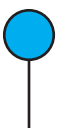
a ranked alphabet

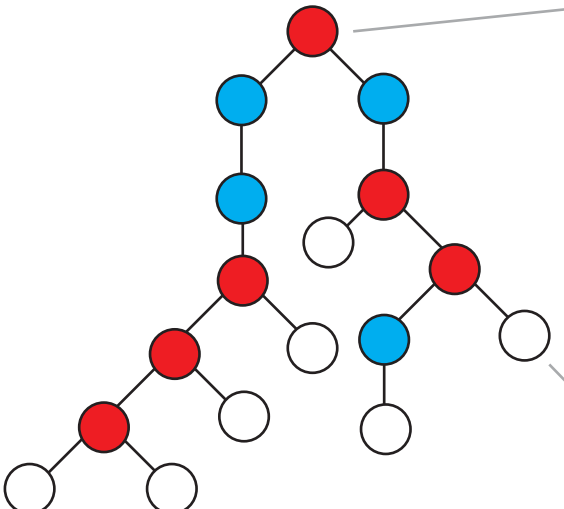
arity 2

arity 1



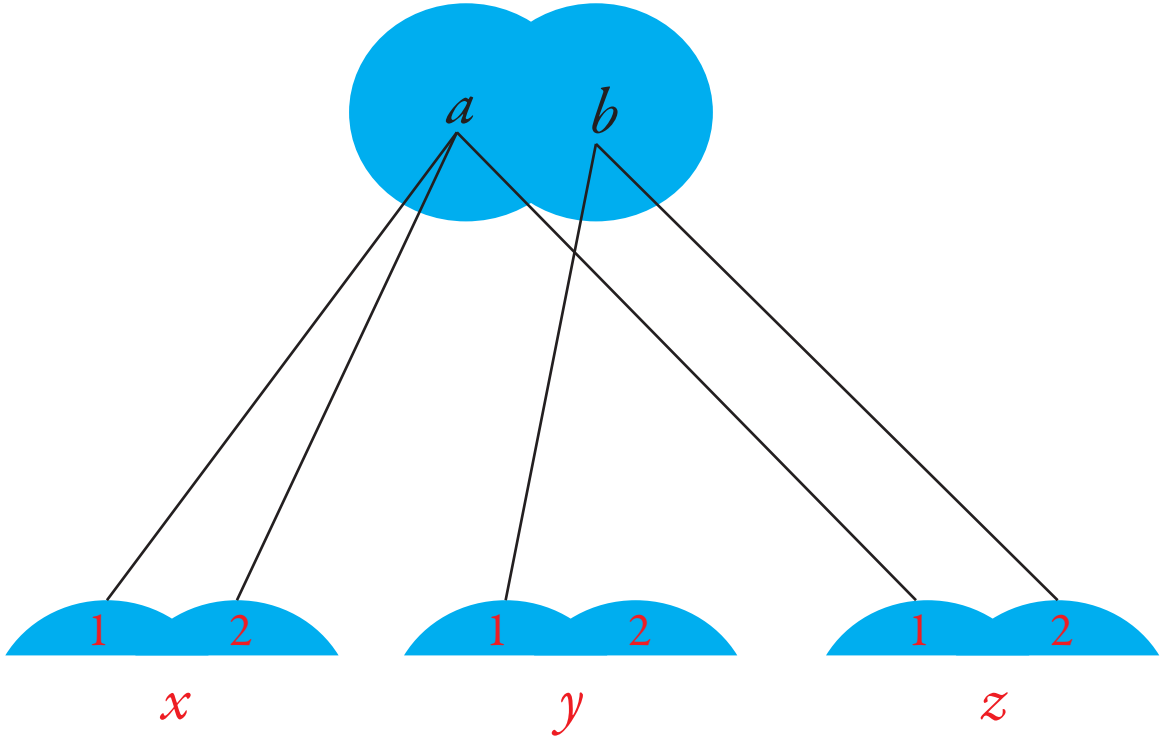
arity 0

a tree

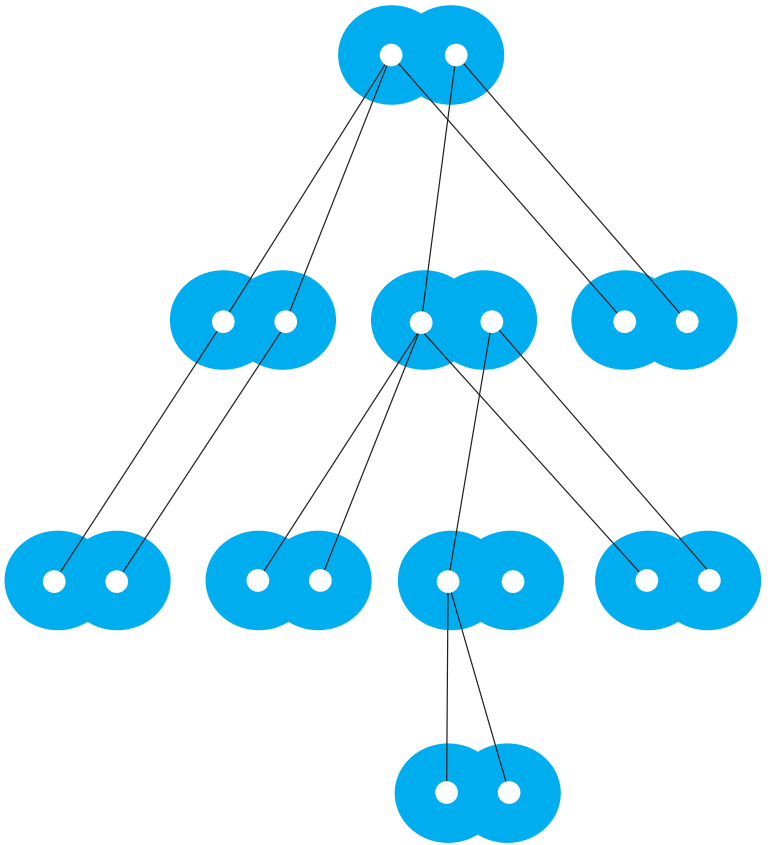


this node has a label of arity 2, and therefore it has 2 children

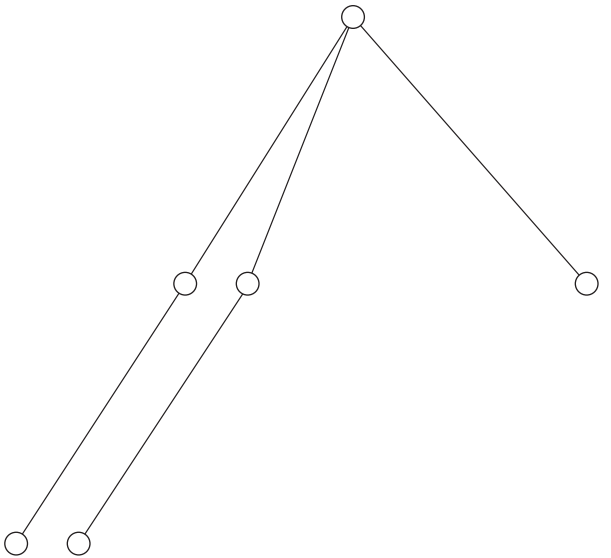
this node is child 2 (children are ordered)


a
b
1
2
1
2
1
2
x
y
z

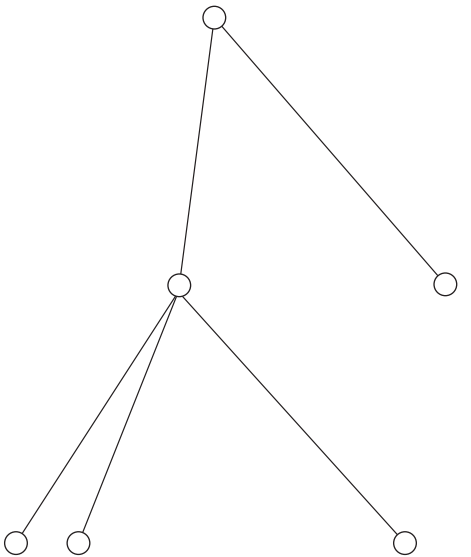
A tree t over $\Sigma^{[2]}$

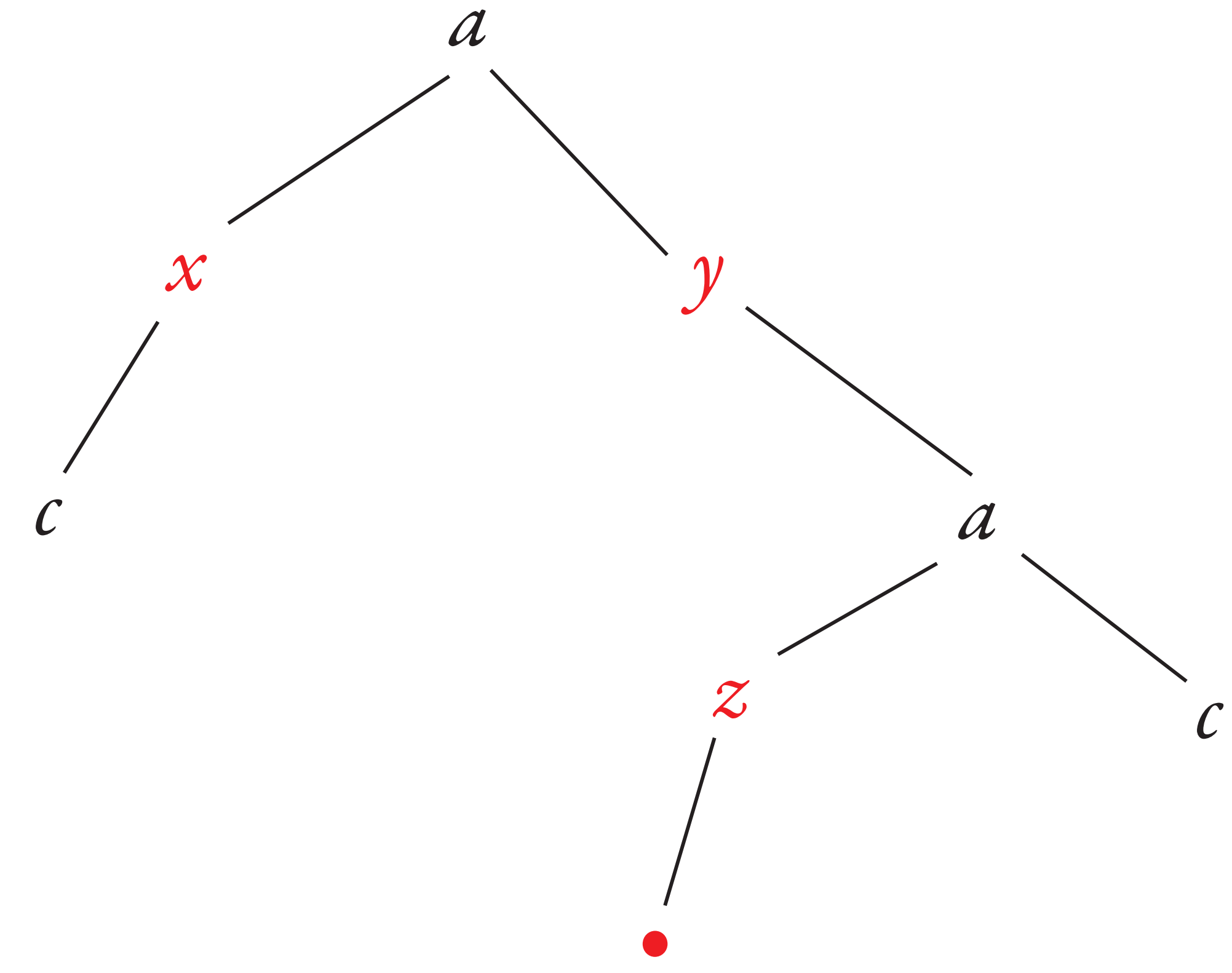


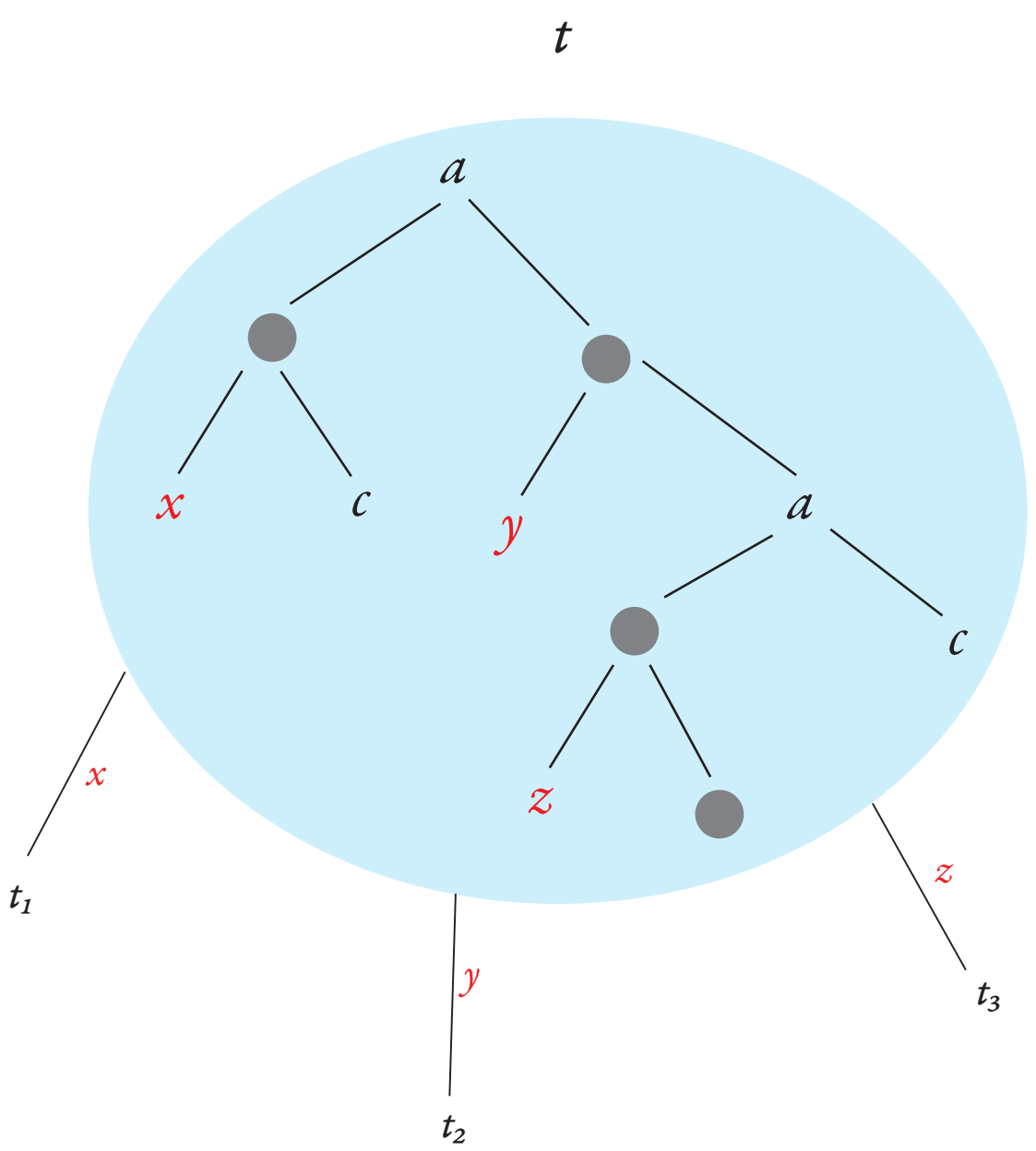
$\mathsf{unfold}_1(t)$

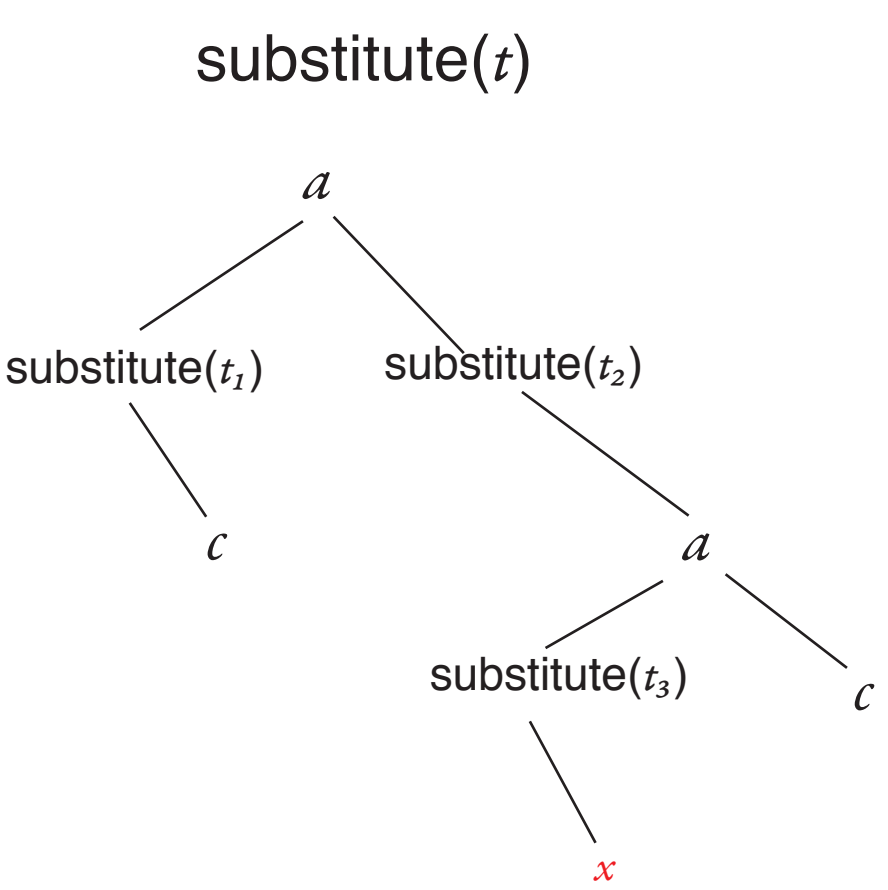


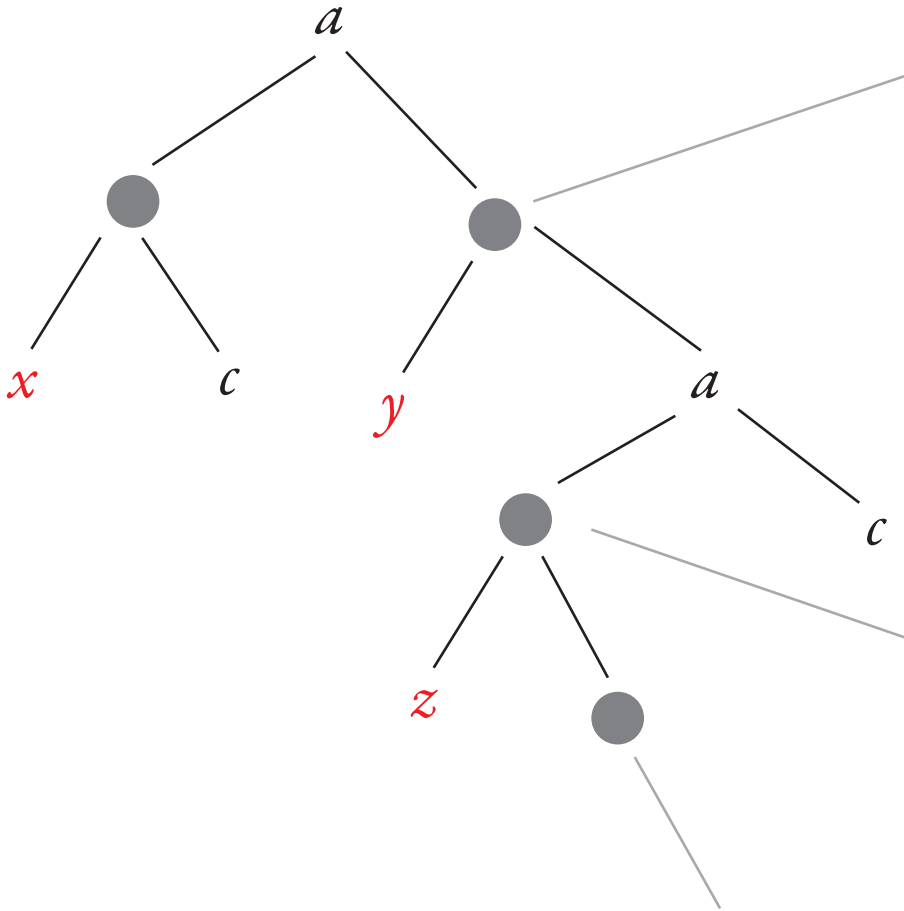
$\mathsf{unfold}_2(t)$




a
x
y
c
a
z
c


t
a
x
c
y
a
c
z
x
t1
y
t2
z
t3

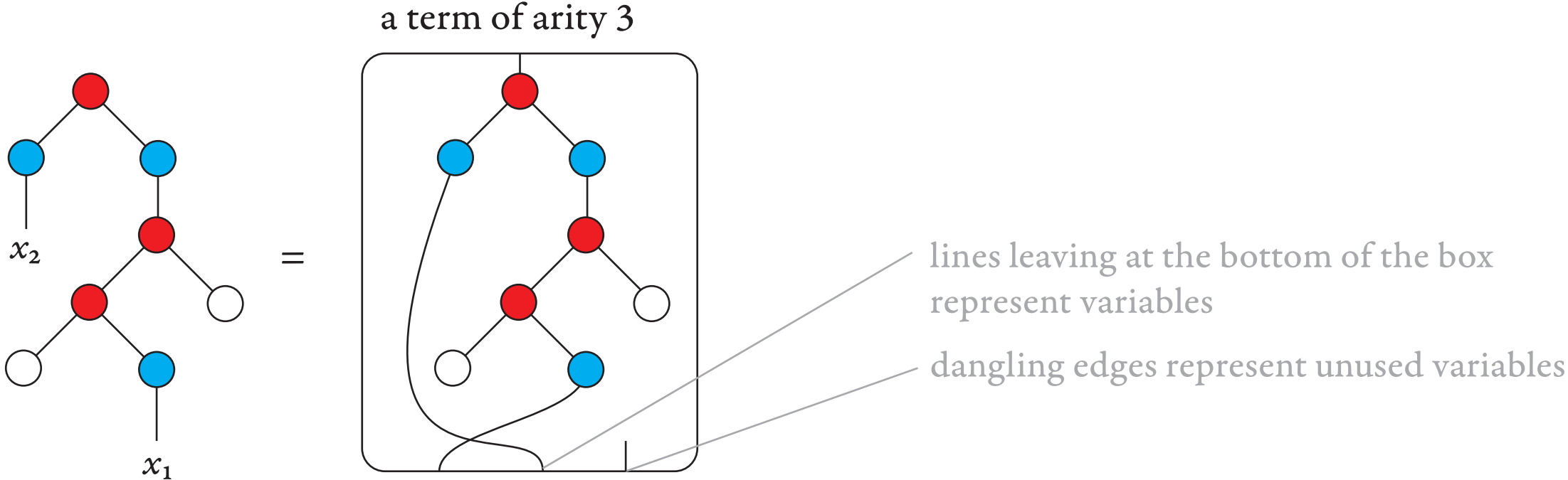

substitute(t)
a
substitute(t1)
substitute(t2)
c
a
substitute(t3)
c
x

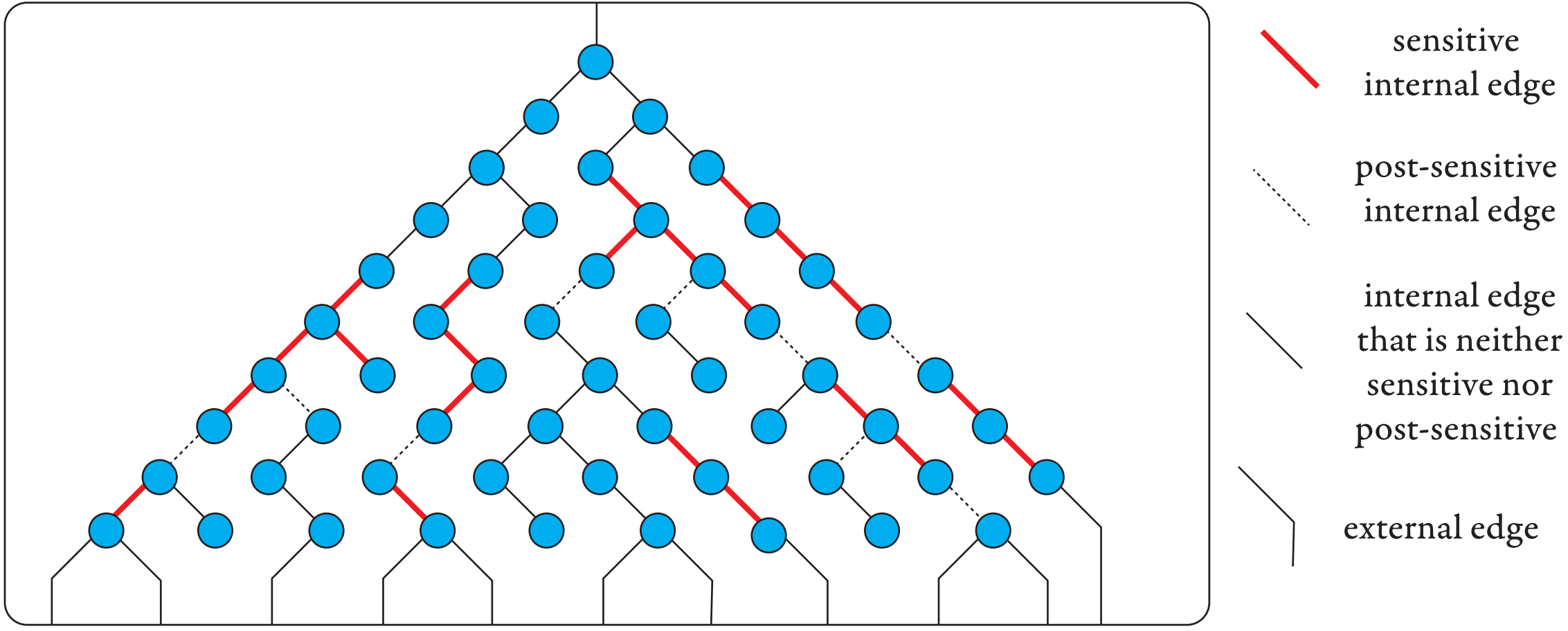


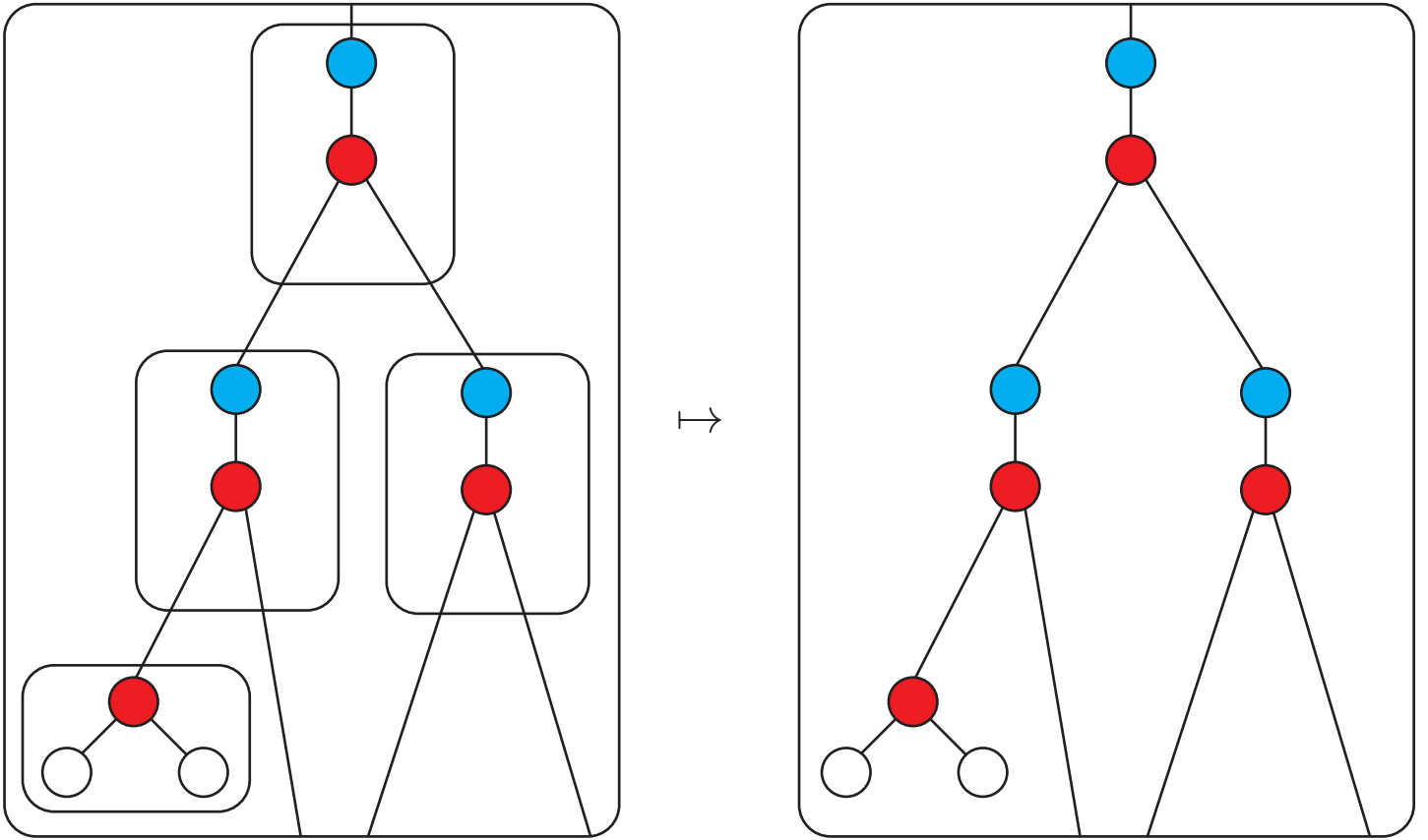
(c) for each binary letter from 1, its left child is a variable

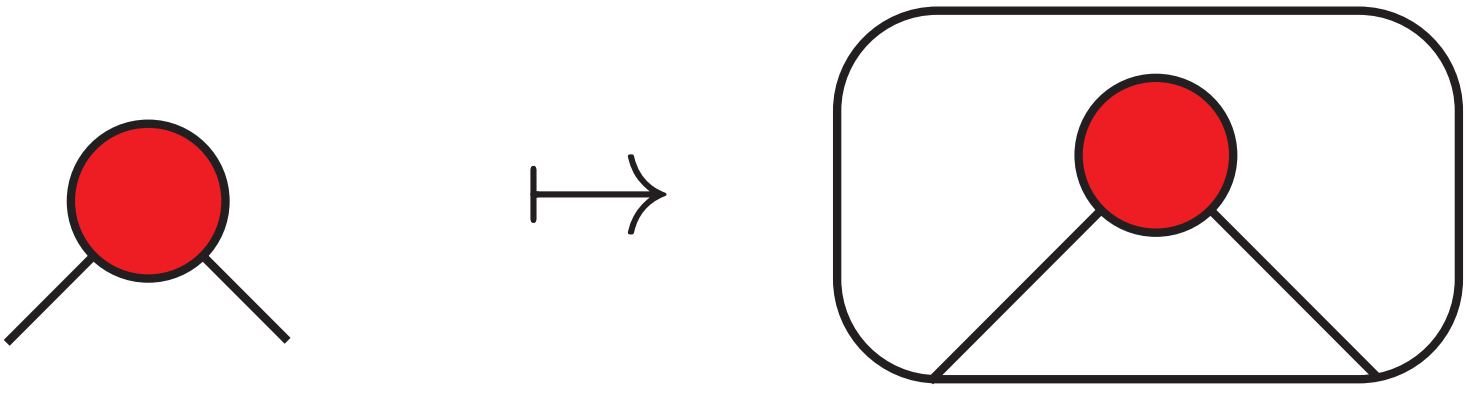
(a) only binary and nullary letters from 1 are allowed

