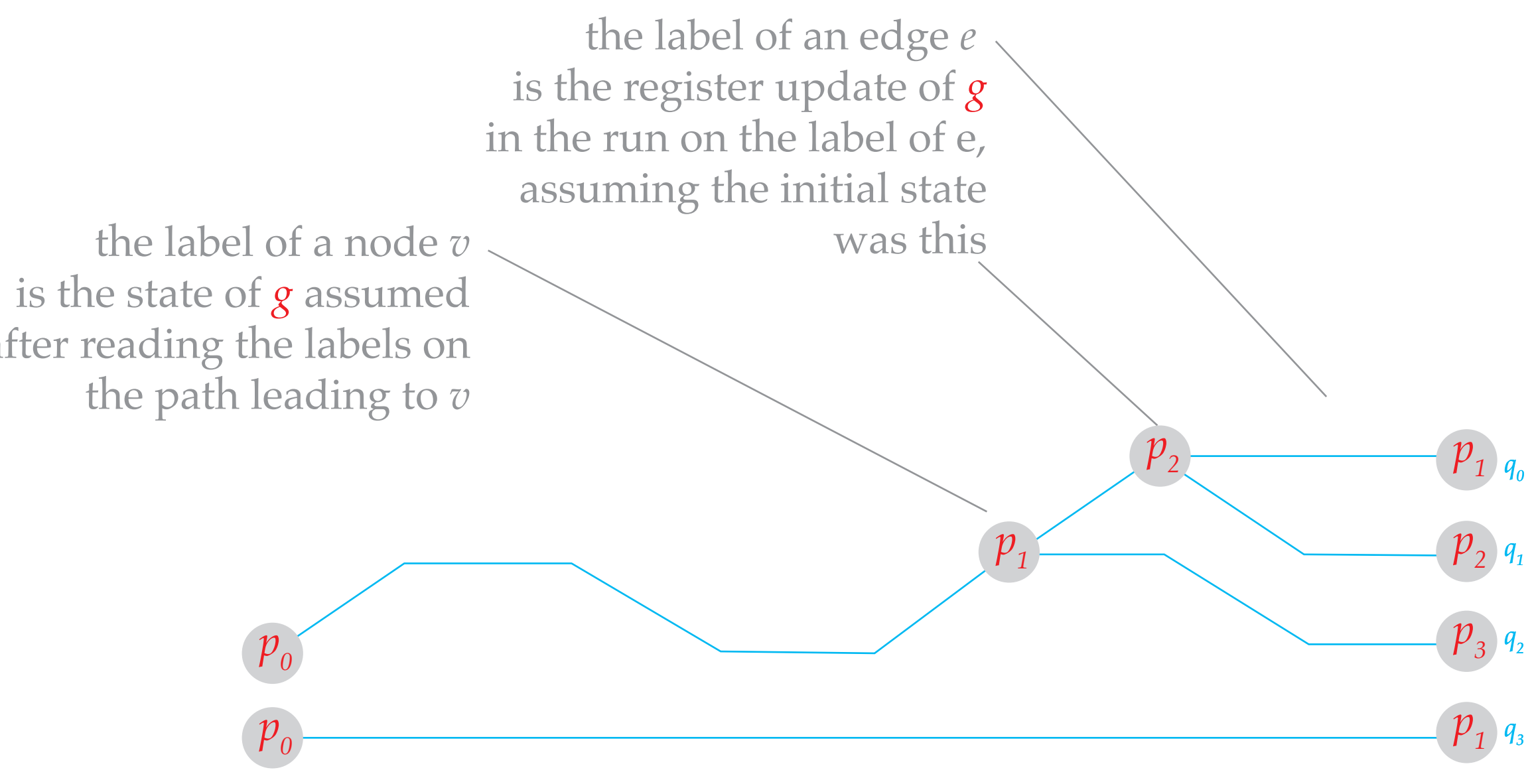


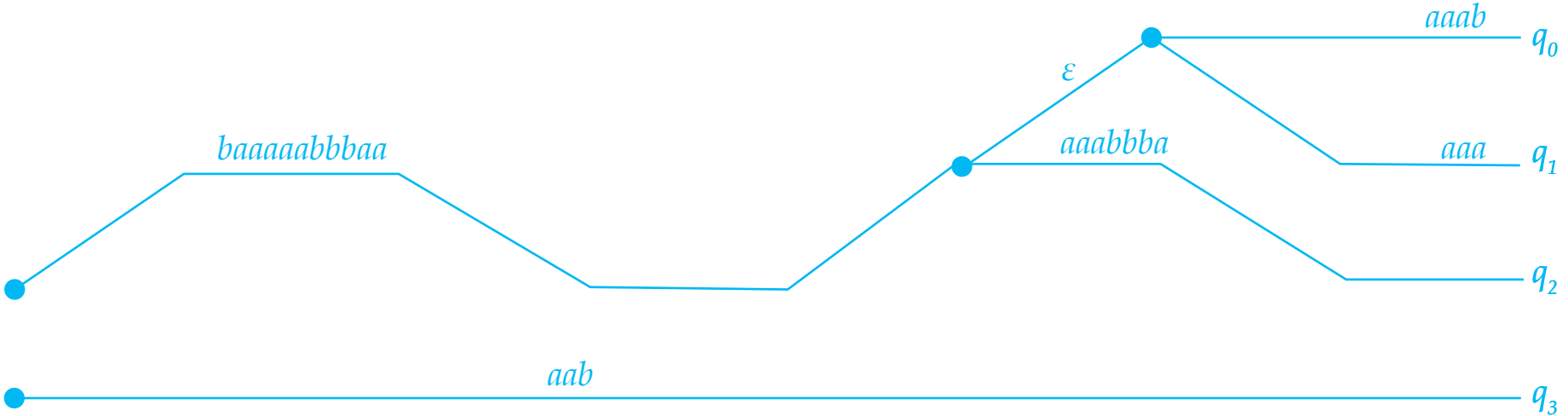
3 registers used to store these words

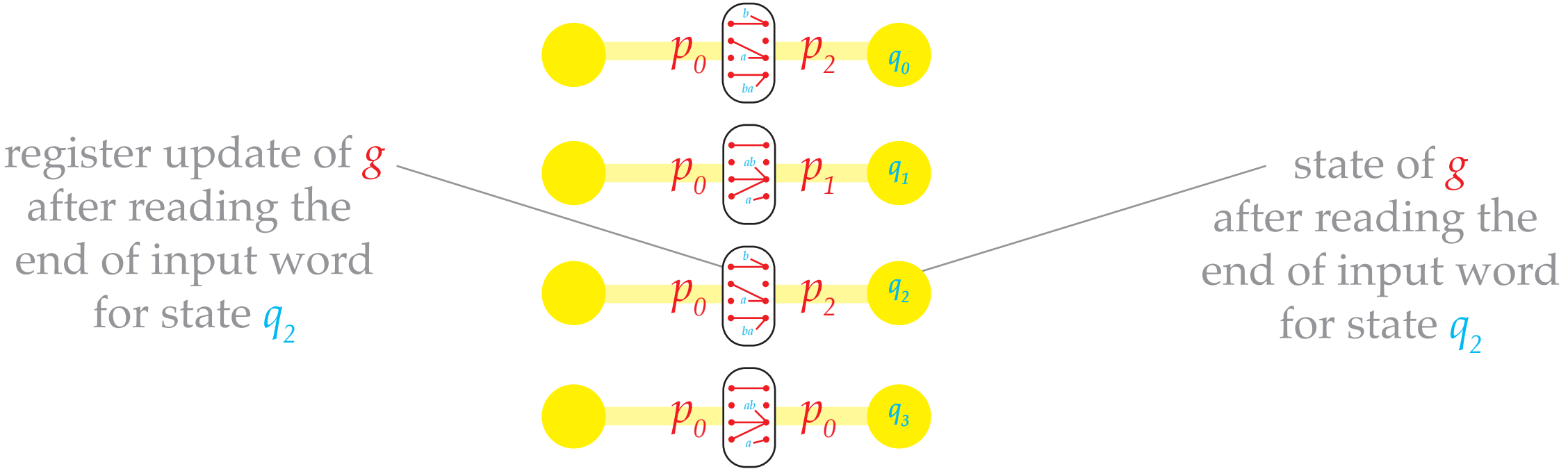
abb aaa bbaa


input word
a b b a a a
run of f
q_6 q_5 q_4 q_3 q_2 q_1 q_0
end of word output
output of f
ba a ba ε aab bb aa
run of g
p_0 p_1 p_2 p_3 p_4 p_5 p_6 p_7
ε ε ε ε
a a ε ba
aba ab a ba
abaab ε ba baa
abaab ε babaa a
aa abaab babaab ab
aaabaab a babaaba ε
aaabaabbbaba ab a ε
register valuation
p_n is the state after reading output produced by f in the last n-1 transitions and end-of-output word.

