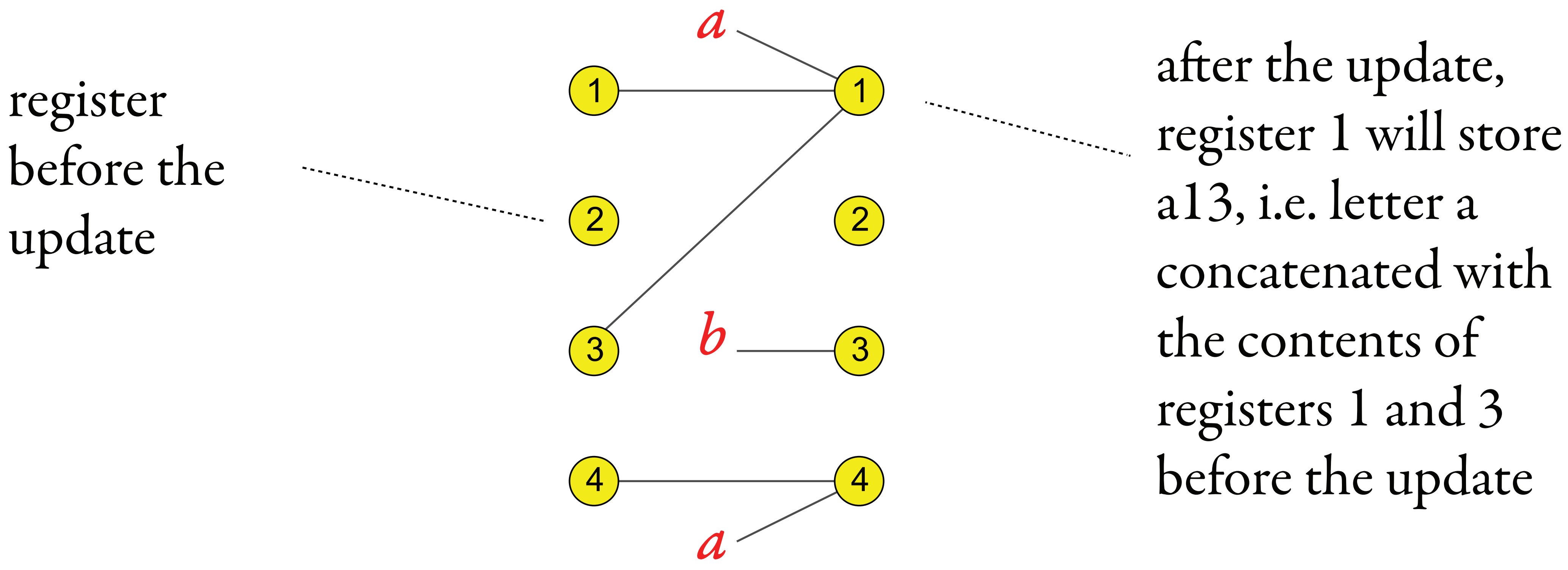
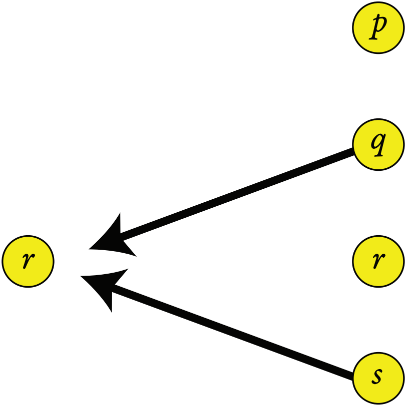
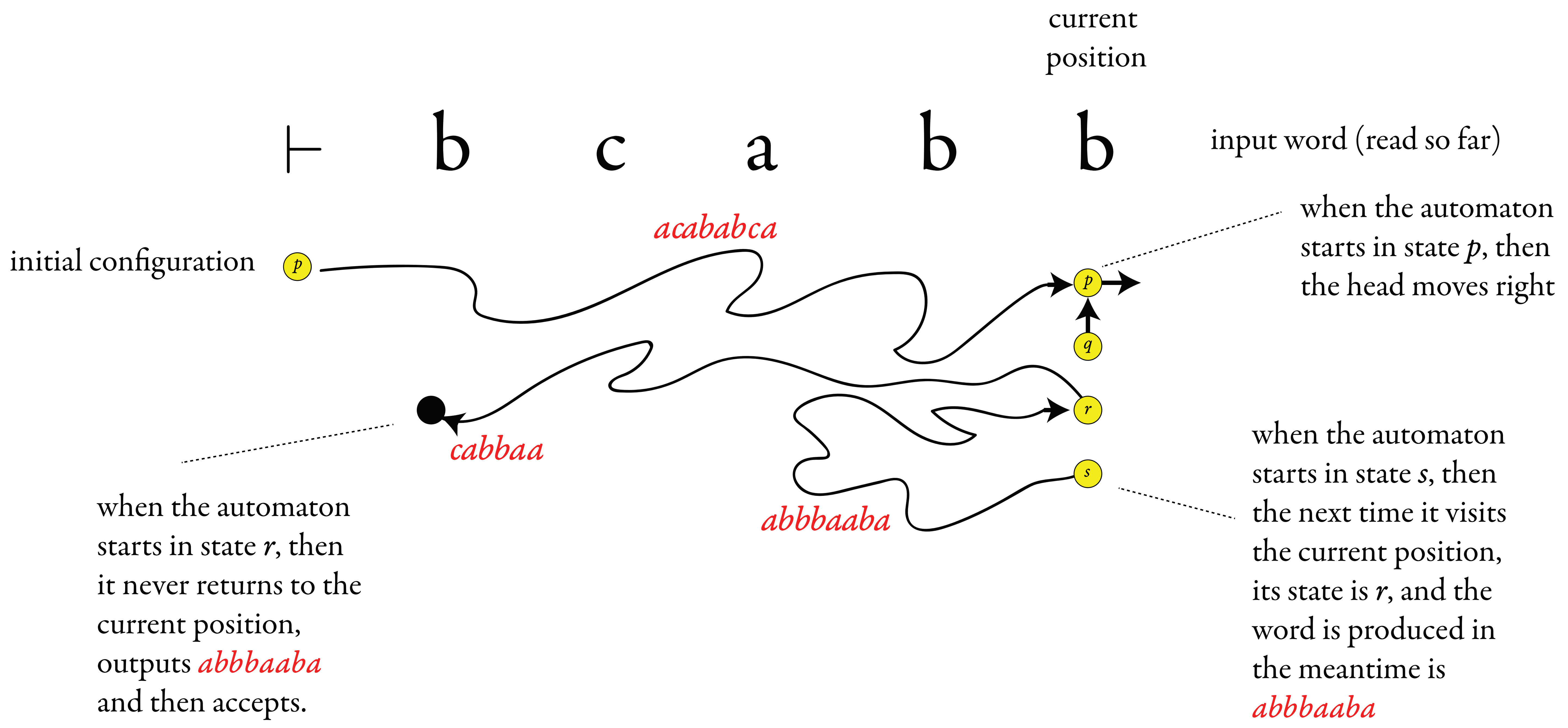


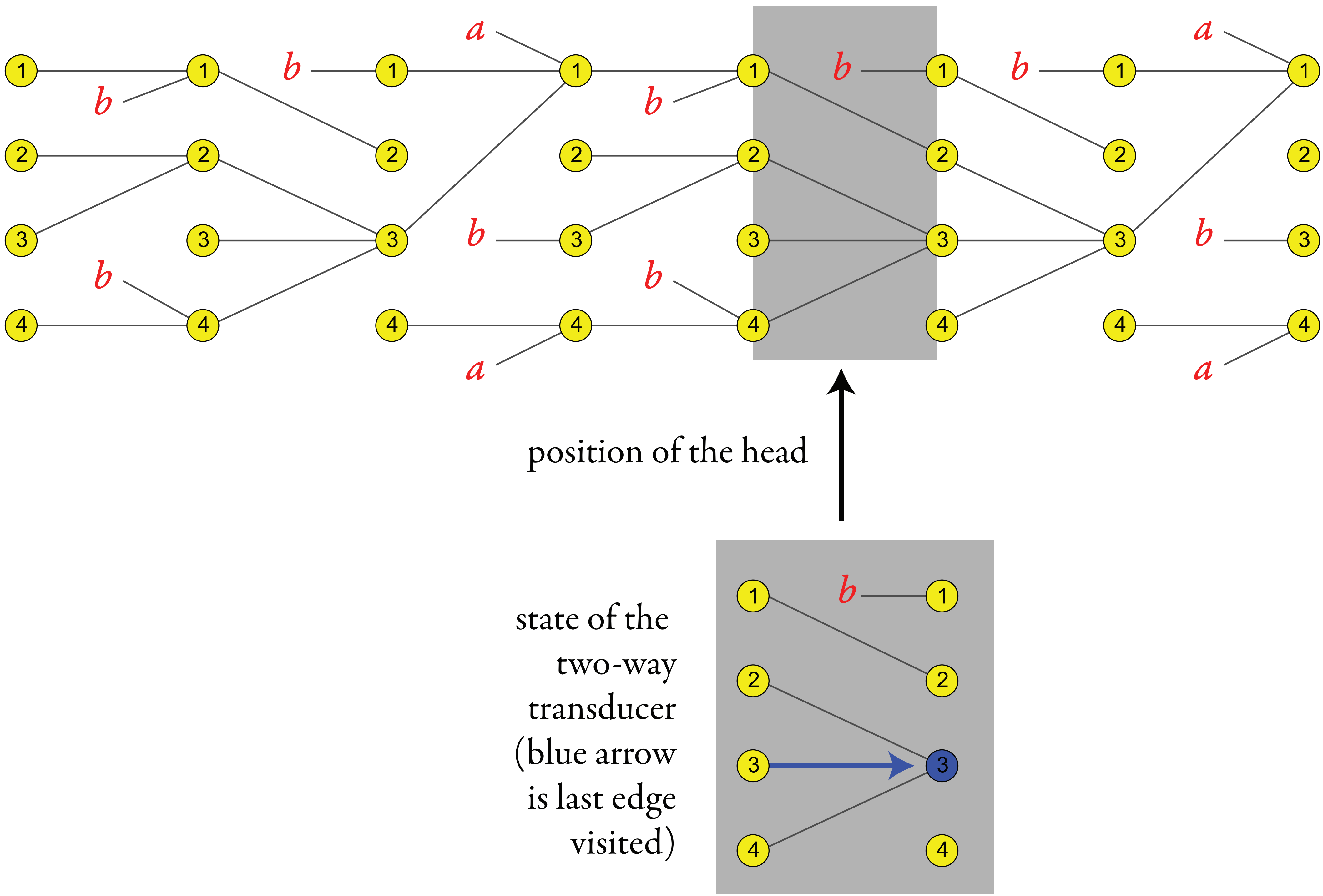
register before the update
a
1
2
3
4
b
a
1
2
3
4
after the update, register 1 will store a13, i.e. letter a concatenated with the contents of registers 1 and 3 before the update

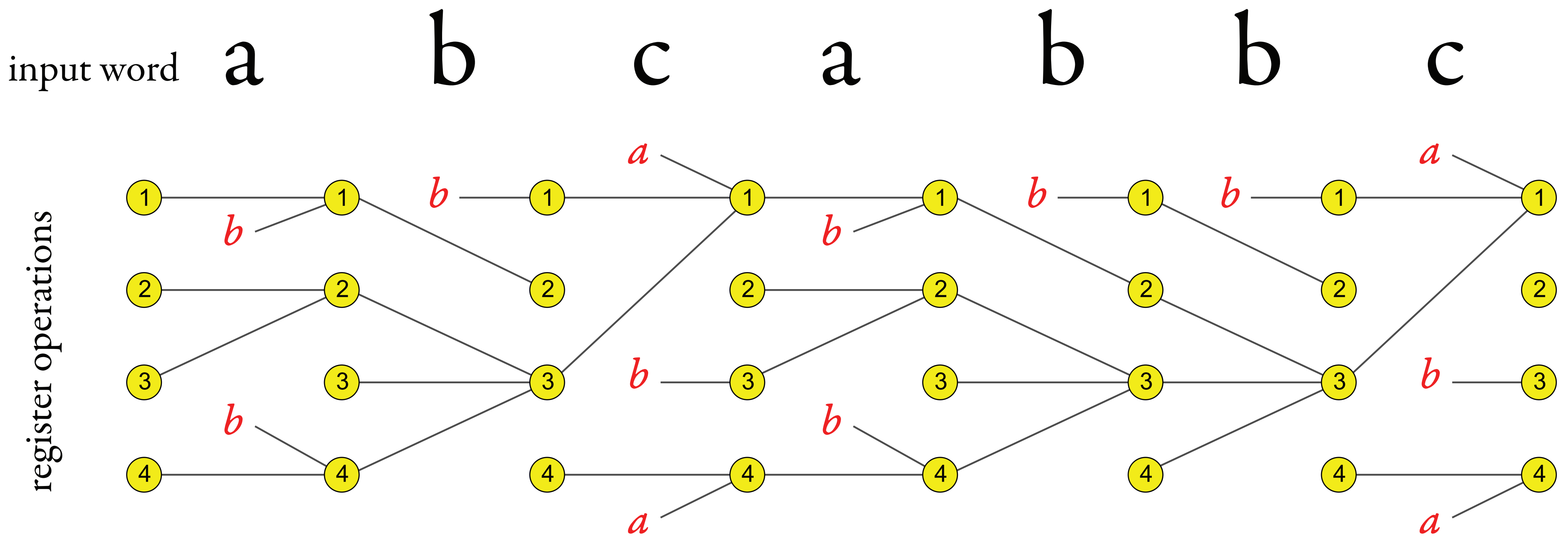
a

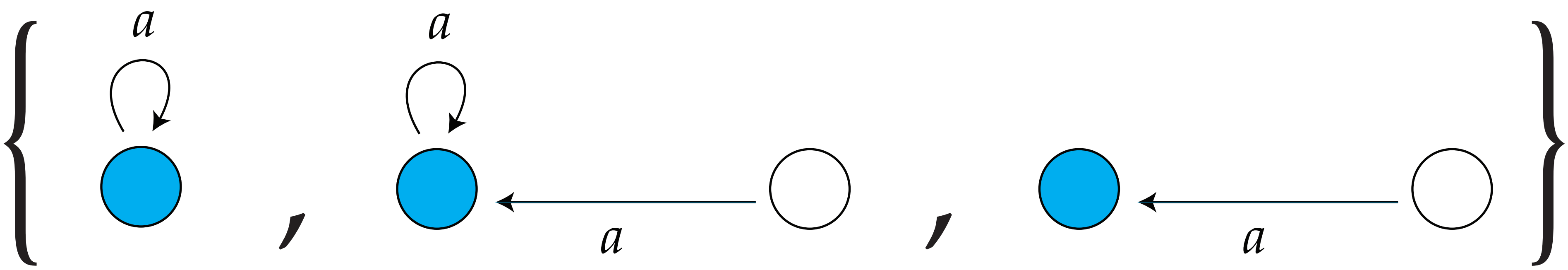




current position
⊢ b c a b b
input word (read so far)
initial configuration
p
acababca
p
q
r
s
when the automaton starts in state p, then the head moves right
cabbaa
abbbaaba
when the automaton starts in state r, then it never returns to the current position, outputs abbbaaba and then accepts.
when the automaton starts in state s, then the next time it visits the current position, its state is r, and the word is produced in the meantime is abbbaaba

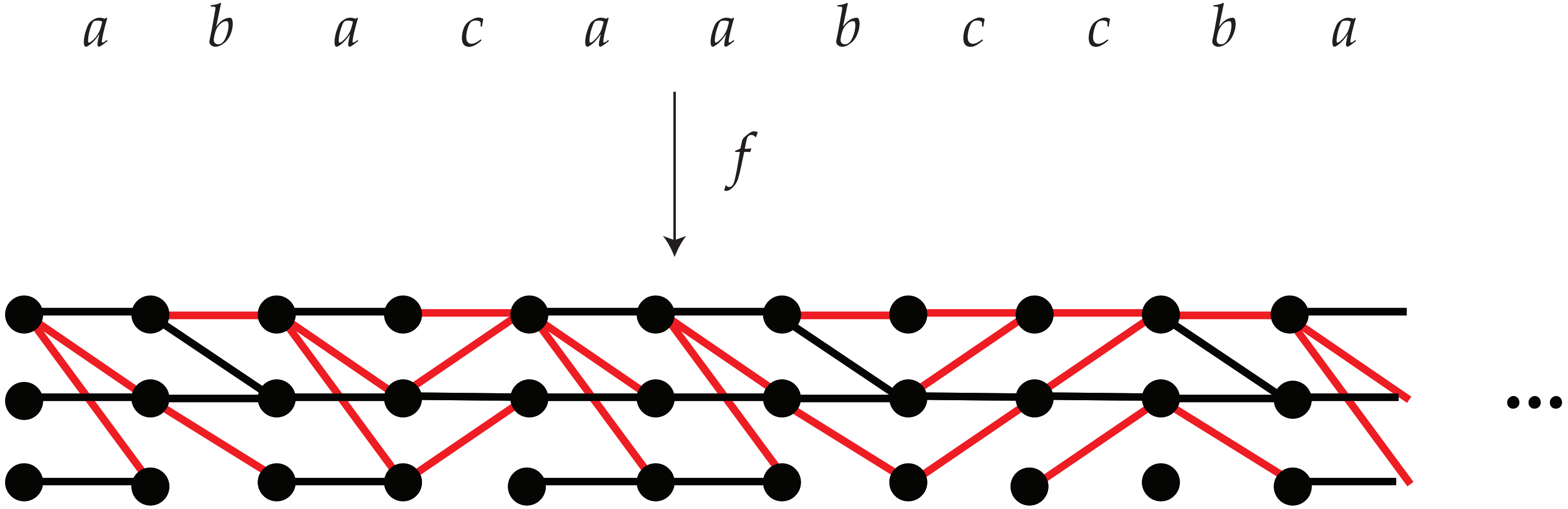

position of the head
state of the two-way transducer (blue arrow is last edge visited)

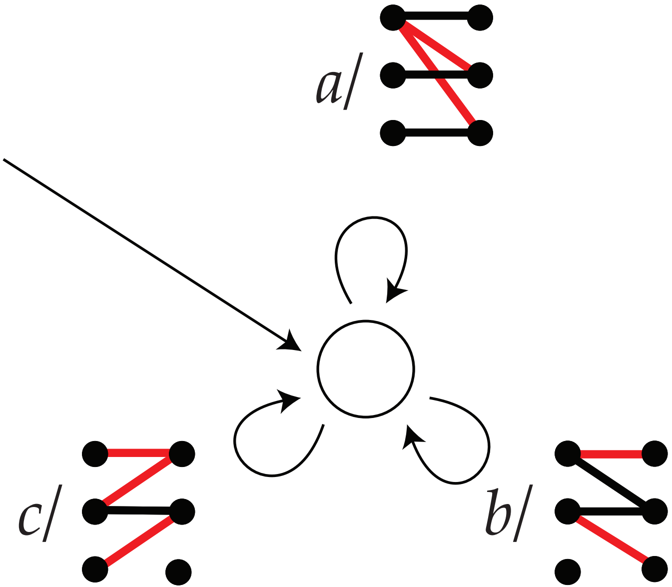

input word
a b c a b b c
register operations
1 2 3 4
a b

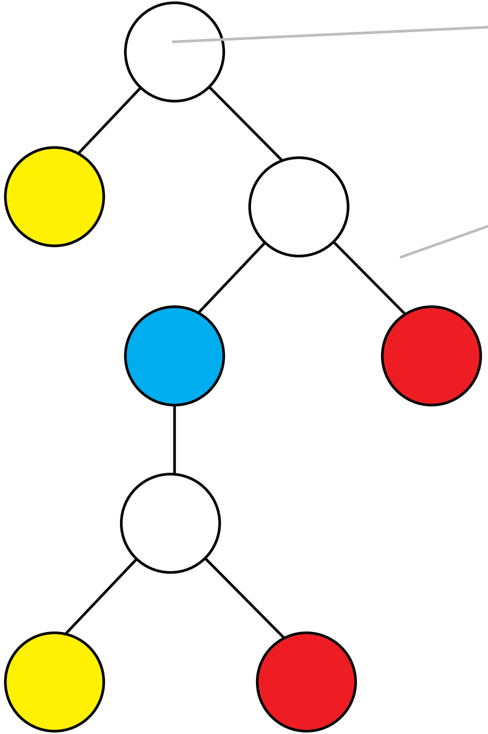


a set of transitions
is visualised as the
part of the automaton
that only uses transitions
from that set

it is impossible to see
this particular set
of transitions (and no
others) infinitely often


a b a c a a b c c b a
f
...