(b) a nullary letter from 1 is allowed exactly once

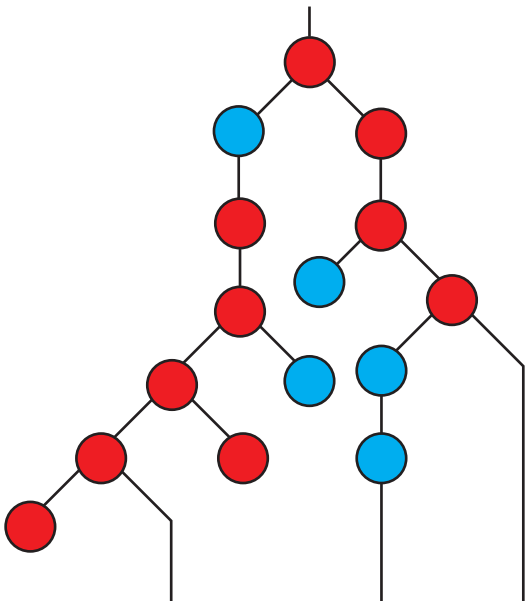

a term of arity 3
x_2
x_1
=
lines leaving at the bottom of the box represent variables
dangling edges represent unused variables


sensitive
internal edge
post-sensitive
internal edge
internal edge
that is neither
sensitive nor
post-sensitive
external edge

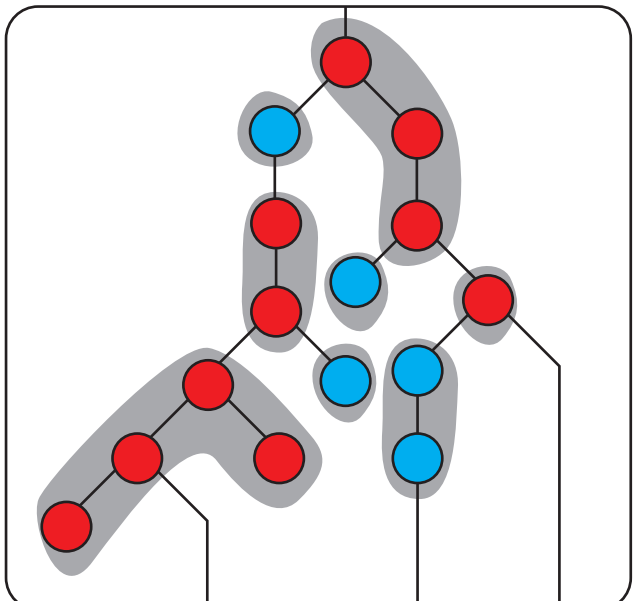




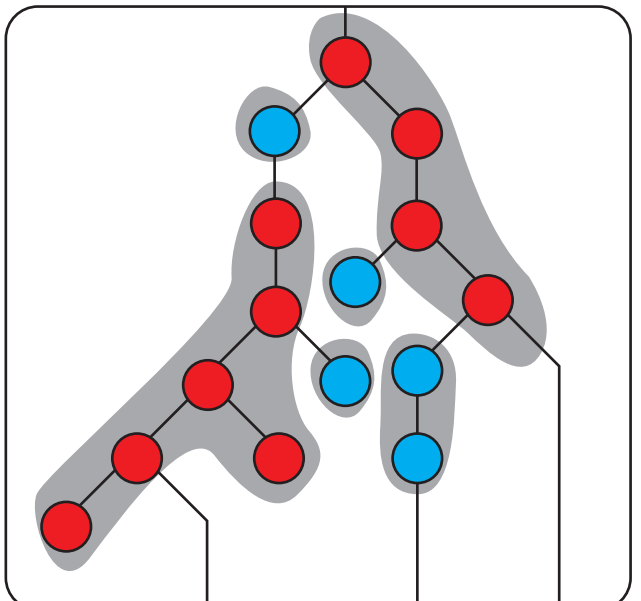
a term

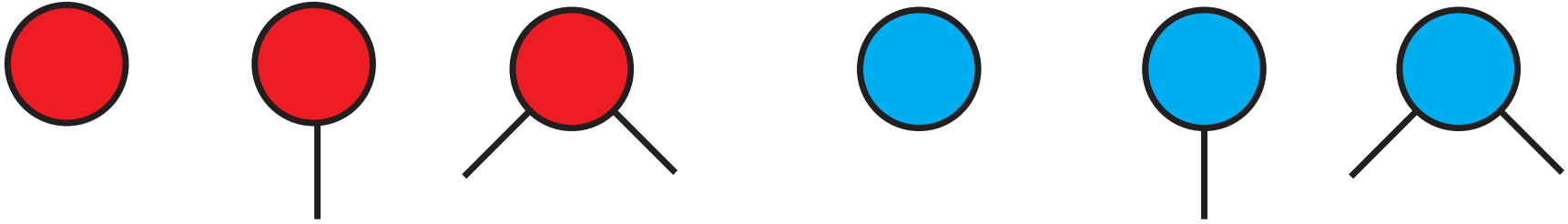


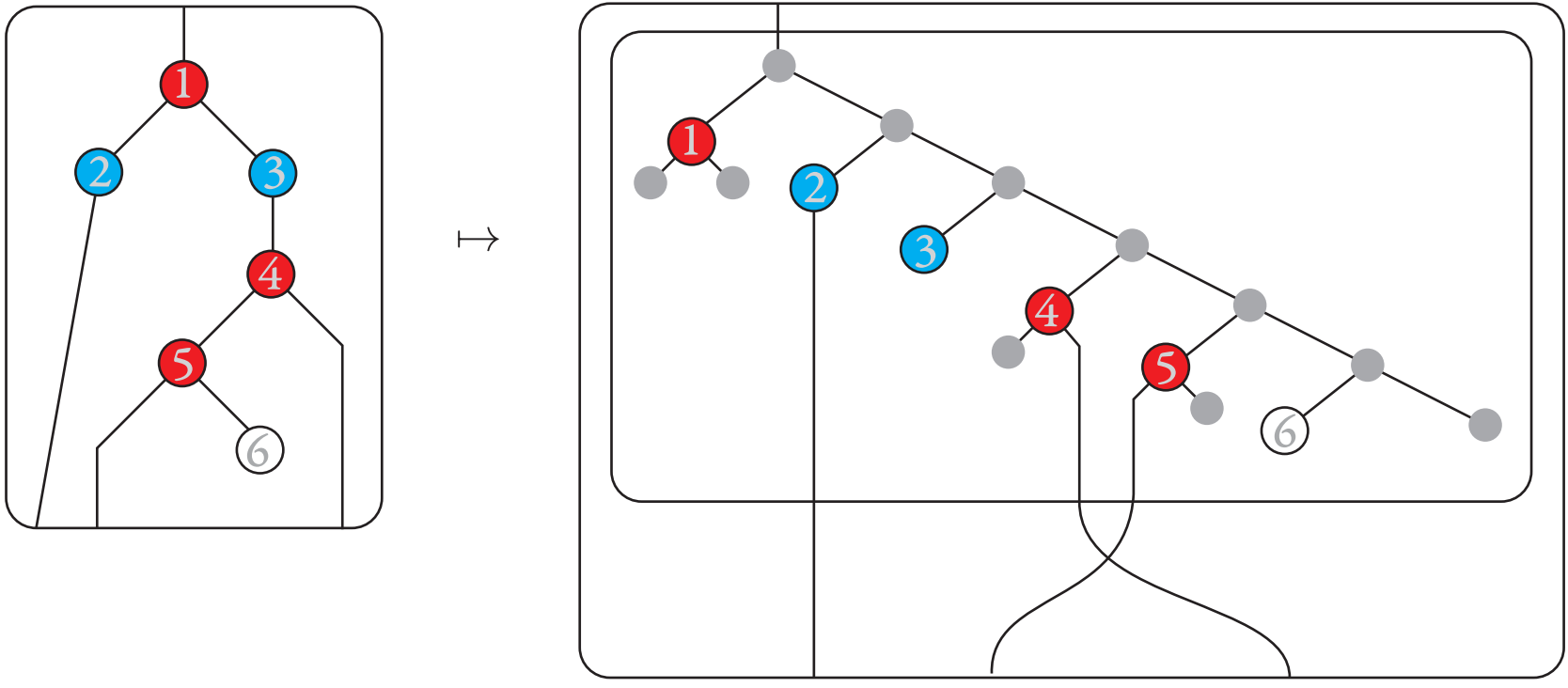
ancestor equivalence



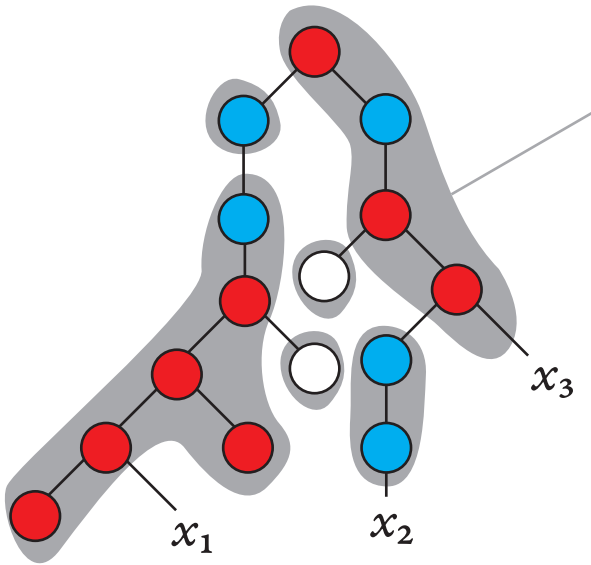
descendant equivalence

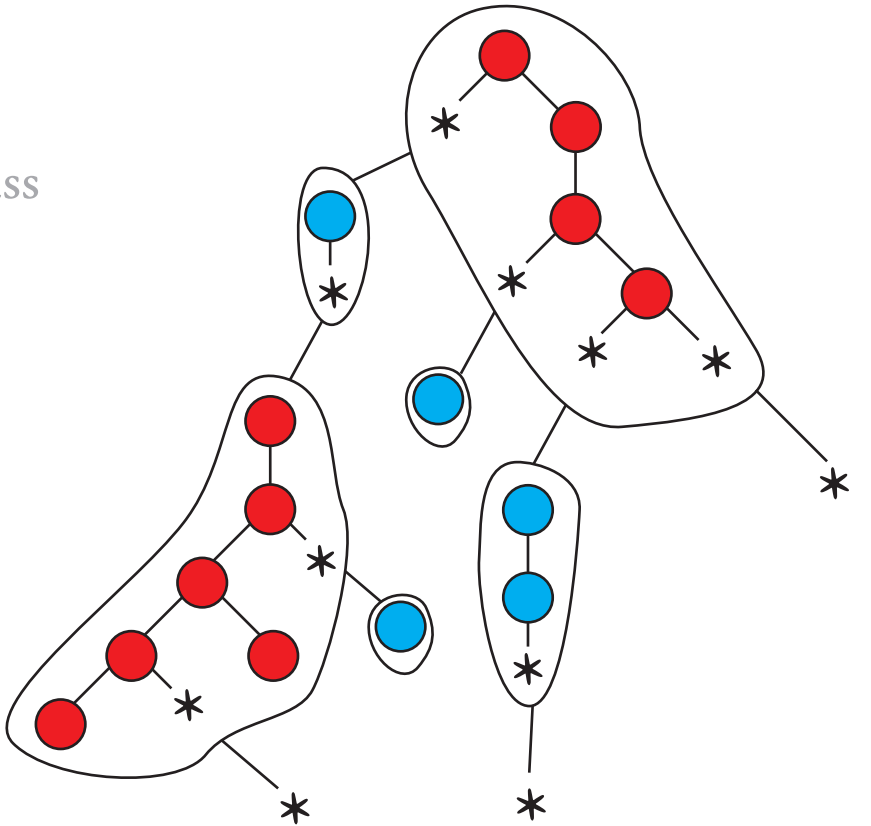




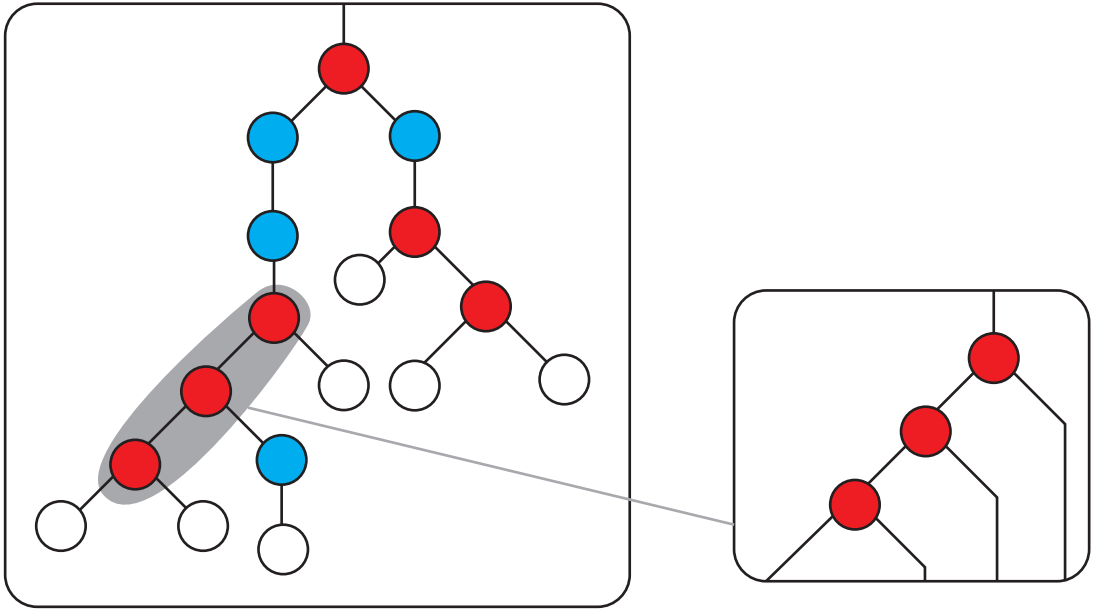

1
2
3
4
5
6
↦
1
2
3
4
5
6

a factorisation equivalence

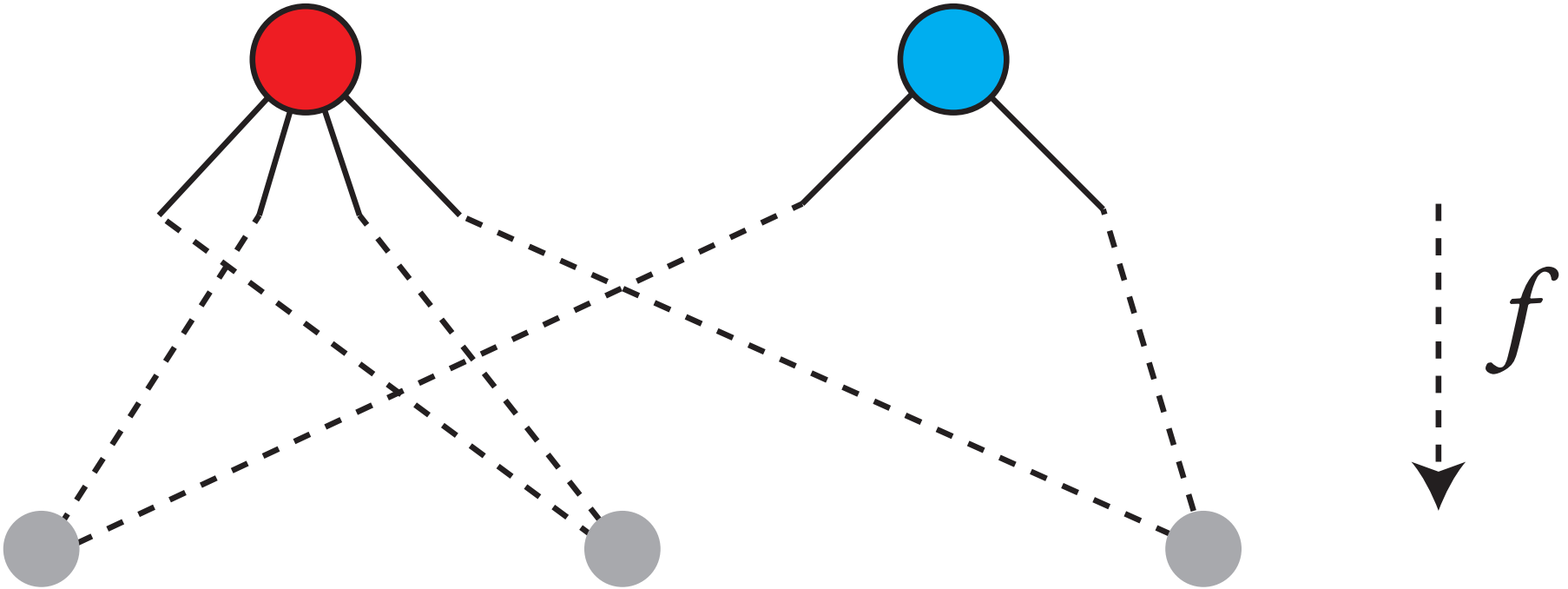


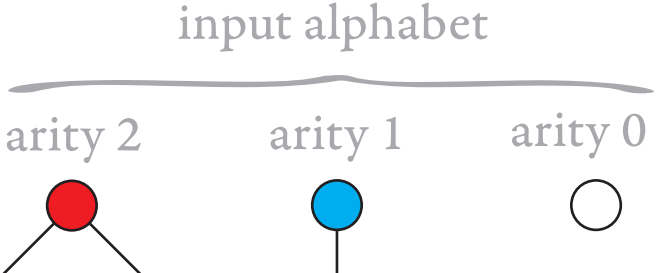


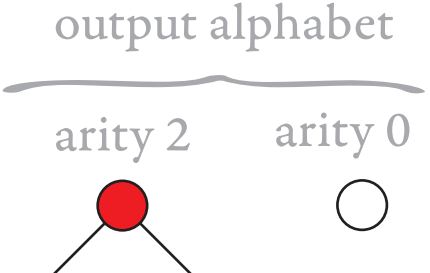
a tree

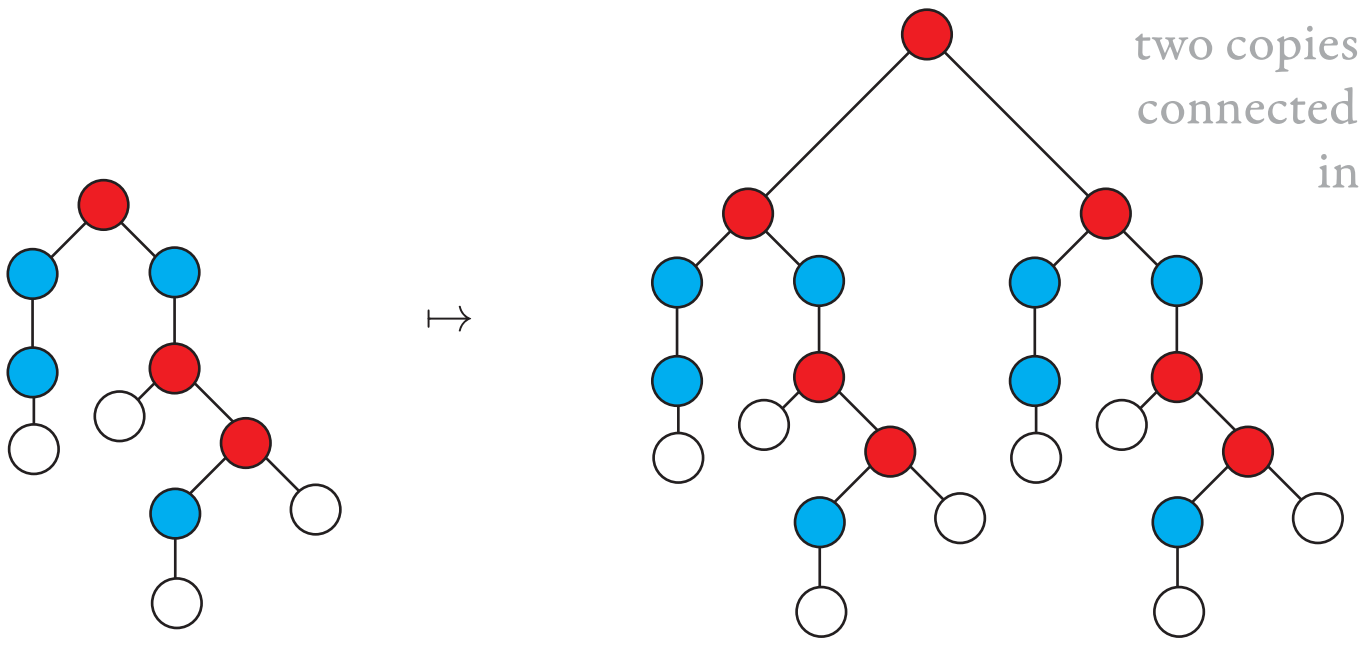


a term with 4 ports that represents part of the tree

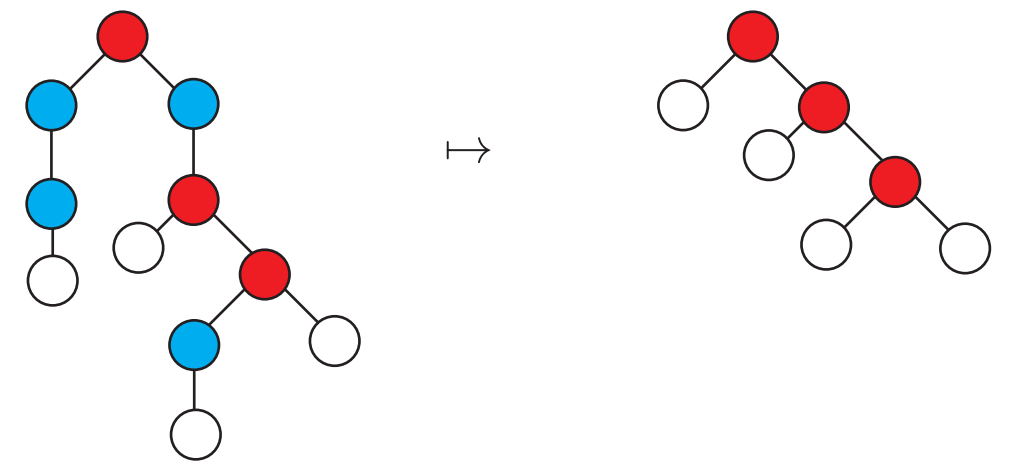

f

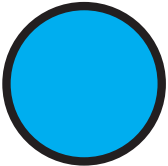

input alphabet
arity 2
arity 1
arity 0

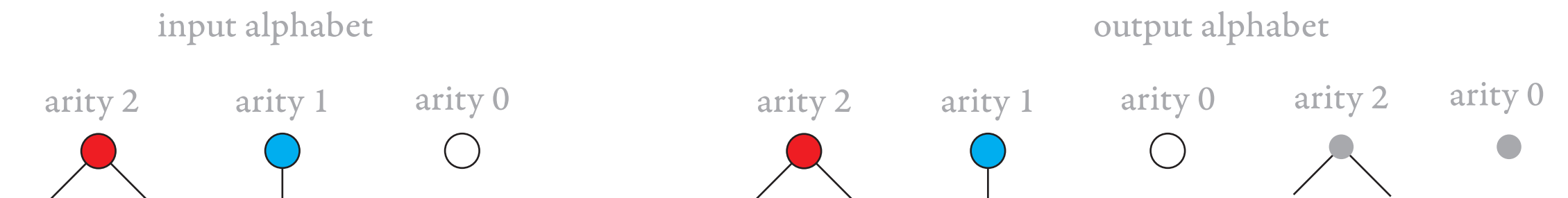

output alphabet
arity 2
arity 0

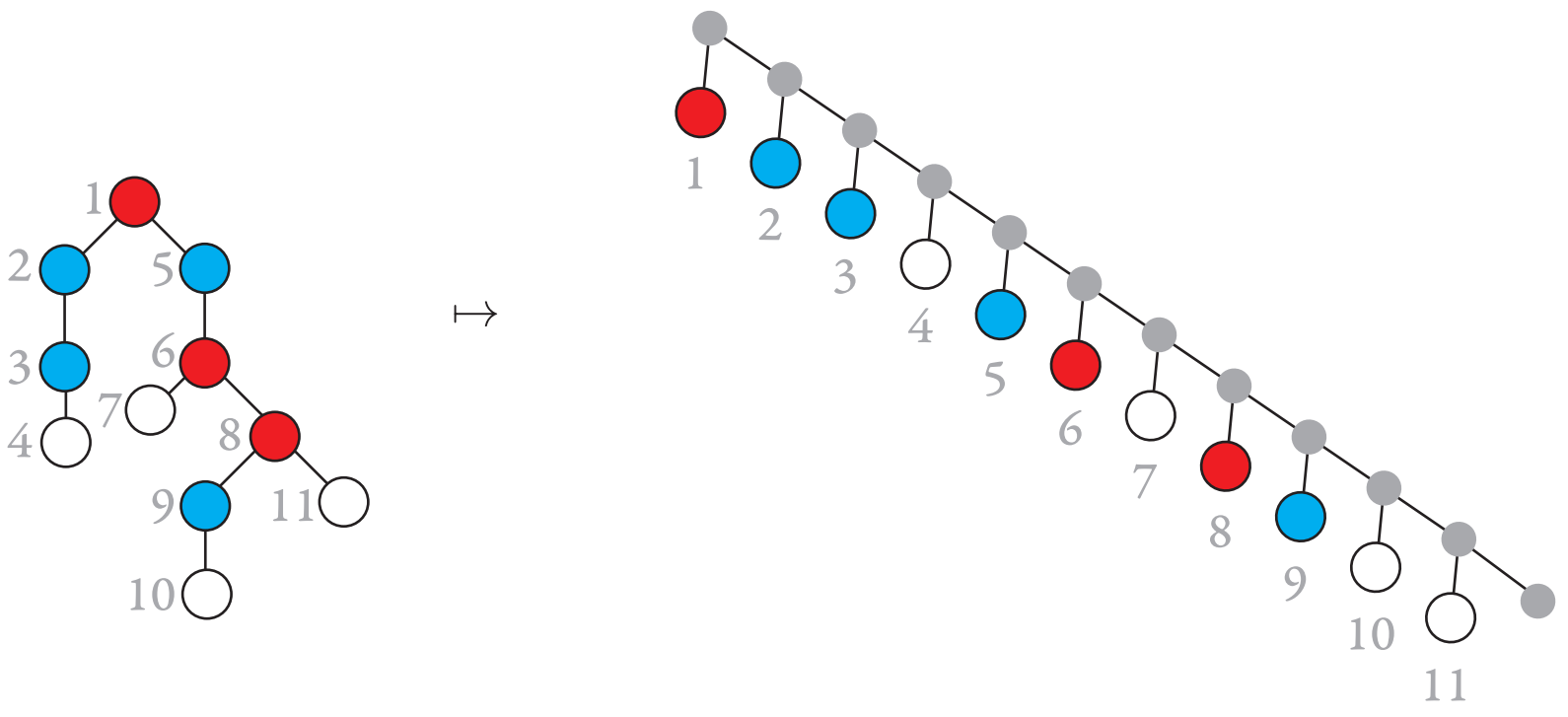


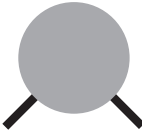
two copies of the input tree,
connected by a binary node
in the root



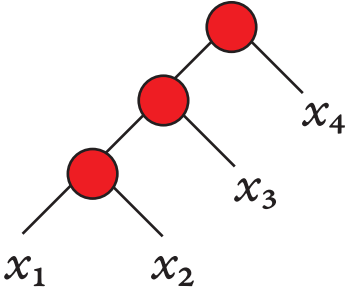



input alphabet
arity 2
arity 1
arity 0
output alphabet
arity 2
arity 1
arity 0
arity 2
arity 0