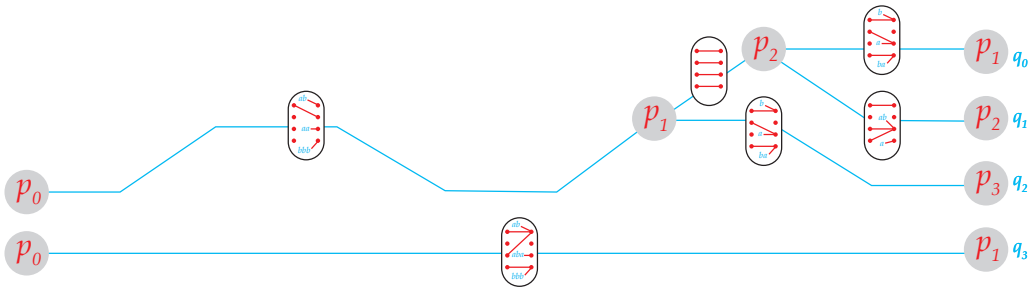

c a b a c a d d
q_0 q_1 q_2 q_3
ε a ba aa
aaa bb b


the label of an edge e
is the register update of g
in the run on the label of e,
assuming the initial state
was this
the label of a node v
is the state of g assumed
after reading the labels on
the path leading to v
p_0
p_0
p_1
p_2
p_1
p_2
p_3
p_1
q_0
q_1
q_2
q_3

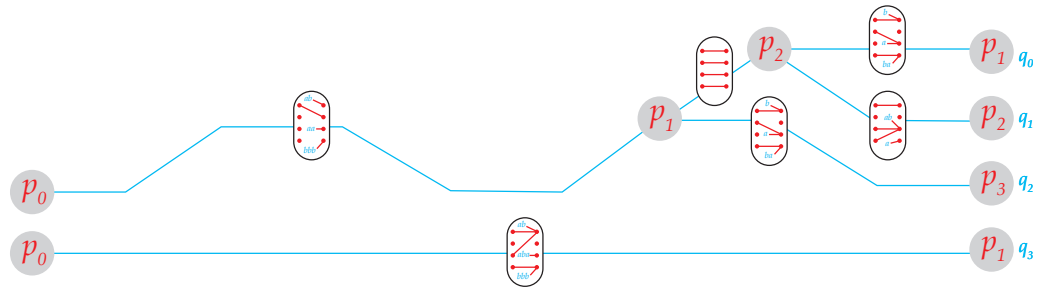

aaab
q_0
ε
baaaaabbbaa
aaabbba
aaa
q_1
q_2
aab
q_3


register update of g after reading the end of input word for state q_2
p_0
p_2
q_0
b
a
ba
p_0
p_1
q_1
ab
a
p_0
p_2
q_2
b
a
ba
p_0
p_0
q_3
ab
a
state of g after reading the end of input word for state q_2

