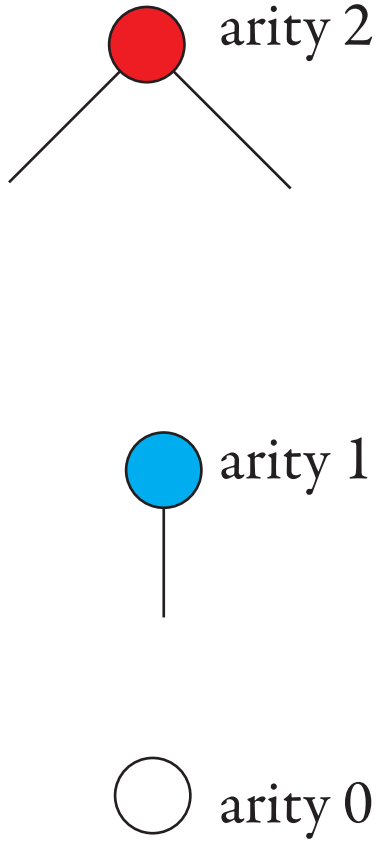
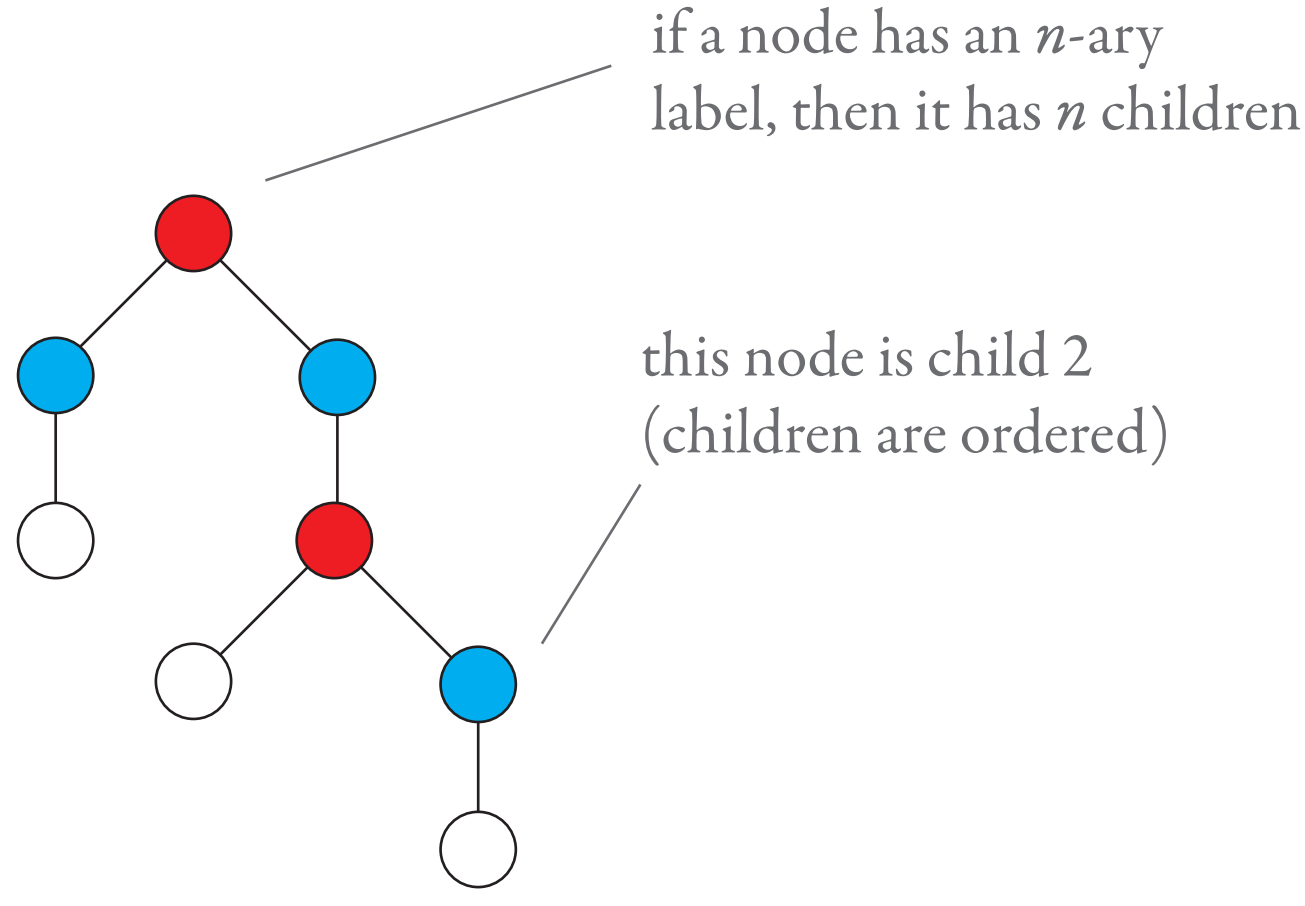
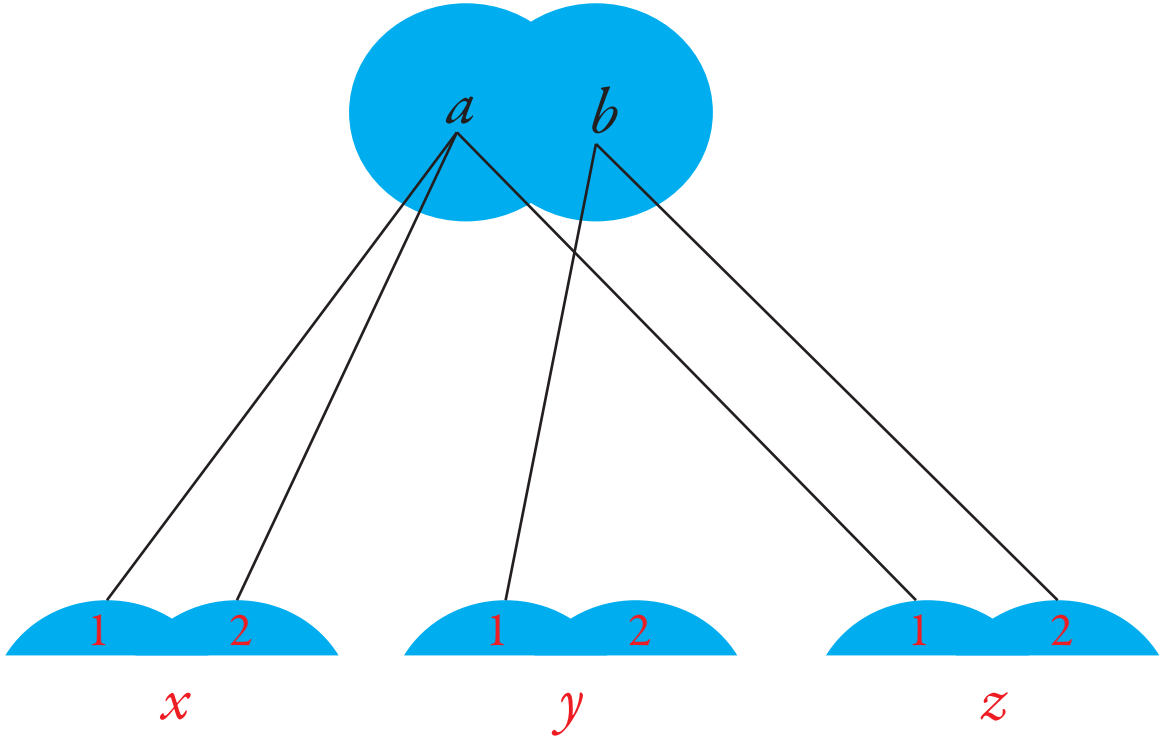
A ranked alphabet Σ

