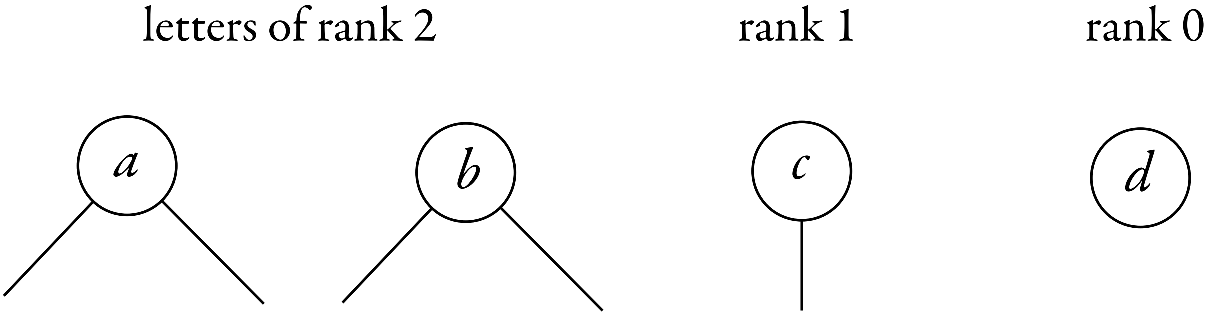
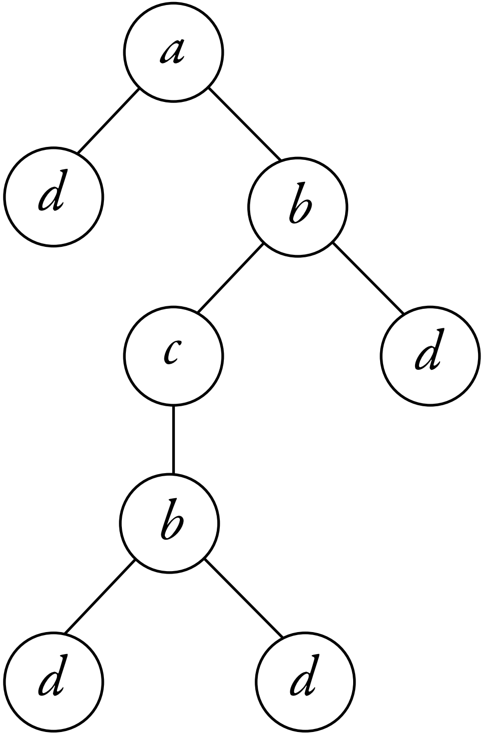
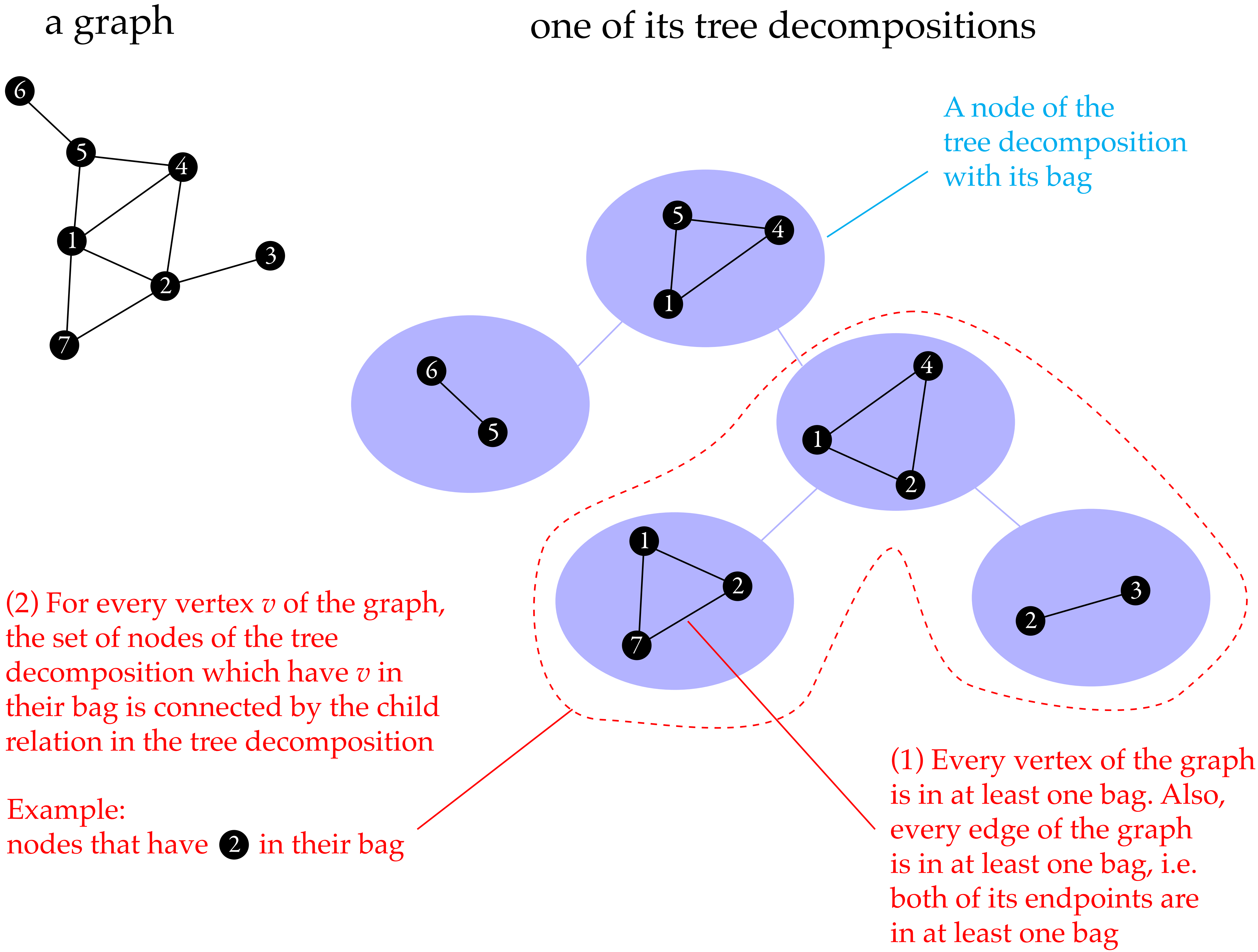

letters of rank 2
rank 1
rank 0
a
b
c
d


a
d
b
c
d
b
d
d

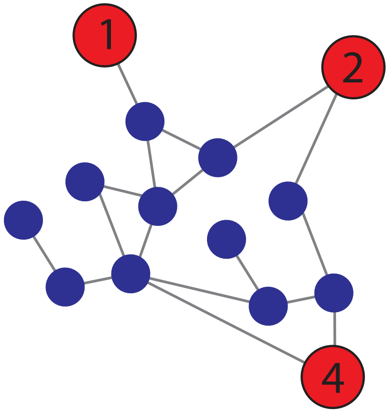
$$f(k) \cdot n^c$$

some computable function

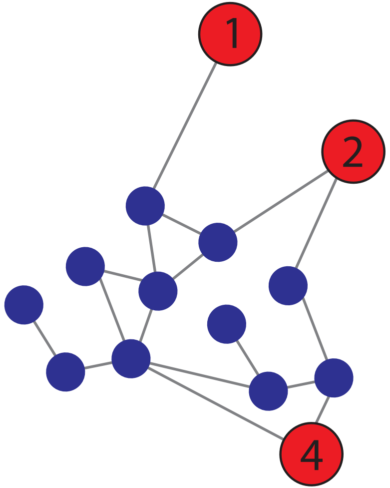
a polynomial with degree independent of k



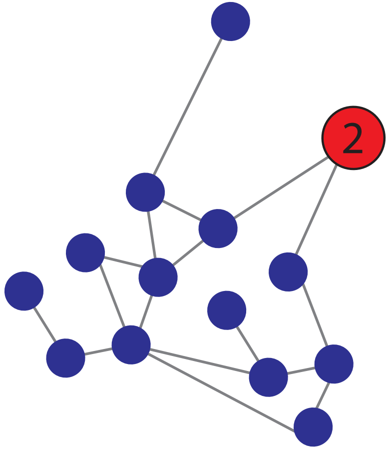
a graph
one of its tree decompositions
6
5
4
1
3
2
7
A node of the tree decomposition with its bag
5
4
1
6
5
4
1
2
1
2
7
3
2
(2) For every vertex v of the graph, the set of nodes of the tree decomposition which have v in their bag is connected by the child relation in the tree decomposition
Example:
nodes that have 2 in their bag
(1) Every vertex of the graph is in at least one bag. Also, every edge of the graph is in at least one bag, i.e. both of its endpoints are in at least one bag

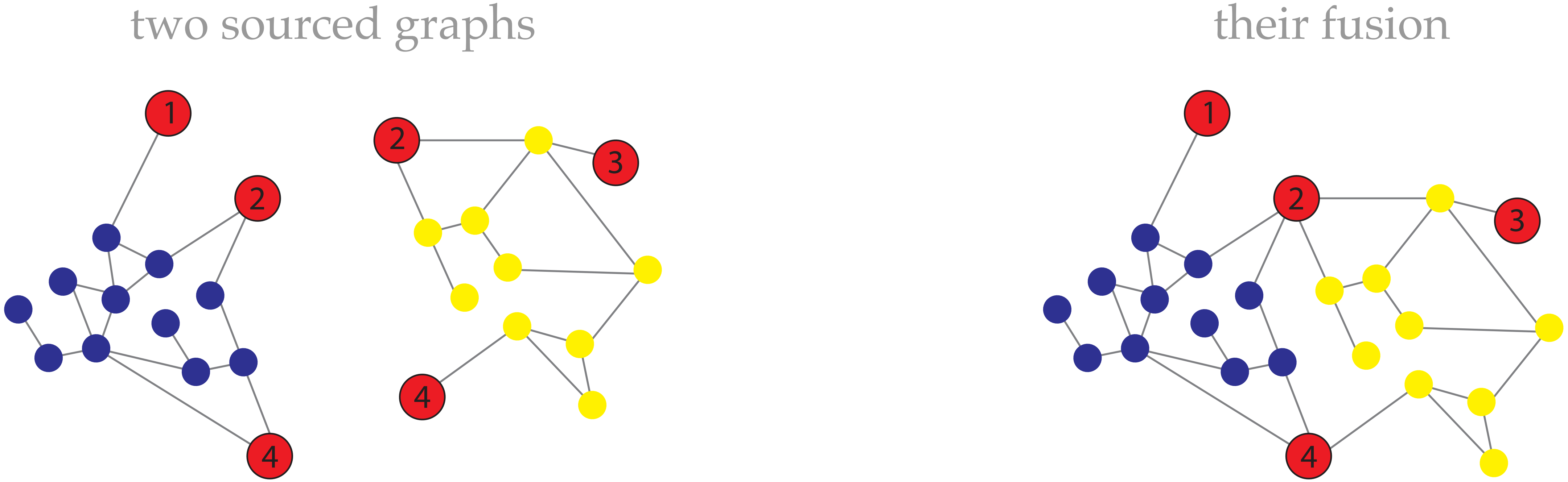

1
2
4

a sourced graph

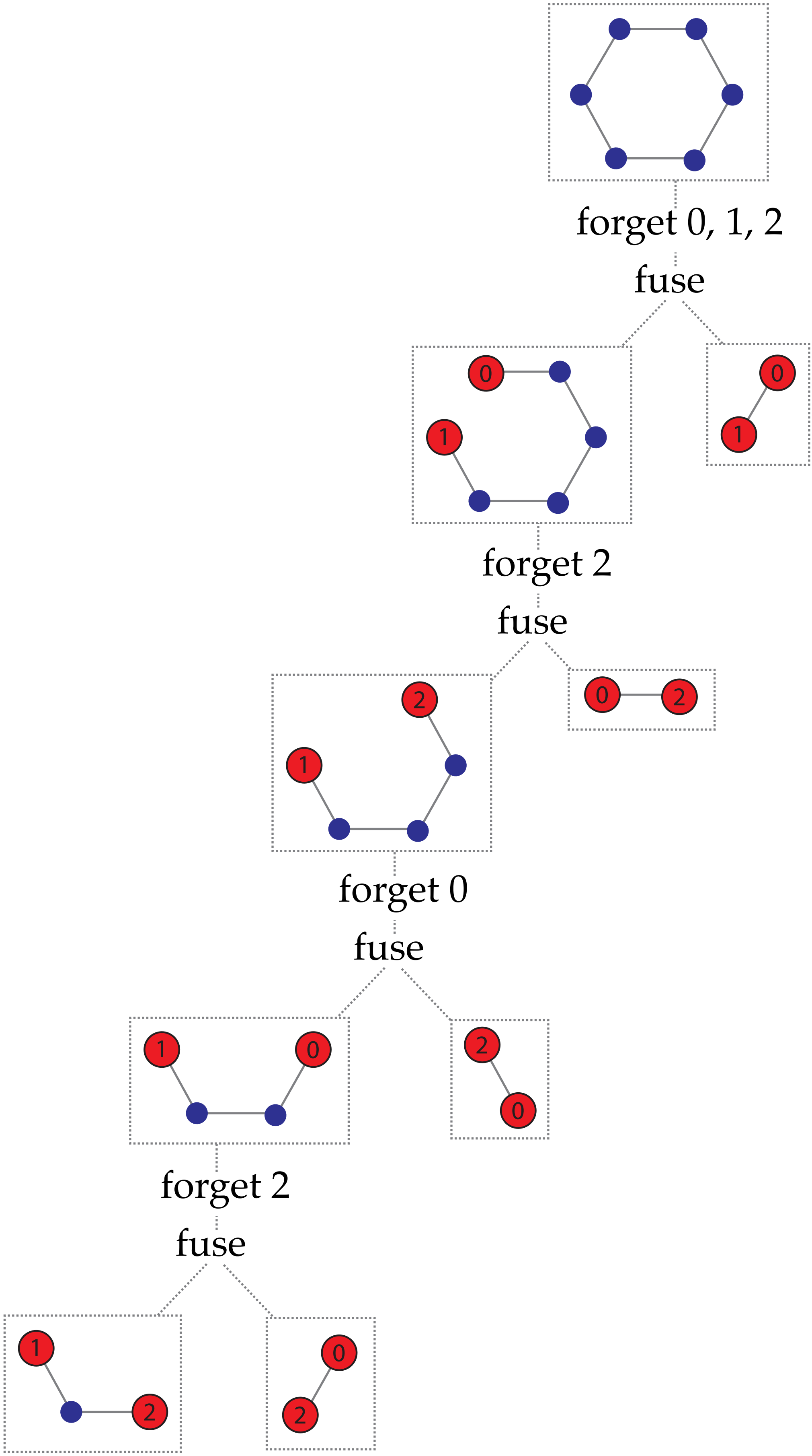


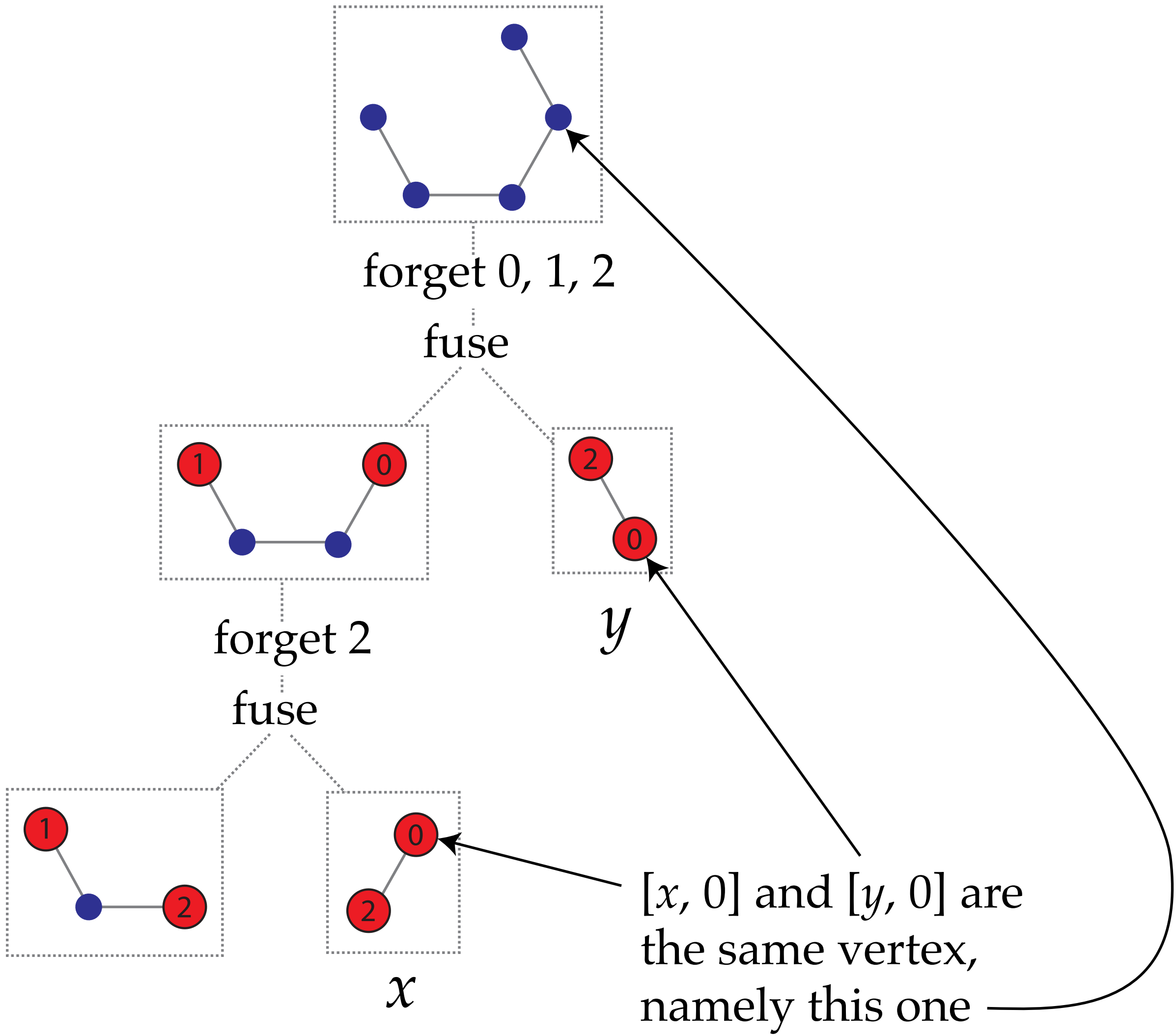
after forgetting 1, 4.

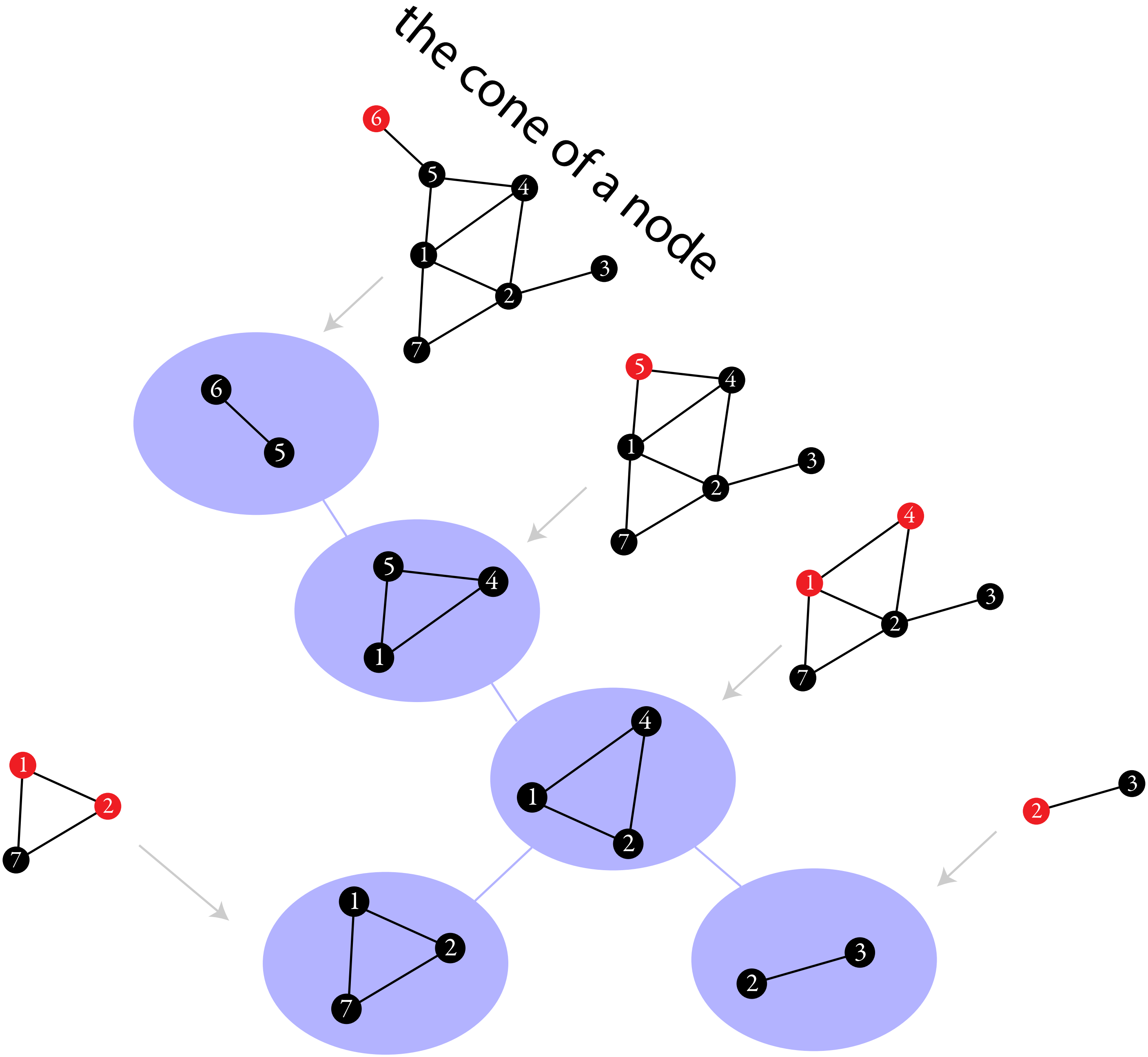


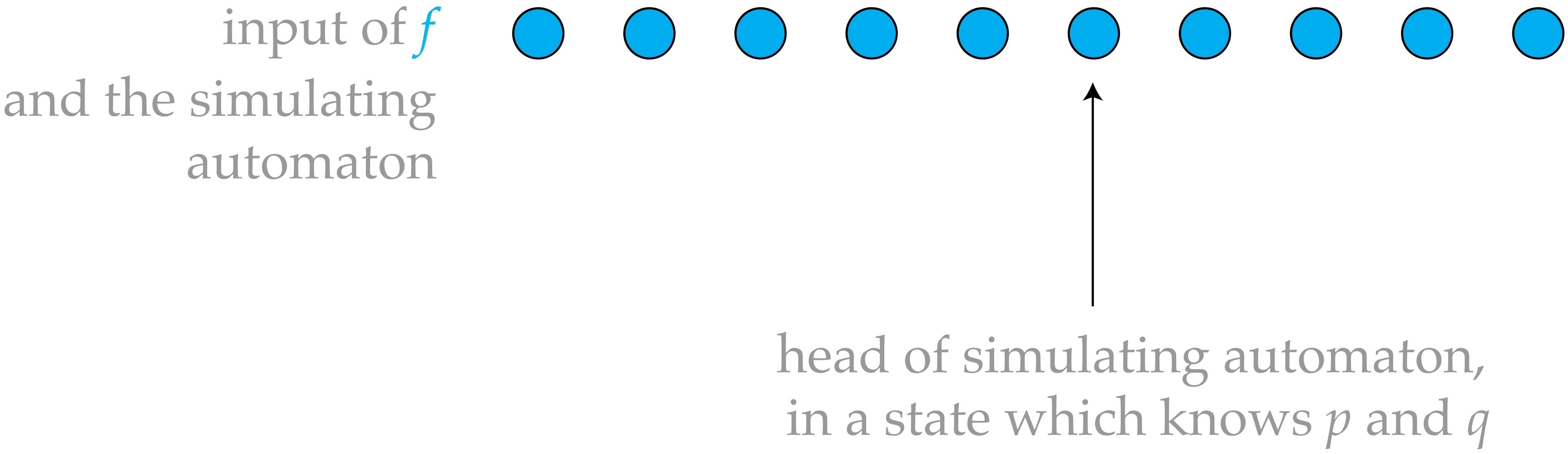


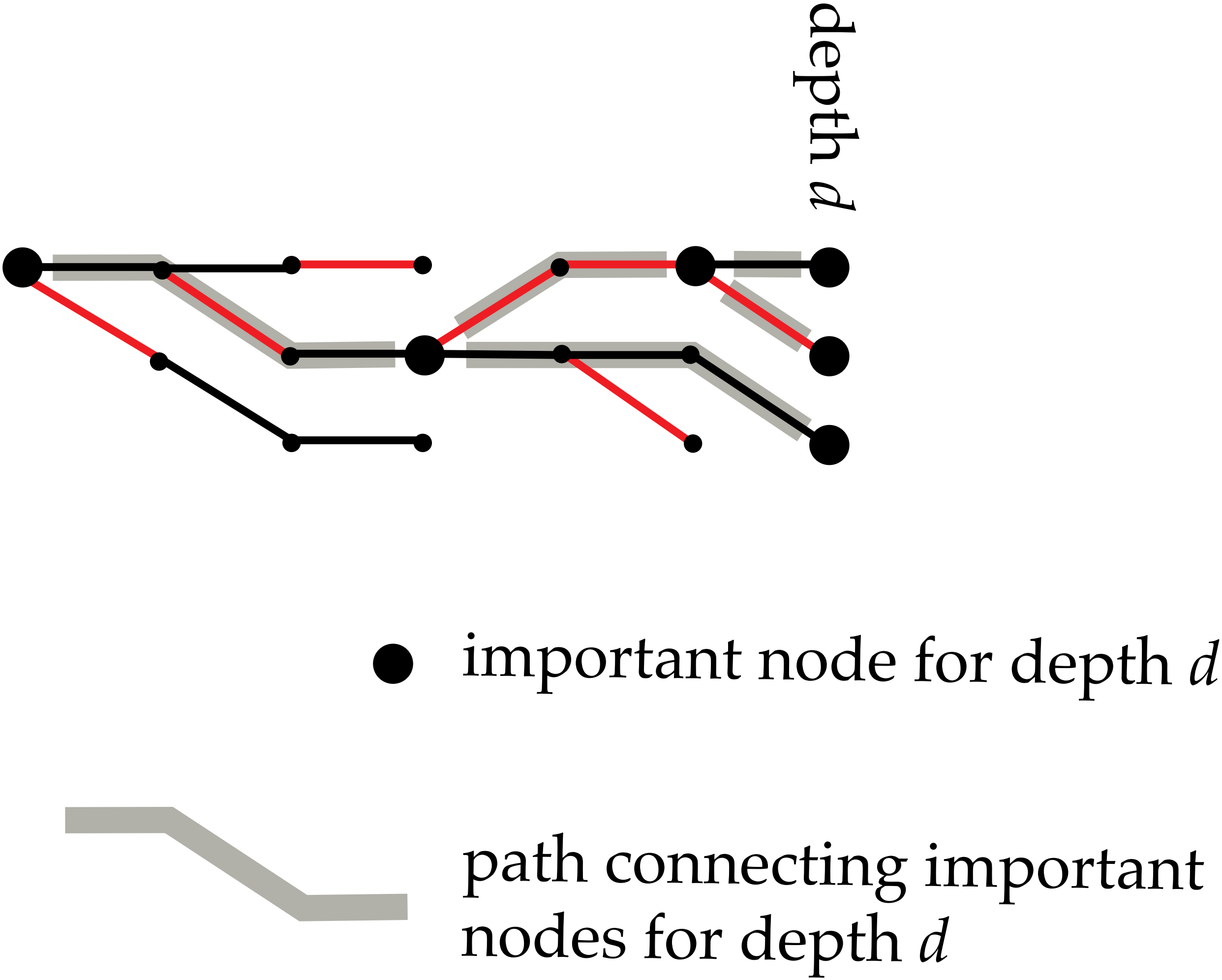
two sourced graphs
their fusion
1
2
4
2
3
4
1
2
3
4

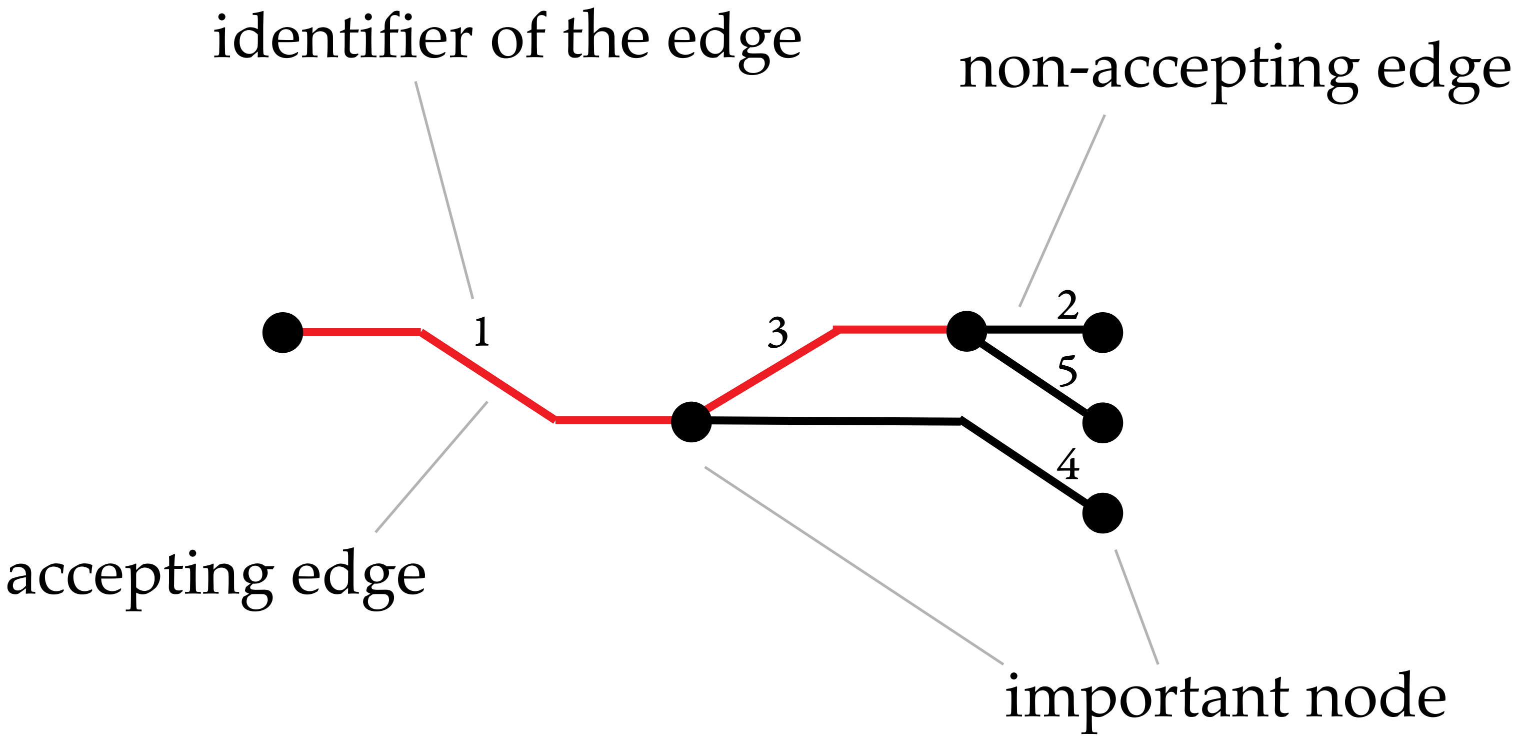

forget 0, 1, 2
fuse
0
1
0
1
forget 2
fuse
2
1
0
2
forget 0
fuse
1
0
2
0
forget 2
fuse
1
2
0
2

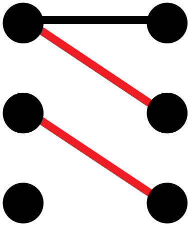

forget 0, 1, 2
fuse
1
0
2
0
forget 2
y
fuse
1
2
0
2
x
[x, 0] and [y, 0] are
the same vertex,
namely this one

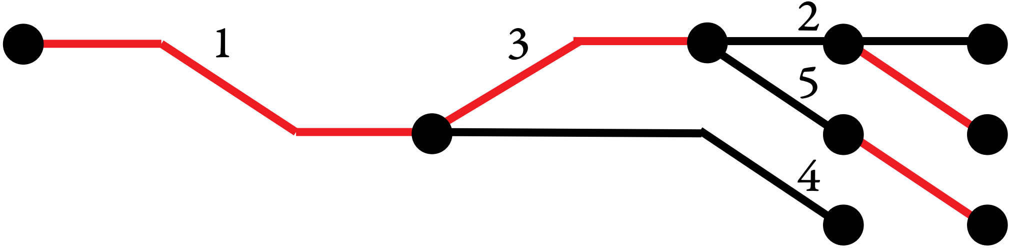

the cone of a node

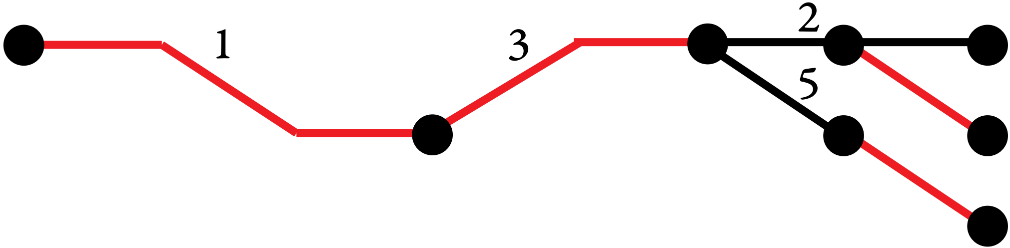

input of f
and the simulating
automaton
head of simulating automaton,
in a state which knows p and q


depth d
important node for depth d
path connecting important nodes for depth d

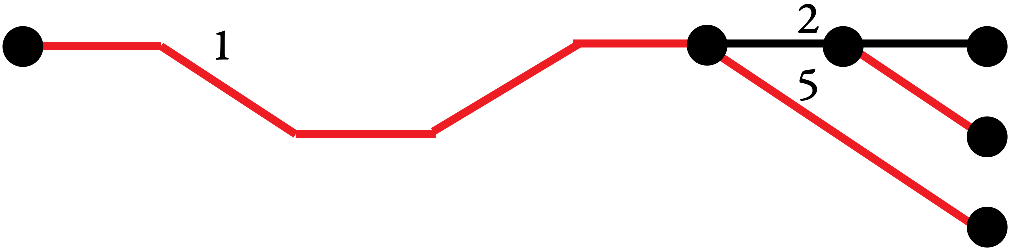

identifier of the edge
non-accepting edge
1
3
2
5
4
accepting edge
important node



