

a ranked alphabet

arity 2



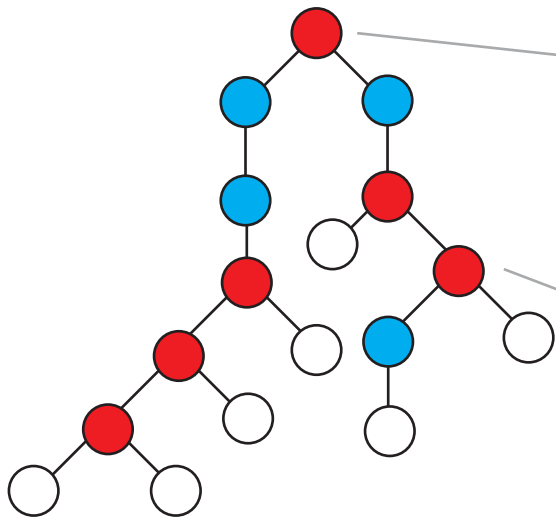
arity 1



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



$\text{unfold}_1(t)$



$\text{unfold}_2(t)$





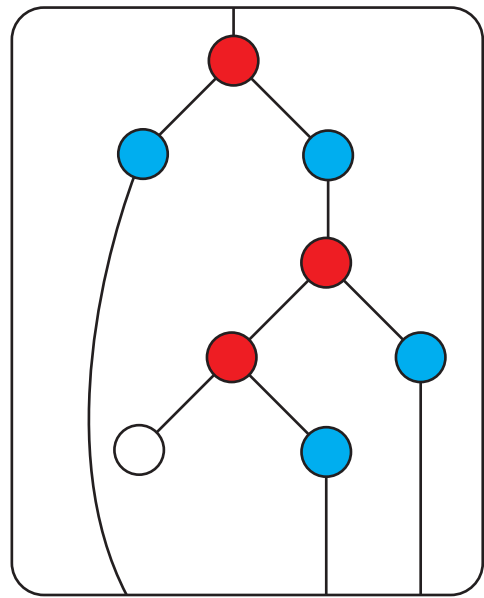
t



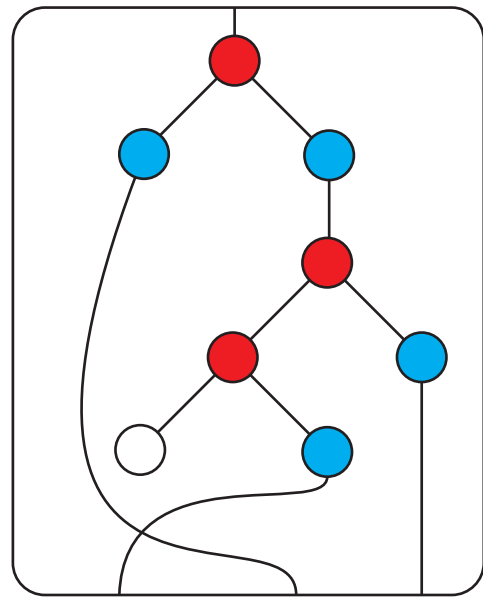
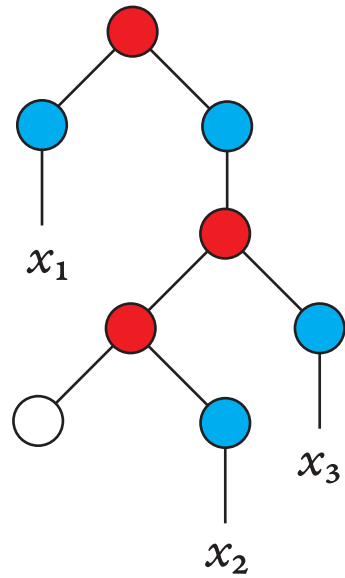
substitute(t)