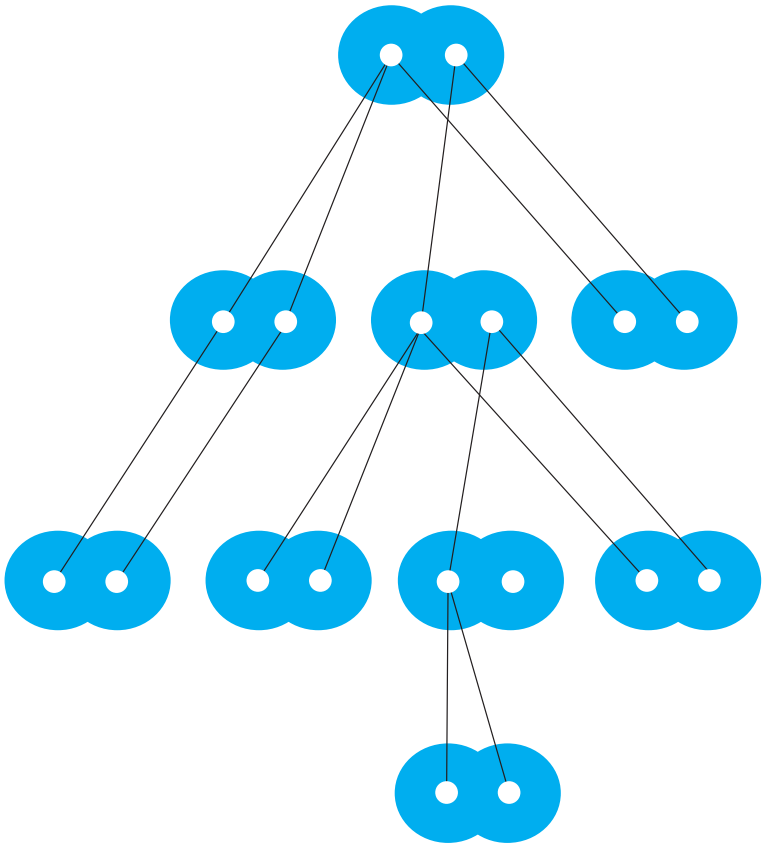


A tree over Σ

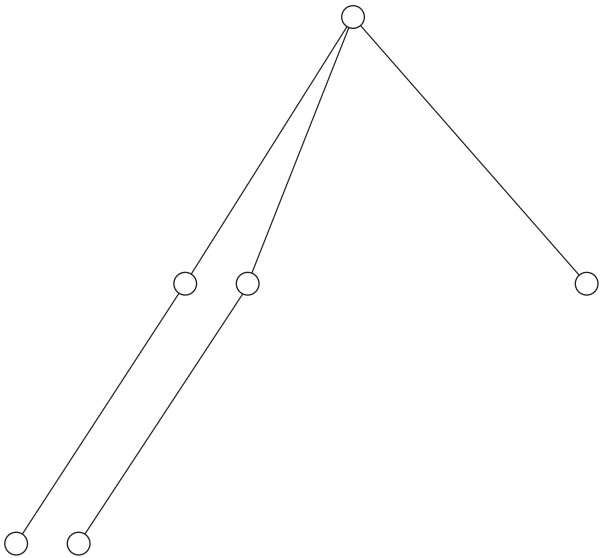



a
b
1
2
1
2
1
2
x
y
z

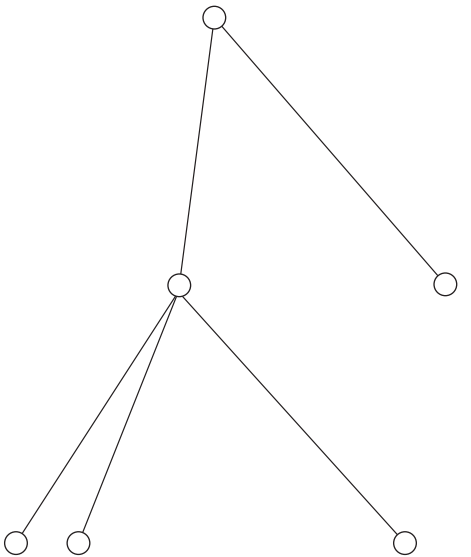
A tree t over $\Sigma^{[2]}$