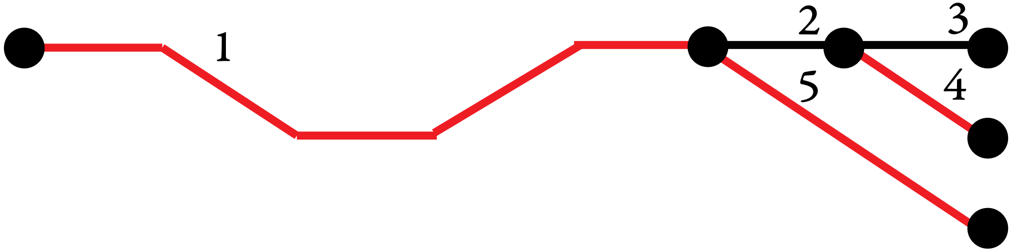

1
3
2
5
4

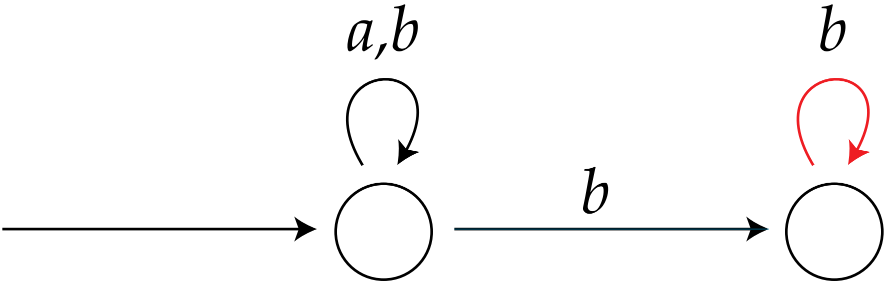


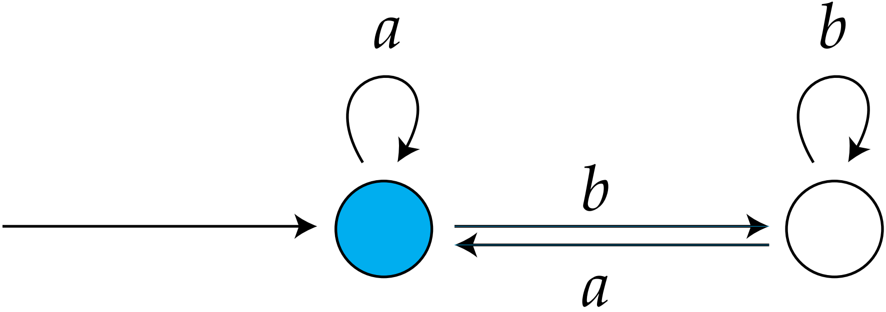
1
3
2
5



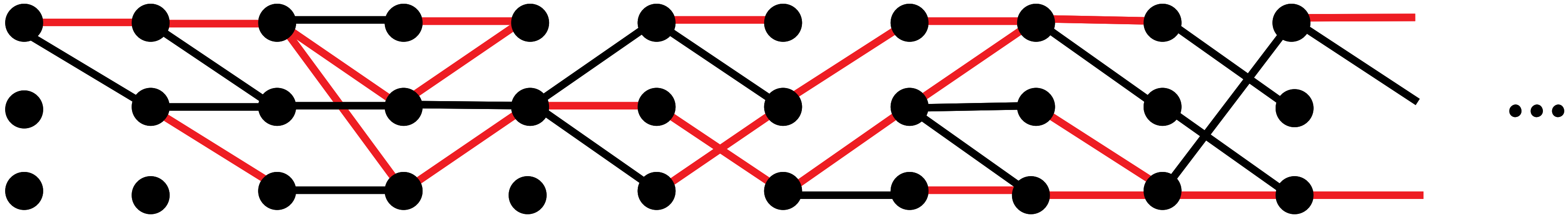
1
2
5


1
2
3
5
4

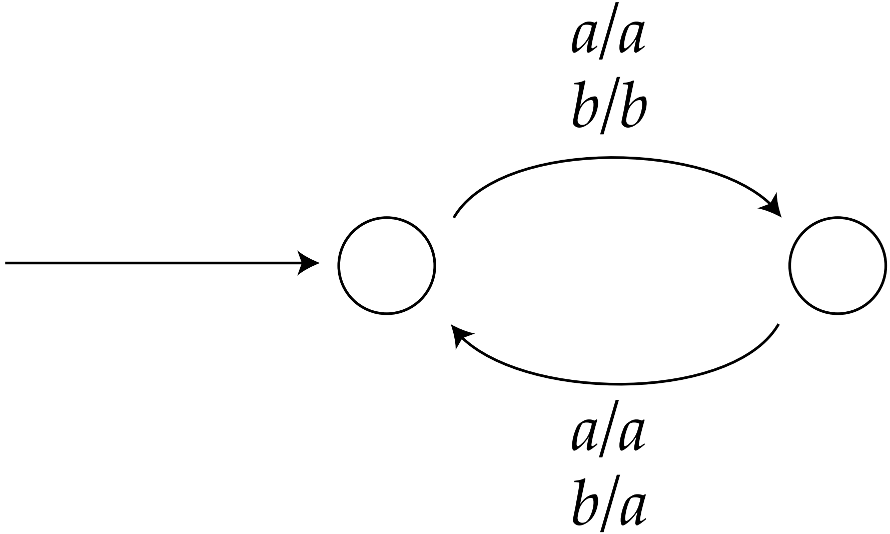

a,b
b
b

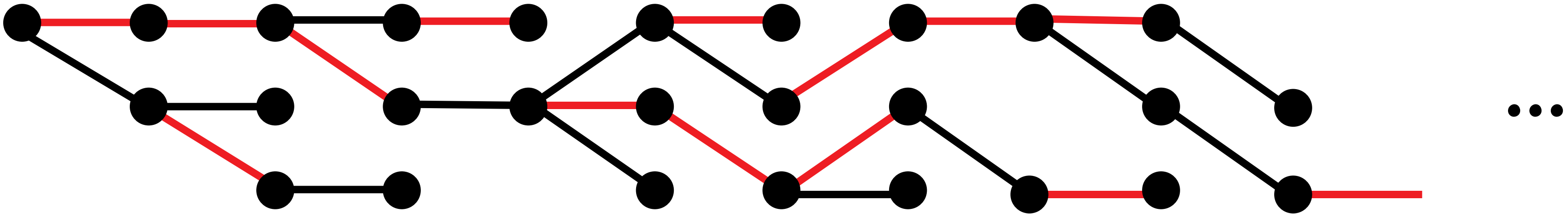


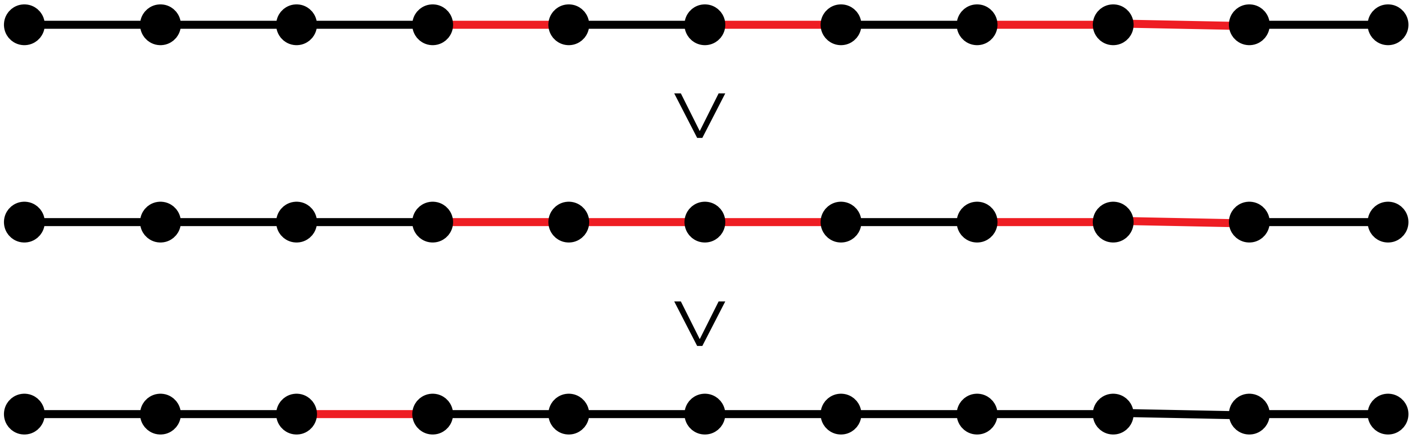
a
b
b
a



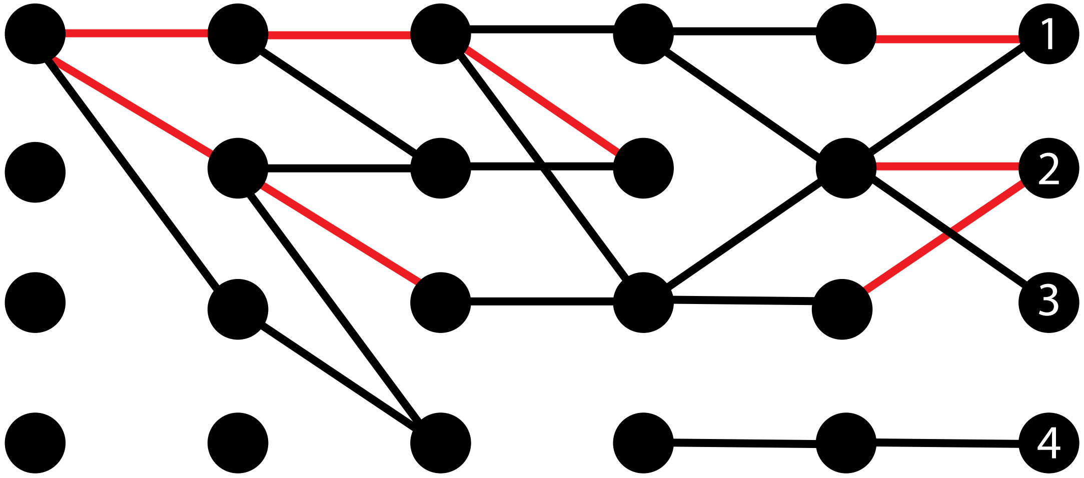
a transition a/b means that
letter a is input, and letter b is output



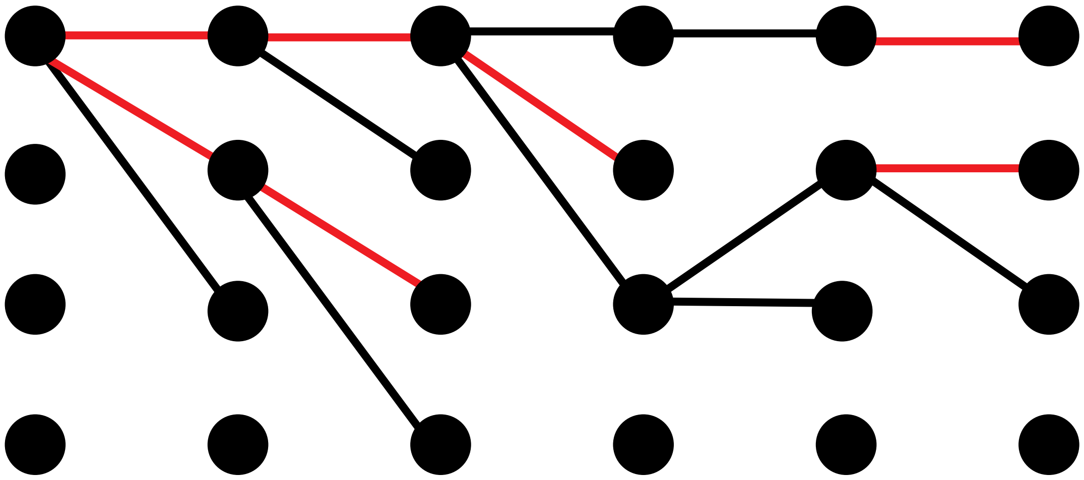




input



output



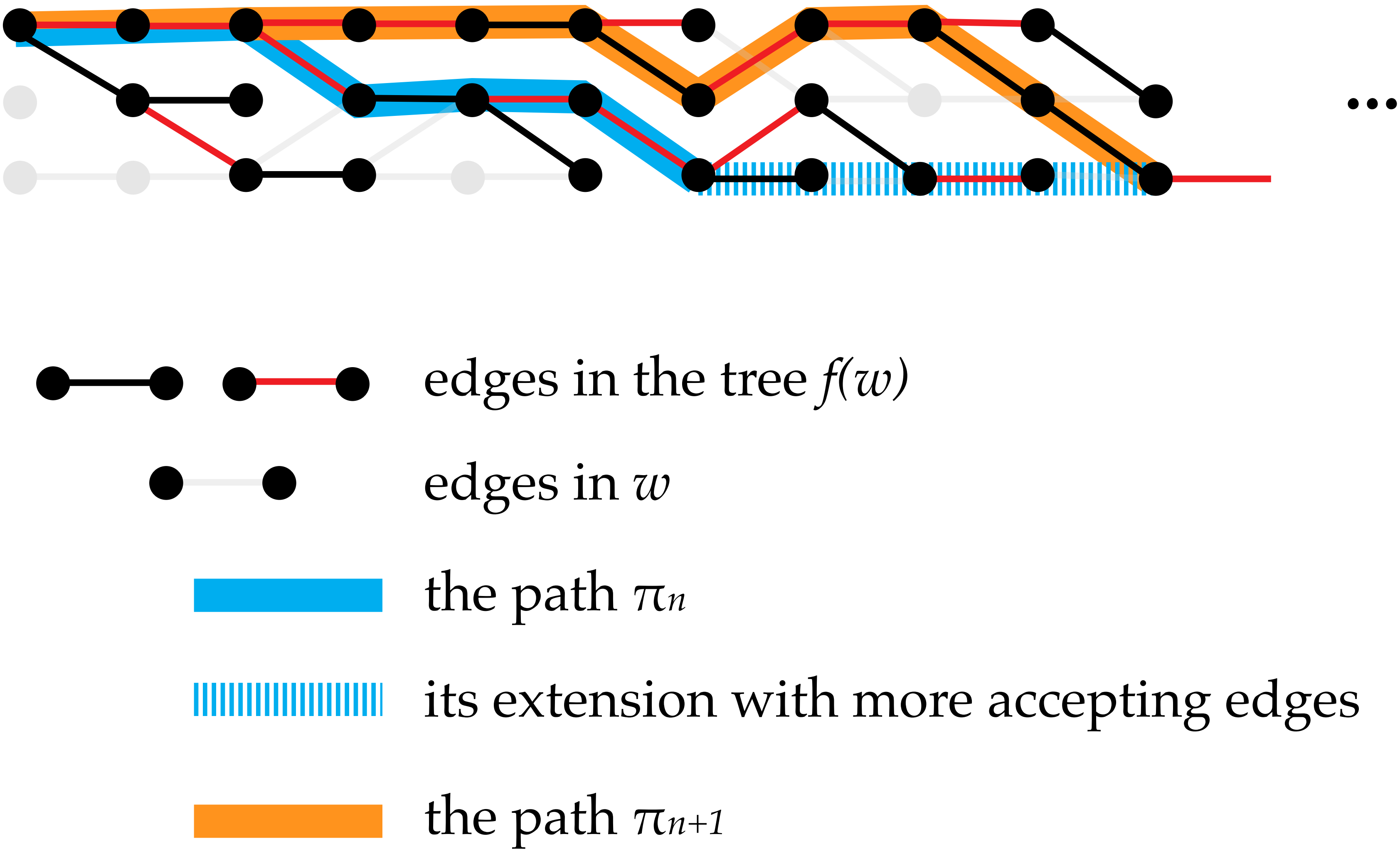
The state of the tranducer is this information:

The reachable vertices are

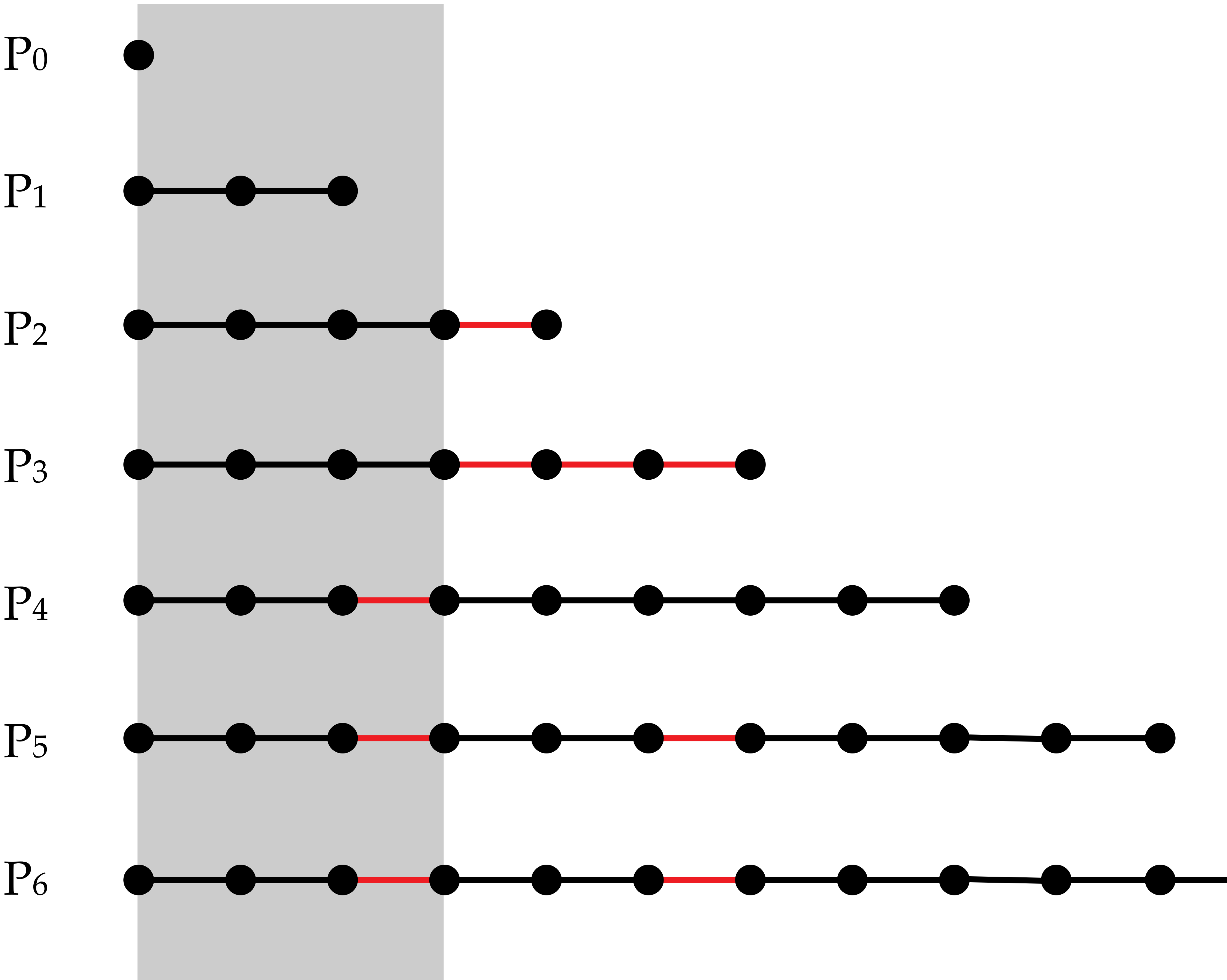
❶ ❷ ❸

and the least profiles for reaching them are ordered as

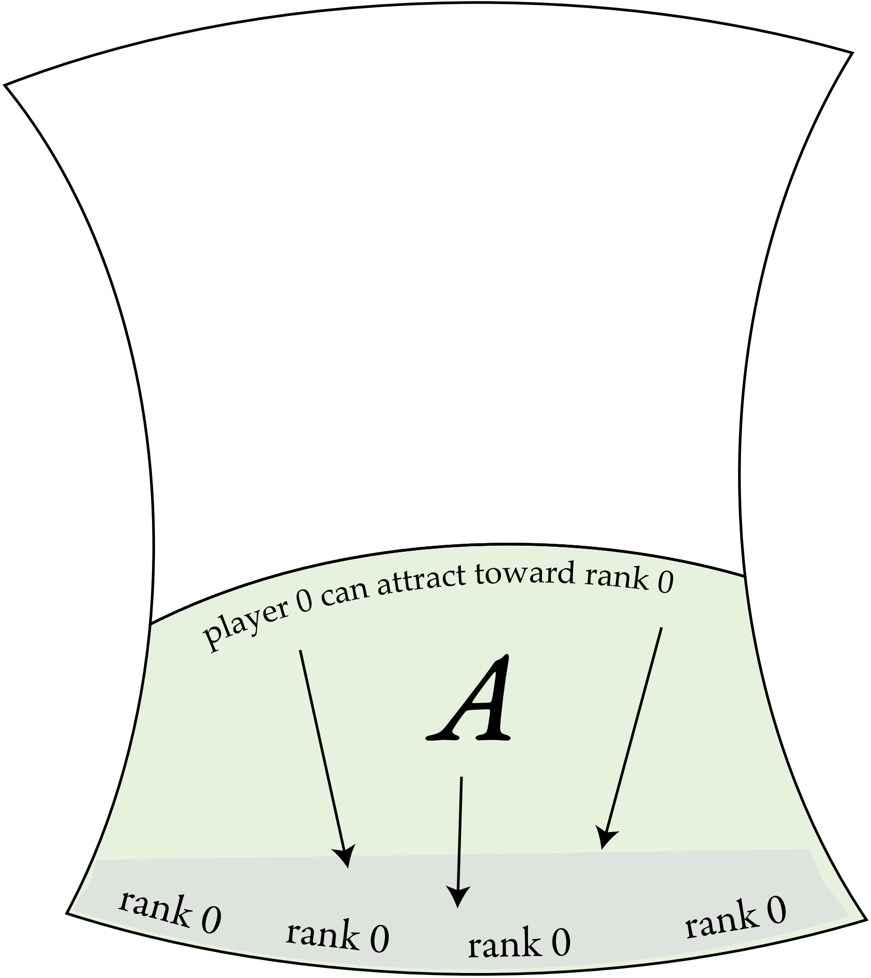
❶ = ❷ < ❸



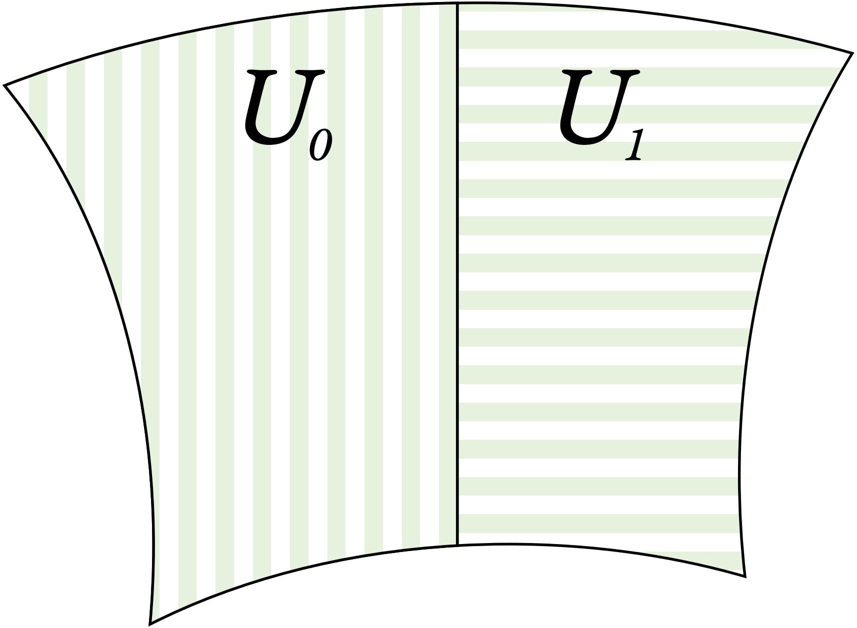
...
edges in the tree $f(w)$
edges in w
the path π_n
its extension with more accepting edges
the path π_{n+1}

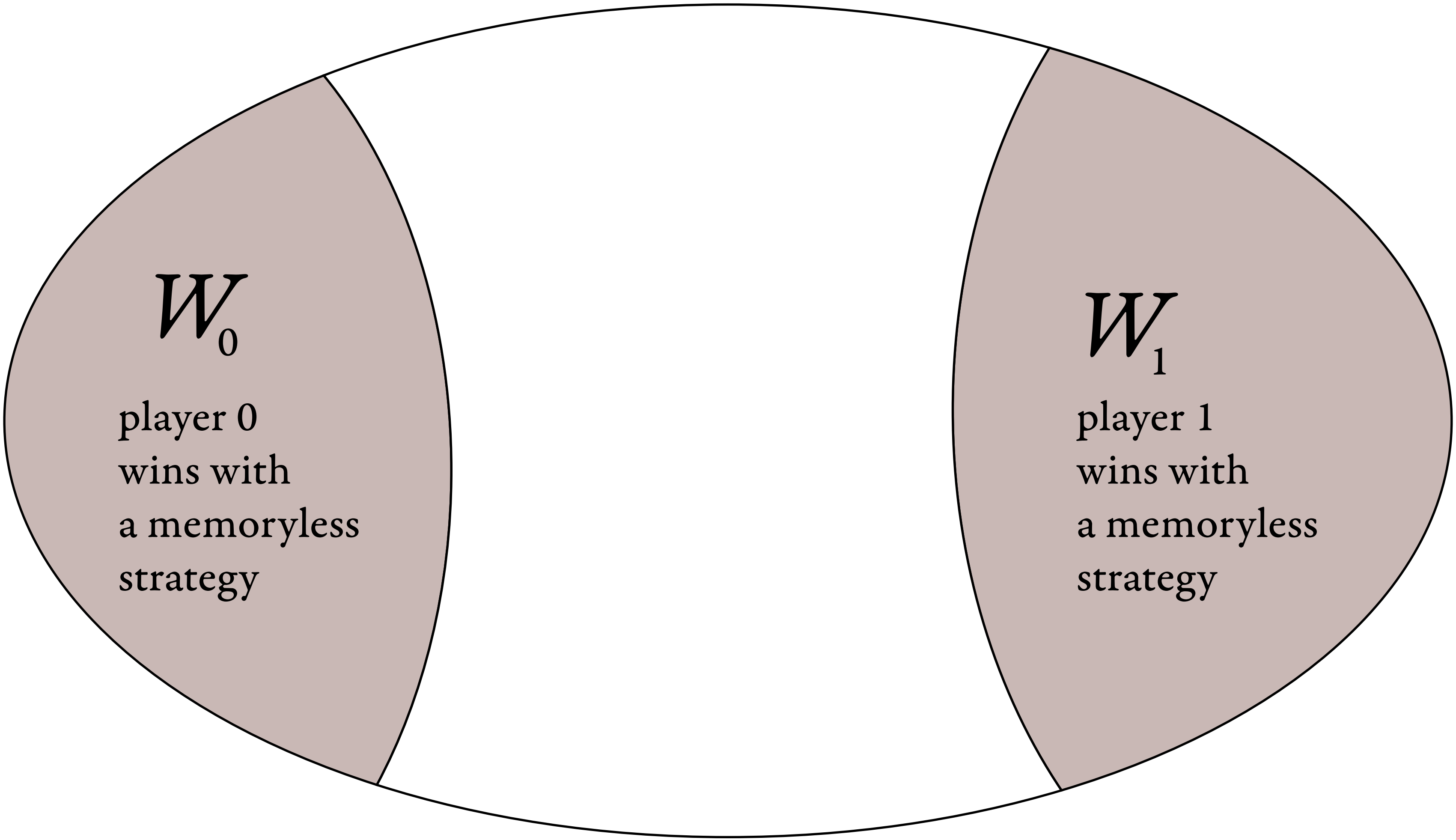


P0
P1
P2
P3
P4
P5
P6

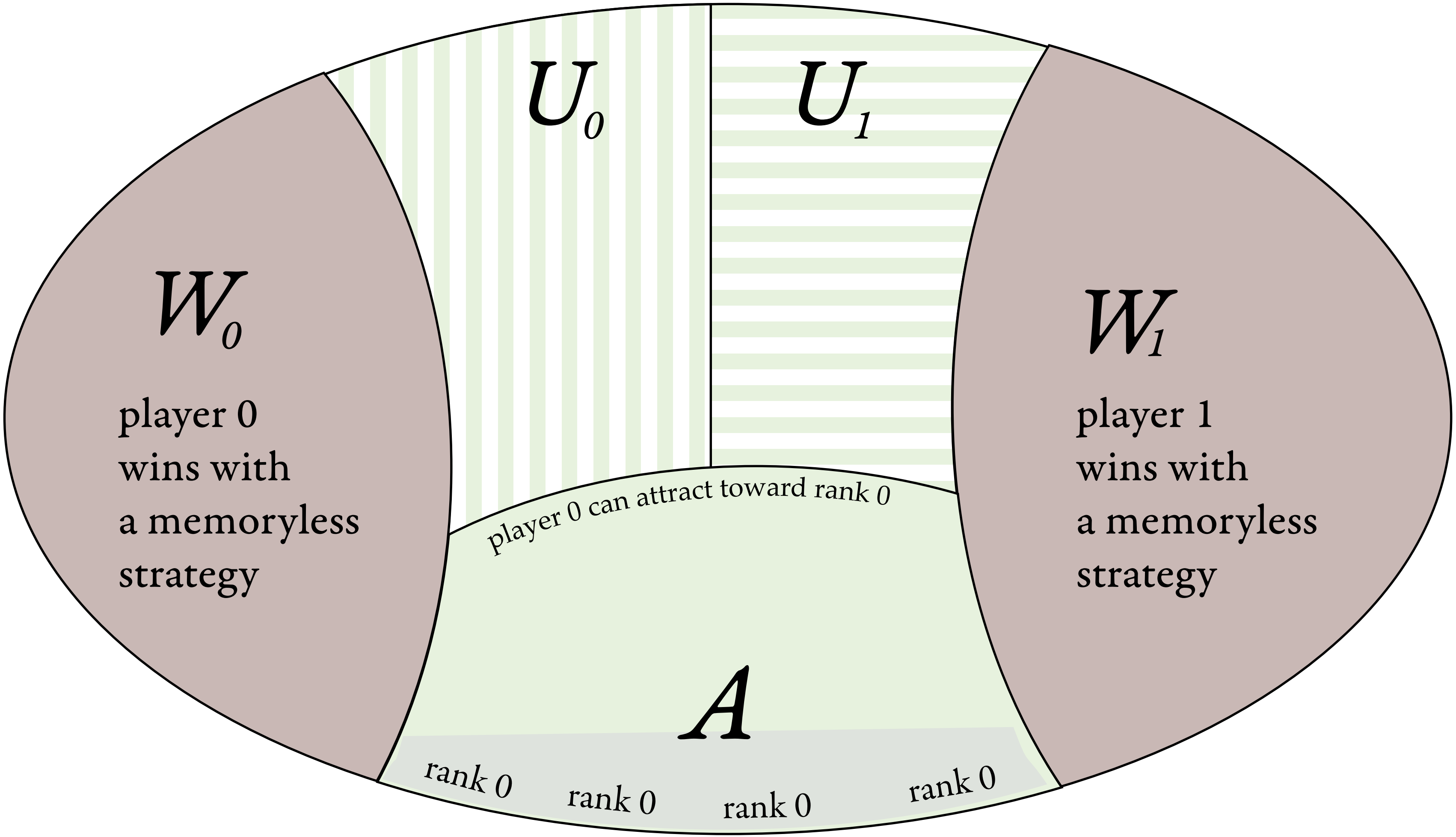


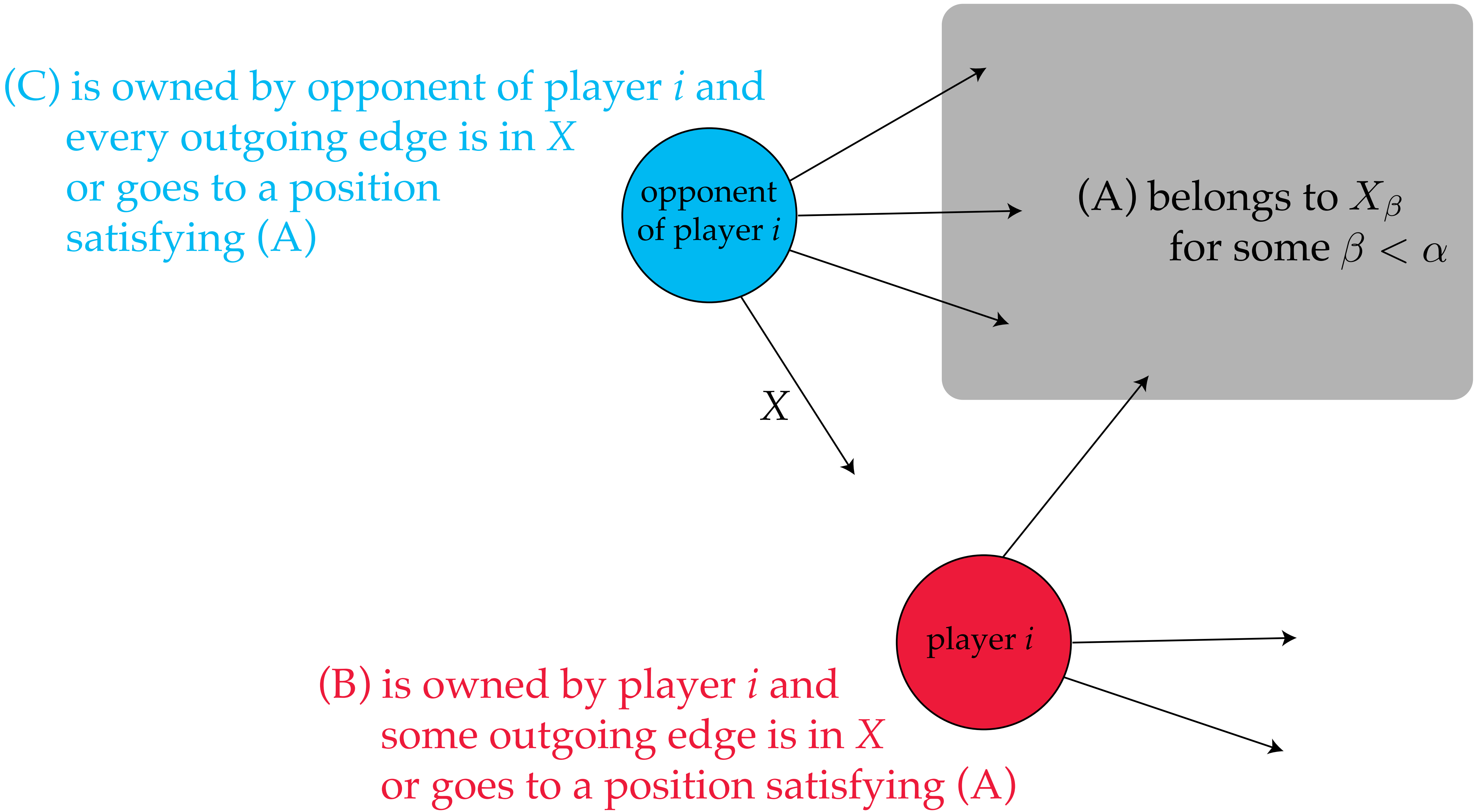
player 0 can attract toward rank 0
A
rank 0
rank 0
rank 0
rank 0