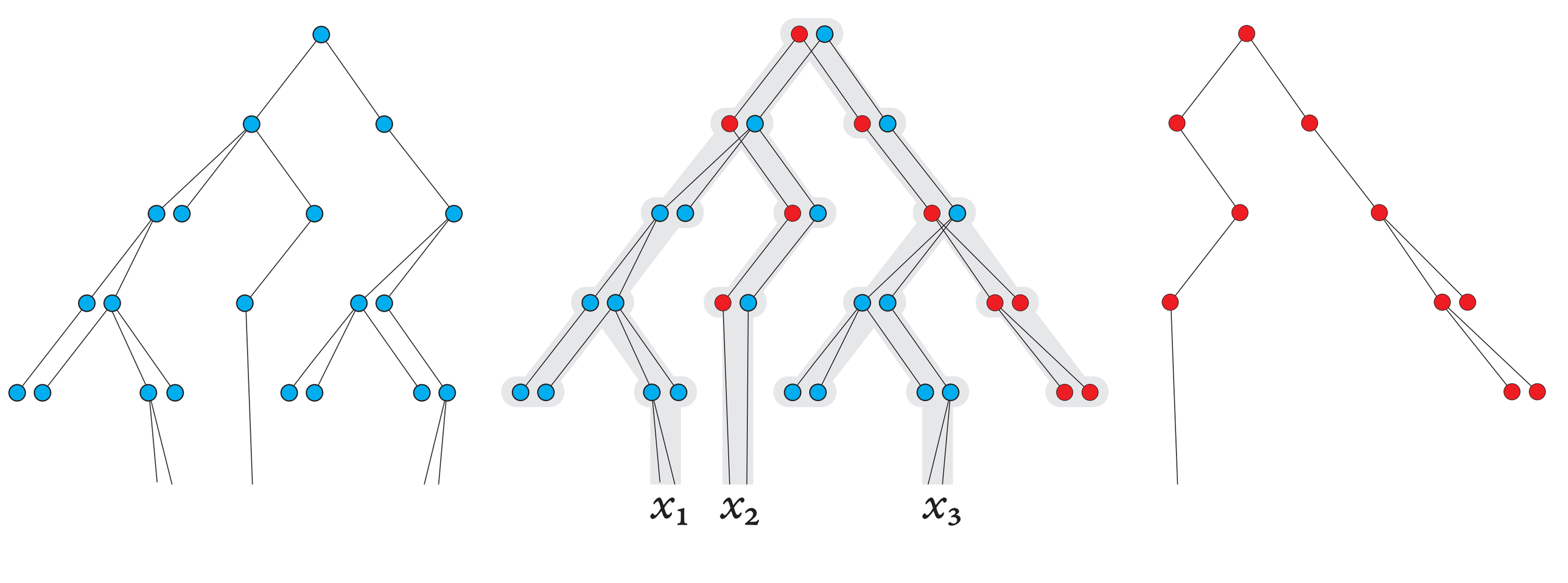

1
2
5
3
6
4
7
8
9
11
10
↦
1
2
3
4
5
6
7
8
9
10
11

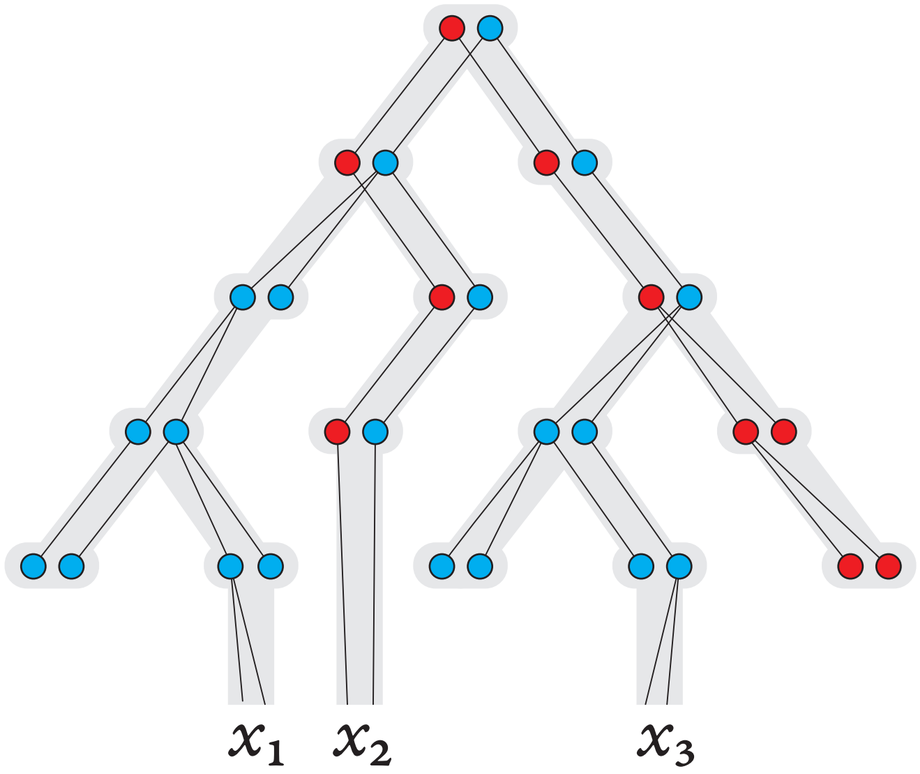


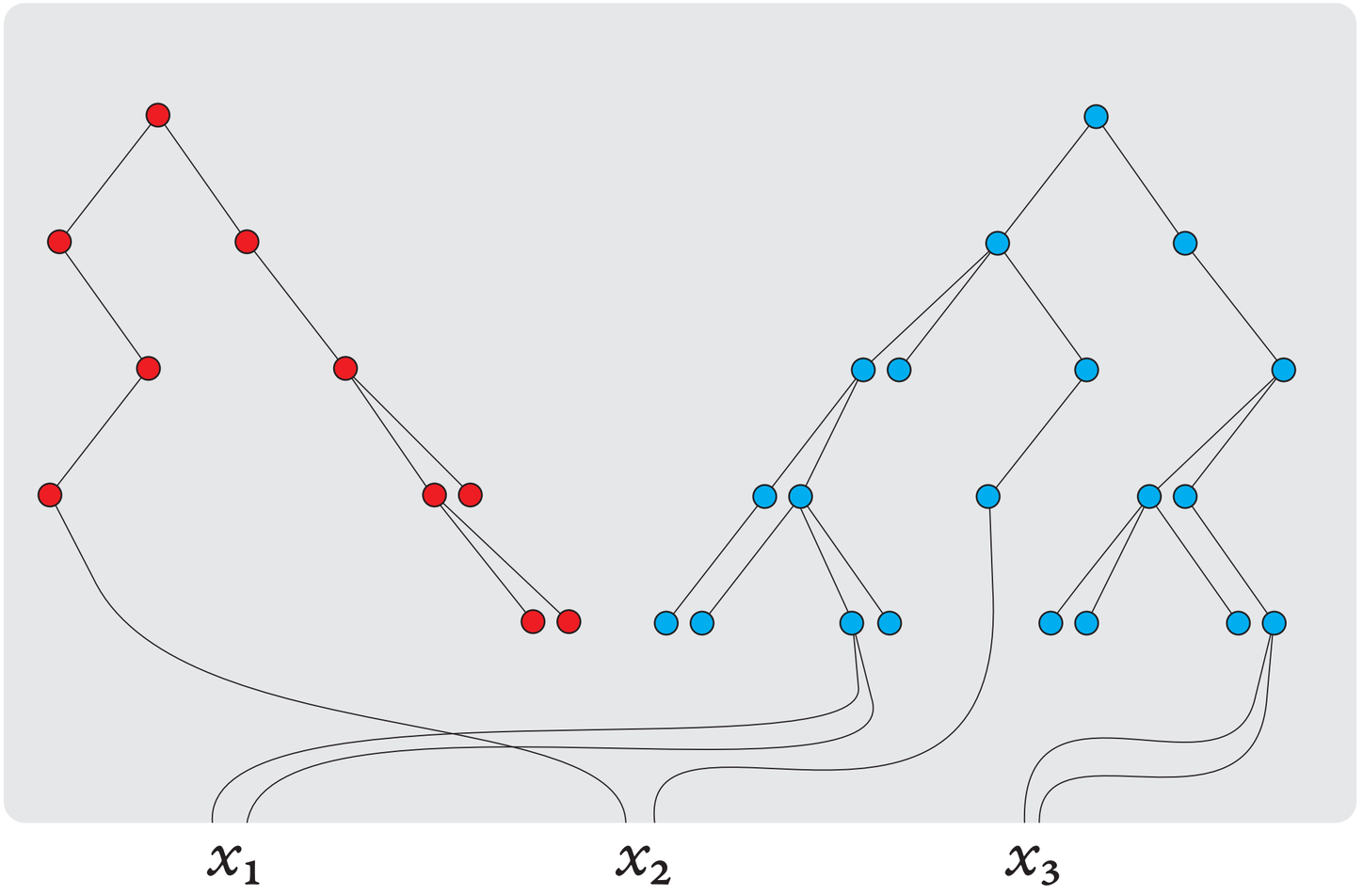
a term of arity 4

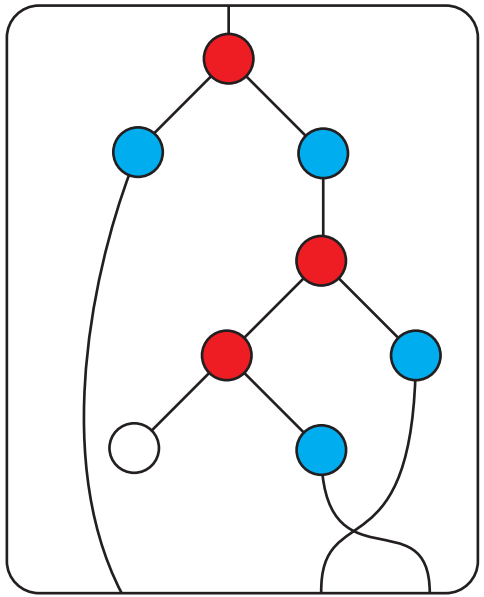


a term of arity 0


x_1
x_2
x_3


x_1
x_2
x_3

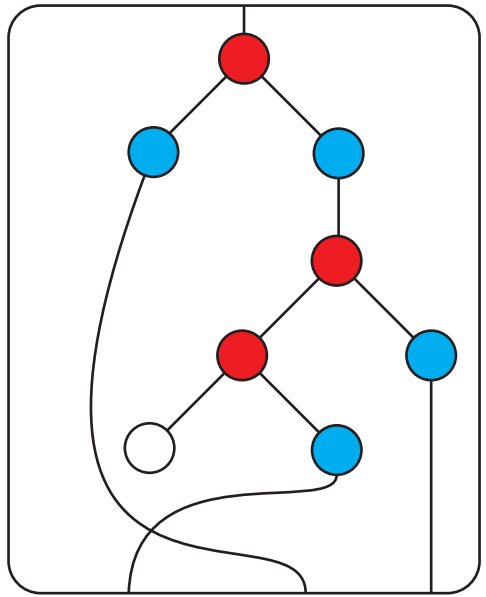

x_1
x_2
x_3



satisfies (*)

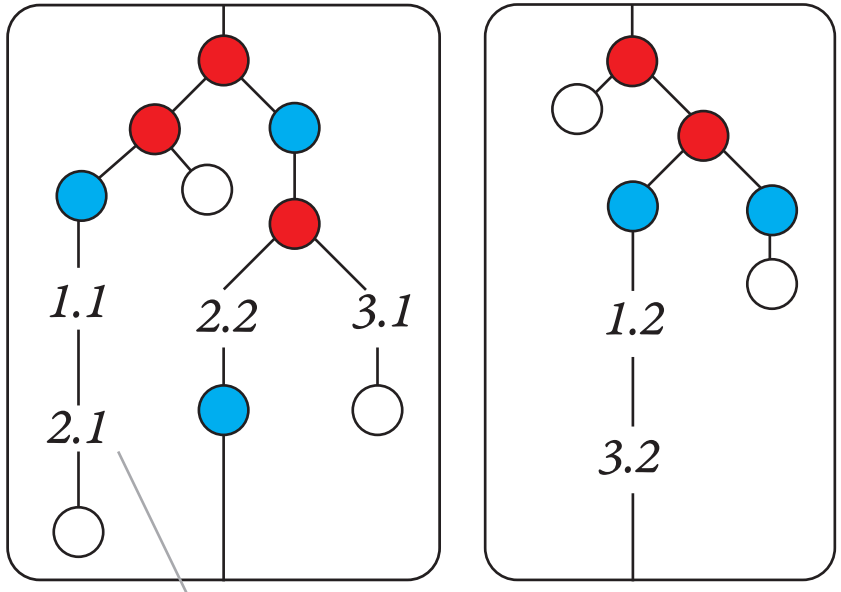
(*)

If the root has arity n,
and $1 \leq i < j \leq n$, then
all ports of the j-th
subterm of the root are
after all ports of the
i-th subterm of the root

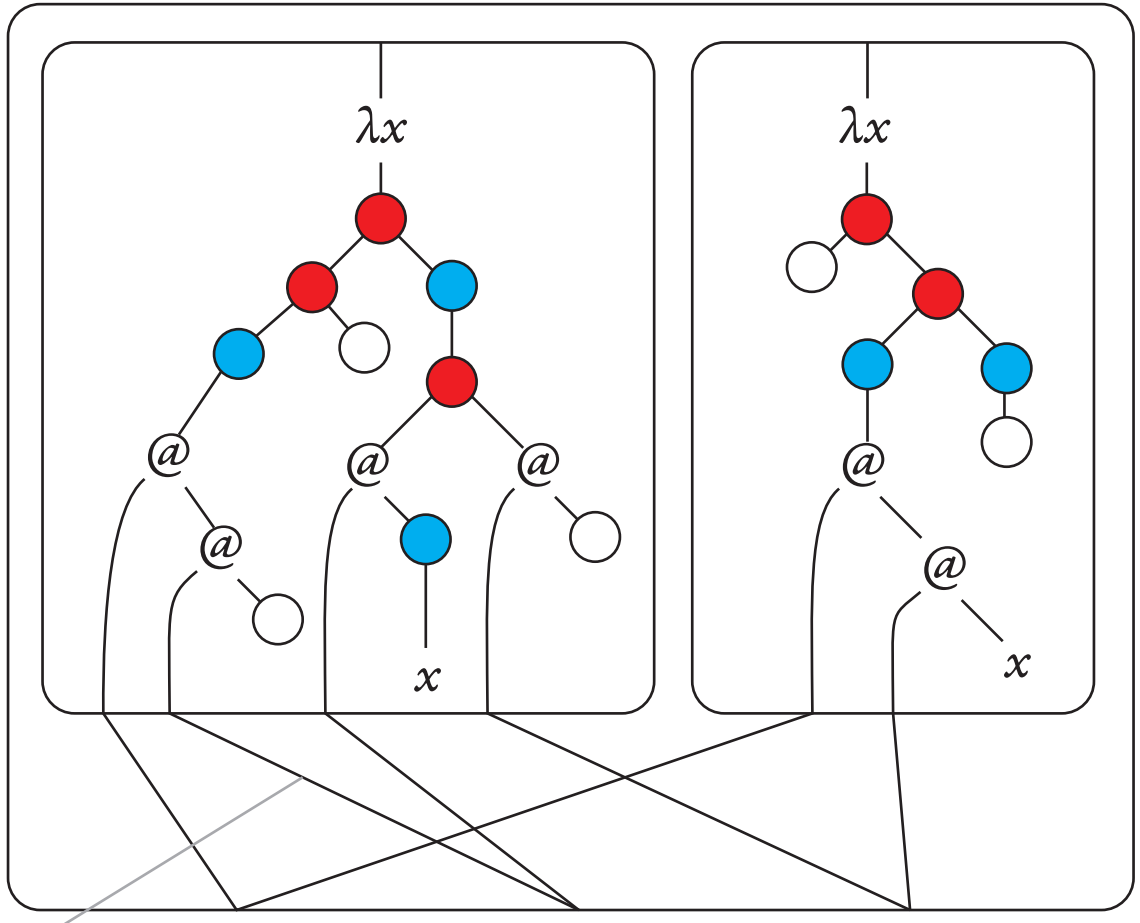


violates (*)

a register update

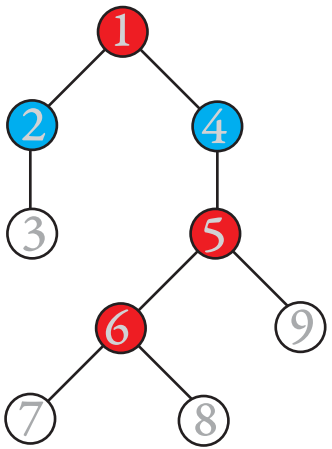


its dual

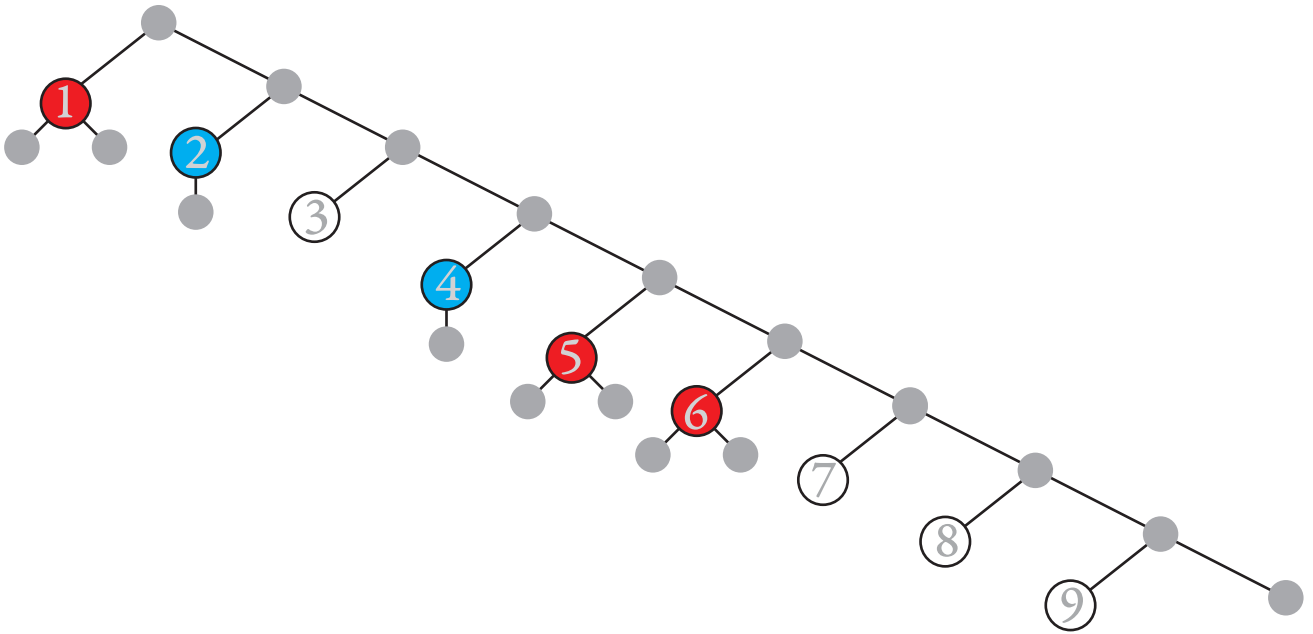


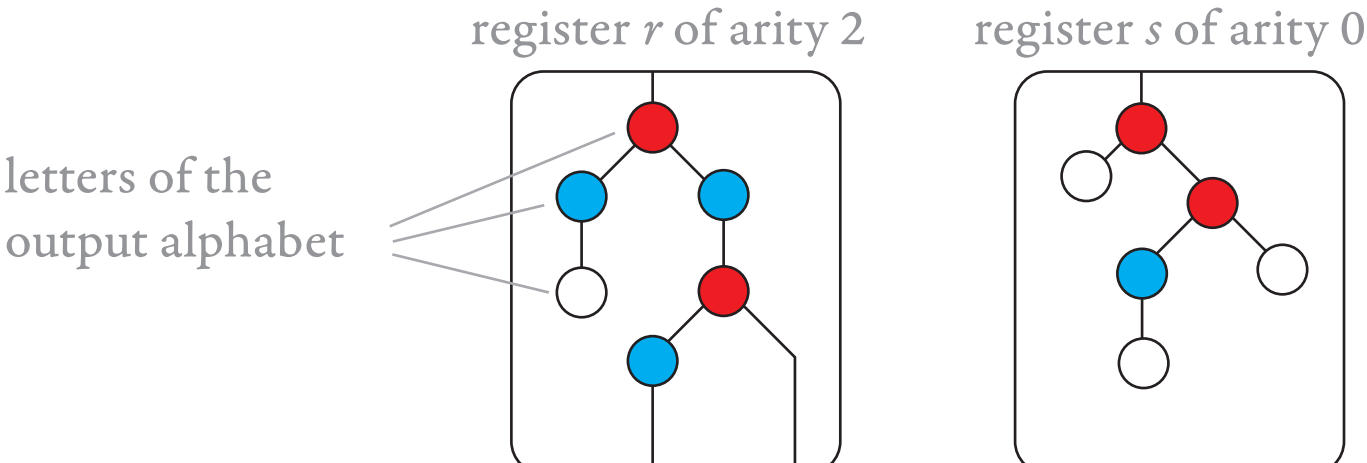
Variable *i.j* represents register *i* in the *j*-th argument of the reigster update.
In the dual, this variable is mapped to the *i*-th edge which enters the *j*-th port of the reducer.

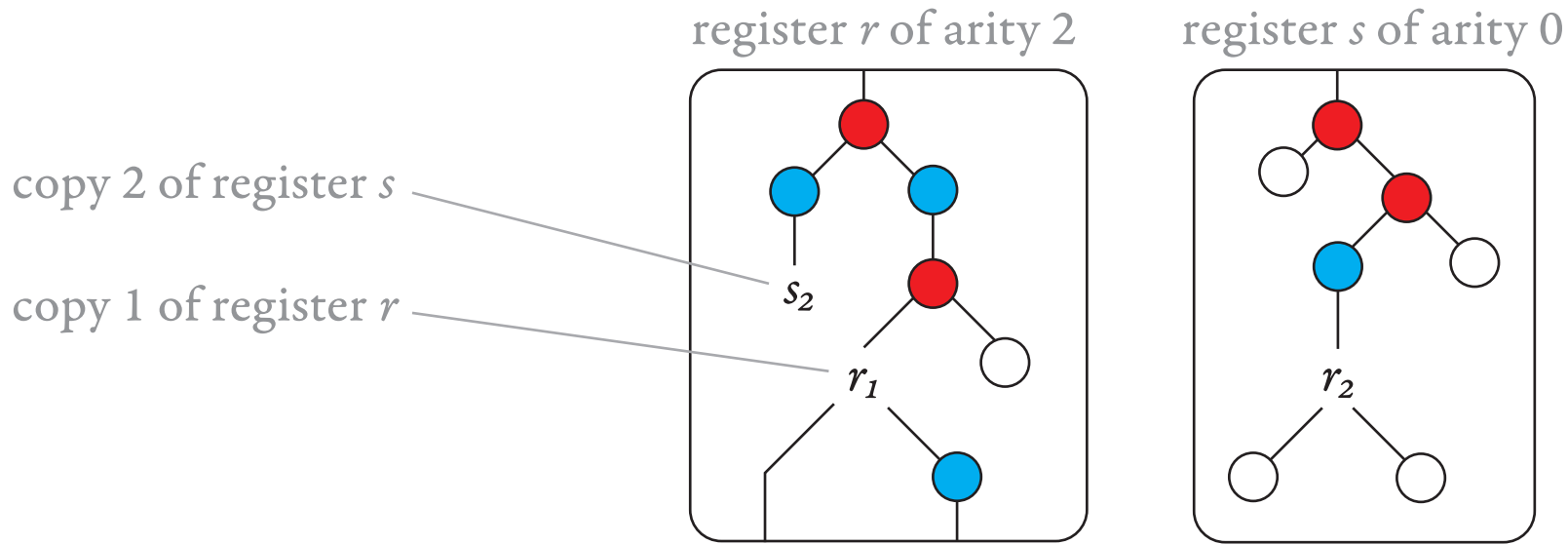
input



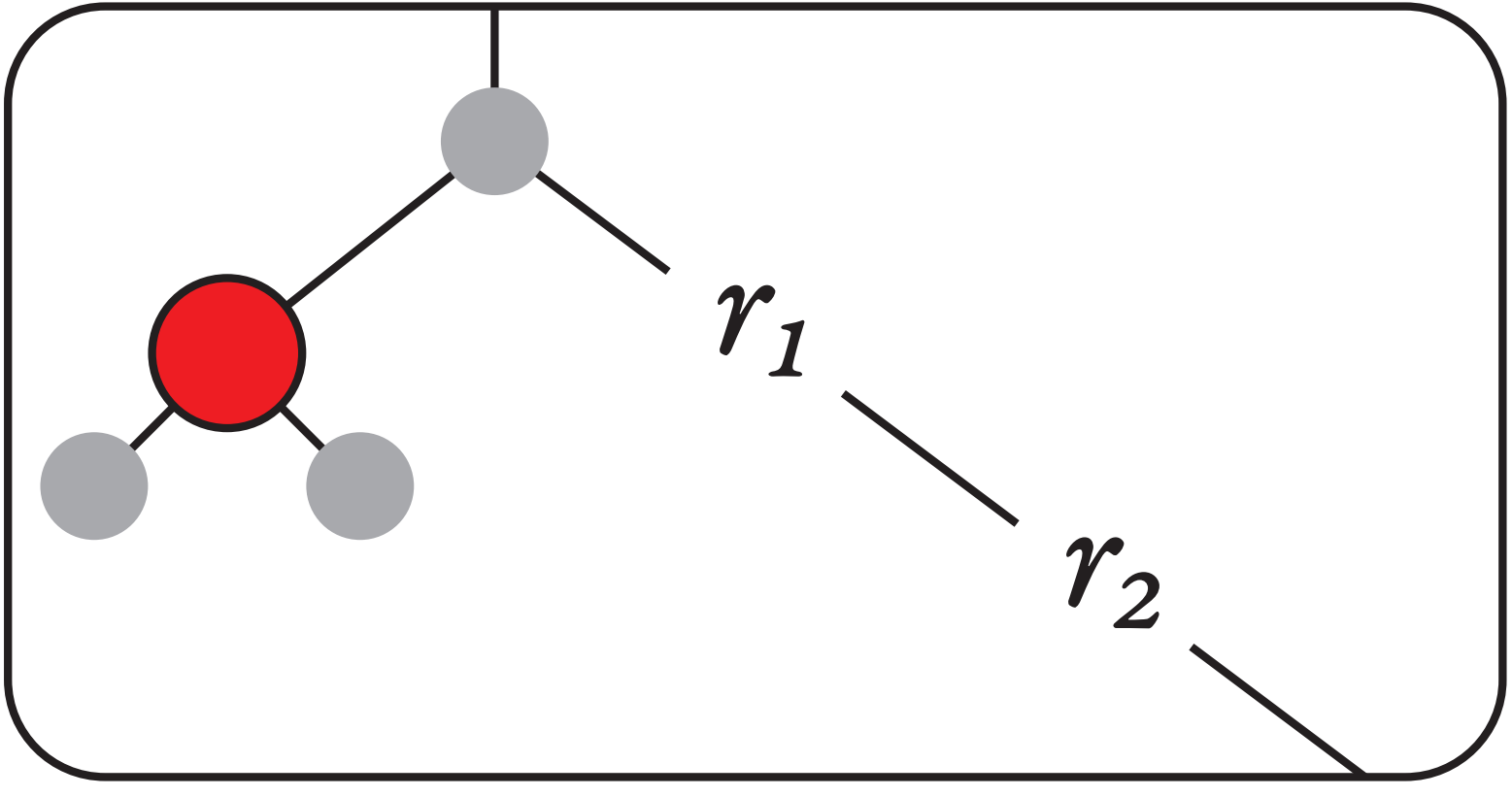
output

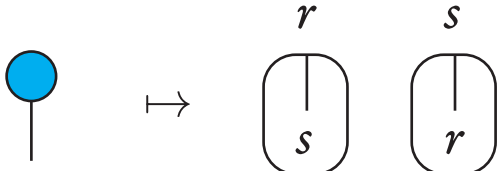


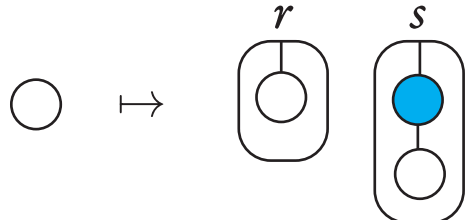

register r of arity 2
register s of arity 0
letters of the
output alphabet

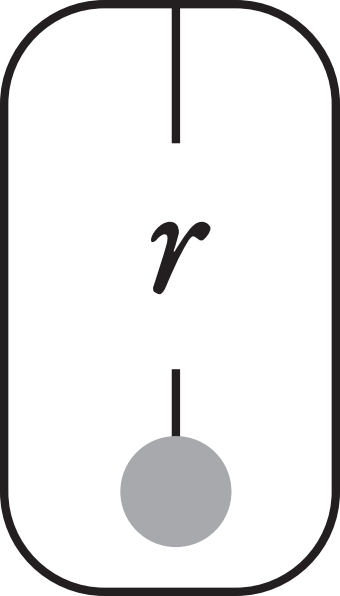

register r of arity 2
register s of arity 0
copy 2 of register s
copy 1 of register r
s_2
r_1
r_2