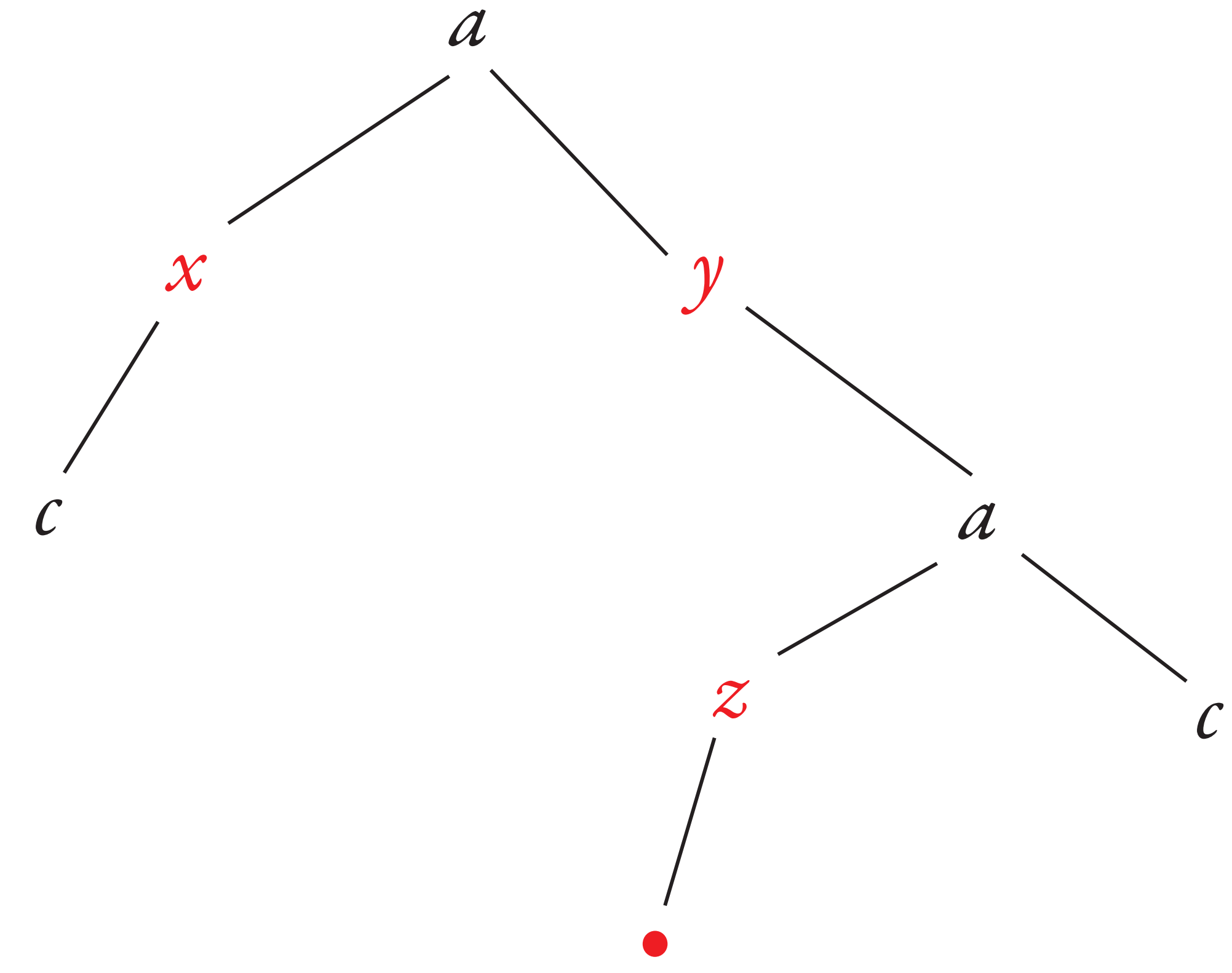


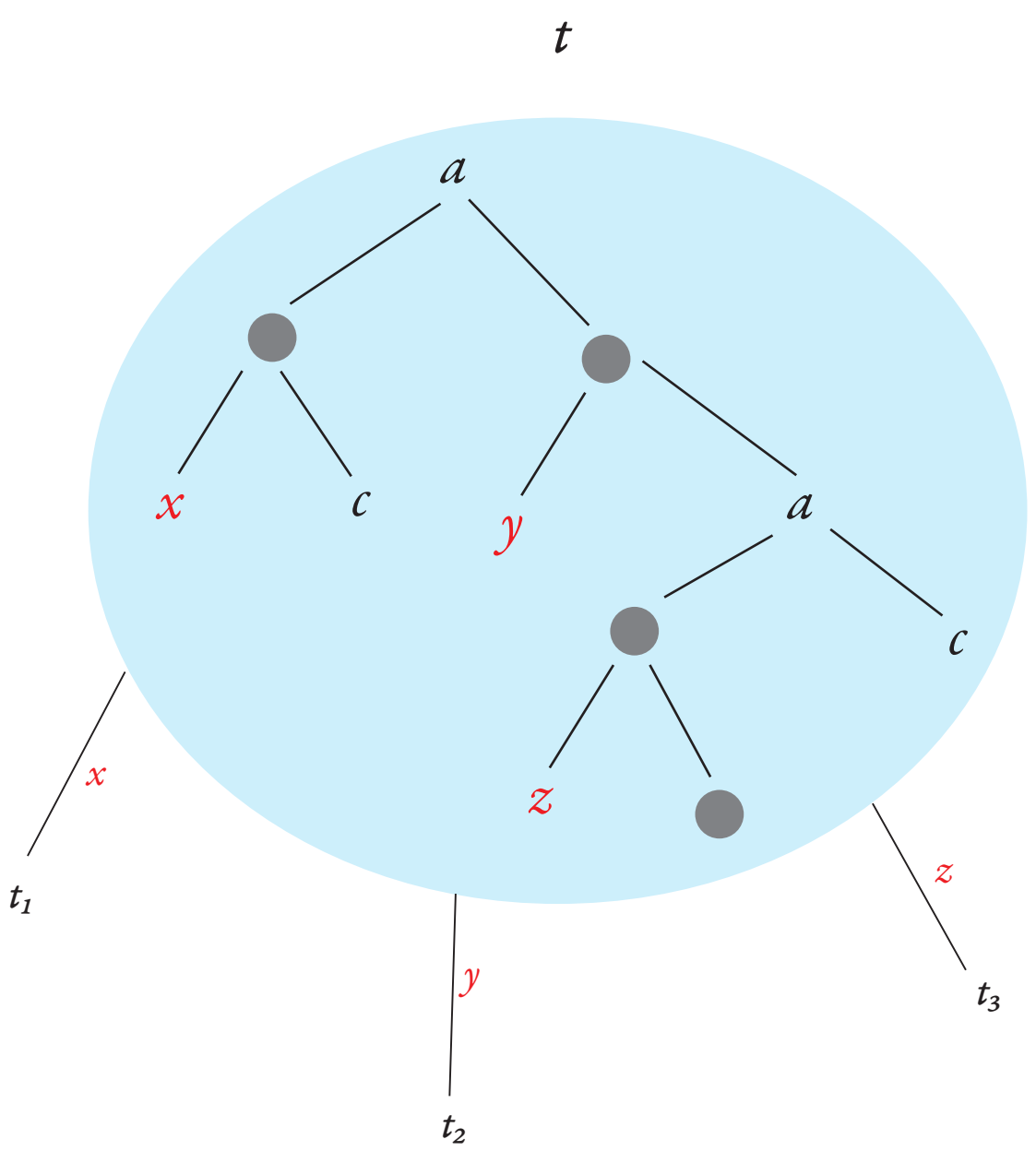
$\mathsf{unfold}_1(t)$

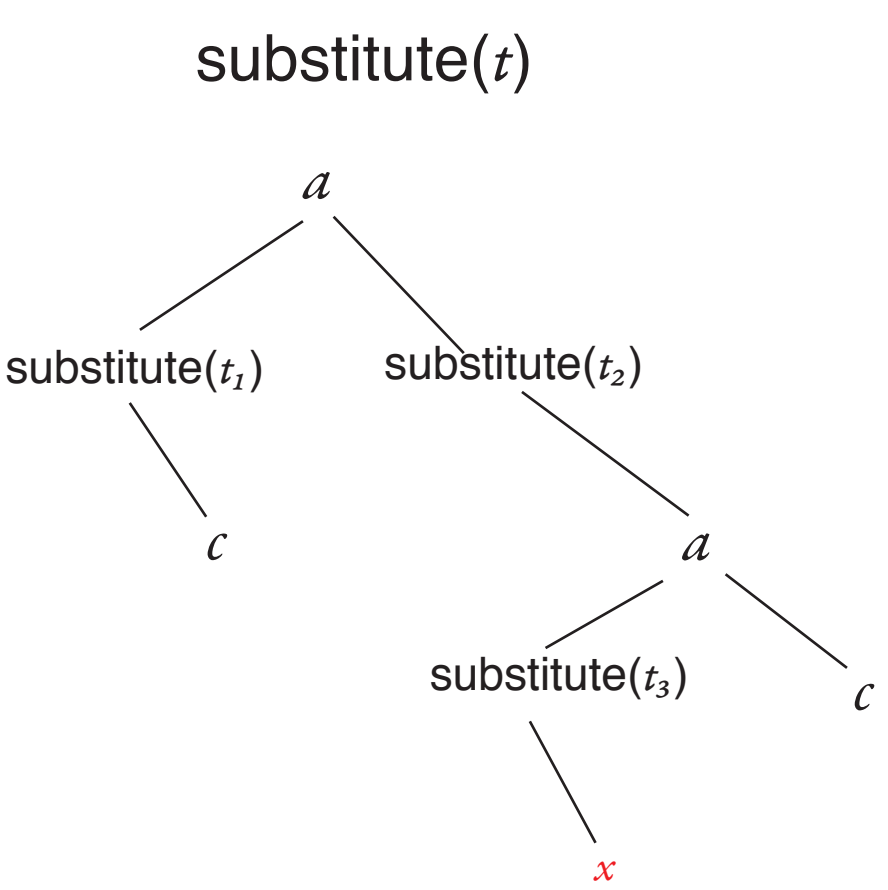


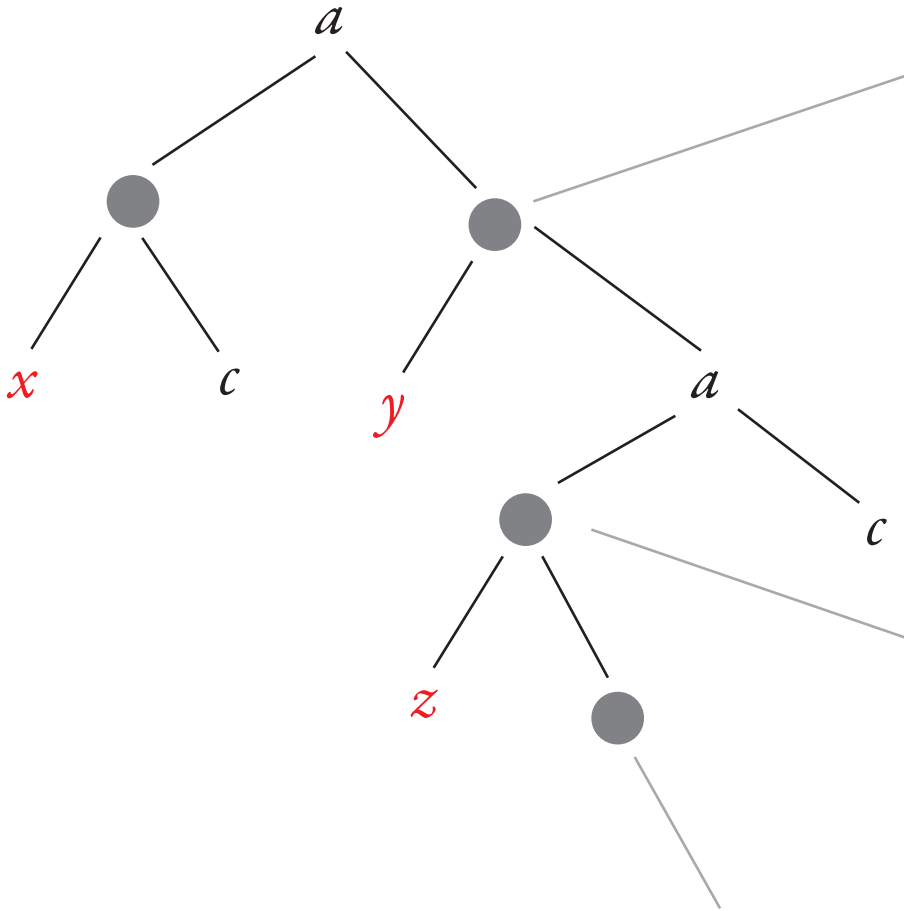
$\mathsf{unfold}_2(t)$




a
x
y