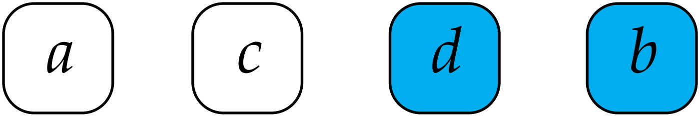

U_0
U_1

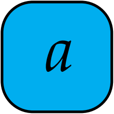


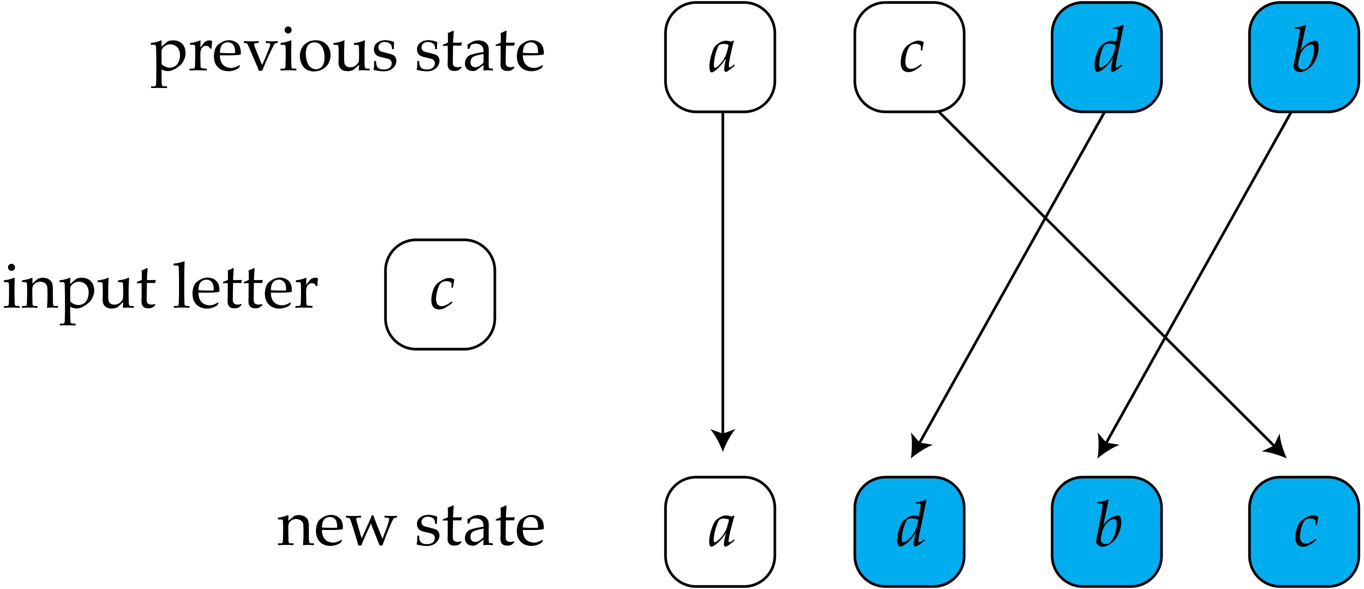
W_0
player 0
wins with
a memoryless
strategy
W_1
player 1
wins with
a memoryless
strategy

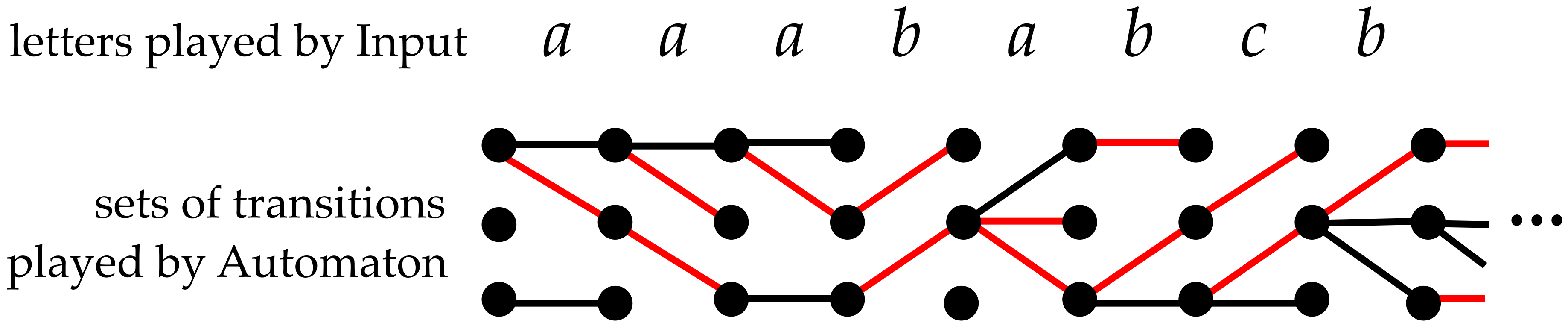

U_0
U_1
W_0
player 0
wins with
a memoryless
strategy
W_1
player 1
wins with
a memoryless
strategy
player 0 can attract toward rank 0
A
rank 0
rank 0
rank 0
rank 0

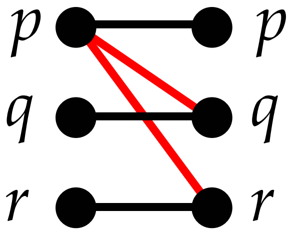

(C) is owned by opponent of player i and
every outgoing edge is in X
or goes to a position
satisfying (A)
opponent
of player i
(A) belongs to X_β
for some $\beta < \alpha$
X
player i
(B) is owned by player i and
some outgoing edge is in X
or goes to a position satisfying (A)

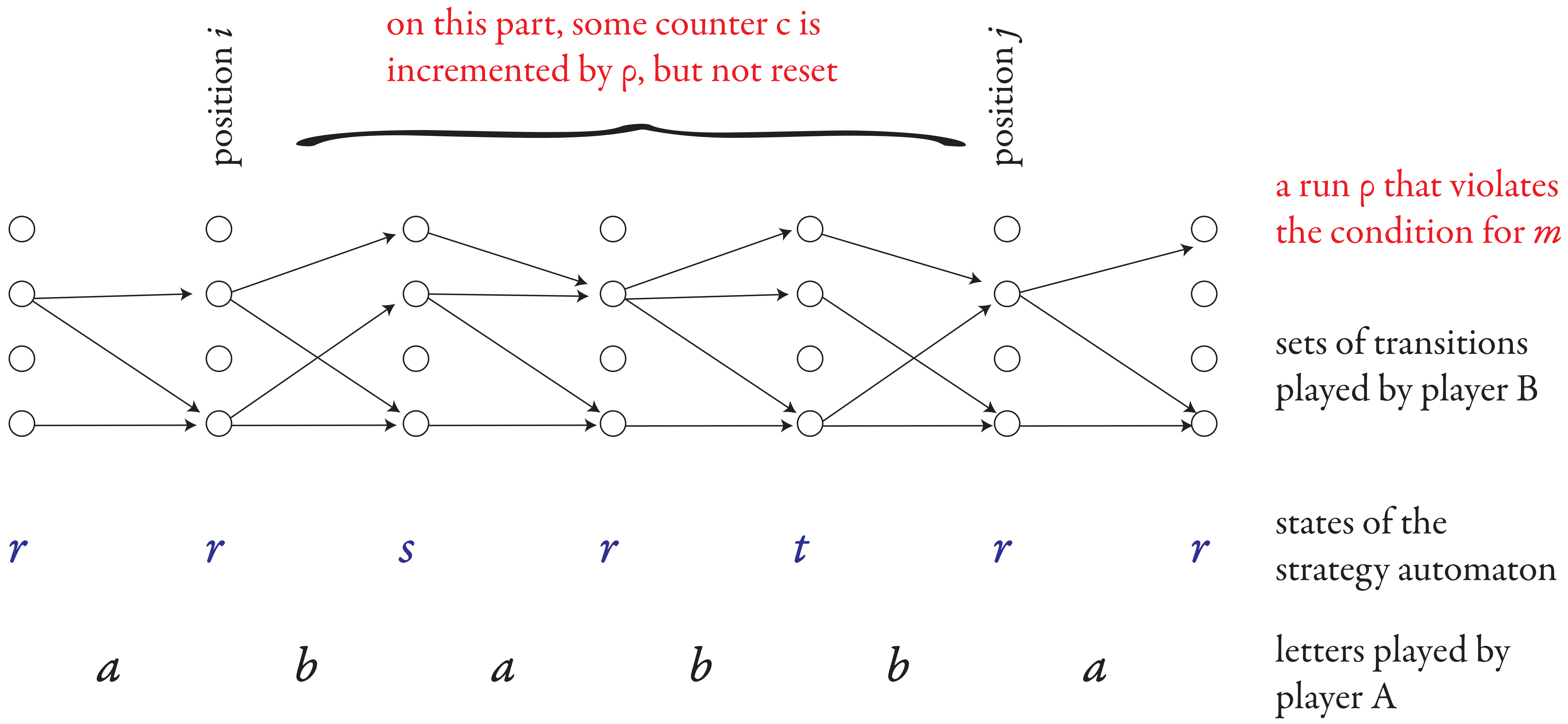

a
c
d
b


a

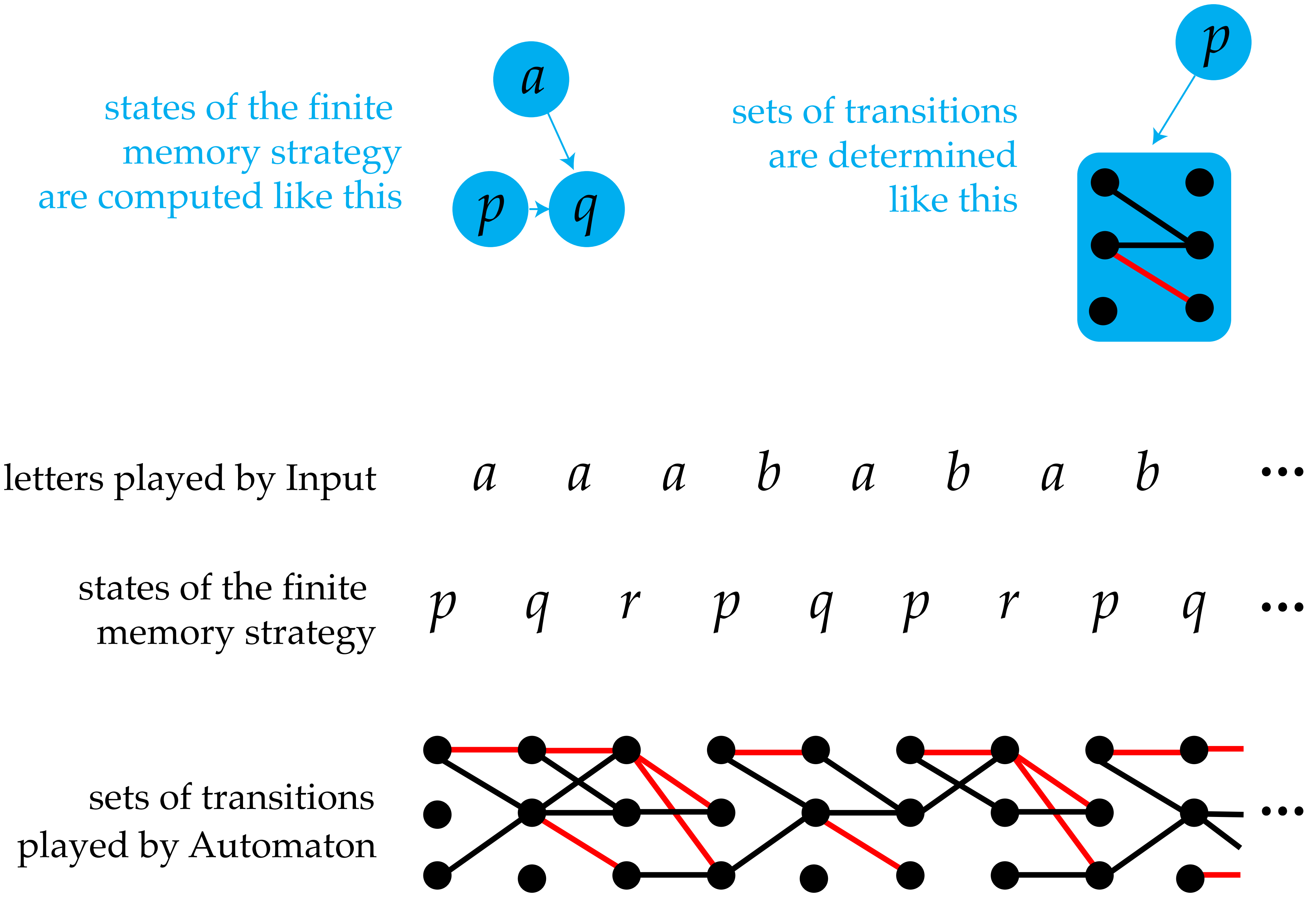

previous state
a
c
d
b
input letter
c
new state
a
d
b
c


letters played by Input
a a a b a b c b
sets of transitions played by Automaton
...

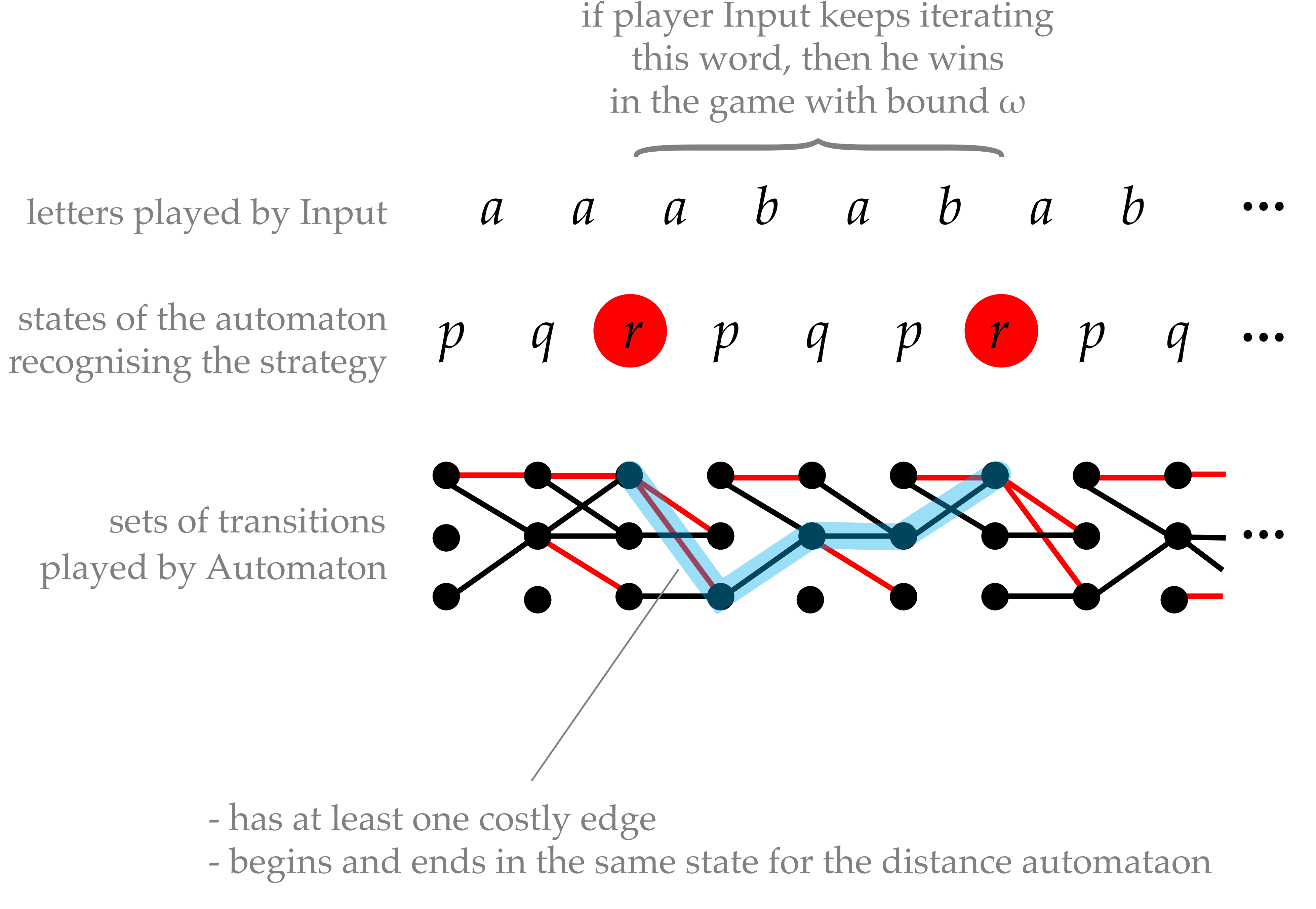

p
q
r
p
q
r



on this part, some counter c is incremented by ρ, but not reset
position i
position j
a run ρ that violates the condition for m
sets of transitions played by player B
r
r
s
r
t
r
r
states of the strategy automaton
a
b
a
b
b
a
letters played by player A

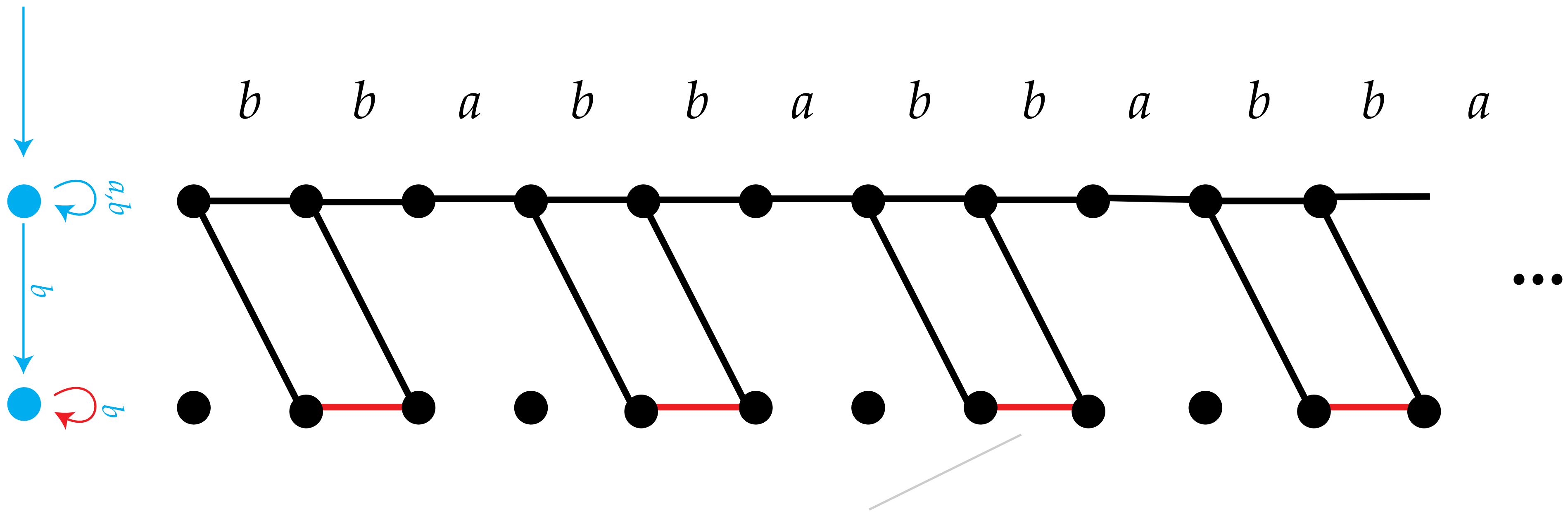


states of the finite memory strategy are computed like this
a
p
q
sets of transitions are determined like this
p
letters played by Input
a a a b a b a b ...
states of the finite memory strategy
p q r p q p r p q ...
sets of transitions played by Automaton
...

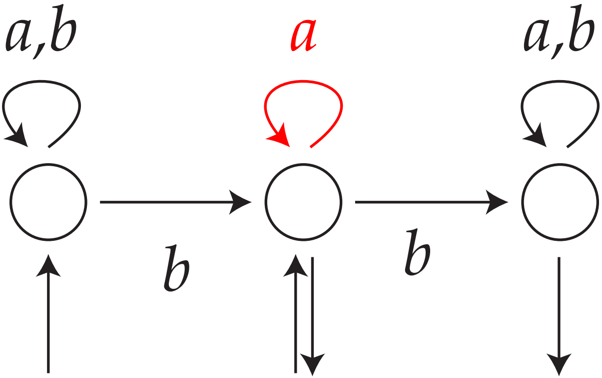

if player Input keeps iterating
this word, then he wins
in the game with bound ω
letters played by Input
a a a b a b a b ...
states of the automaton
recognising the strategy
p q r p q p r p q ...
sets of transitions
played by Automaton
...
- has at least one costly edge
- begins and ends in the same state for the distance automataon

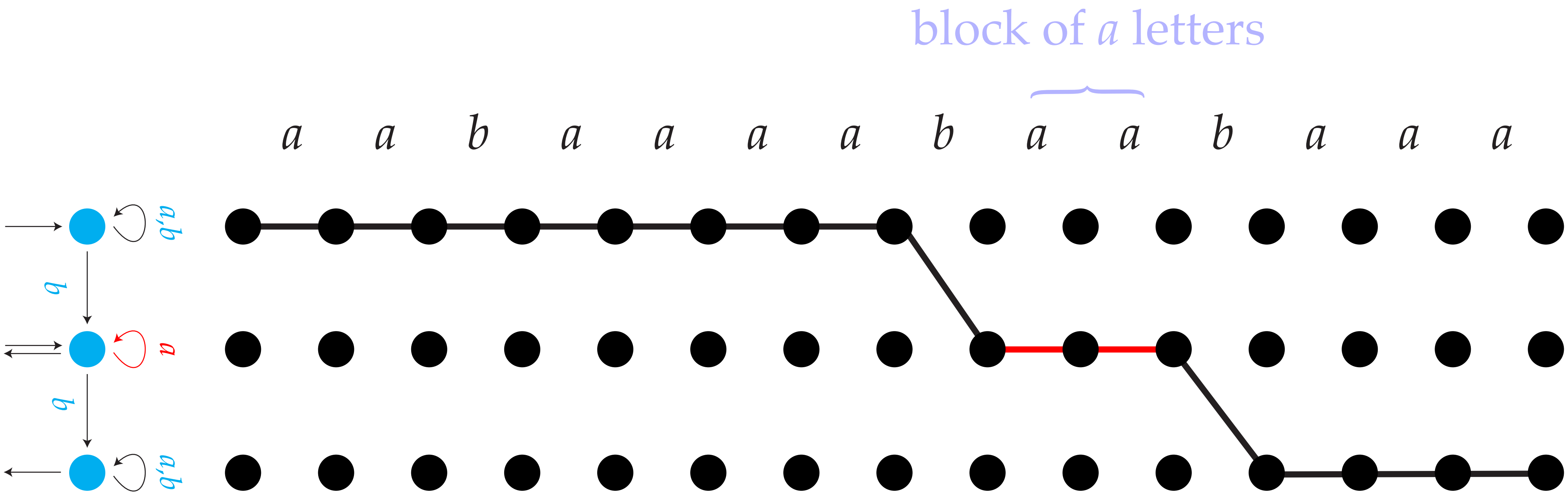
the automaton

the runs of the automaton over $(bba)^\omega$

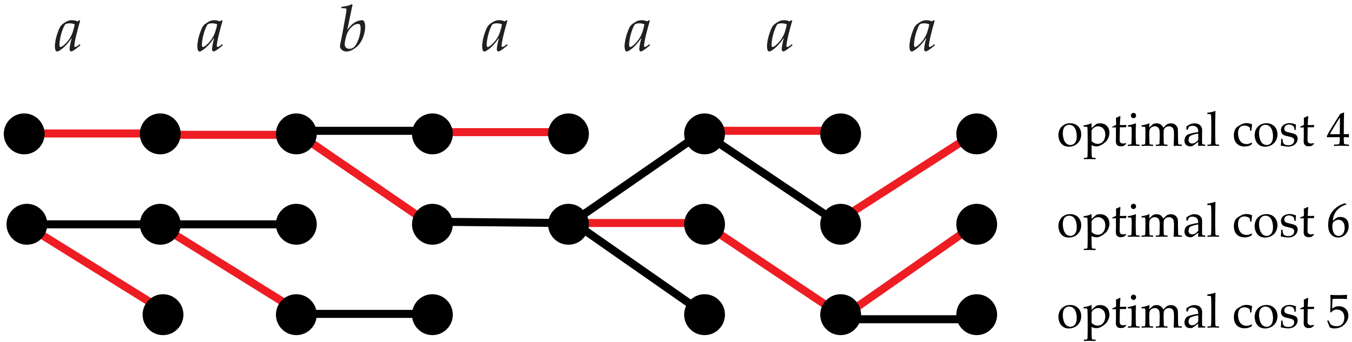


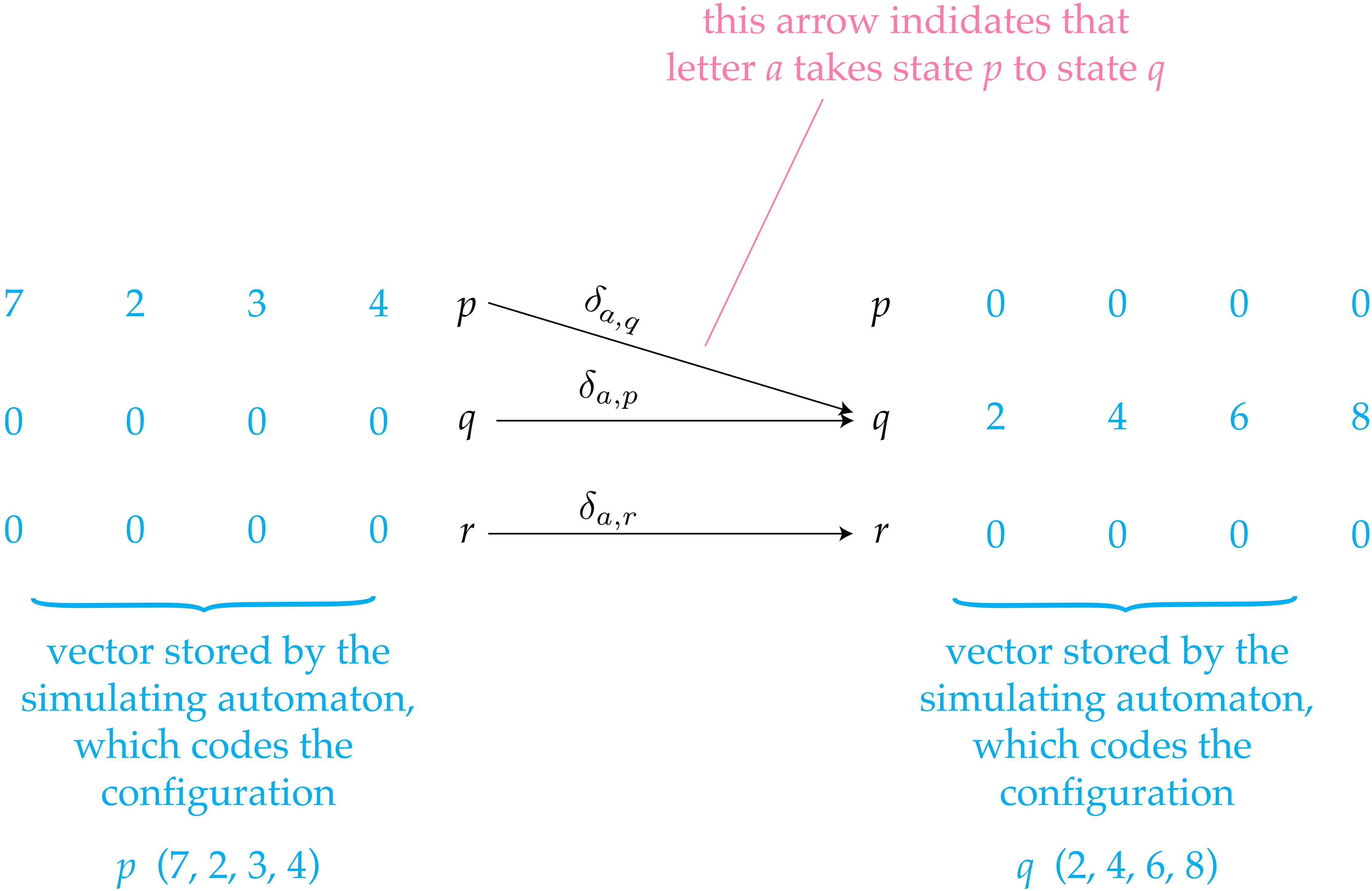
an accepting transition is seen infinitely often


a,b
a
a,b
b
b

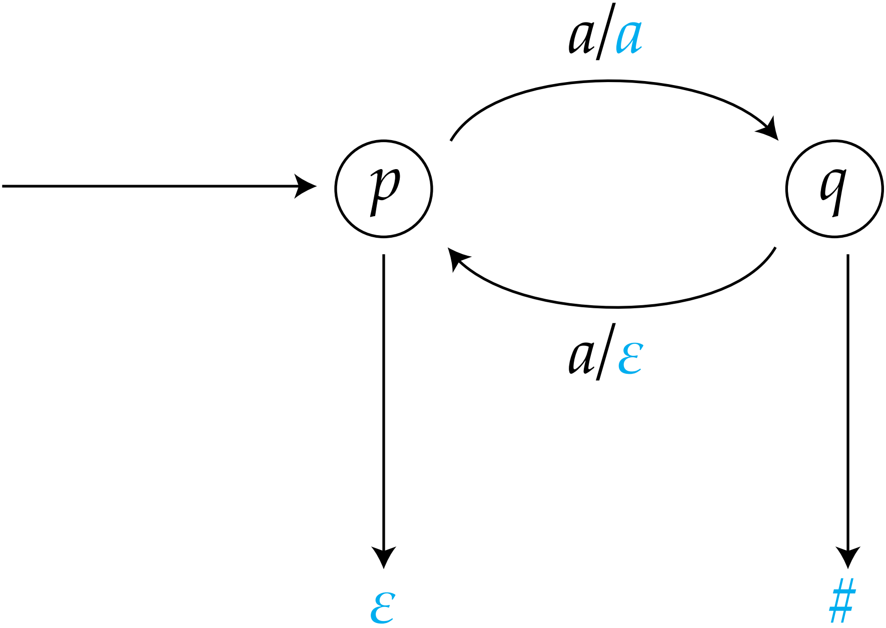


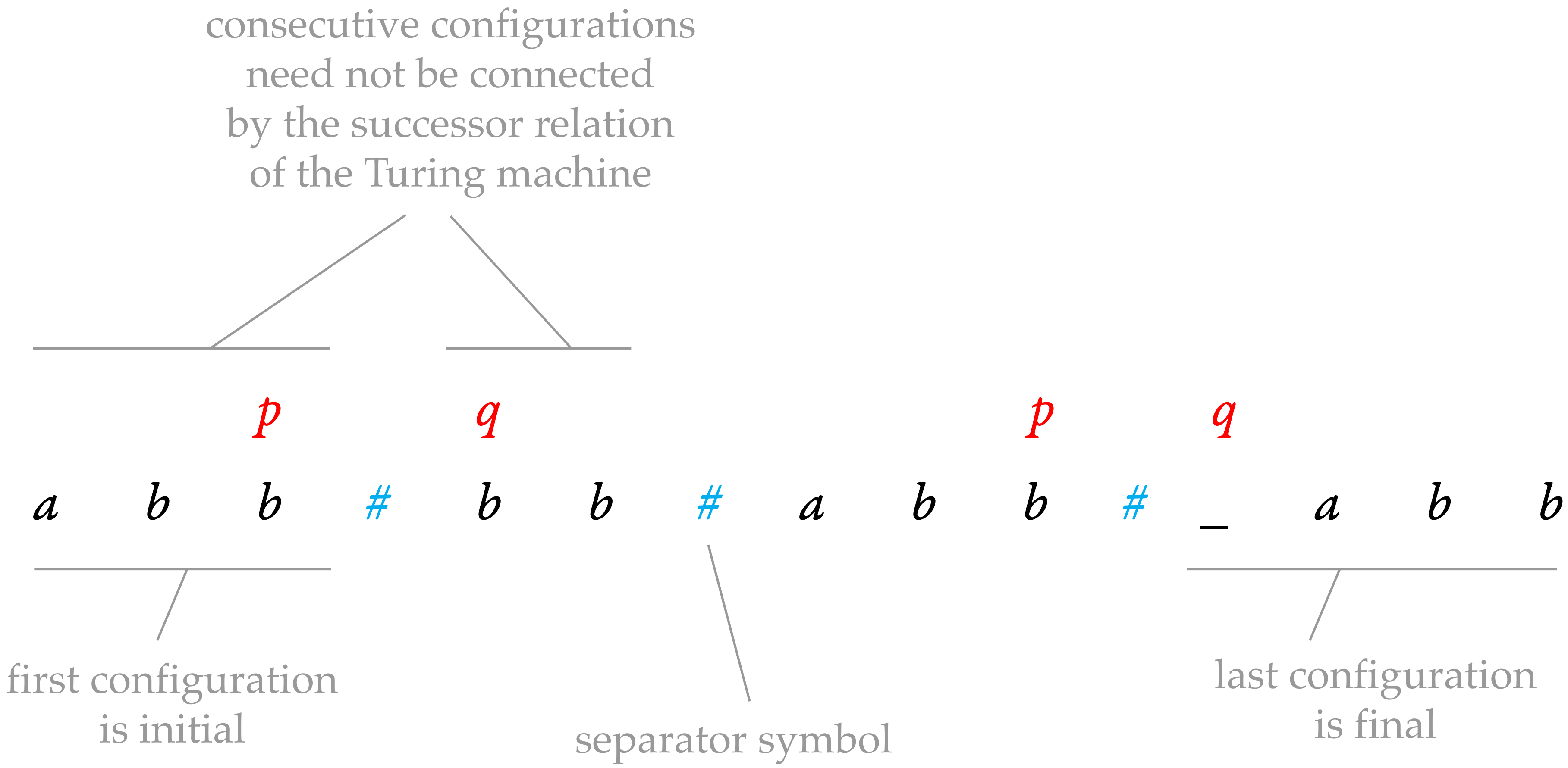
block of a letters
a a b a a a a b a a b a a a
a,b
b
a
b
a,b

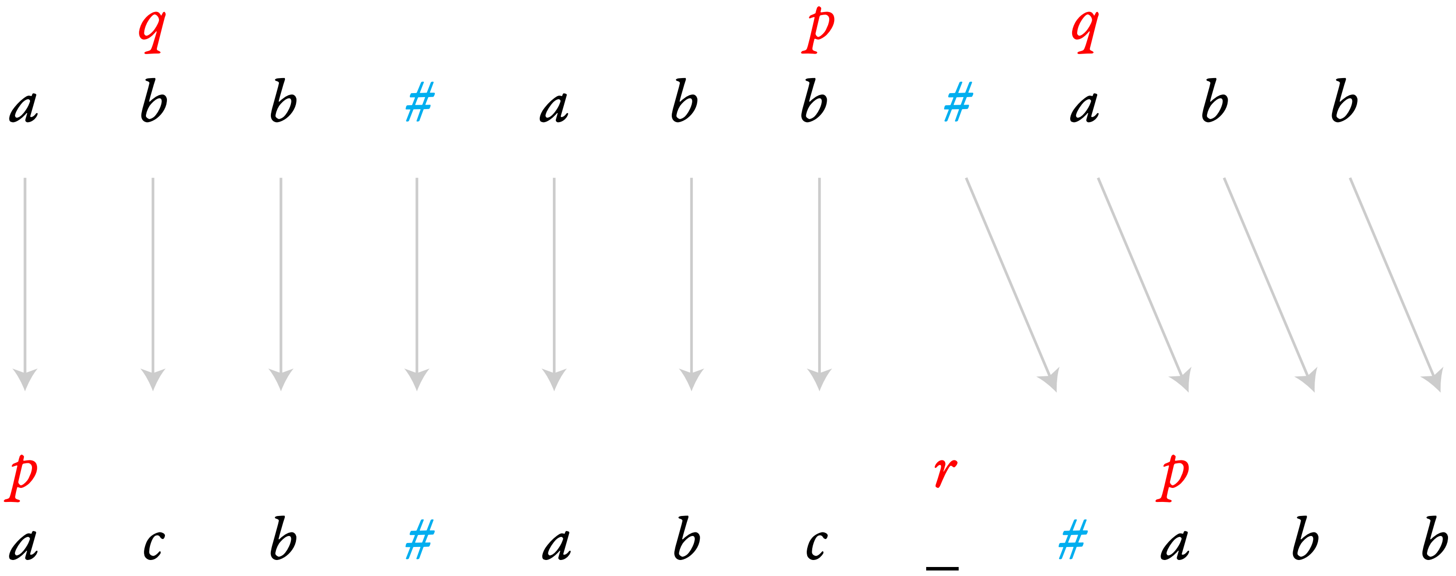

a a b a a a a
optimal cost 4
optimal cost 6
optimal cost 5



this arrow indidates that
letter a takes state p to state q
7 2 3 4 p
0 0 0 0 q
0 0 0 0 r
$\delta_{a,q}$
$\delta_{a,p}$
$\delta_{a,r}$
p 0 0 0 0
q 2 4 6 8
r 0 0 0 0
vector stored by the simulating automaton, which codes the configuration
p (7, 2, 3, 4)
vector stored by the simulating automaton, which codes the configuration
q (2, 4, 6, 8)


a/a
p
q
a/ε
ε
#

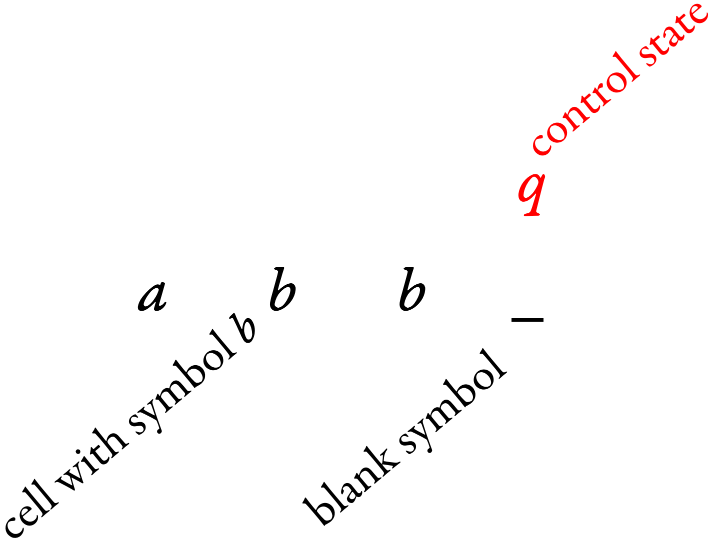

consecutive configurations need not be connected by the successor relation of the Turing machine
p
q
p
q
a b b # b b # a b b # _ a b b
first configuration is initial
separator symbol
last configuration is final