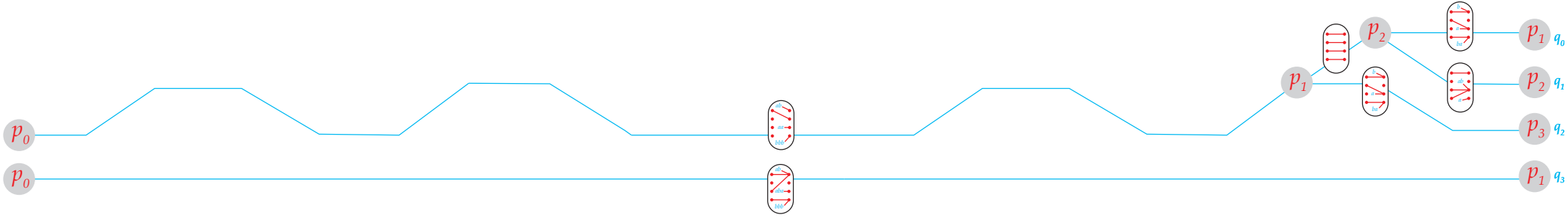
update forest s

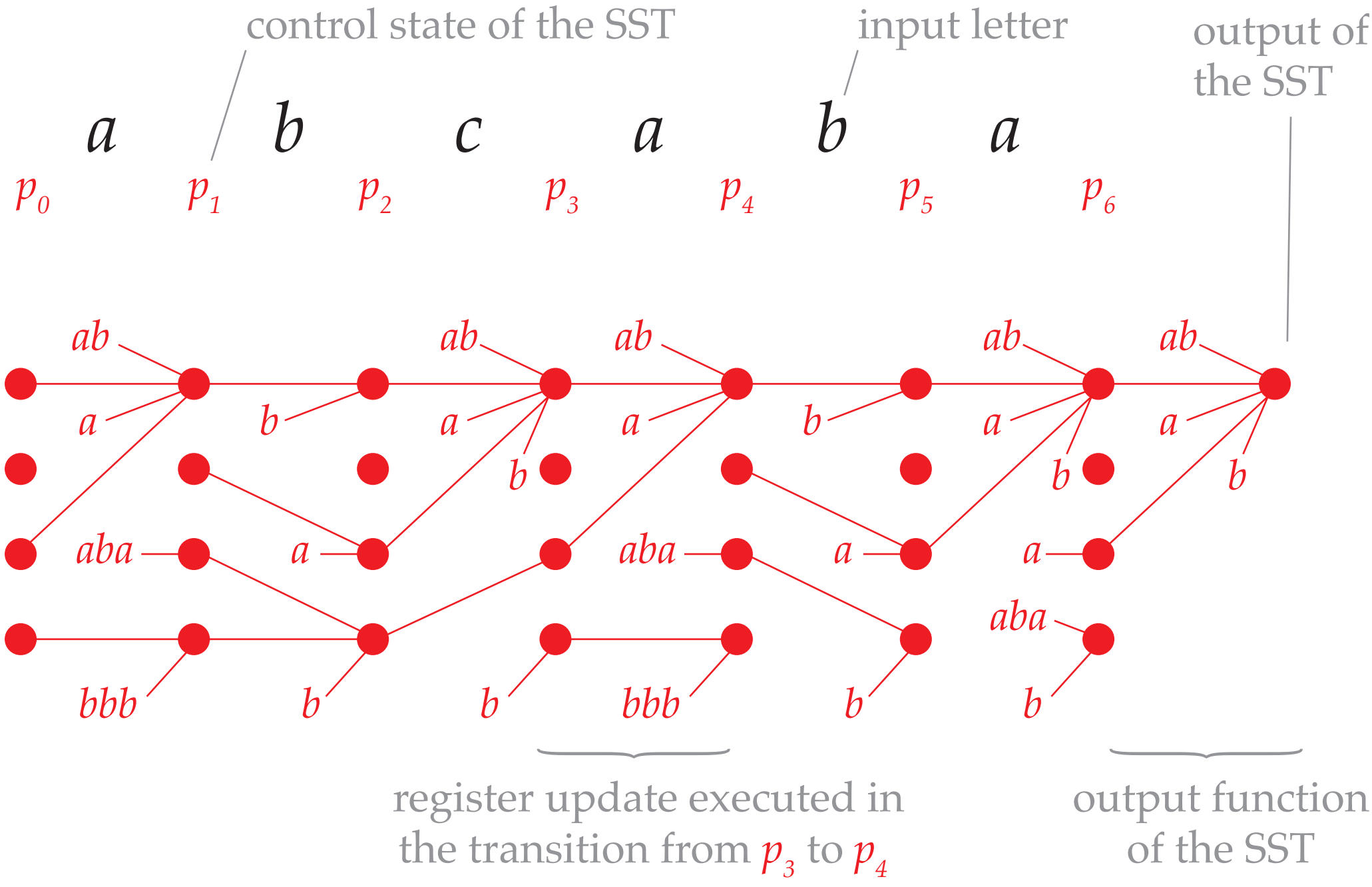


update forest t

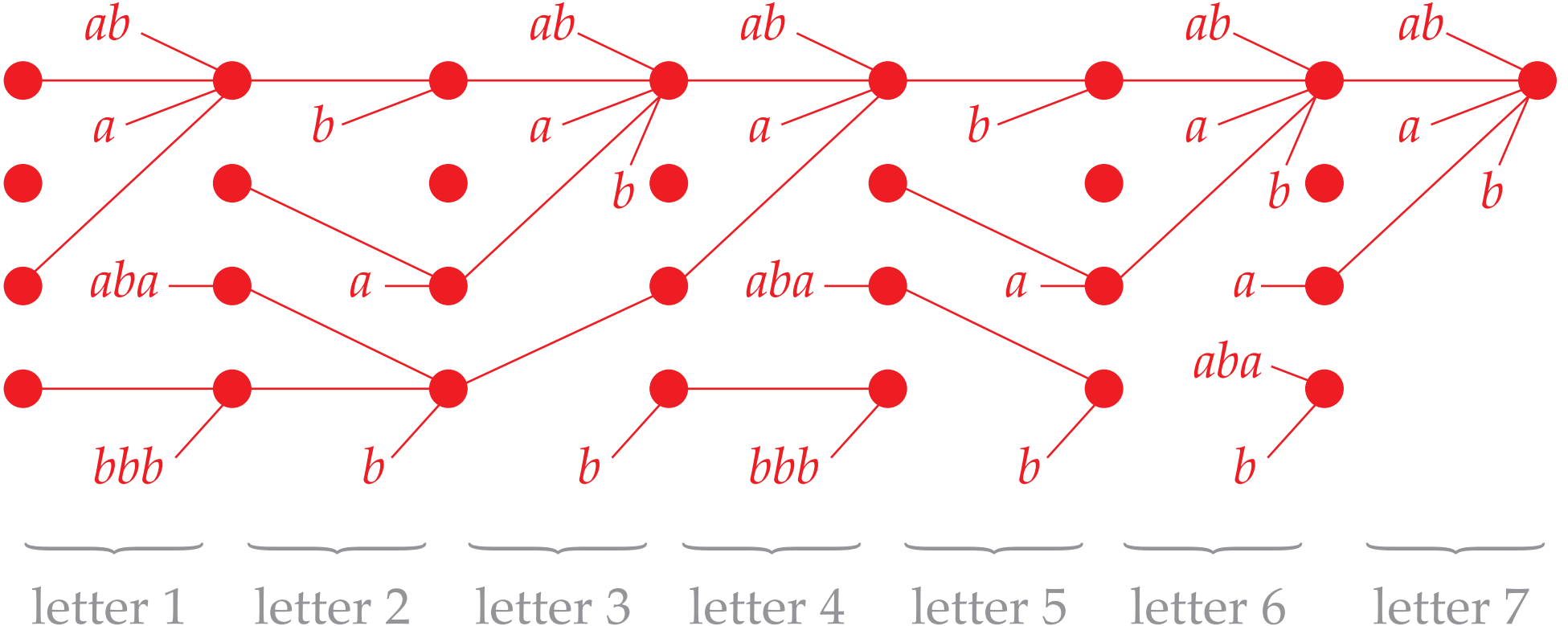


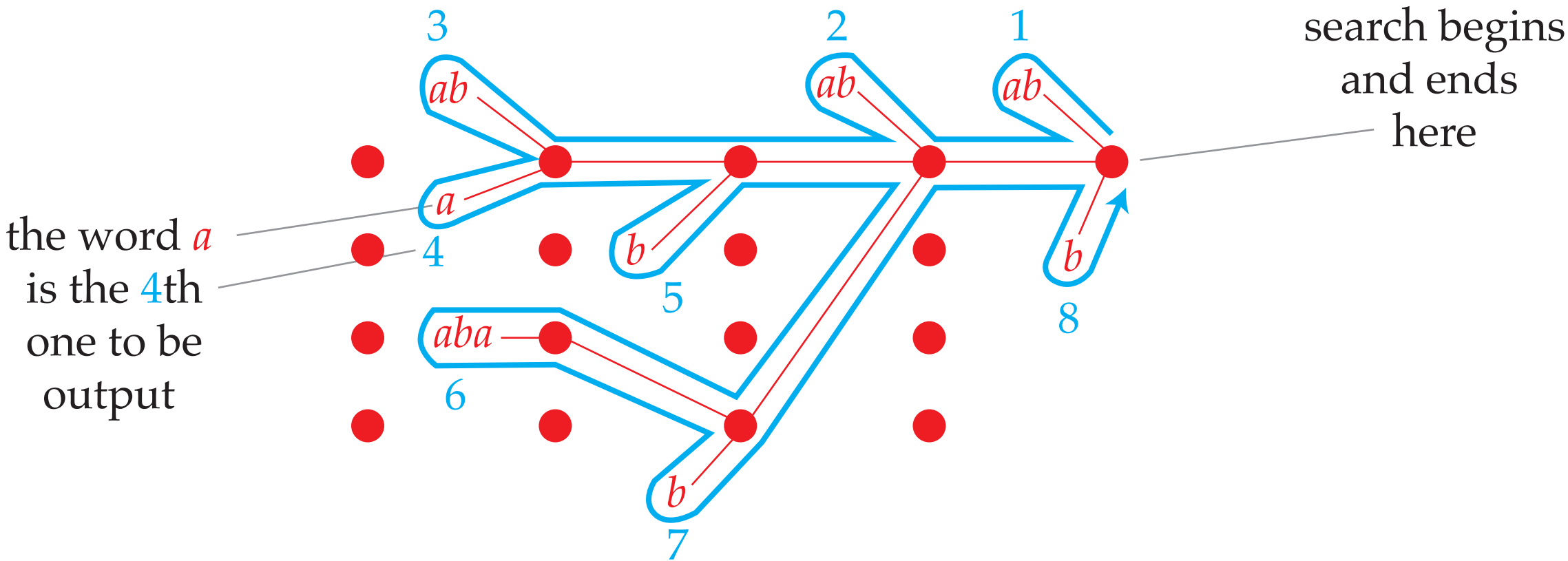
their composition st



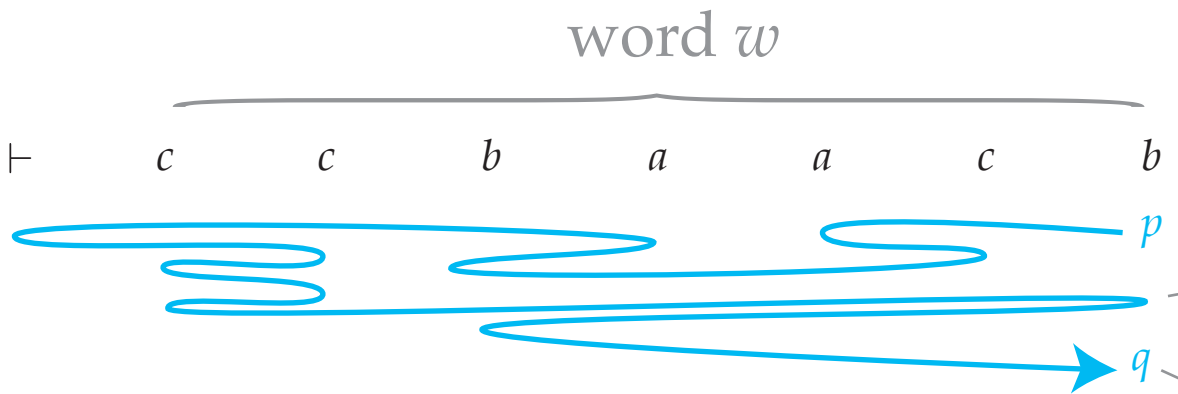

control state of the SST
input letter
output of the SST
a b c a b a
p_0 p_1 p_2 p_3 p_4 p_5 p_6
ab
a
b
aba
bbb
register update executed in the transition from p_3 to p_4
output function of the SST

a *b* *c* *a* *b* *a*


ab
a
aba
bbb
b
ab
b
a
b
ab
a
b
b
ab
a
aba
bbb
b
ab
b
a
a
ab
a
b
aba
b
letter 1
letter 2
letter 3
letter 4
letter 5
letter 6
letter 7


3
2
1
search begins
and ends
here
ab
ab
ab
a
b
b
4
5
8
the word a
is the 4th
one to be
output
aba
6
b
7