c
a
z
c


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

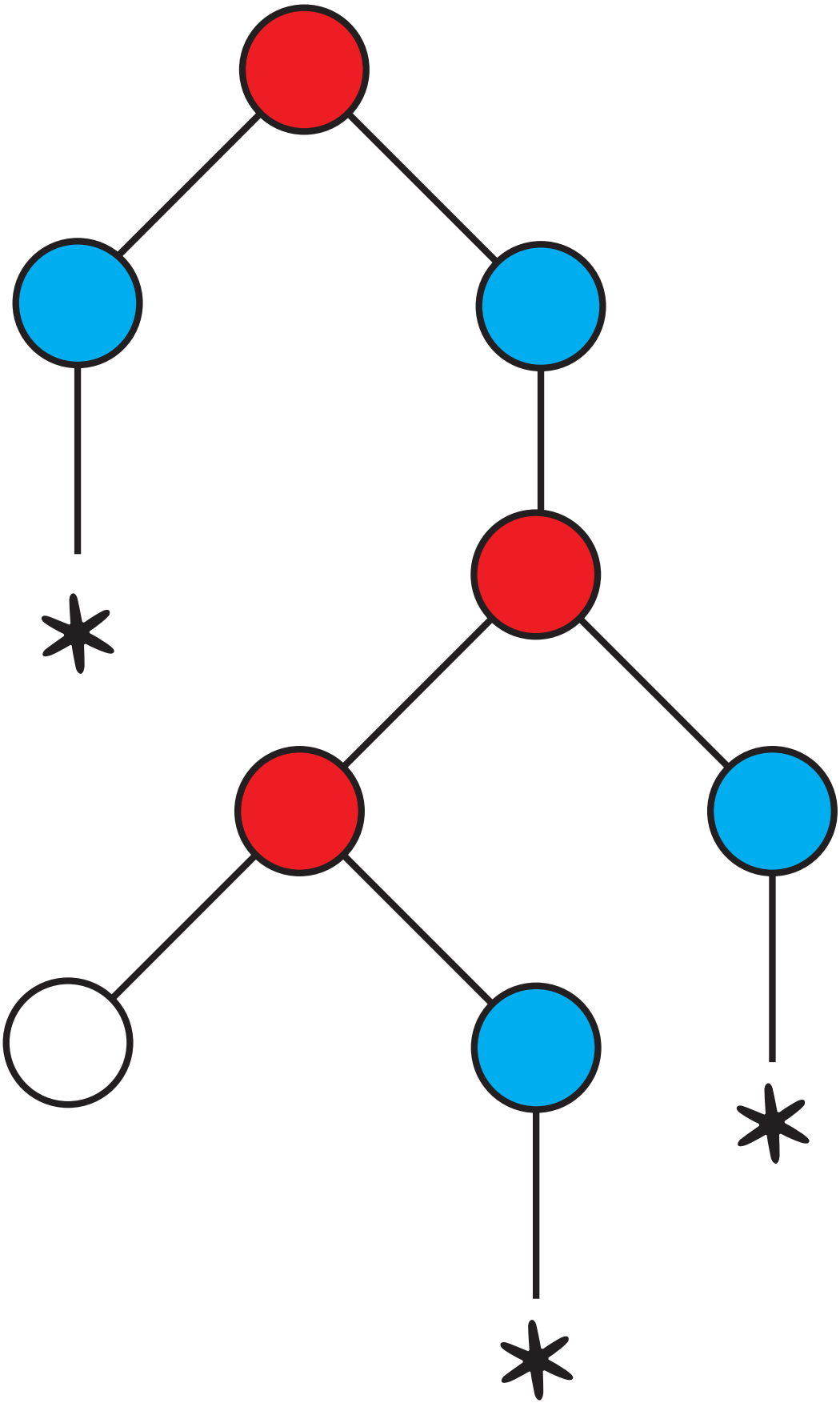

substitute(t)
a
substitute(t1)
substitute(t2)
c
a
substitute(t3)
c
x

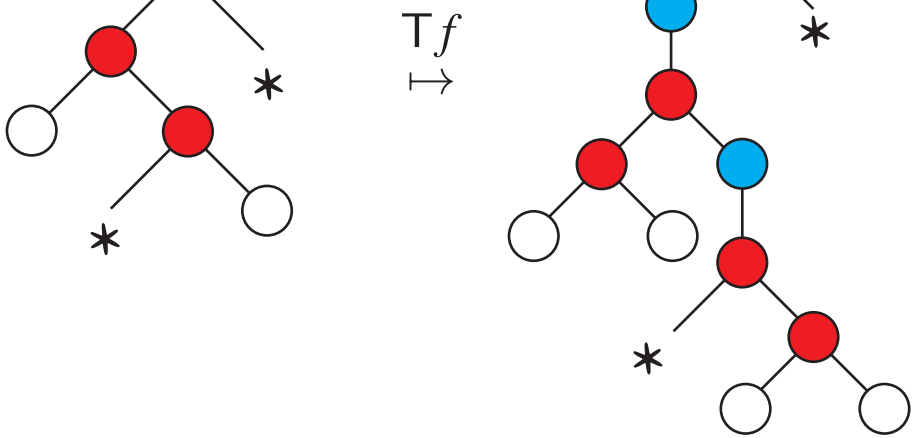