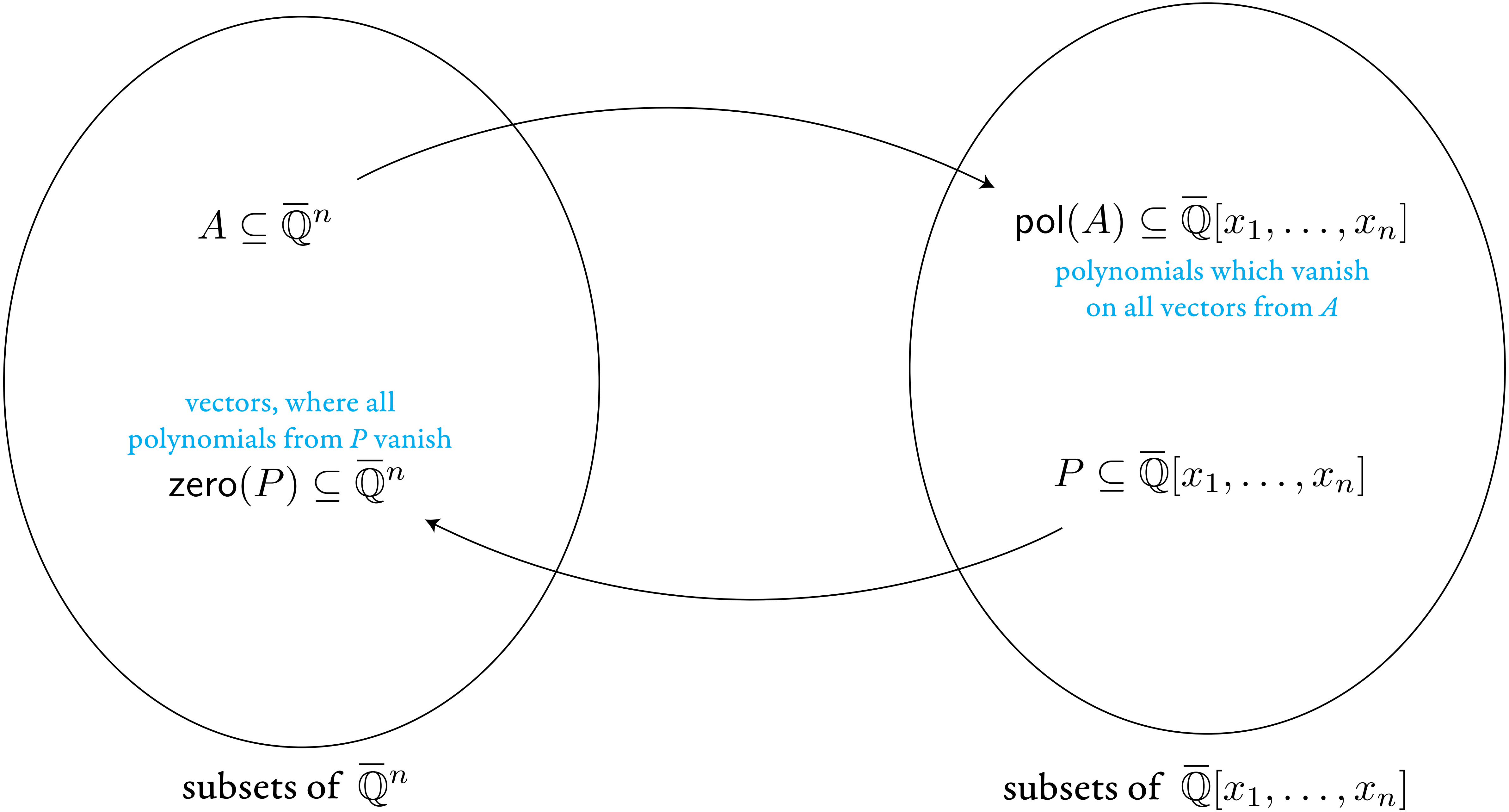


the head wrote *c* and moved one step to the left in state *p*

the head wrote *c* and moved one step to the right in state *r*

	q					p		q			
a	b	b	#	a	b	b	#	_	a	b	b

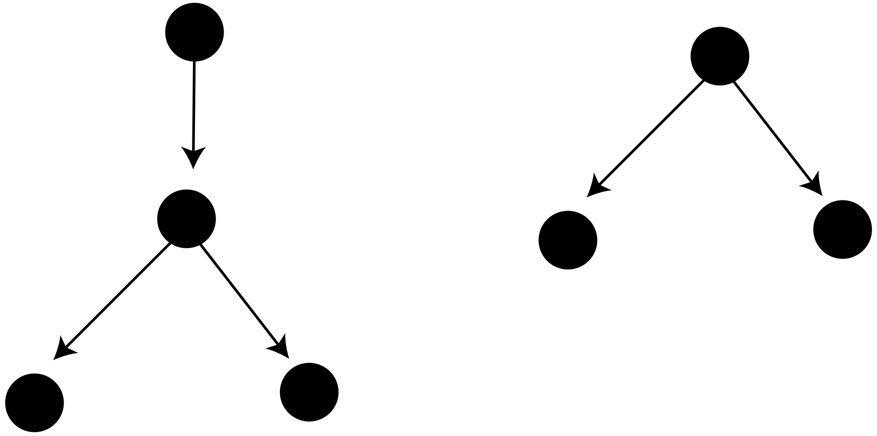

control state
q
a
b
b
_
cell with symbol b
blank symbol


$A \subseteq \overline{\mathbb{Q}}^n$
$\mathsf{pol}(A) \subseteq \overline{\mathbb{Q}}[x_1, \ldots, x_n]$
polynomials which vanish
on all vectors from A
vectors, where all
polynomials from P vanish
$\mathsf{zero}(P) \subseteq \overline{\mathbb{Q}}^n$
$P \subseteq \overline{\mathbb{Q}}[x_1, \ldots, x_n]$
subsets of $\overline{\mathbb{Q}}^n$
subsets of $\overline{\mathbb{Q}}[x_1, \ldots, x_n]$

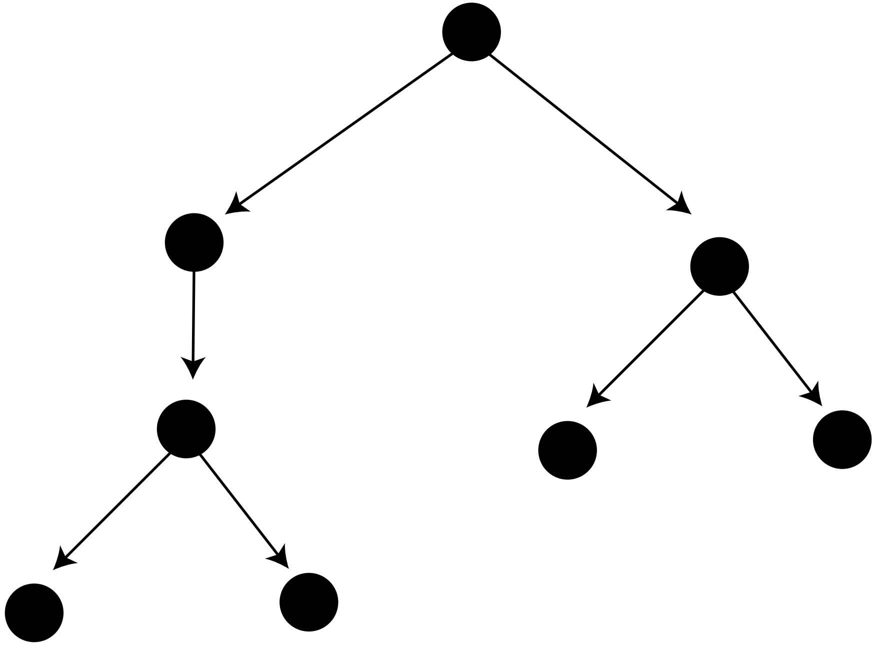
dimension of X sum of dimensions of $X_1, \dots, X_k$

$$p : \mathbb{F}^n \to \mathbb{F}^m$$

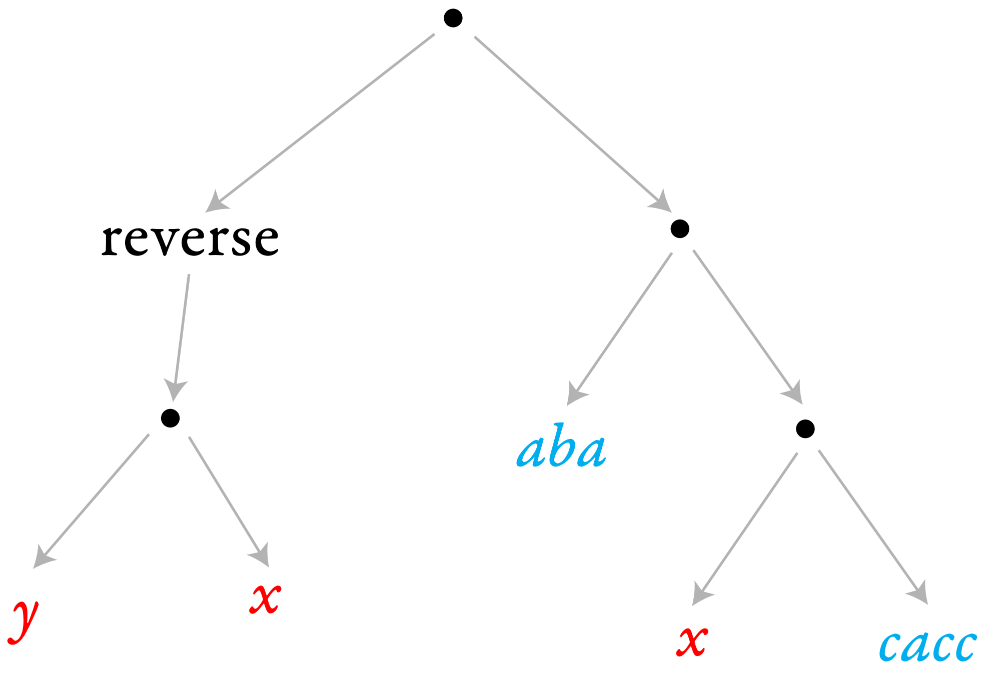
inputs two trees

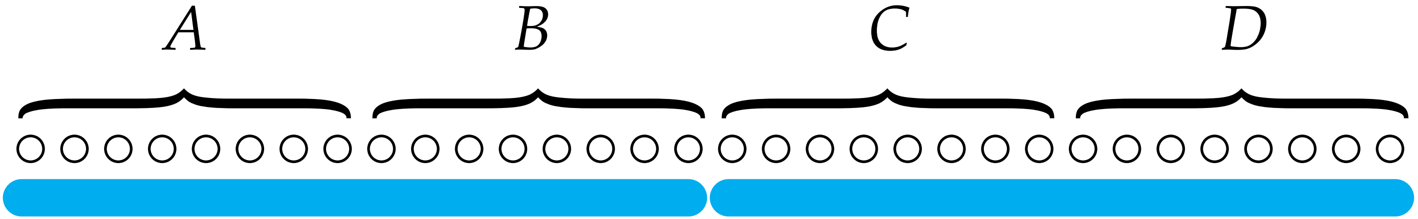


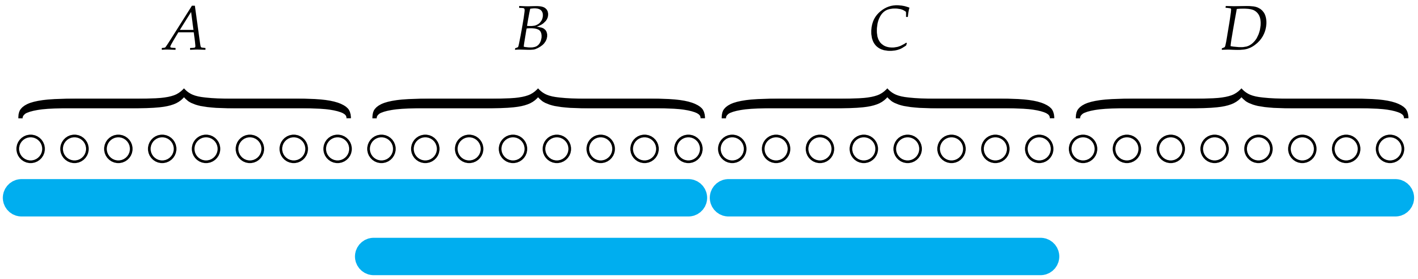
joins them into one

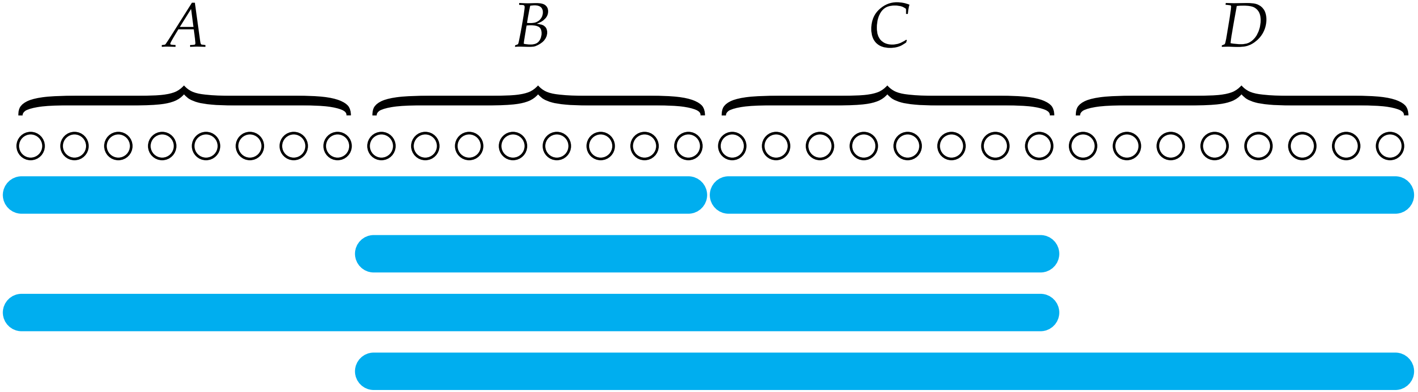


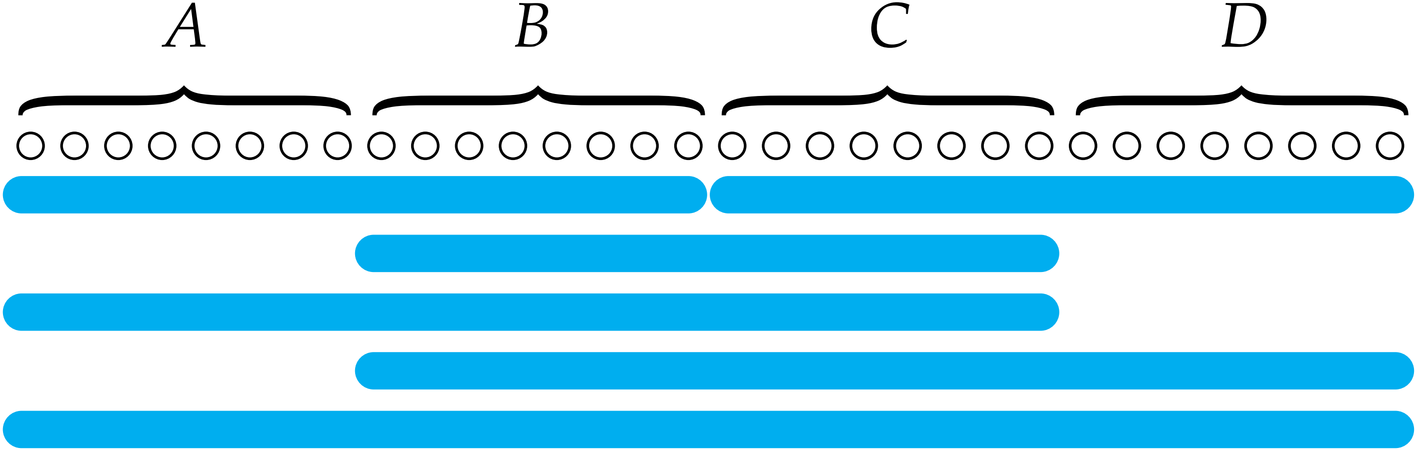
aba	elements of the universe of the algebra
x	variables
reverse	operations in the algebra

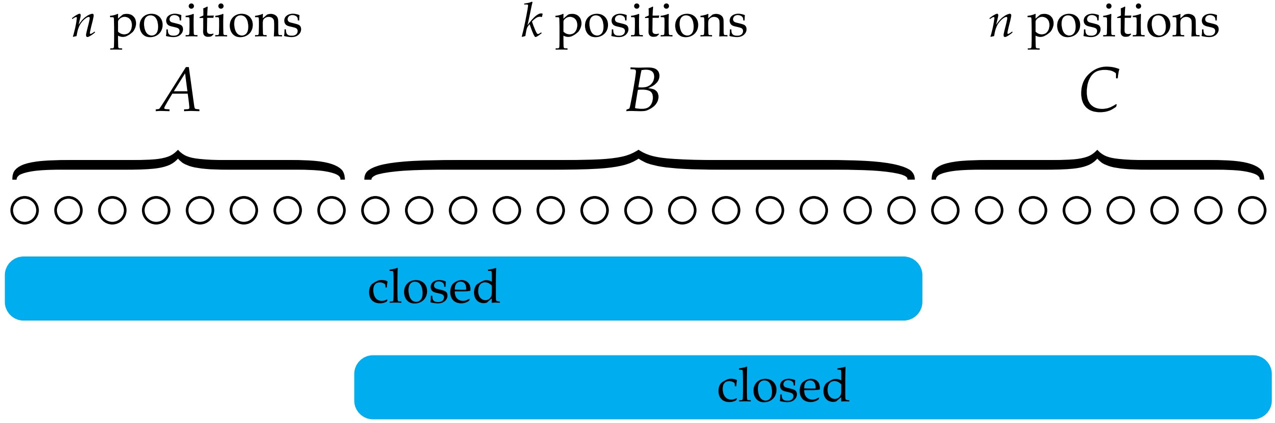


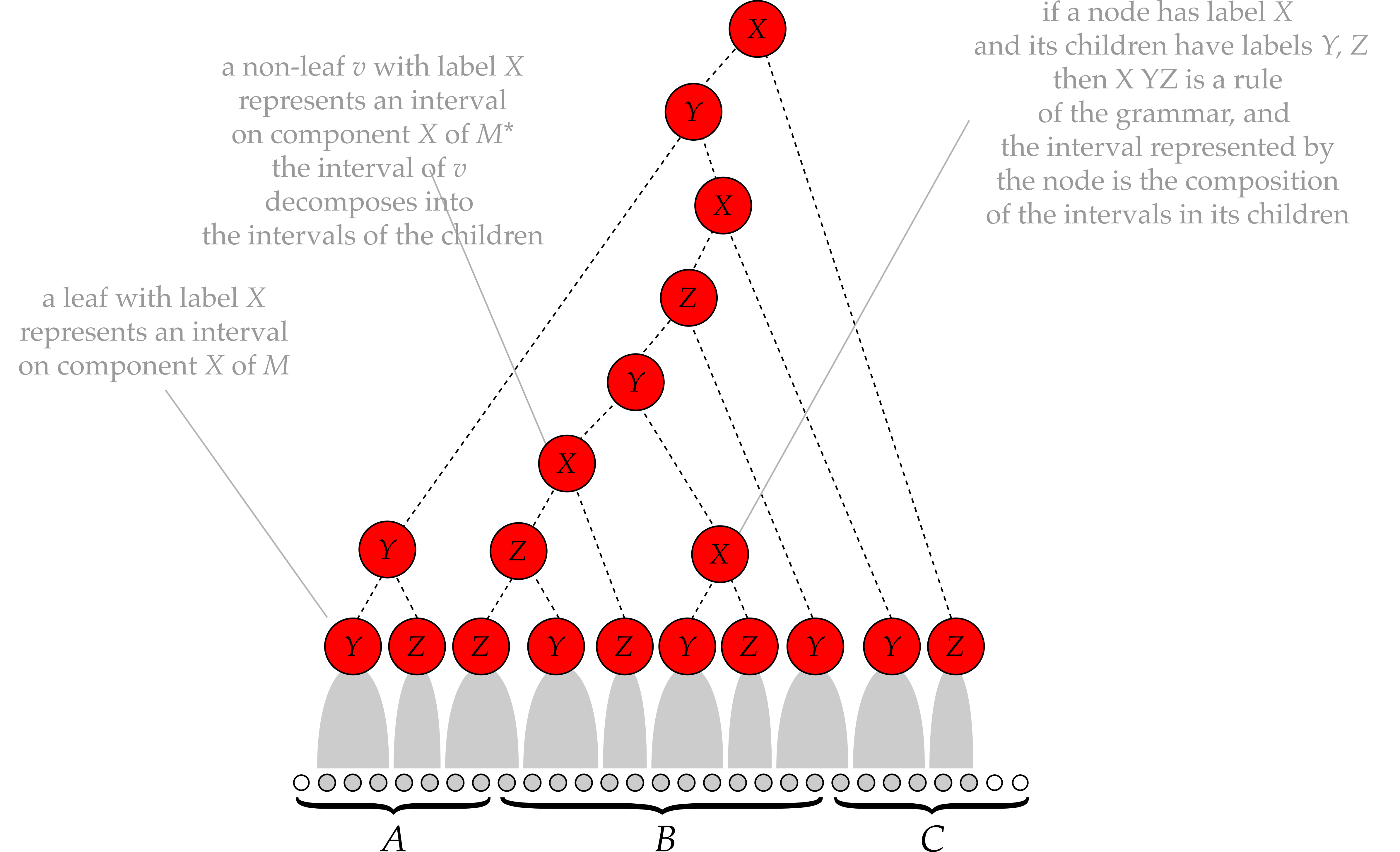

A
B
C
D

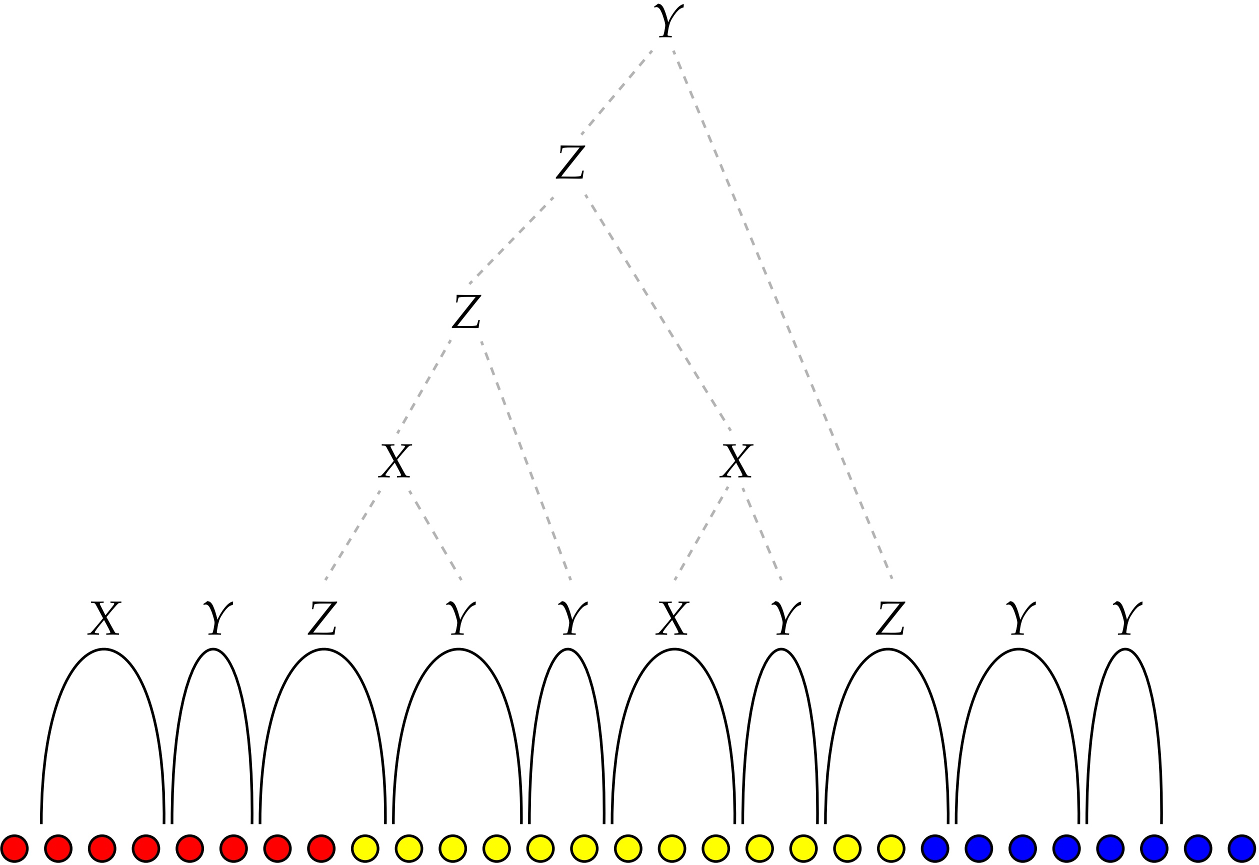

A
B
C
D

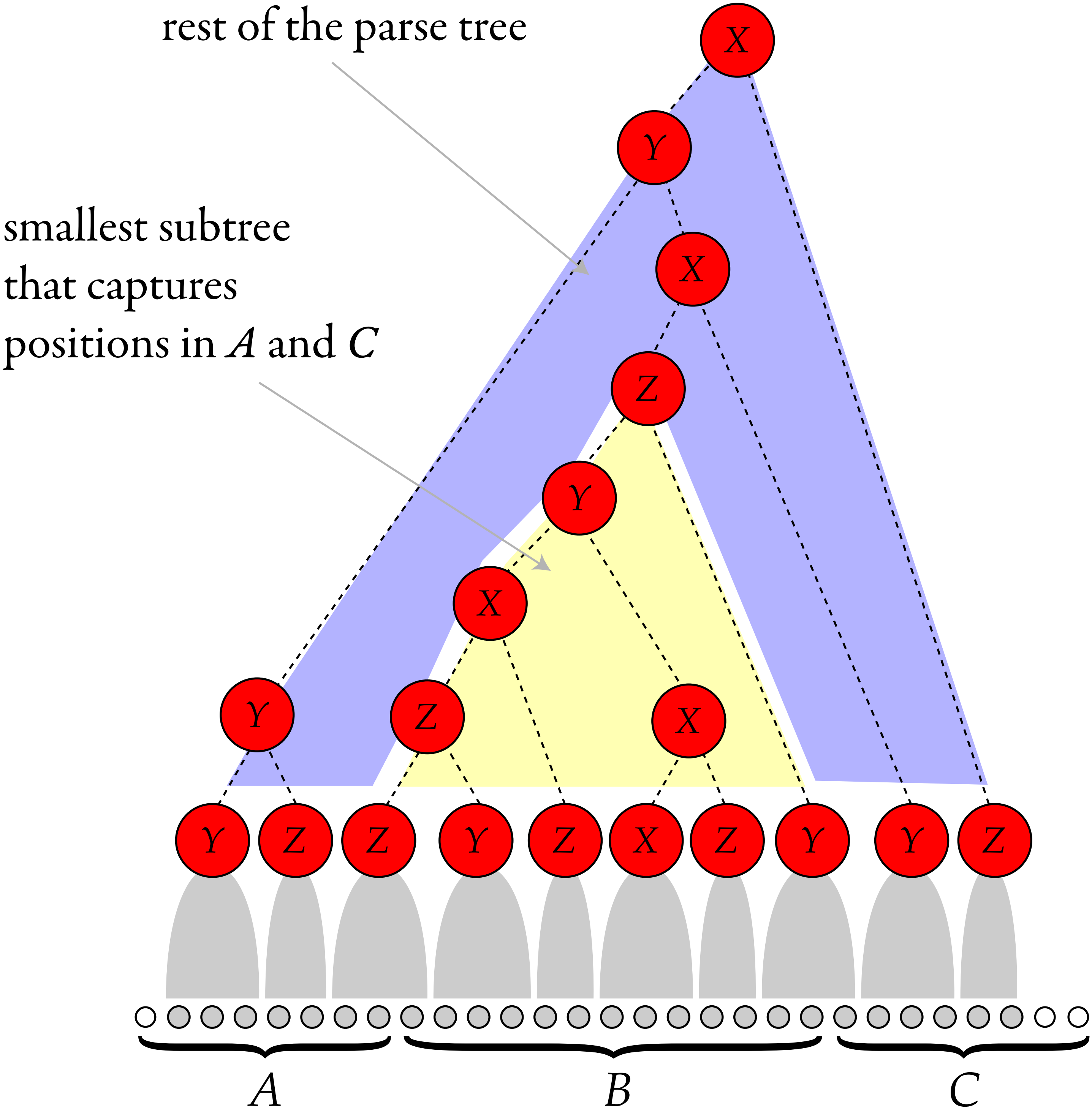

A
B
C
D


A
B
C
D

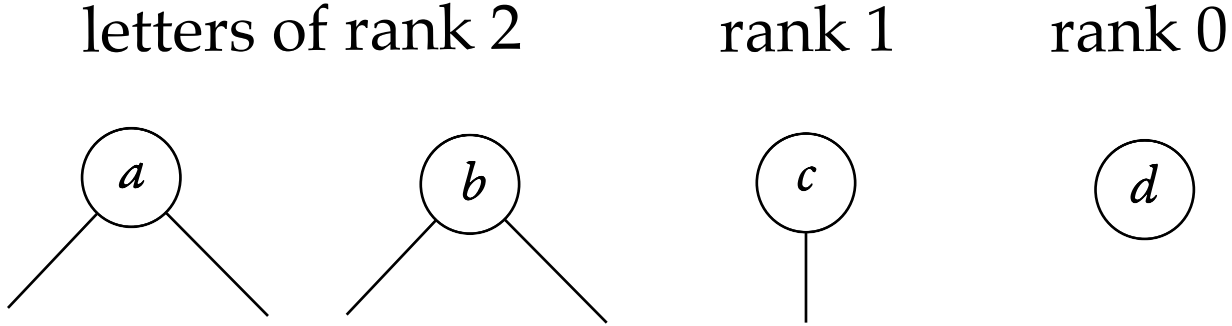

n positions
k positions
n positions
A
B
C
closed
closed

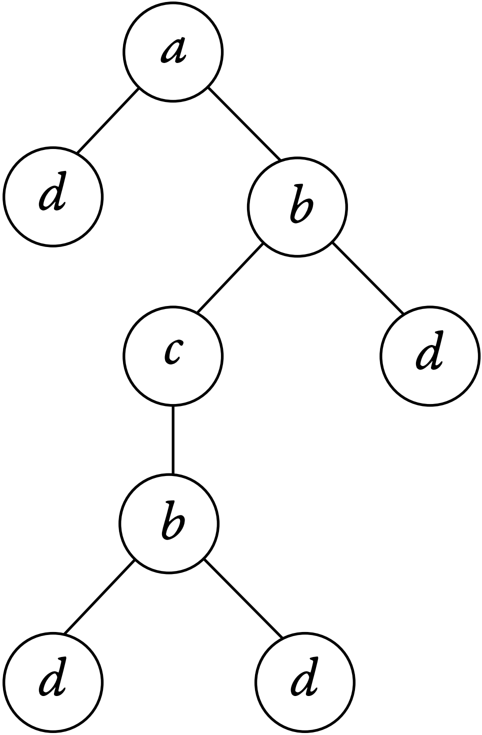

if a node has label X
and its children have labels Y, Z
then X YZ is a rule
of the grammar, and
the interval represented by
the node is the composition
of the intervals in its children
a non-leaf v with label X
represents an interval
on component X of M^*
the interval of v
decomposes into
the intervals of the children
a leaf with label X
represents an interval
on component X of M
X
Y
Z
A
B
C