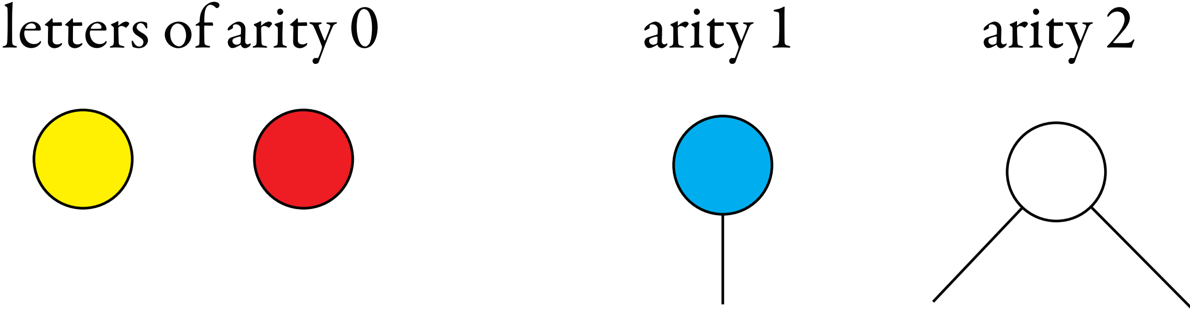

a/
b/
c/

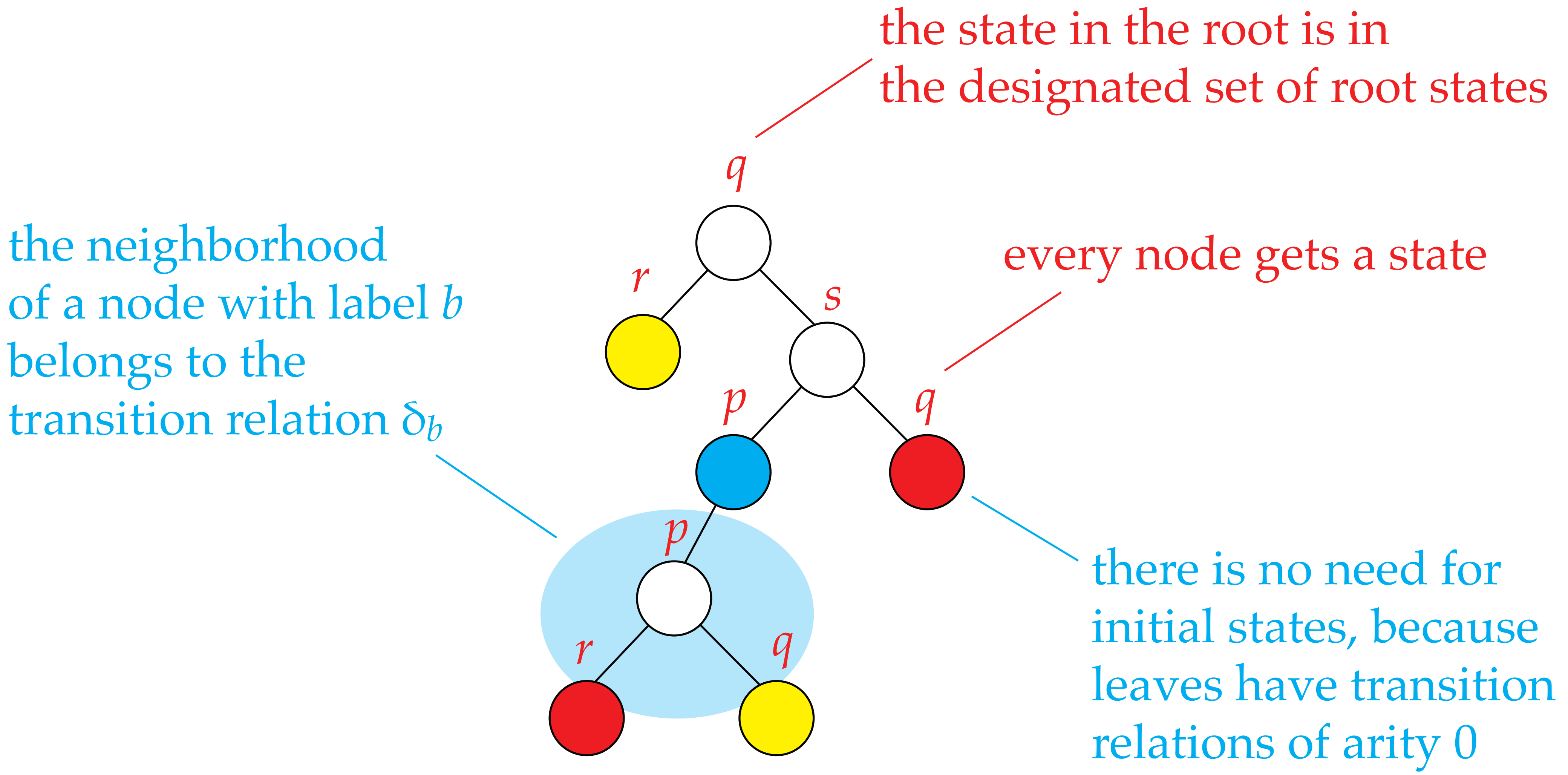


every node gets a label from the alphabet

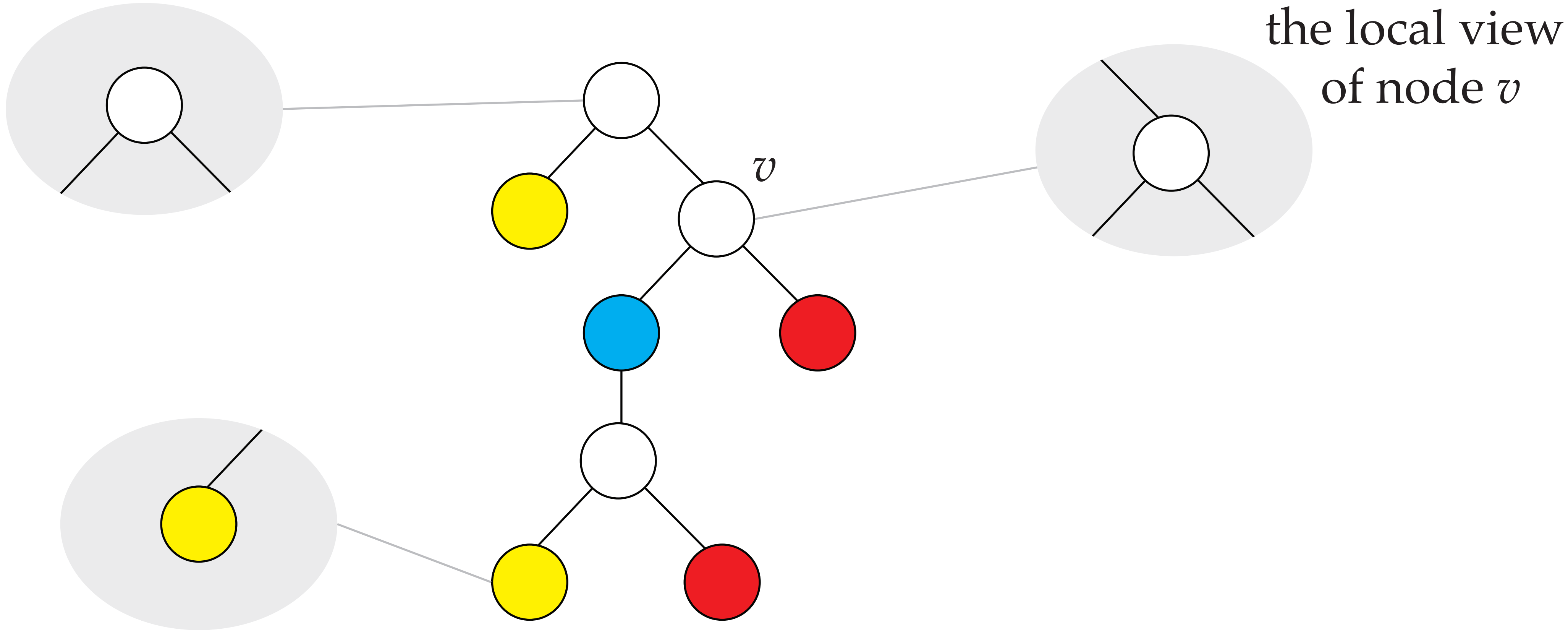
if a node has a label of arity *n*,
then it has exactly *n* children

children are ordered, so one can
speak of the first child, second child, etc.

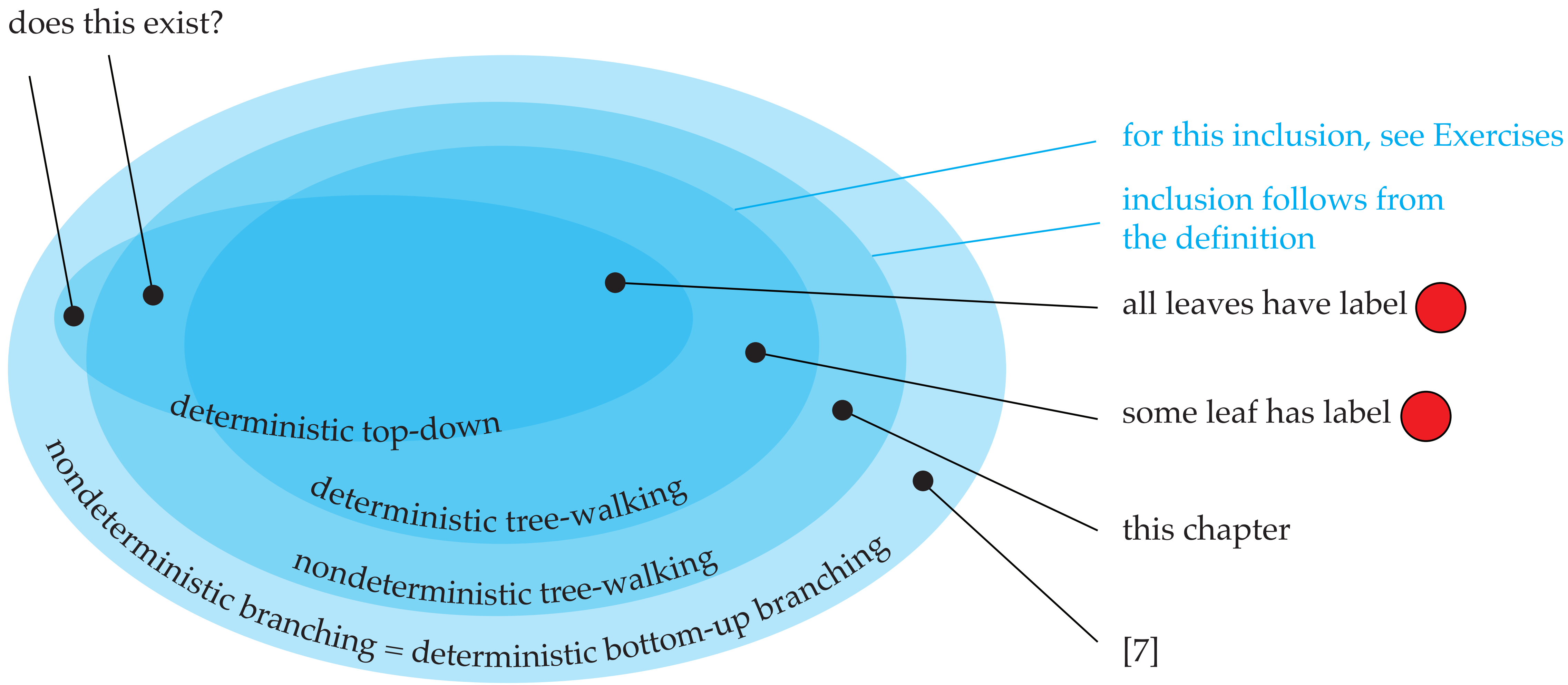

letters of arity 0
arity 1
arity 2

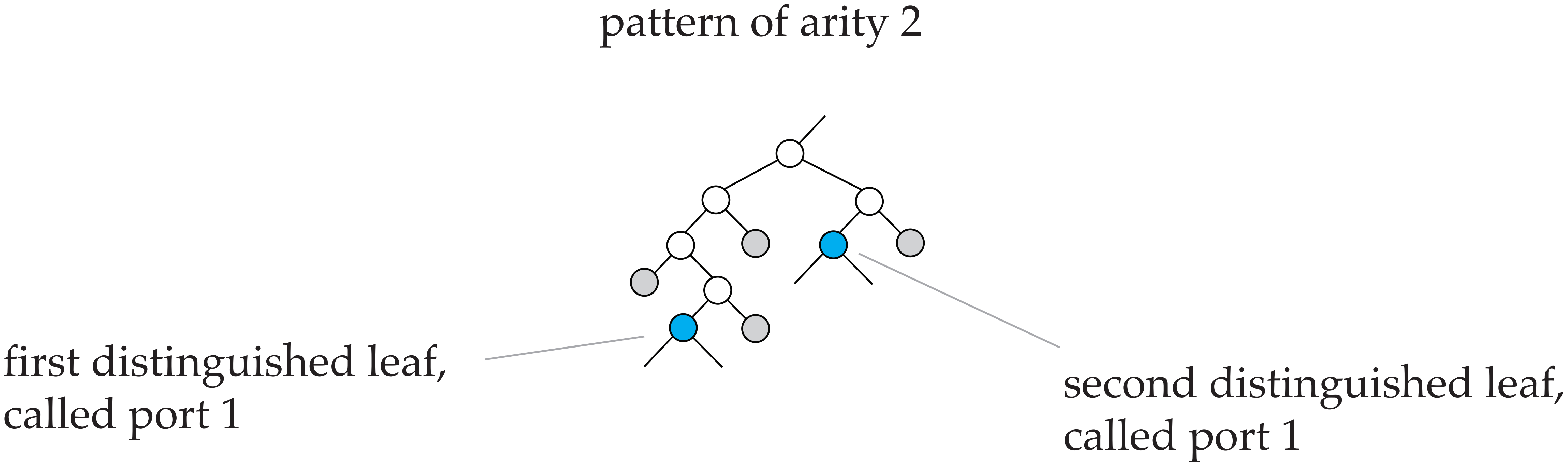


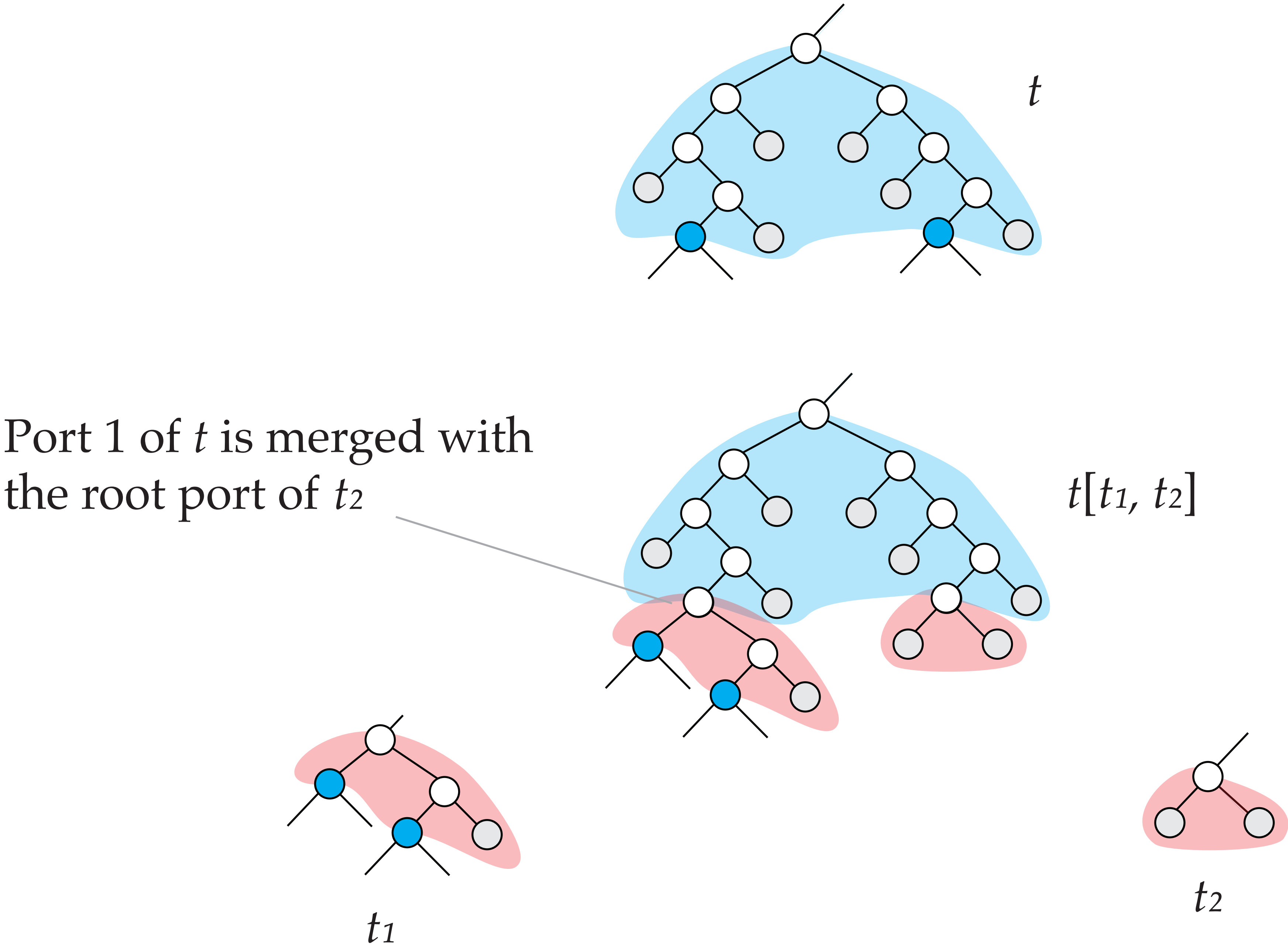
the state in the root is in
the designated set of root states
q
r
s
every node gets a state
the neighborhood
of a node with label b
belongs to the
transition relation δ_b
p
q
p
r
q
there is no need for
initial states, because
leaves have transition
relations of arity 0

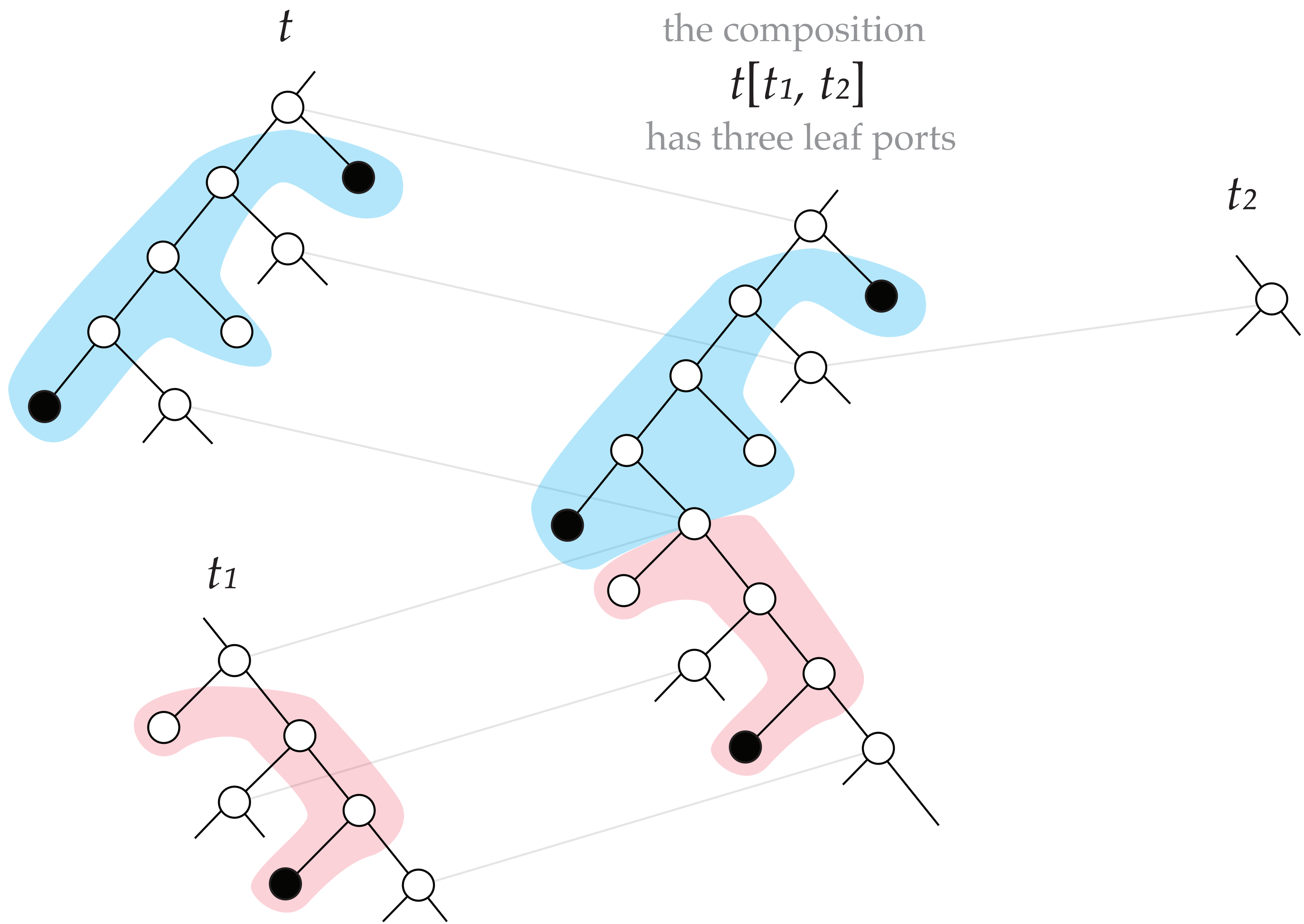


the local view
of node v
v

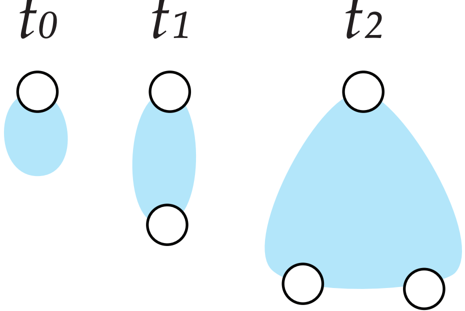

does this exist?
for this inclusion, see Exercises
inclusion follows from the definition
all leaves have label
some leaf has label
this chapter
[7]
deterministic top-down
deterministic tree-walking
nondeterministic tree-walking
nondeterministic branching = deterministic bottom-up branching