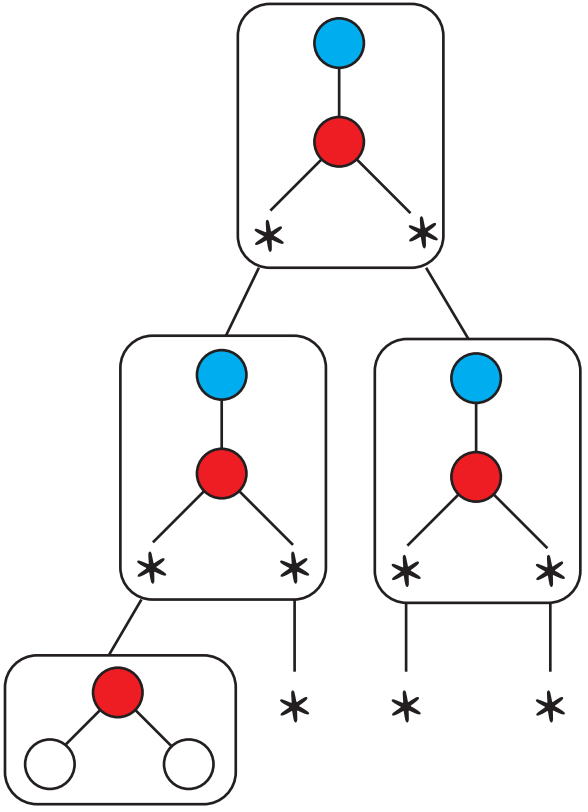
(c) for each binary letter from 1, its left child is a variable

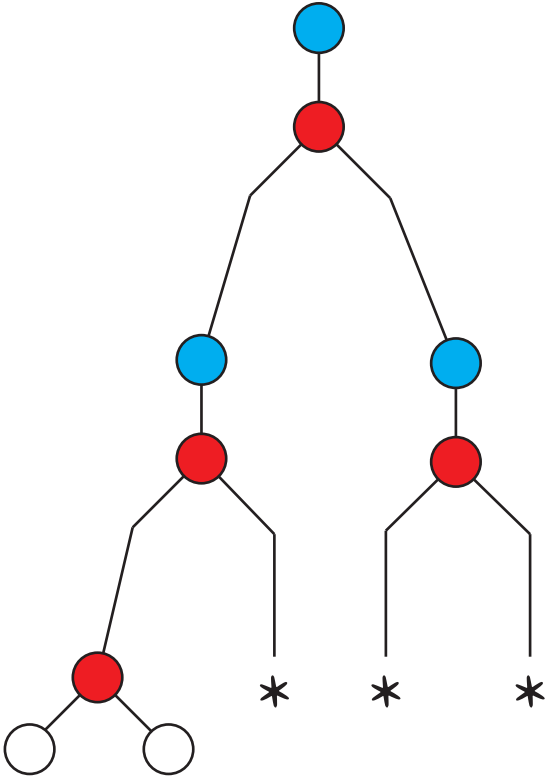
(a) only binary and nullary letters from 1 are allowed