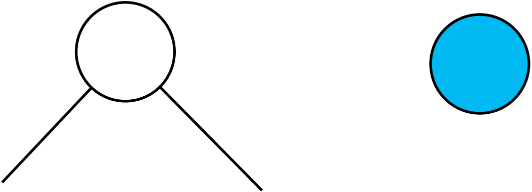

Y
Z
Z
X
X
X
Y
Z
Y
Y
X
Y
Z
Y
Y

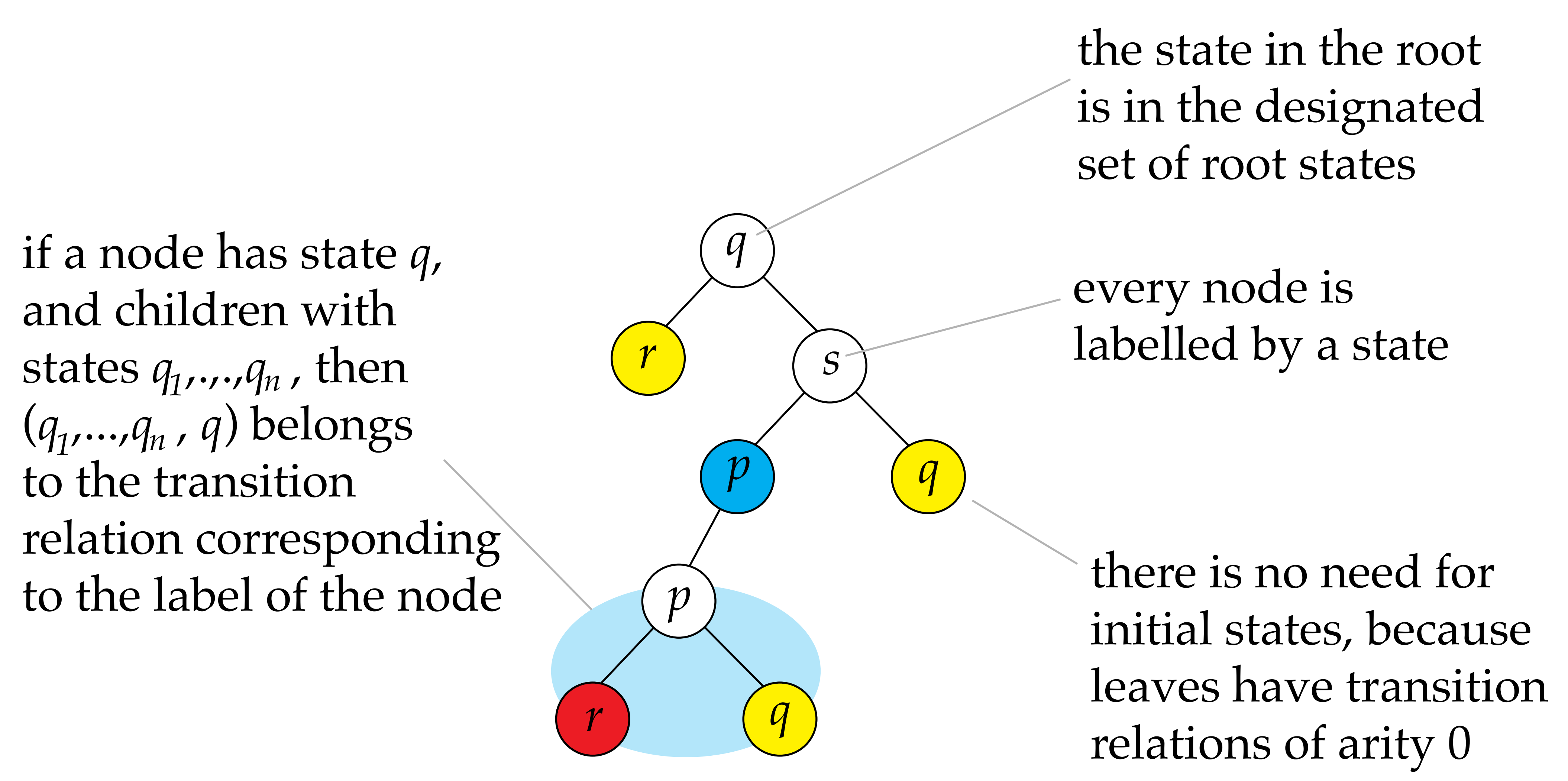


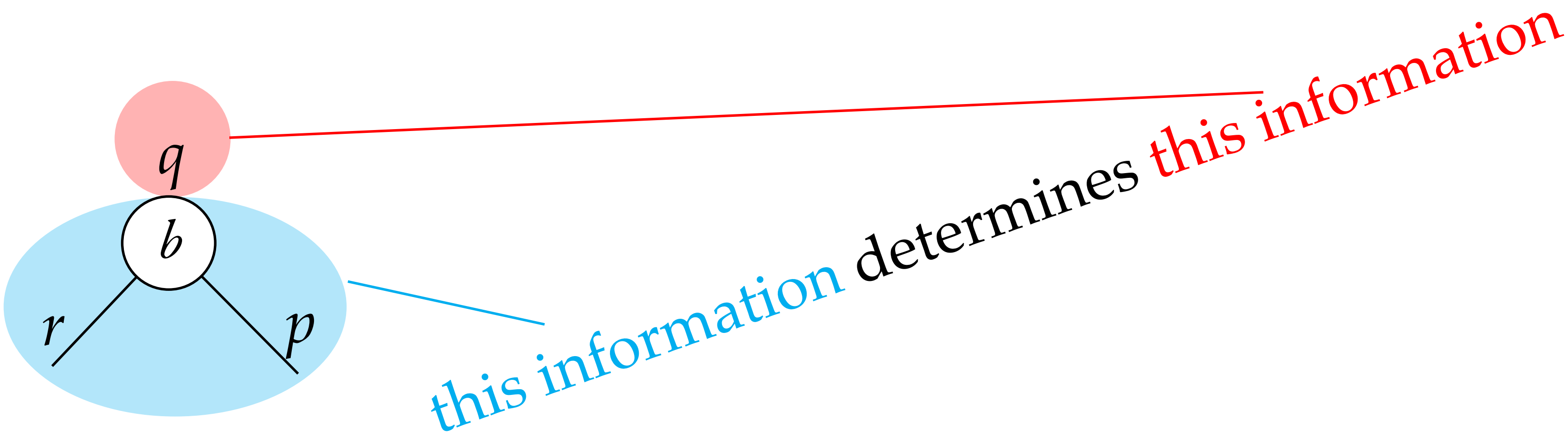
rest of the parse tree
smallest subtree
that captures
positions in A and C
X
Y
X
Z
Y
X
Y
Z
X
Y
Z
Z
Y
Z
X
Z
Y
Y
Z
A
B
C

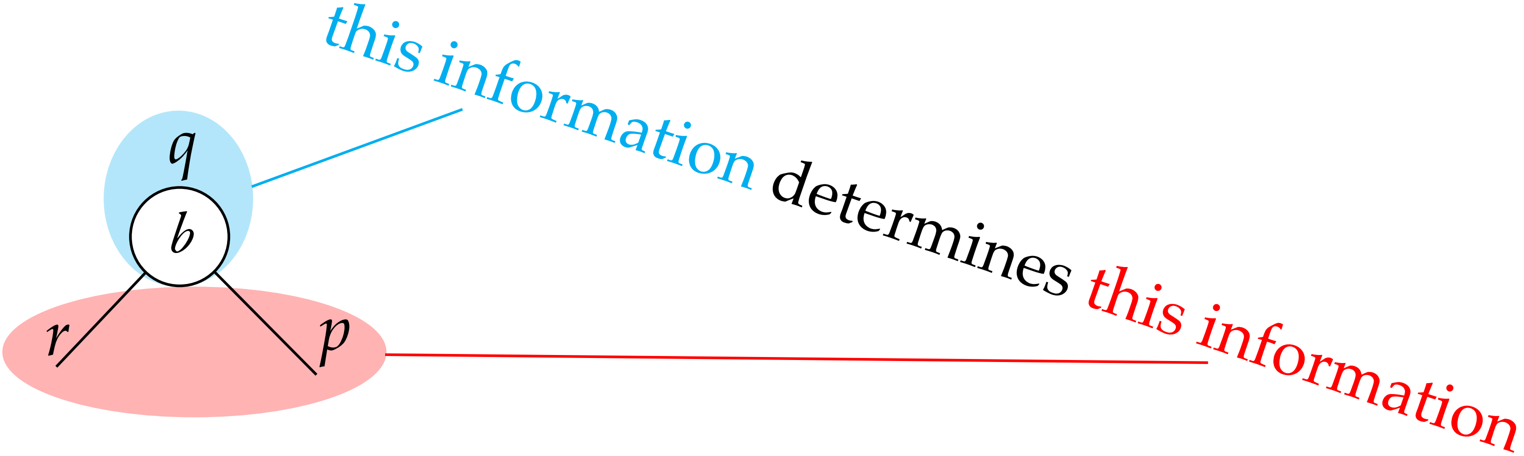

letters of rank 2
rank 1
rank 0
a
b
c
d


a
d
b
c
d
b
d
d




the state in the root is in the designated set of root states
every node is labelled by a state
there is no need for initial states, because leaves have transition relations of arity 0
if a node has state q, and children with states q_1,.,.,q_n , then (q_1,...,q_n , q) belongs to the transition relation corresponding to the label of the node
q
r
s
p
q
p
r
q


q
b
r
p
this information determines this information



this information determines this information
q
b
r
p

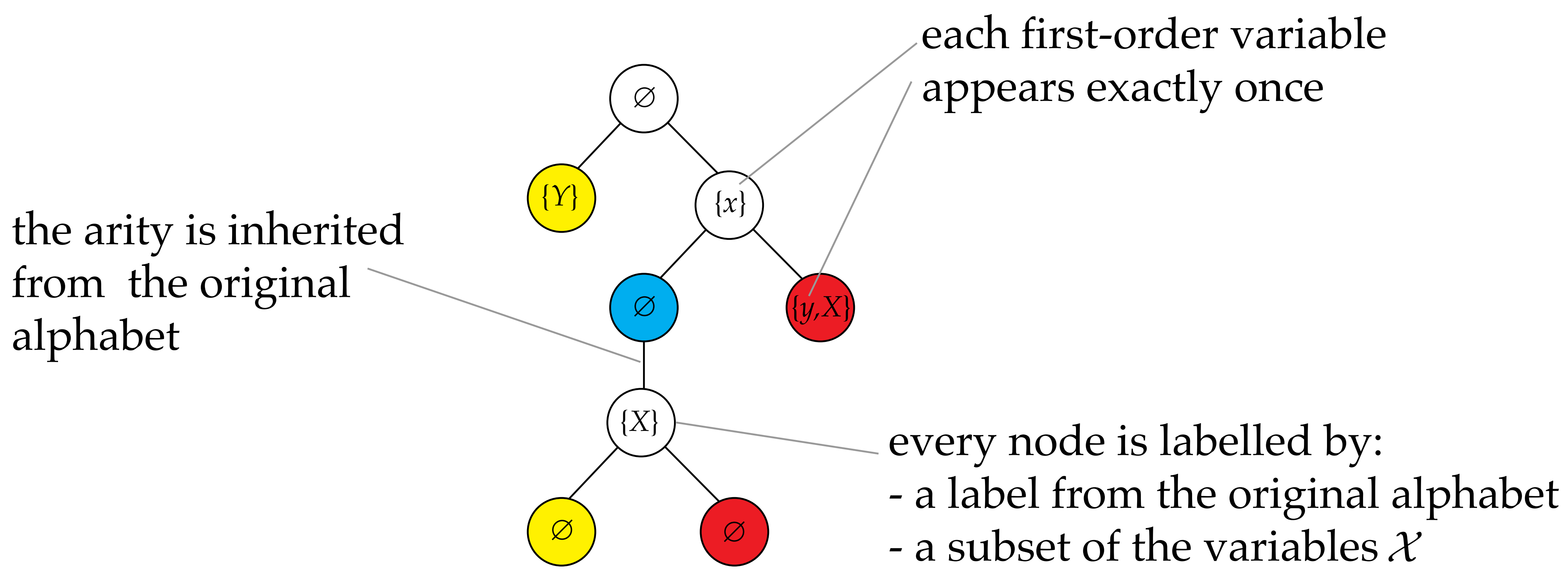
there exists a labelling of nodes with states

every node has exactly one state

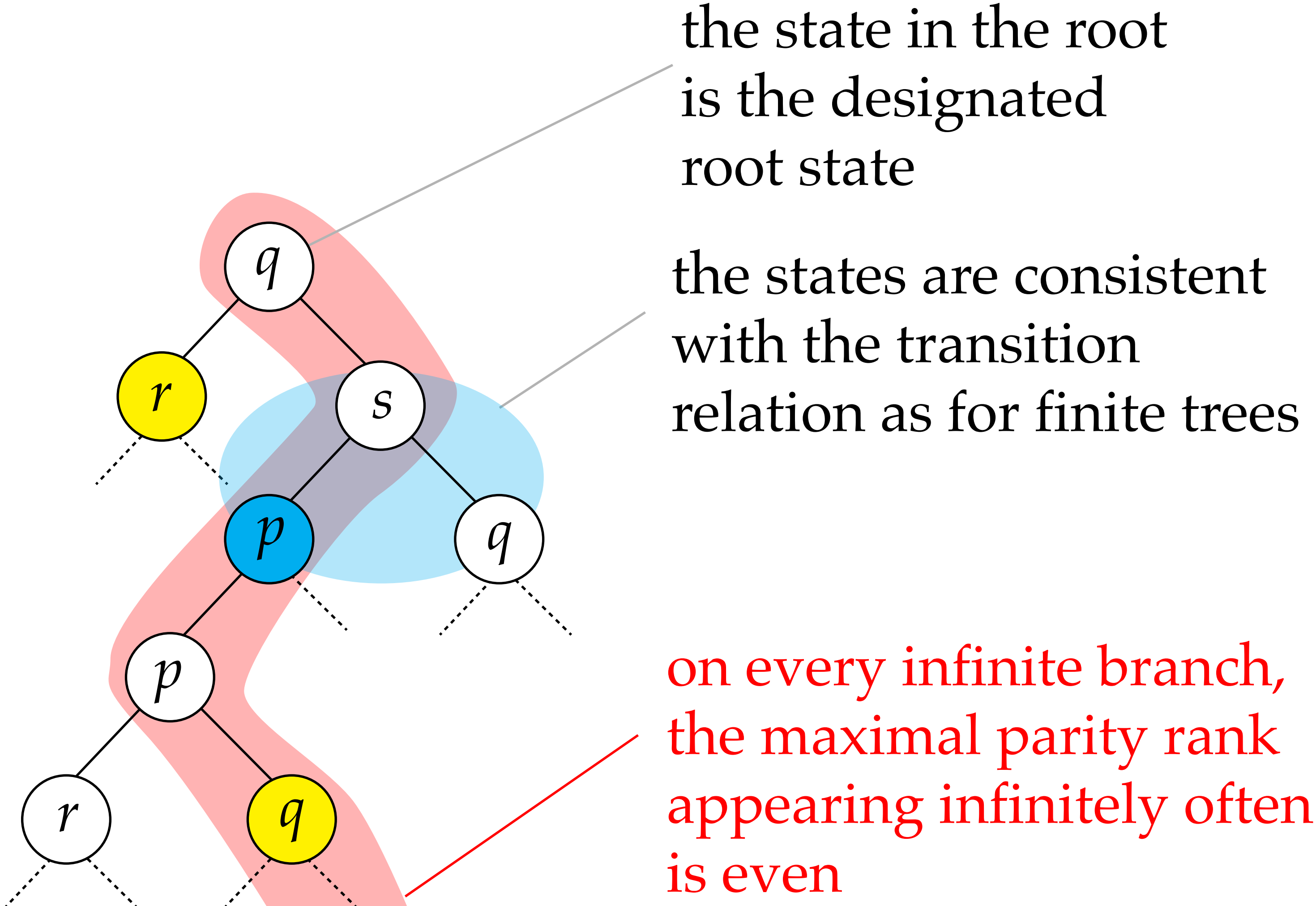
the root has a root state

for every node, a transition of the automaton is used

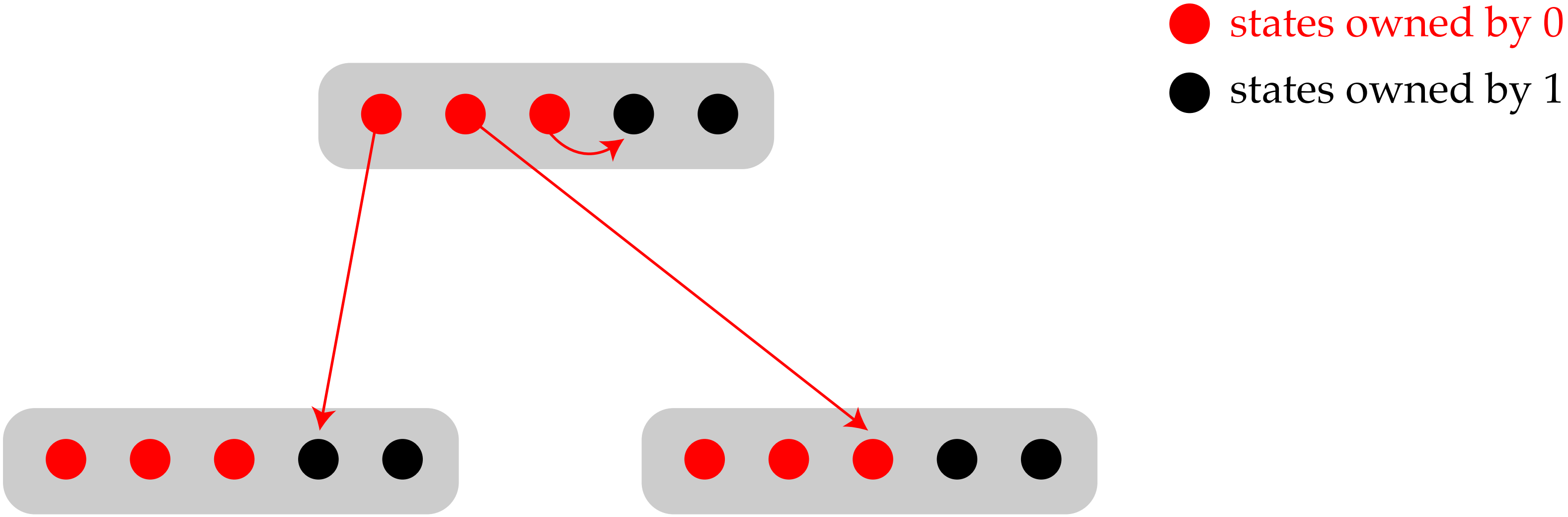
$$\exists X_1 \cdots \exists X_n \quad \wedge \left\{ \begin{array}{l} \forall x \bigvee_{q \in \{1,\ldots,n\}} x \in X_q \wedge \bigwedge_{p \neq q} x \notin X_p \\[2em] \forall x \; \mathrm{root}(x) \Rightarrow \bigvee_{i \in R} x \in X_i \\[2em] \bigwedge_{a \in \Sigma} \forall x \; a(x) \Rightarrow \bigvee_{(q_1,\ldots,q_k,q) \in \delta_a} \left(x \in X_q \wedge \bigwedge_{i \in \{1,\ldots,k\}} \mathrm{child}_i(x) \in X_{q_i} \right) \end{array} \right.$$

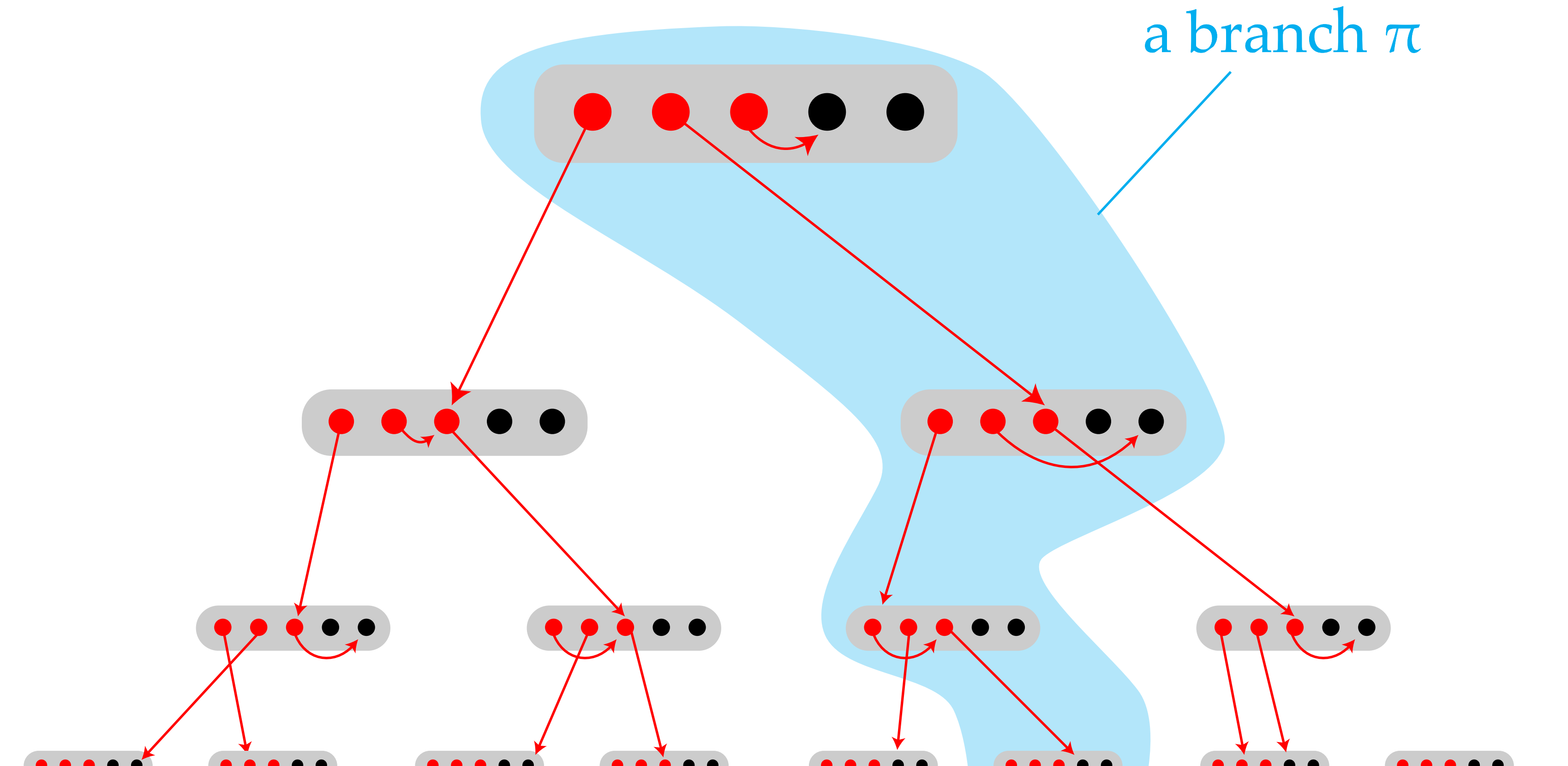


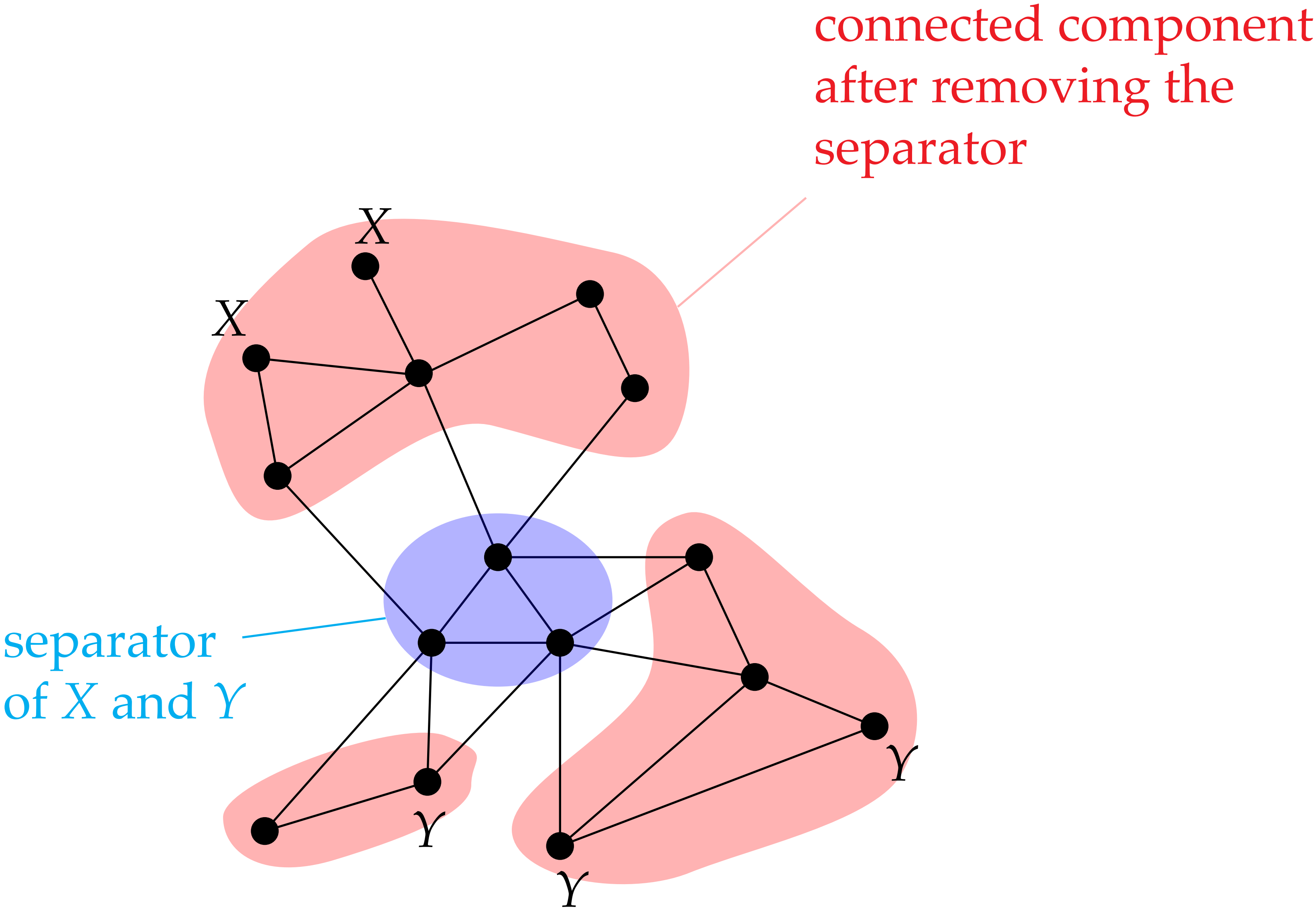
each first-order variable
appears exactly once
the arity is inherited
from the original
alphabet
every node is labelled by:
- a label from the original alphabet
- a subset of the variables $\mathcal{X}$
$\emptyset$
$\{Y\}$
$\{x\}$
$\emptyset$
$\{y,X\}$
$\{X\}$
$\emptyset$
$\emptyset$

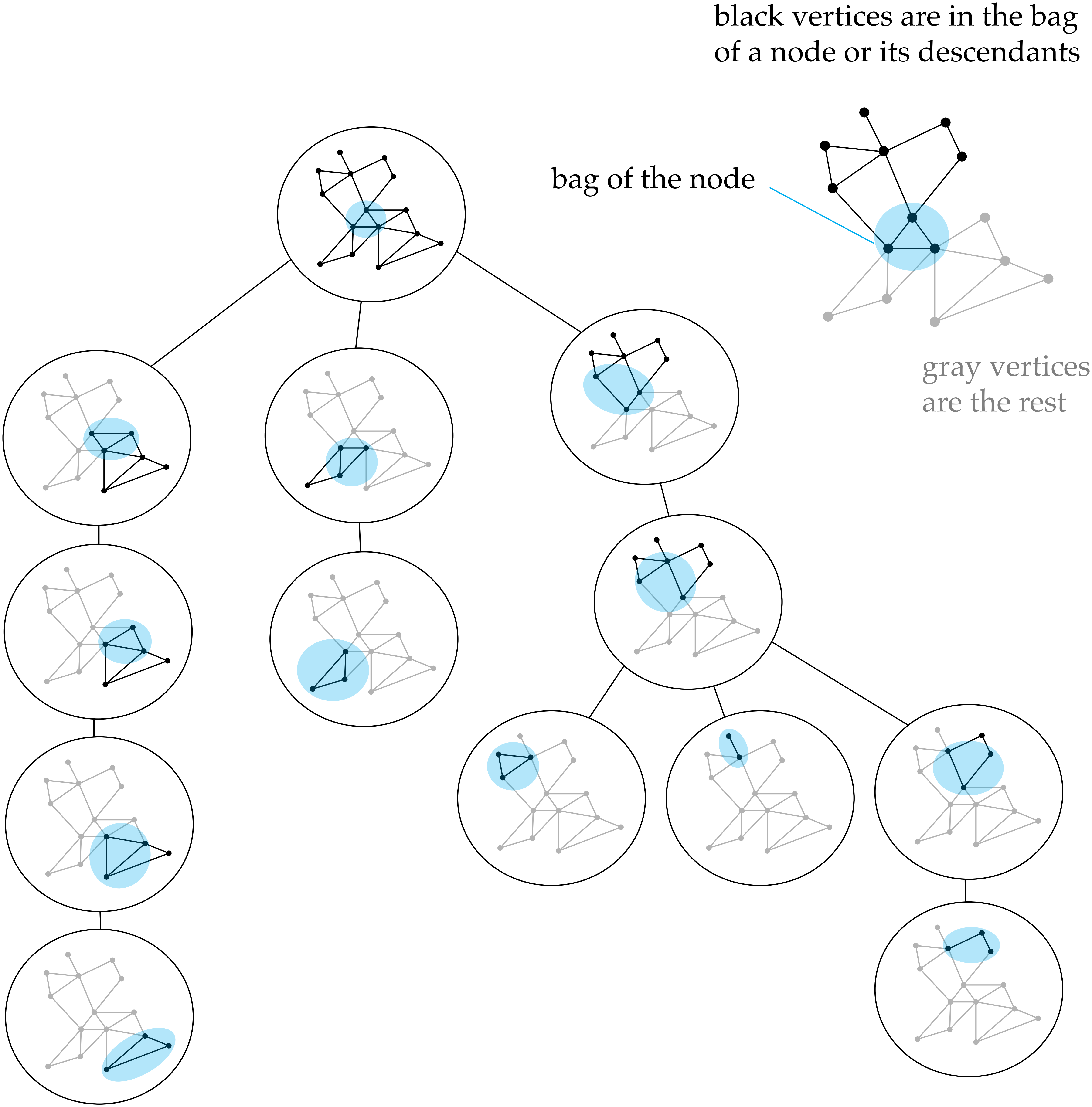


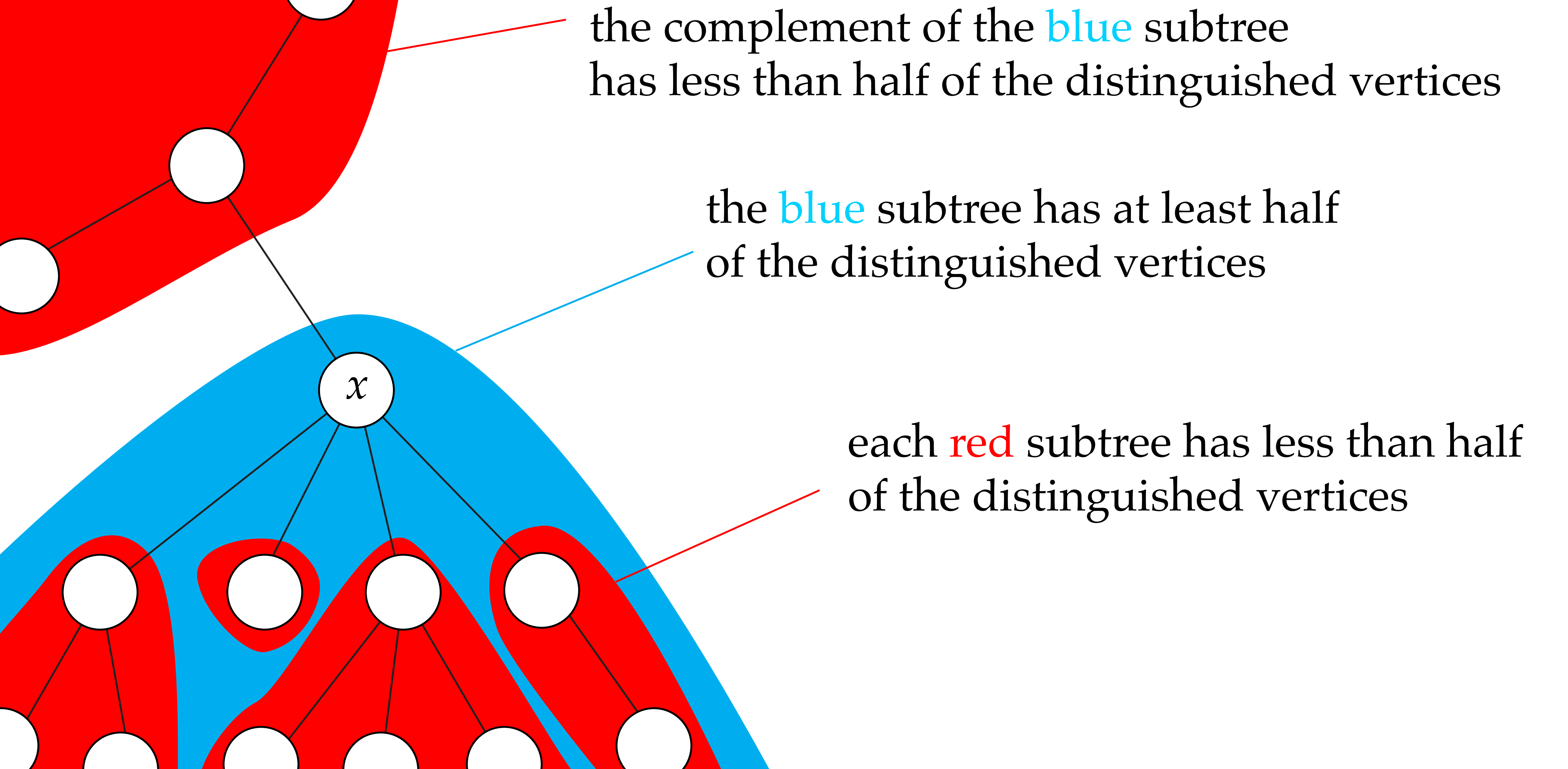
the state in the root
is the designated
root state
the states are consistent
with the transition
relation as for finite trees
on every infinite branch,
the maximal parity rank
appearing infinitely often
is even
q
r
s
p
q
p
r
q