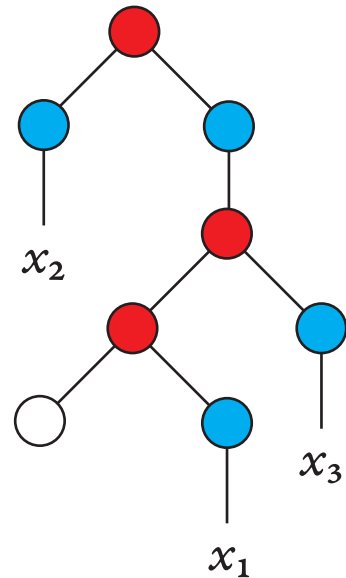


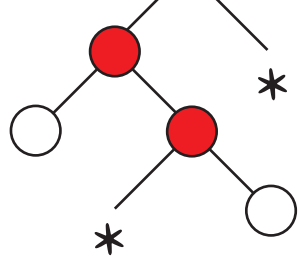


=

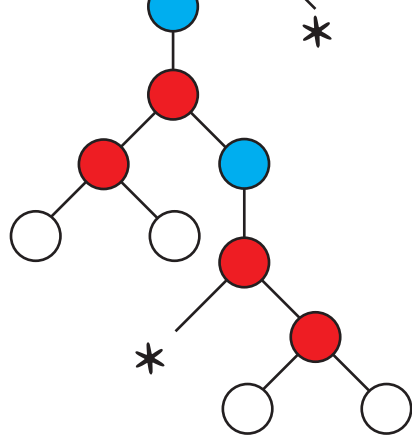


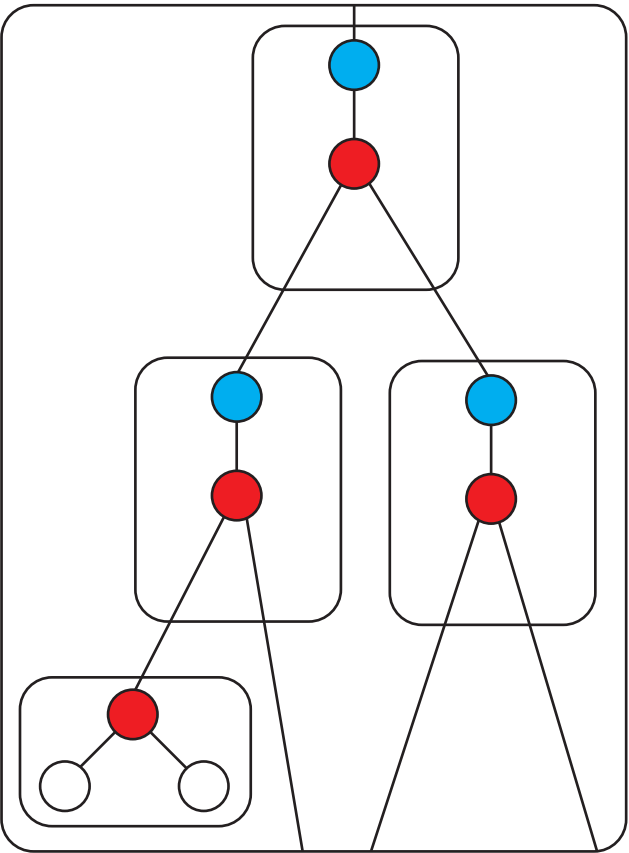
=



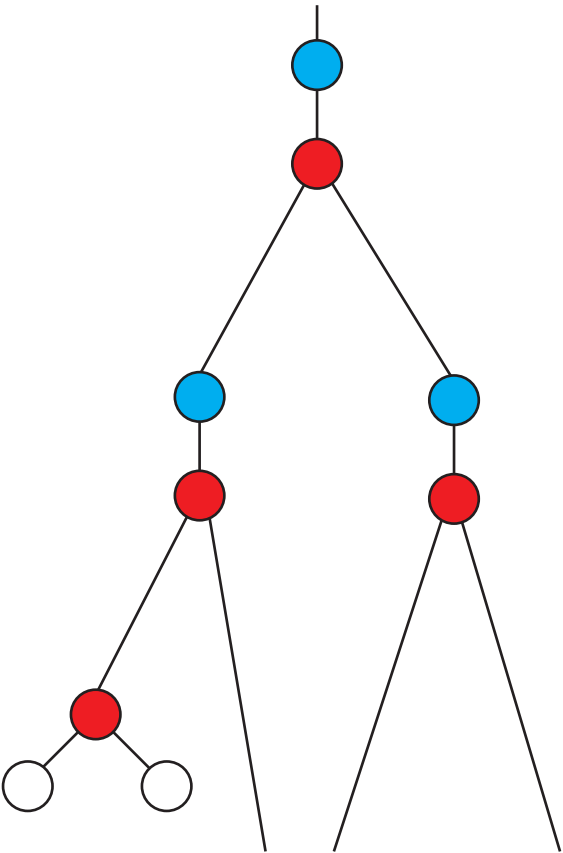


$\mathsf{T}f$
 \mapsto



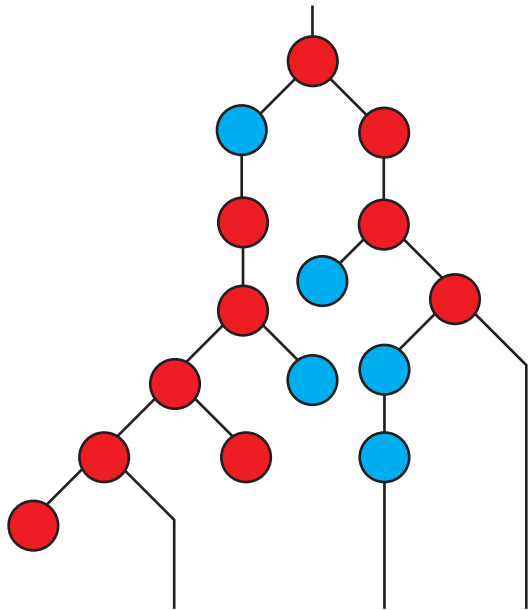


\mapsto

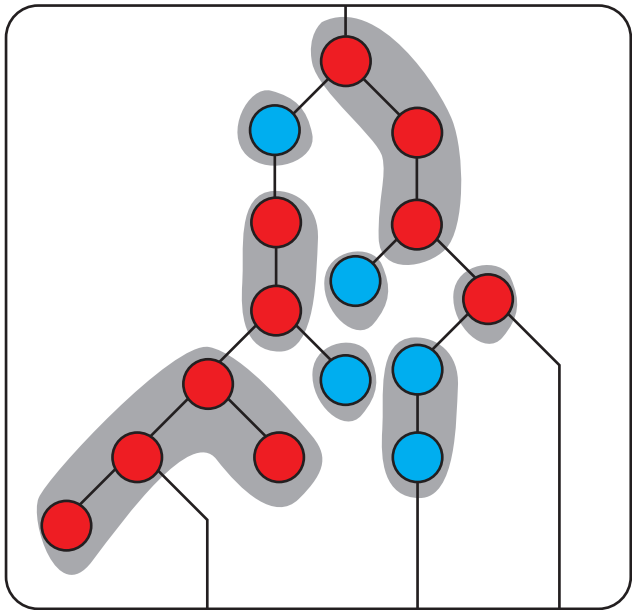




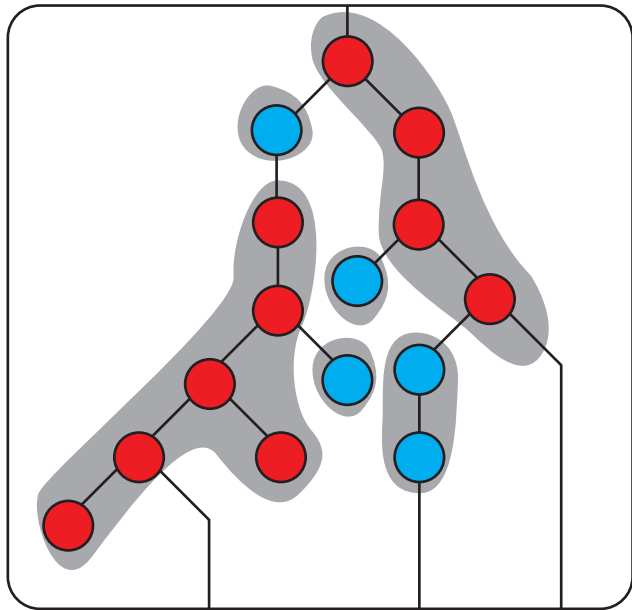
a term