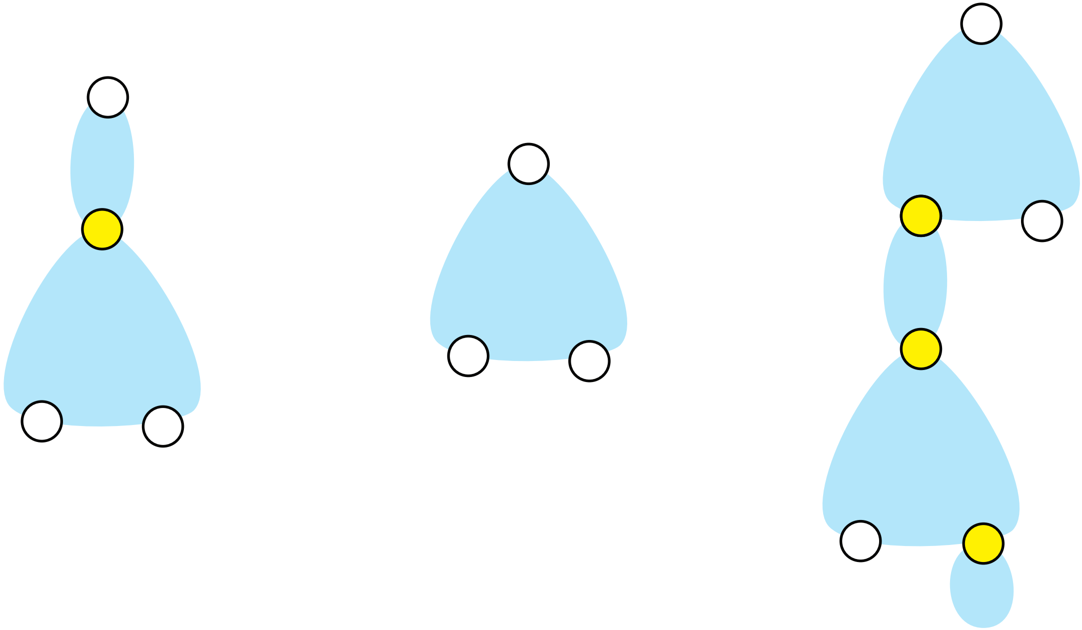

pattern of arity 2
first distinguished leaf,
called port 1
second distinguished leaf,
called port 1

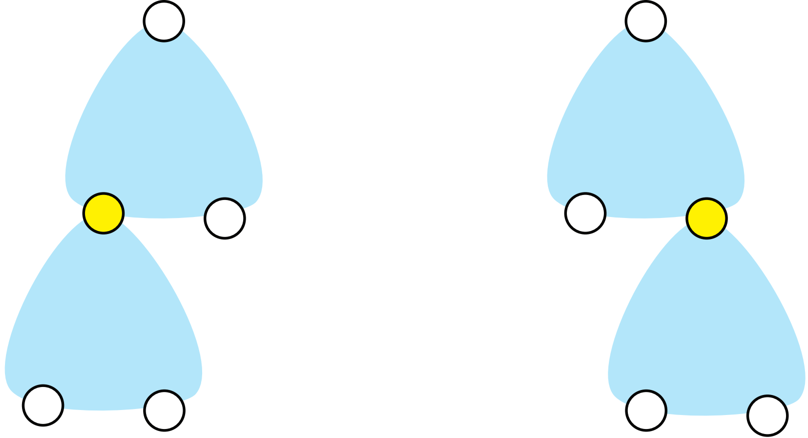

t
Port 1 of t is merged with the root port of t2
t[t1, t2]
t1
t2

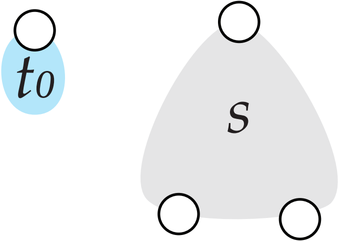


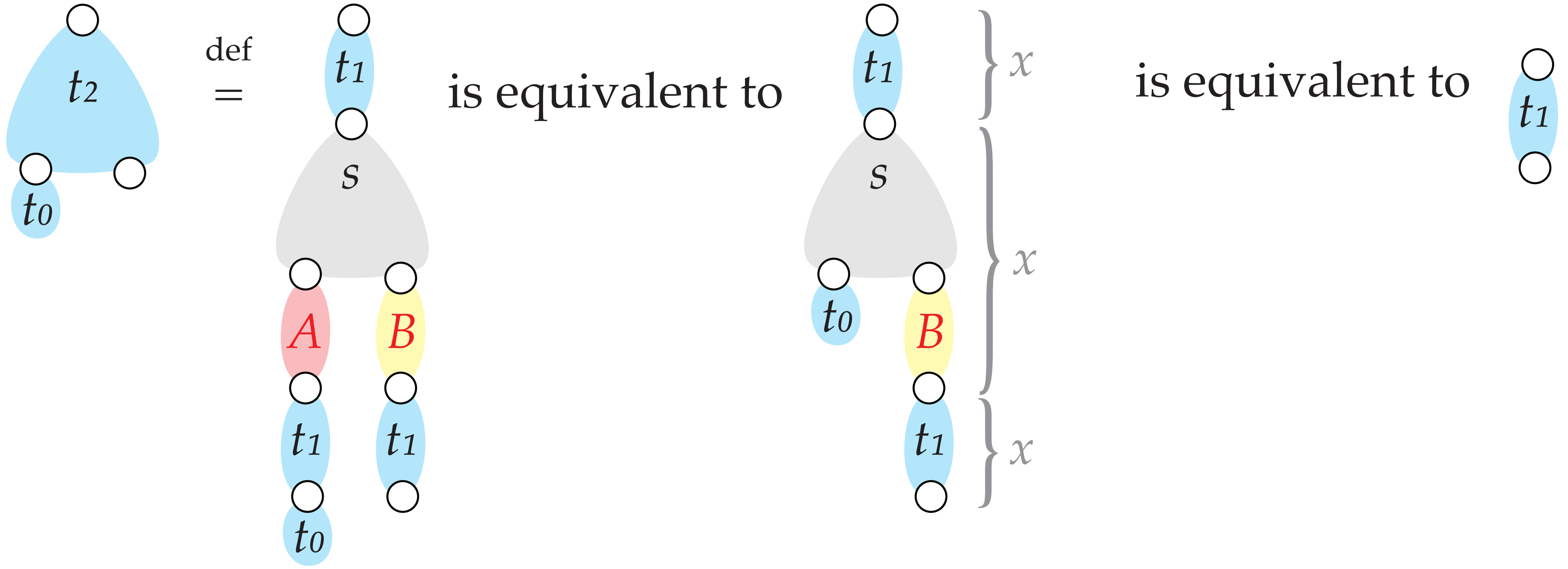
t
the composition
t[t1, t2]
has three leaf ports
t2
t1


t_0
t_1
t_2

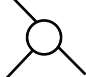


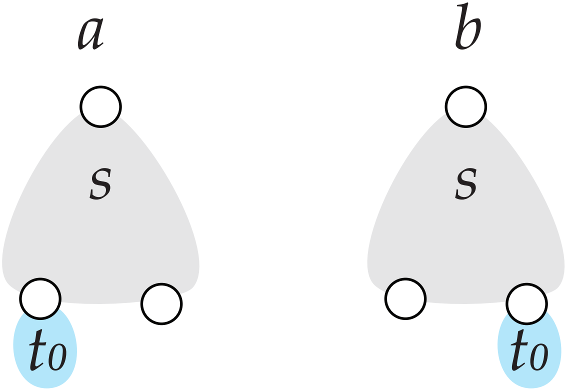



t_0
s

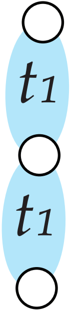


t2
t0
def
=
t1
s
A
B
t1
t1
t0
is equivalent to
t1
x
s
t0
B
x
t1
x
is equivalent to
t1

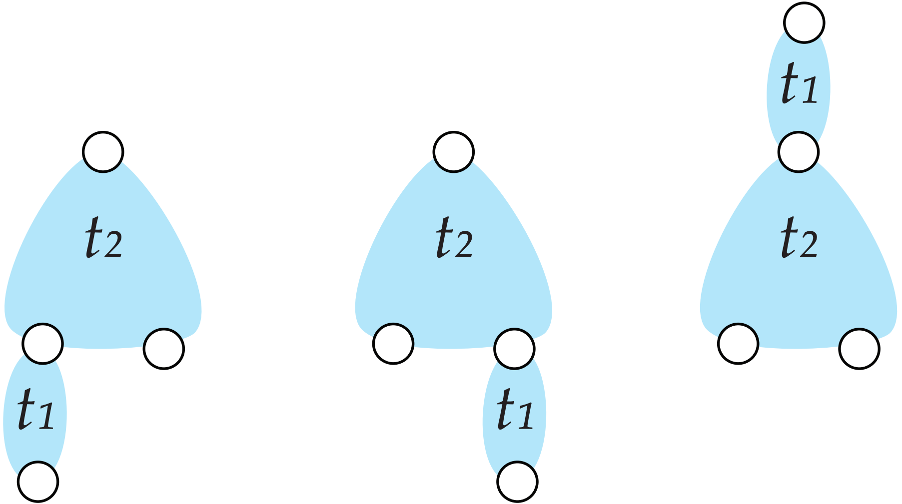




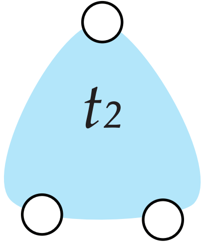
a
s
t0
b
s
t0

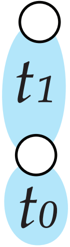


is equivalent to

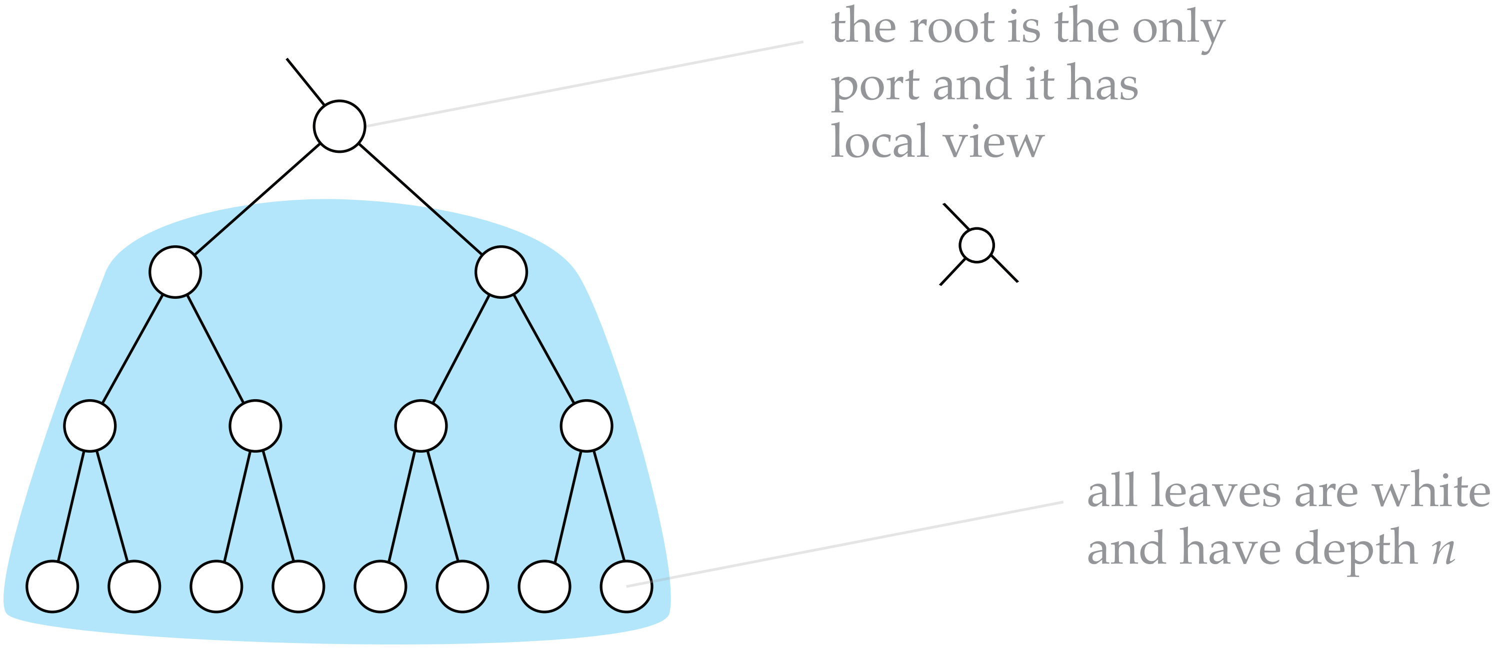


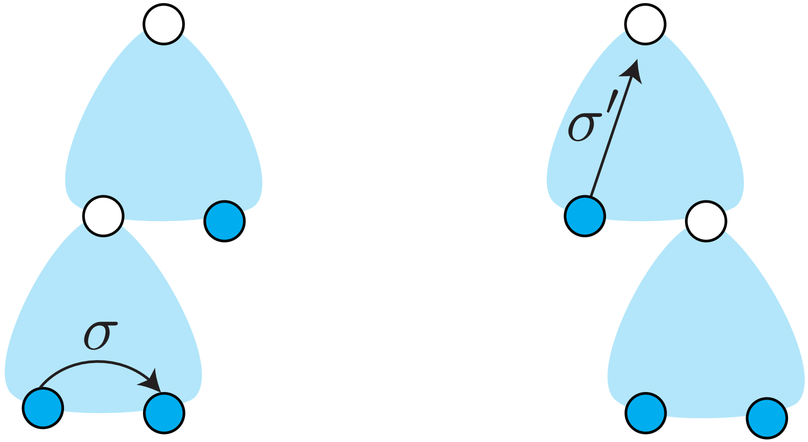
are equivalent to

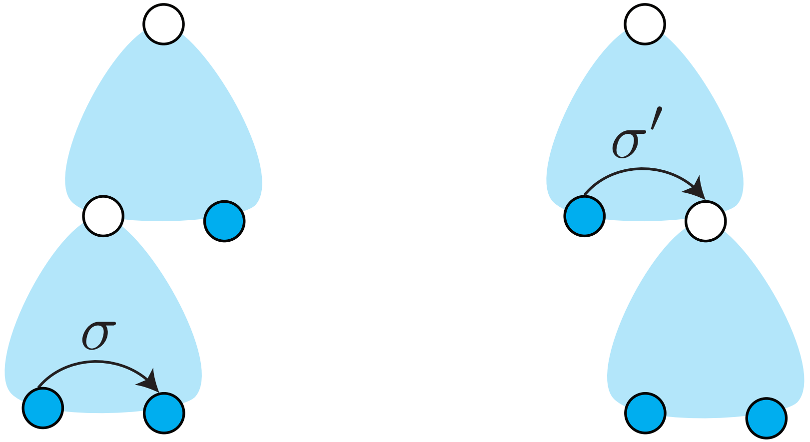




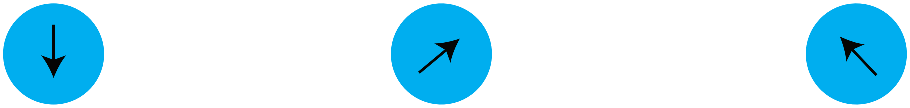
is equivalent to

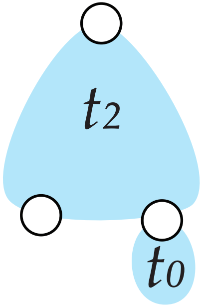

the root is the only
port and it has
local view
all leaves are white
and have depth n


σ
σ'