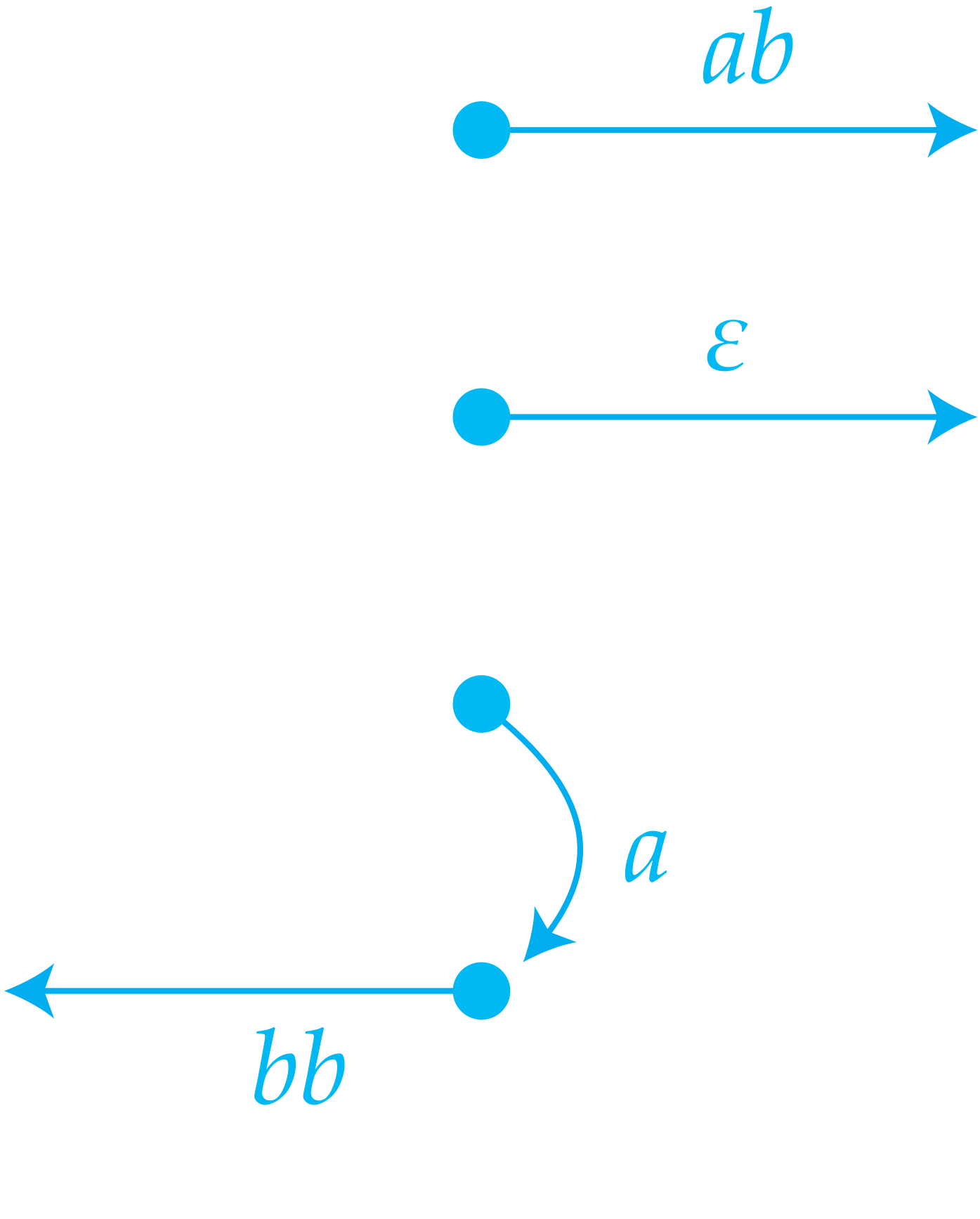
c *b* *a* *a* *c* *b* *a*

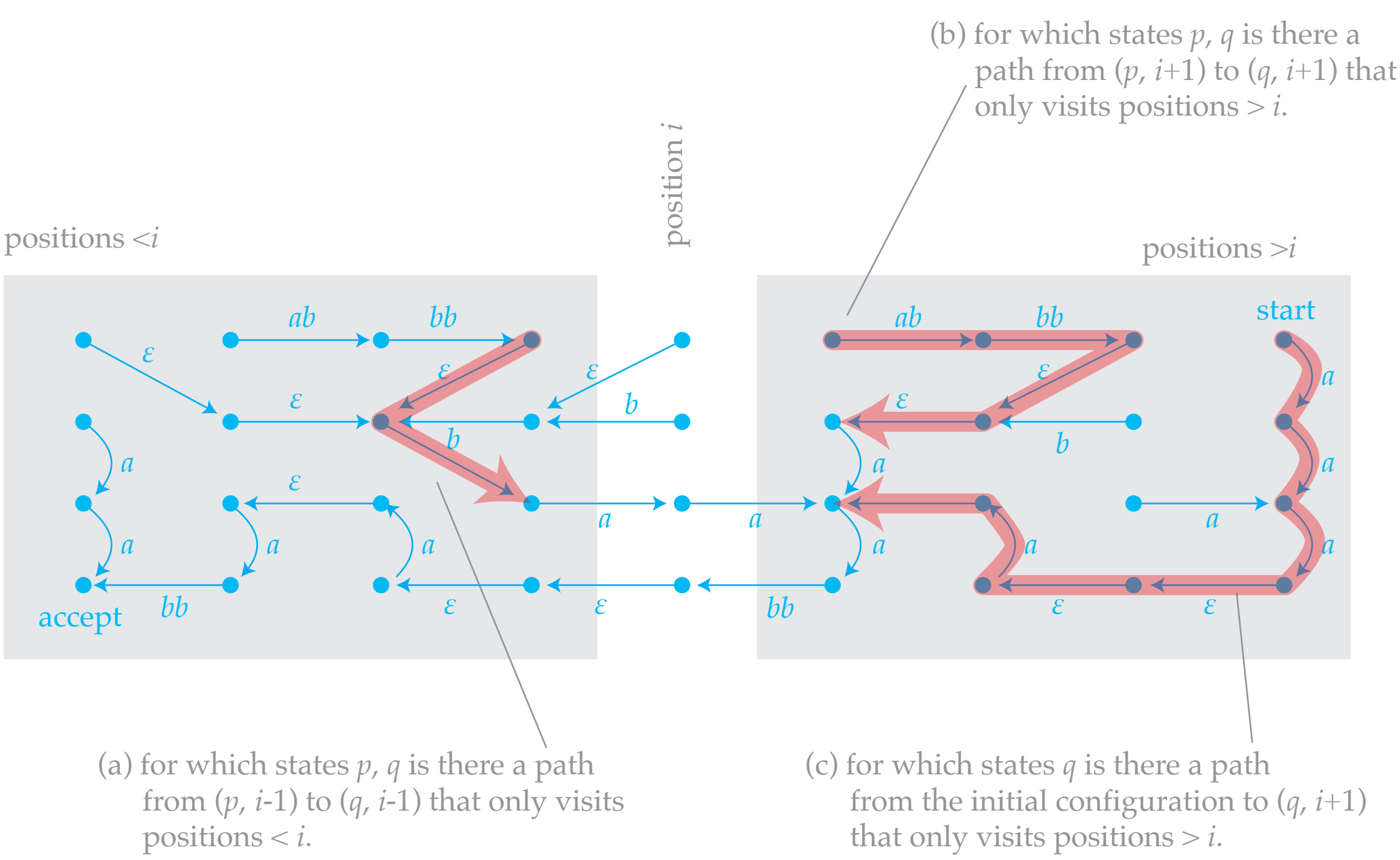


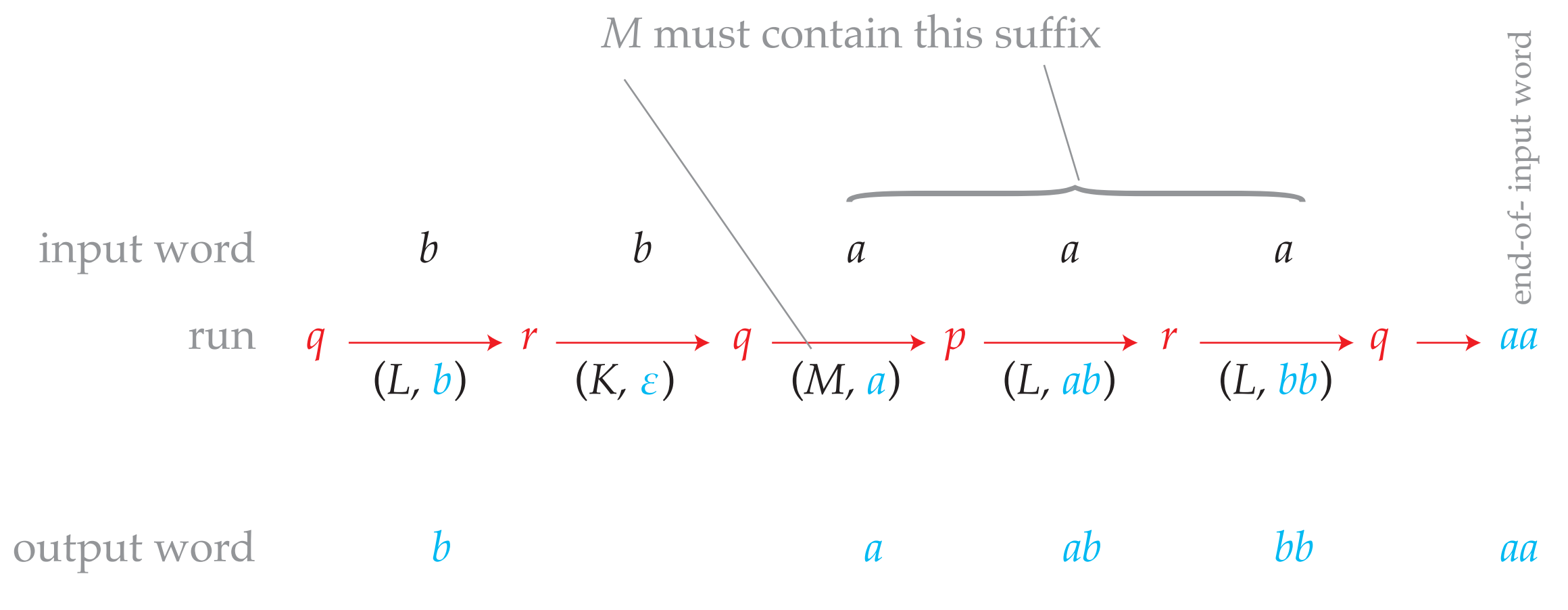
begins in p in last position

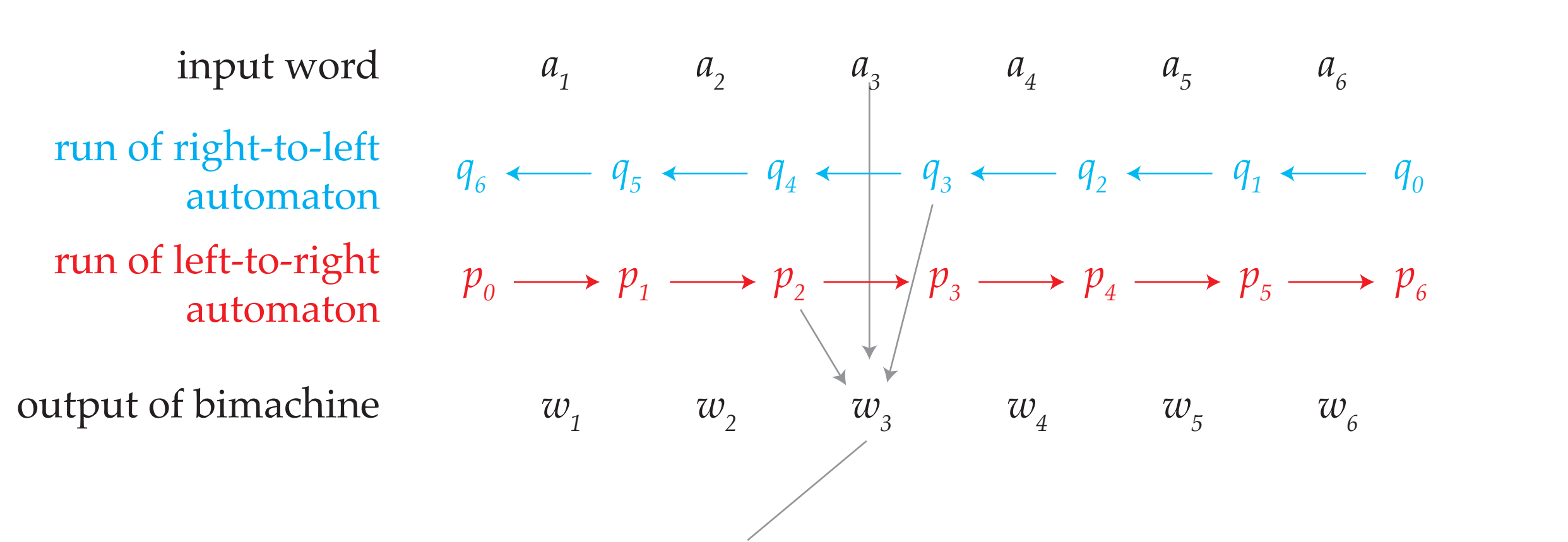
is allowed to return to last position, but to move right in the last position