ancestor equivalence

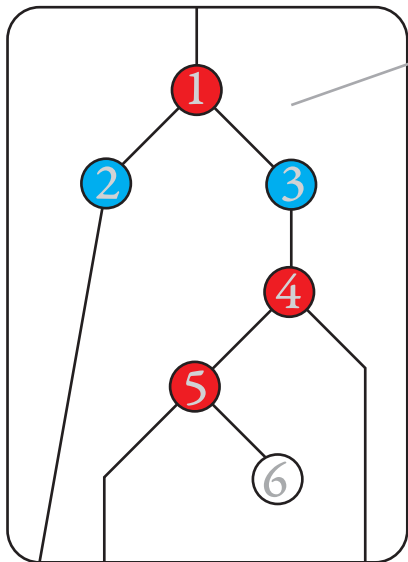


descendant equivalence





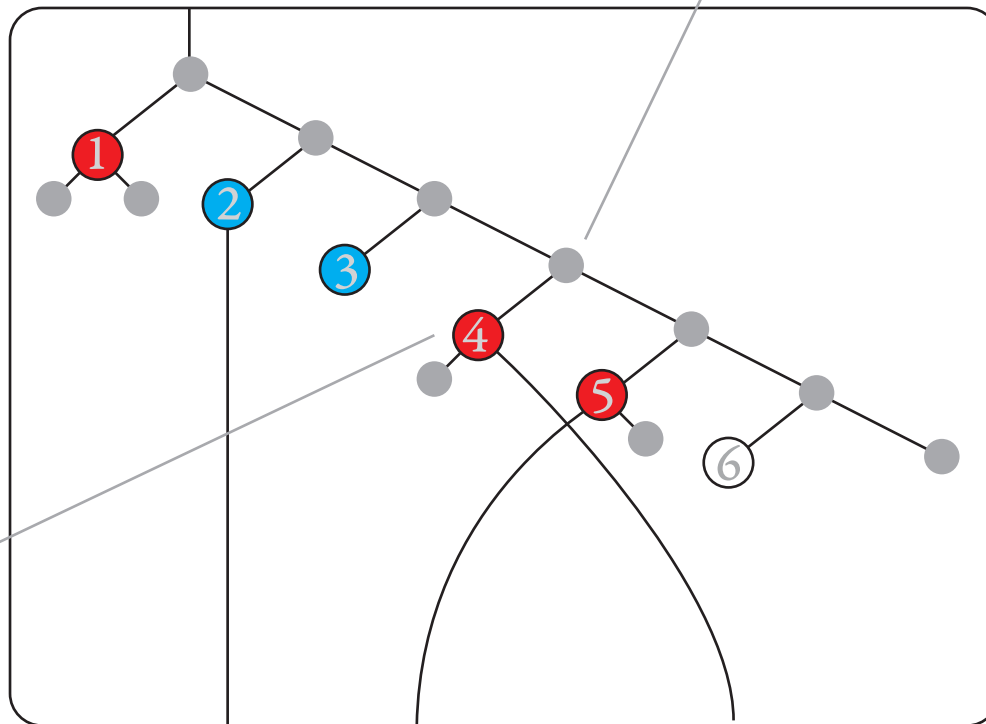
input



number the non-port
nodes in the input term
according to their
appearance in the
pre-order traversal

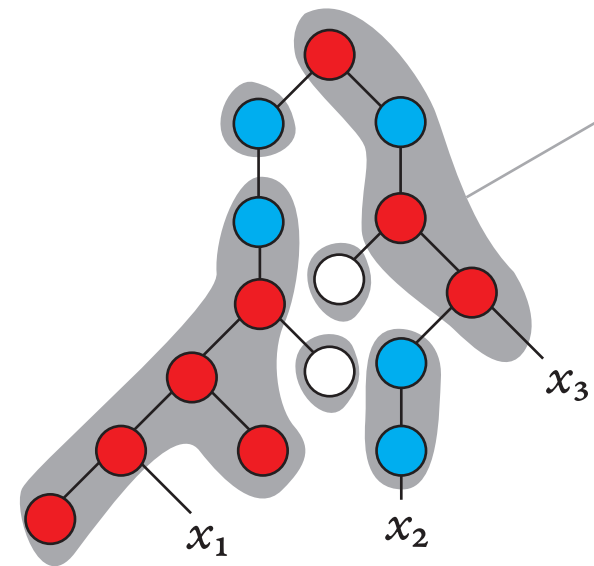
use a copy of the corresponding node, with
edges to the ports inherited, and other edges
plugged by ●