r
r
r
s


r_1
r_2

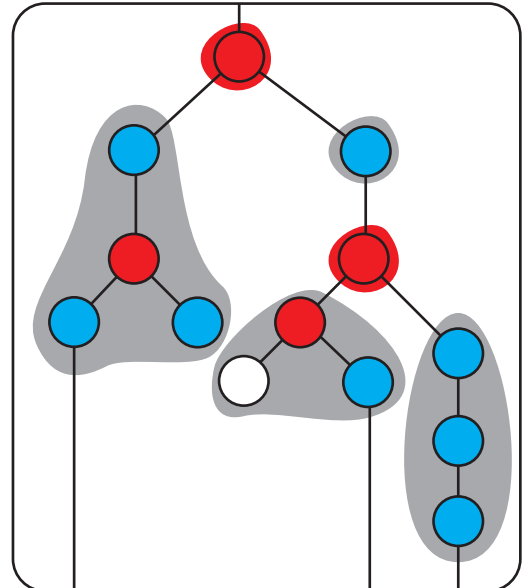

r
s
s
r

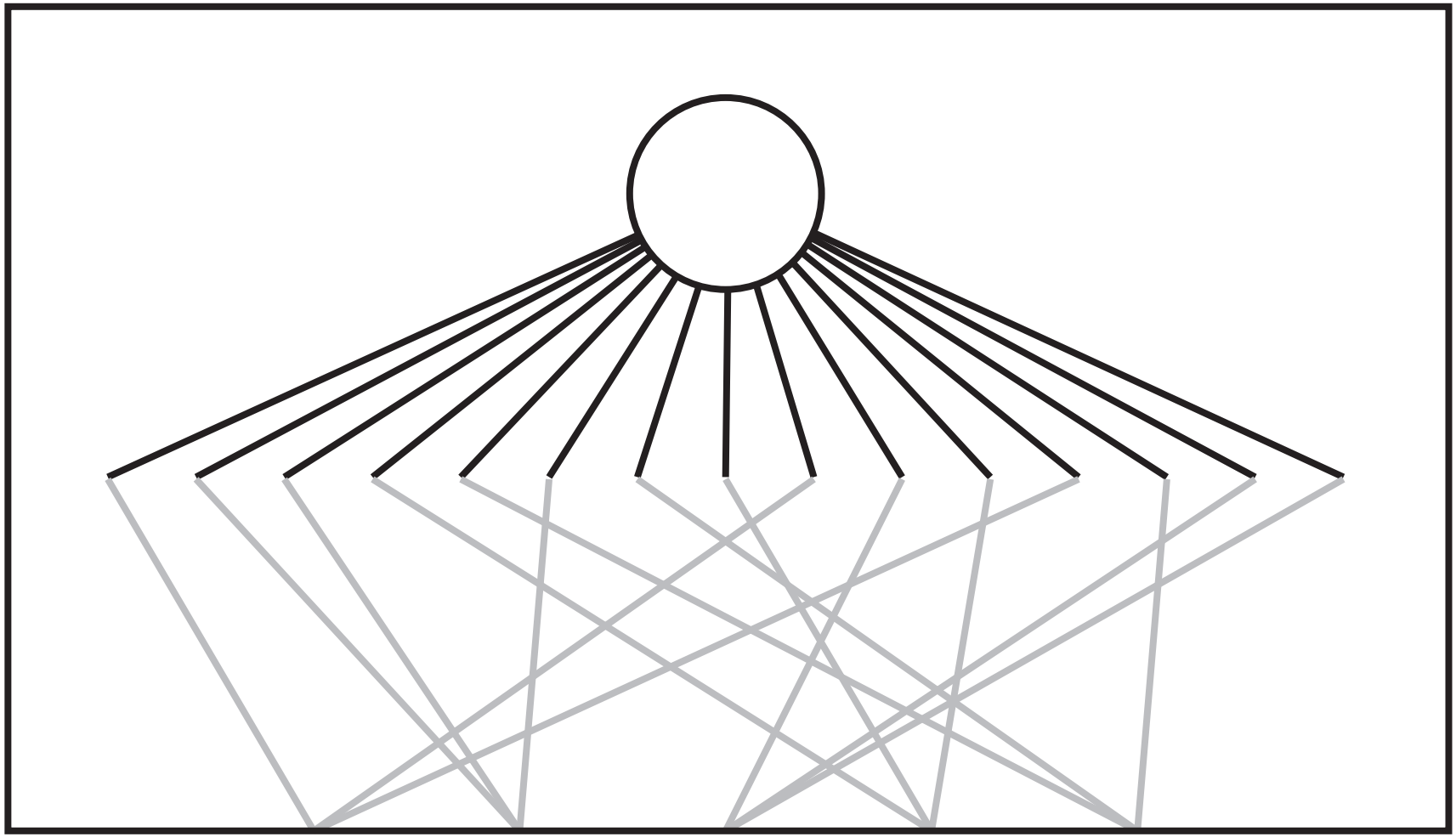

r
s


r

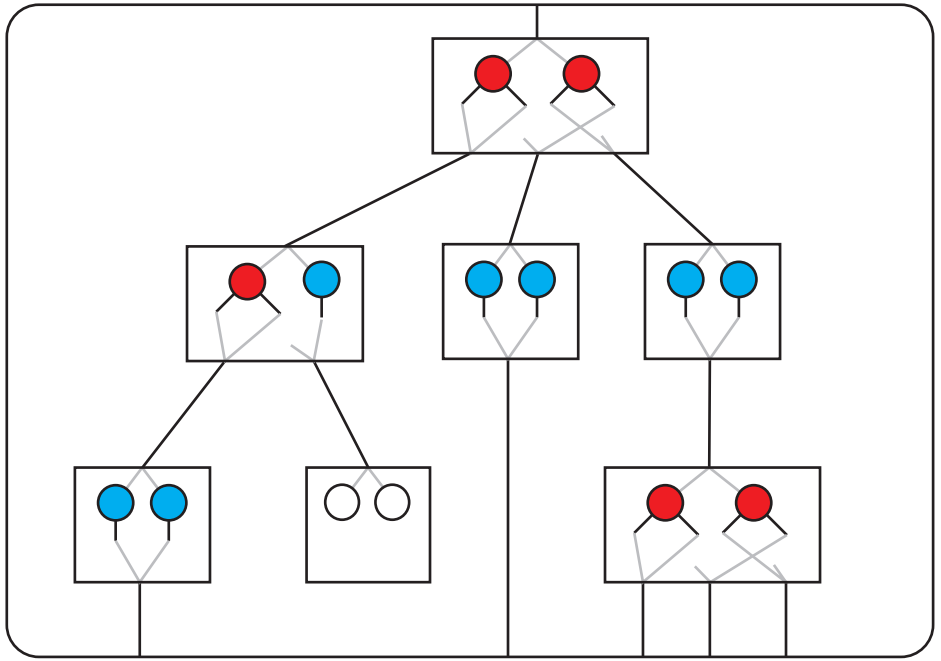
factors without
branching nodes

factors with
branching nodes

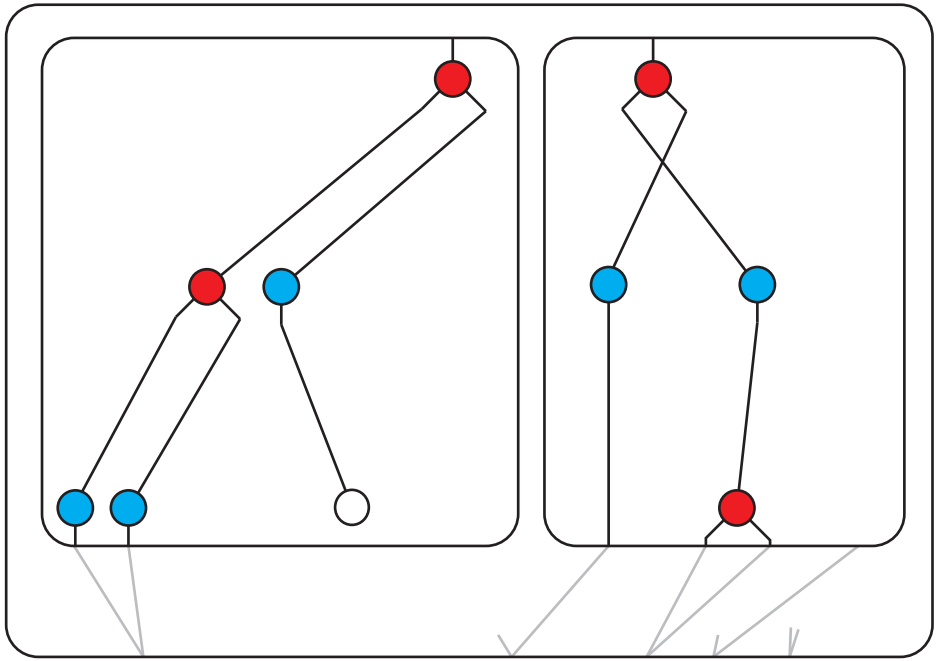


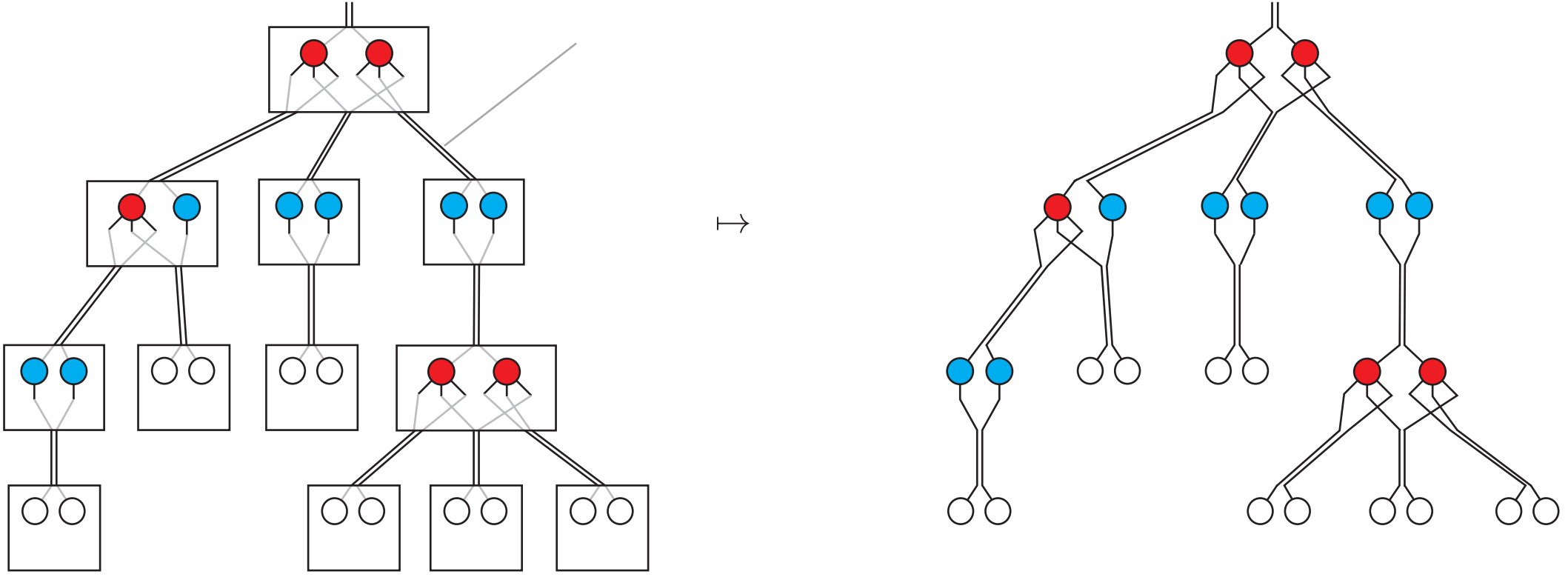


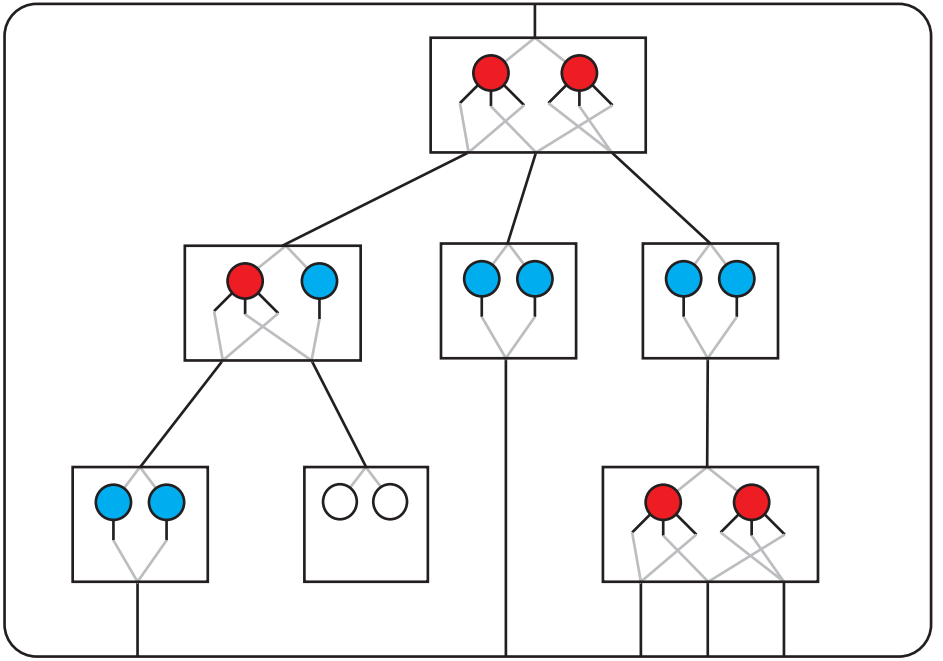
a term of matrix powers

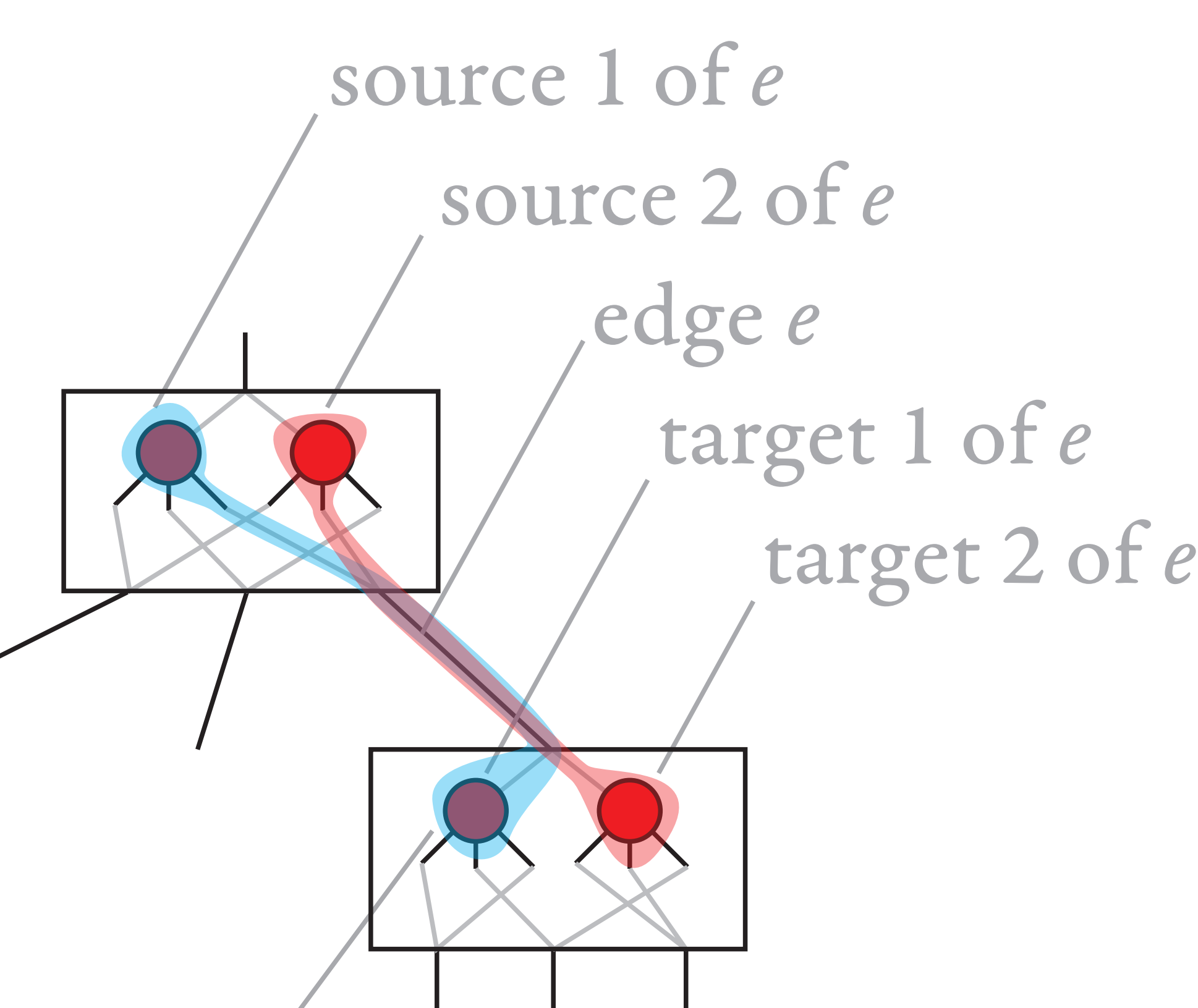


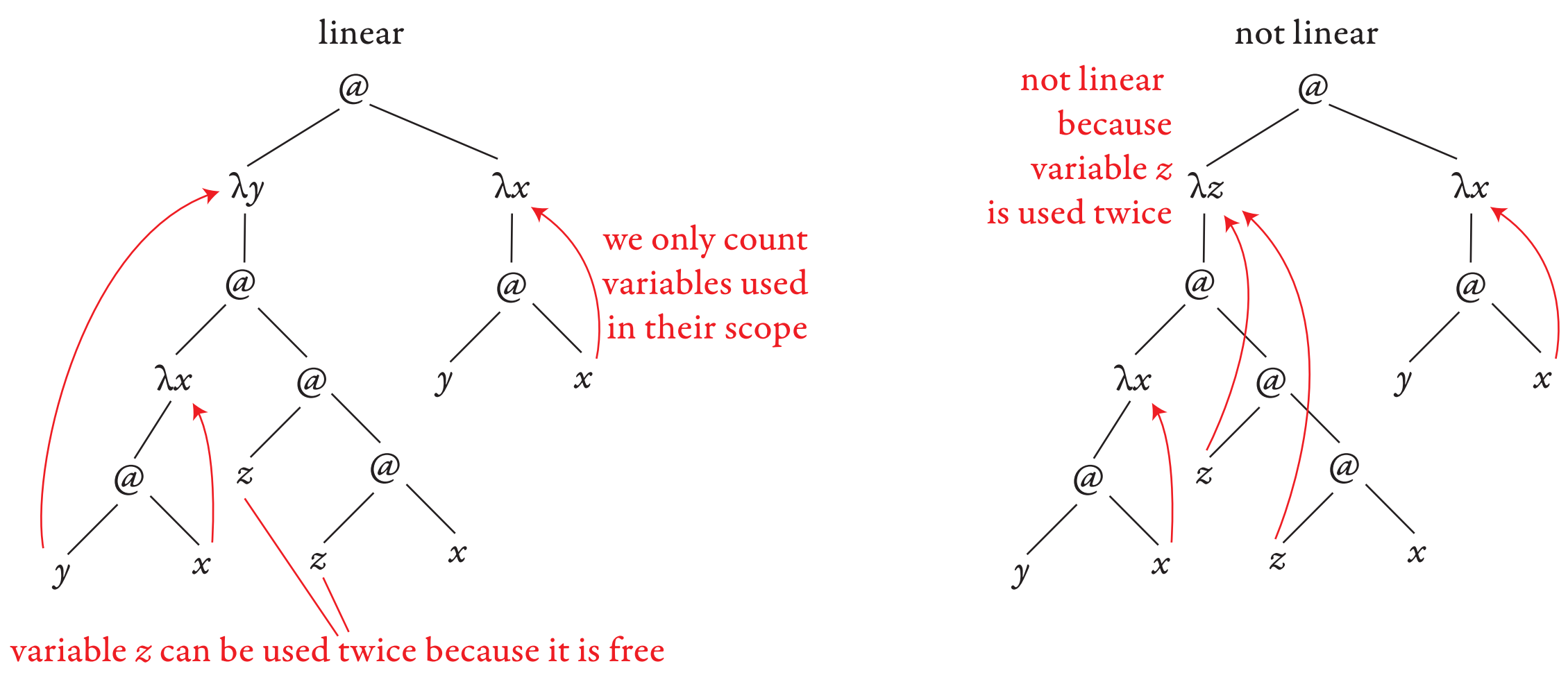
its term unfolding



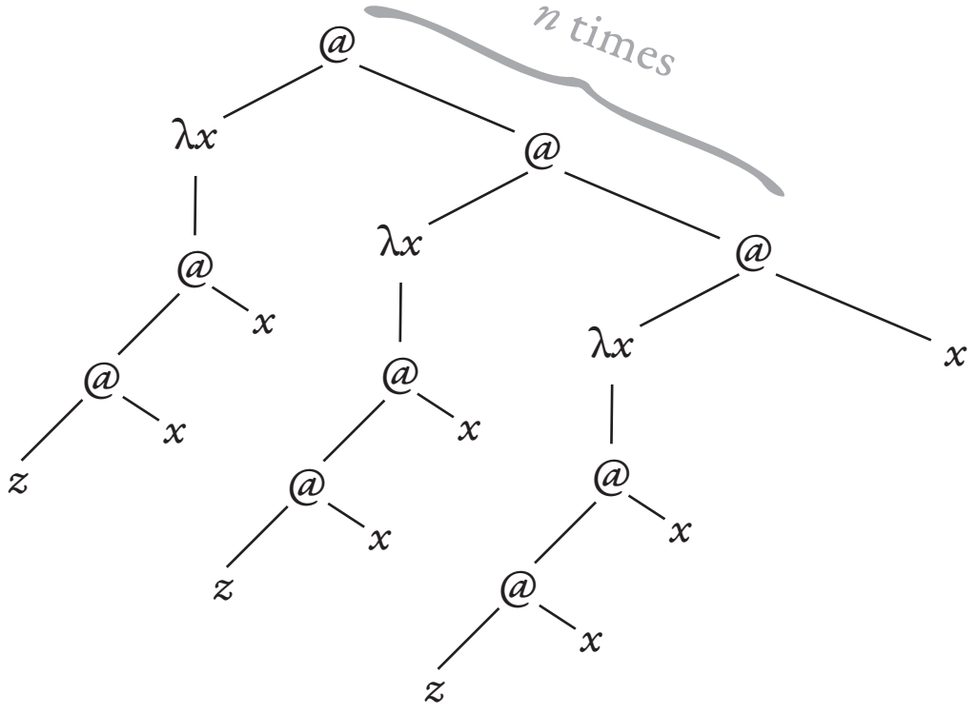




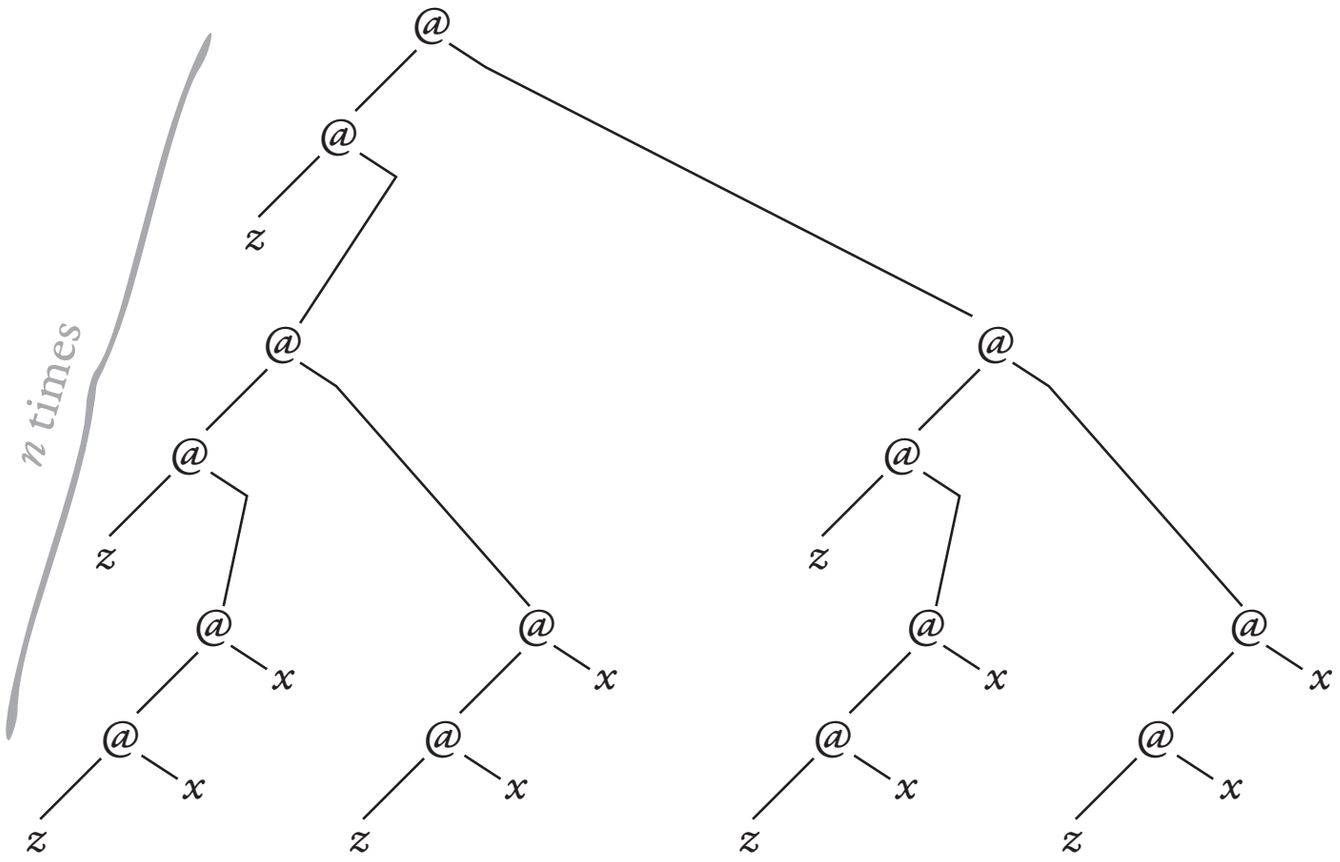

source 1 of e
source 2 of e
edge e
target 1 of e
target 2 of e


linear
@
λy
λx
@
@
λx
@
y
x
@
z
@
y
x
z
x
we only count
variables used
in their scope
variable z can be used twice because it is free
not linear
not linear
because
variable z
is used twice
@
λz
λx
@
@
λx
@
y
x
@
z
@
y
x
z
x

a λ-term of size $O(n)$

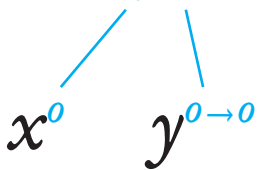


its normal form of size $O(2^n)$

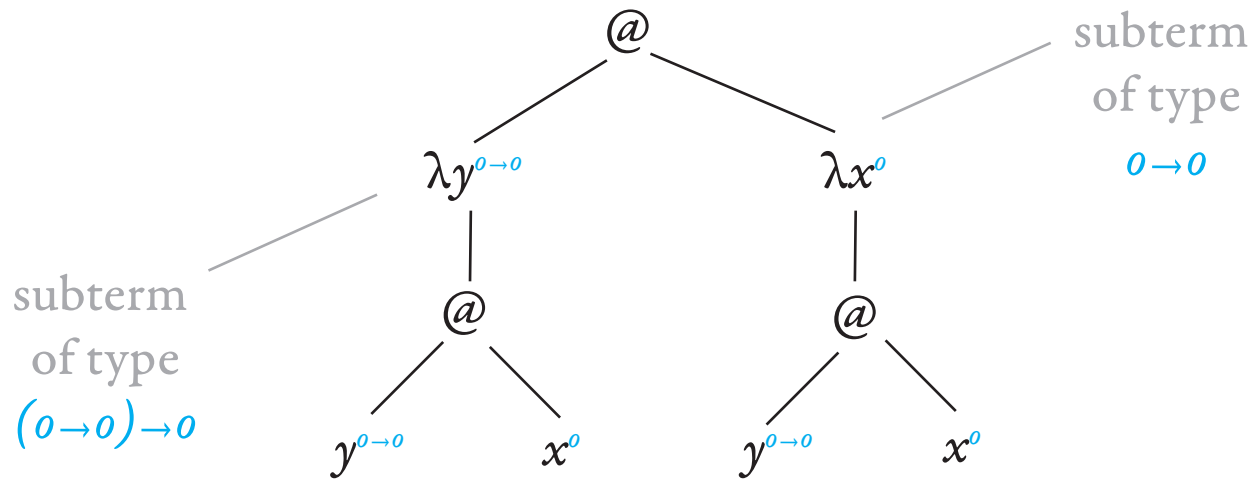


variables

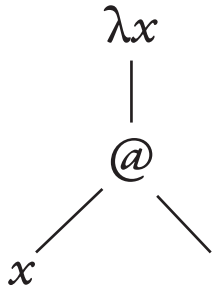
types of variables in superscript



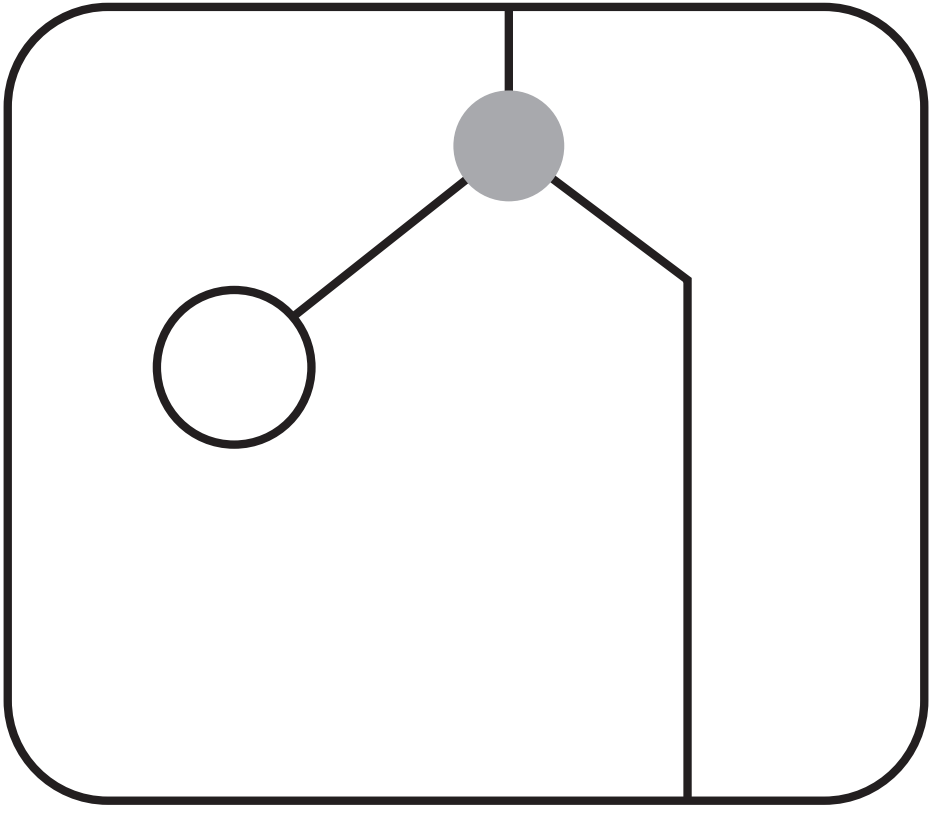
λ-term of type o

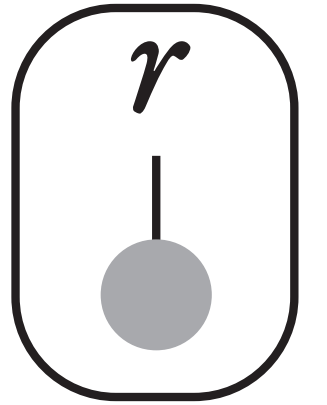


@

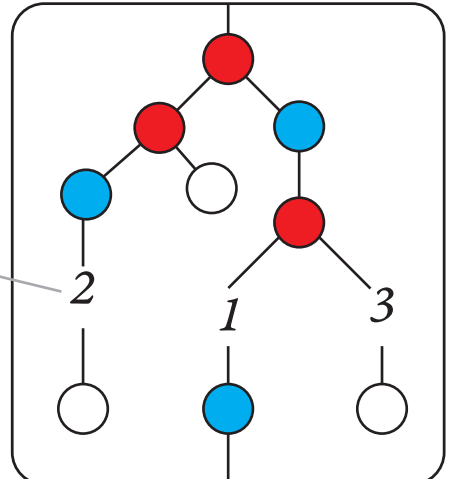


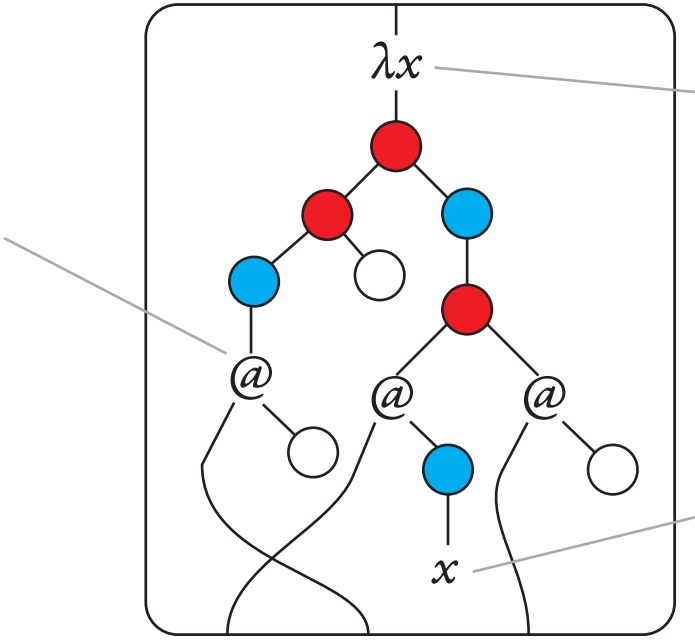
λx.




r

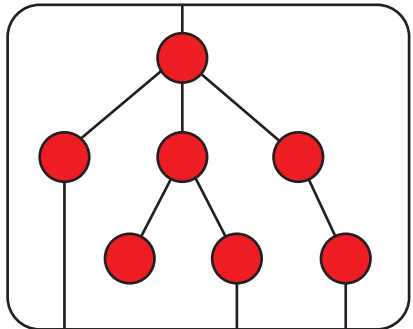
placeholder for the term
stored in the unique register
of the 2nd child