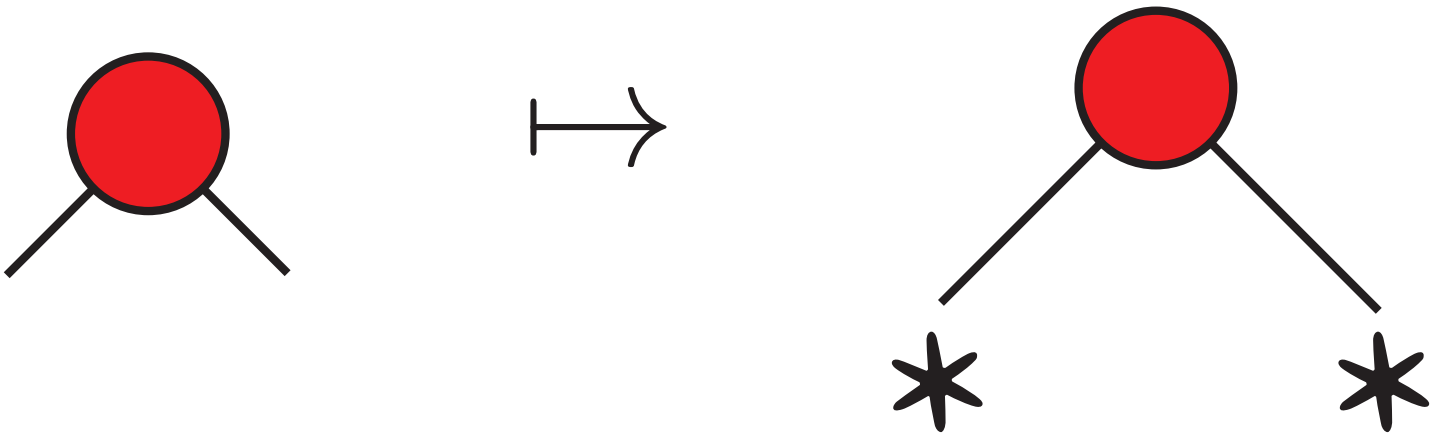
(b) a nullary letter from 1 is allowed exactly once