output

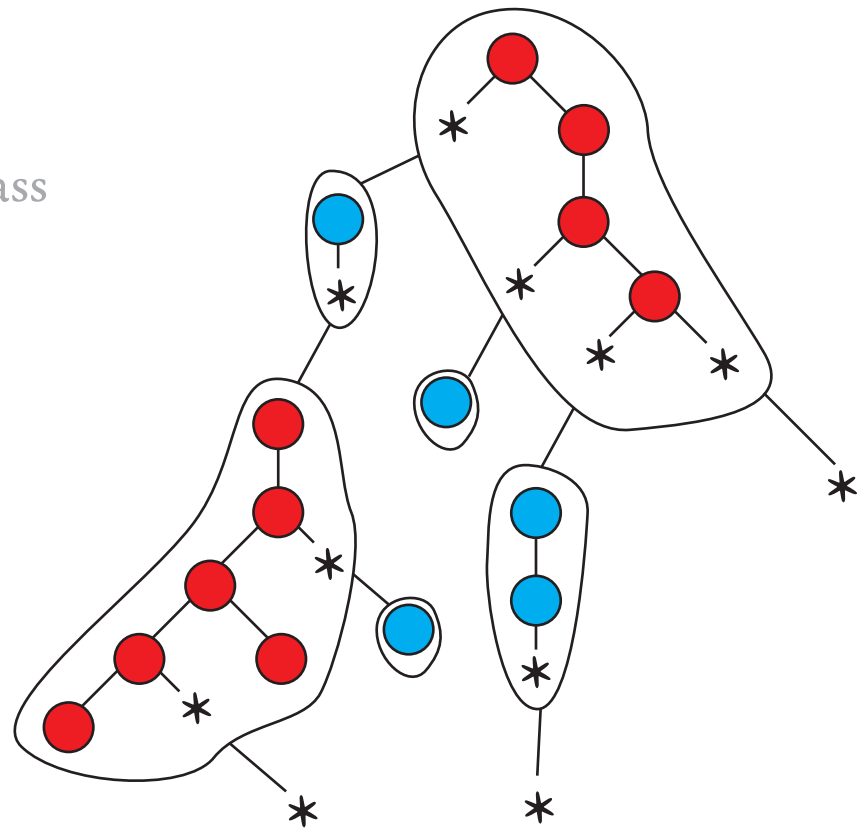


create a binary node for
each non-port node in
the input term

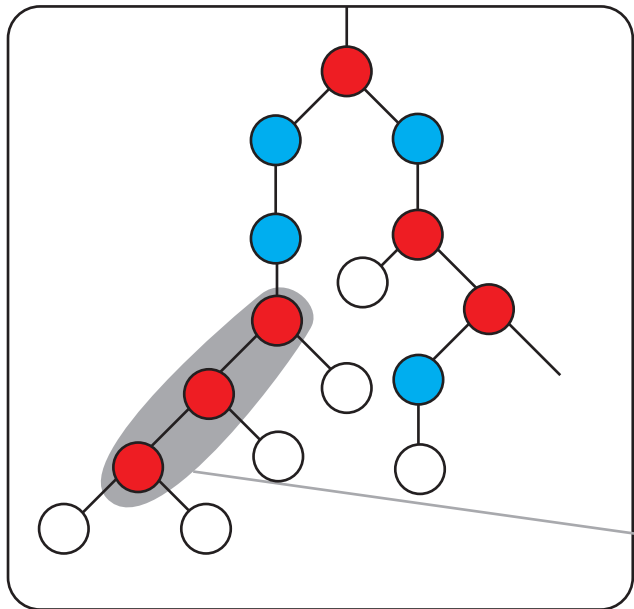
a factorisation equivalence



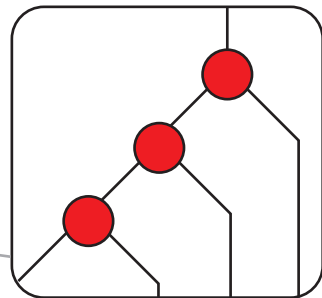
an equivalence class



a tree



a factor of the
tree, viewed
as a term





input alphabet

arity 2



arity 1



arity 0



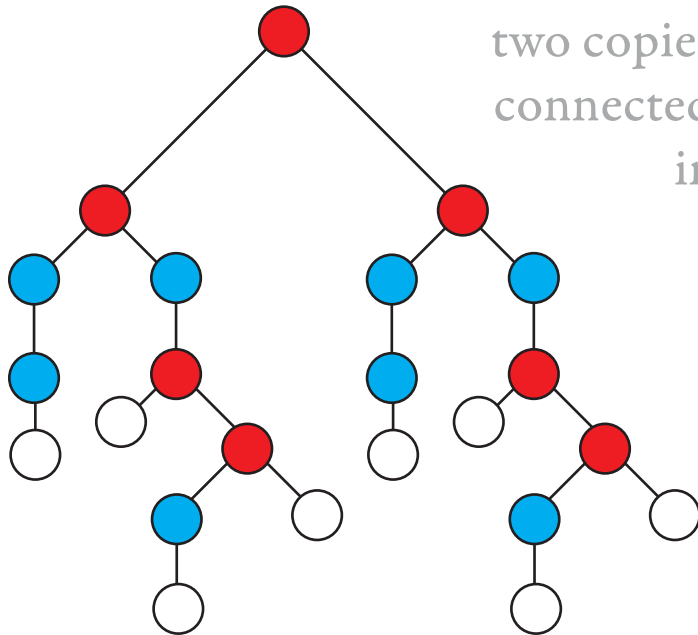
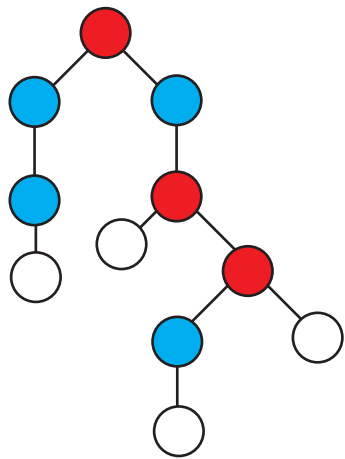
output alphabet