ends in q in last position

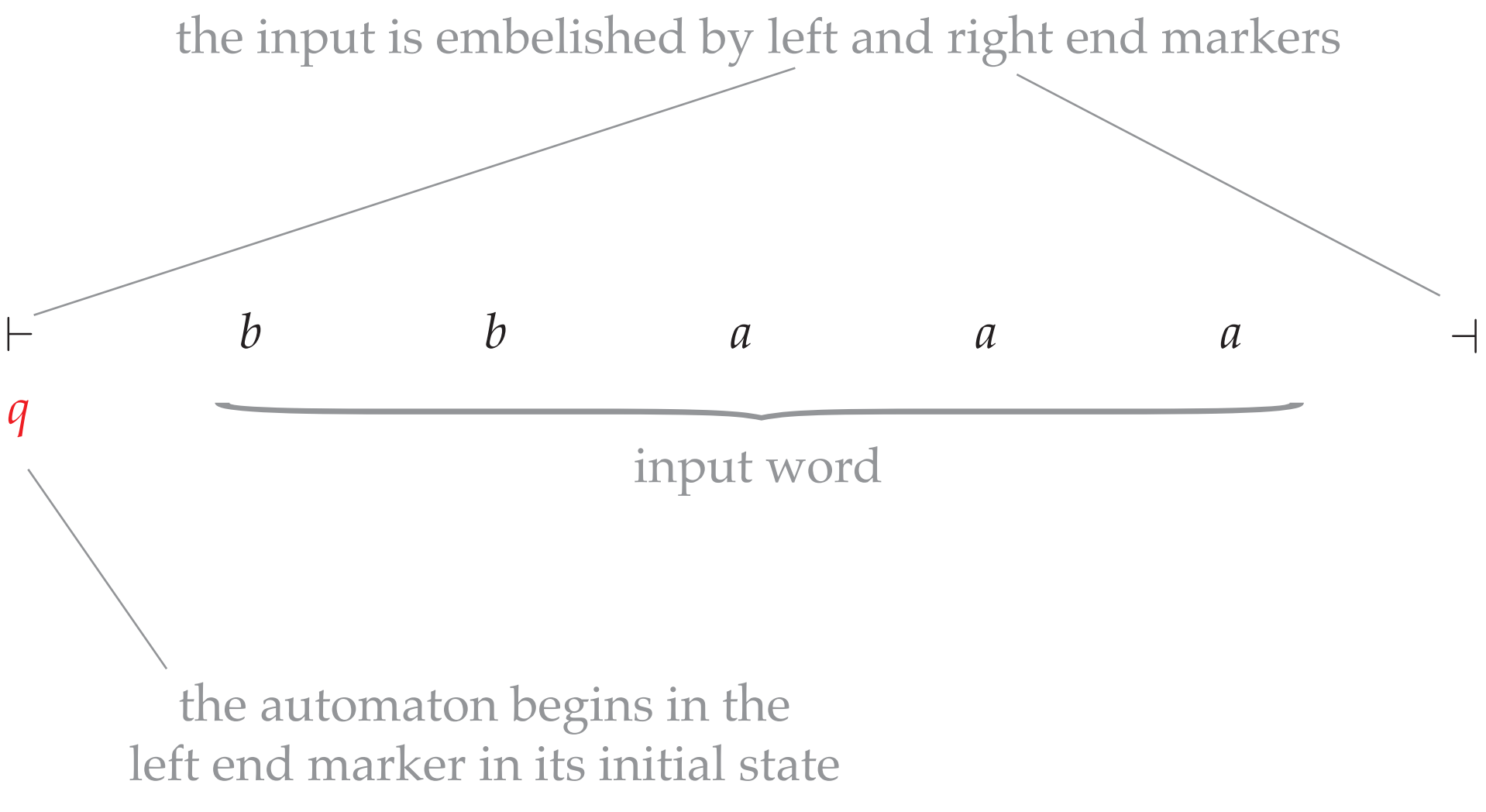

ab
ε
a
bb

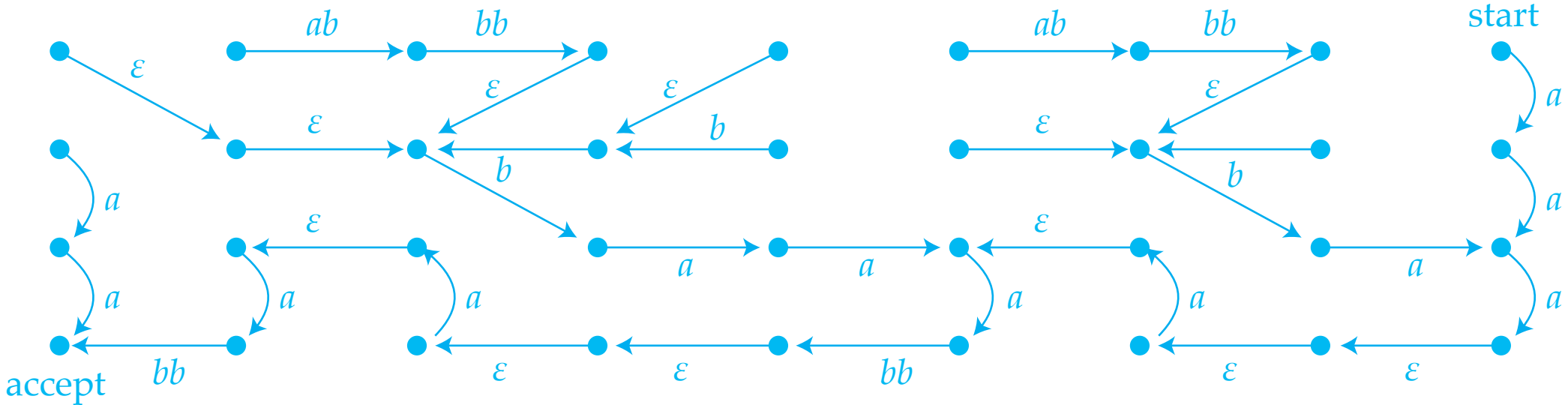

(b) for which states p, q is there a path from (p, i+1) to (q, i+1) that only visits positions > i.
positions <i
position i
positions >i
ab
bb
start
ε
b
a
accept
bb
(a) for which states p, q is there a path from (p, i-1) to (q, i-1) that only visits positions < i.
(c) for which states q is there a path from the initial configuration to (q, i+1) that only visits positions > i.

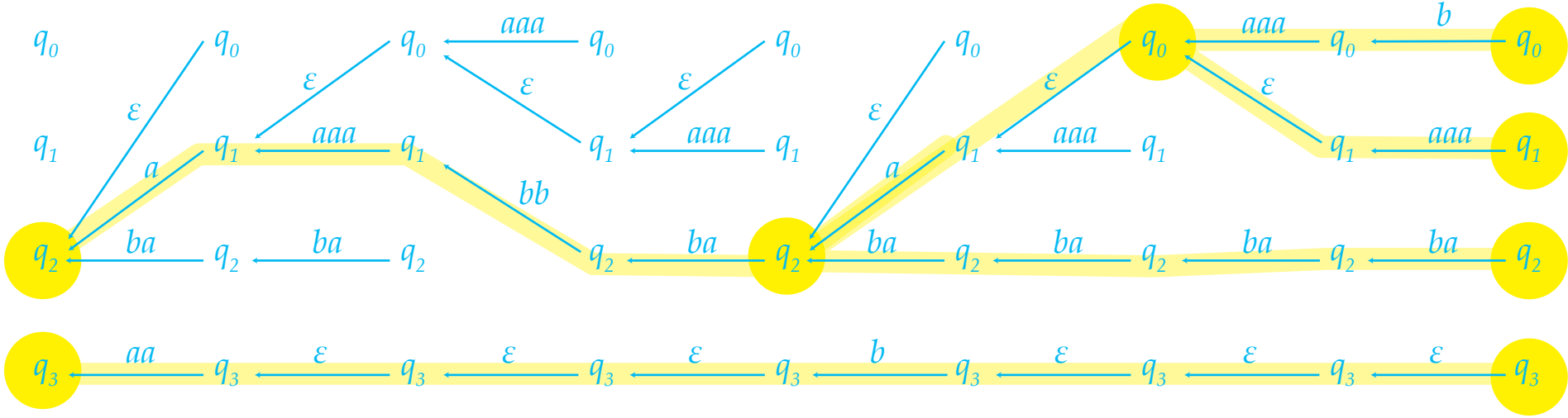

M must contain this suffix
input word
b
b
a
a
a
end-of- input word
run
q
r
q
p
r
q
aa
(L, b)
(K, ε)
(M, a)
(L, ab)
(L, bb)
output word
b
a
ab
bb
aa



the i-th output word is the value of the output function on
- the state of left-to-right automaton after reading letters $<i$
- the i-th letter
- the state of right-to-left automaton after reading letters $>i$


the input is embelished by left and right end markers
⊢
b
b
a
a
a
⊣
q
input word
the automaton begins in the
left end marker in its initial state

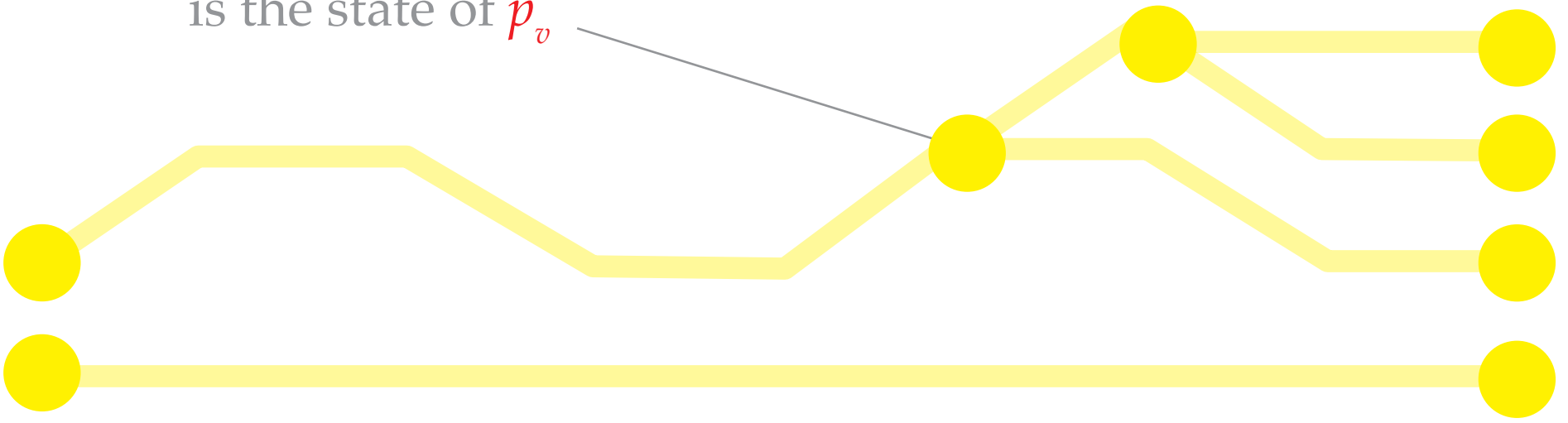

ab
bb
ε
ε
ε
ε
b
b
a
ε
a
a
a
a
a
a
a
a
bb
ε
ε
bb
accept
ab
bb
start
ε
ε
b
a
a
ε
a
a
a
a
ε
ε