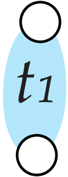


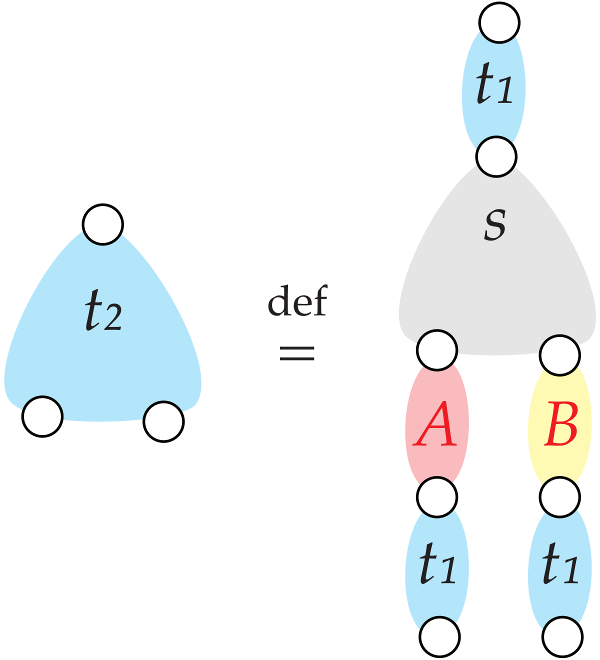
σ
σ′



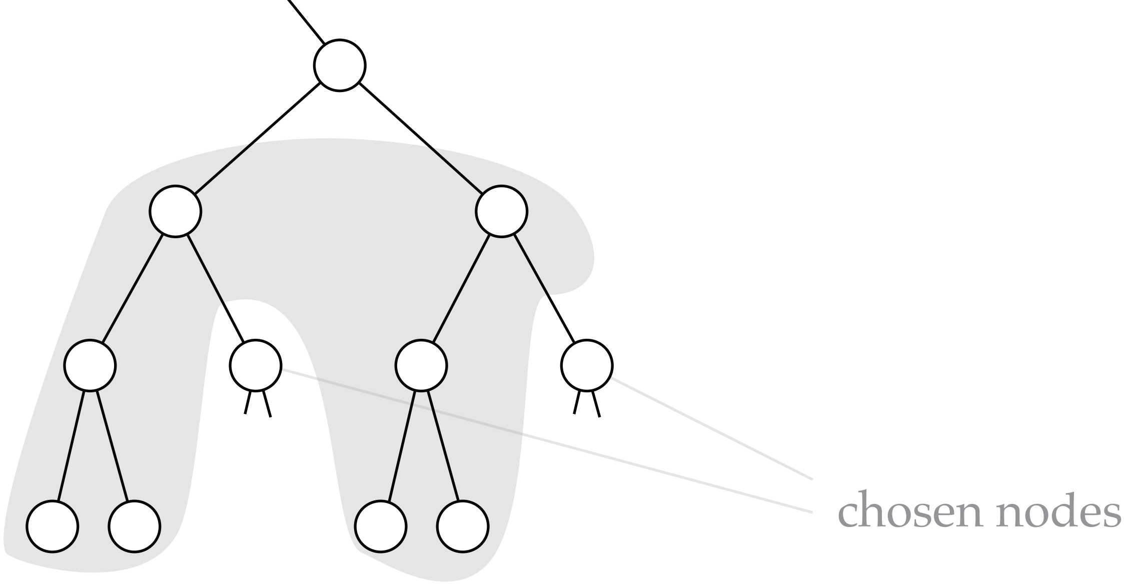


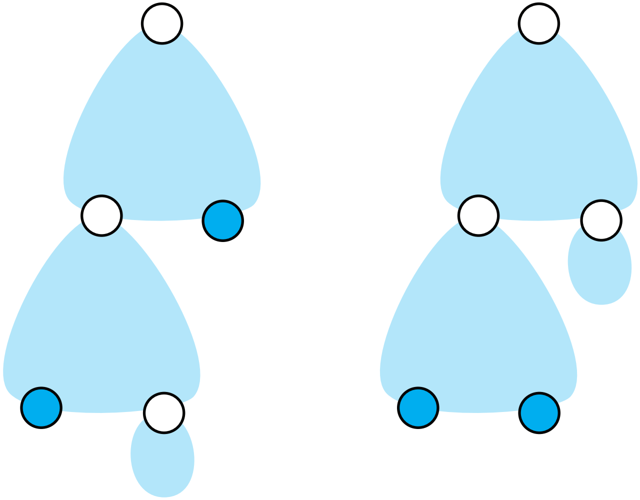
is equivalent to

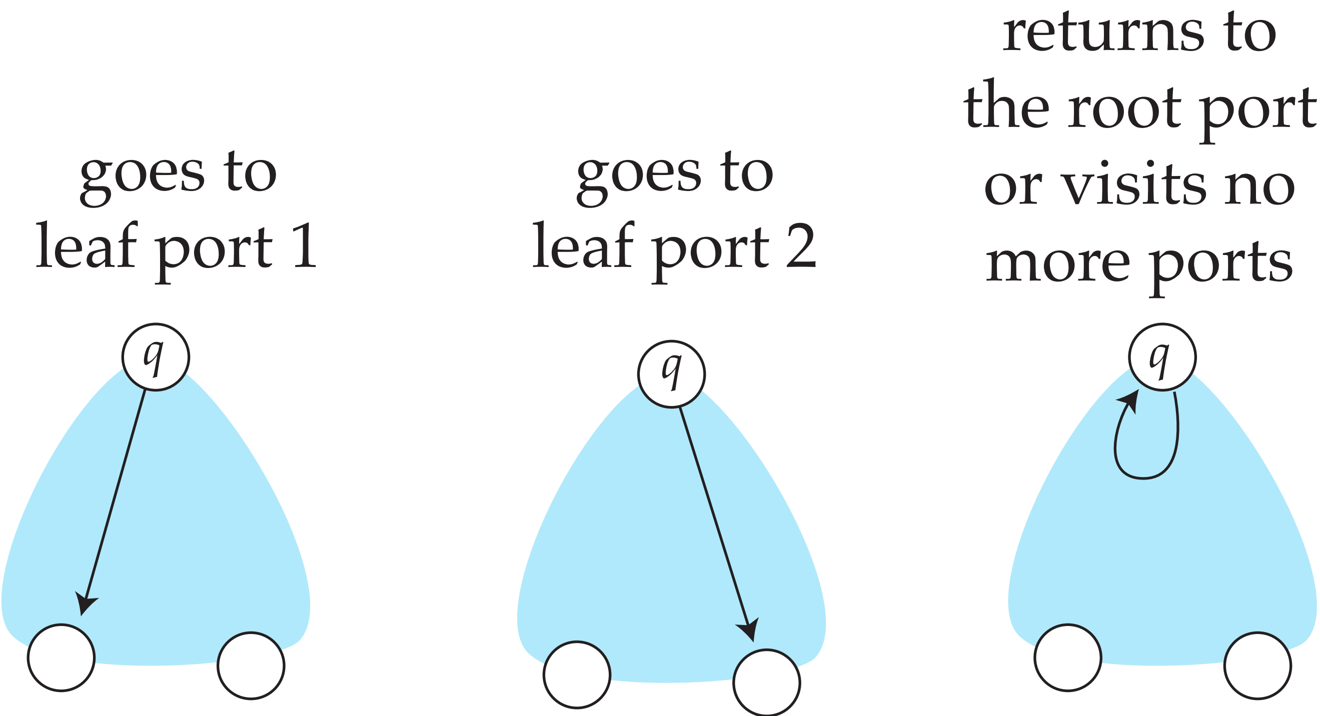


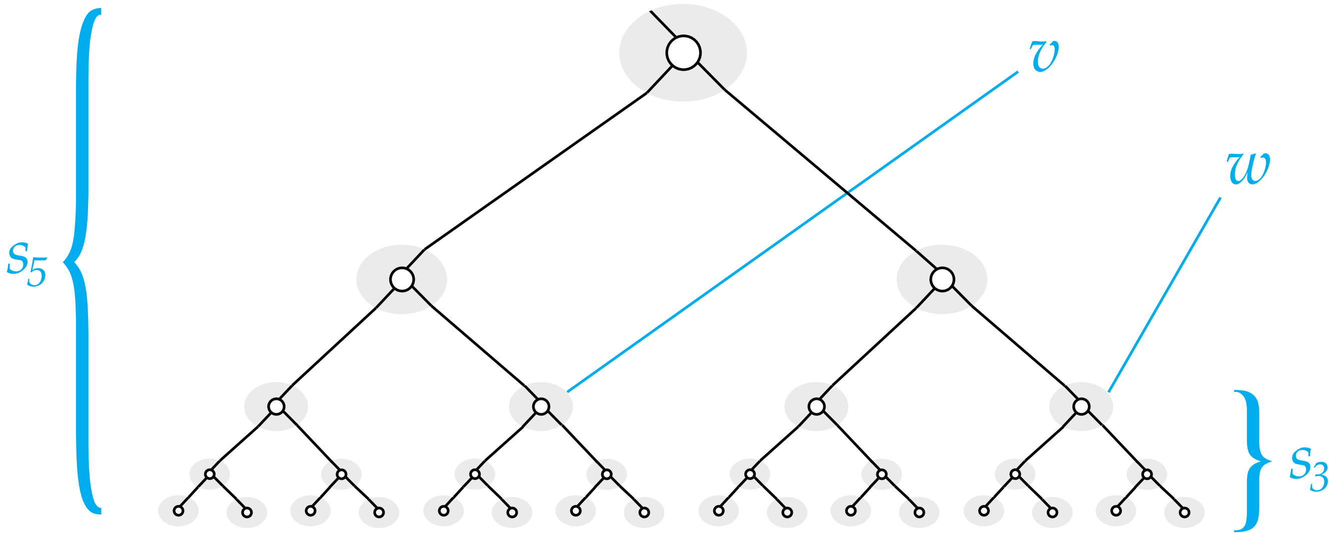


t2
def
=
t1
s
A
B
t1
t1

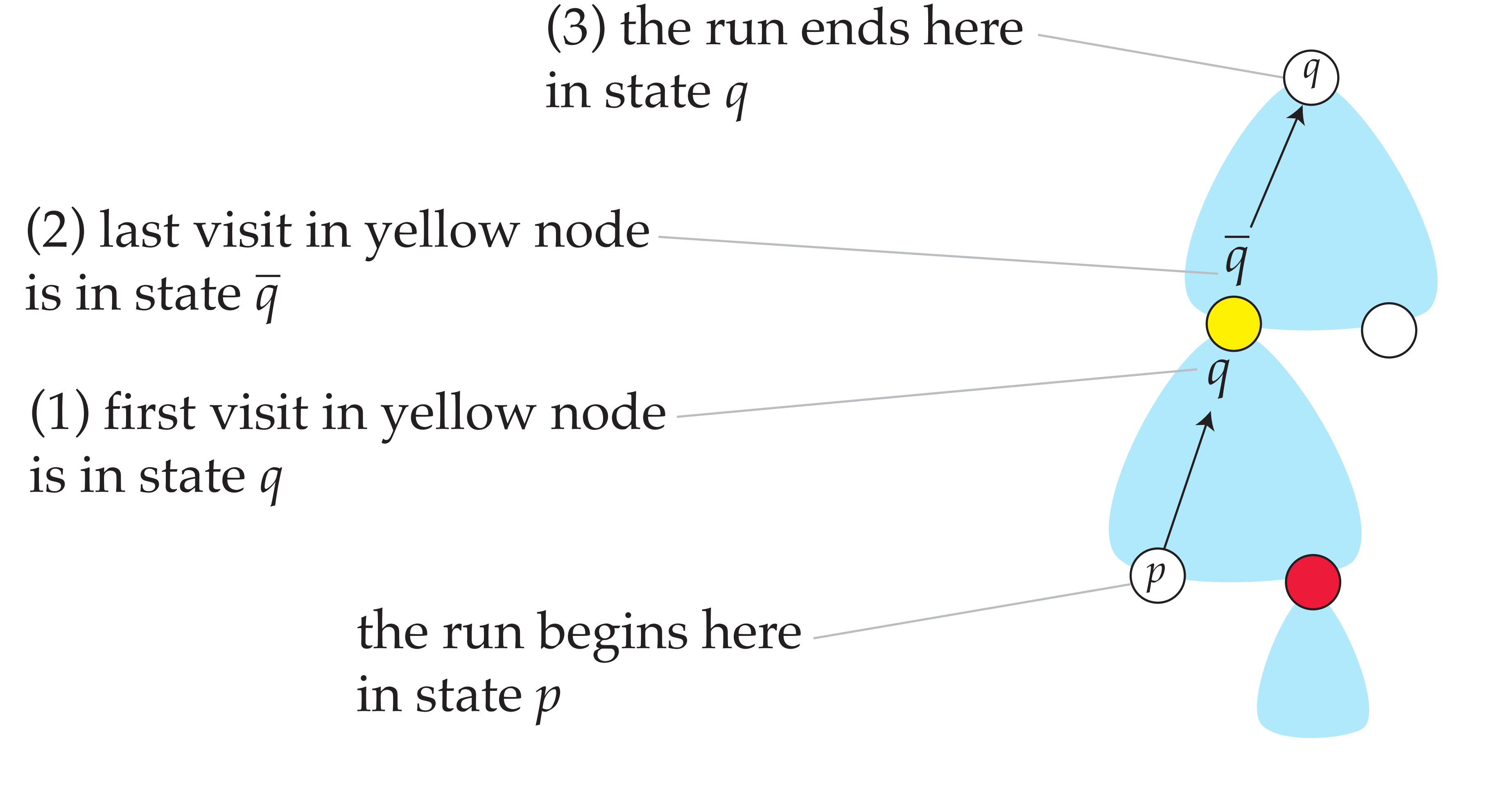

chosen nodes



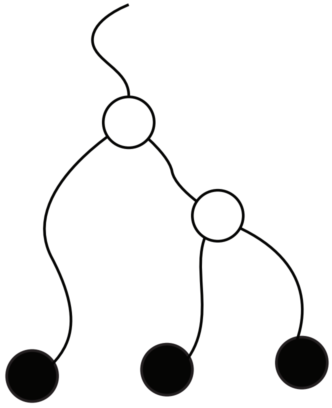

goes to leaf port 1
goes to leaf port 2
returns to the root port or visits no more ports
q
q
q

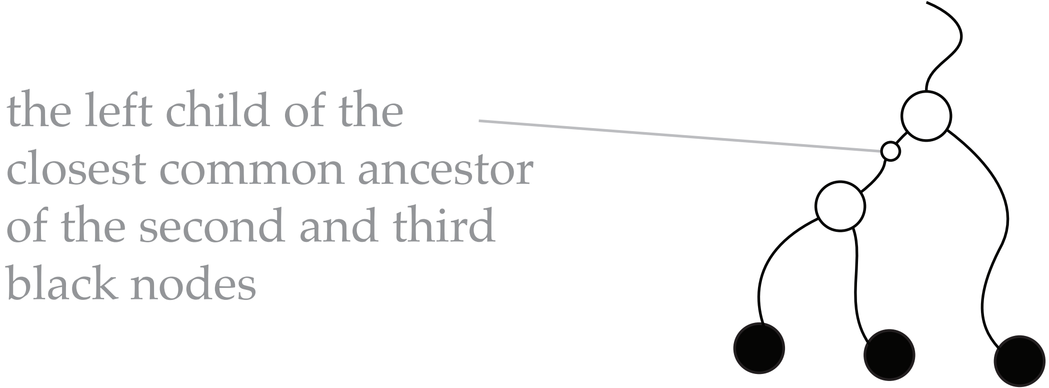


v
w
S_5
S_3

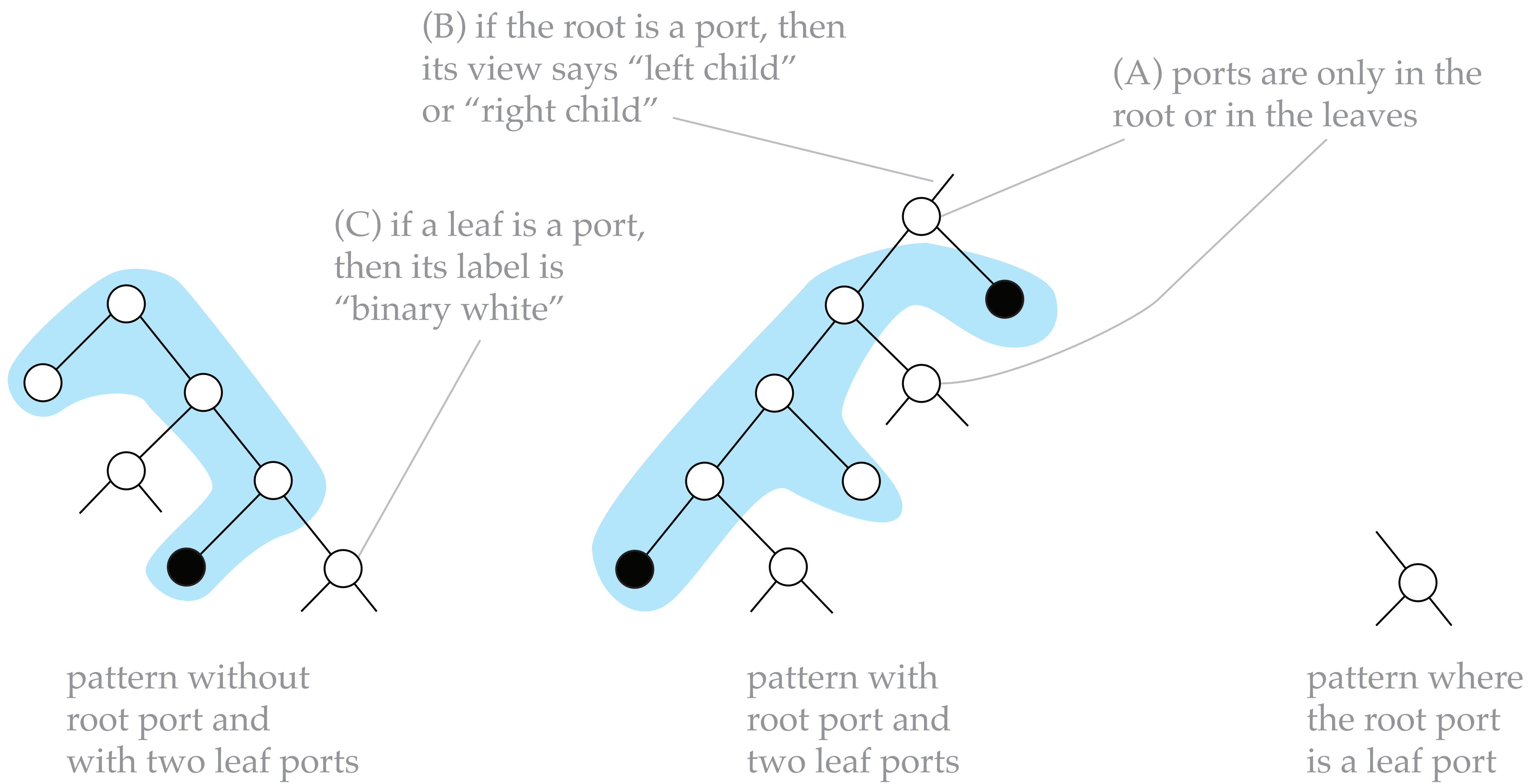


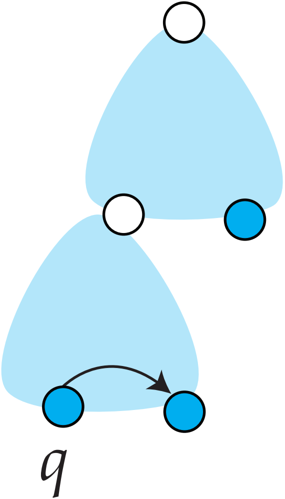
(3) the run ends here
in state q
q
(2) last visit in yellow node
is in state $\overline{q}$
$\overline{q}$
q
(1) first visit in yellow node
is in state q
p
the run begins here
in state p



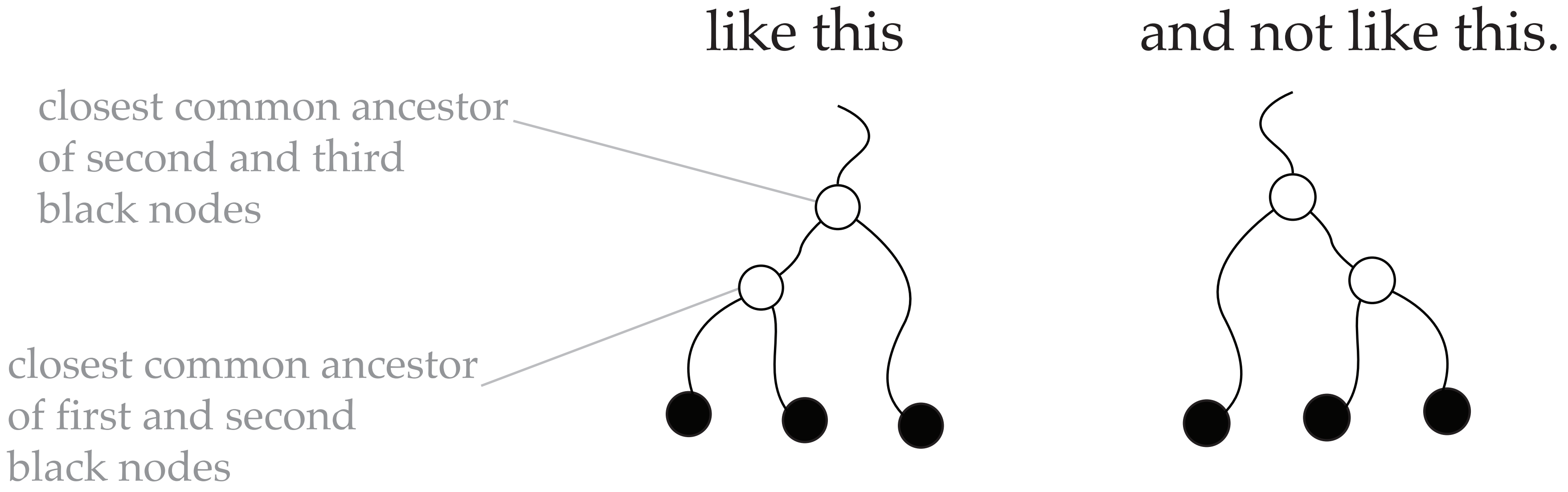

the left child of the closest common ancestor of the second and third black nodes

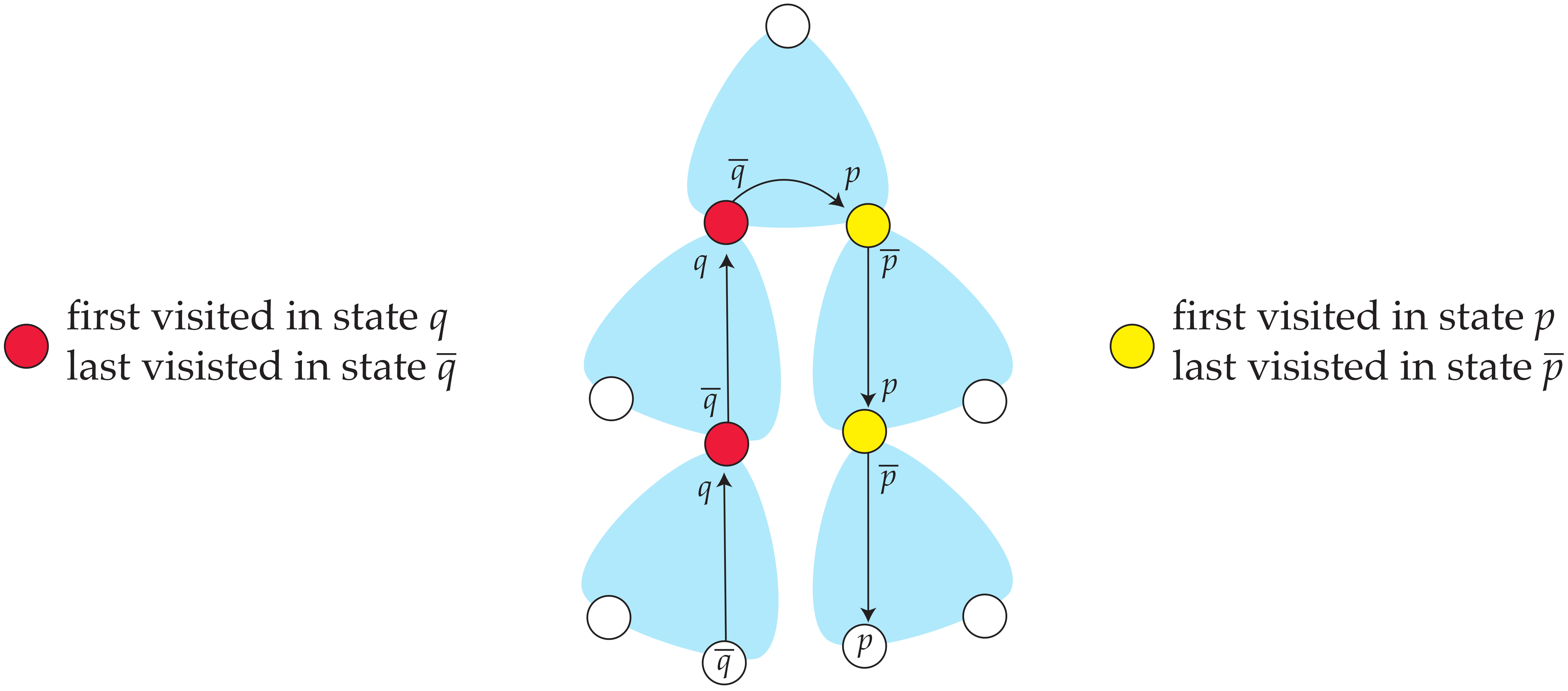
(*)

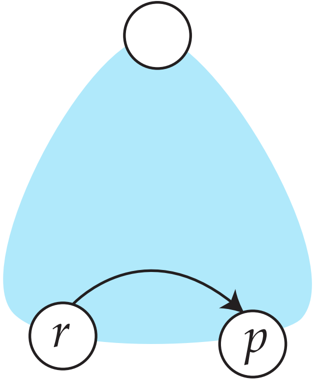

(B) if the root is a port, then
its view says "left child"
or "right child"
(A) ports are only in the
root or in the leaves
(C) if a leaf is a port,
then its label is
"binary white"
pattern without
root port and
with two leaf ports
pattern with
root port and
two leaf ports
pattern where
the root port
is a leaf port

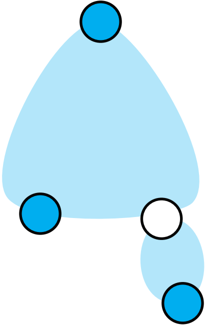

q

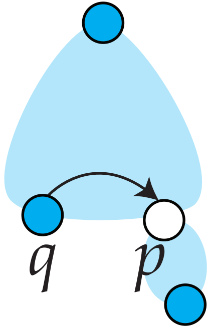

arity zero
white binary
white leaf
black leaf

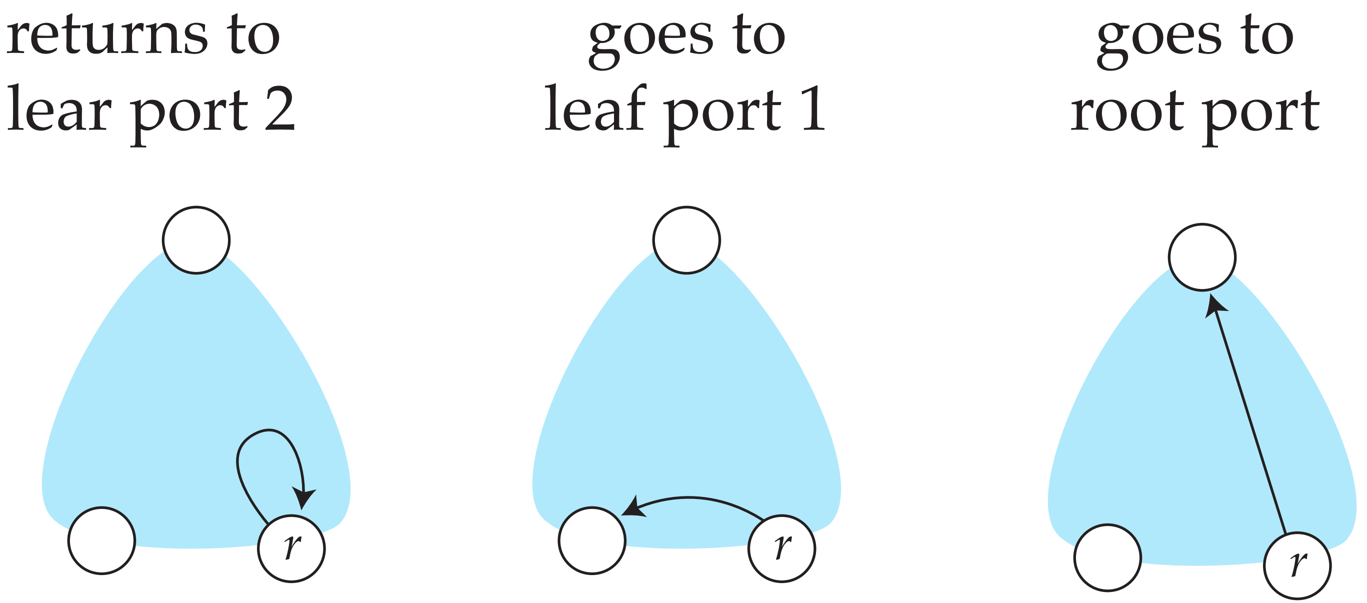

like this
and not like this.
closest common ancestor of second and third black nodes
closest common ancestor of first and second black nodes