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



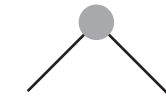
arity 1



arity 0

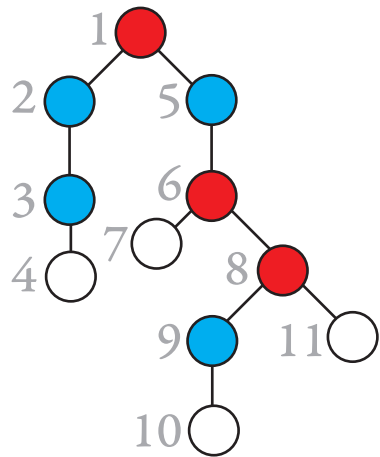


arity 2

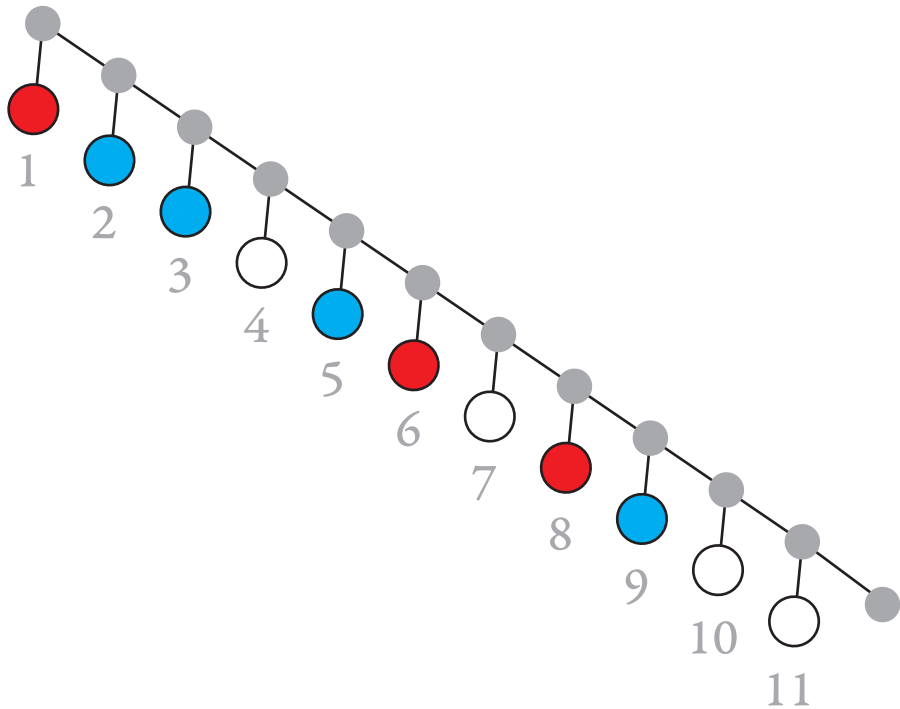


arity 0





\mapsto





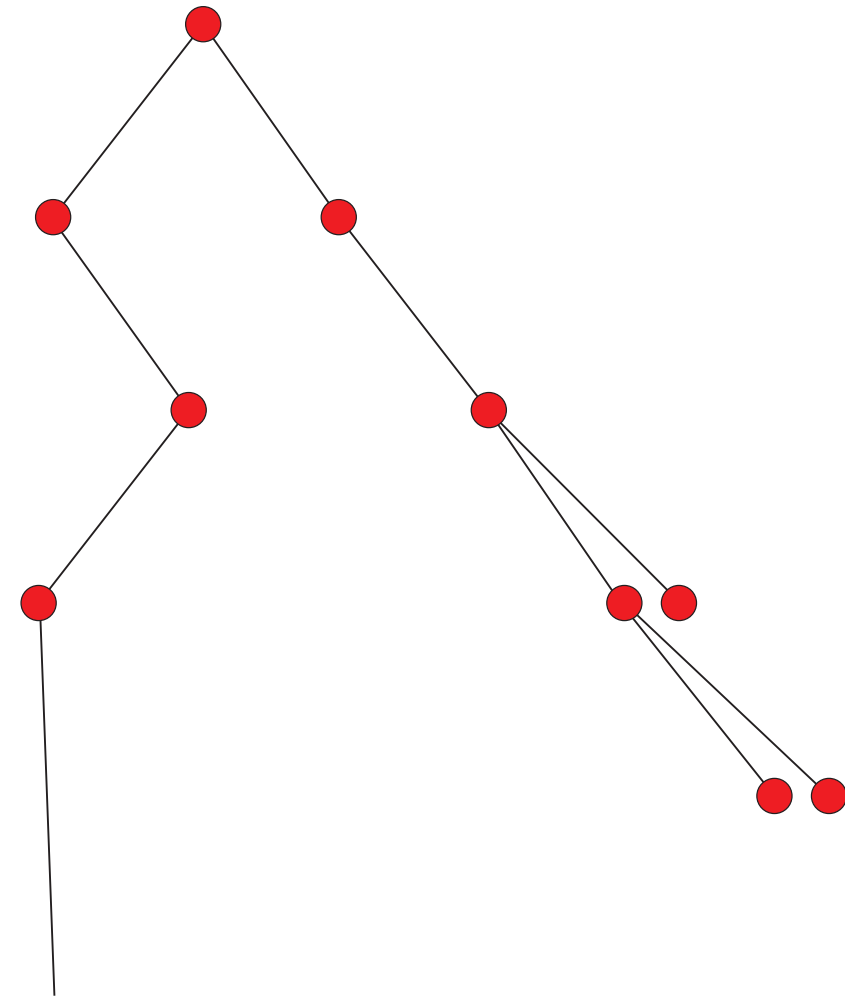
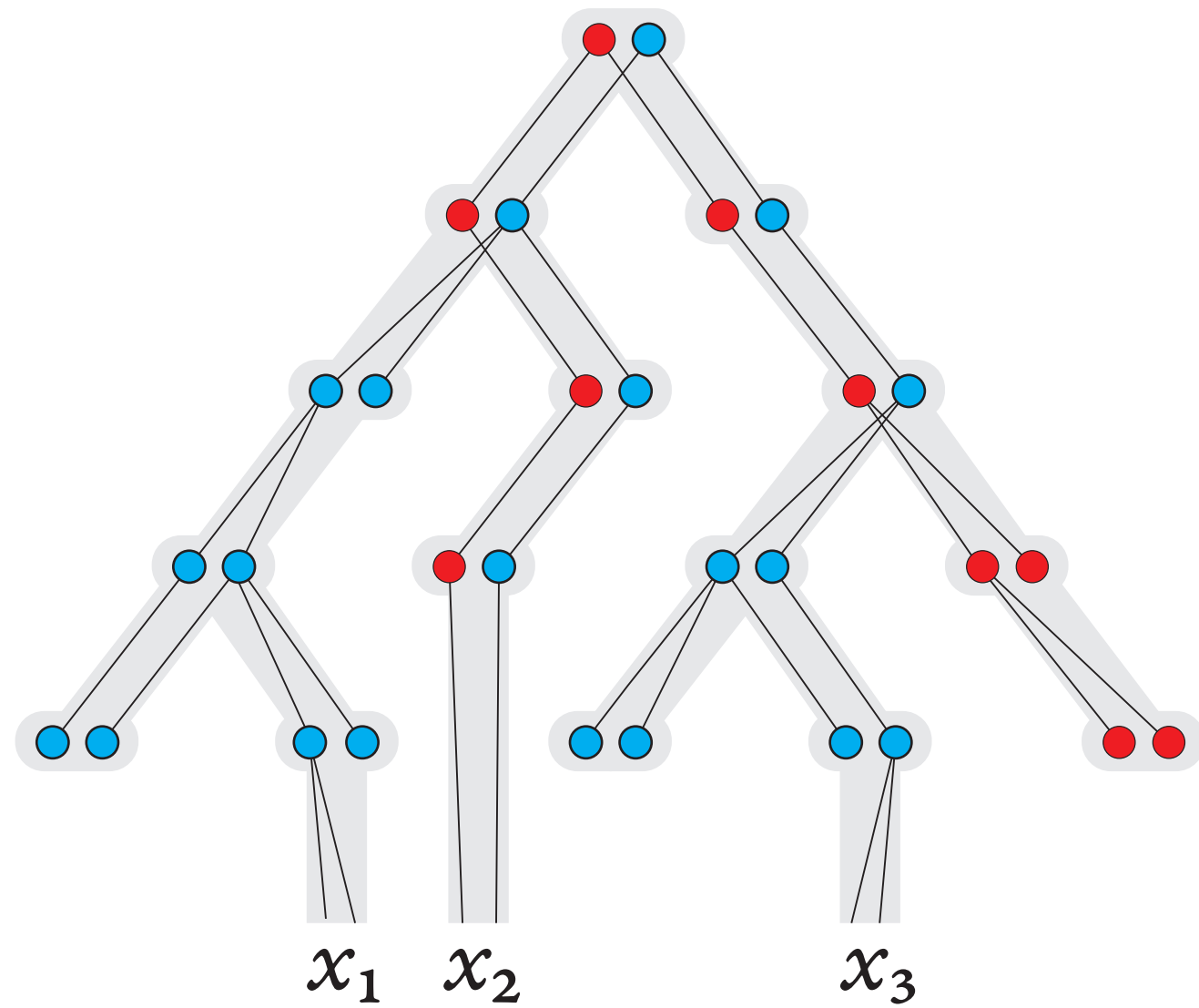