q_0
q_0
q_0
aaa
q_0
q_0
q_0
q_0
q_0
b
q_0
ε
ε
ε
ε
ε
ε
q_1
q_1
a
q_1
aaa
q_1
q_1
aaa
q_1
q_1
aaa
q_1
q_1
aaa
q_1
bb
q_2
ba
q_2
ba
q_2
q_2
ba
q_2
q_2
ba
q_2
q_2
ba
q_2
q_2
ba
q_2
q_3
aa
q_3
ε
q_3
ε
q_3
ε
q_3
b
q_3
ε
q_3
ε
q_3
ε
q_3

the label of an edge
that corresponds to a path π
is the register update τ_π

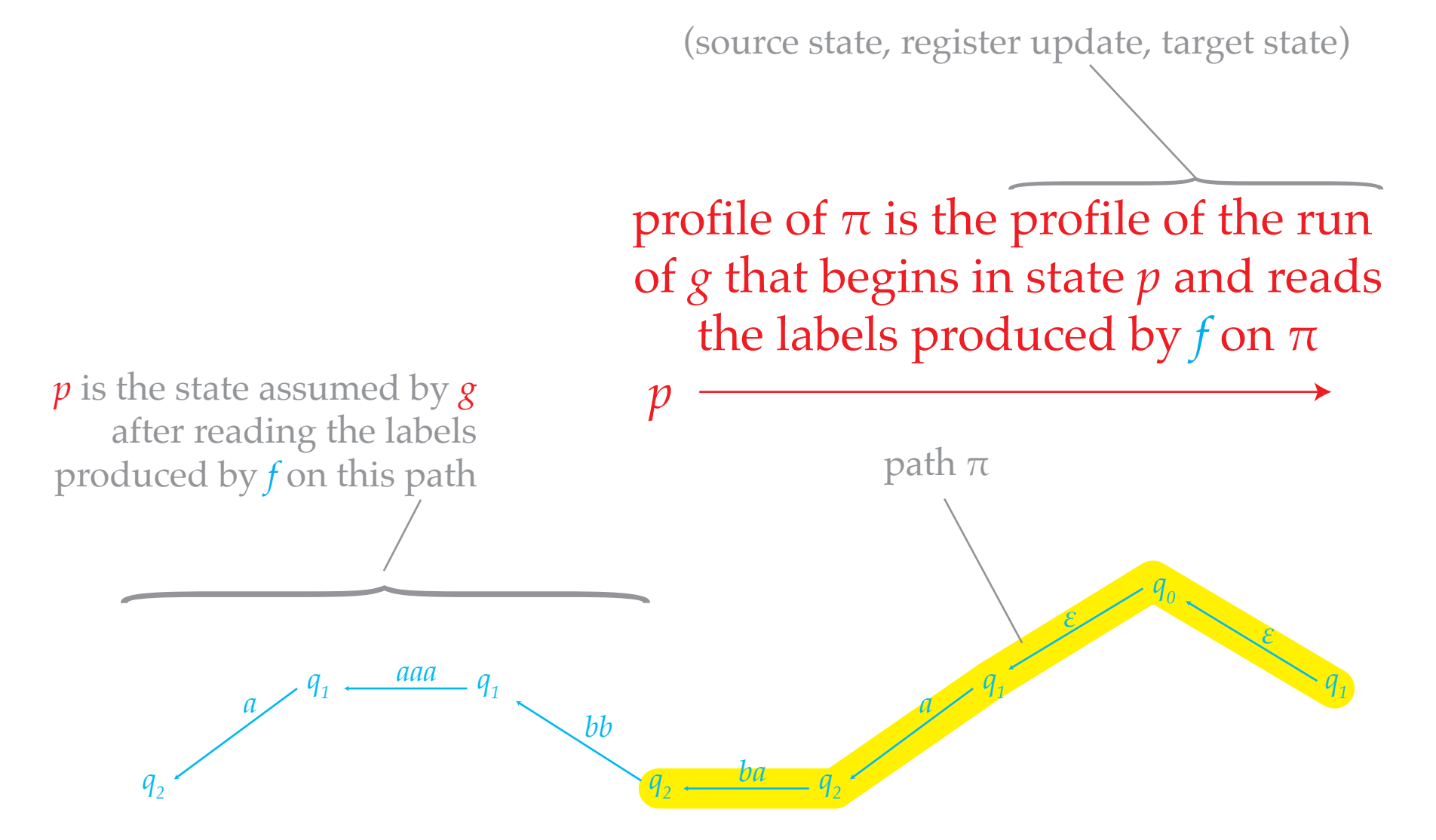
for leaves, we also
remember their
state in f