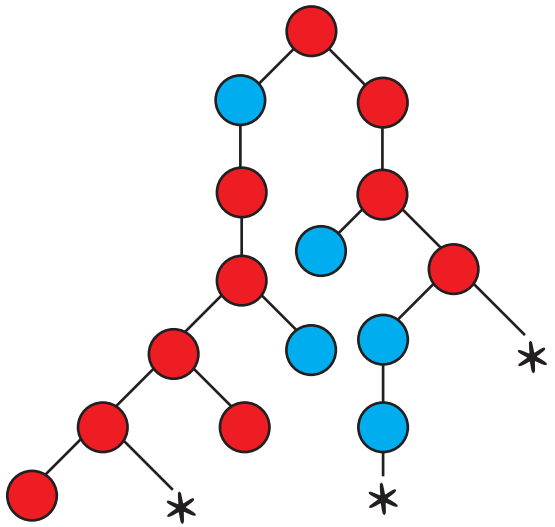

Tf
↦



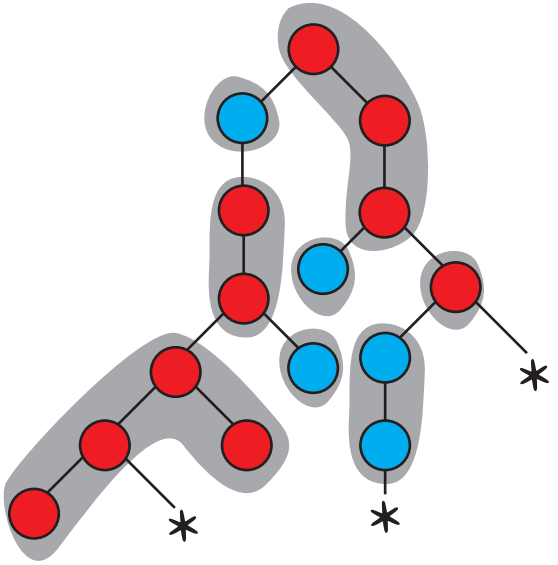




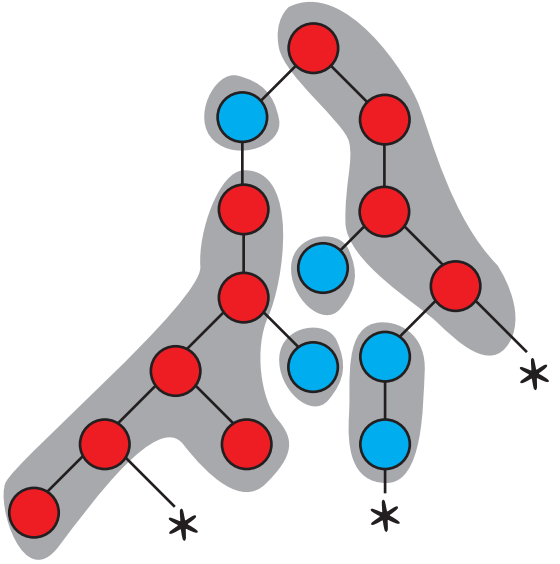
a term

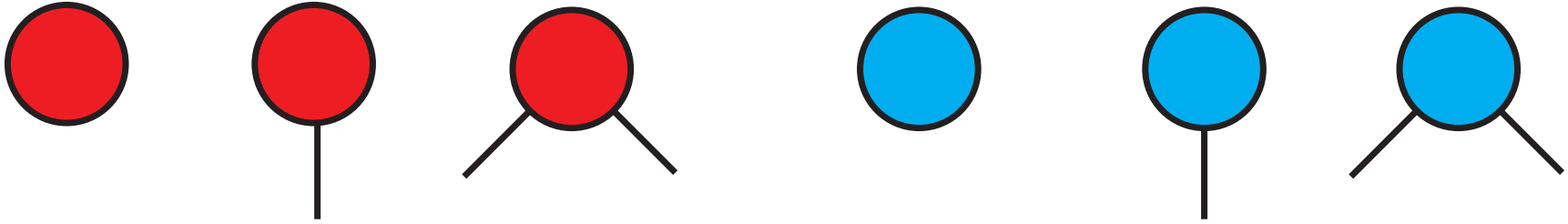


ancestor equivalence

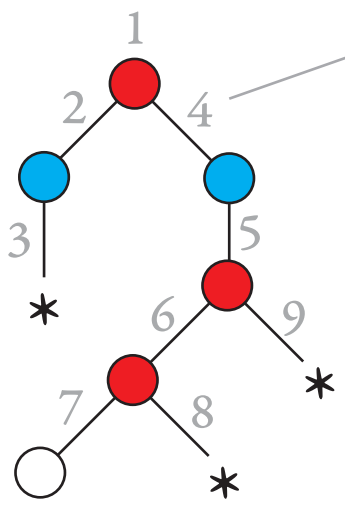


descendant equivalence



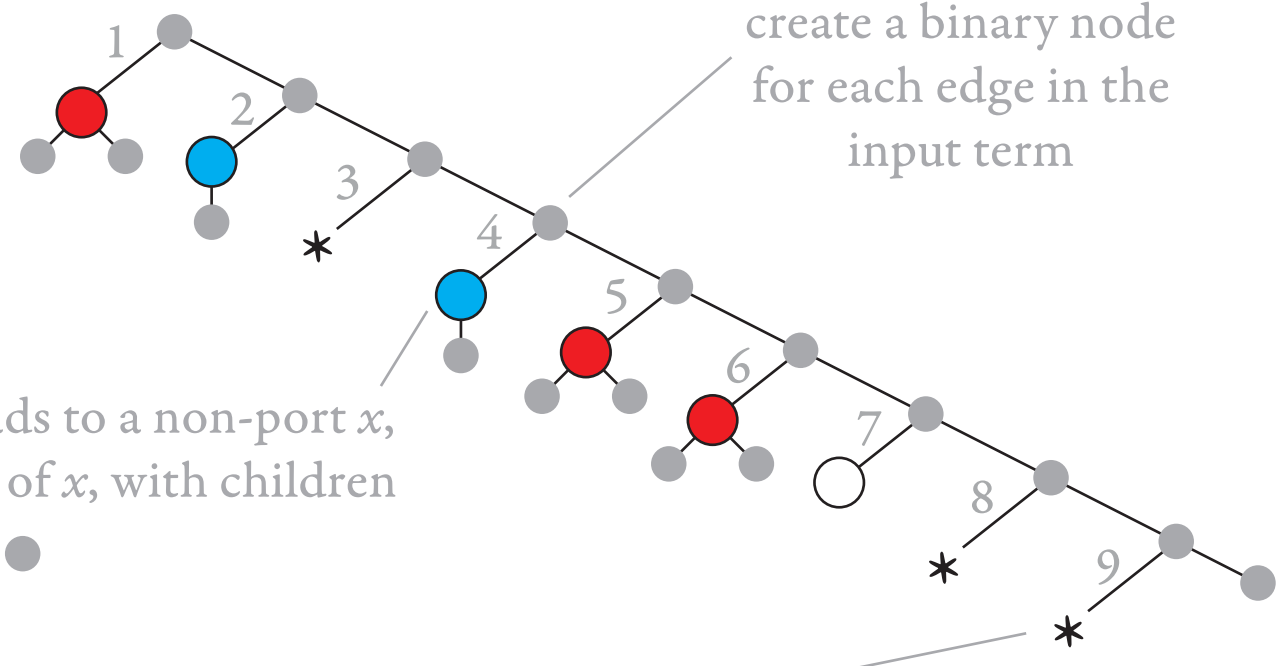


input



number the edges in the input term according to their appearance in the pre-order traversal