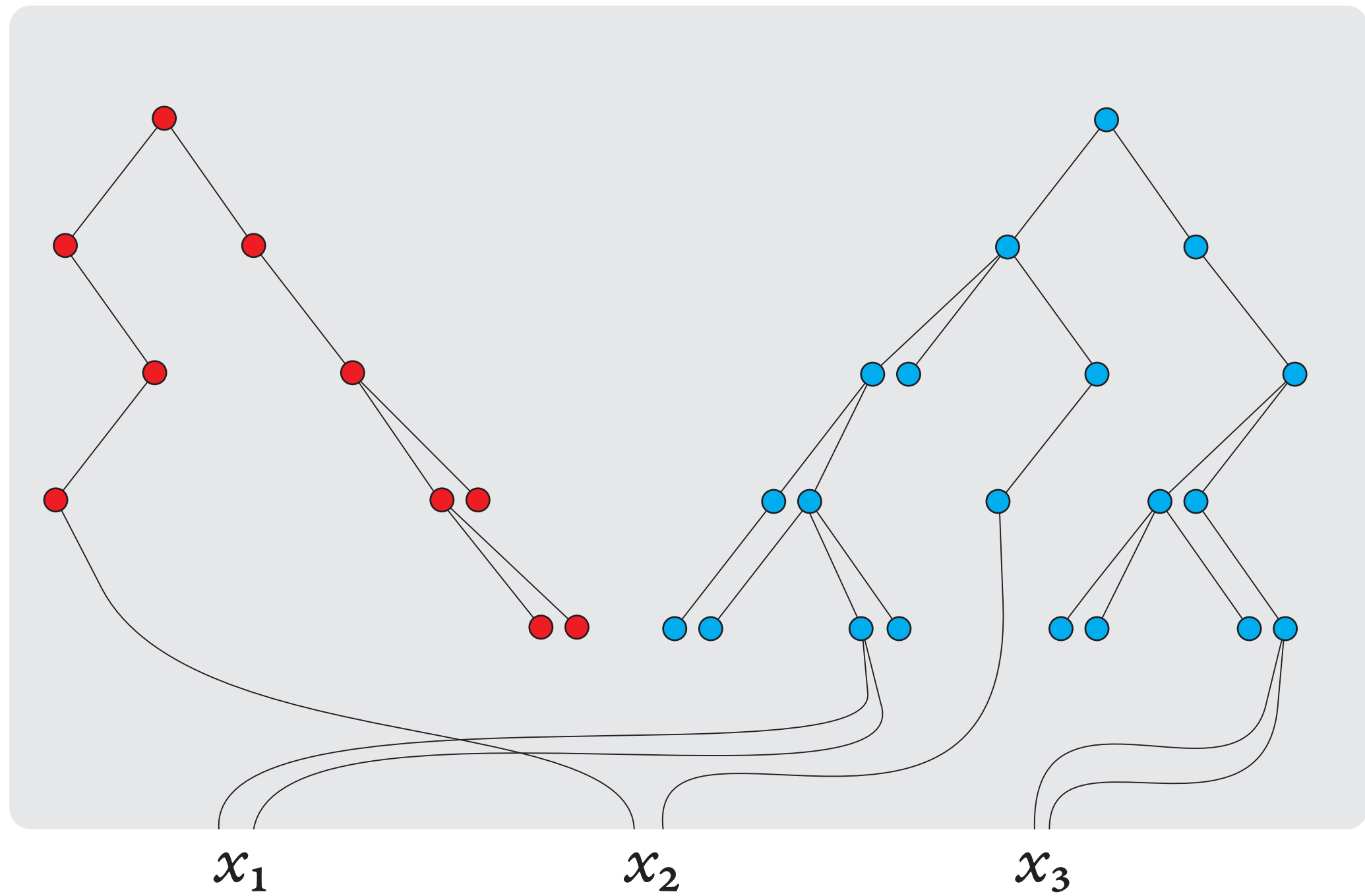
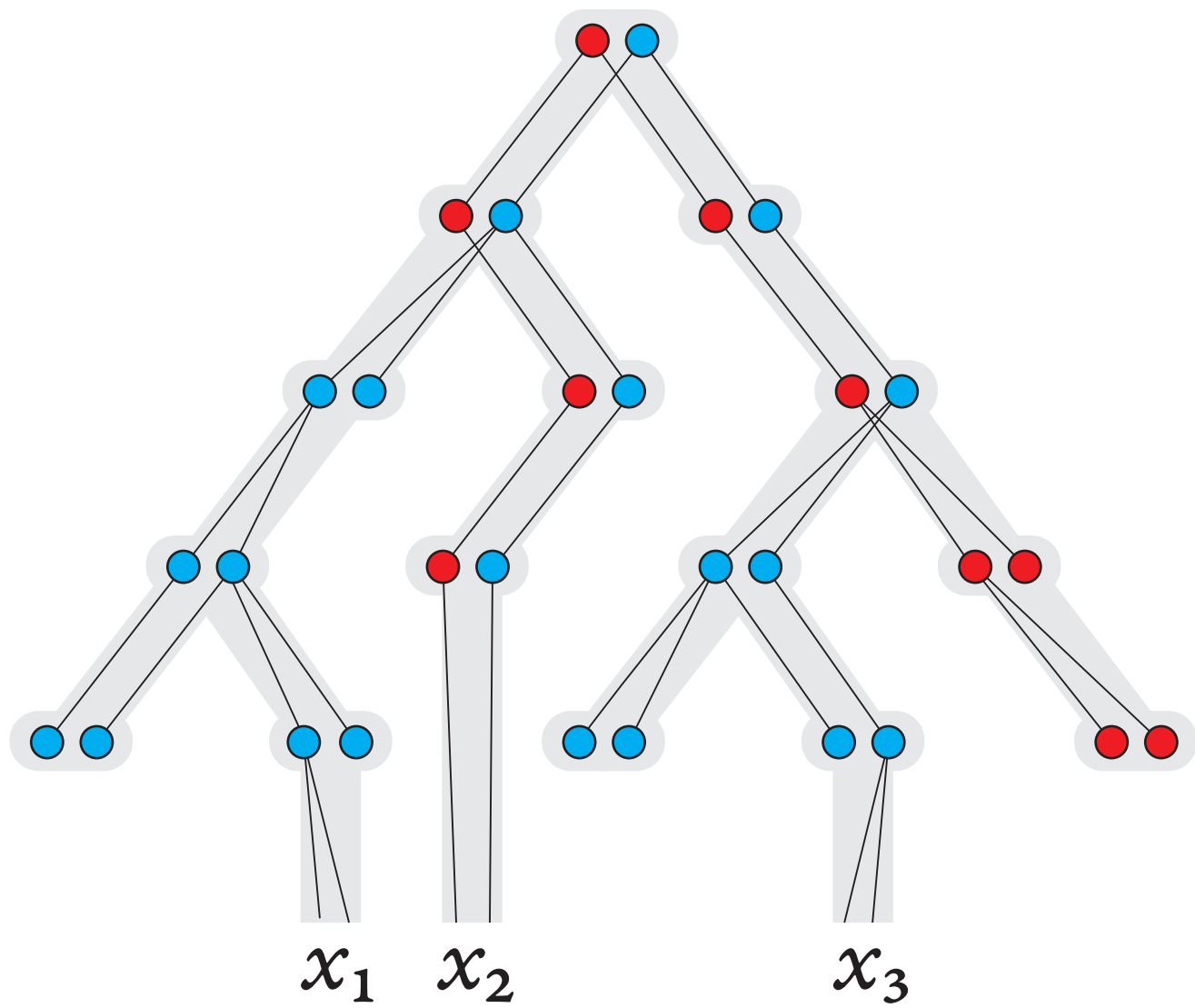
a term of arity 4



a term of arity 0





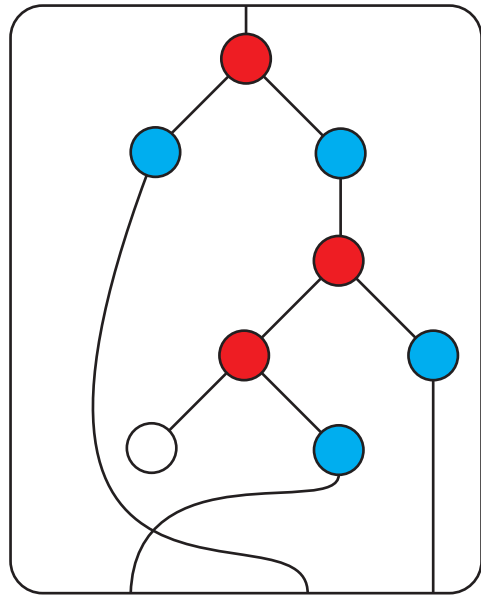




satisfies (*)

(*)

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



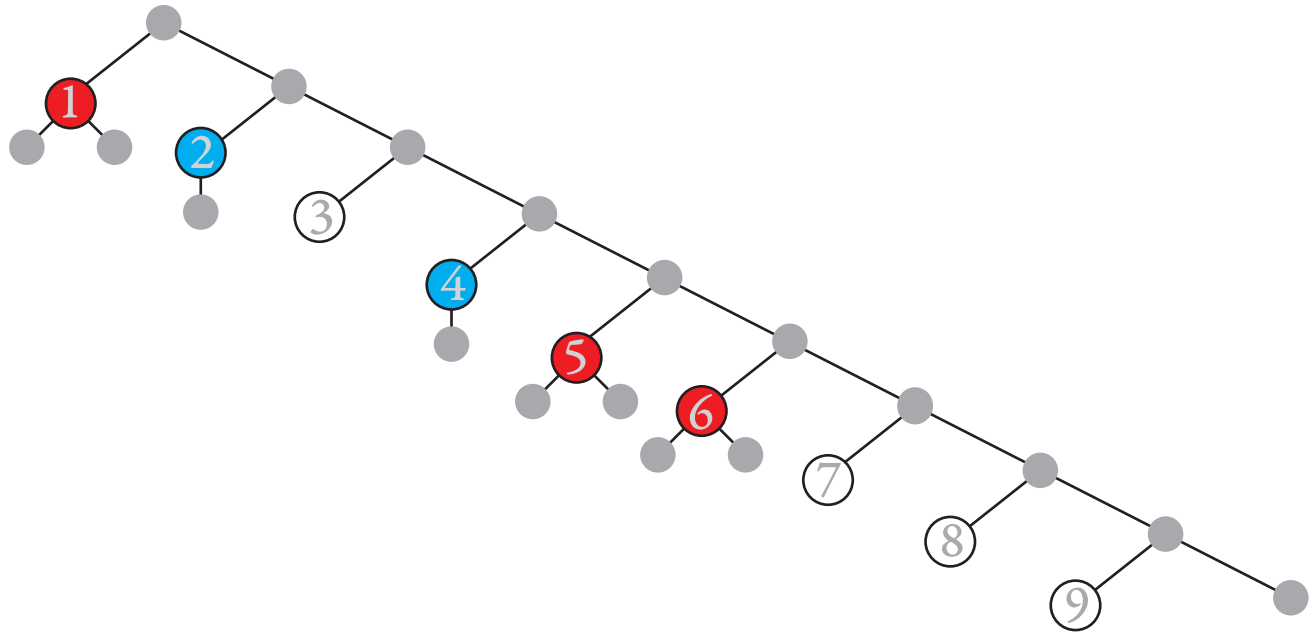
violates (*)

In the dual, this variable is mapped to the i -th edge which enters the j -th port of the reducer.