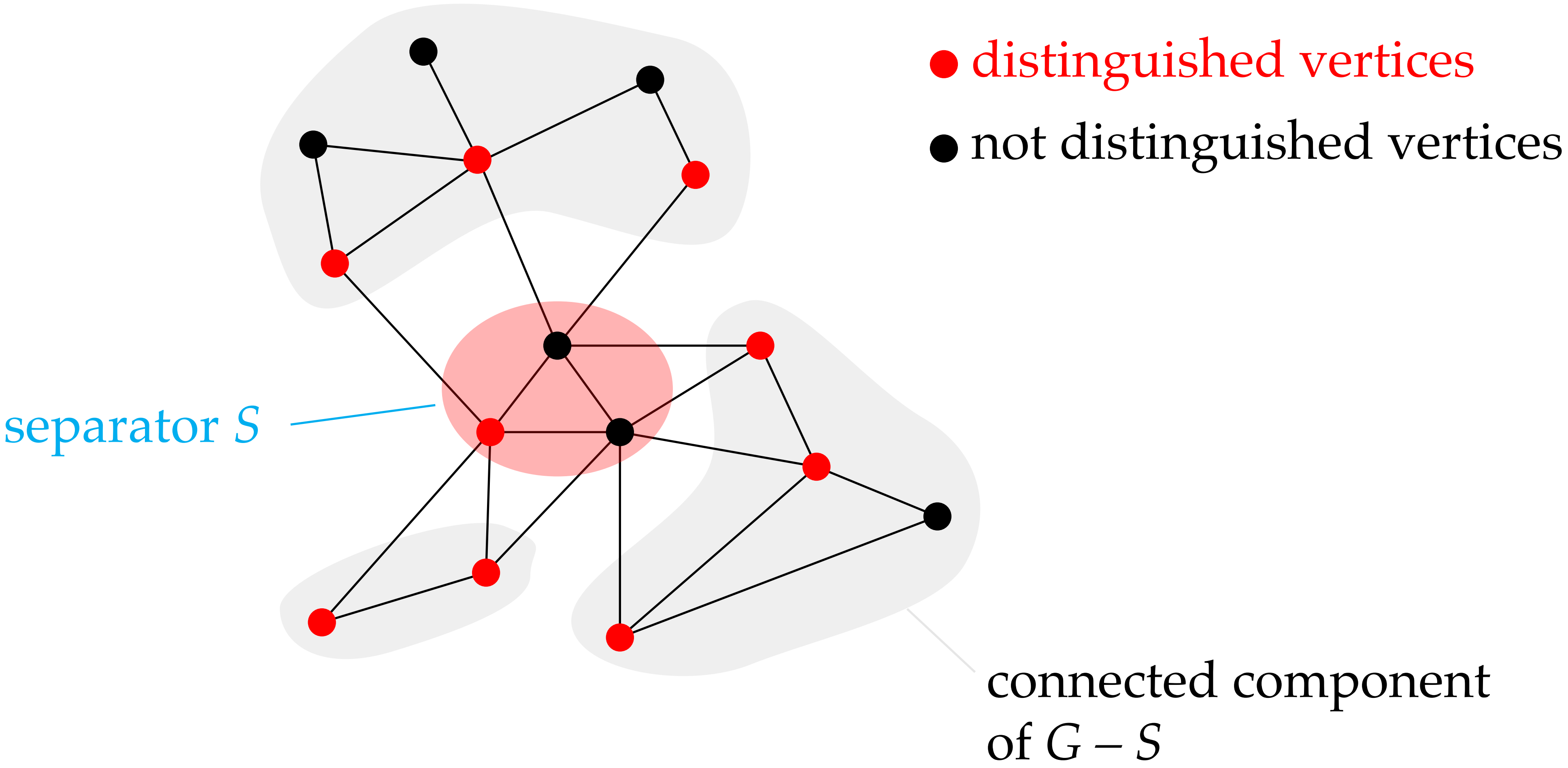

states owned by 0
states owned by 1

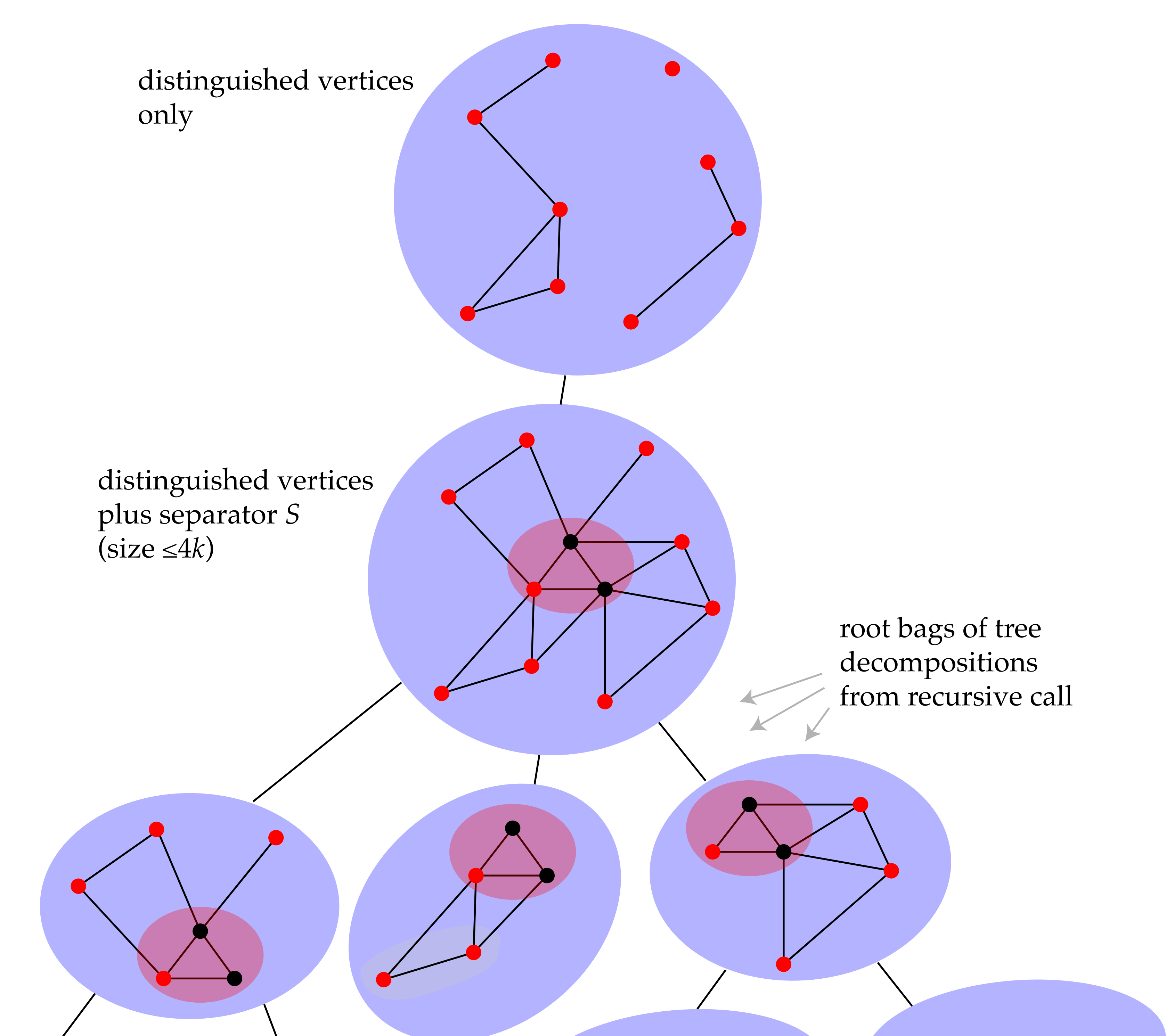

a branch π

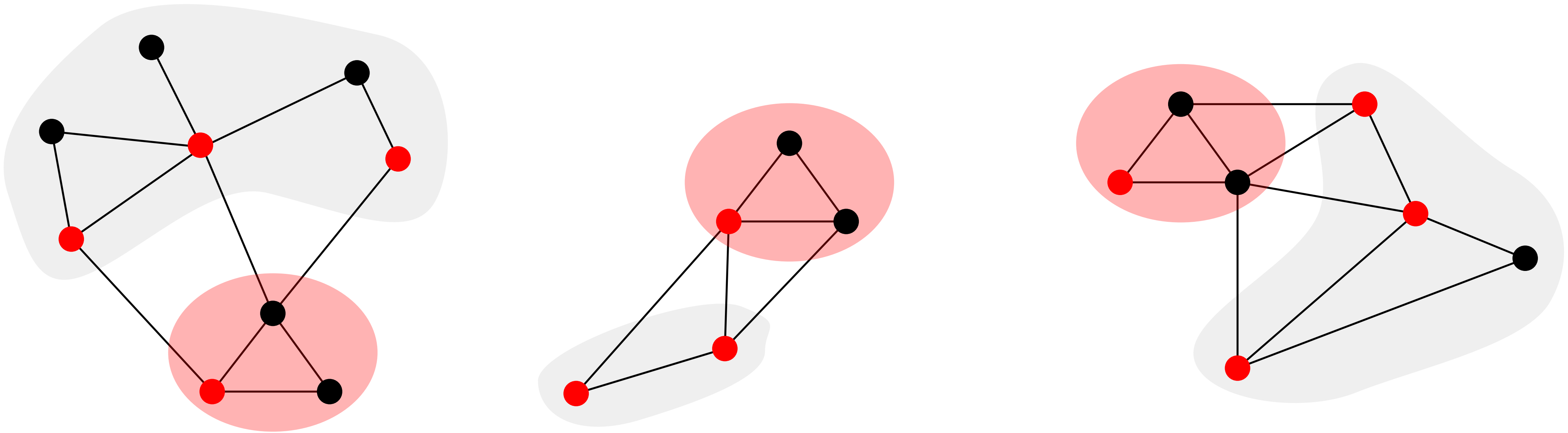

connected component after removing the separator
X
X
separator of X and Y
Y
Y
Y

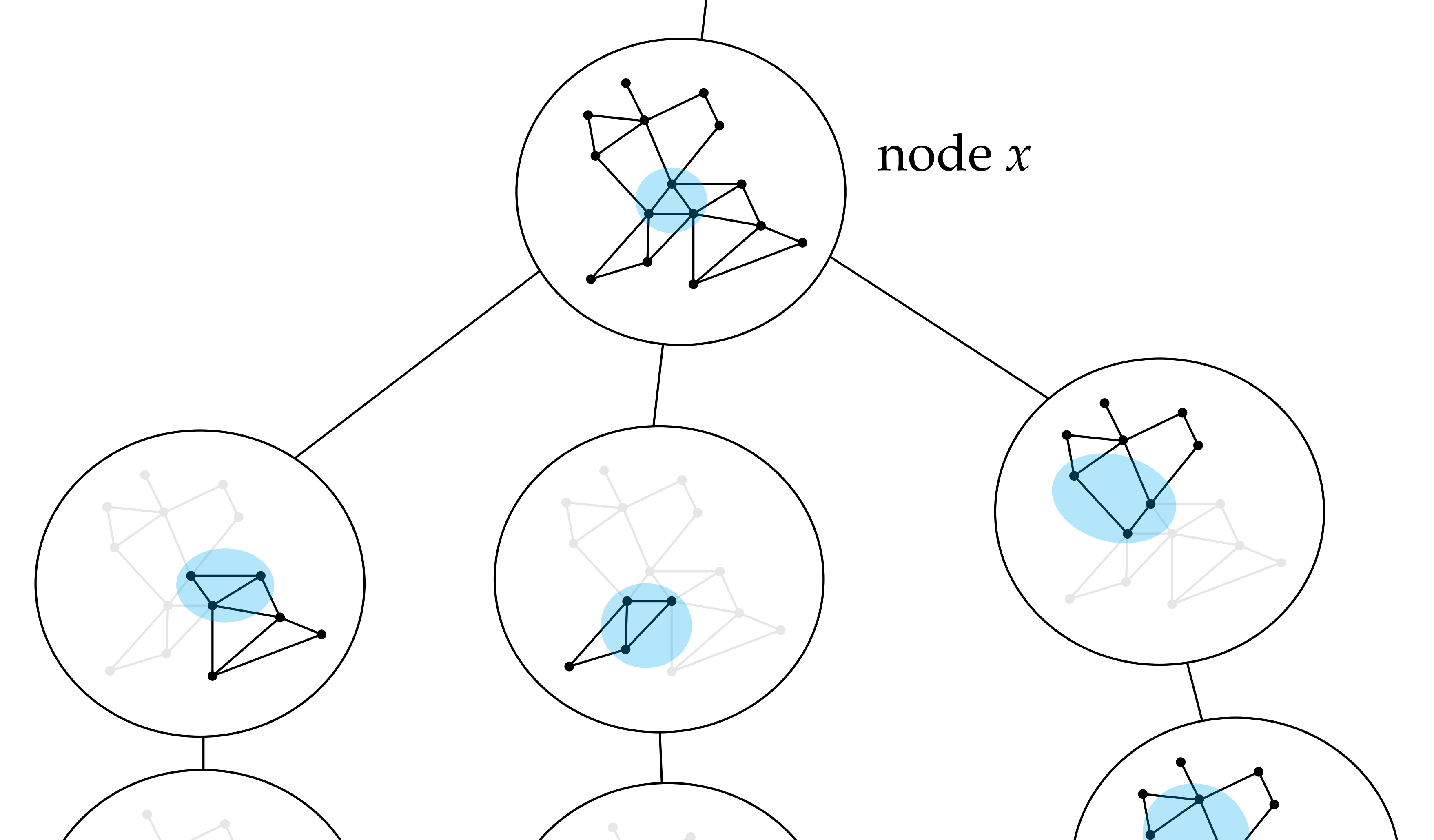

black vertices are in the bag
of a node or its descendants
bag of the node
gray vertices
are the rest

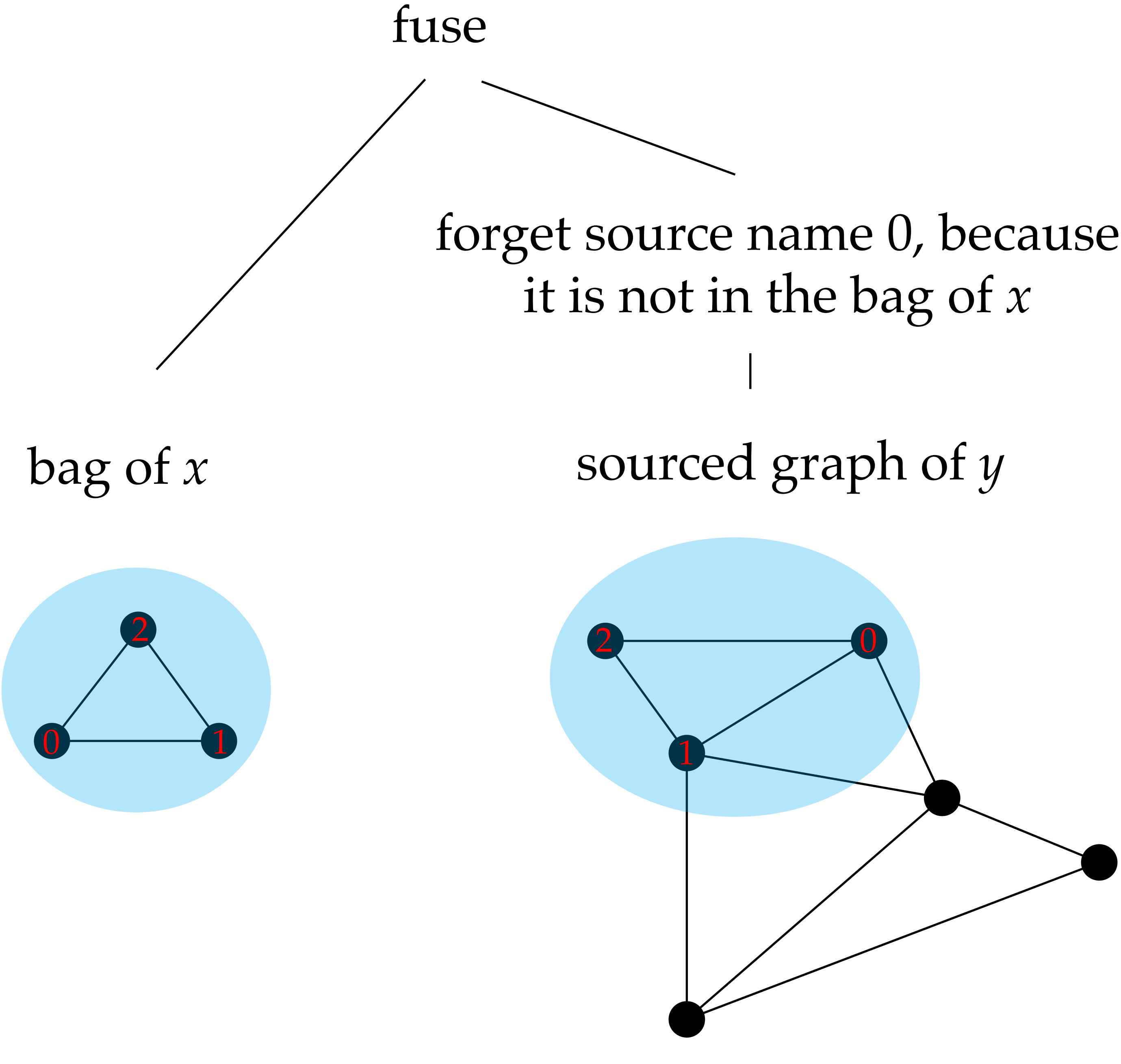

the complement of the blue subtree
has less than half of the distinguished vertices
the blue subtree has at least half
of the distinguished vertices
x
each red subtree has less than half
of the distinguished vertices

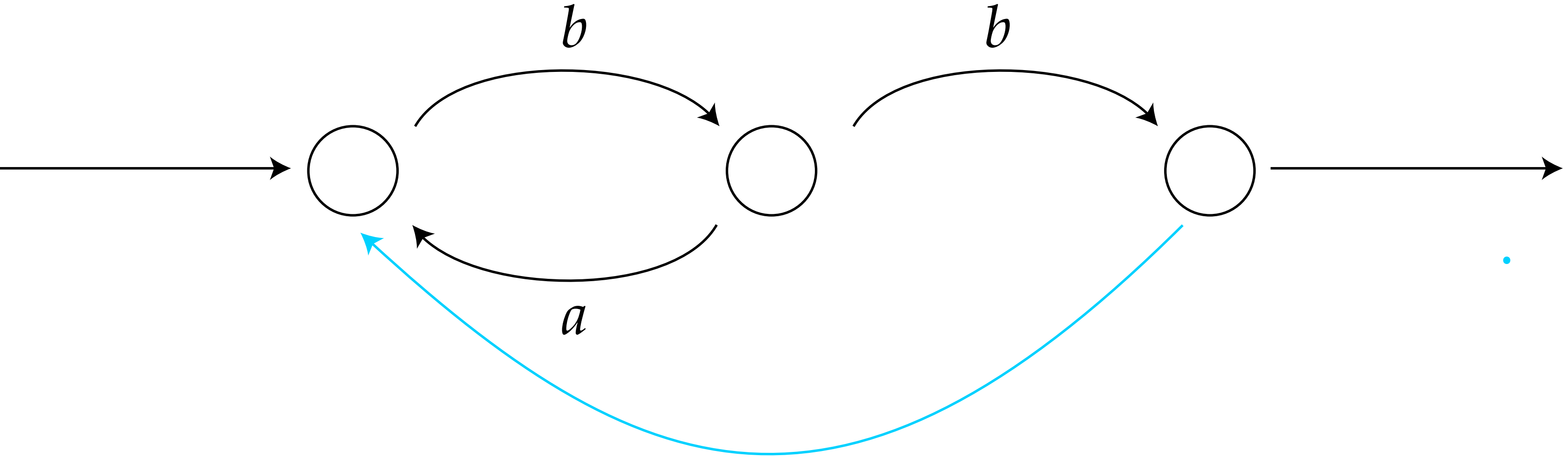

distinguished vertices
not distinguished vertices
separator S
connected component
of $G - S$


distinguished vertices only
distinguished vertices plus separator S (size $\leq 4k$)
root bags of tree decompositions from recursive call

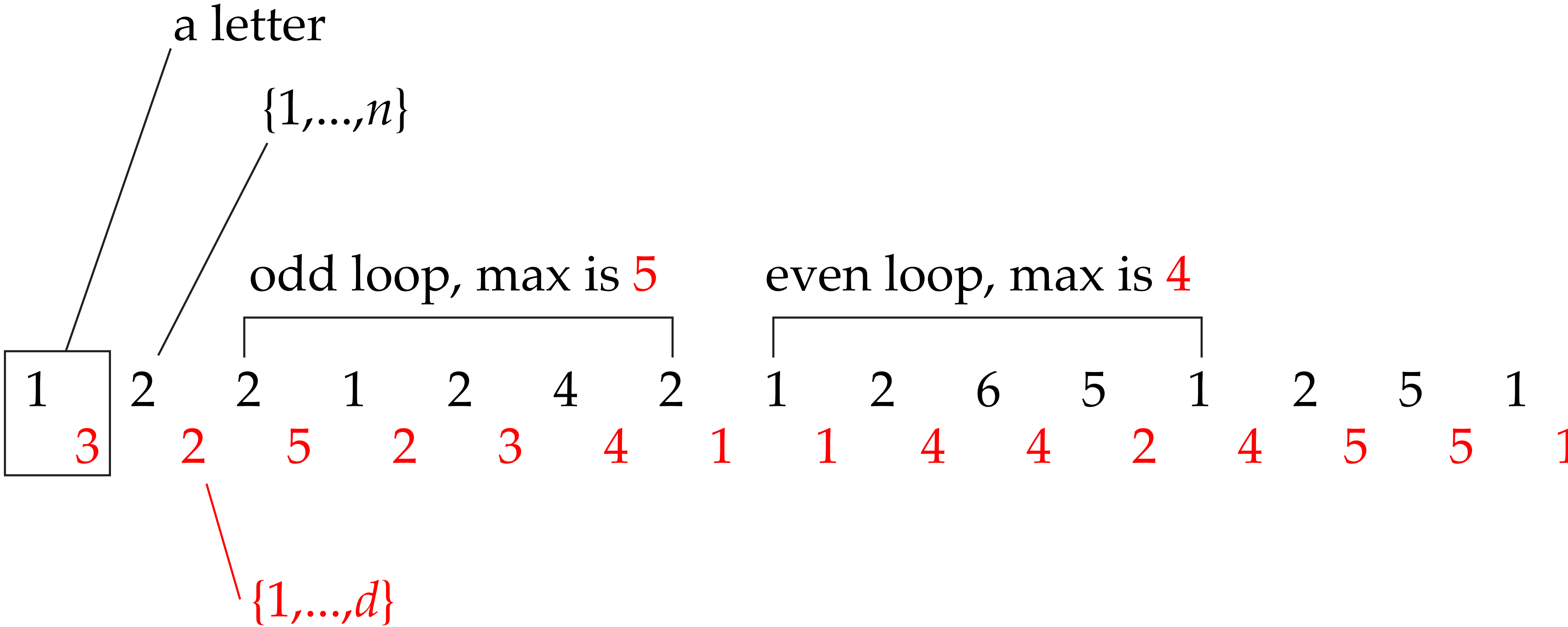


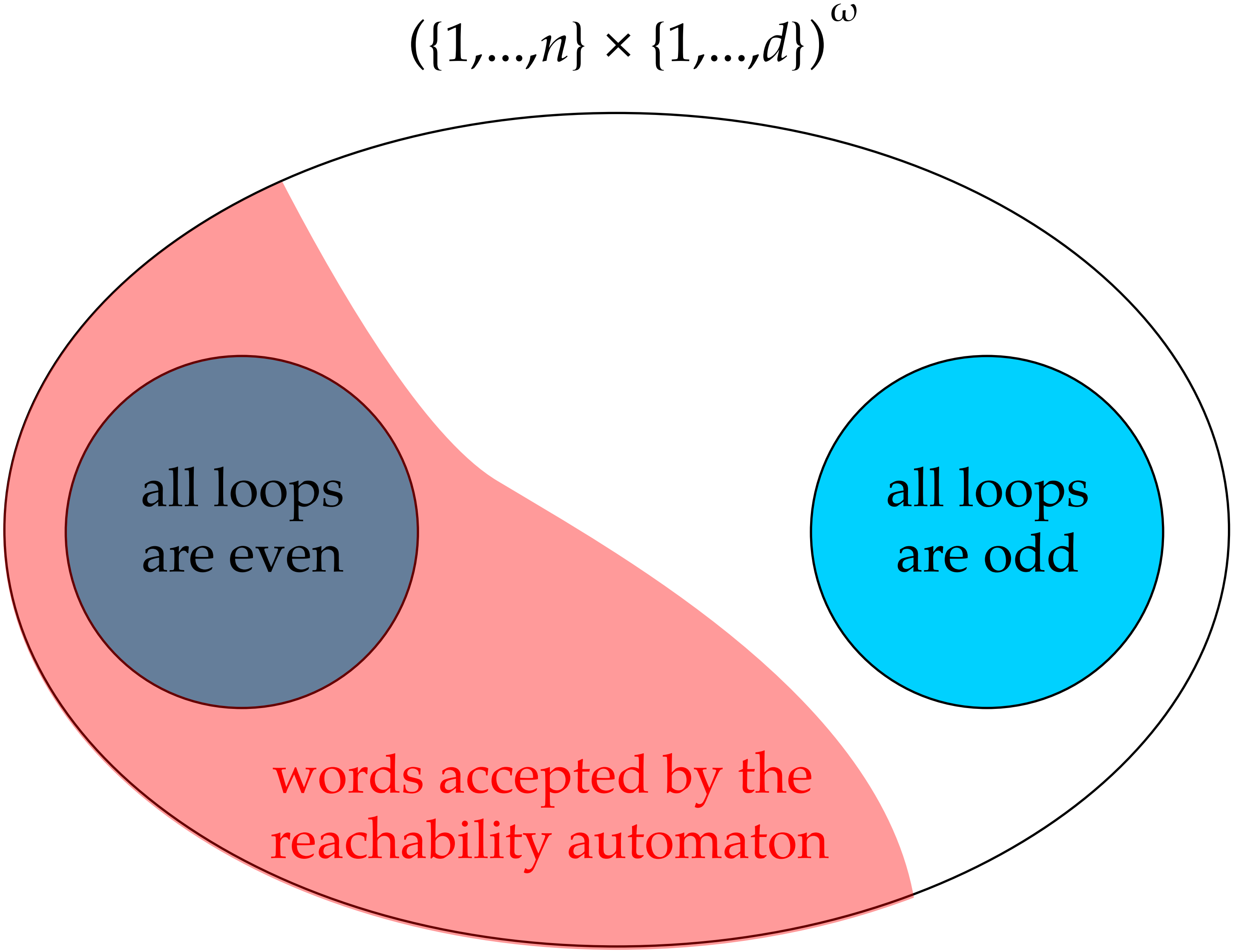

node x

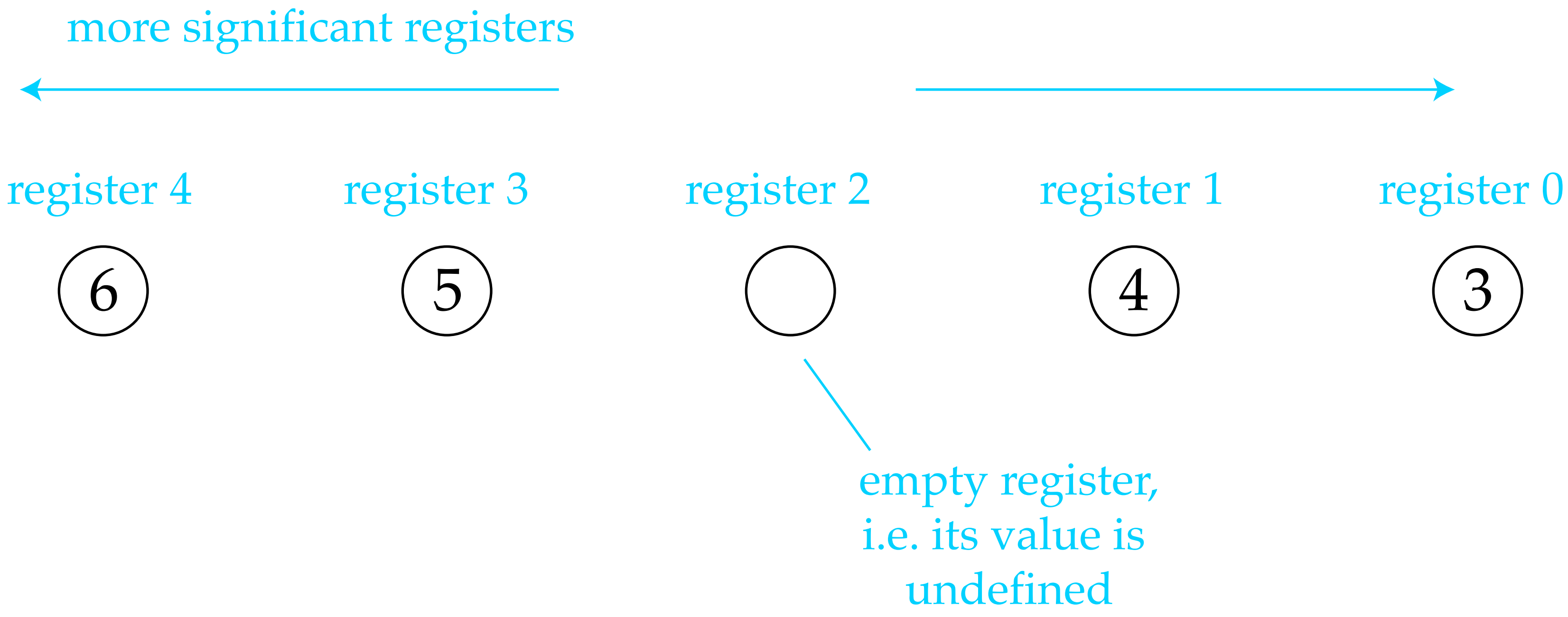

fuse
forget source name 0, because it is not in the bag of x
bag of x
sourced graph of y
2
0
1
2
0
1

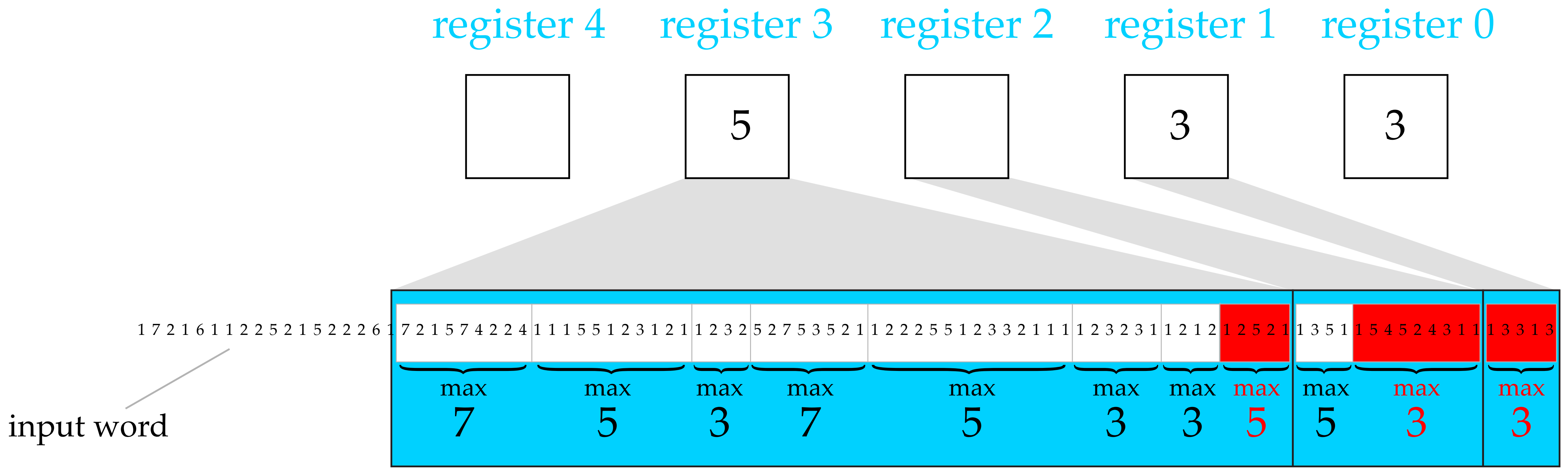


the blue arrow is not necessary,
because once the automaton
enters an accepting state, the
computation ends with acceptance

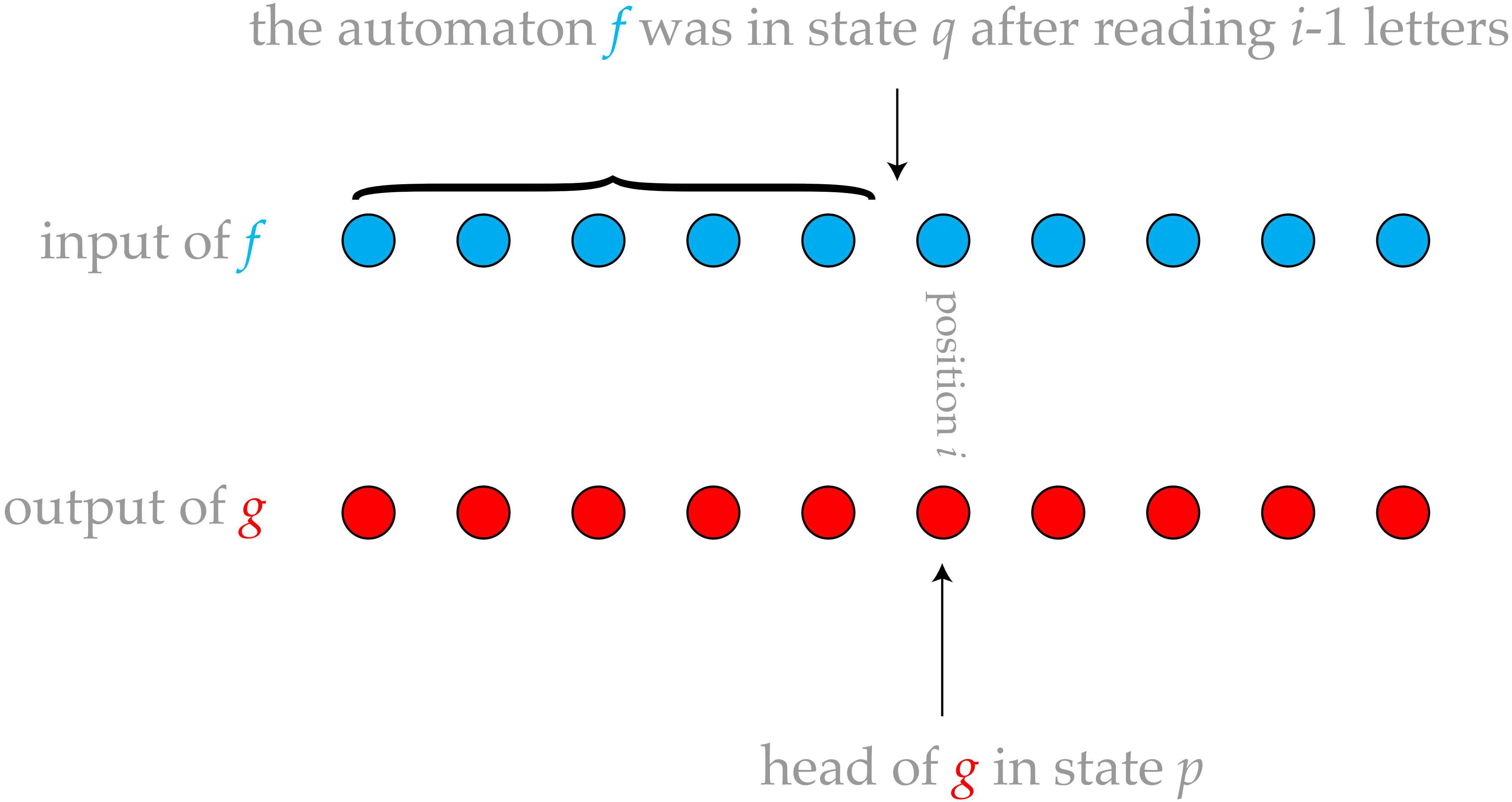

a letter
{1,...,n}
odd loop, max is 5
even loop, max is 4
1 2 2 1 2 4 2 1 2 6 5 1 2 5 1
3 2 5 2 3 4 1 1 4 4 2 4 5 5 1
{1,...,d}

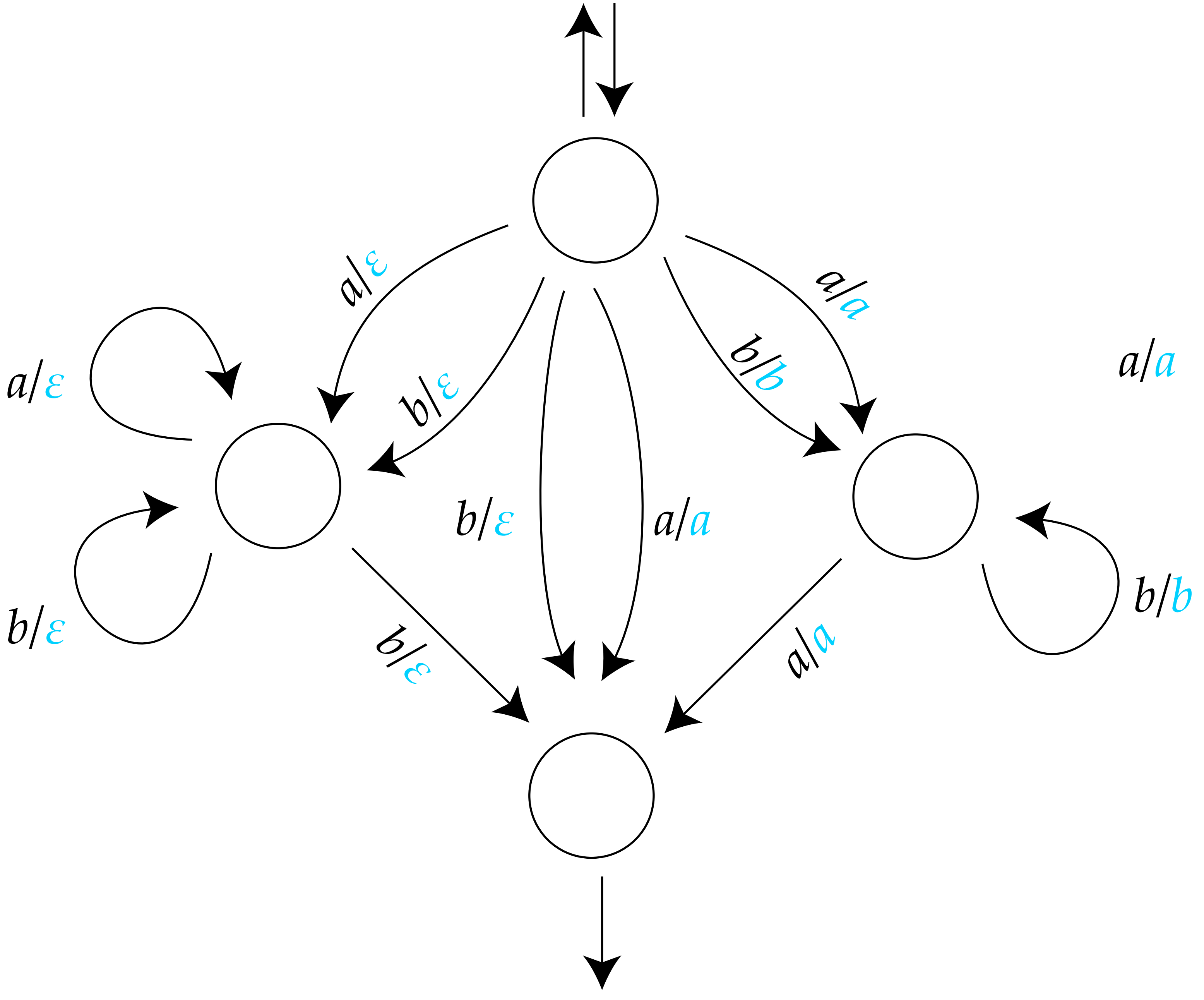

$(\{1,\ldots,n\} \times \{1,\ldots,d\})^{\omega}$
all loops are even
all loops are odd
words accepted by the reachability automaton


more significant registers
register 4
register 3
register 2
register 1
register 0
6
5
4
3
empty register,
i.e. its value is
undefined