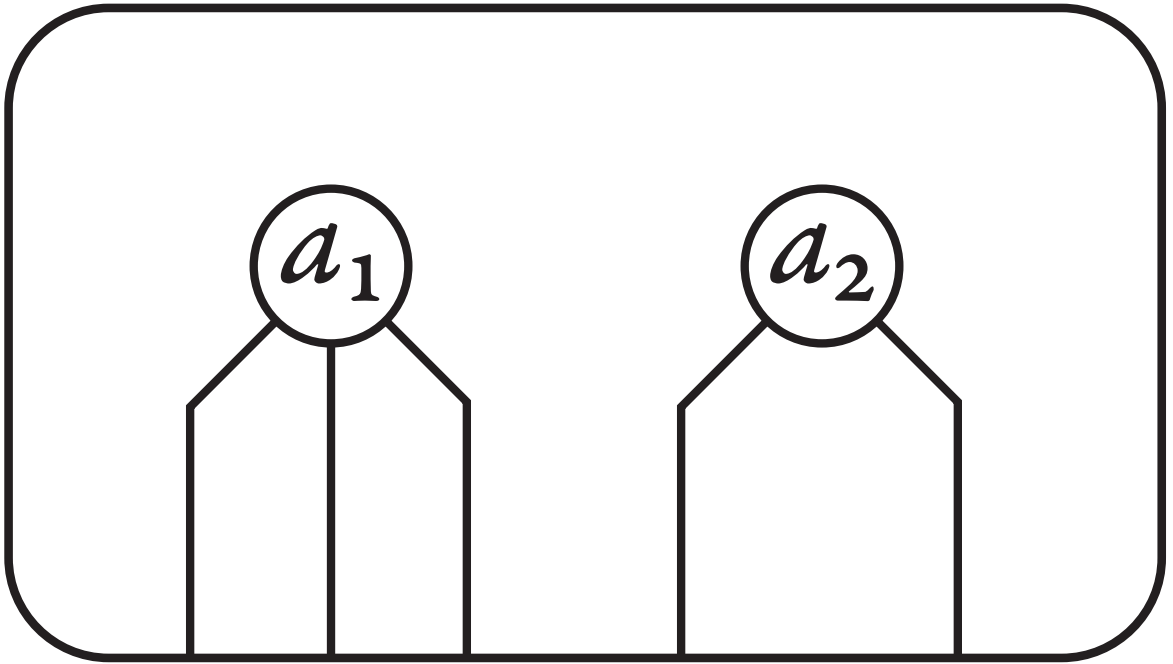


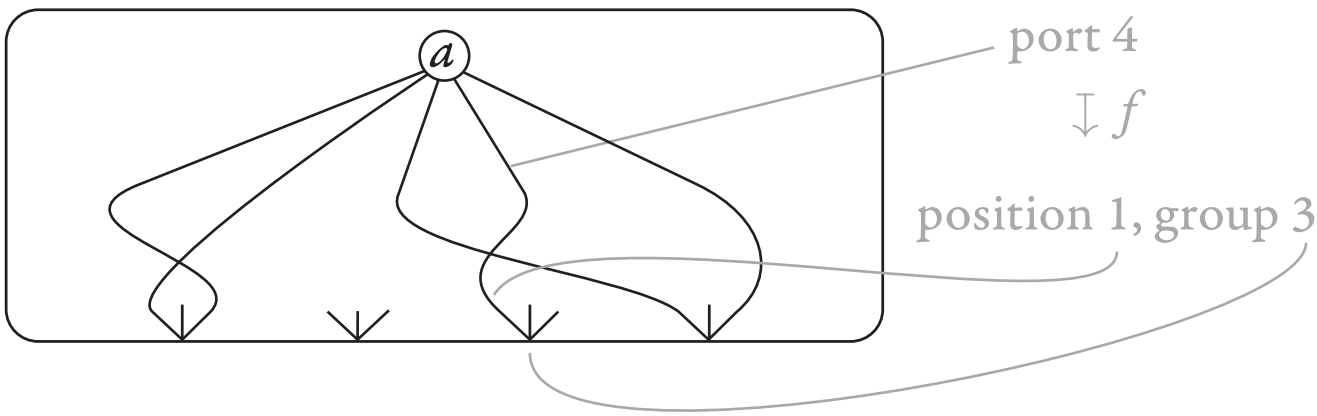
variable x is bound in the root

the original port is replaced by x



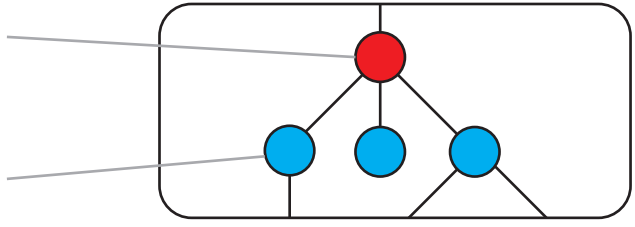
dangling edges
represent ports


a_1
a_2

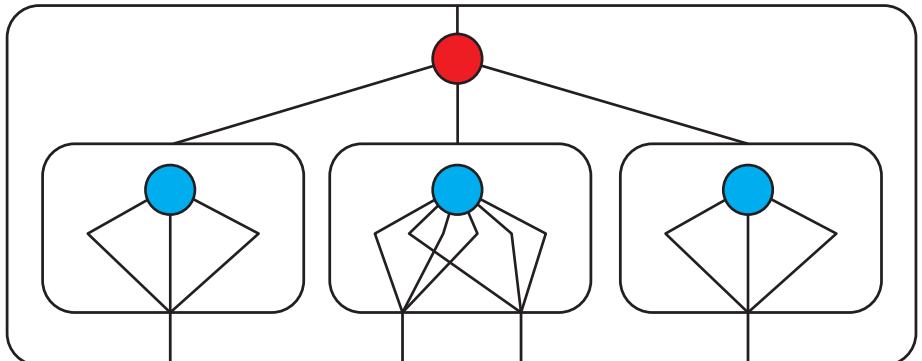

a
port 4
↧ f
position 1, group 3

the root is from Σ

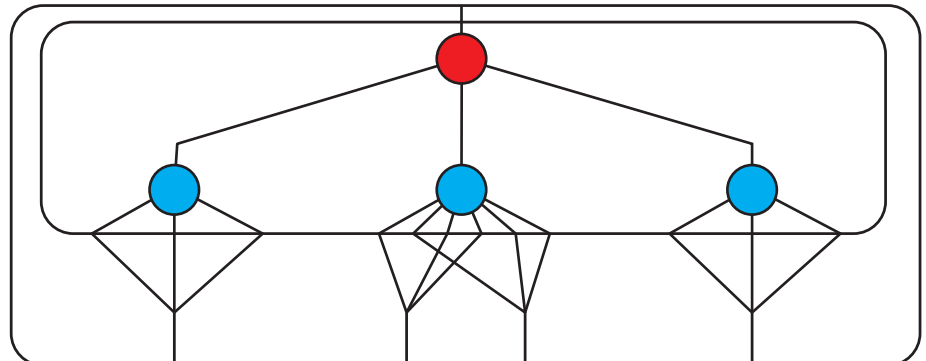
all children are from Γ

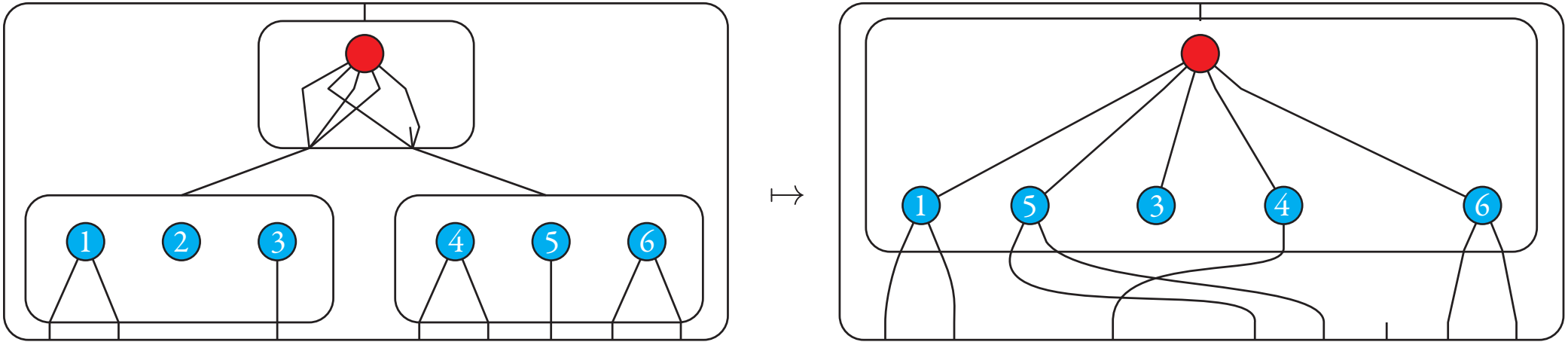


input



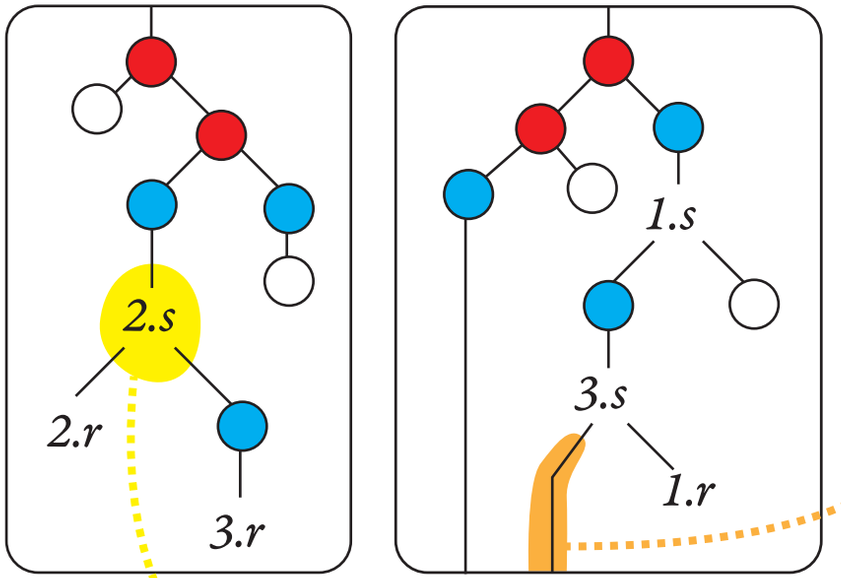
output

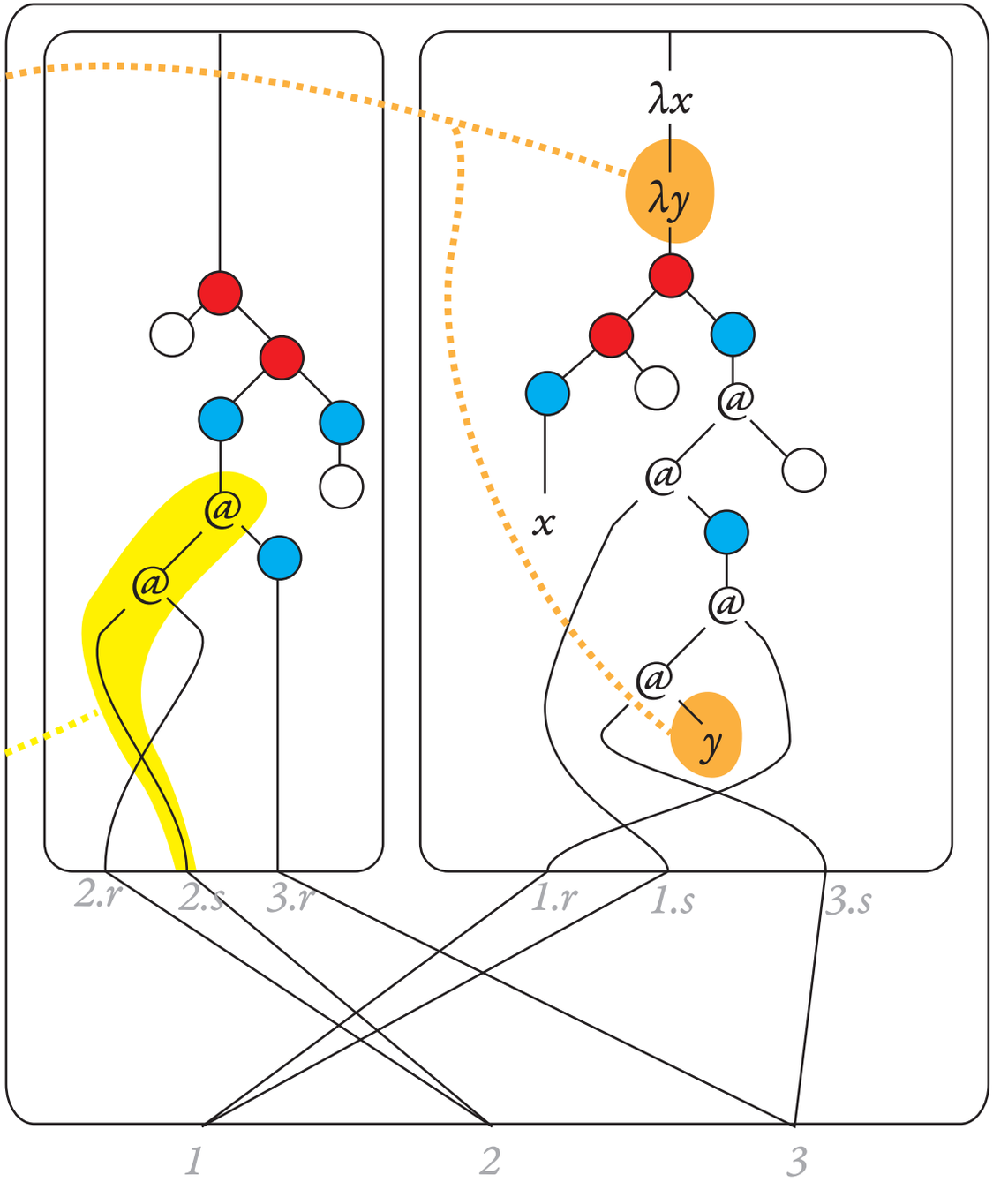


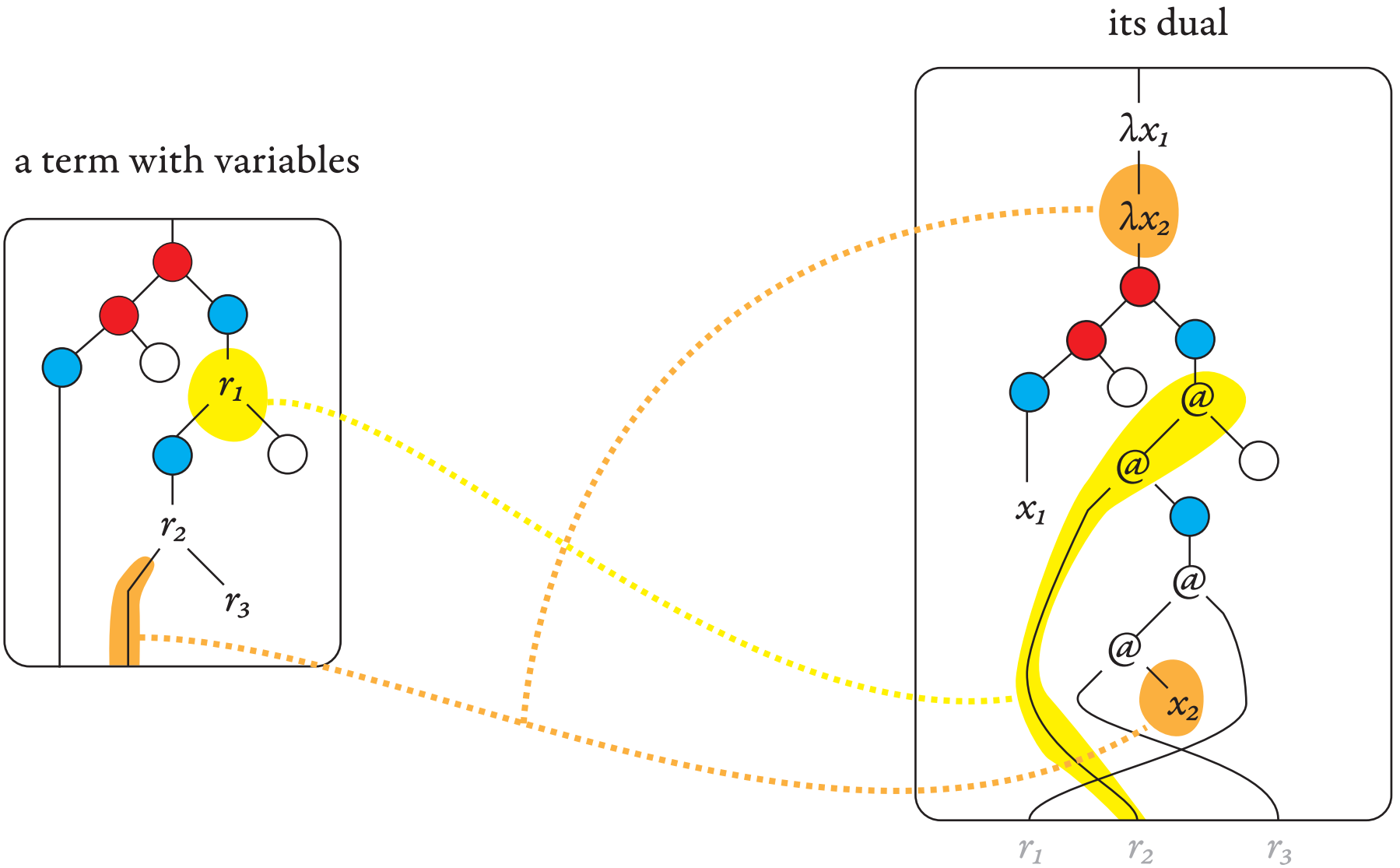

1
2
3
4
5
6
↦
1
5
3
4
6

a register update

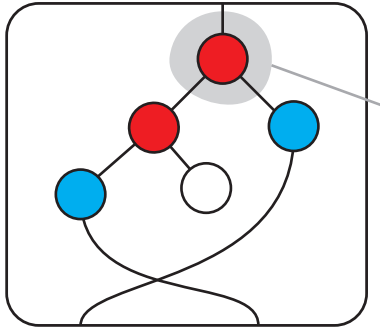
its dual





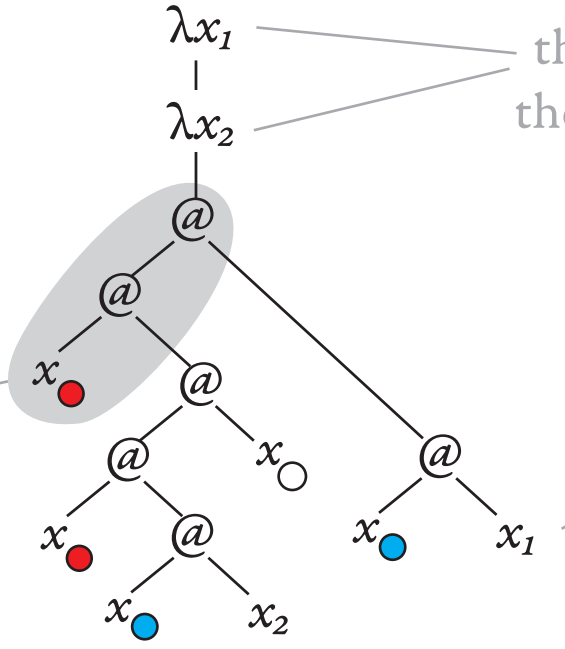

its dual
a term with variables
r_1
r_2
r_3
λx_1
λx_2
@
@
@
@
x_1
x_2
r_1
r_2
r_3

a term of arity 2



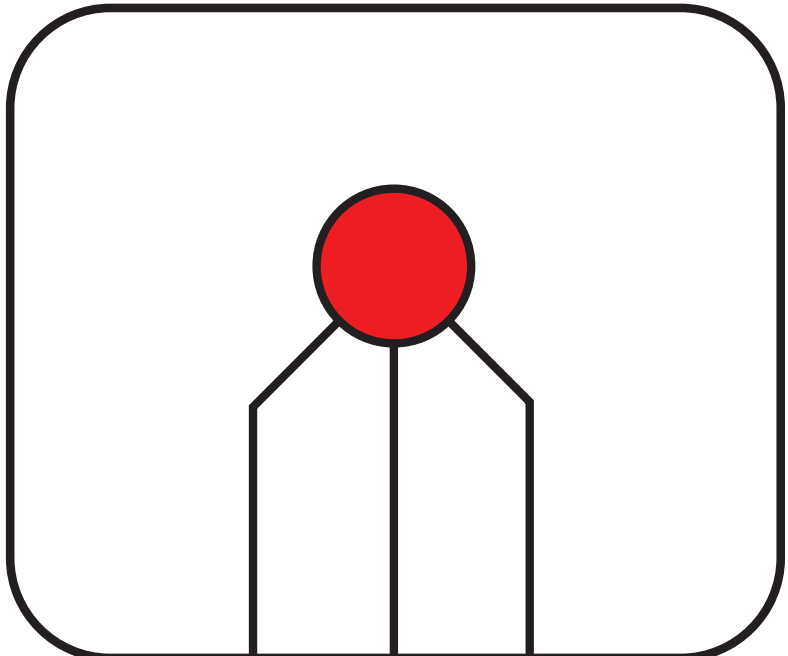
a non-port node is represented by a variable, corresponding to the label, applied to the children of the node

its representation as a λ-term

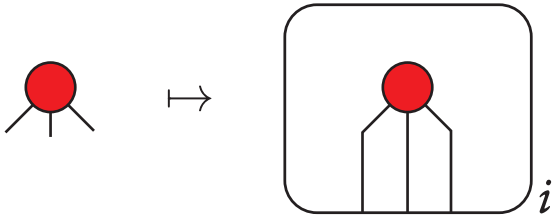


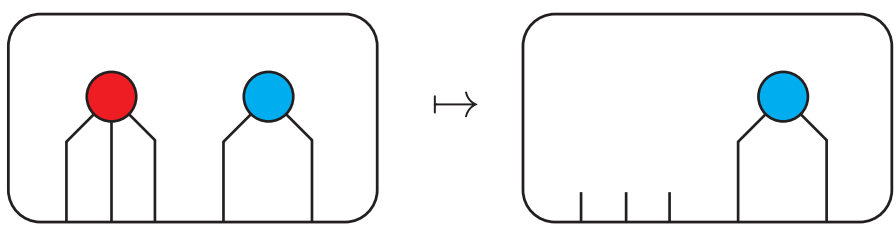
the variables representing the ports are bound outside

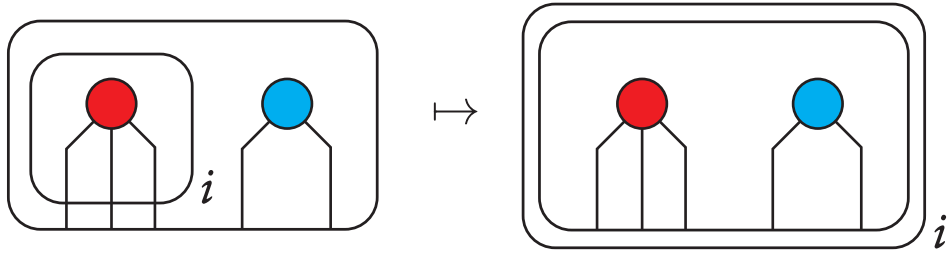
the i-th port is represented by a variable x_i of type o