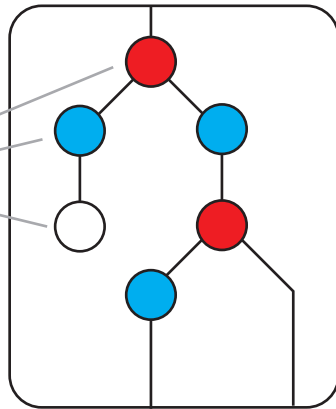
input



output

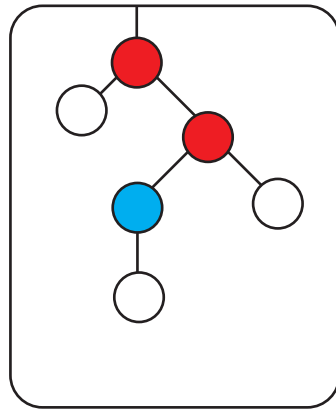


register r

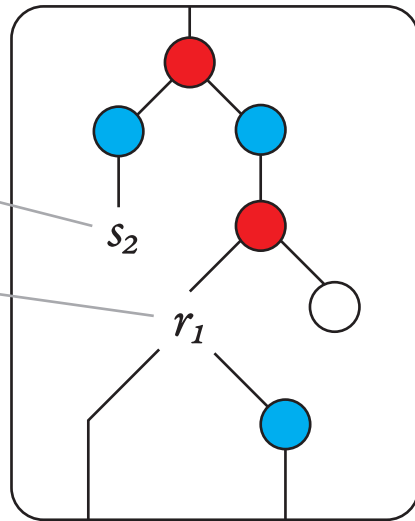


letters of the output alphabet

register s



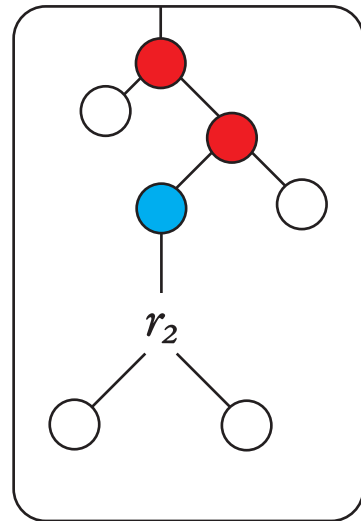
register r



copy 2 of register s

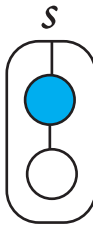
copy 1 of register r

register s









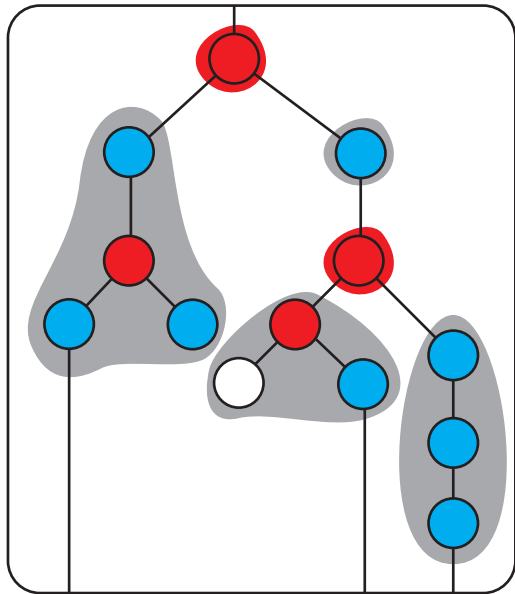