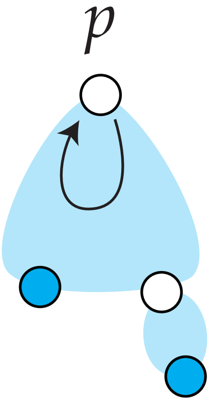

first visited in state q
last visisted in state $\overline{q}$
first visited in state p
last visisted in state $\overline{p}$
$\overline{q}$
p
q
$\overline{p}$
$\overline{q}$
p
q
$\overline{p}$
$\overline{q}$
p

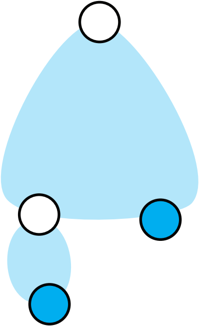

r
p

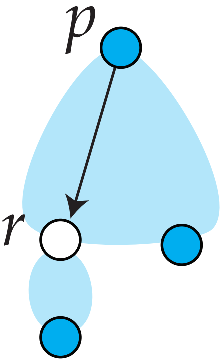


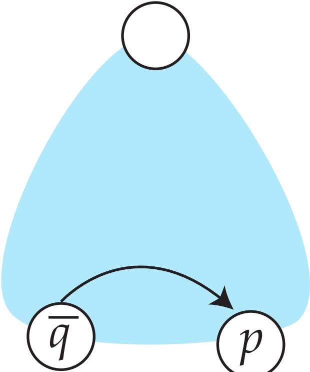

q
p


returns to
lear port 2
goes to
leaf port 1
goes to
root port
r
r
r


p

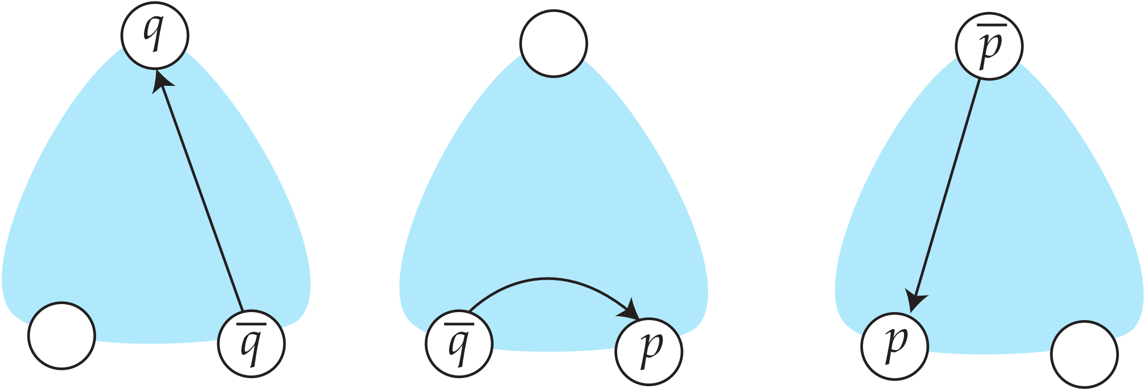



p
r

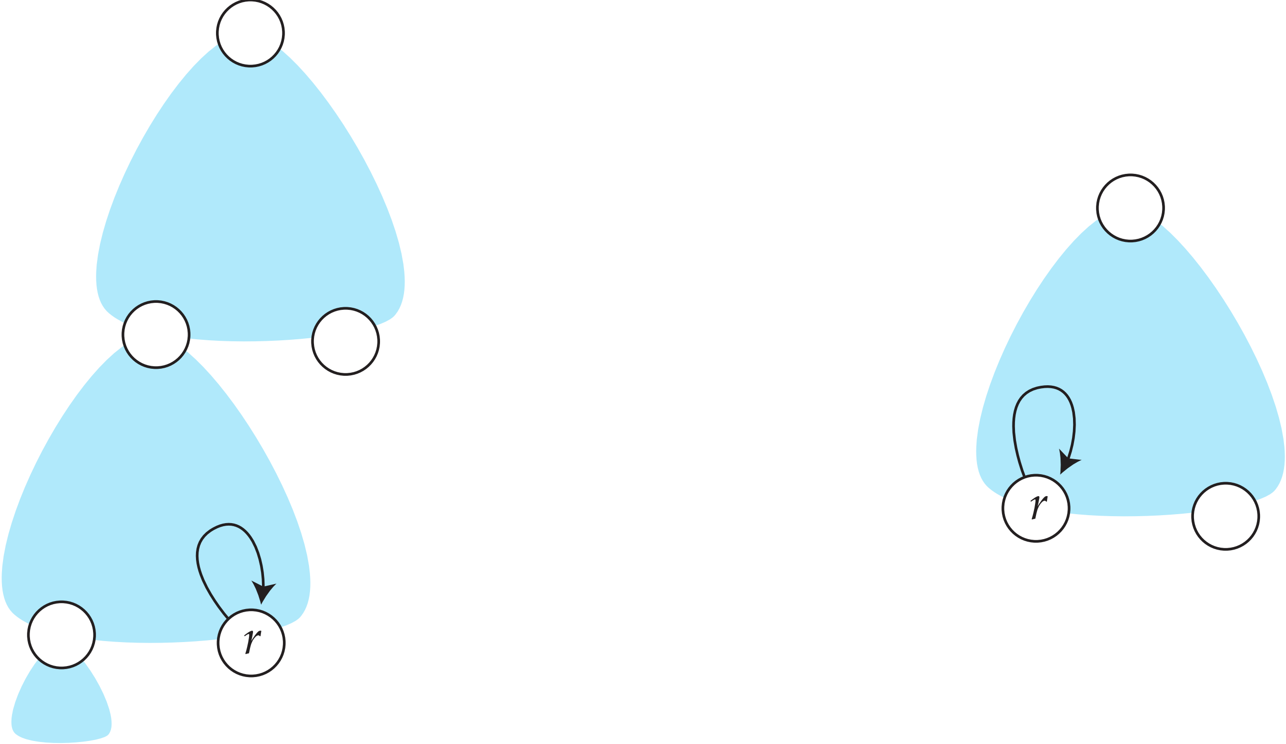


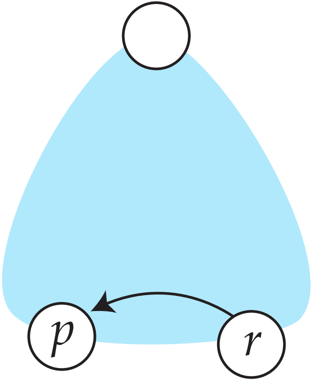
$\overline{q}$
p

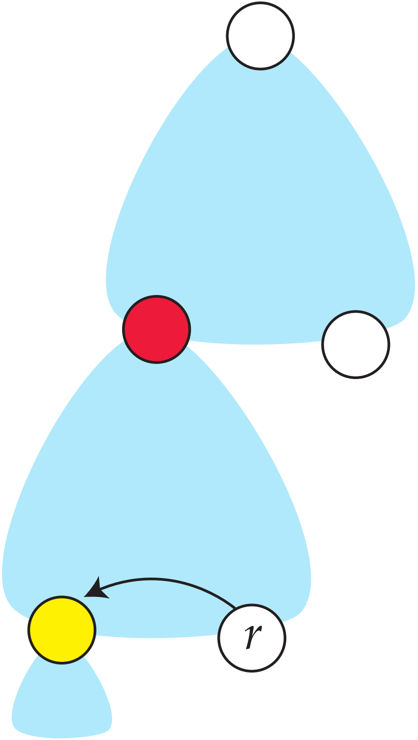
(**)

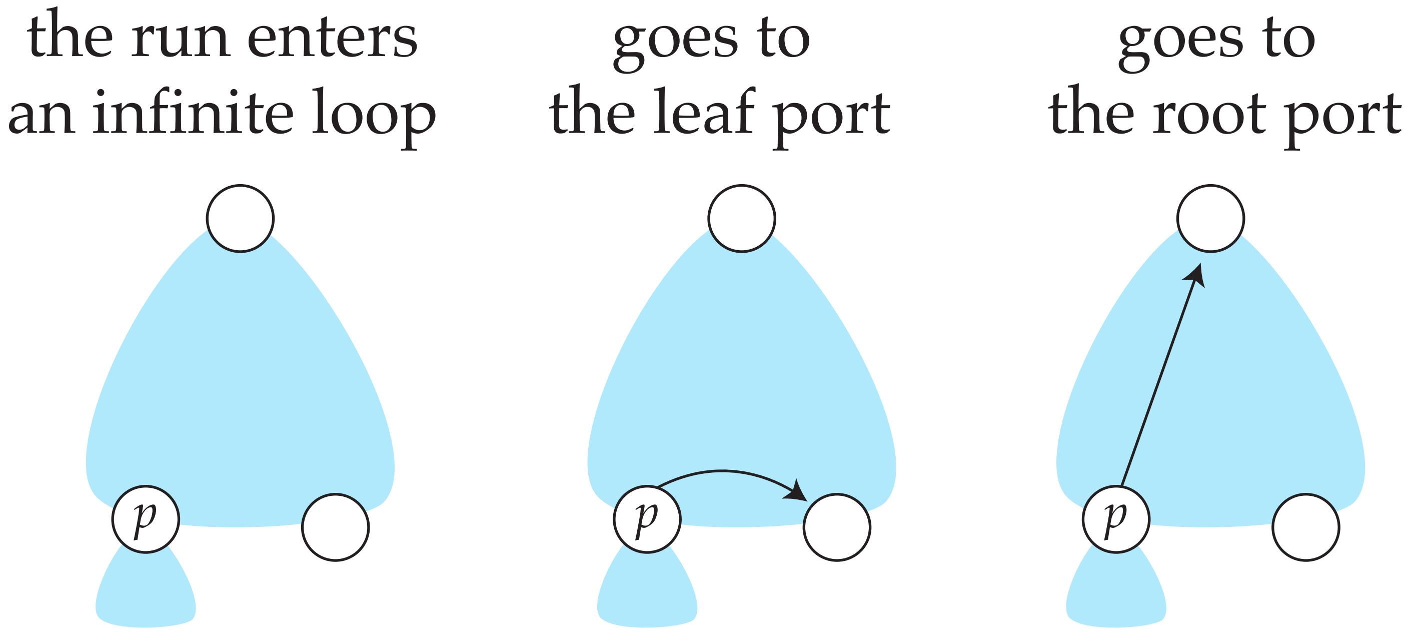


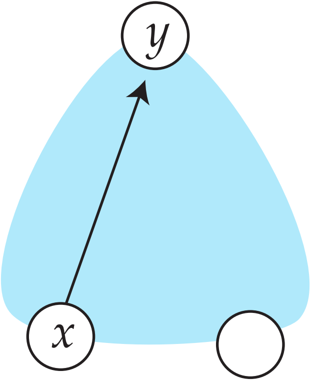
q
q̄
q̄
p
p̄
p


r
r

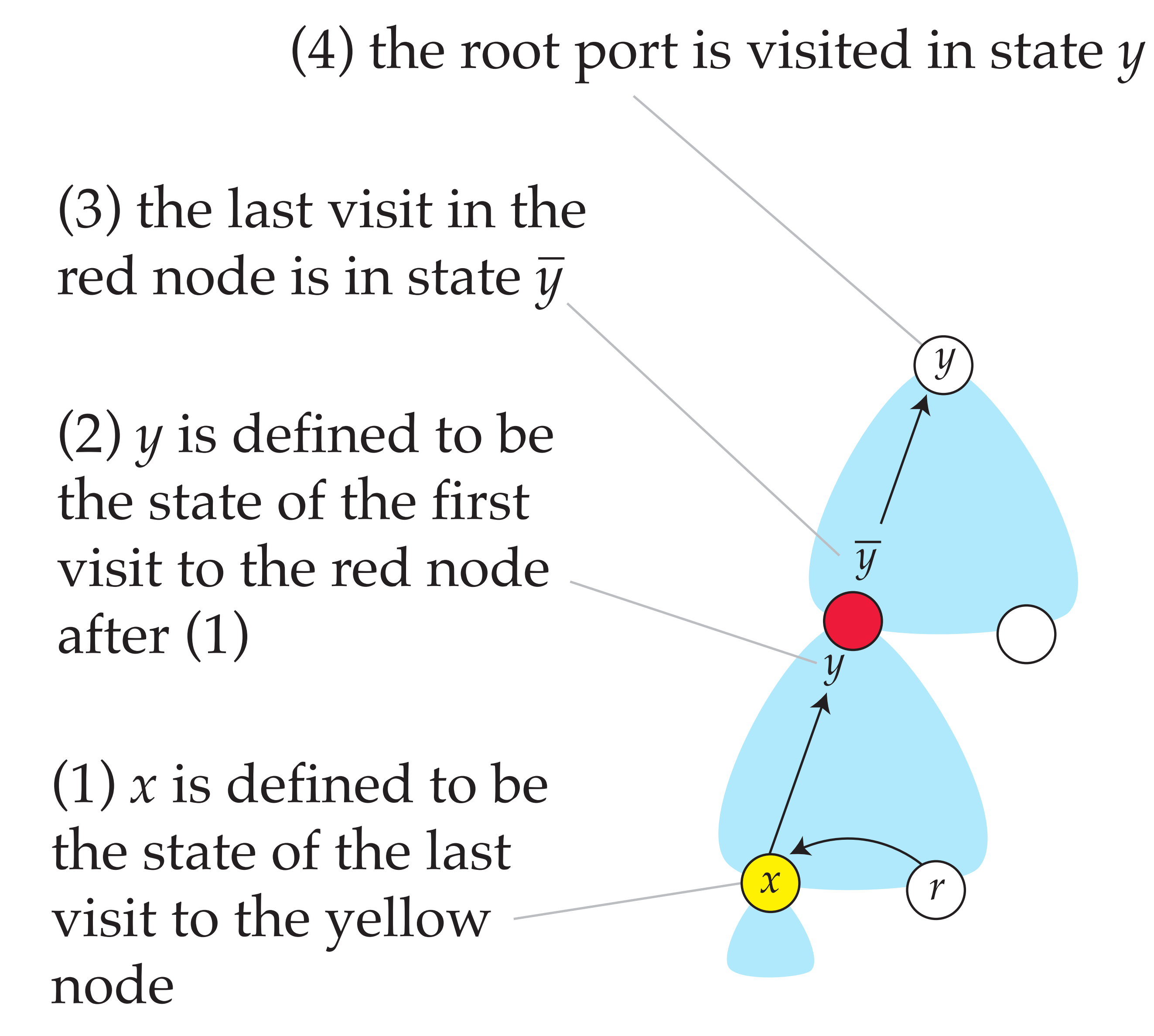

p
r

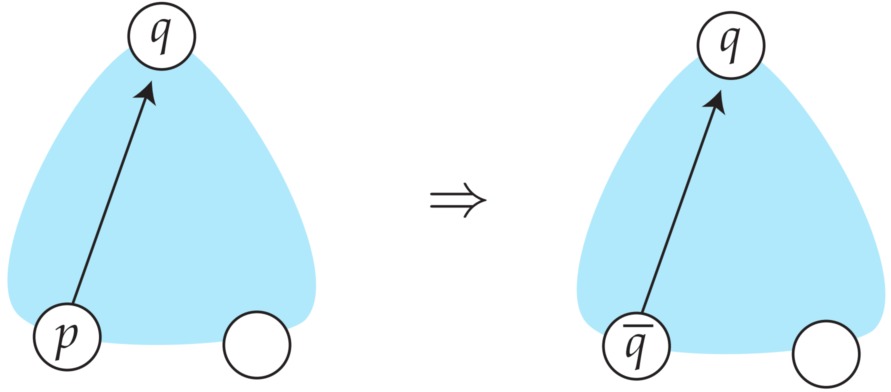

r


the run enters an infinite loop
goes to the leaf port
goes to the root port
p
p
p

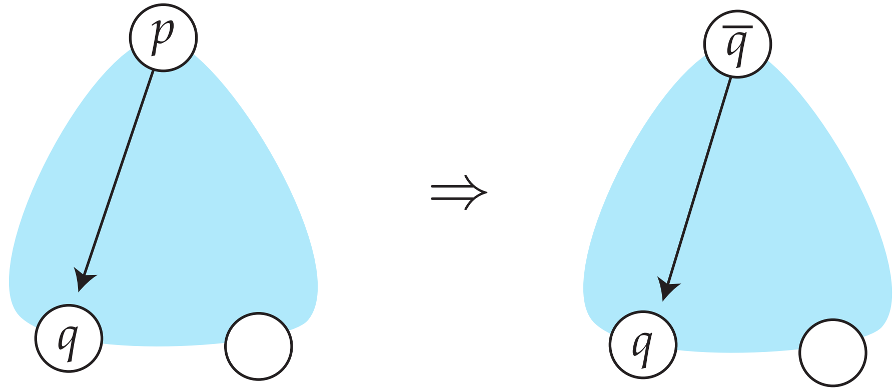


y
x


(4) the root port is visited in state y
(3) the last visit in the red node is in state $\overline{y}$
(2) y is defined to be the state of the first visit to the red node after (1)
(1) x is defined to be the state of the last visit to the yellow node
y
$\overline{y}$
y
x
r

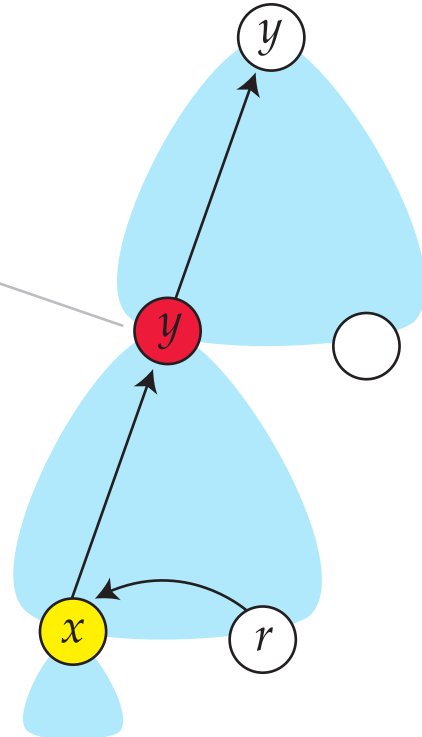


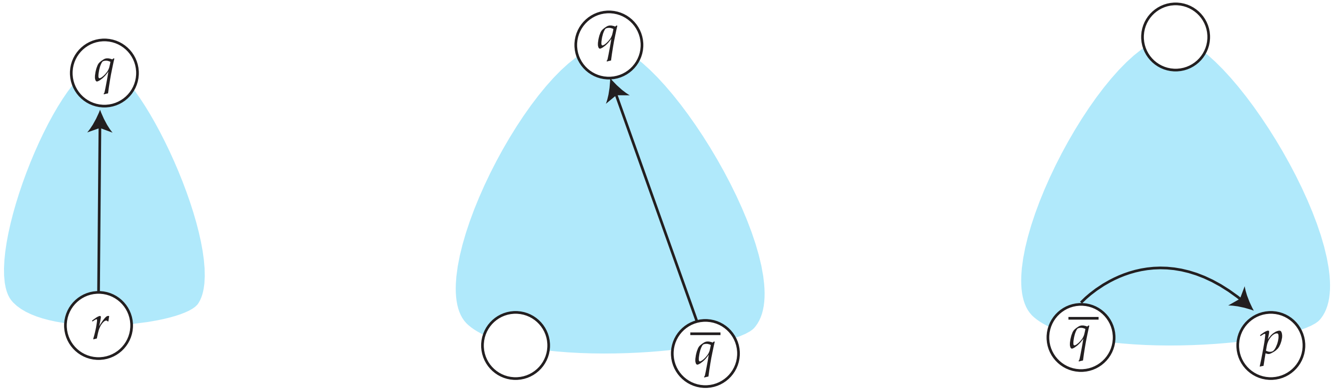
q
p
⇒
q
$\overline{q}$

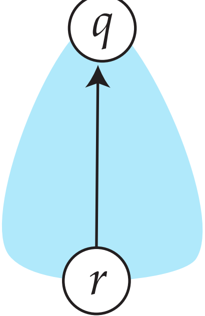


p
q
⇒
$\overline{q}$
q

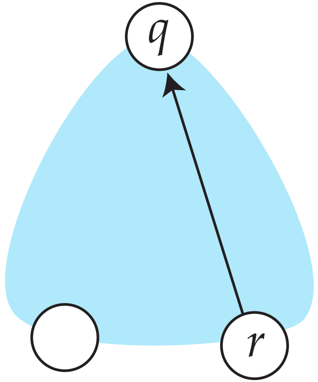
this node is visited in state y, then in state $\overline{y}$.