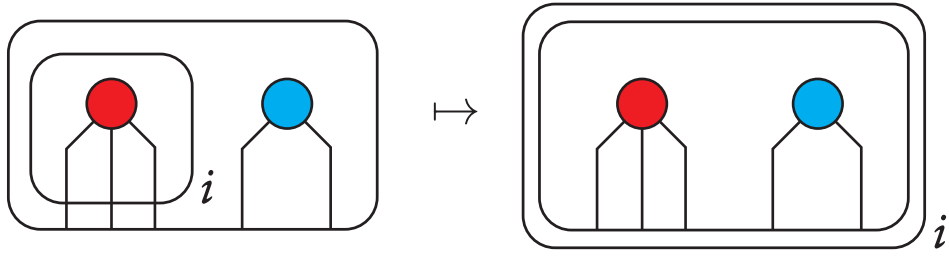


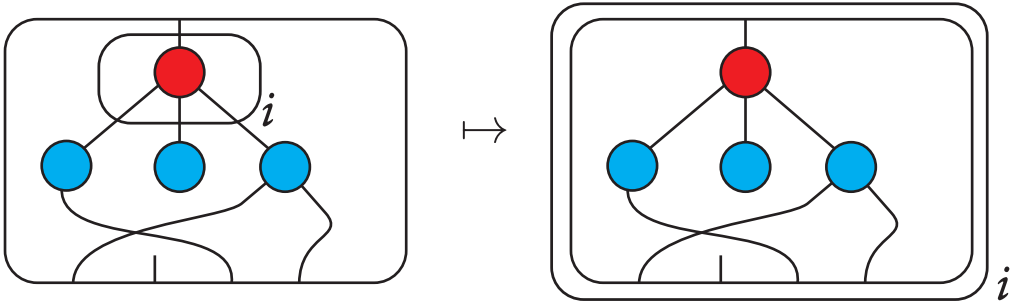
i

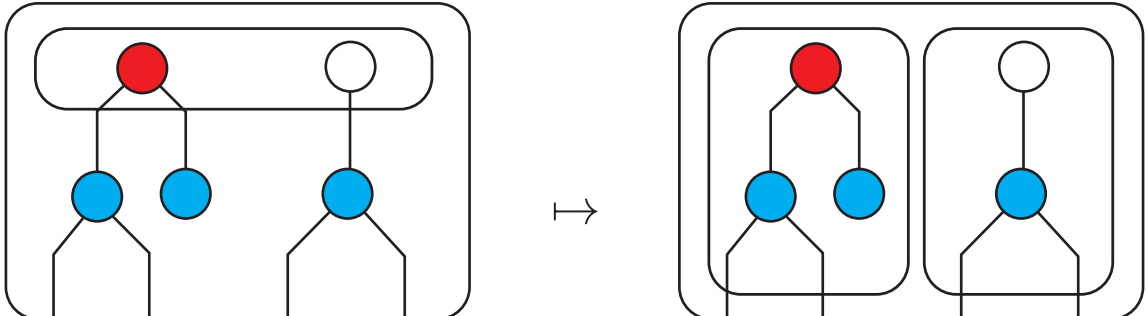

i

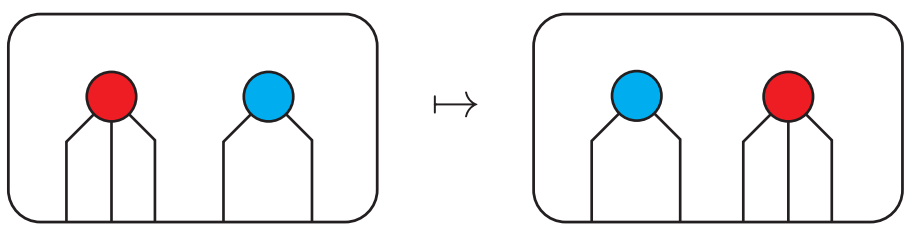


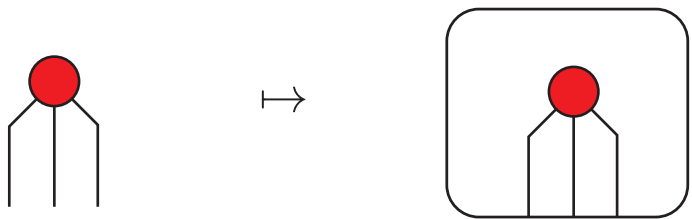

i
i

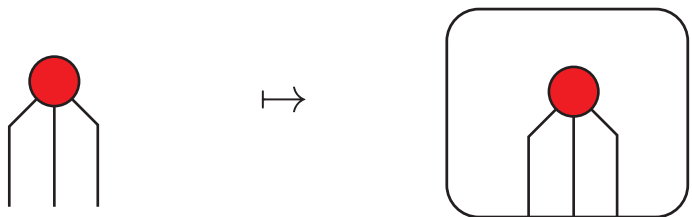

i
i

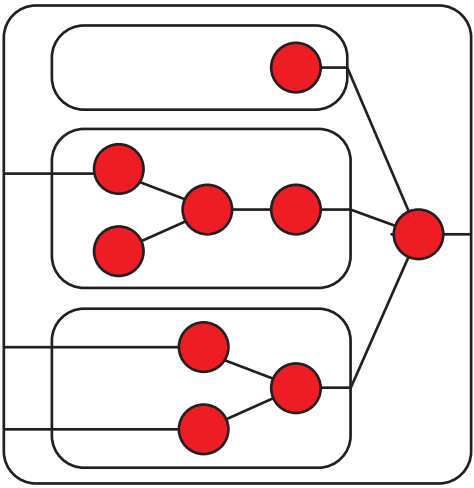

i
↦
i

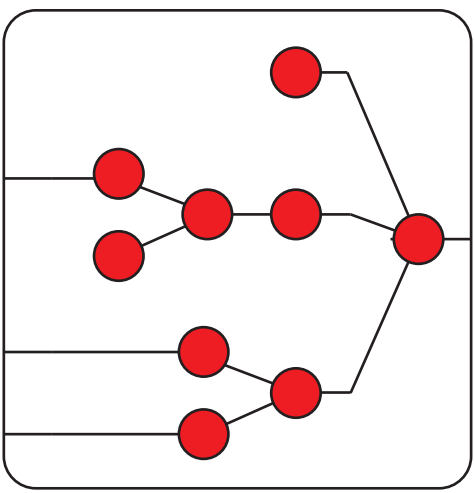


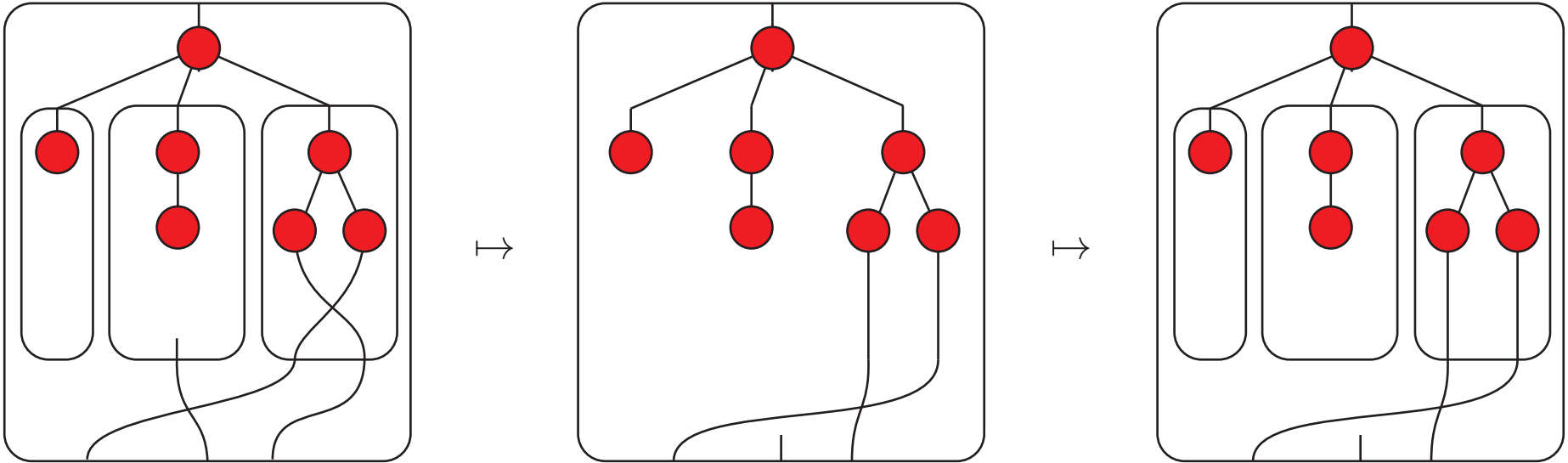


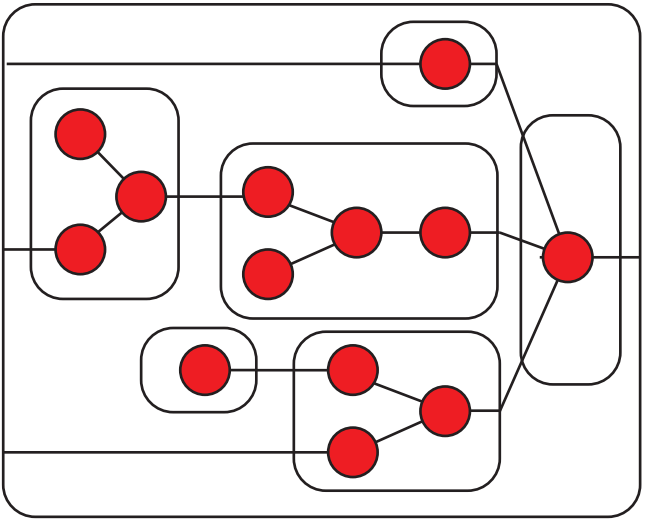


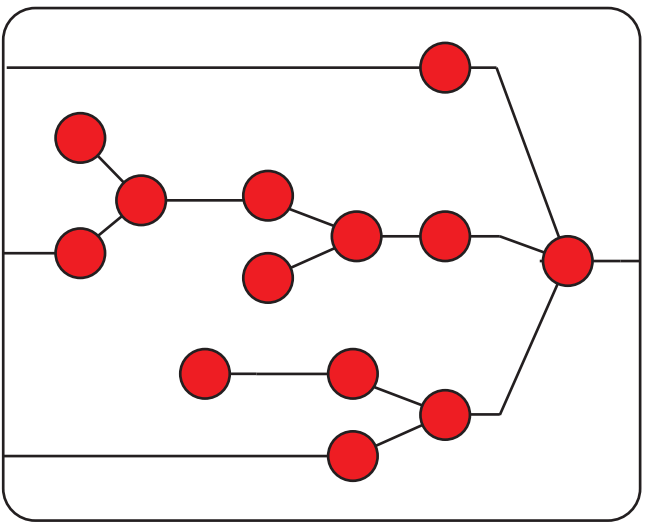


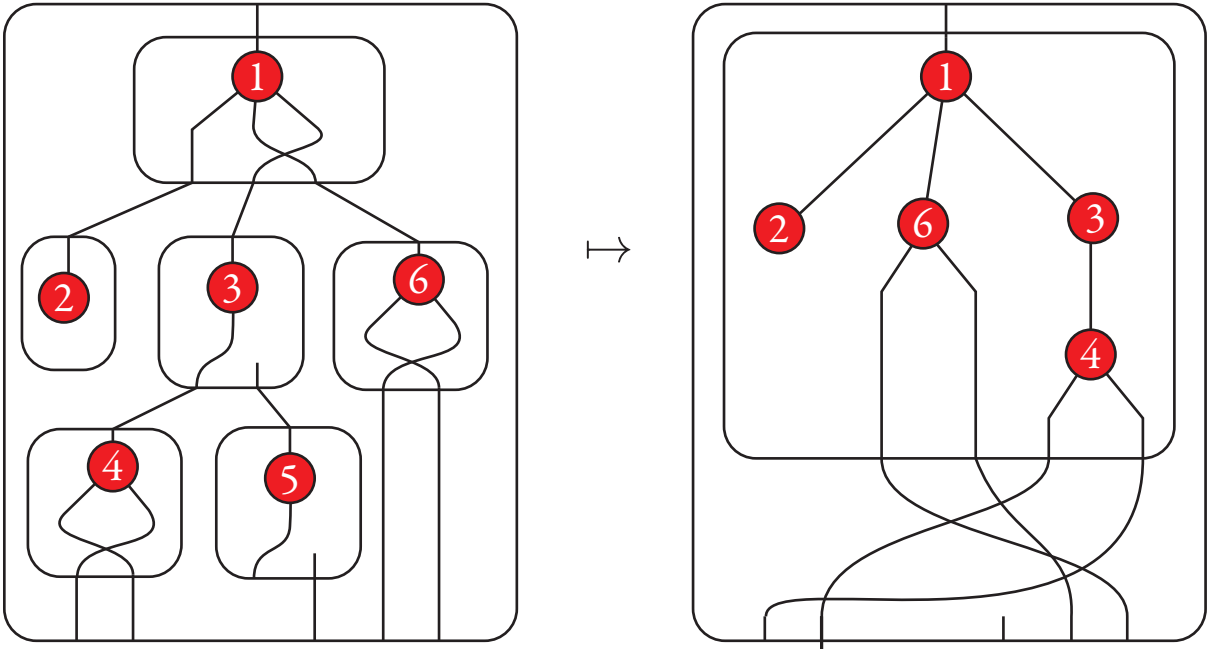


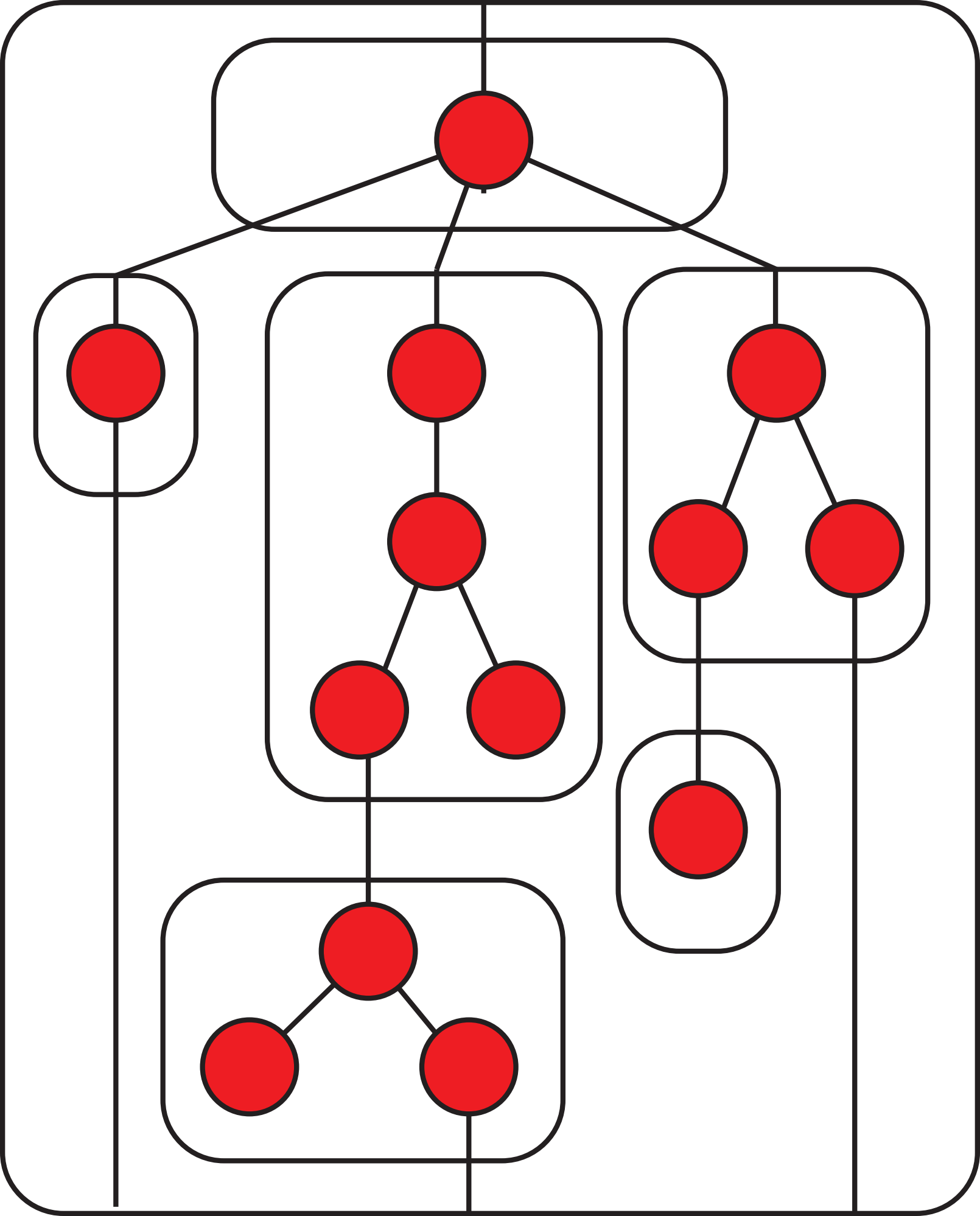


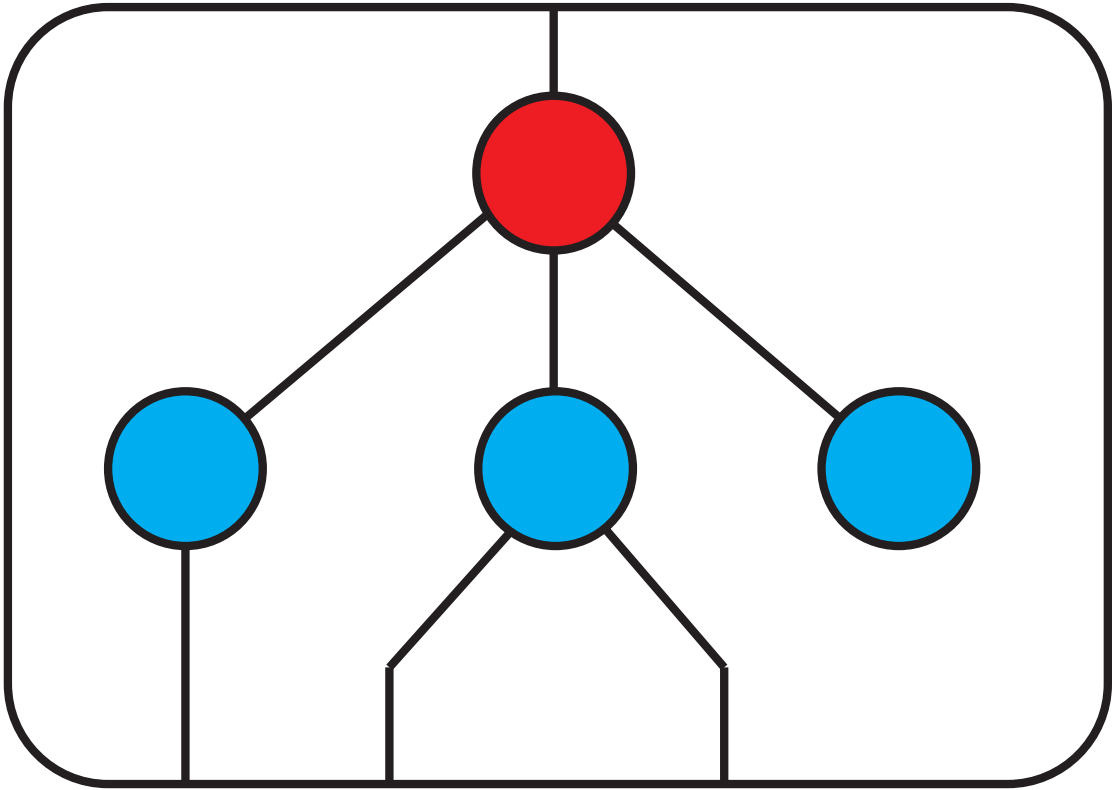


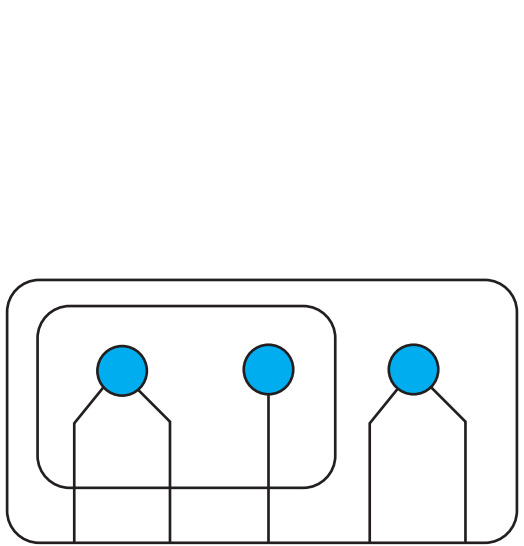


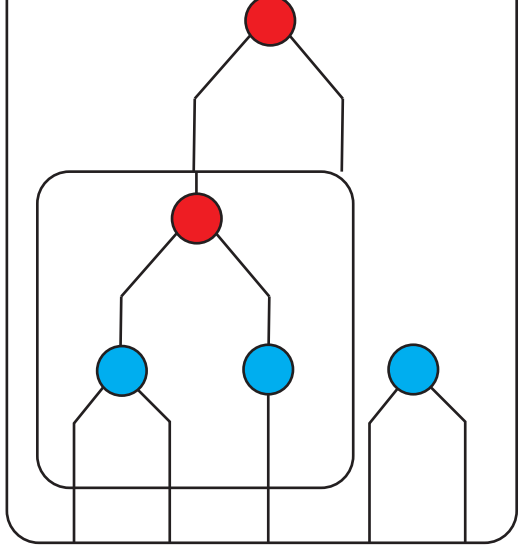


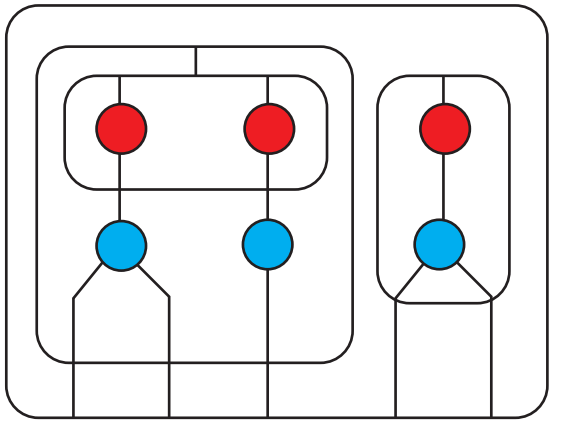

1
2
3
6
4
5
↦
1
2
6
3
4

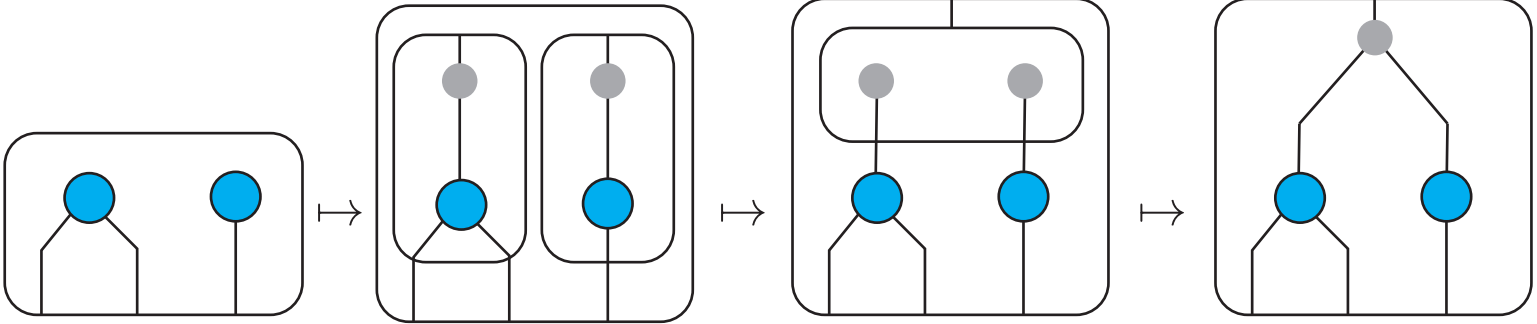


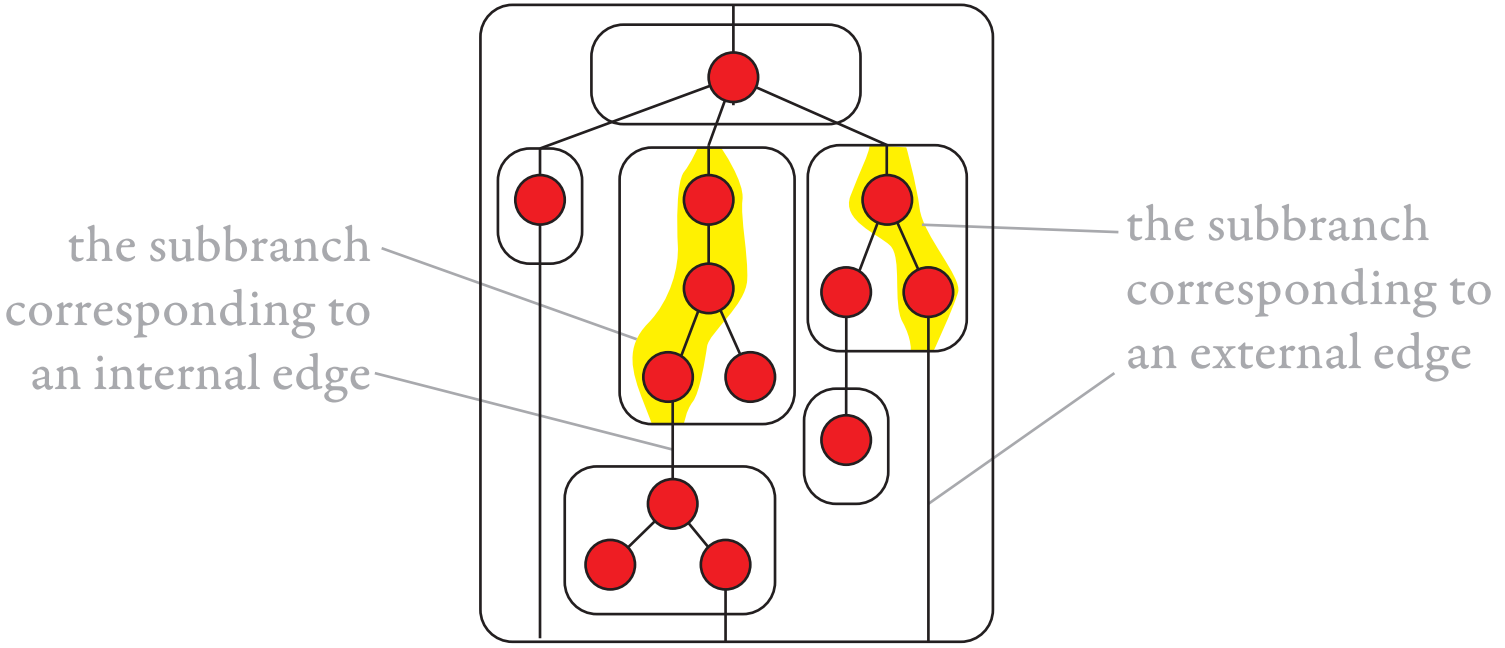


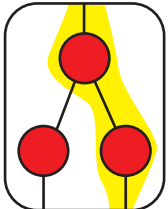




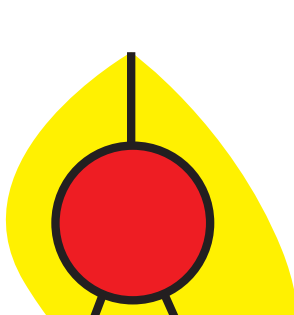


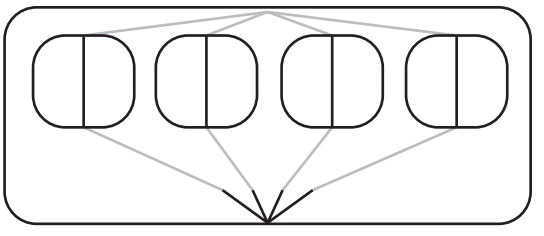



the subbranch
corresponding to
an internal edge
the subbranch
corresponding to
an external edge



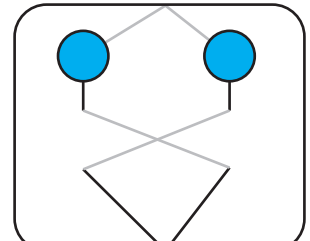
a branch can be visualised as
a term with a distinguished
root-to-port path