the label a node v
is the state of p_v



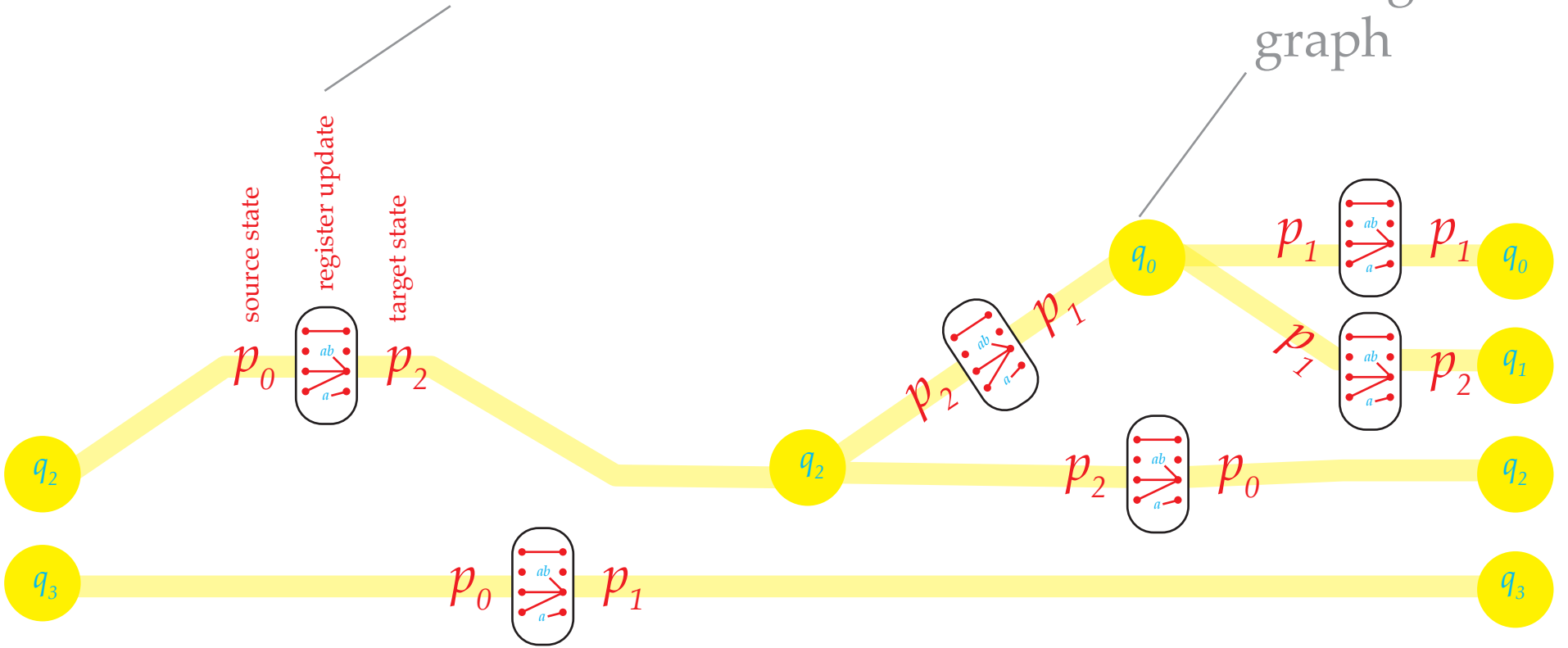
τ

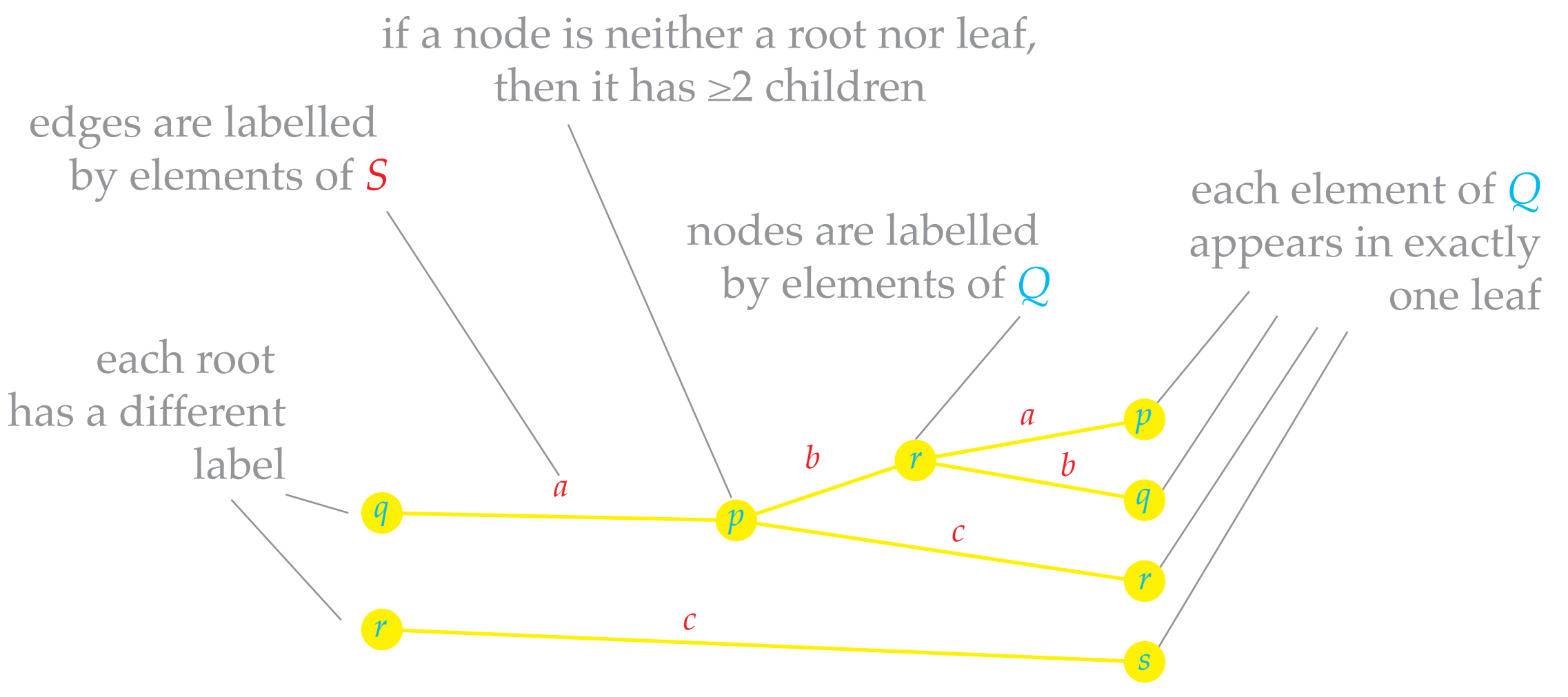
σ


(source state, register update, target state)
profile of π is the profile of the run of g that begins in state p and reads the labels produced by f on π
p
p is the state assumed by g after reading the labels produced by f on this path
path π
q0
ε
ε
q1
q1
q1
a
aaa
a
bb
q2
ba
q2
q2

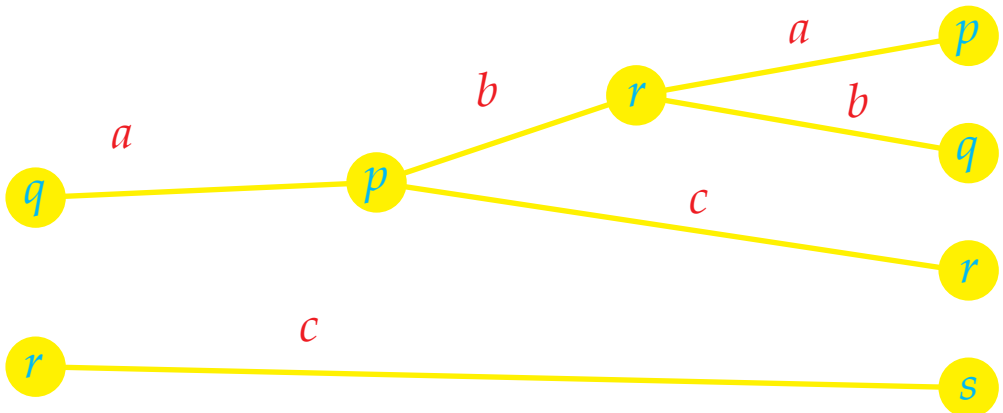
each edge, which correponds to a path π connecting two important nodes in the configuration graph, is labelled by the profile of π

nodes are the important nodes with their labels from the configuration graph

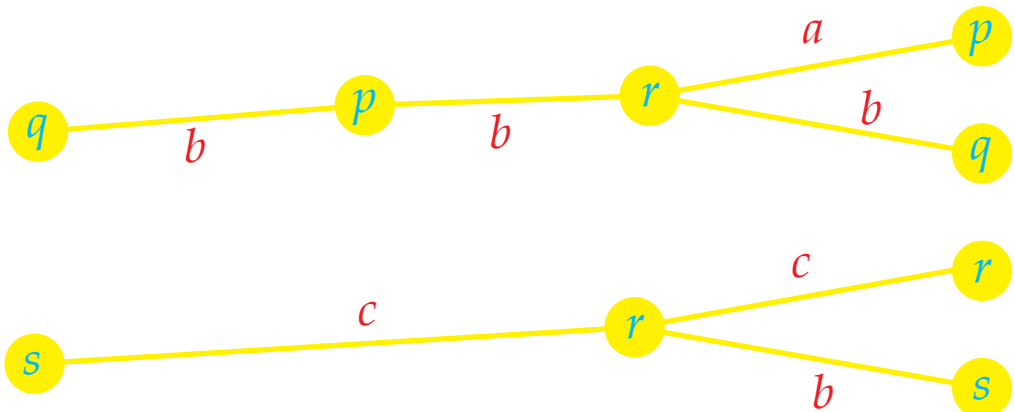


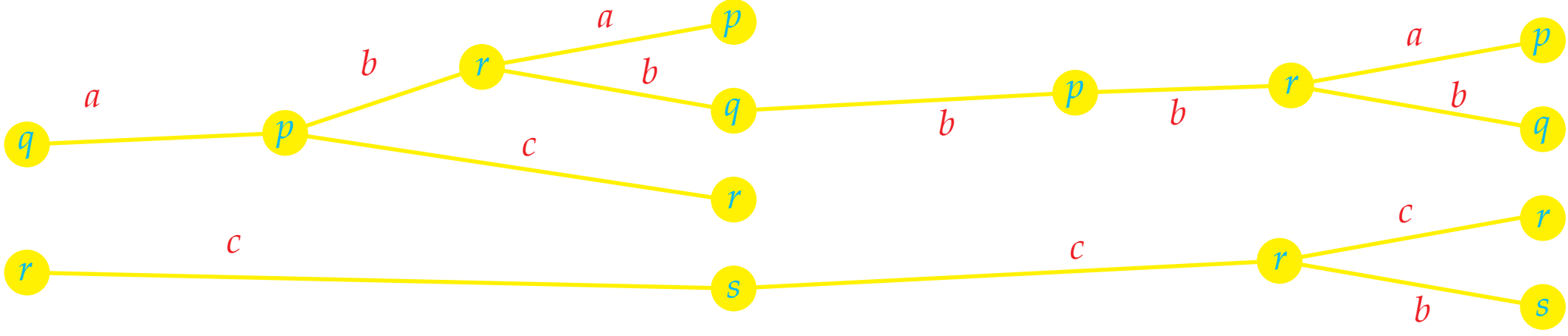

if a node is neither a root nor leaf,
then it has ≥2 children
edges are labelled
by elements of S
nodes are labelled
by elements of Q
each element of Q
appears in exactly
one leaf
each root
has a different
label
q
r
p
r
p
q
r
s
a
c
b
a
b
c