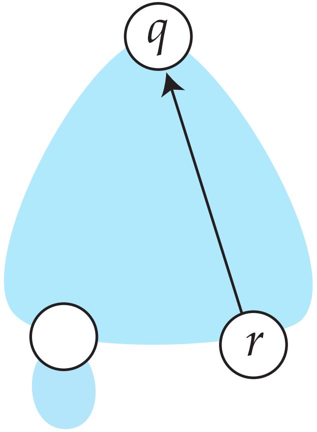

q
r
q
q̄
q̄
p



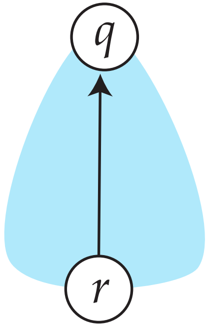
$$(\uparrow)$$

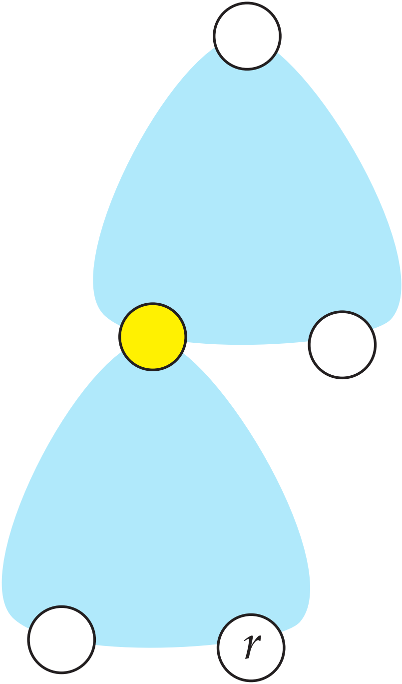


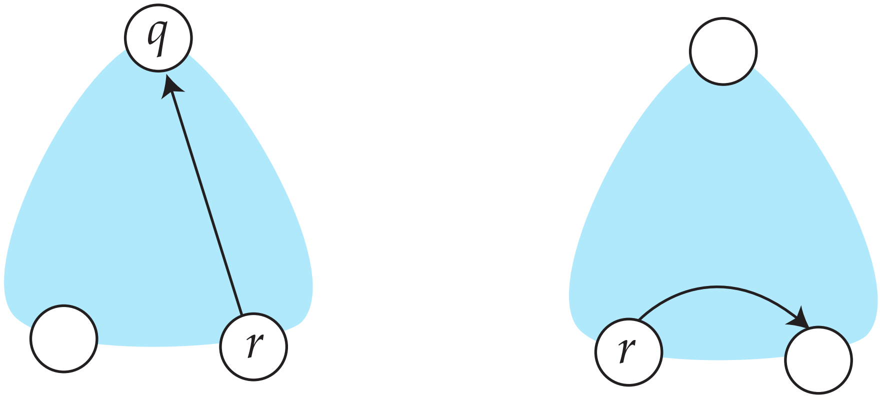
and therefore also

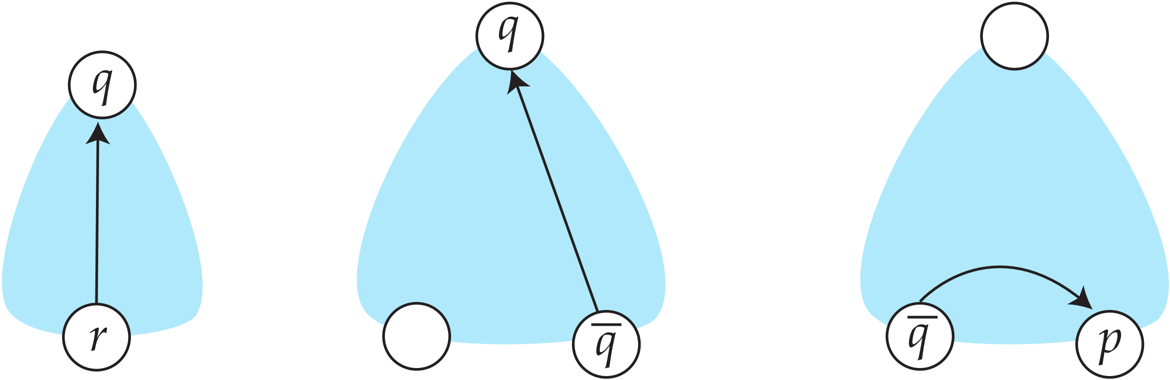


which is equivalent to

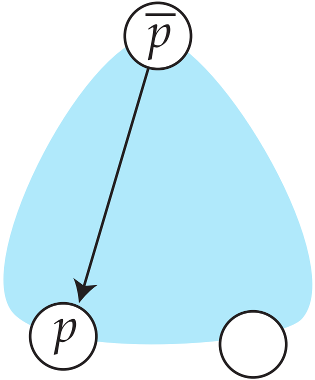


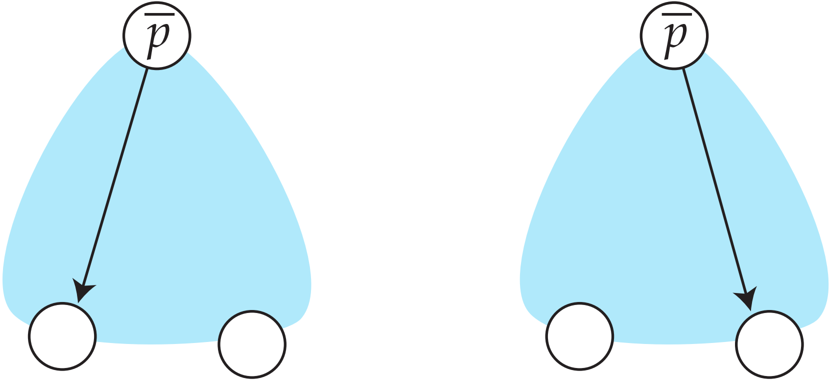

r

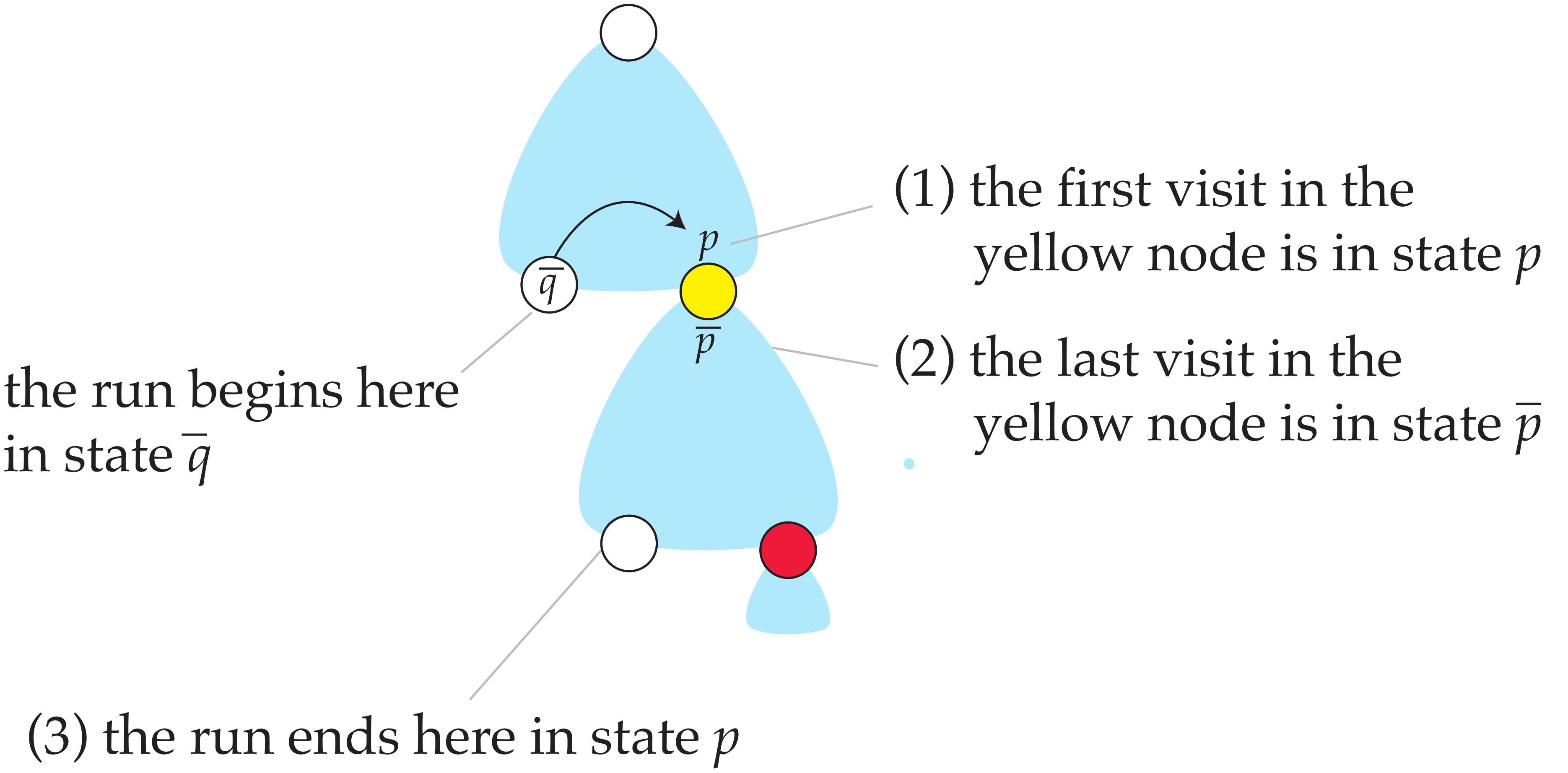

q
r
r



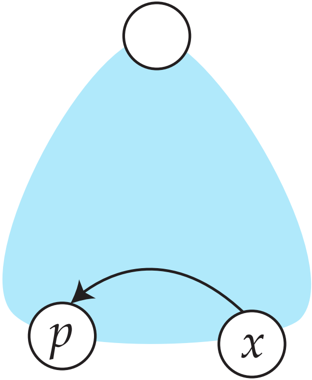
q
r
q
q̄
q̄
p


$\overline{p}$
p

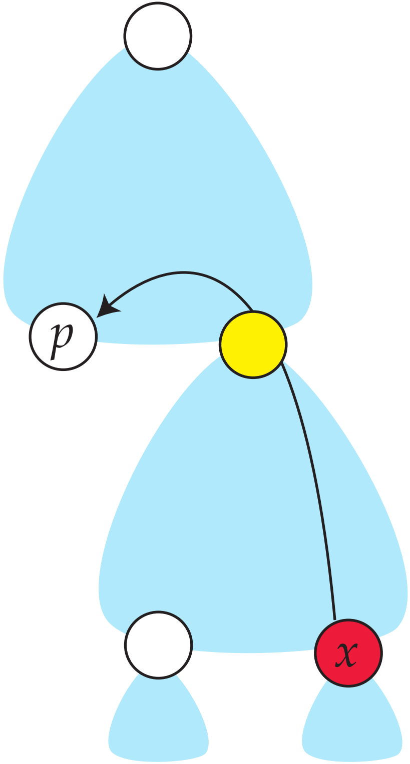

$\overline{p}$
$\overline{p}$

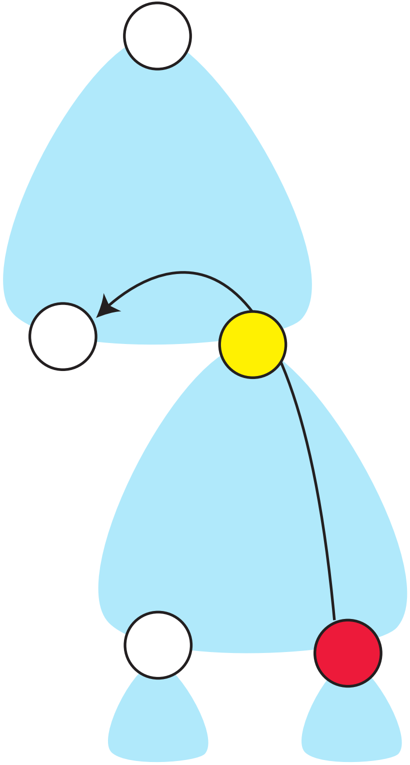

(1) the first visit in the yellow node is in state p
(2) the last visit in the yellow node is in state $\overline{p}$
the run begins here in state $\overline{q}$
(3) the run ends here in state p
p
$\overline{p}$
$\overline{q}$

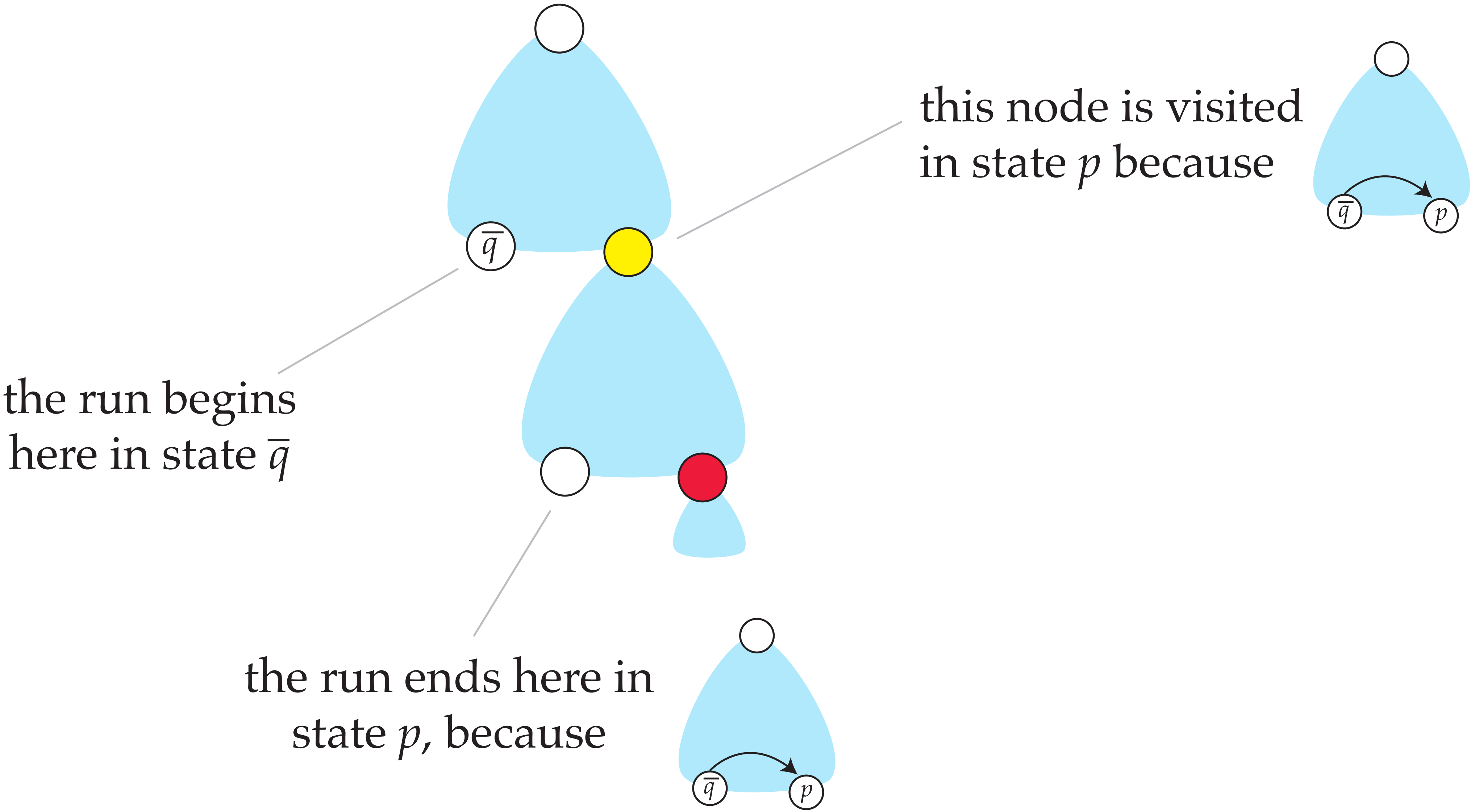
(♠)