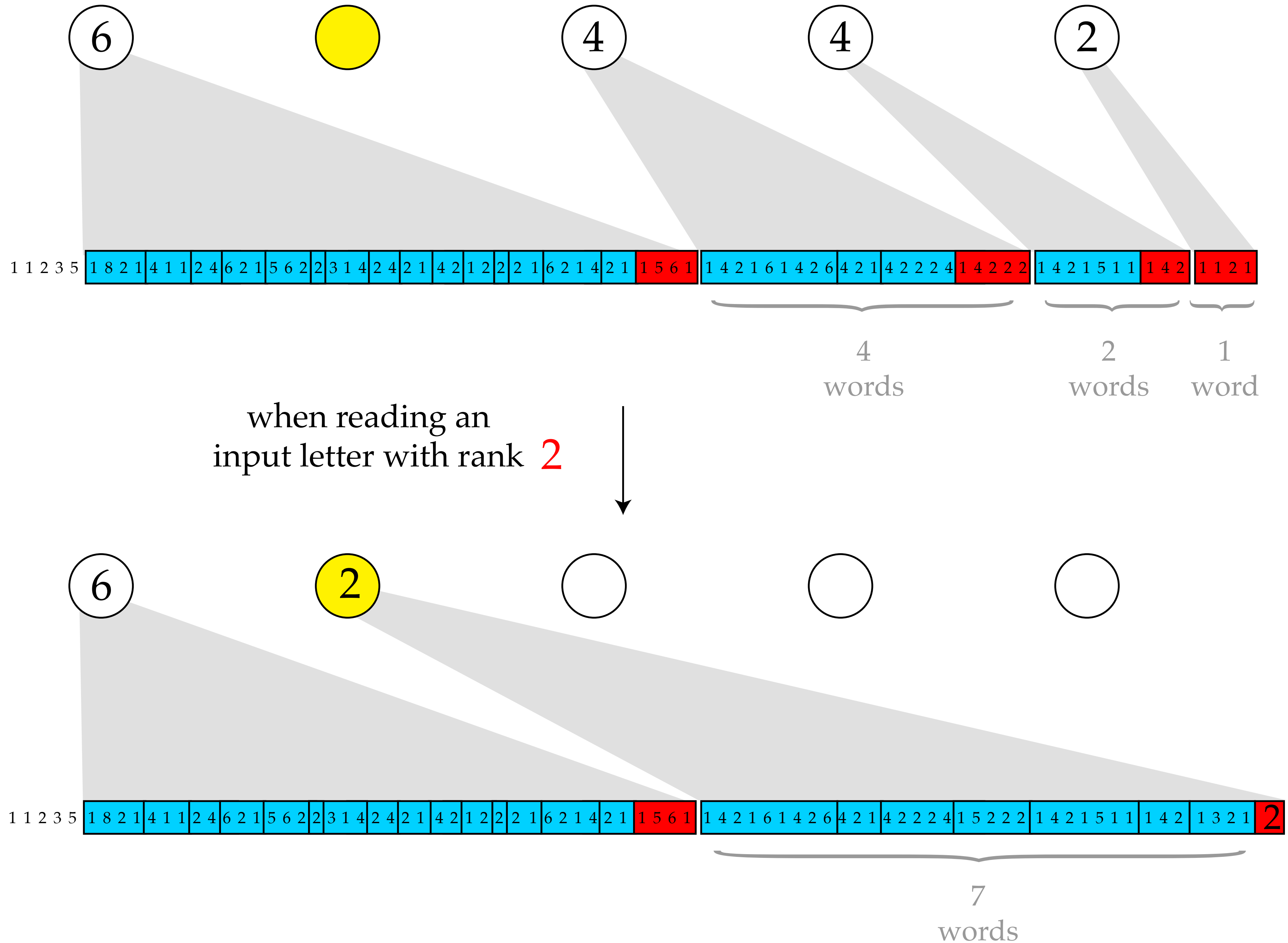


register 4
register 3
register 2
register 1
register 0
5
3
3
input word
1 7 2 1 6 1 1 2 2 5 2 1 5 2 2 2 6 1 7 2 1 5 7 4 2 2 4 1 1 1 5 5 1 2 3 1 2 1 1 2 3 2 5 2 7 5 3 5 2 1 1 2 2 2 5 5 1 2 3 3 2 1 1 1 1 2 3 2 3 1 1 2 1 2 1 2 5 2 1 1 3 5 1 1 5 4 5 2 4 3 1 1 1 3 3 1 3
max 7
max 5
max 3
max 7
max 5
max 3
max 3
max 5
max 5
max 3
max 3

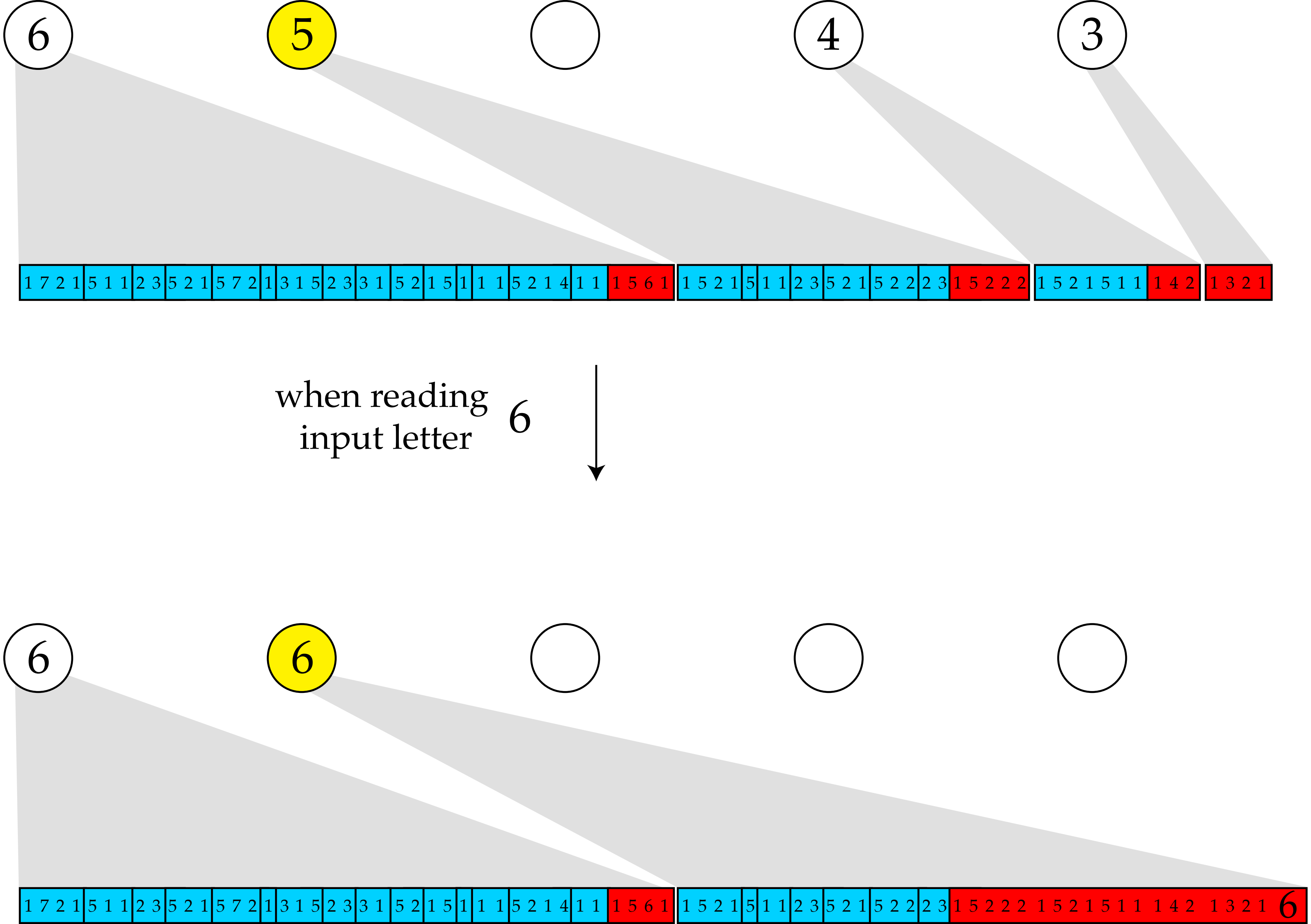

the automaton f was in state q after reading i-1 letters
input of f
position i
output of g
head of g in state p

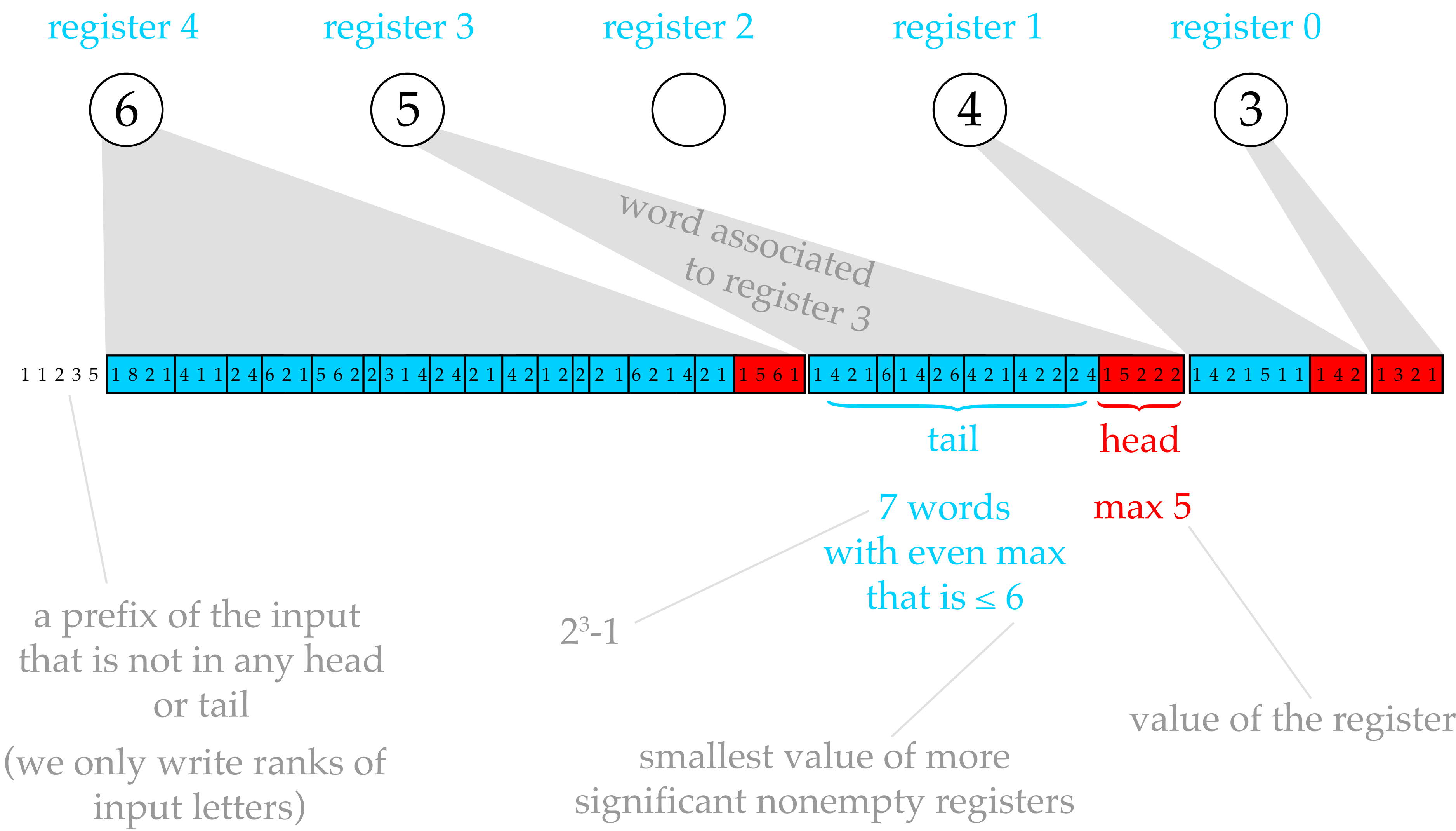


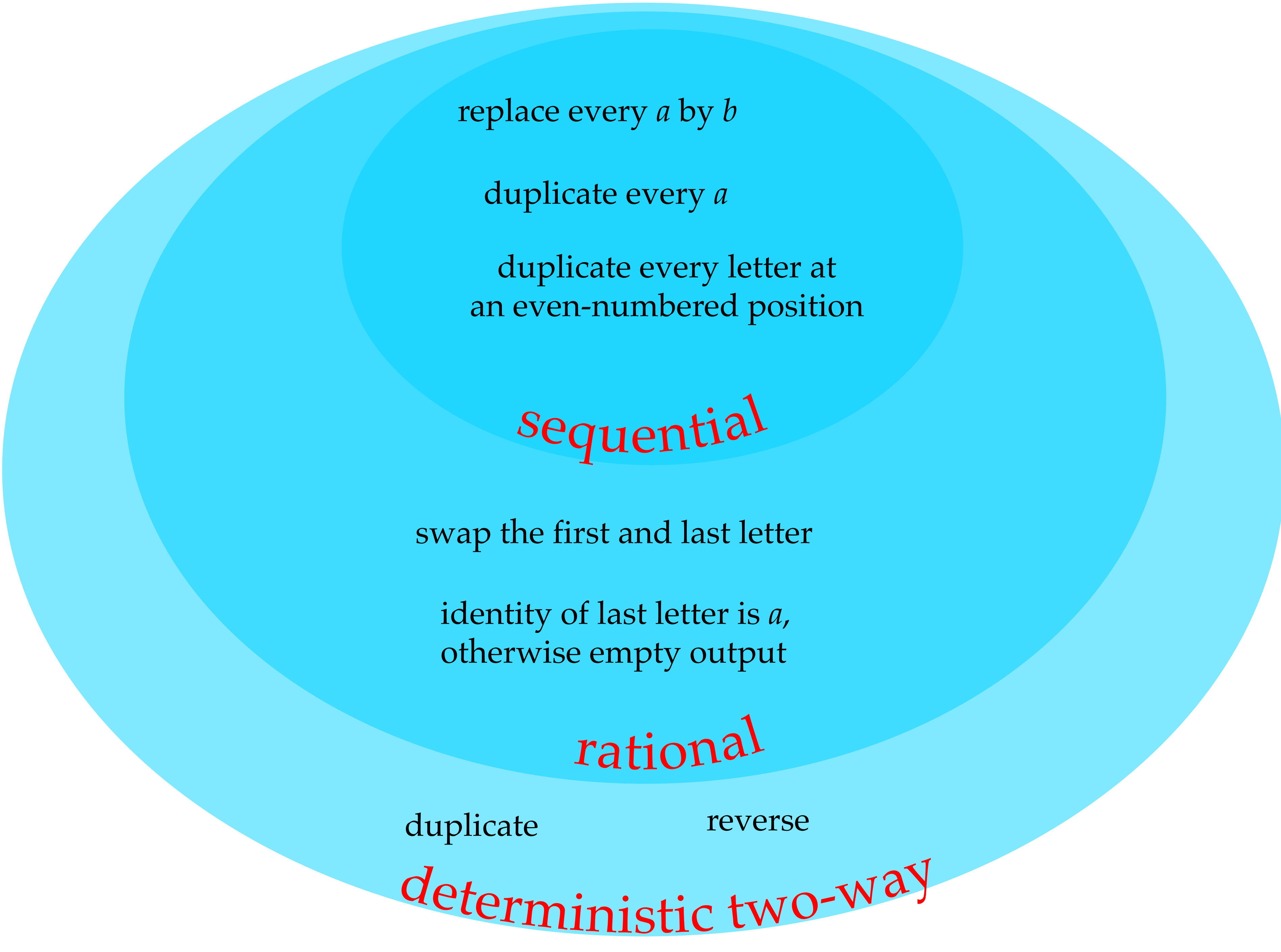
a/ε
b/ε
a/a
b/b
a/a
a/ε
b/ε
b/ε
a/a
b/ε
a/a
b/b

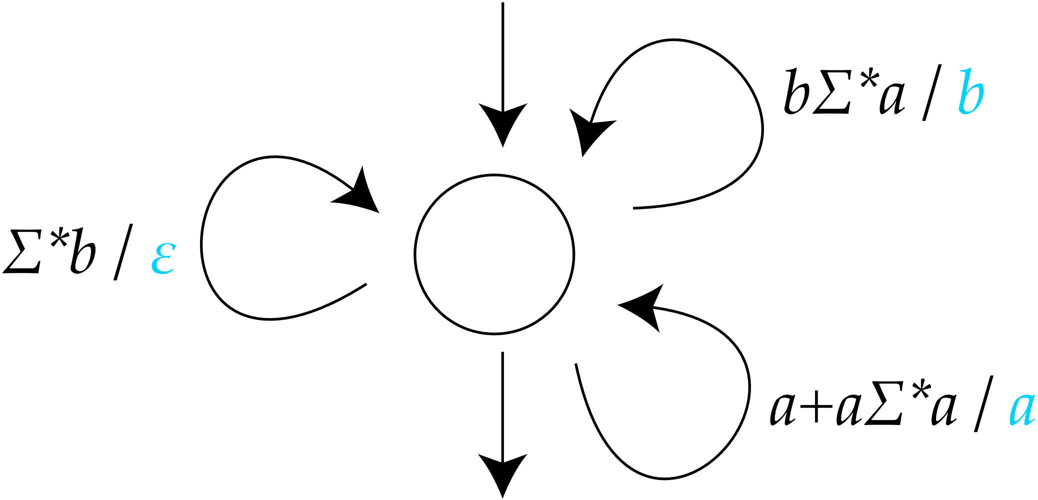


6
4
4
2
1 1 2 3 5
4 words
2 words
1 word
when reading an input letter with rank 2
6
2
7 words


6
5
4
3
1 7 2 1 5 1 1 2 3 5 2 1 5 7 2 1 3 1 5 2 3 3 1 5 2 1 5 1 1 1 5 2 1 4 1 1 1 5 6 1
1 5 2 1 5 1 1 2 3 5 2 1 5 2 2 2 3 1 5 2 2 2
1 5 2 1 5 1 1 1 4 2
1 3 2 1
when reading input letter 6
6
6
1 7 2 1 5 1 1 2 3 5 2 1 5 7 2 1 3 1 5 2 3 3 1 5 2 1 5 1 1 1 5 2 1 4 1 1 1 5 6 1
1 5 2 1 5 1 1 2 3 5 2 1 5 2 2 2 3 1 5 2 2 2 1 5 2 1 5 1 1 1 4 2 1 3 2 1 6


register 4
register 3
register 2
register 1
register 0
6
5
4
3
word associated to register 3
1 1 2 3 5
1 8 2 1 4 1 1 2 4 6 2 1 5 6 2 2 3 1 4 2 4 2 1 4 2 1 2 2 2 1 6 2 1 4 2 1 1 5 6 1
1 4 2 1 6 1 4 2 6 4 2 1 4 2 2 2 4 1 5 2 2 2
1 4 2 1 5 1 1 1 4 2
1 3 2 1
tail
head
7 words with even max that is ≤ 6
max 5
2³-1
a prefix of the input that is not in any head or tail
(we only write ranks of input letters)
smallest value of more significant nonempty registers
value of the register


replace every *a* by *b*
duplicate every *a*
duplicate every letter at an even-numbered position
sequential
swap the first and last letter
identity of last letter is *a*, otherwise empty output
rational
duplicate
reverse
deterministic two-way

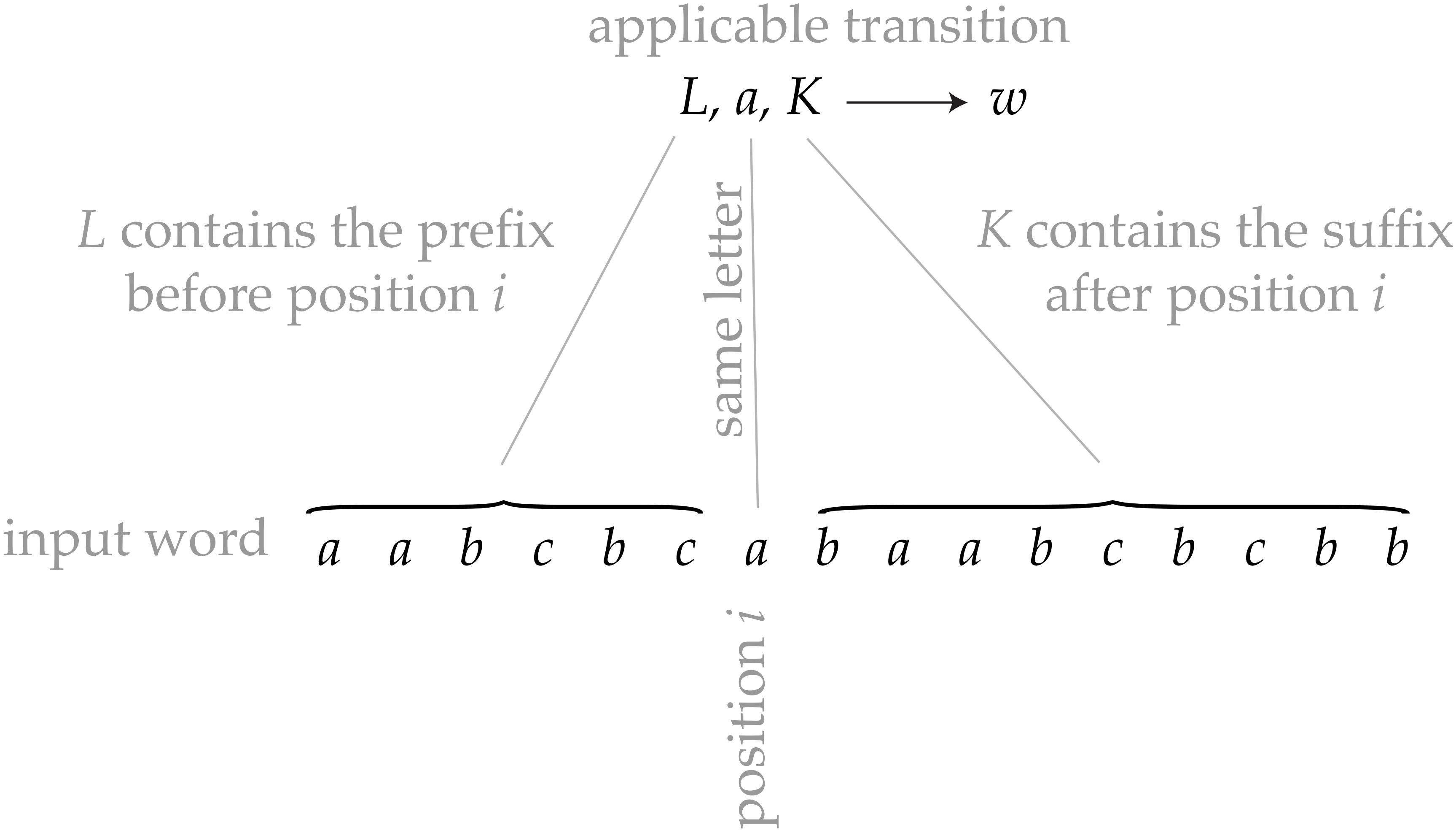

$b\Sigma^*a$ / b
Σ^*b / ε
$a+a\Sigma^*a$ / a

letter from
input alphabet

possibly empty word
over output alphabet

$$L, a, K \longrightarrow w$$

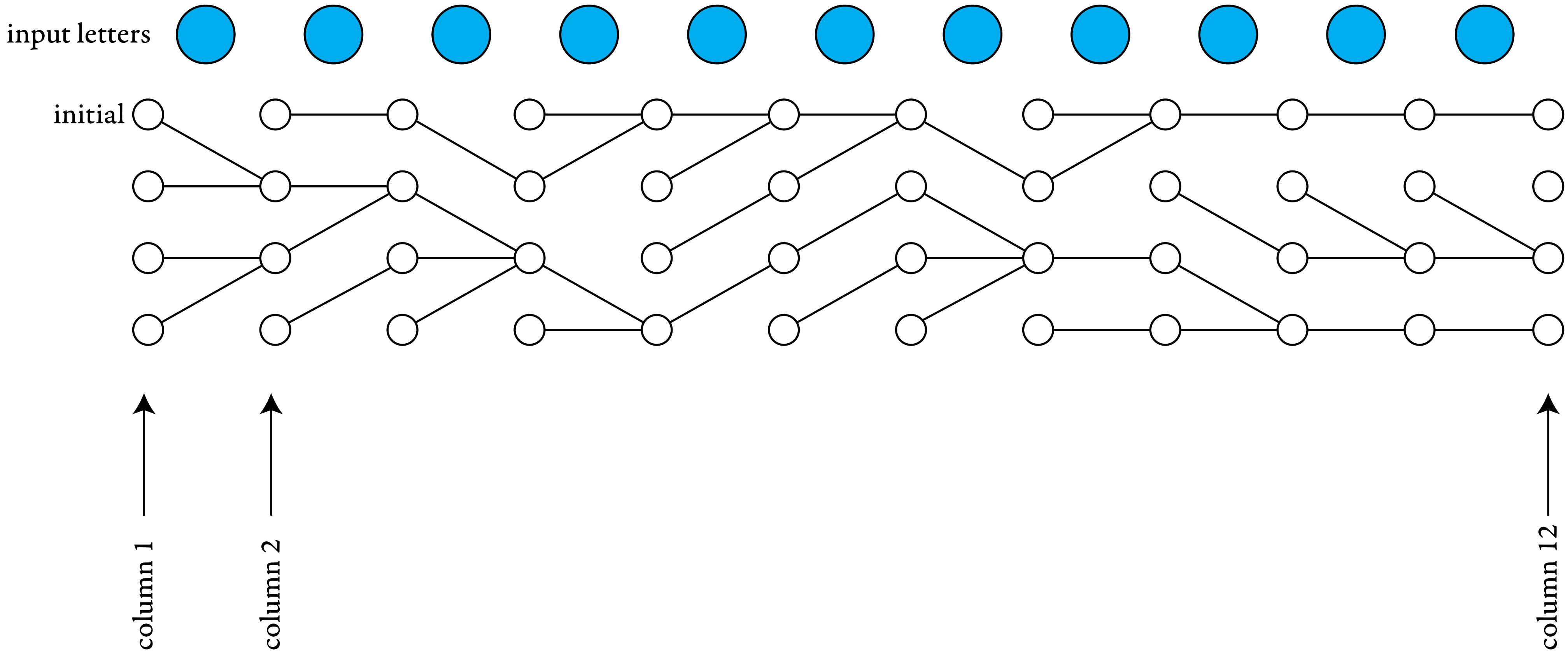
regular languages
over input alphabet

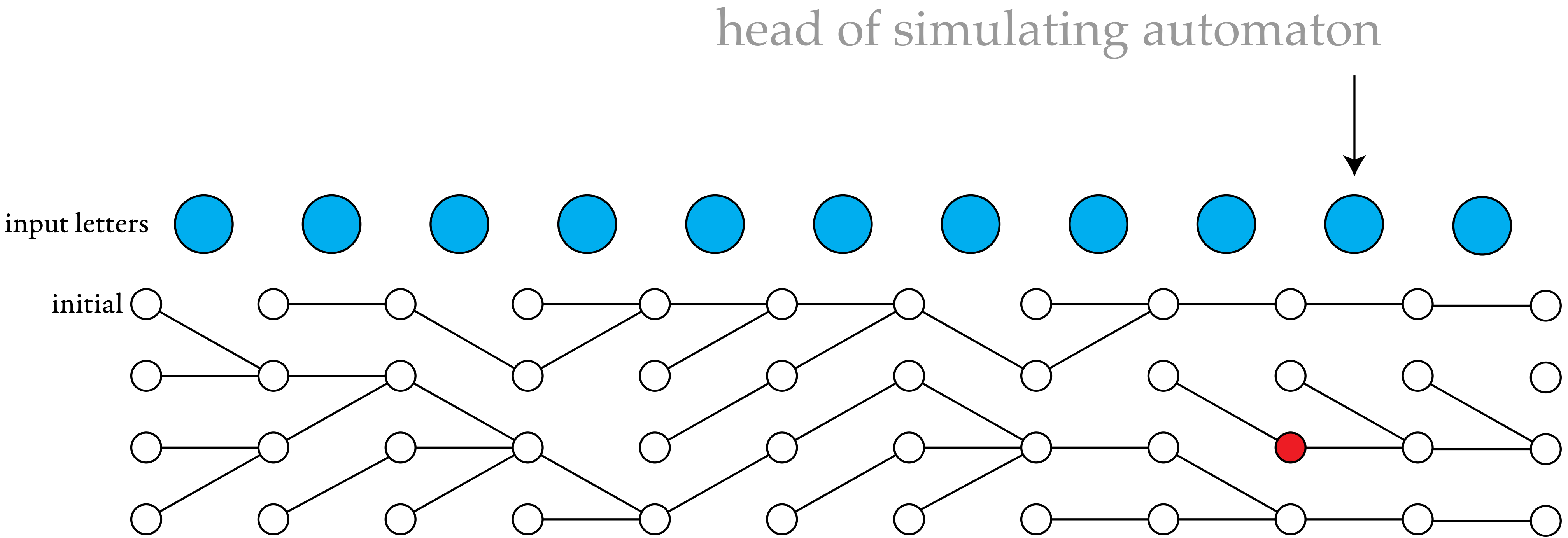

applicable transition
$L, a, K \longrightarrow w$
L contains the prefix before position i
same letter
K contains the suffix after position i
input word
$a\ a\ b\ c\ b\ c\ a\ b\ a\ a\ b\ c\ b\ c\ b\ b$
position i

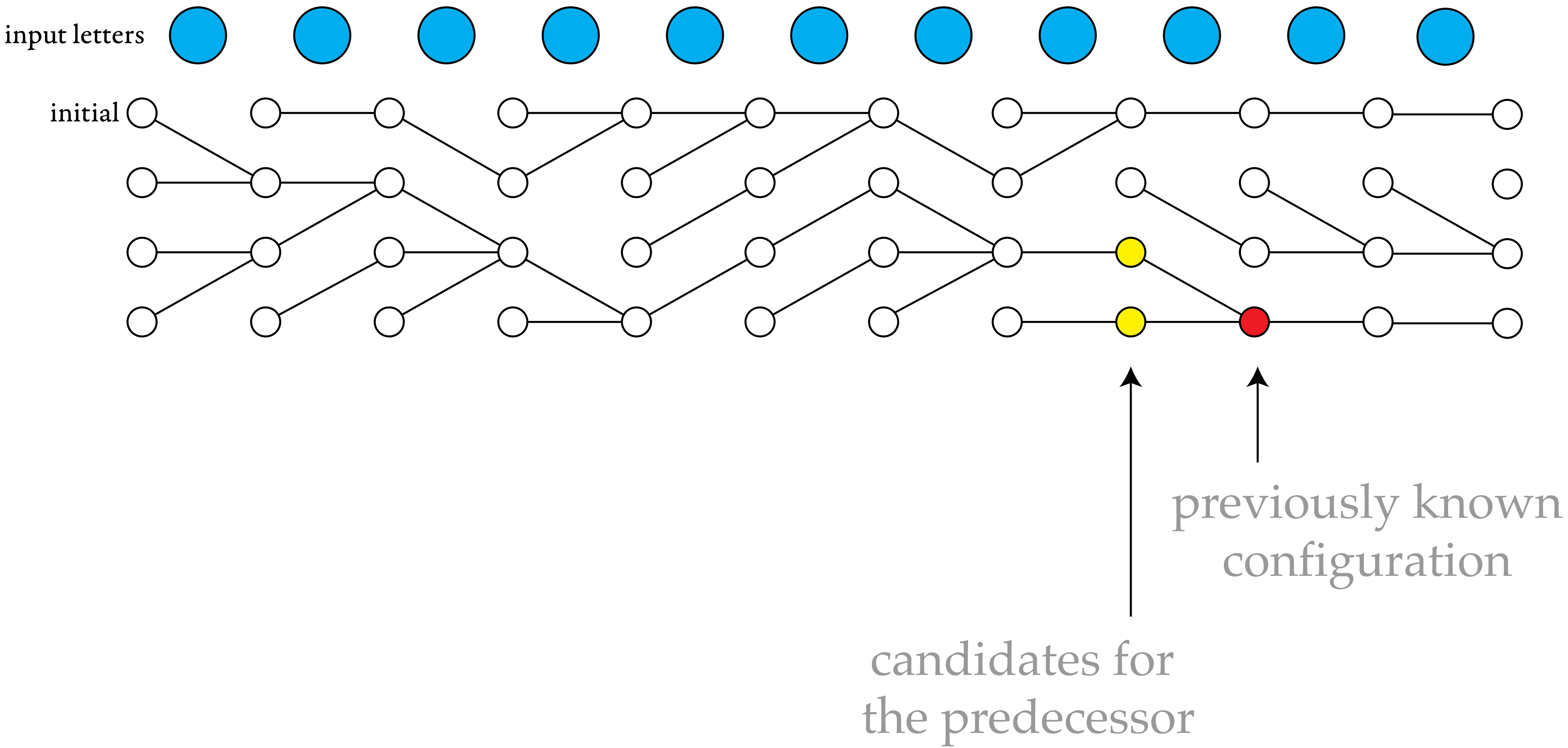
$$L,\ a,\ K \longrightarrow w$$

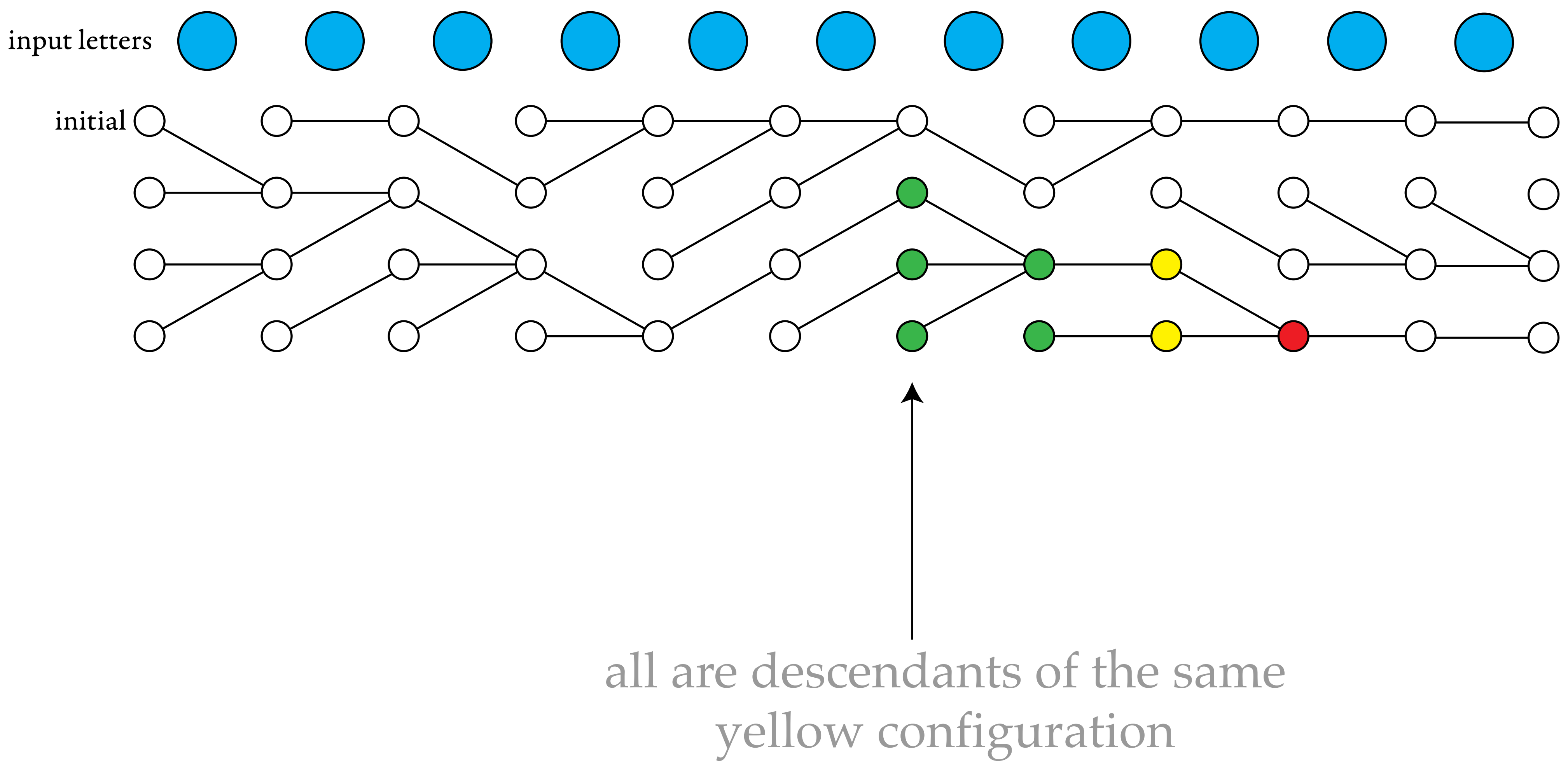
words that admit a run from some initial state to p