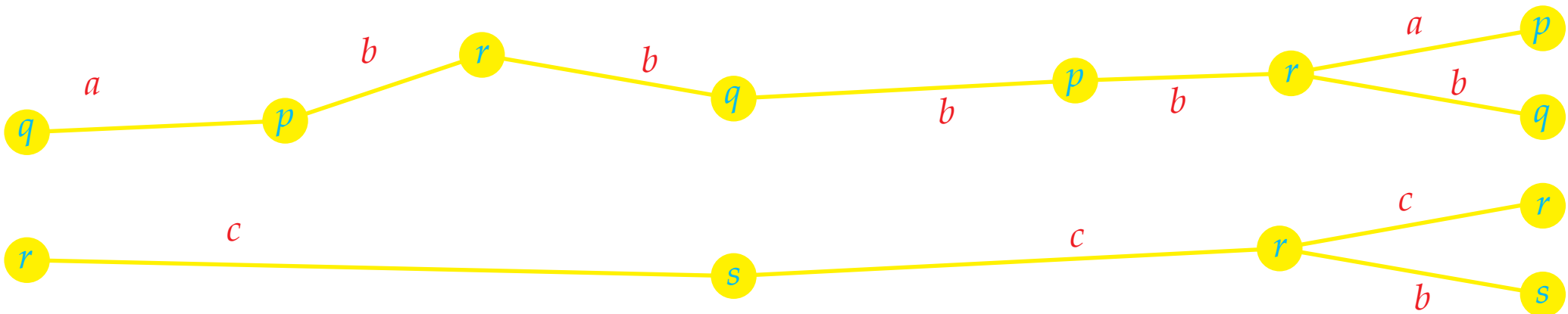
τ

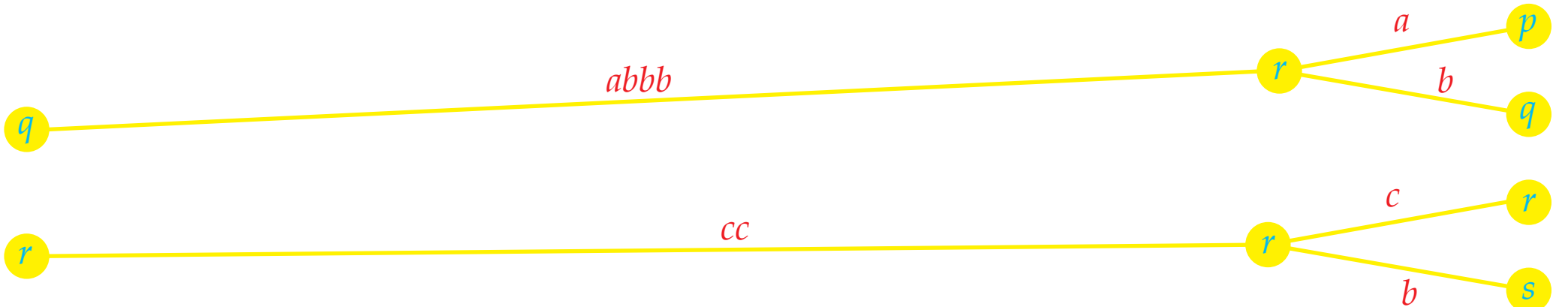


σ



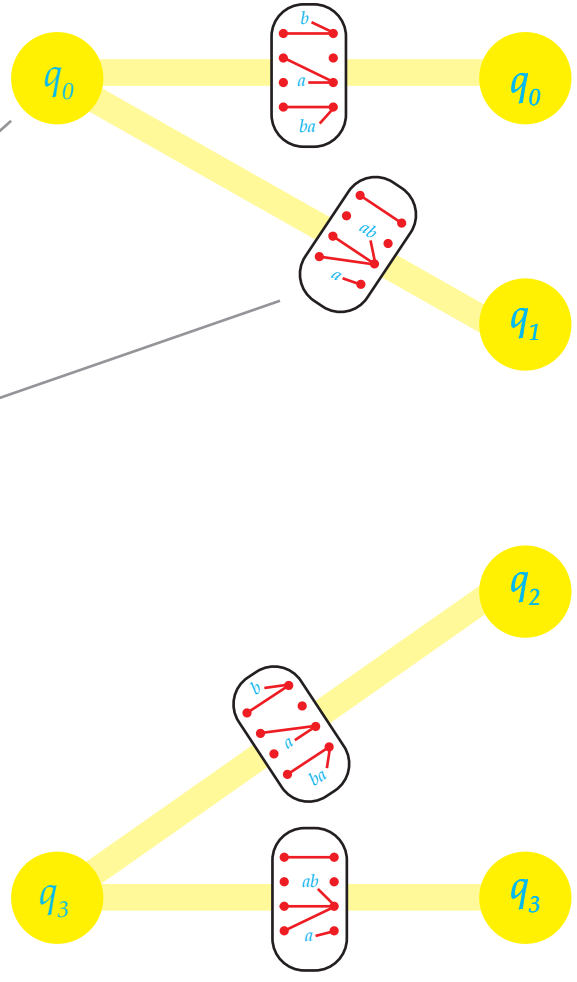

q
a
p
b
r
a
p
b
q
c
r
r
c
s
p
b
r
a
p
b
q
r
c
r
c
b
s


q
a
p
b
r
b
q
b
p
b
r
a
p
b
q
r
c
s
c
r
c
r
b
s


q
abbb
r
a
p
b
q
r
cc
r
c
r
b
s

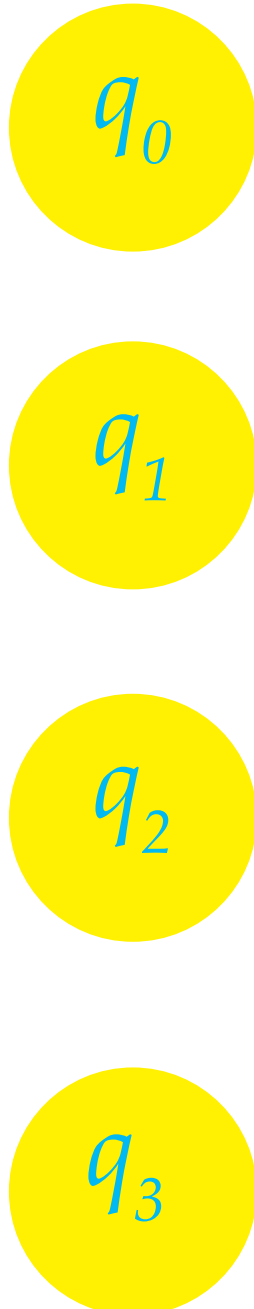
the parent of a leaf with label q
is a root with label aq

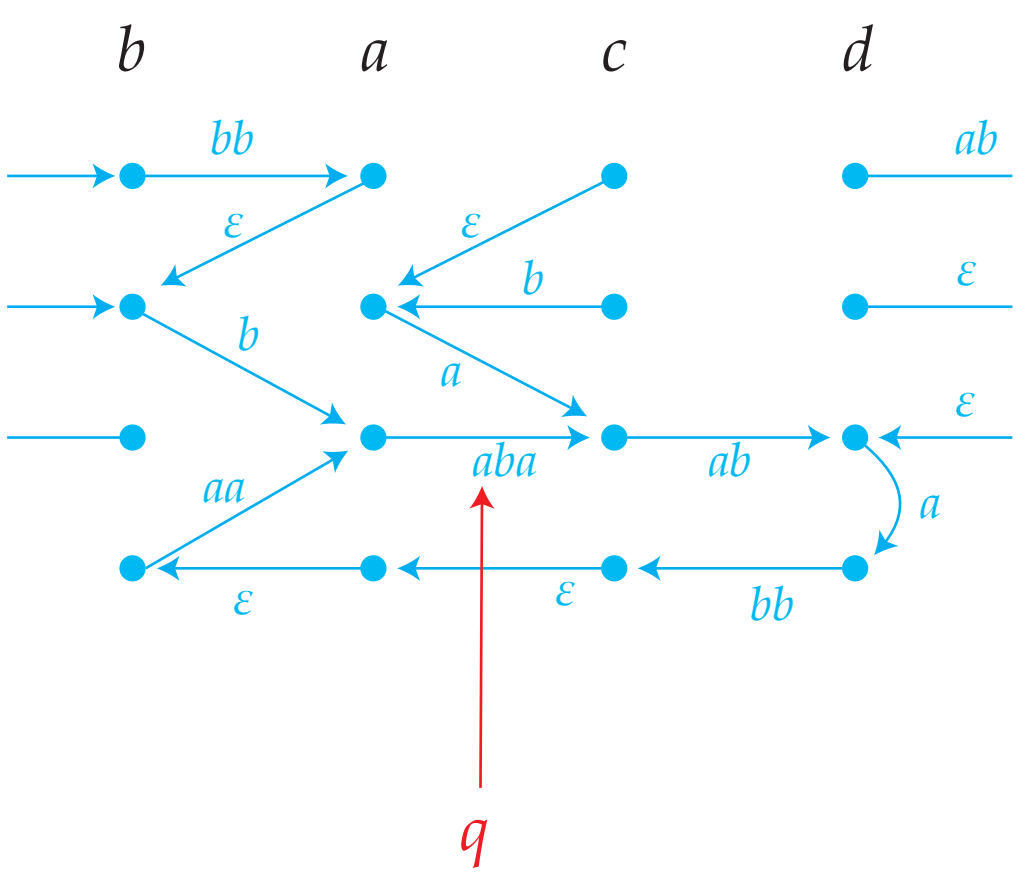
the edge to a leaf with label q
is labelled by the register update
done by the run $[\delta_w(aq), a, q]$



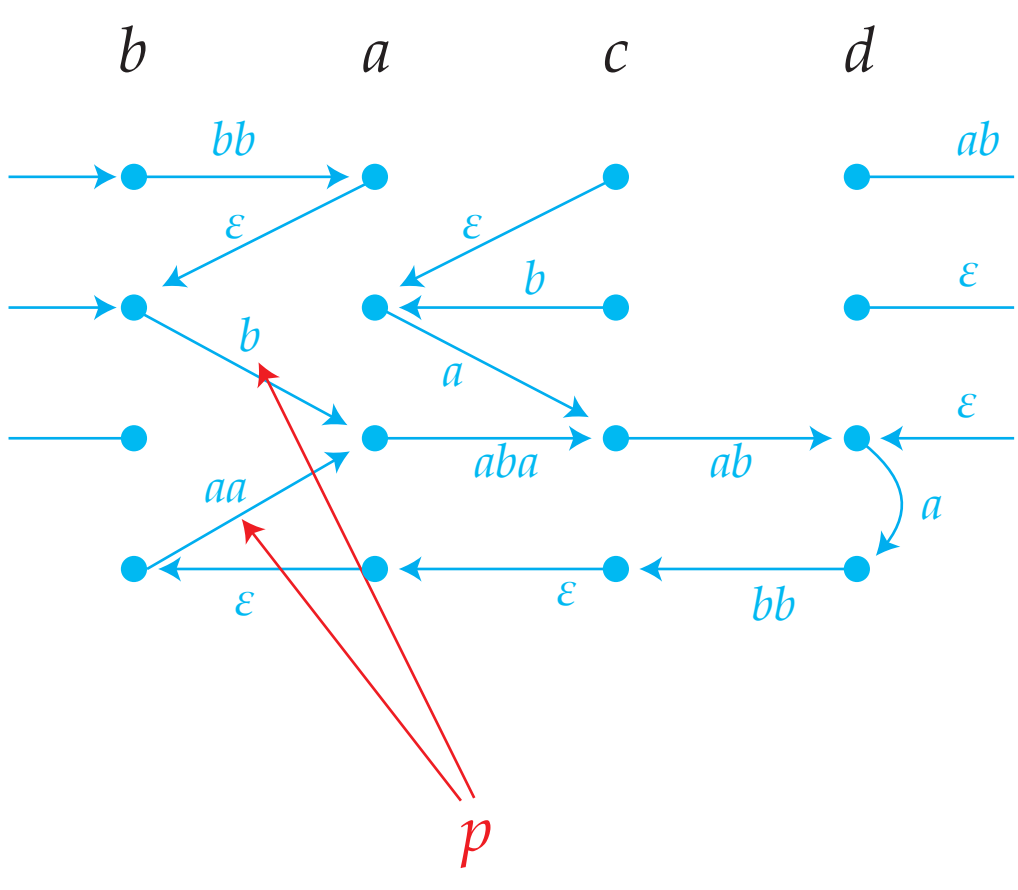
a leaf for
each state of f

every leaf is also a root, and there are no edges





state of the automaton g


b
a
c
d
bb
ab
ε
ε
ε
b
b
a
aba
ab
ε
a
aa
ε
ε
bb
p