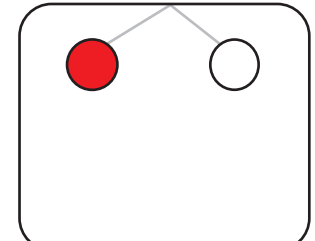


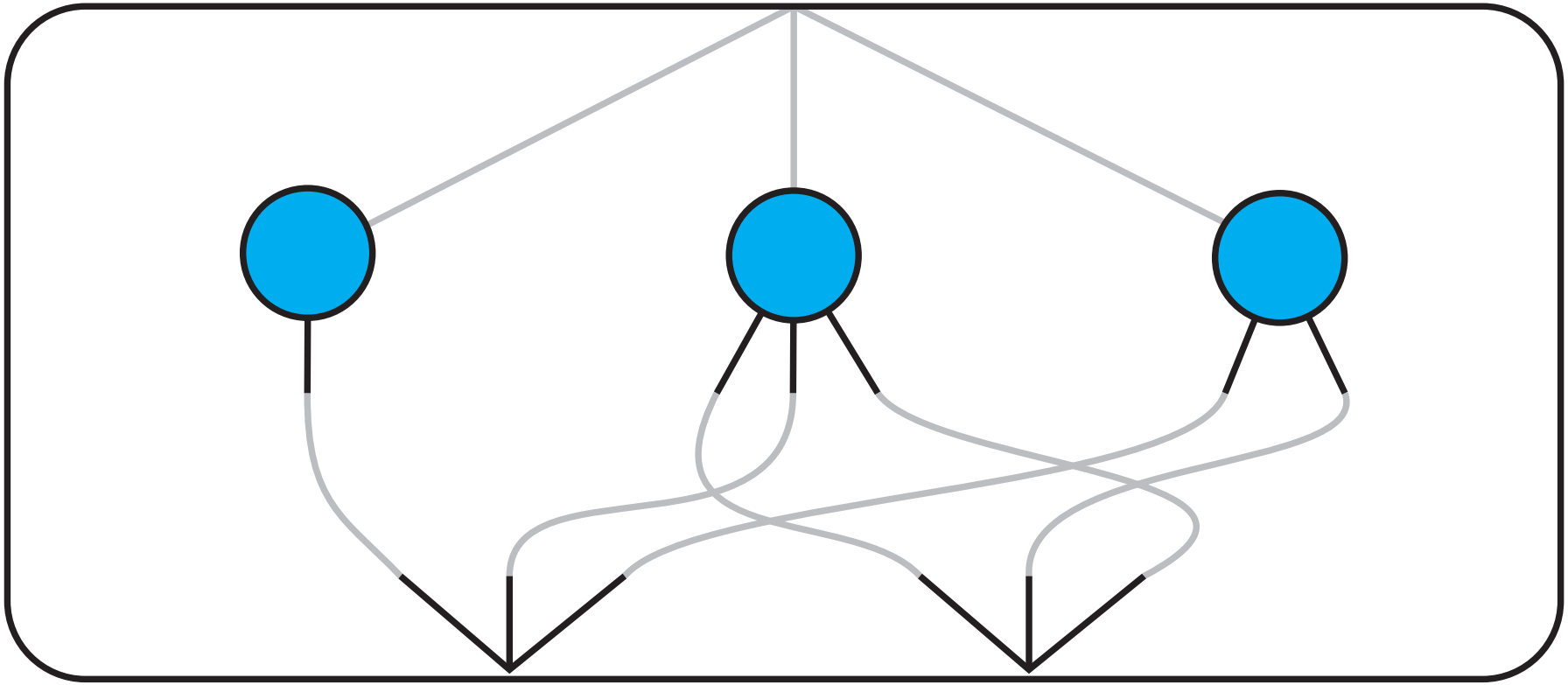


a tuple of k identity terms with all their ports folded into one

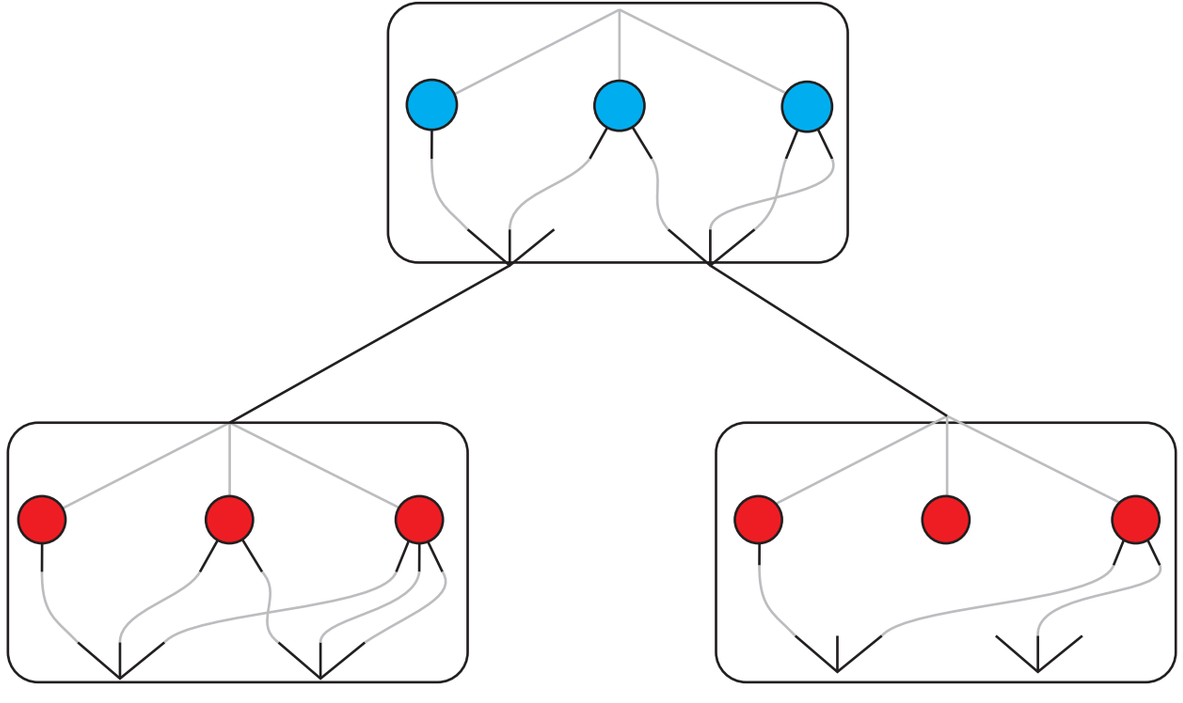
$\Sigma = \{$  ,  ,  $\}$

$a \in \Sigma^{[2]}$ 

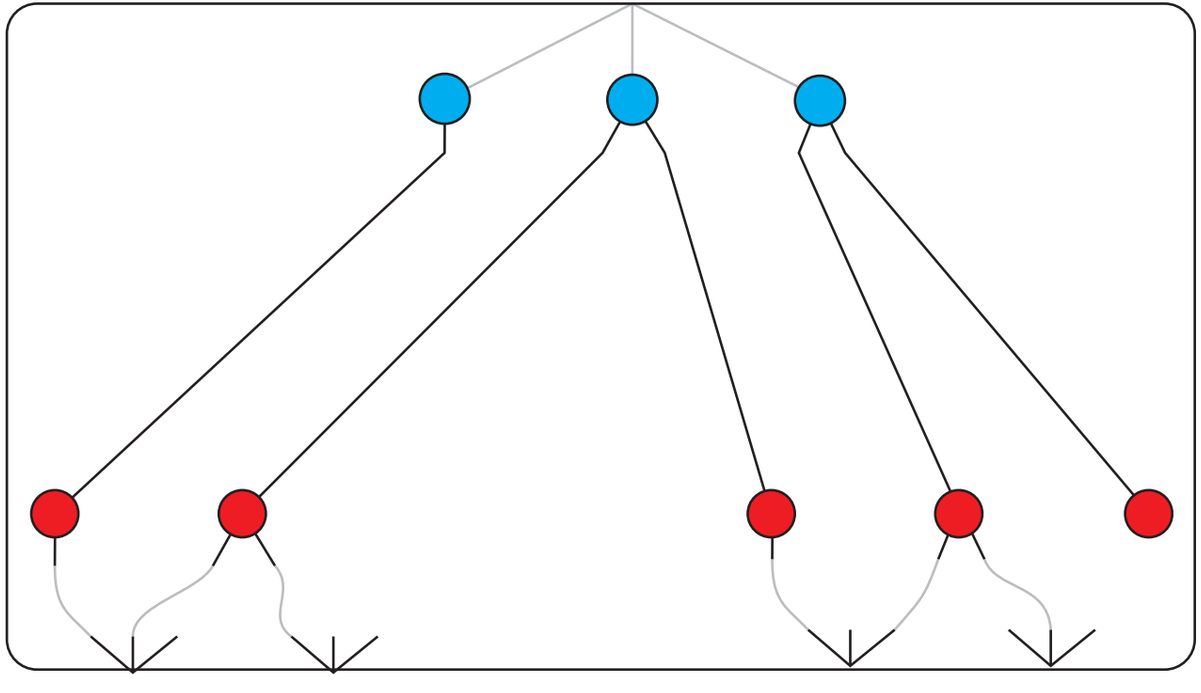
$b \in \Sigma^{[2]}$ 

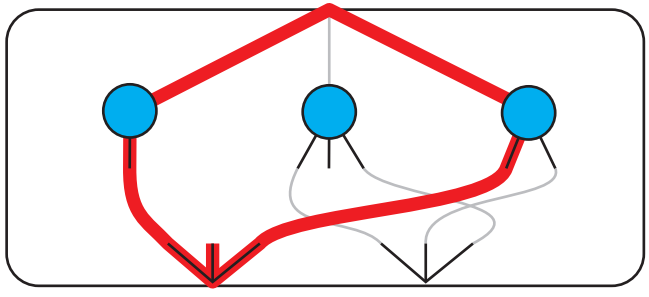


a shallow term of matrix powers

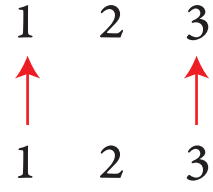


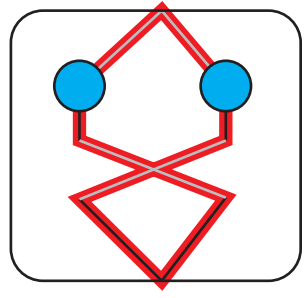
its shallow unfolding



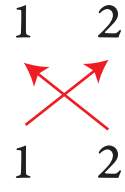


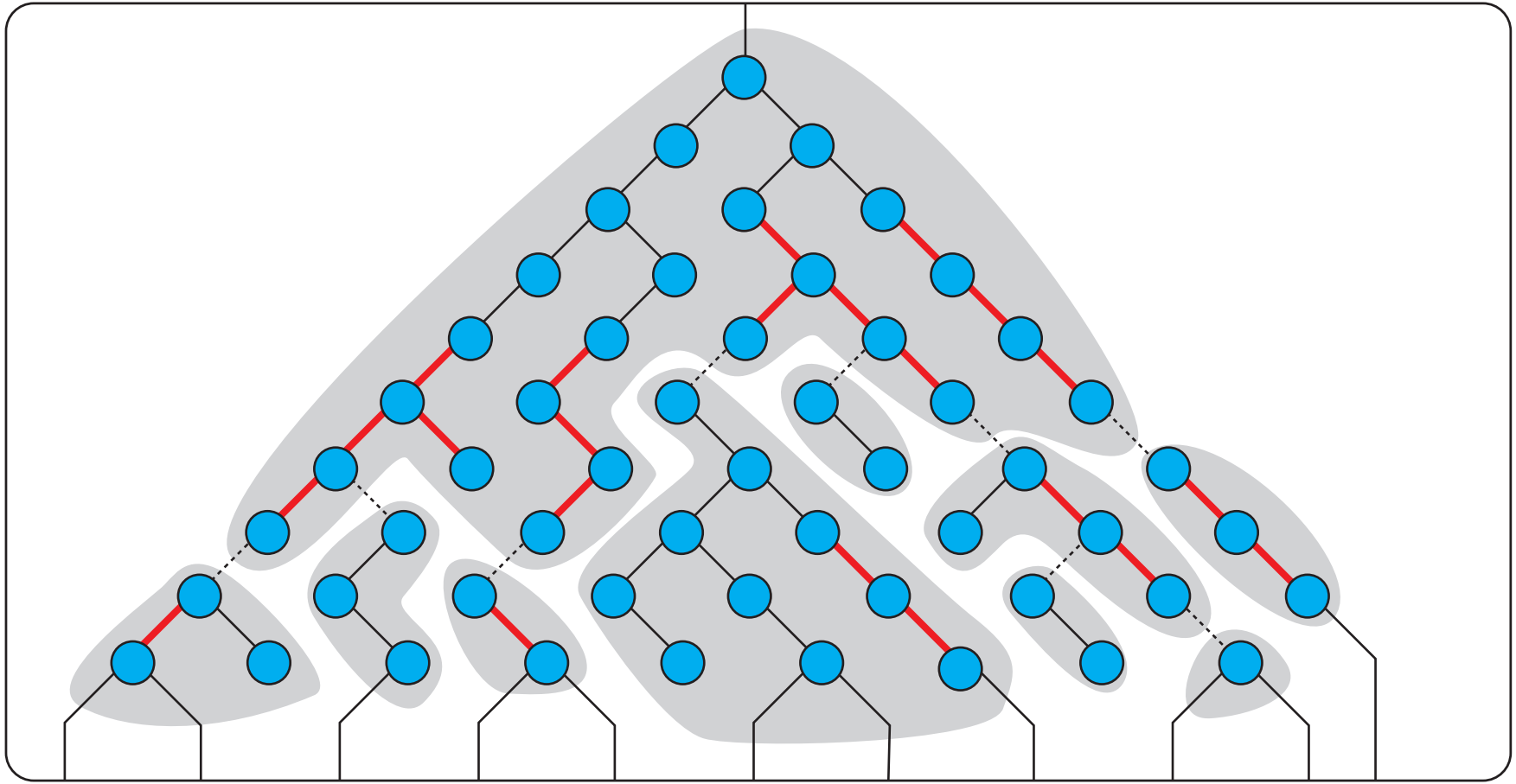
twist of port 1

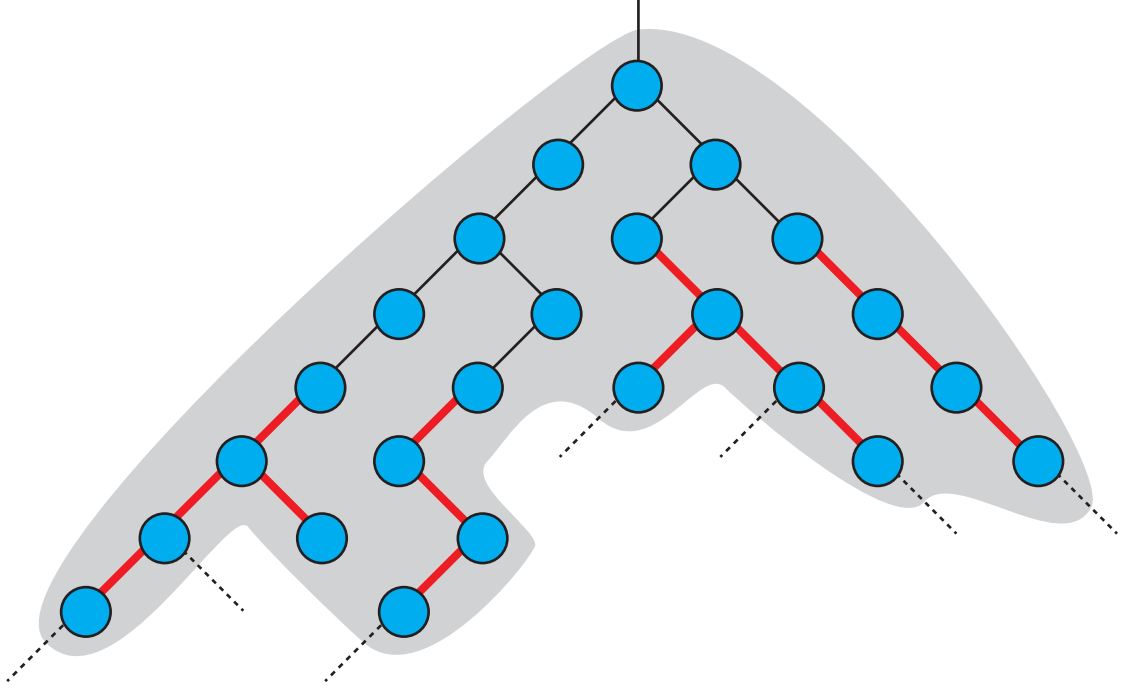


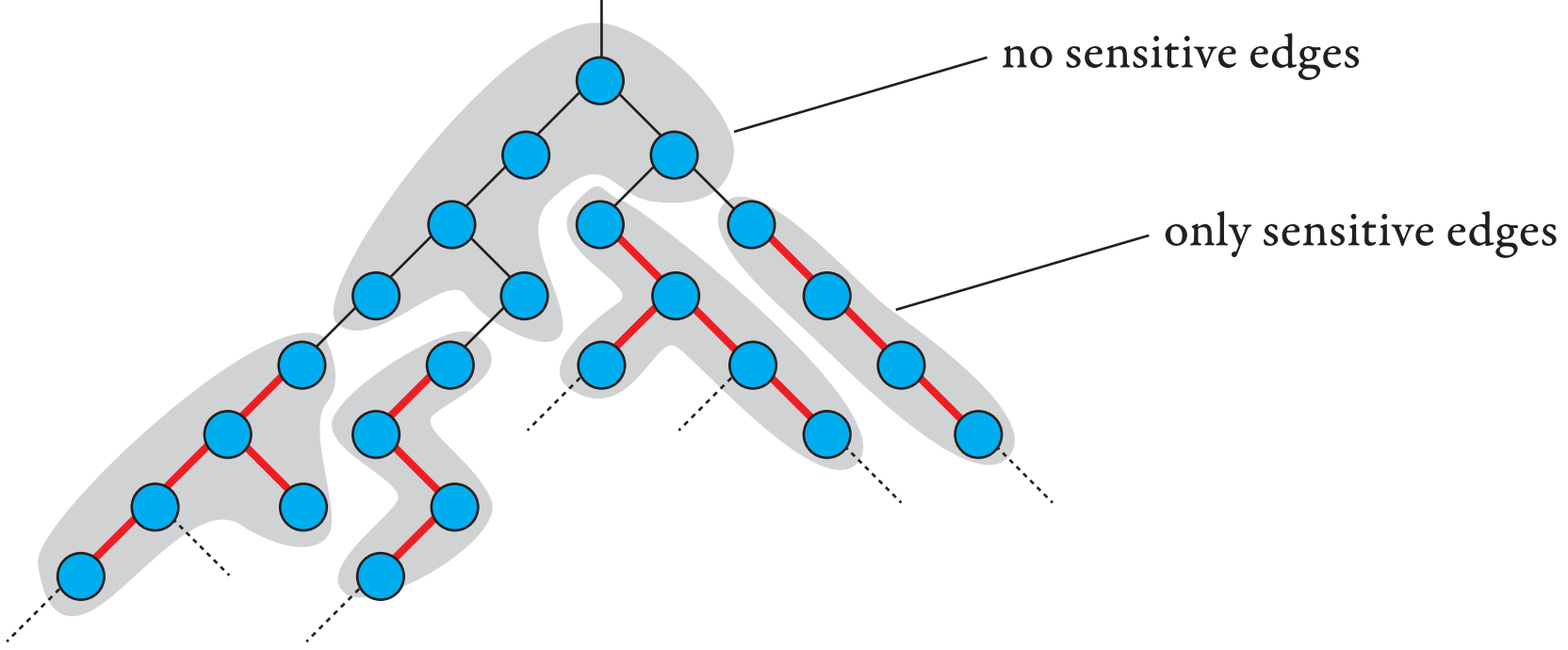


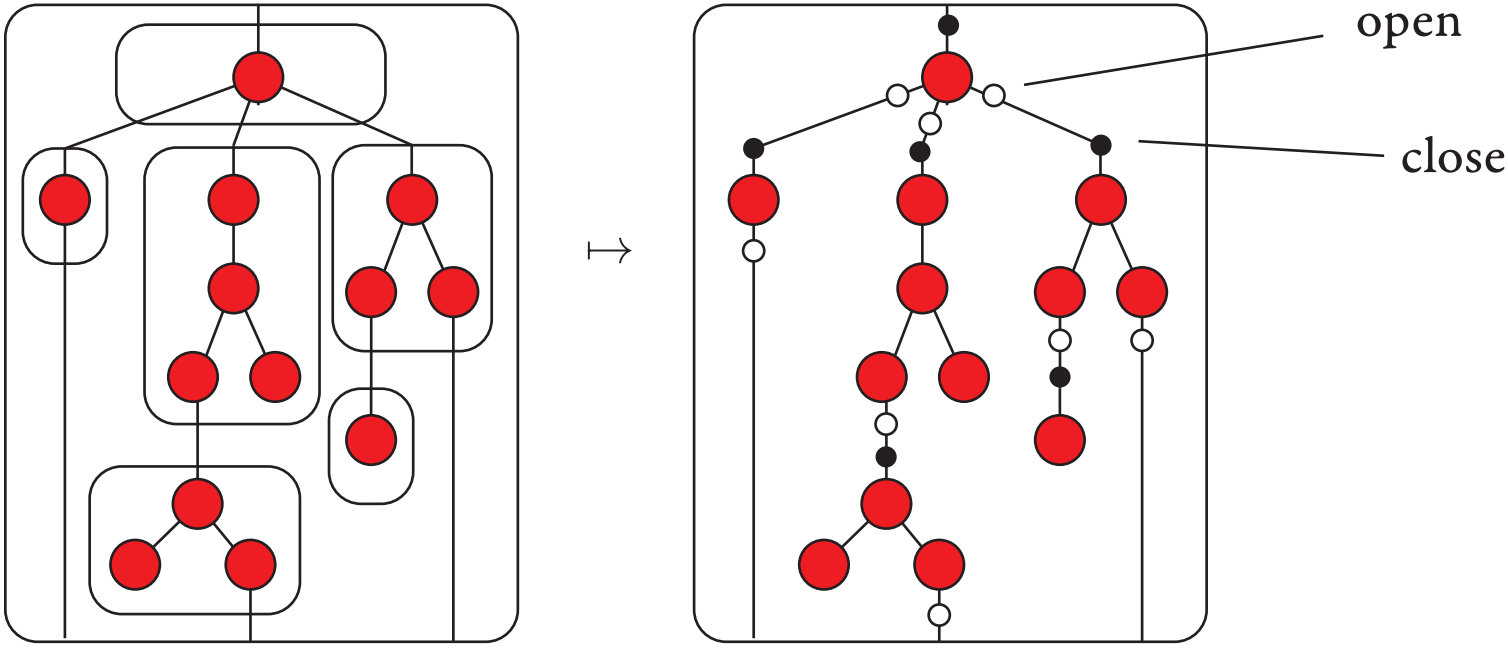
twist of port 1

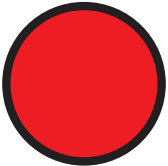


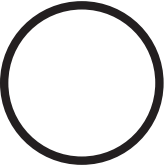


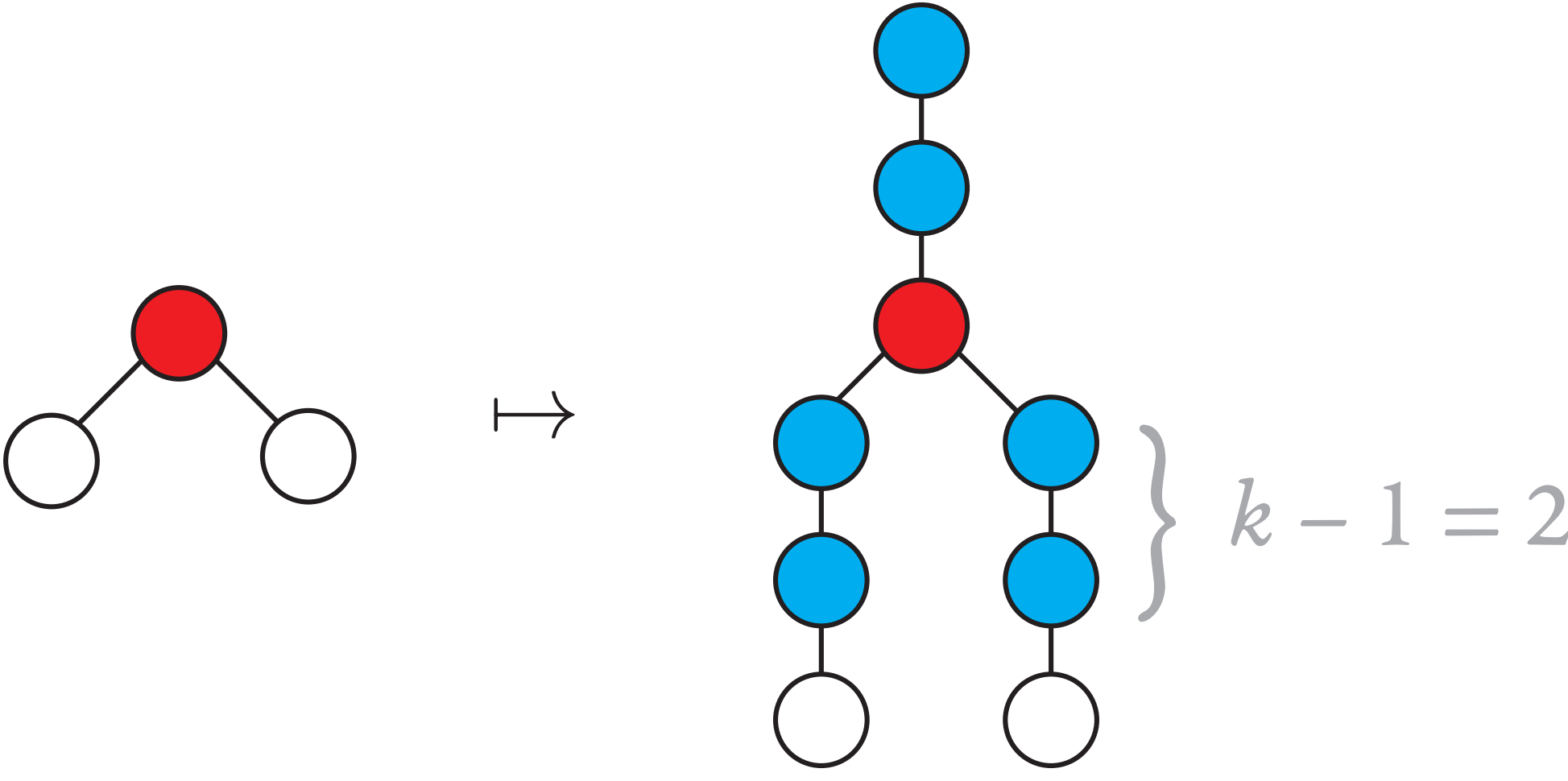


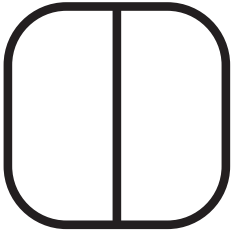

no sensitive edges
only sensitive edges

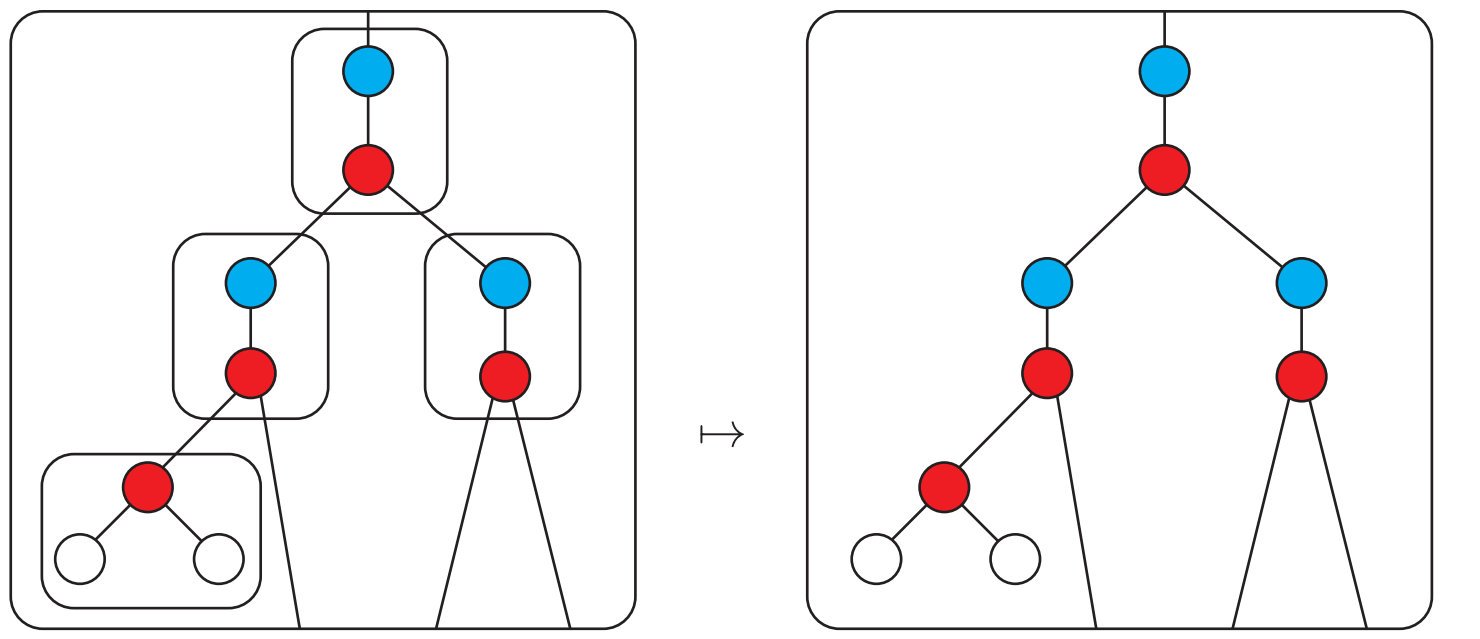

open
close

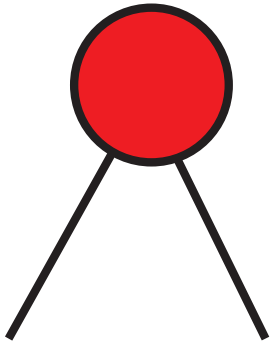


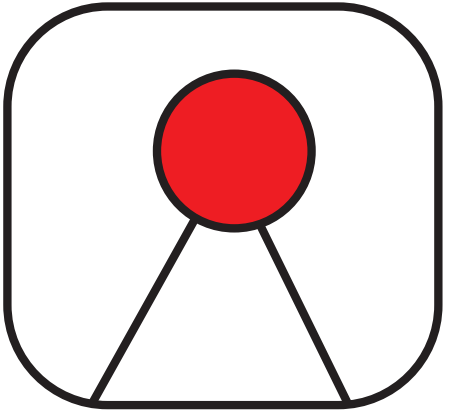


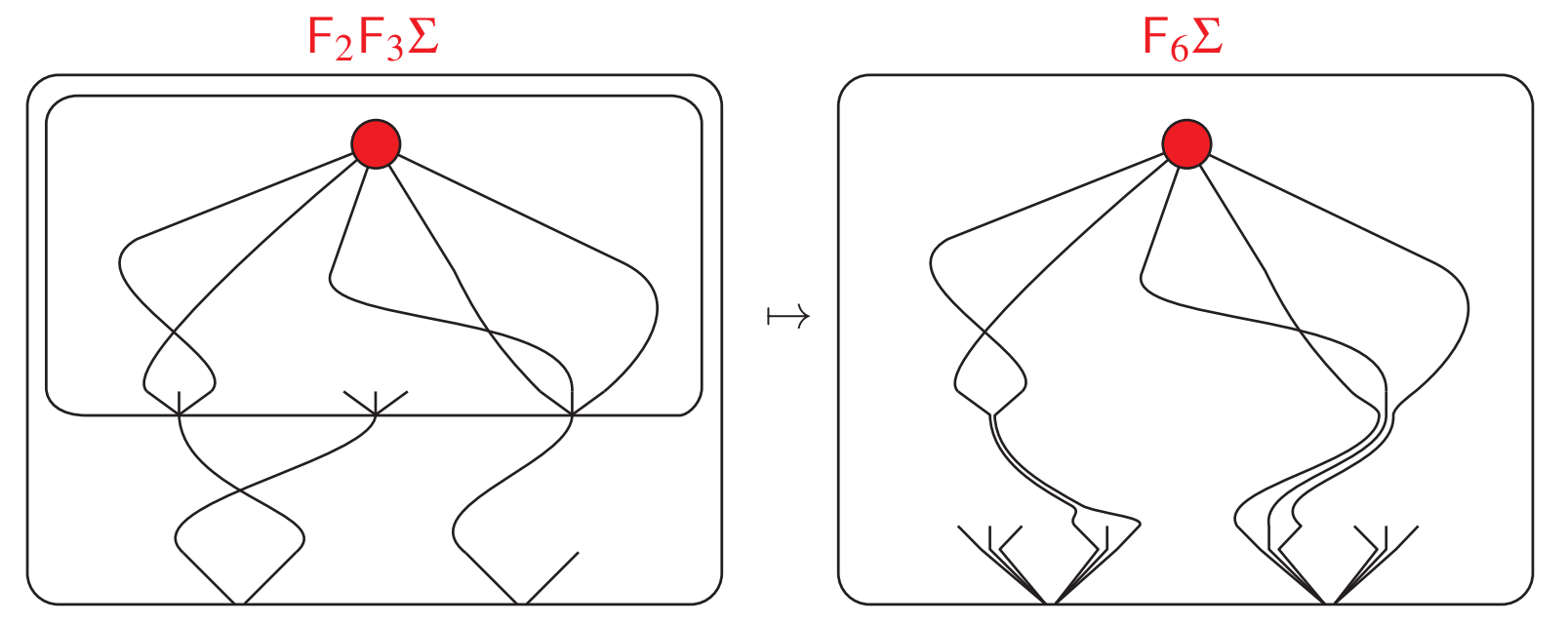

$k-1=2$

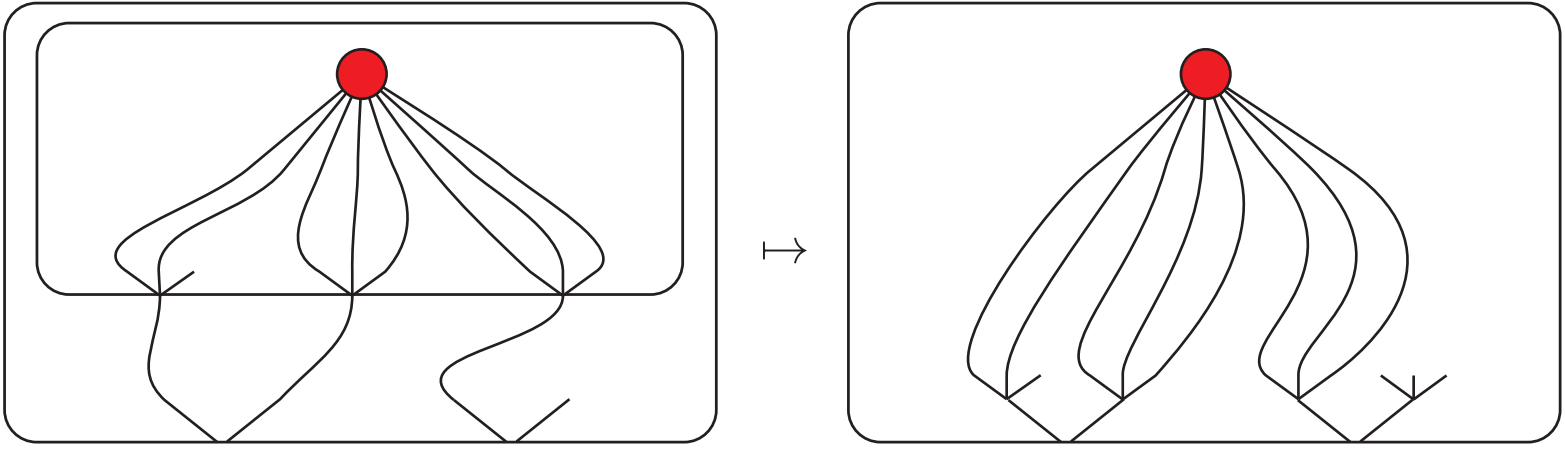


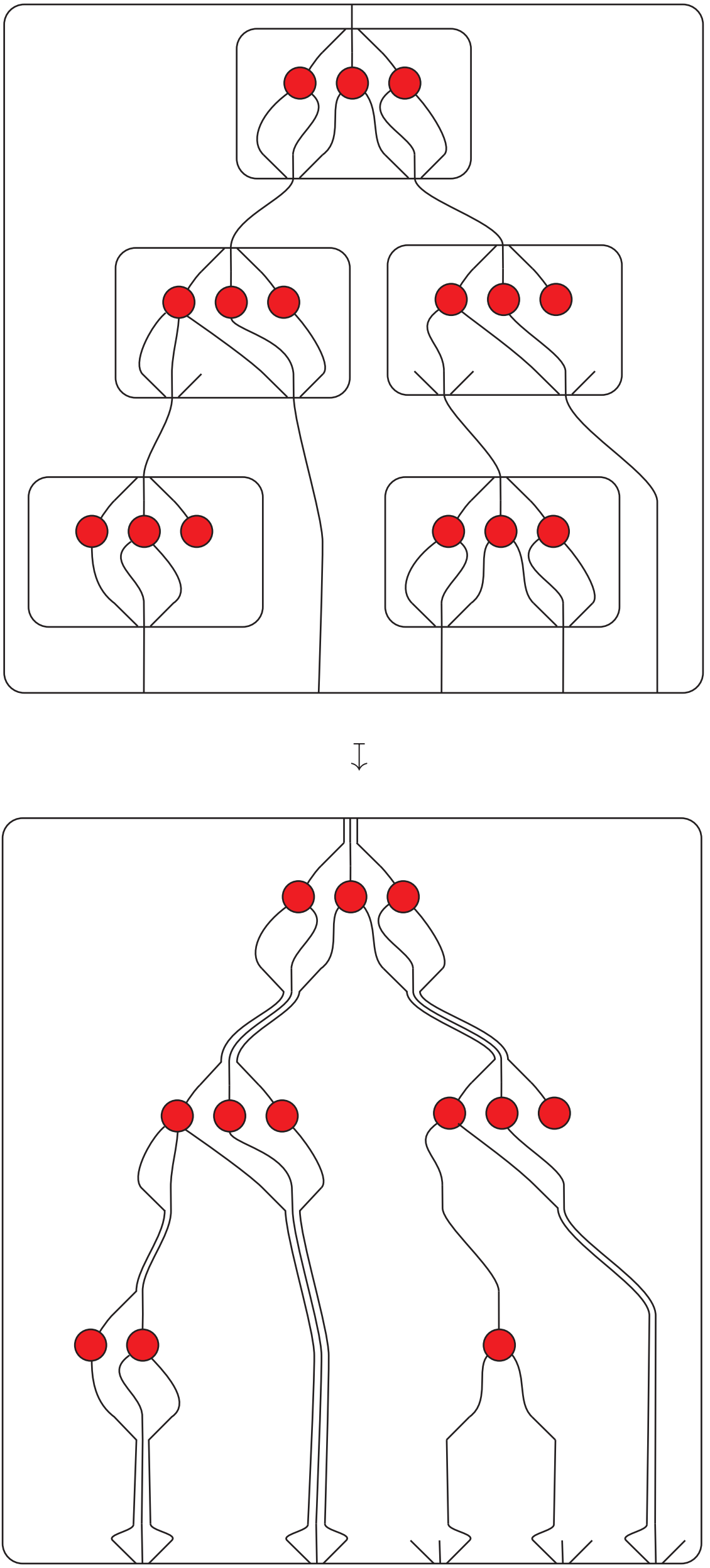




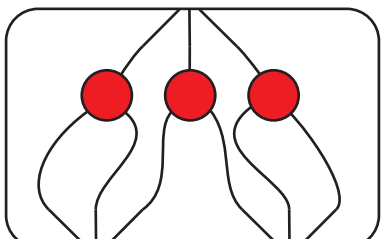



$F_2F_3\Sigma$
$\mapsto$
$F_6\Sigma$

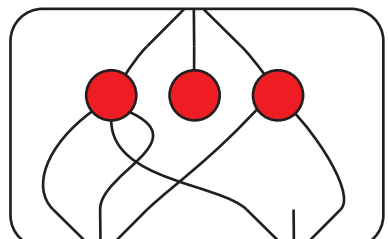




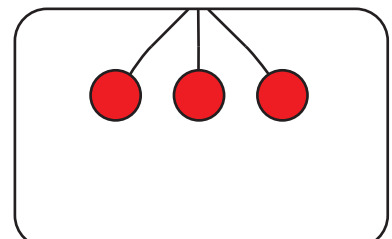