output

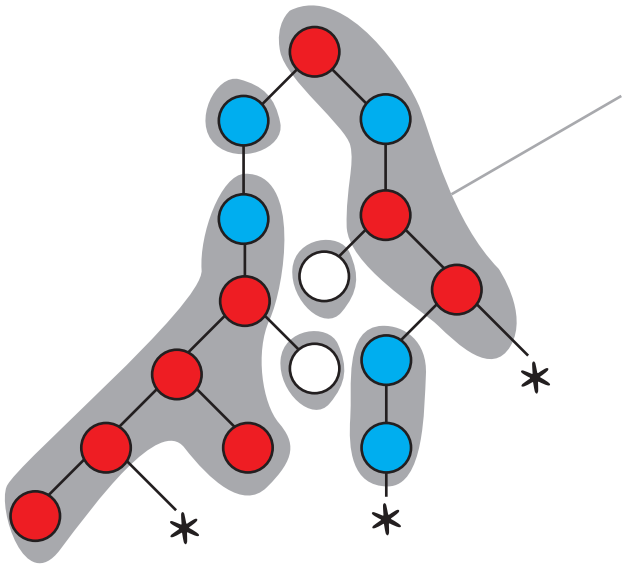


create a binary node for each edge in the input term

if an edge in the input term leads to a non-port x, in the output append the label of x, with children plugged by ●

if an edge in the input term leads to a port, keep the port

a factorisation equivalence



the corresponding factorisation term