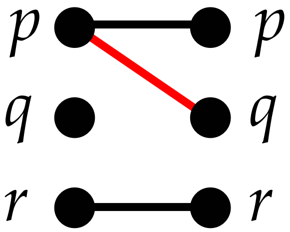
words that admit a run from q to some accepting state


input letters
initial
column 1
column 2
column 12

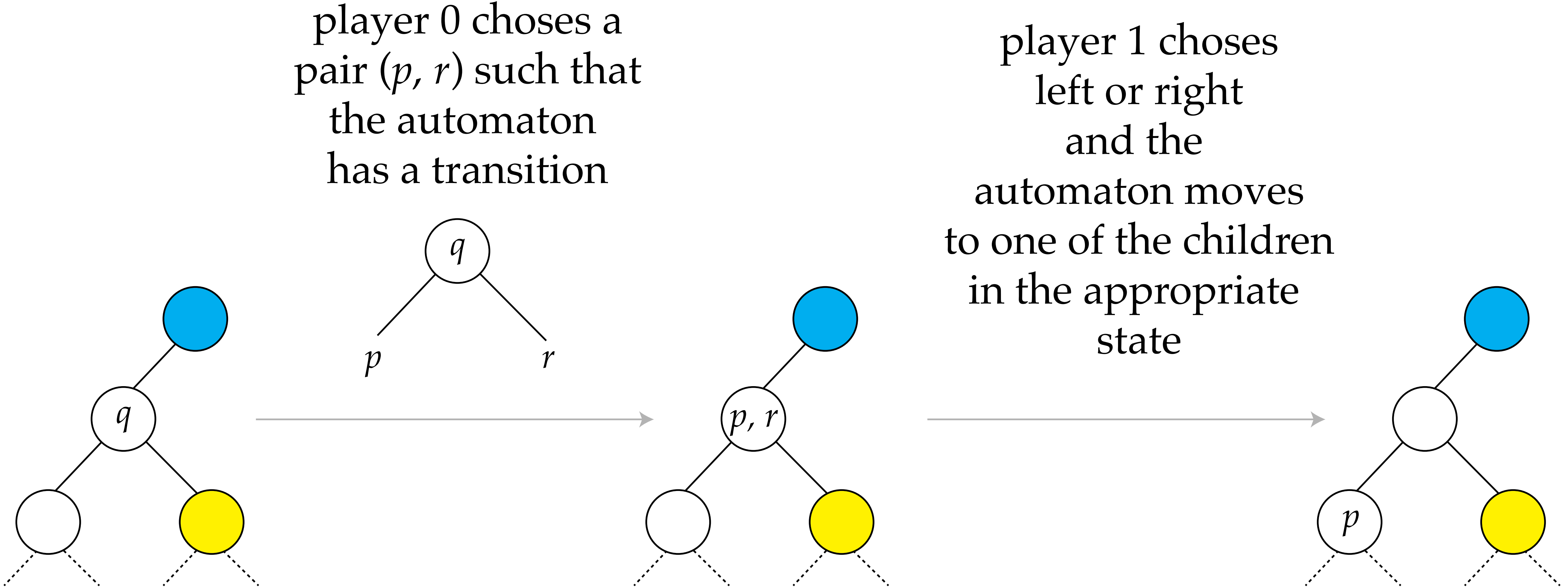

head of simulating automaton
input letters
initial

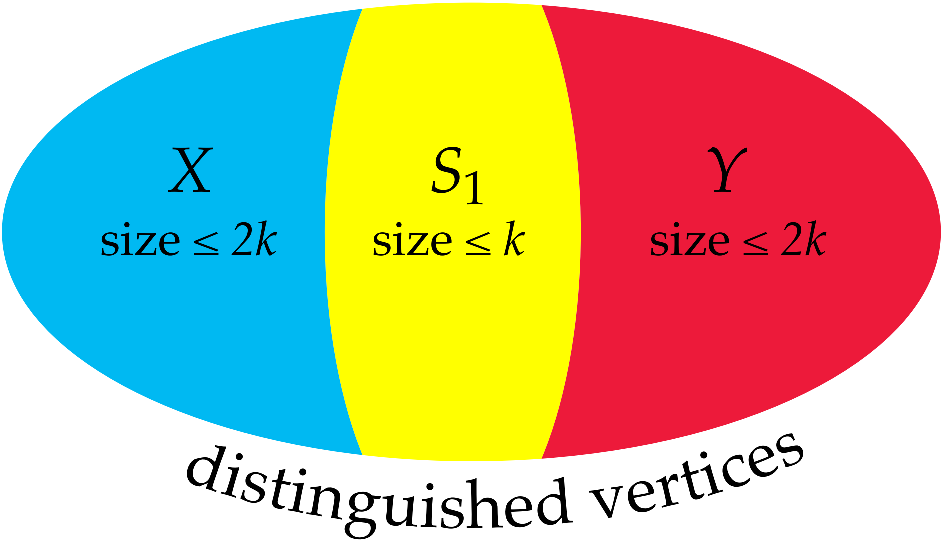

input letters
initial
previously known configuration
candidates for the predecessor

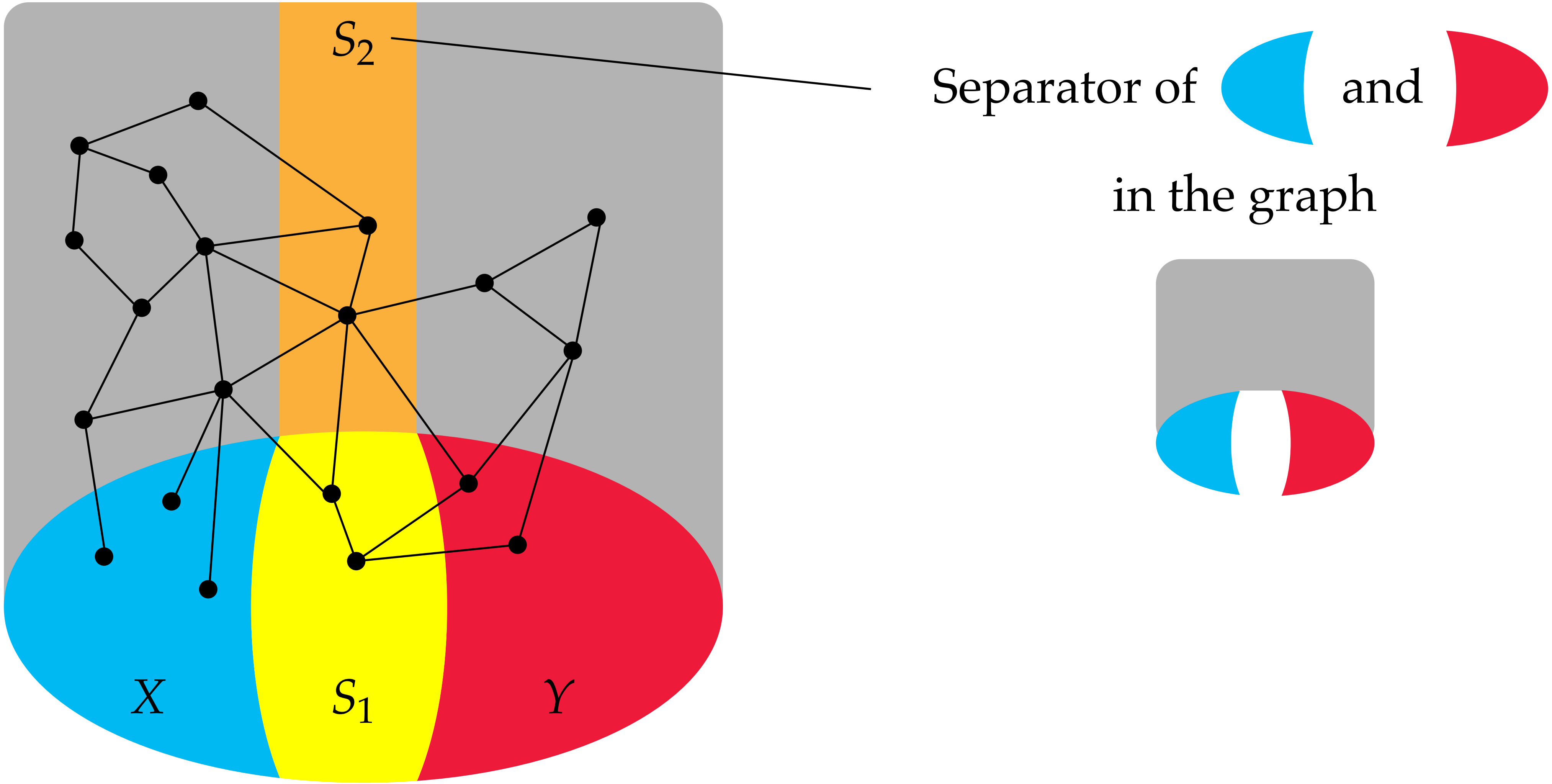

input letters
initial
all are descendants of the same yellow configuration



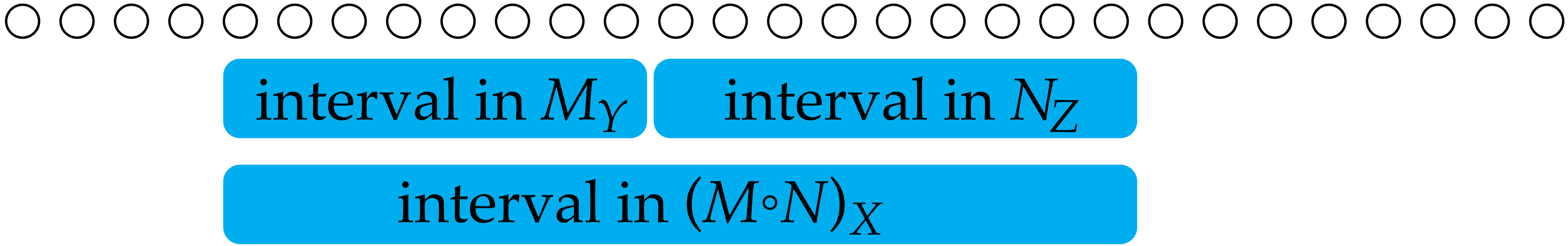
every vertex in the right column
has at most one incoming edge

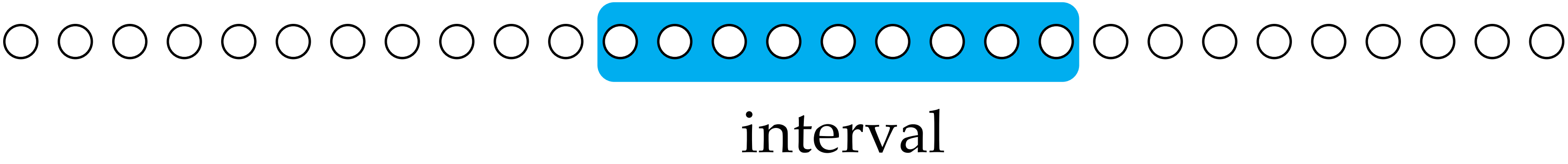

player 0 choses a pair (p, r) such that the automaton has a transition
q
p
r
player 1 choses left or right and the automaton moves to one of the children in the appropriate state
q
p, r
p

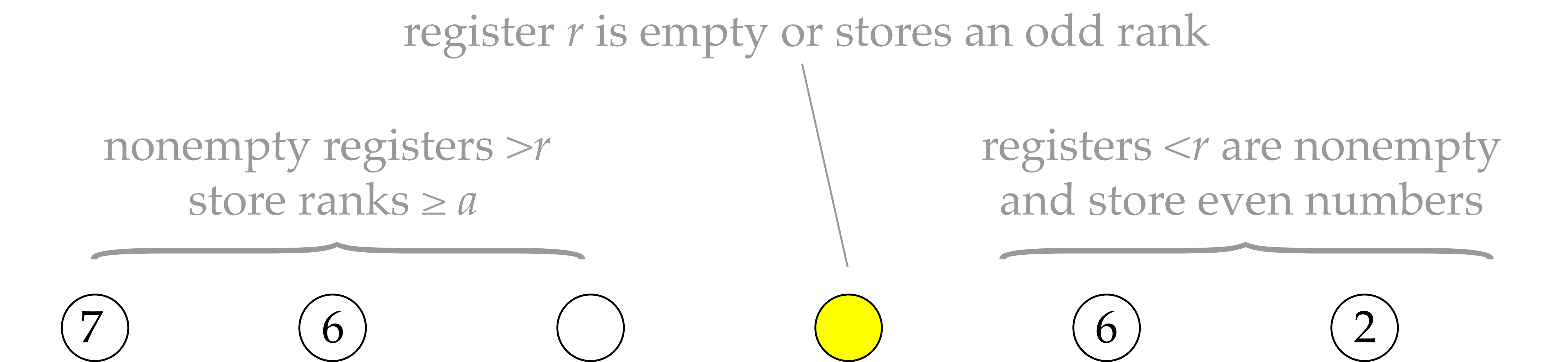

X
size $\leq 2k$
S_1
size $\leq k$
Y
size $\leq 2k$
distinguished vertices

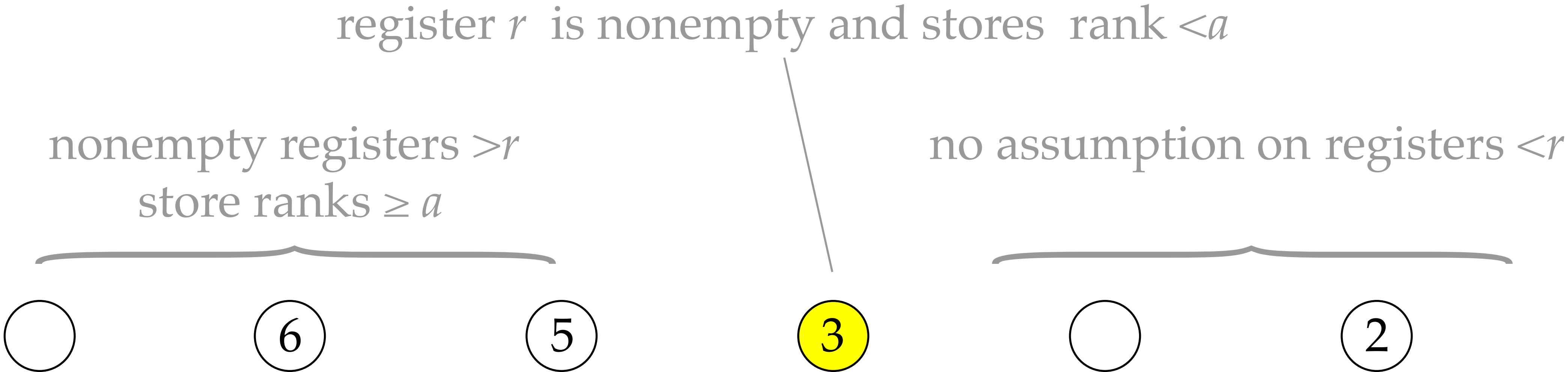


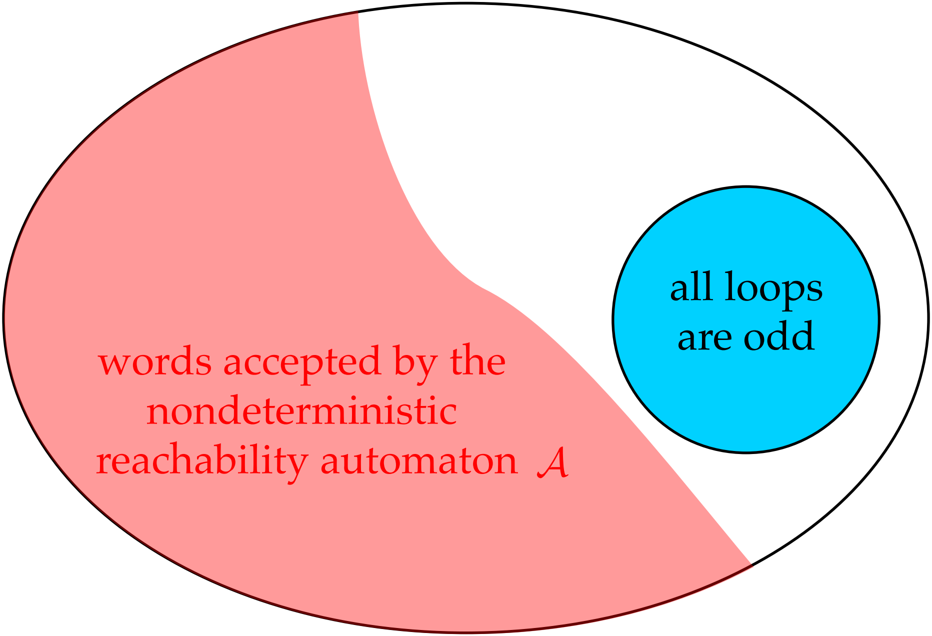
S2
X
S1
Y
Separator of
and
in the graph

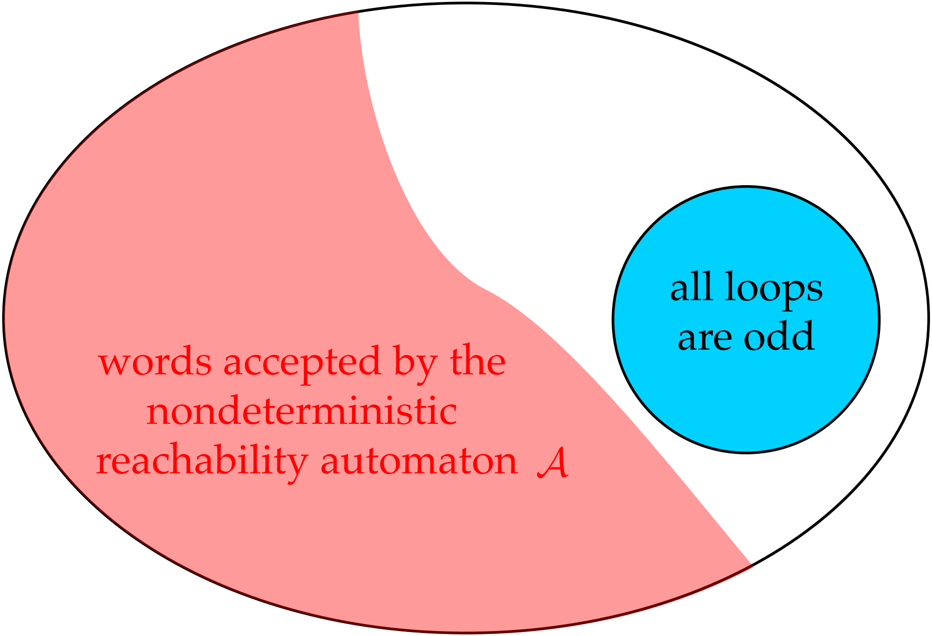

interval in M_Y
interval in N_Z
interval in $(M \circ N)_X$


interval

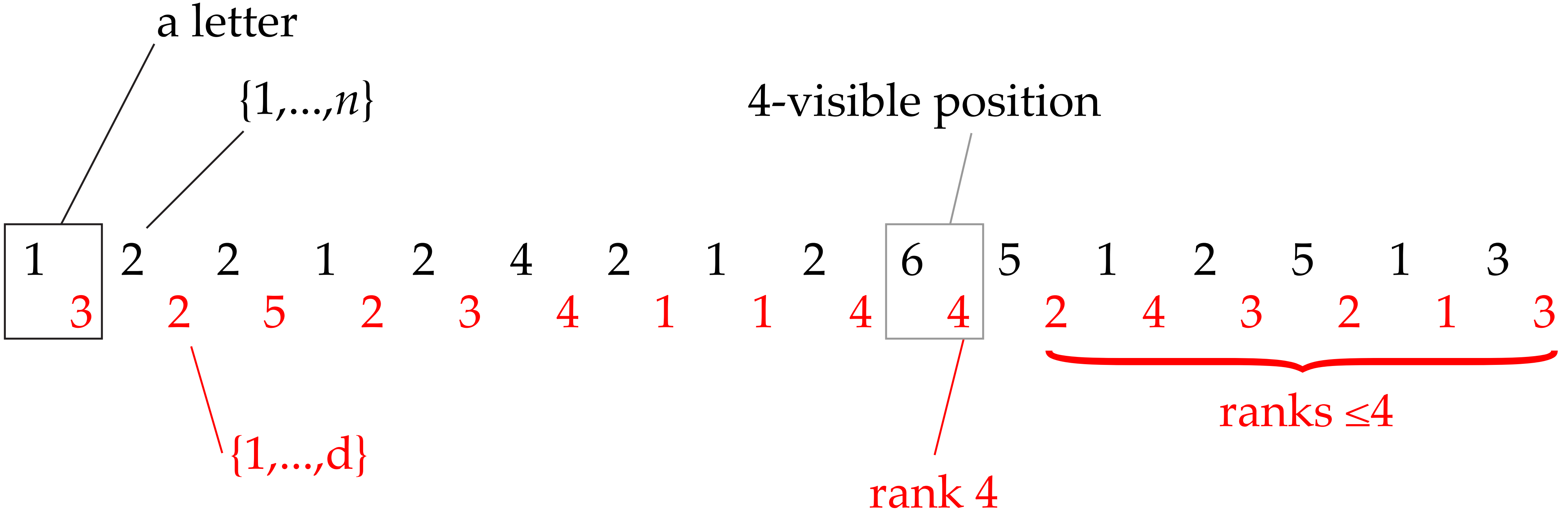

register r is empty or stores an odd rank
nonempty registers $>r$ store ranks $\geq a$
registers $<r$ are nonempty and store even numbers
7
6
6
2

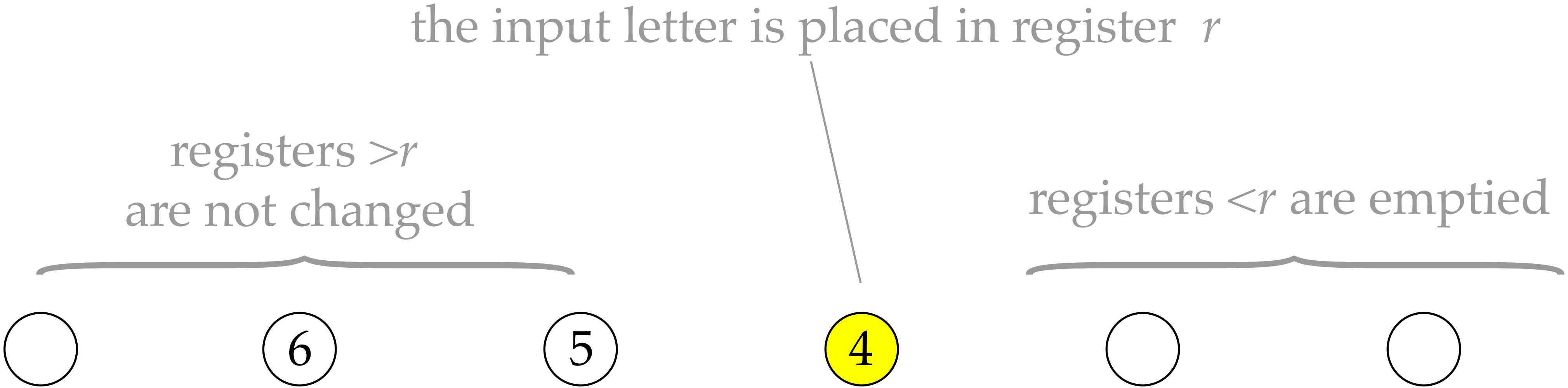

register r is nonempty and stores rank <a
nonempty registers >r
store ranks ≥ a
no assumption on registers <r
6
5
3
2

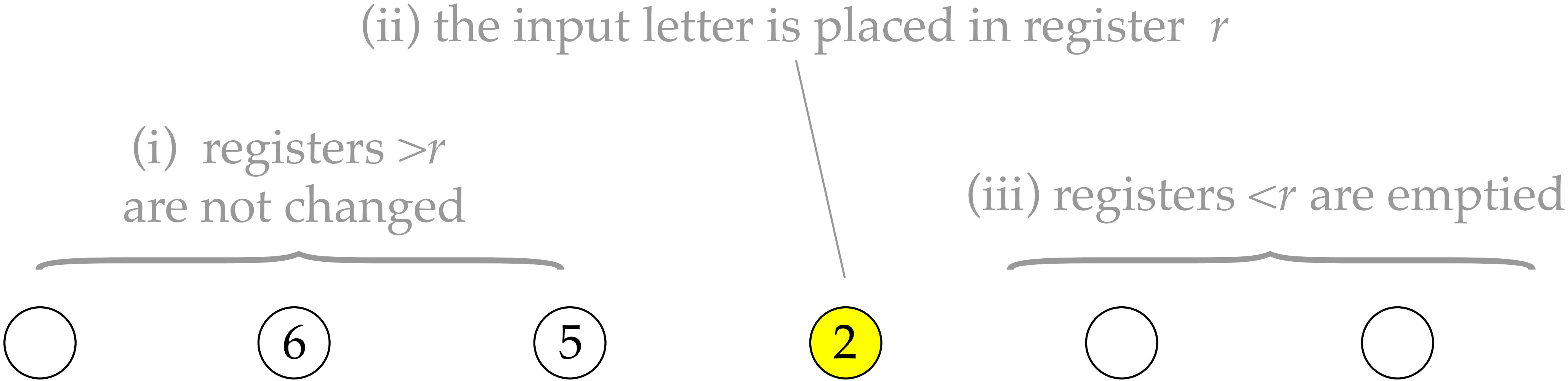

words accepted by the nondeterministic reachability automaton $\mathcal{A}$
all loops are odd

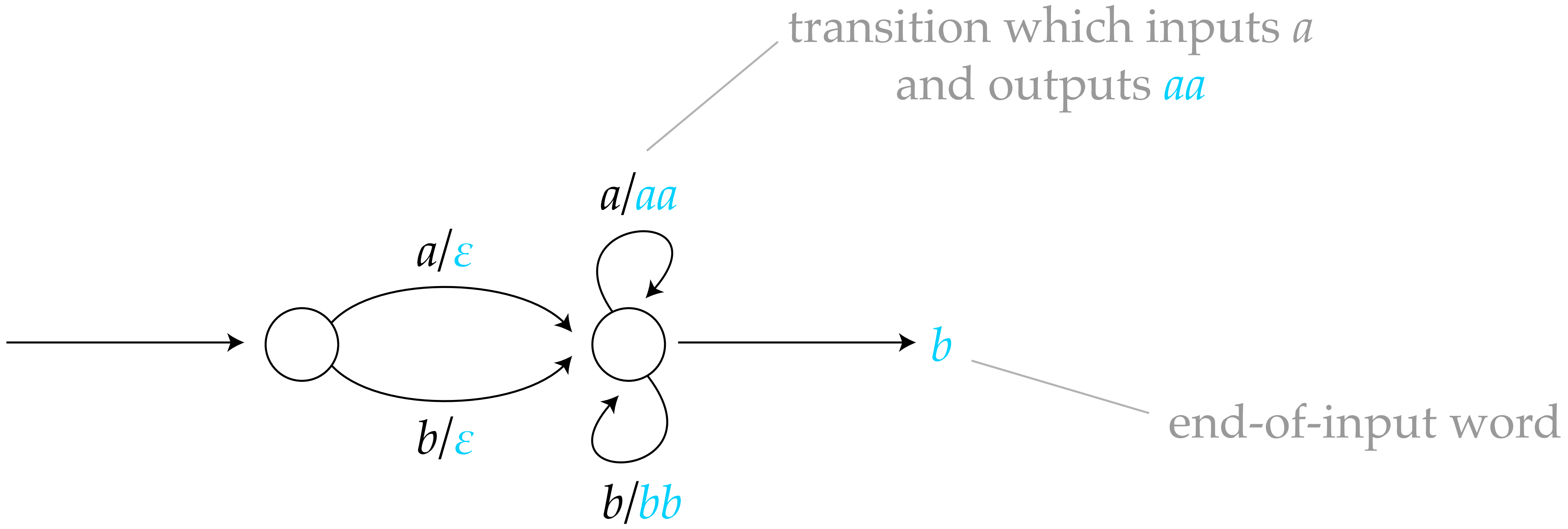


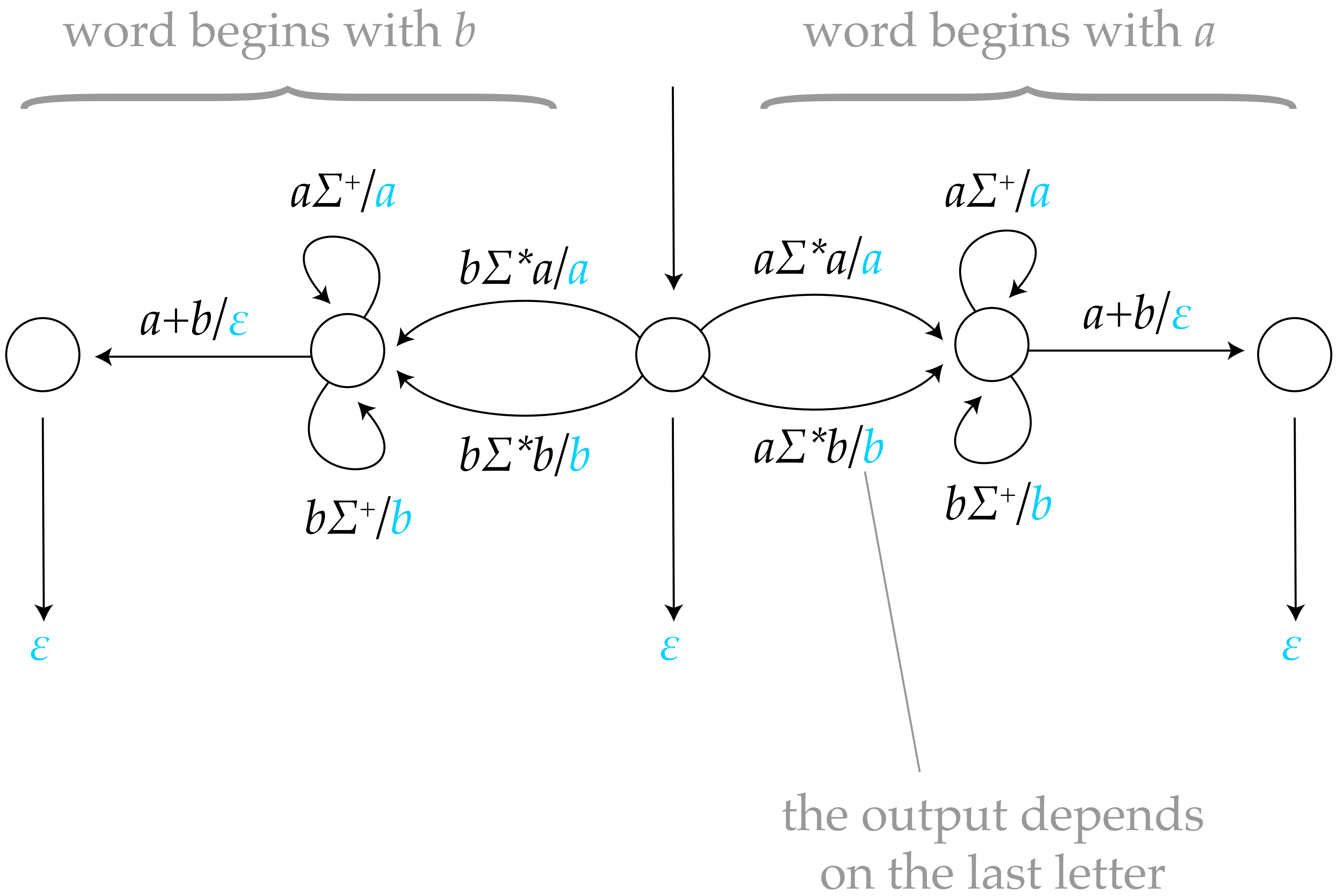
all loops
are odd
words accepted by the
nondeterministic
reachability automaton $\mathcal{A}$


a letter
{1,...,n}
4-visible position
1 2 2 1 2 4 2 1 2 6 5 1 2 5 1 3
3 2 5 2 3 4 1 1 4 4 2 4 3 2 1 3
ranks ≤4
{1,...,d}
rank 4


the input letter is placed in register r
registers >r
are not changed
registers <r are emptied
6
5
4


(ii) the input letter is placed in register r
(i) registers $>r$ are not changed
(iii) registers $<r$ are emptied
6
5
2


transition which inputs a
and outputs aa
a/aa
a/ε
b/ε
b
end-of-input word
b/bb


word begins with b
word begins with a
$a\Sigma^+/a$
$b\Sigma^*a/a$
$a\Sigma^*a/a$
$a\Sigma^+/a$
$a+b/\varepsilon$
$a+b/\varepsilon$
$b\Sigma^*b/b$
$a\Sigma^*b/b$
$b\Sigma^+/b$
$b\Sigma^+/b$
ε
ε
ε
the output depends
on the last letter