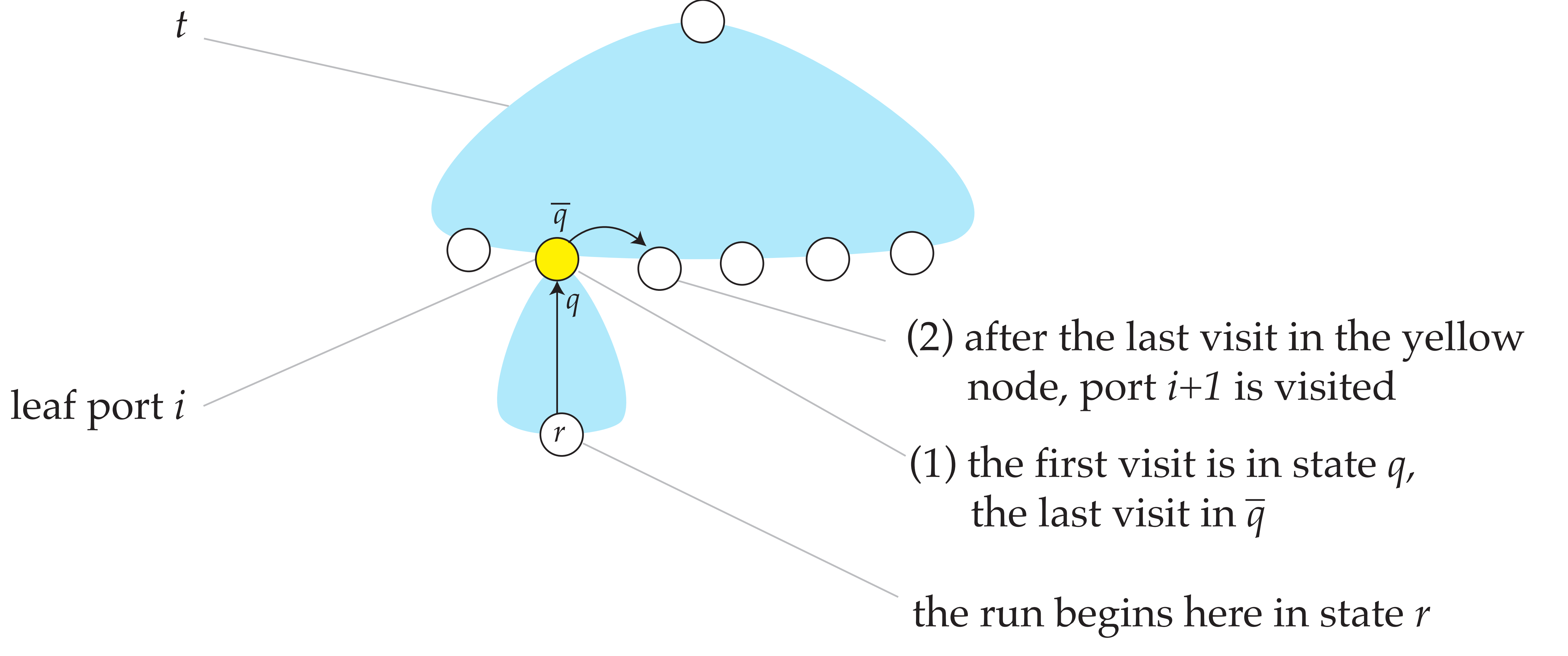


and therefore also

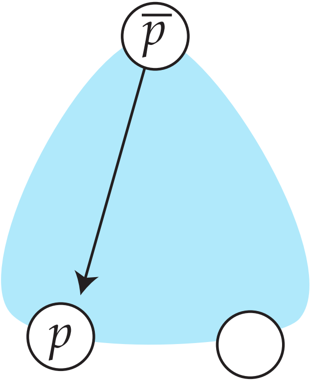


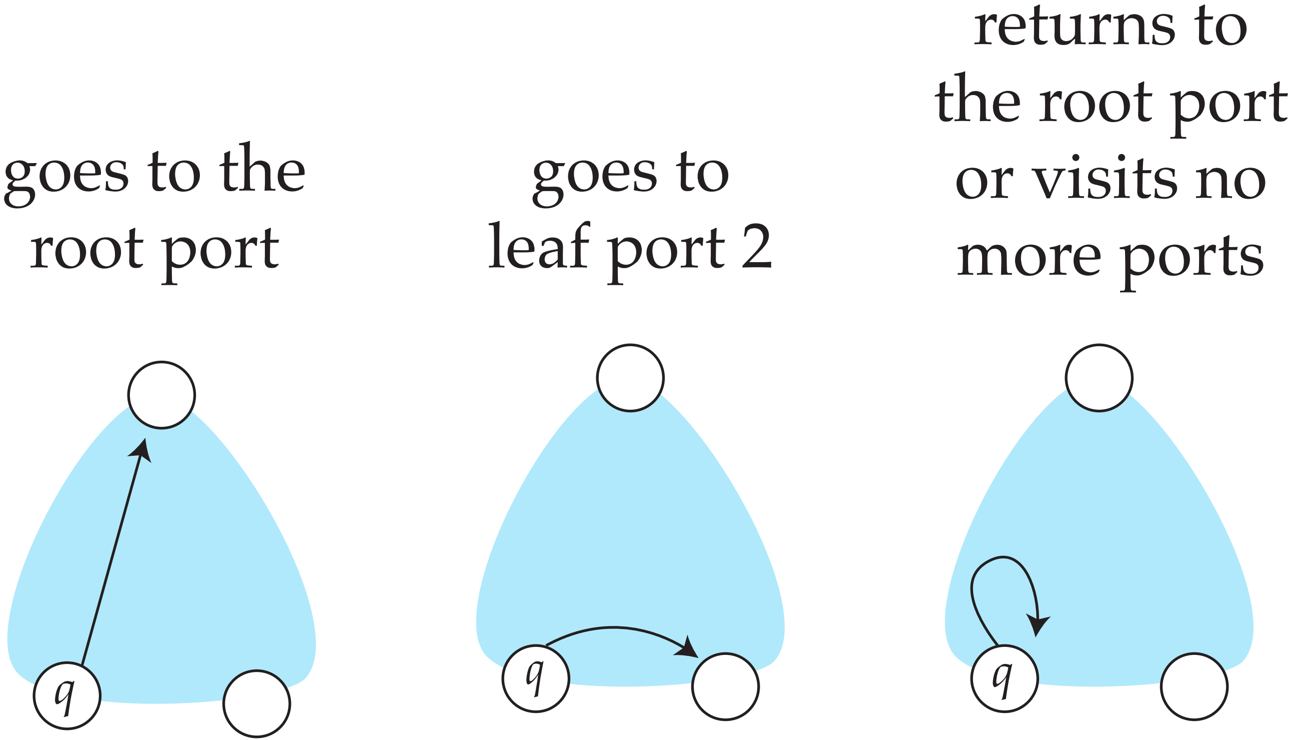


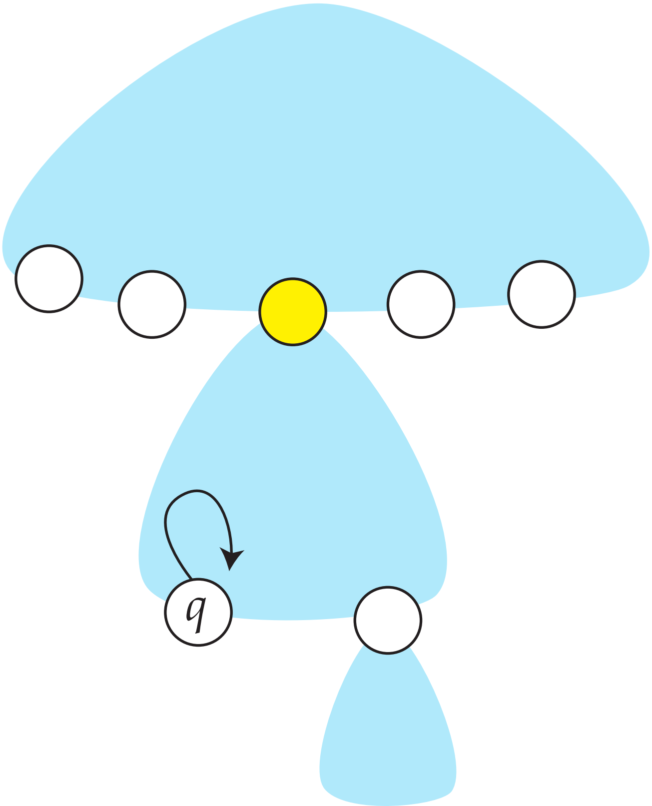

this node is visited in state p because
$\overline{q}$
p
the run begins here in state $\overline{q}$
$\overline{q}$
the run ends here in state p, because
$\overline{q}$
p



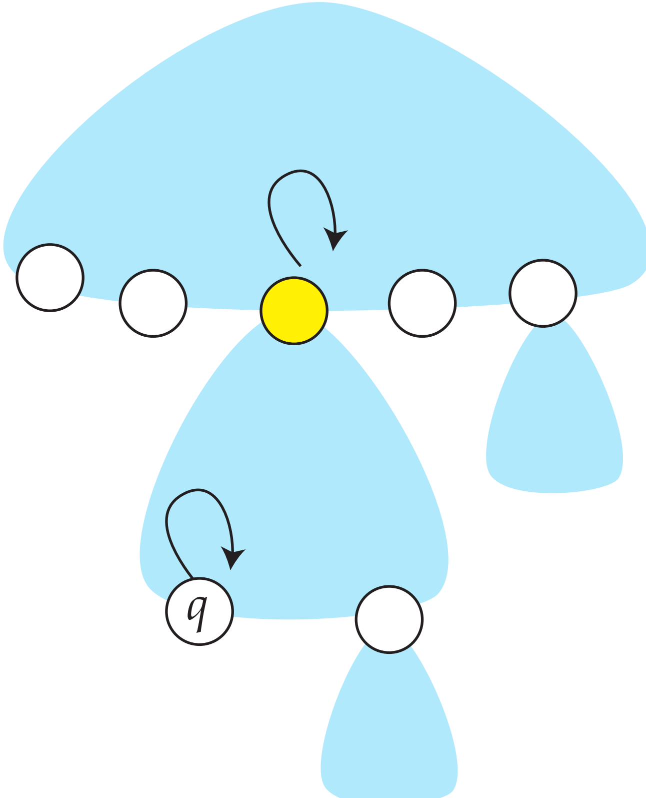
t
$\overline{q}$
q
r
leaf port i
(2) after the last visit in the yellow node, port i+1 is visited
(1) the first visit is in state q, the last visit in $\overline{q}$
the run begins here in state r


$\overline{p}$
p

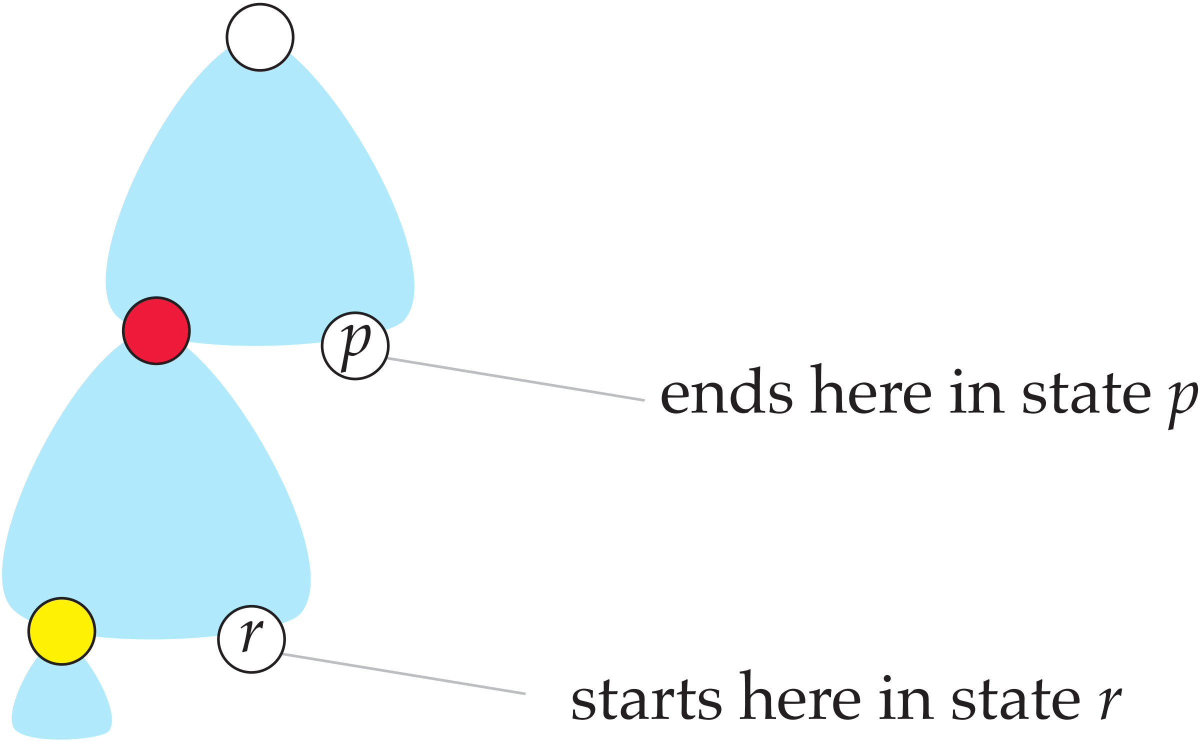

goes to the
root port
goes to
leaf port 2
returns to
the root port
or visits no
more ports
q
q
q



q

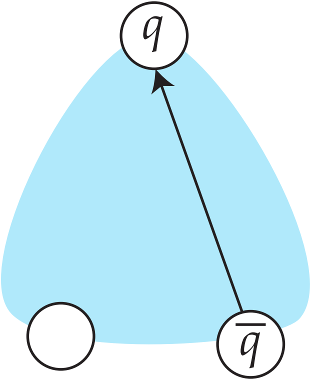


like in this picture

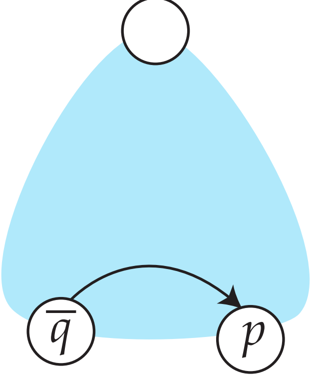


p
ends here in state p
r
starts here in state r

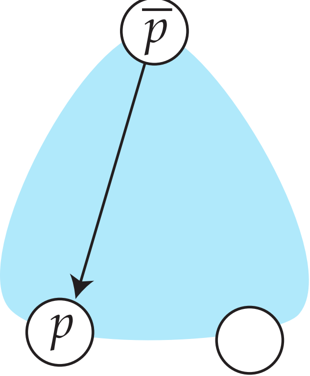
(♣)


q
$\overline{q}$

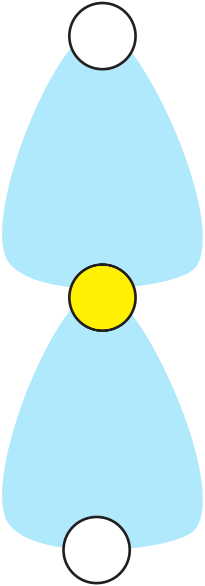
(*)

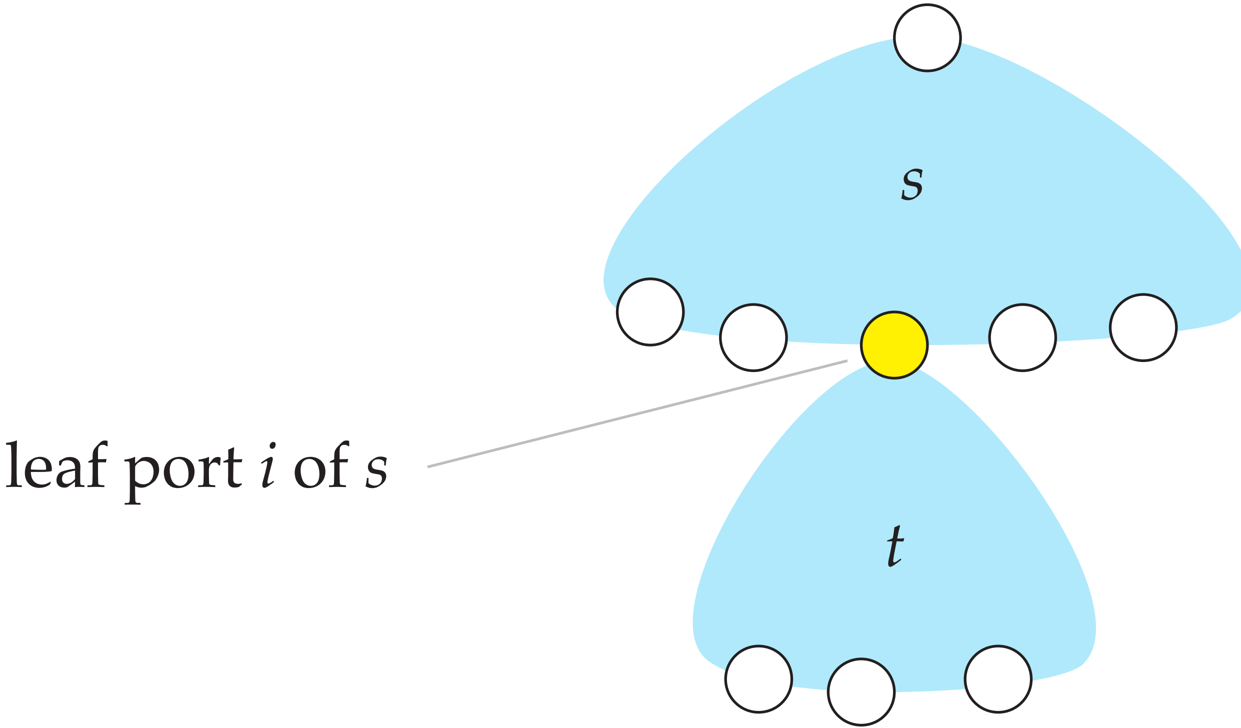

$\overline{q}$
p

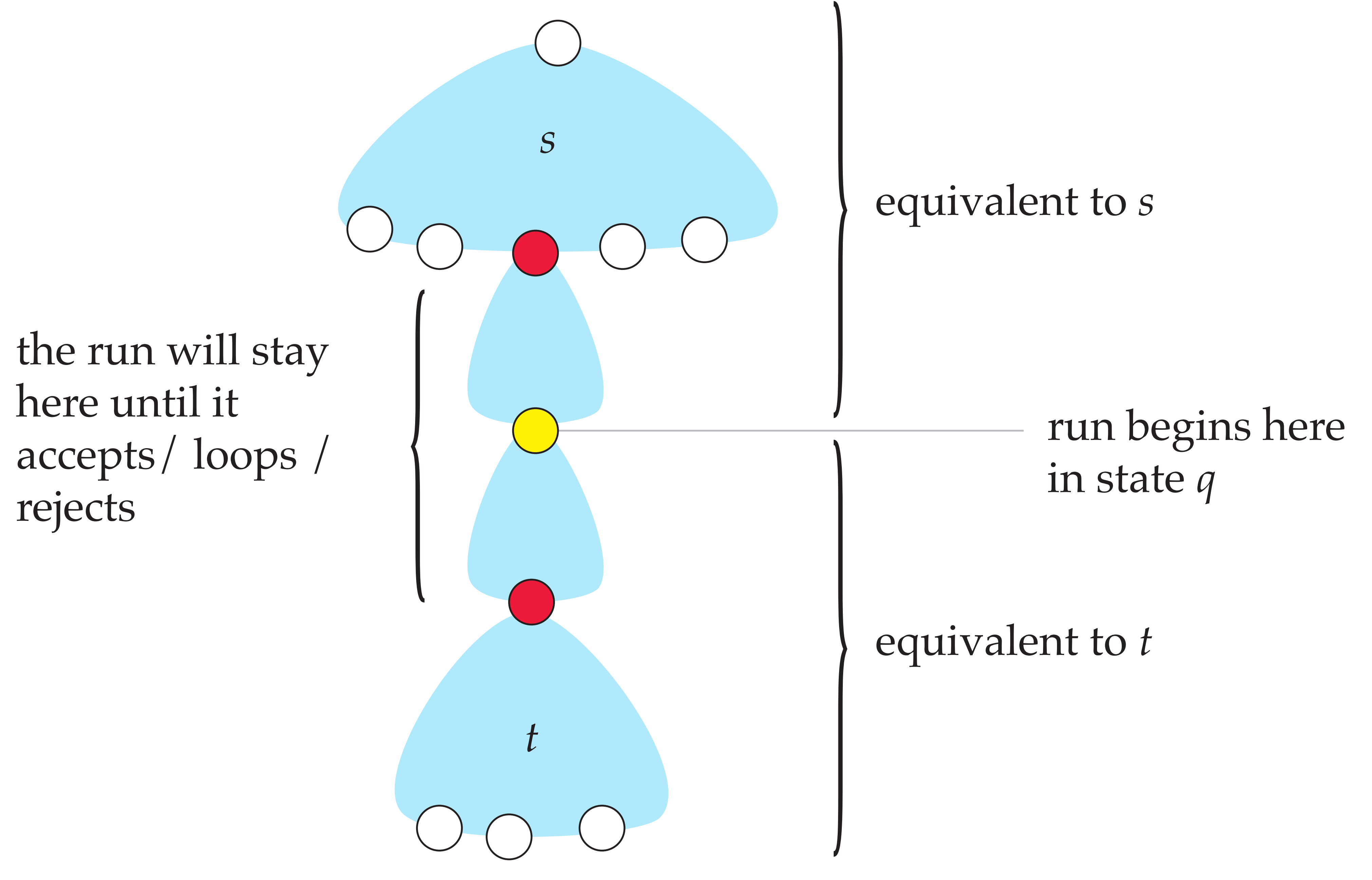
(**)

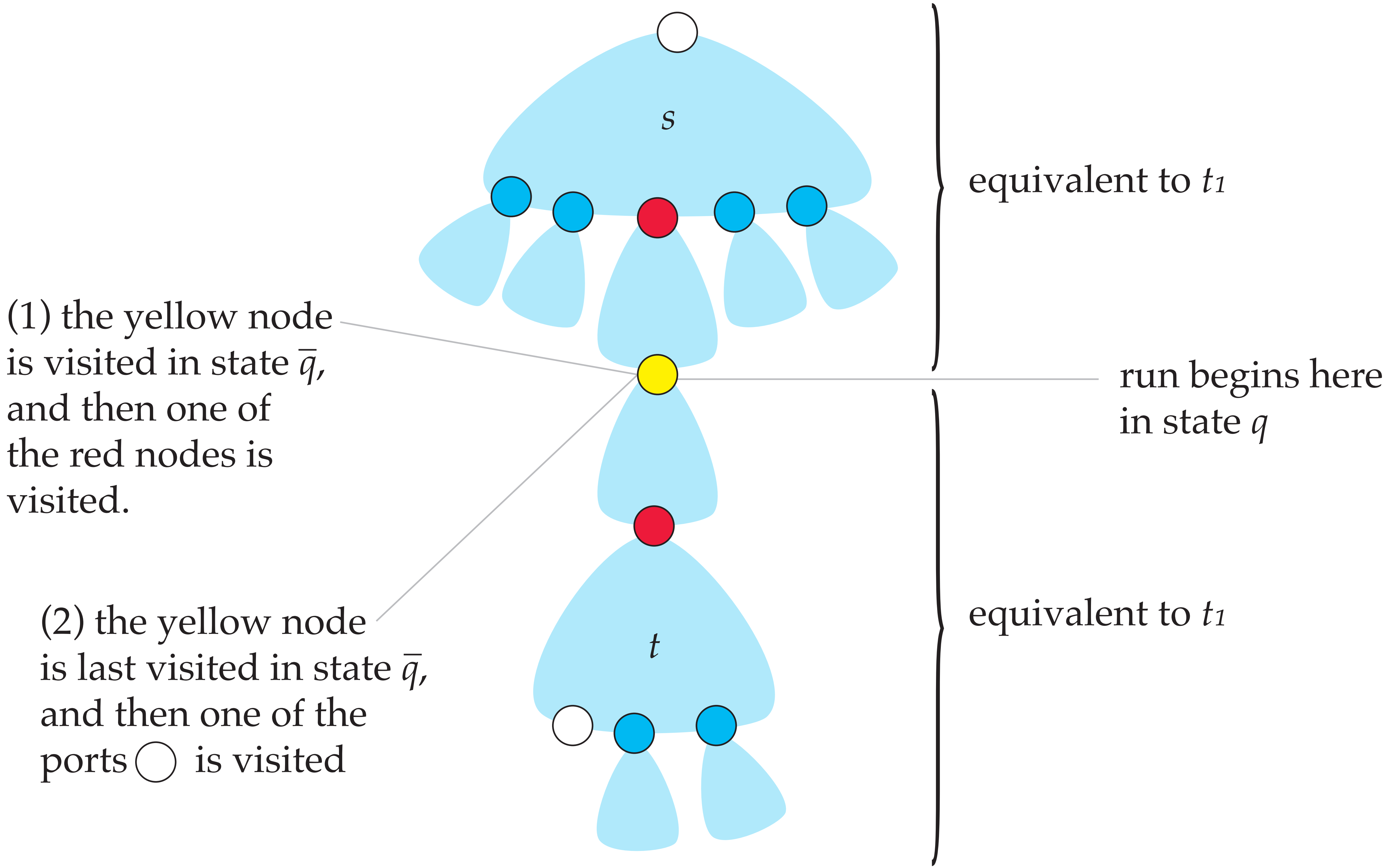

$\overline{p}$
p

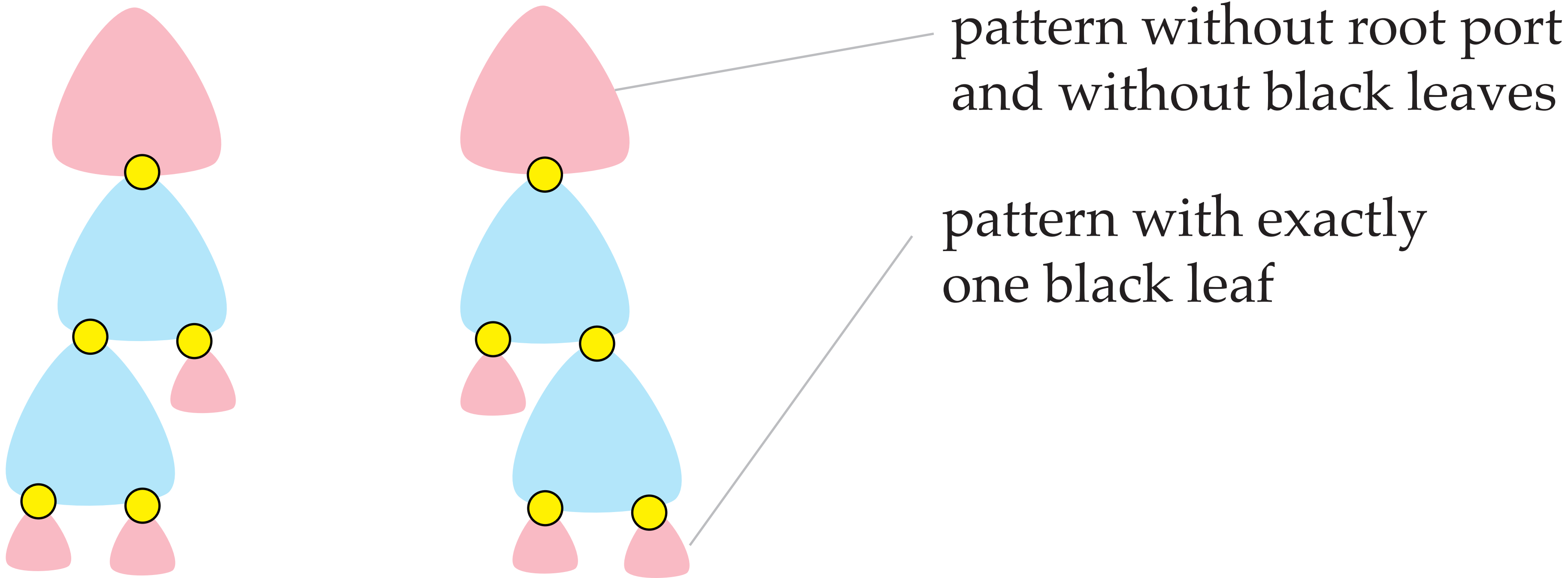
(***)