arity 2

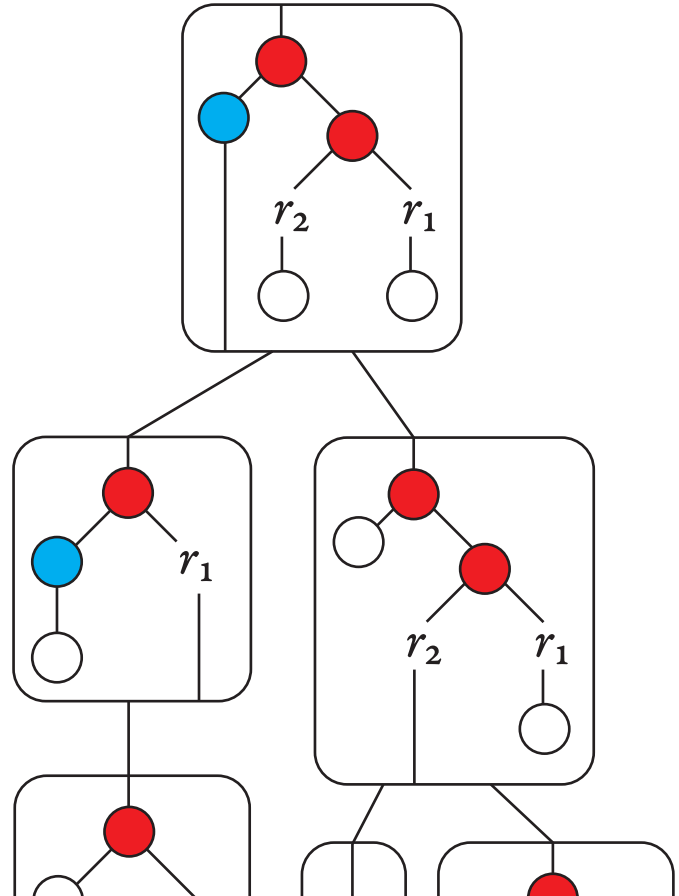


arity 2



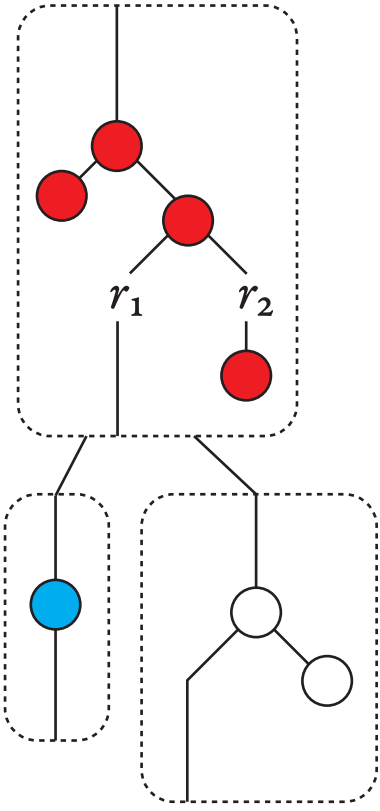
arity 0



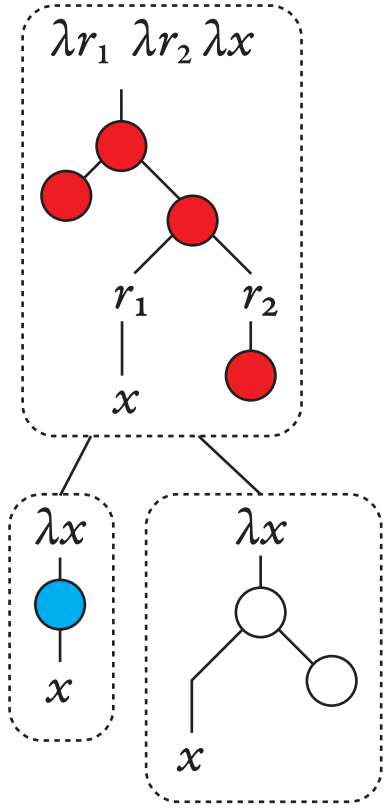

r_2
r_1
r_1
r_2
r_1

tree of register updates

λ-term

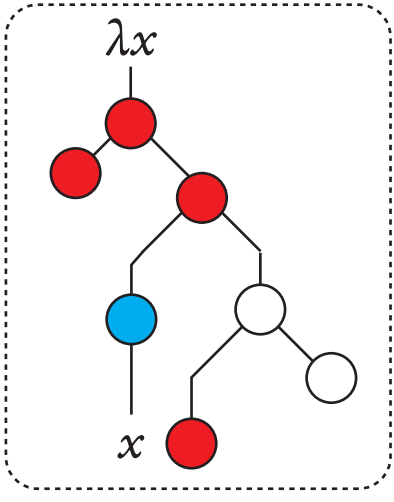


represent
as a λ-term
$\mapsto$

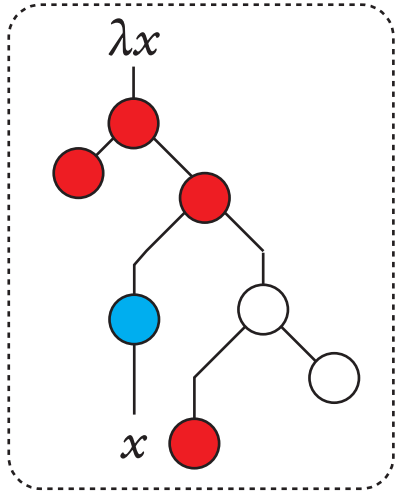


evaluate ↧

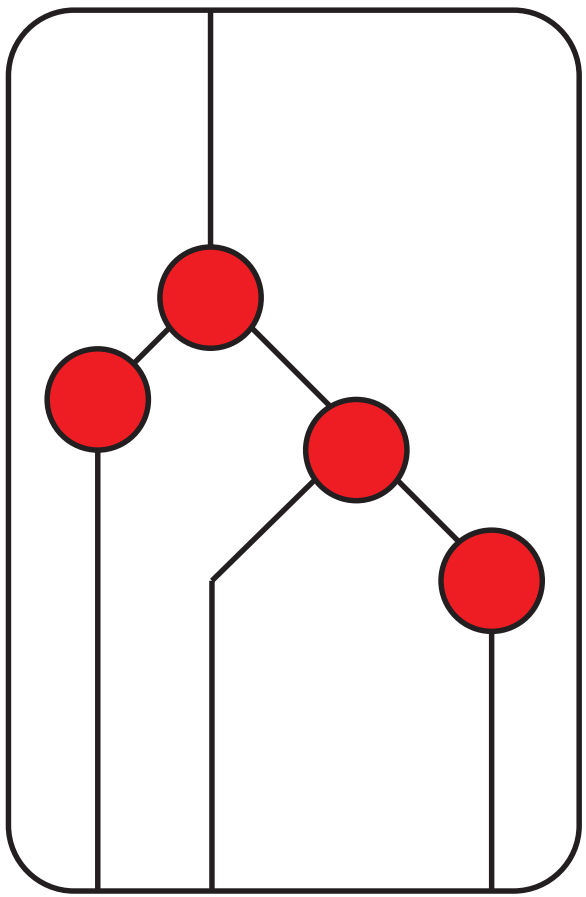
↧ evaluate



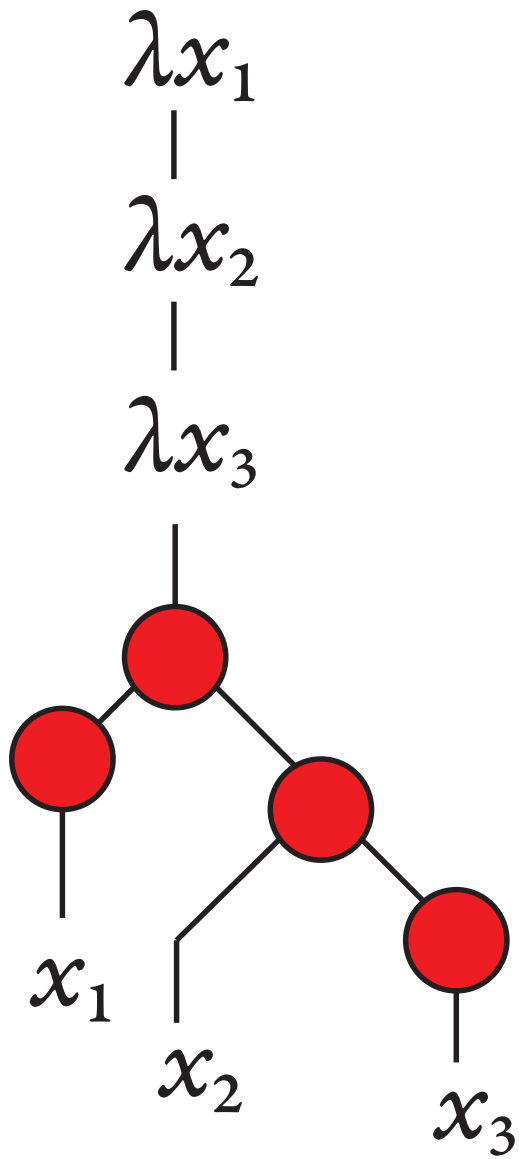
represent
as a λ-term
$\mapsto$

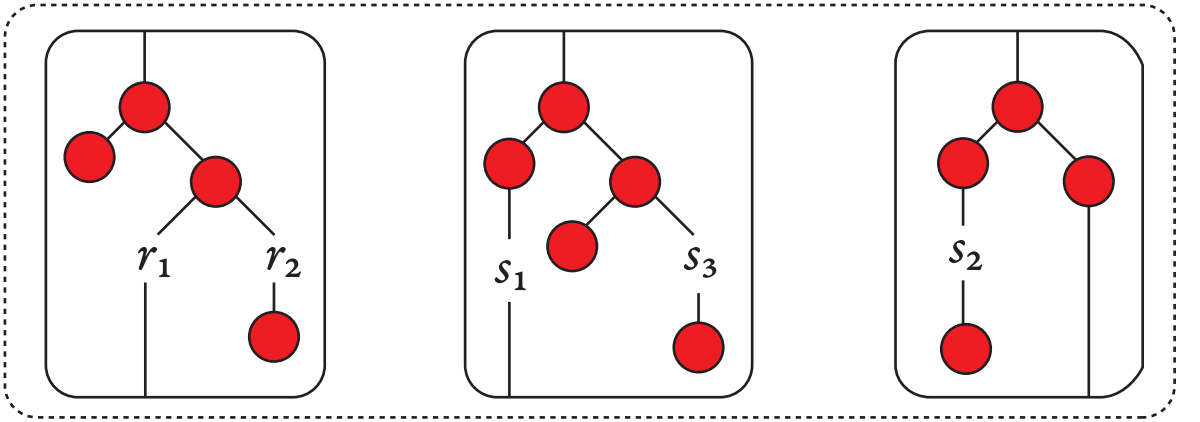


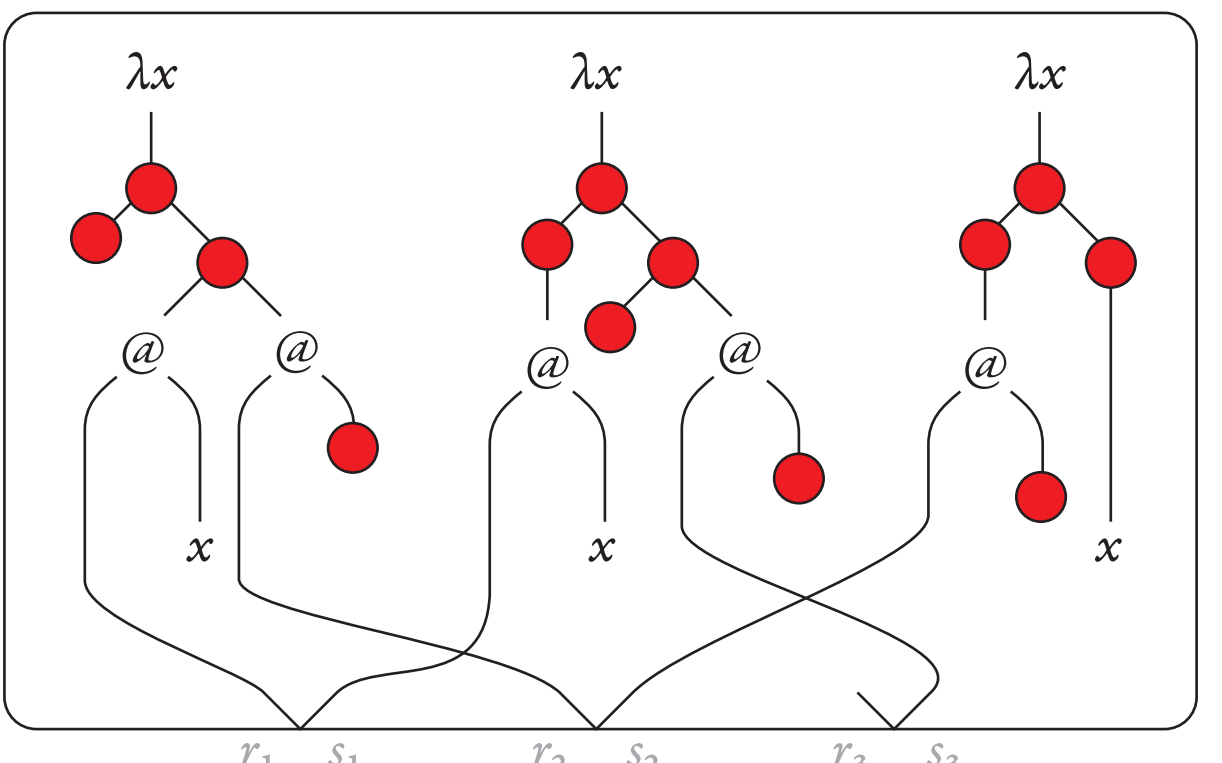
a term



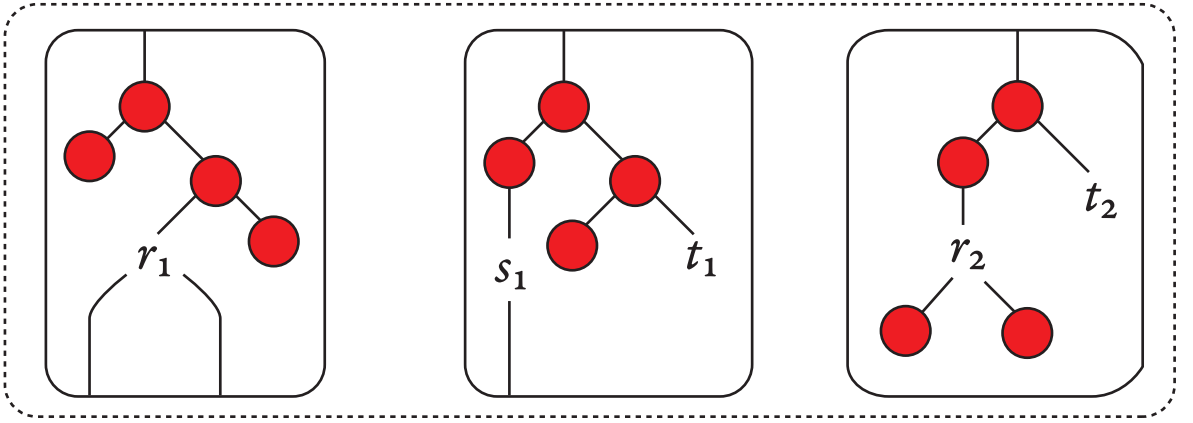
its λ-representation



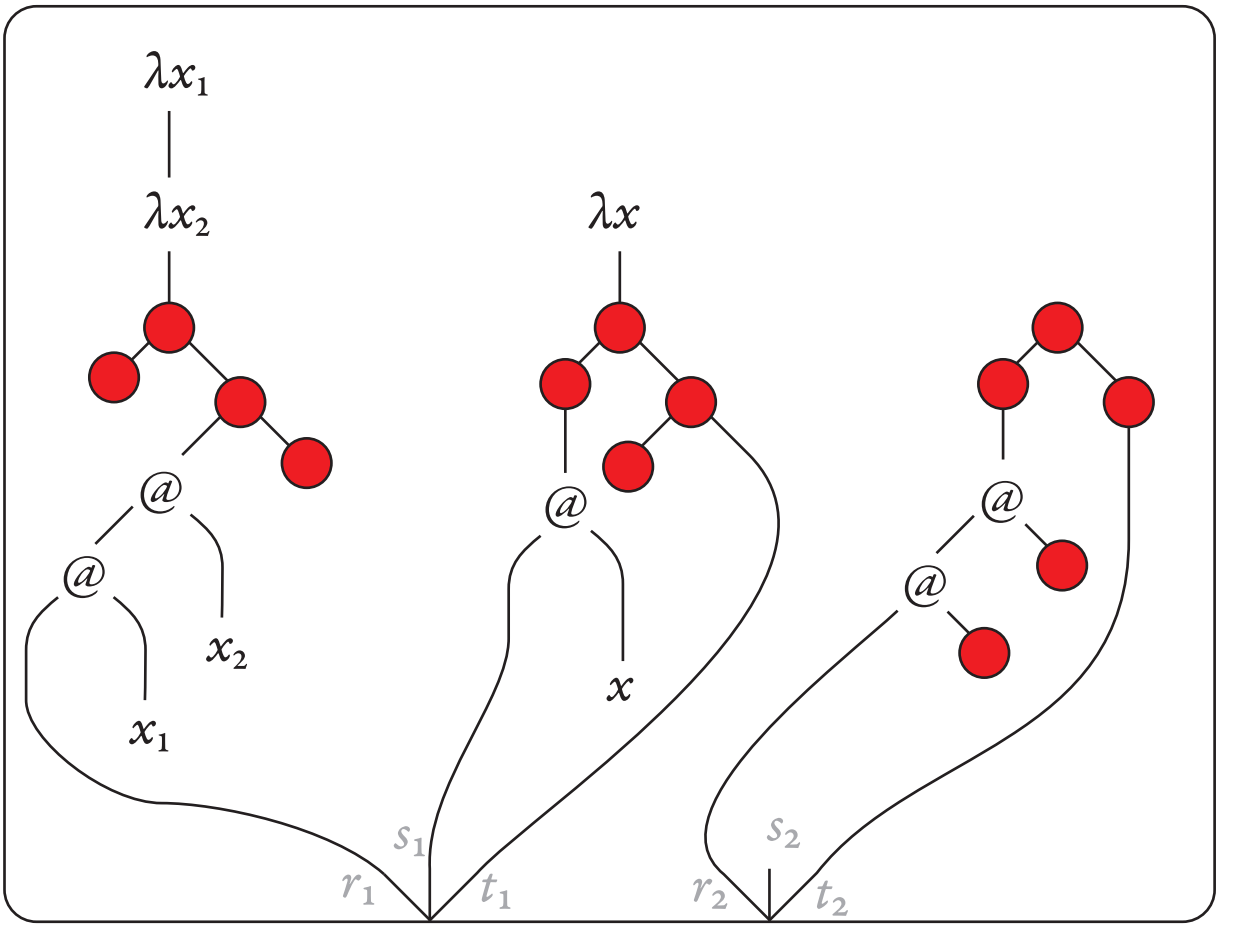

r_1
r_2
s_1
s_3
s_2

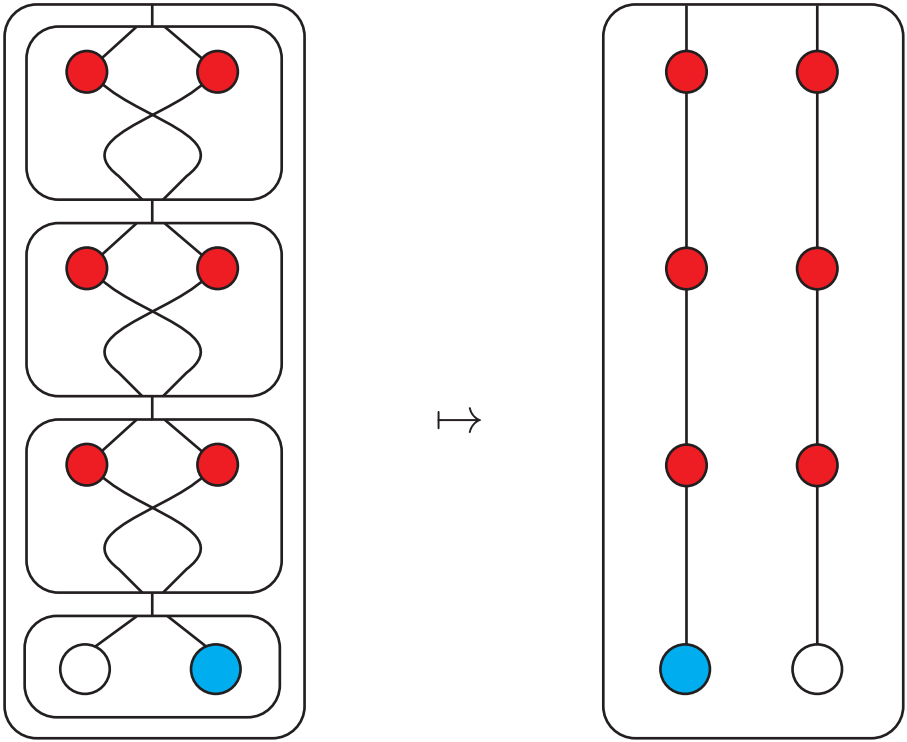

λx
λx
λx
@
@
@
@
@
x
x
x
r_1
s_1
r_2
s_2
r_3
s_3

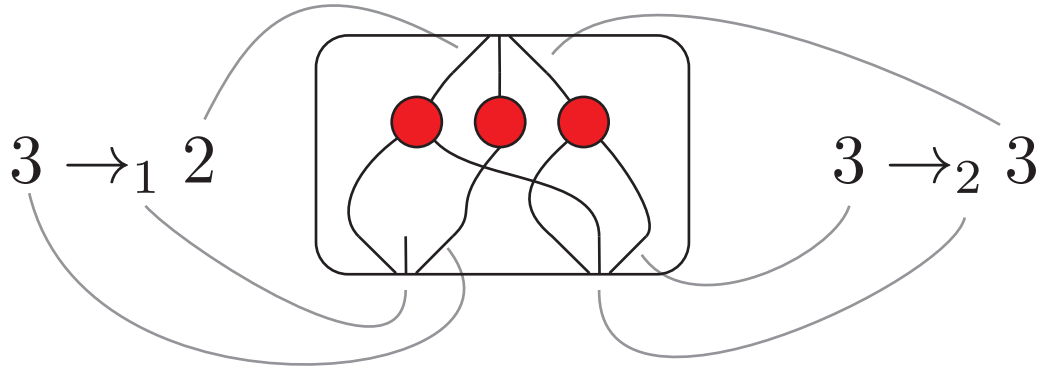
a register update

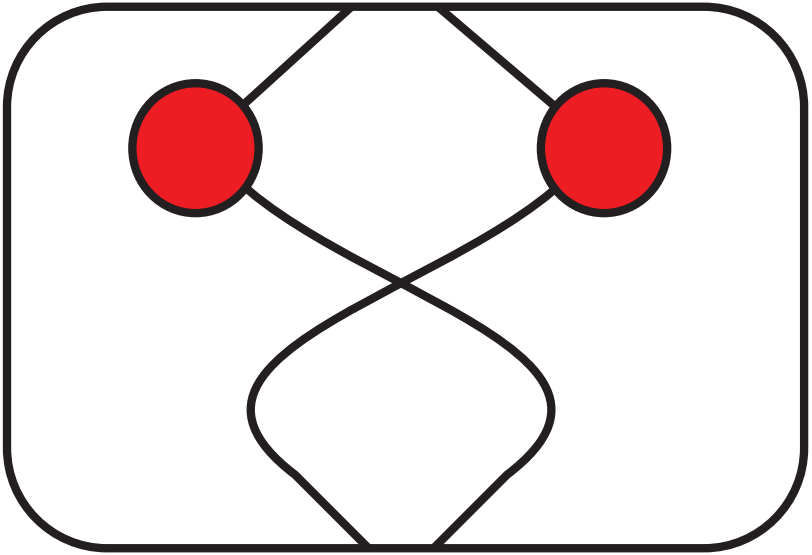


its λ-representation

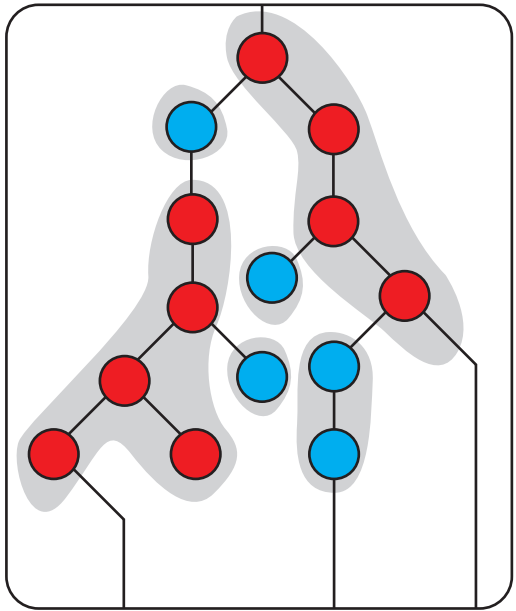




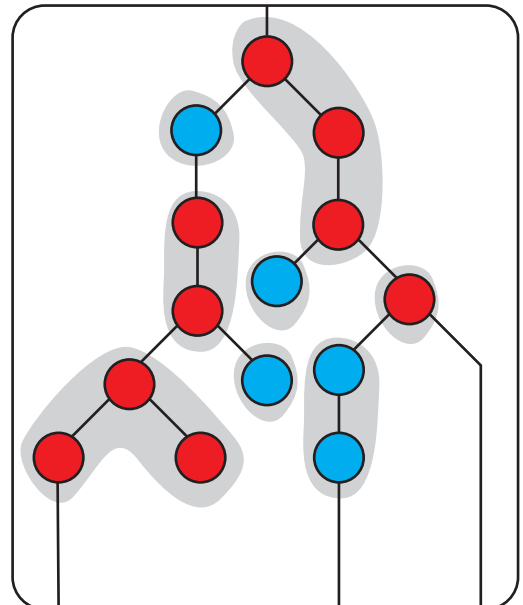

$3 \to_1 2$
$3 \to_2 3$

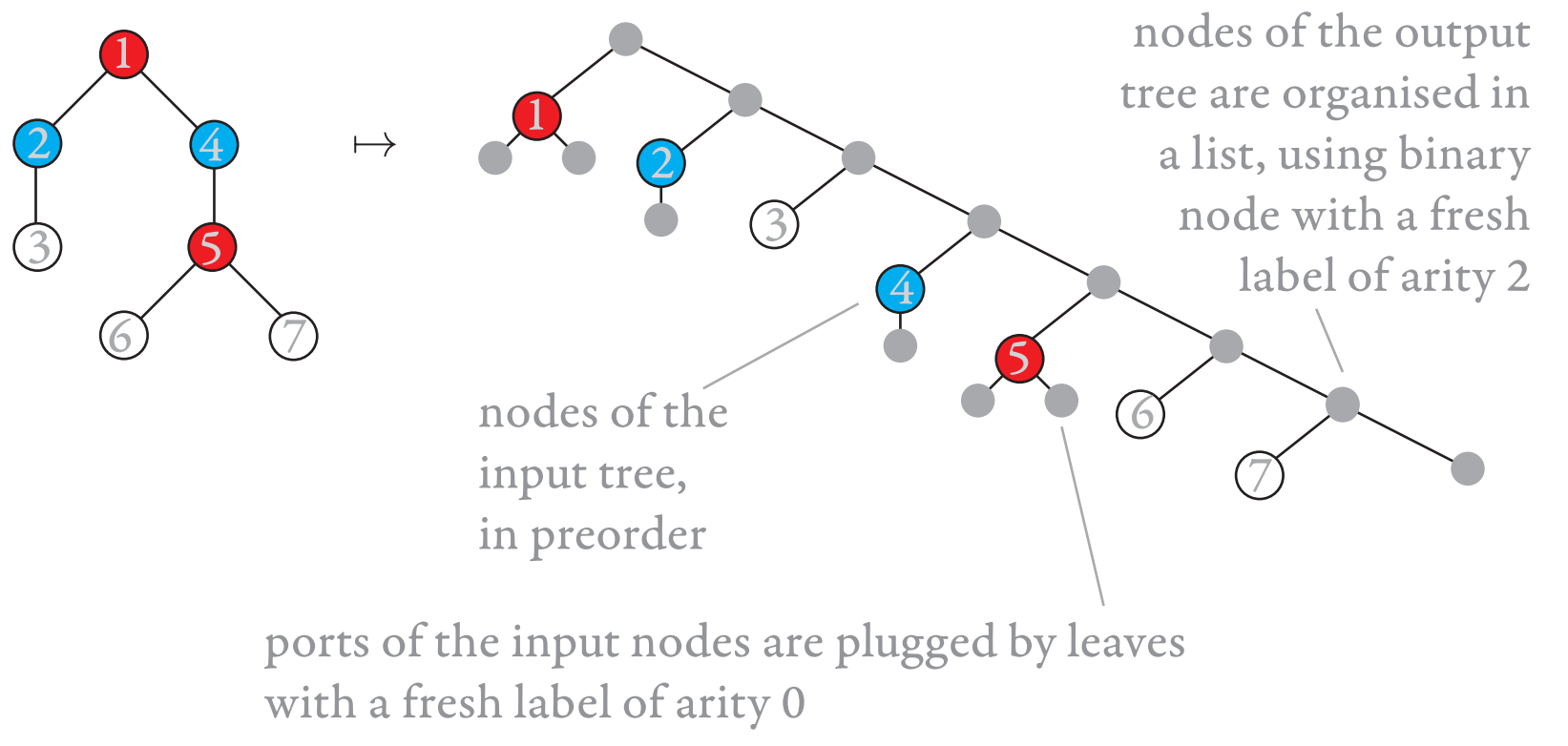


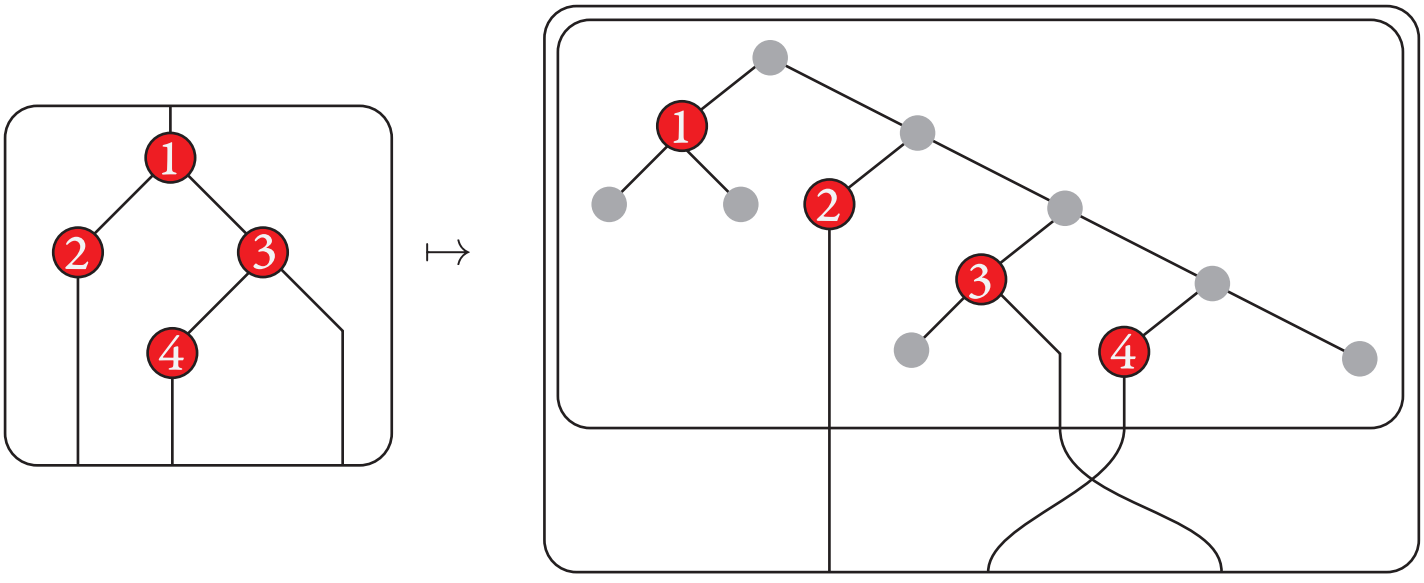
↑-equivalence



↓-equivalence




1
2
4
3
5
6
7
↦
1
2
3
4
5
6
7
nodes of the output tree are organised in a list, using binary node with a fresh label of arity 2
nodes of the input tree, in preorder
ports of the input nodes are plugged by leaves with a fresh label of arity 0


1
2
3
4
↦
1
2
3
4