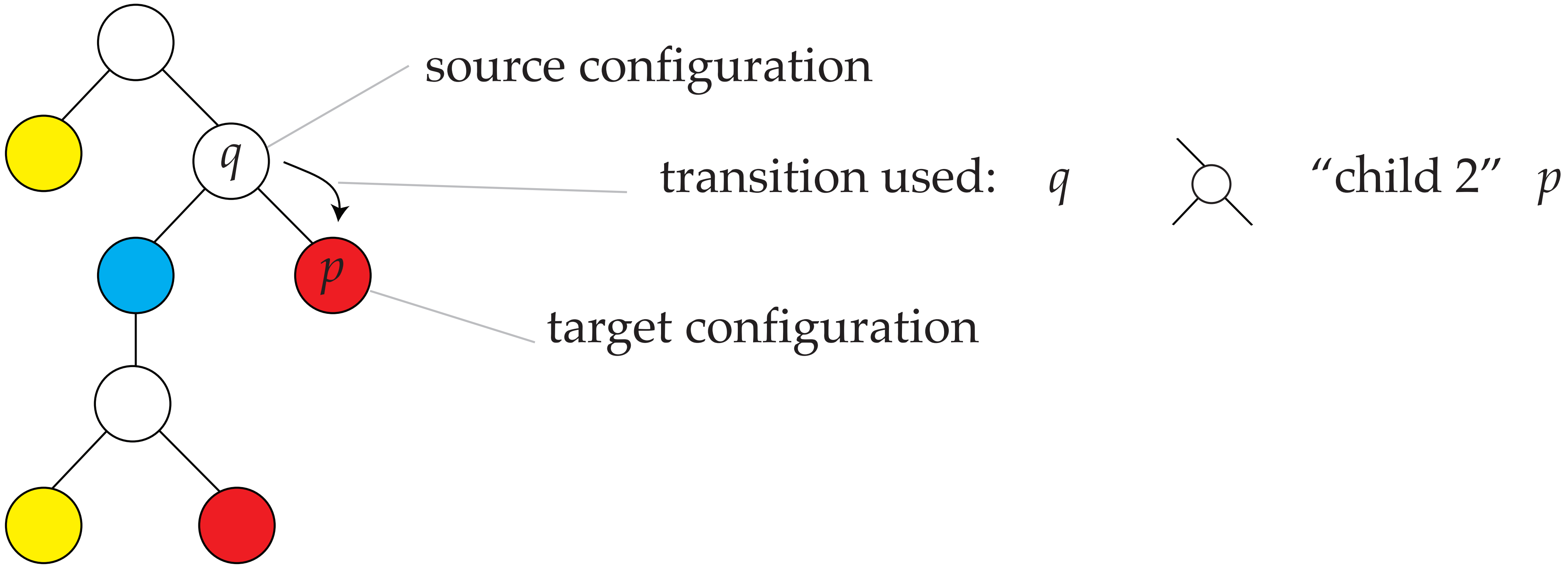


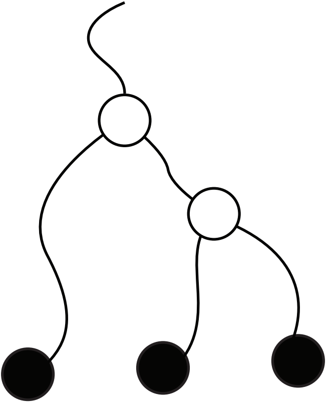

s
leaf port i of s
t

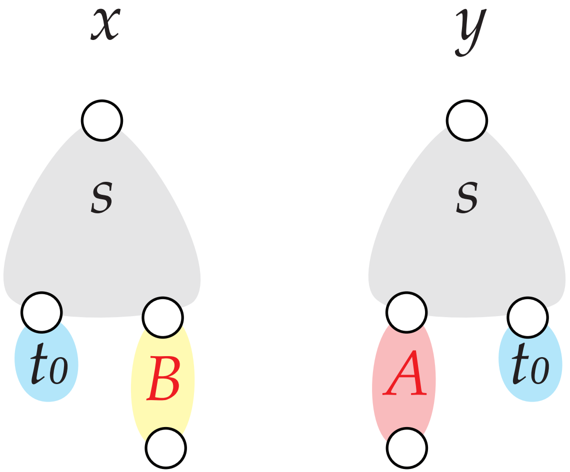

s
equivalent to s
the run will stay here until it accepts / loops / rejects
run begins here in state q
t
equivalent to t

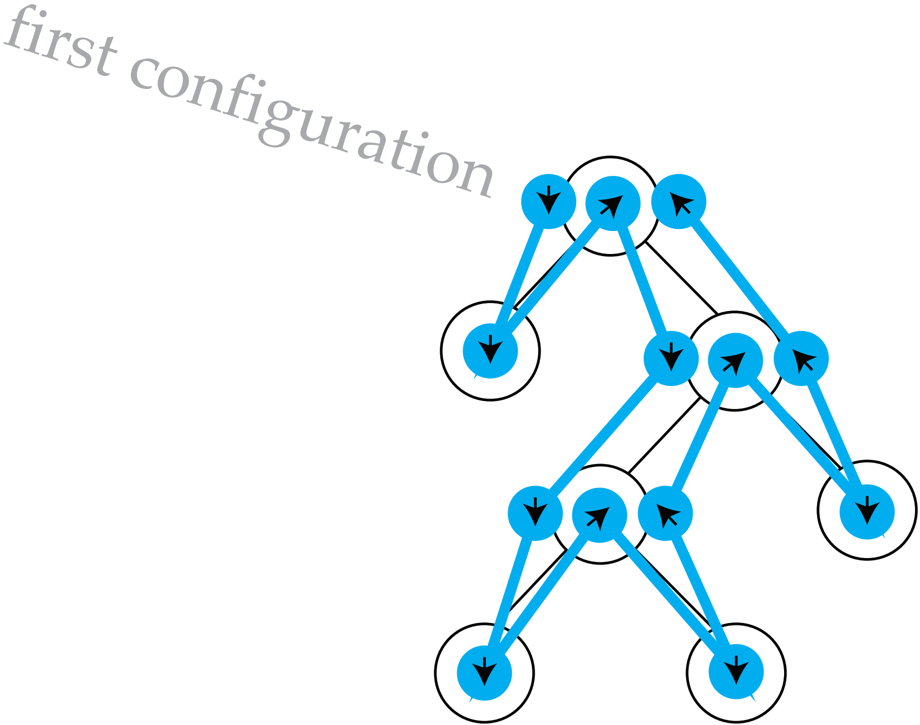

s
equivalent to t_1
(1) the yellow node is visited in state $\overline{q}$, and then one of the red nodes is visited.
run begins here in state q
(2) the yellow node is last visited in state $\overline{q}$, and then one of the ports ◯ is visited
t
equivalent to t_1

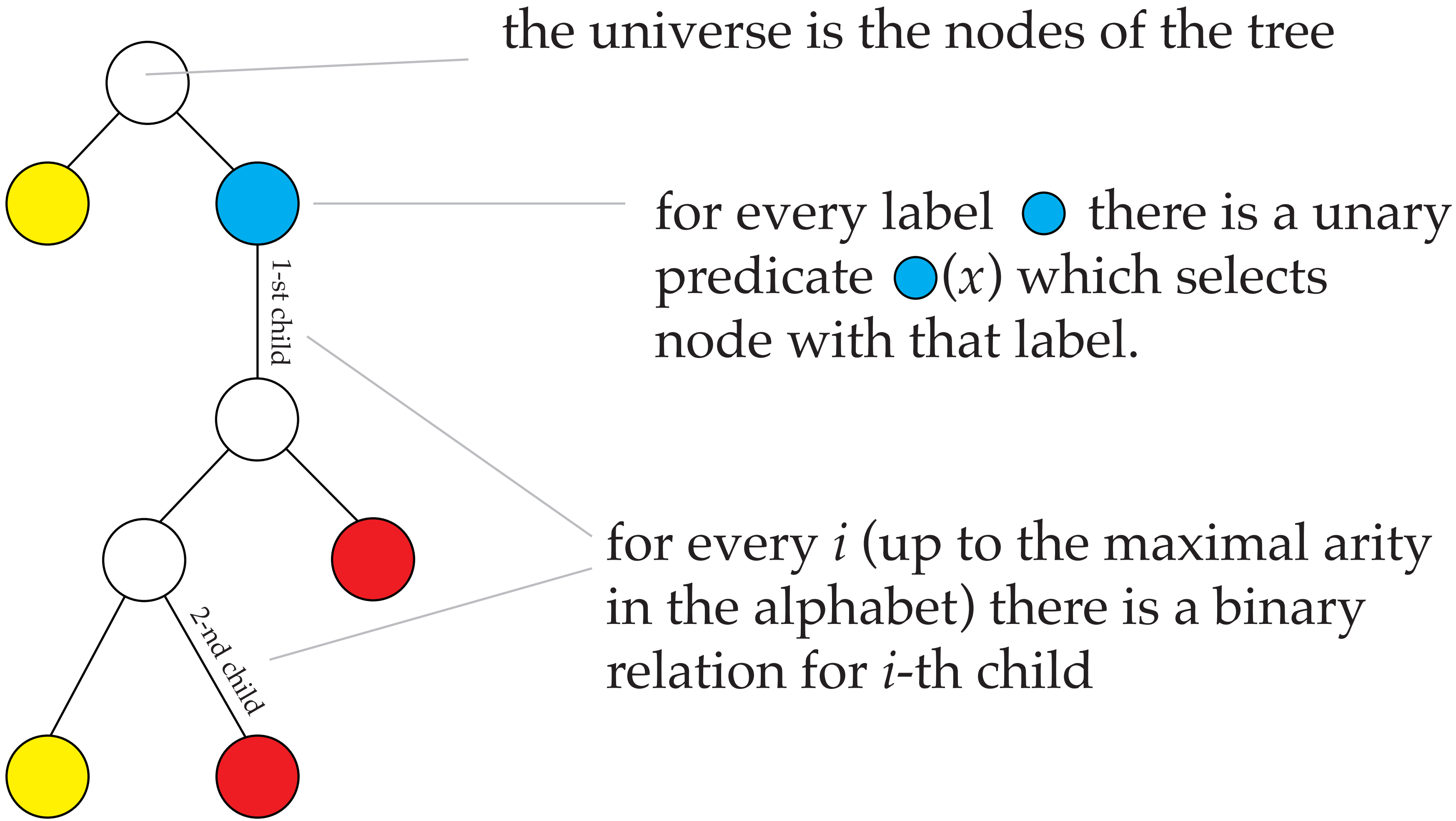

pattern without root port
and without black leaves
pattern with exactly
one black leaf

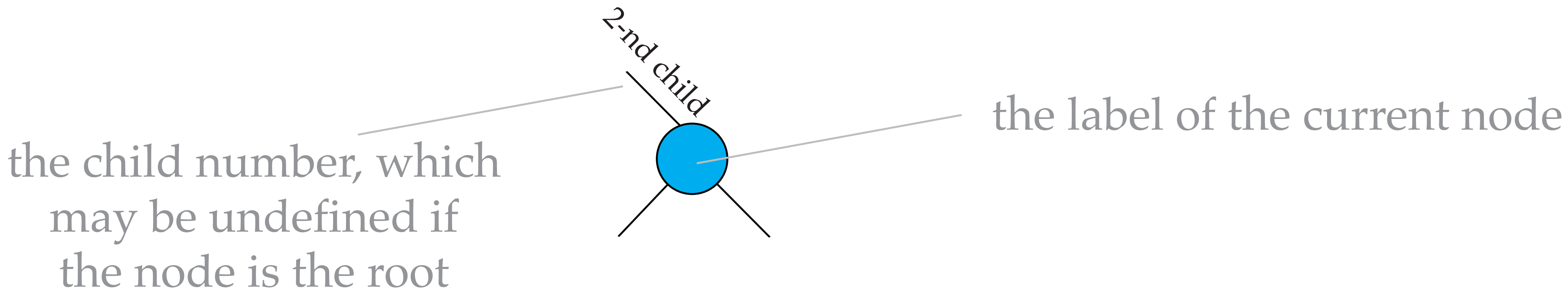

q
p
source configuration
transition used: q “child 2” p
target configuration




x
y
s
s
t0
B
A
t0

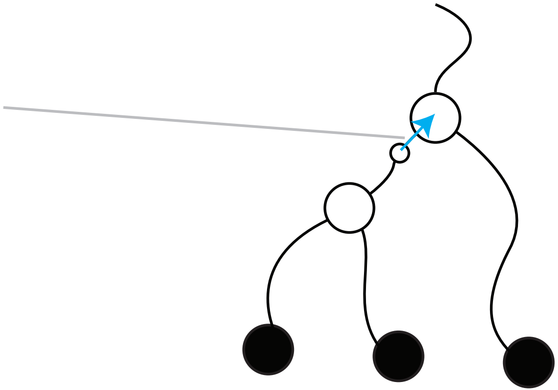

first configuration


the universe is the nodes of the tree
for every label there is a unary predicate (x) which selects node with that label.
1-st child
for every i (up to the maximal arity in the alphabet) there is a binary relation for i-th child
2-nd child


2-nd child
the label of the current node
the child number, which
may be undefined if
the node is the root

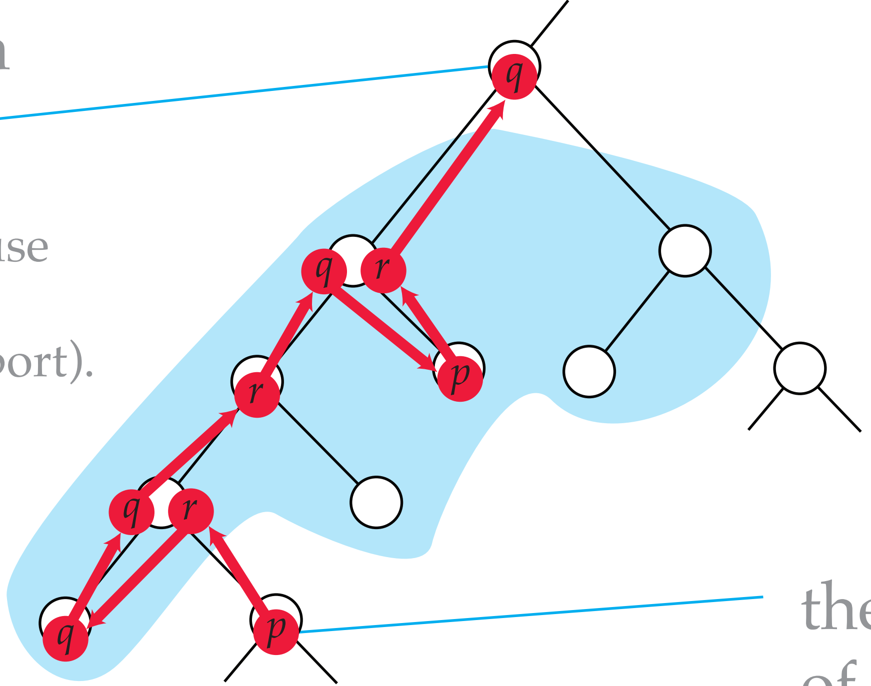
$$\delta \subseteq \overset{\text{source state}}{Q} \times \underbrace{\text{local views over } \Sigma}_{\text{current local view}} \times \overset{\text{target state}}{Q} \times \{\overbrace{\text{parent, stay, child 1, ..., child } \underset{\text{maximal arity in } \Sigma}{n}}^{\text{where to move}}\}.$$

first move of the
depth-first search



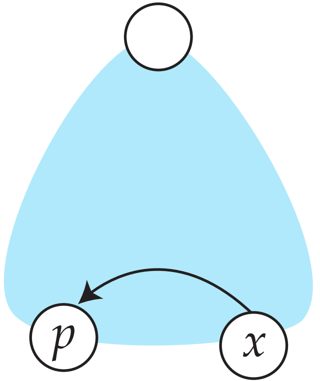
the run is cut off as soon as it visits a port

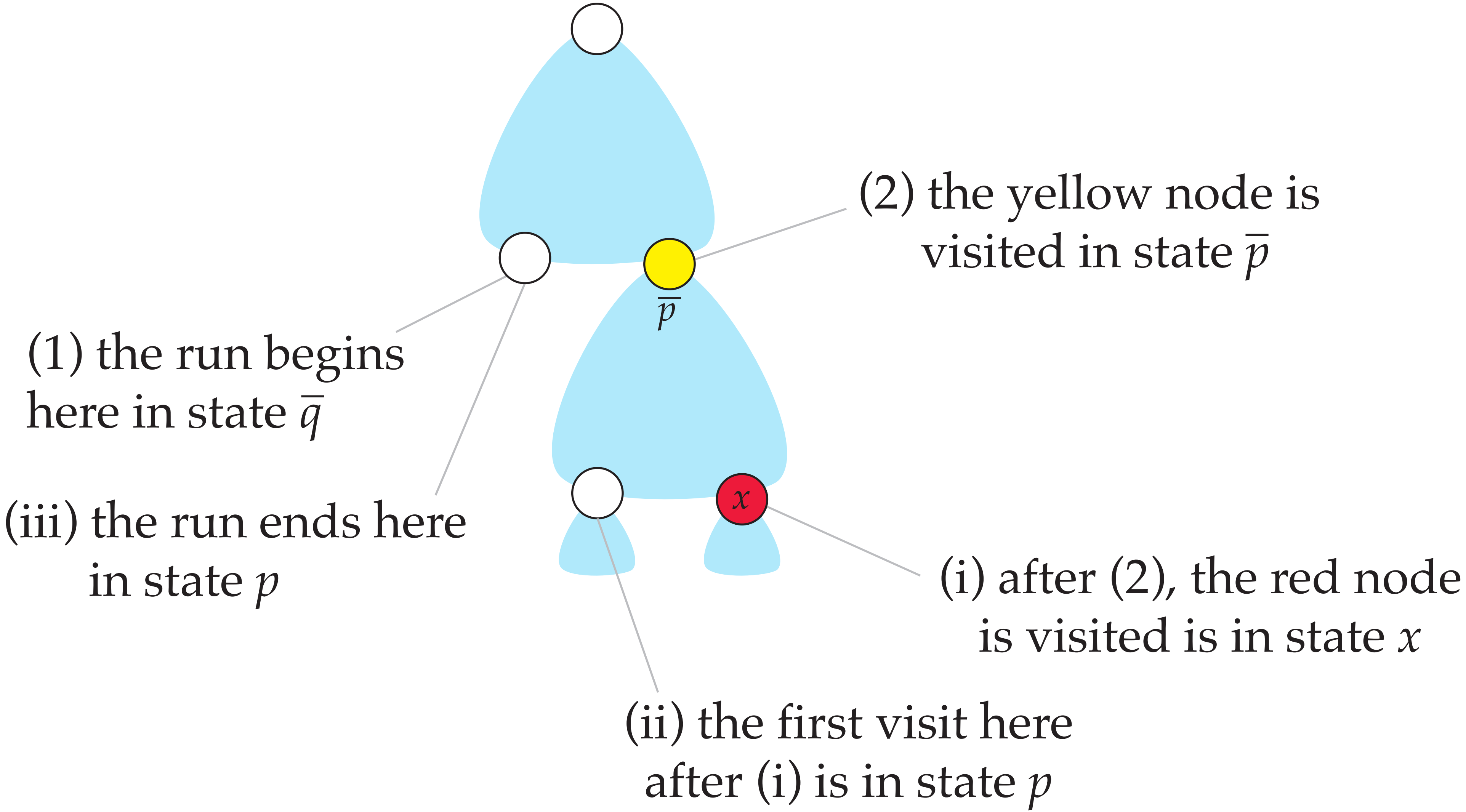
(this might never happen, because the run might accept, reject or loop before visiting another port).



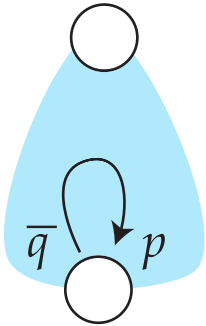
the automaton begins in one of the ports

the behaviour in the ports is well-defined, because a pattern includes the local view of each port


p
x



(2) the yellow node is
visited in state $\overline{p}$
$\overline{p}$
(1) the run begins
here in state $\overline{q}$
x
(iii) the run ends here
in state p
(i) after (2), the red node
is visited is in state x
(ii) the first visit here
after (i) is in state p



and therefore also