factors without
branching nodes

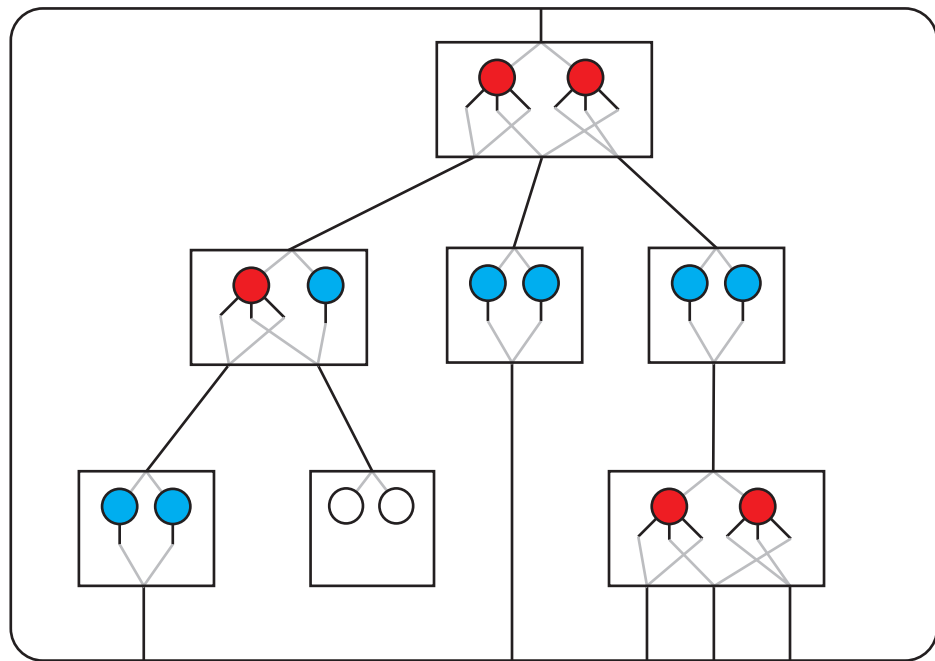


factors with
branching nodes

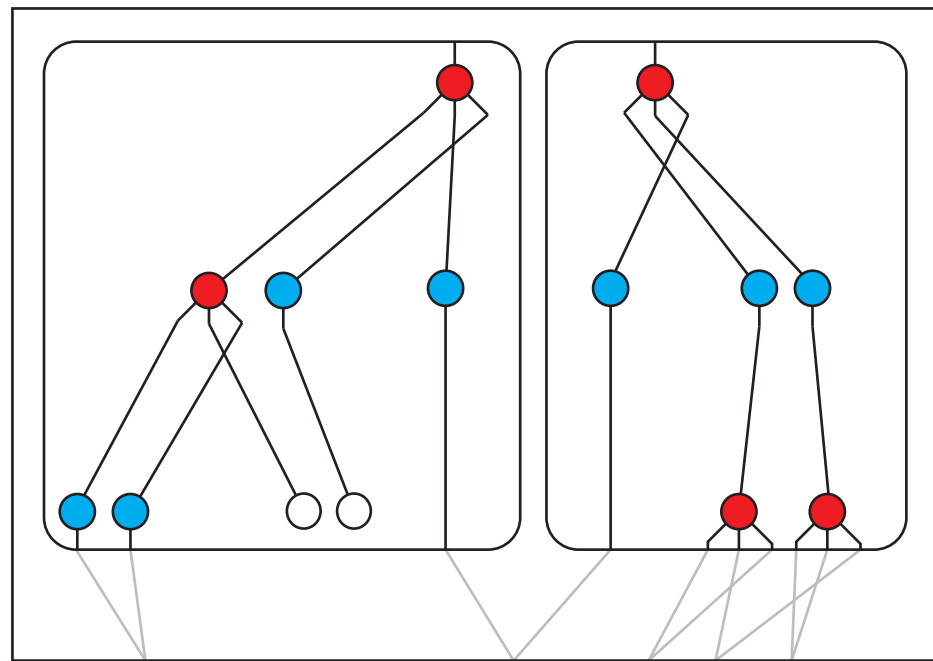




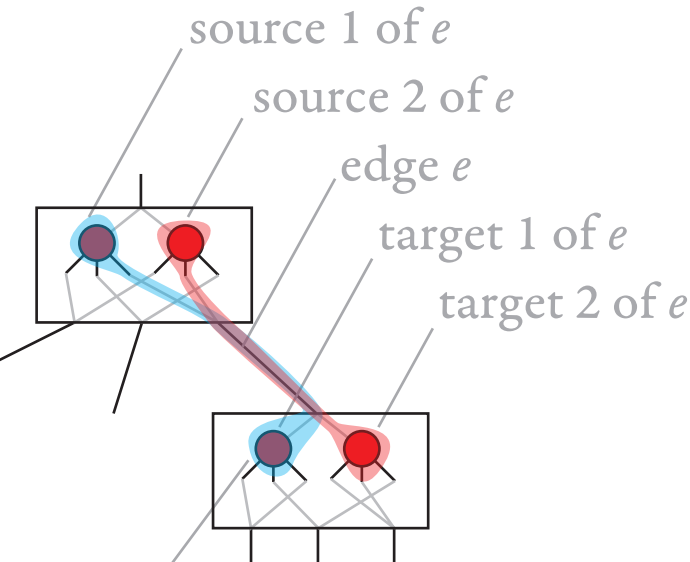
input



output







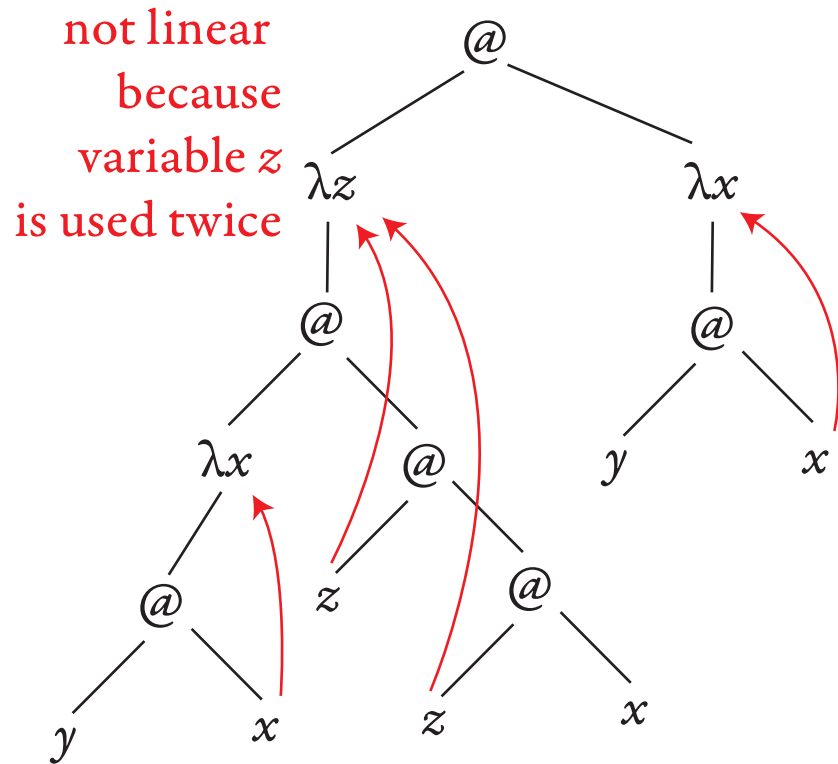
linear



we only count
variables used
in their scope

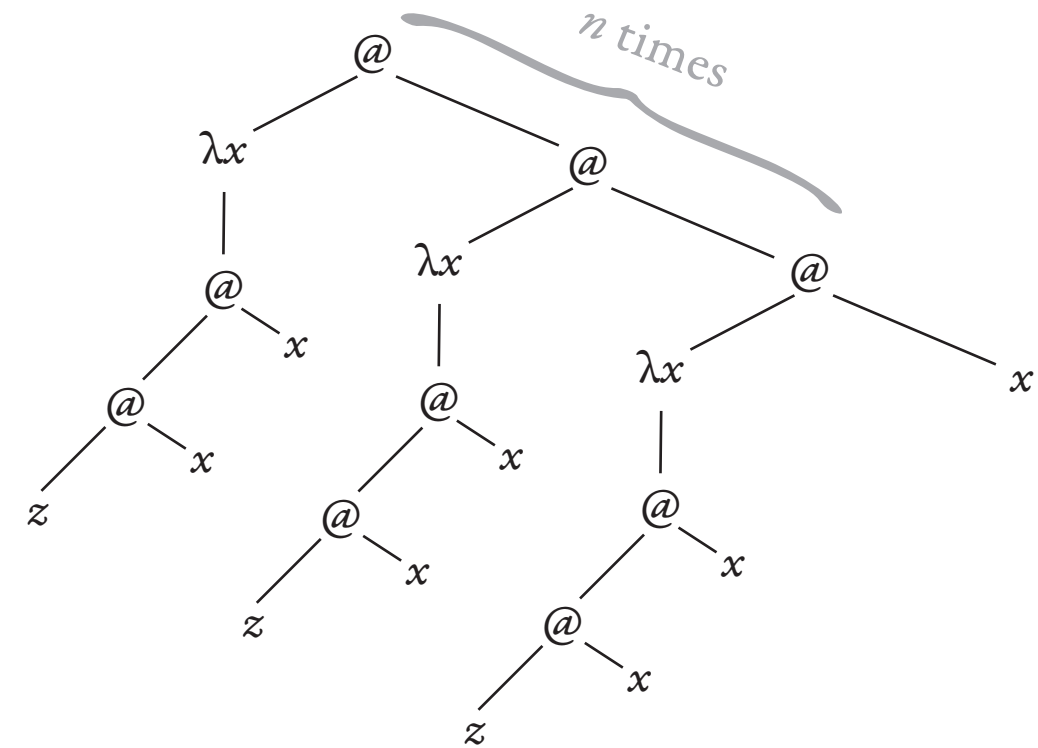
variable z can be used twice because it is free

not linear

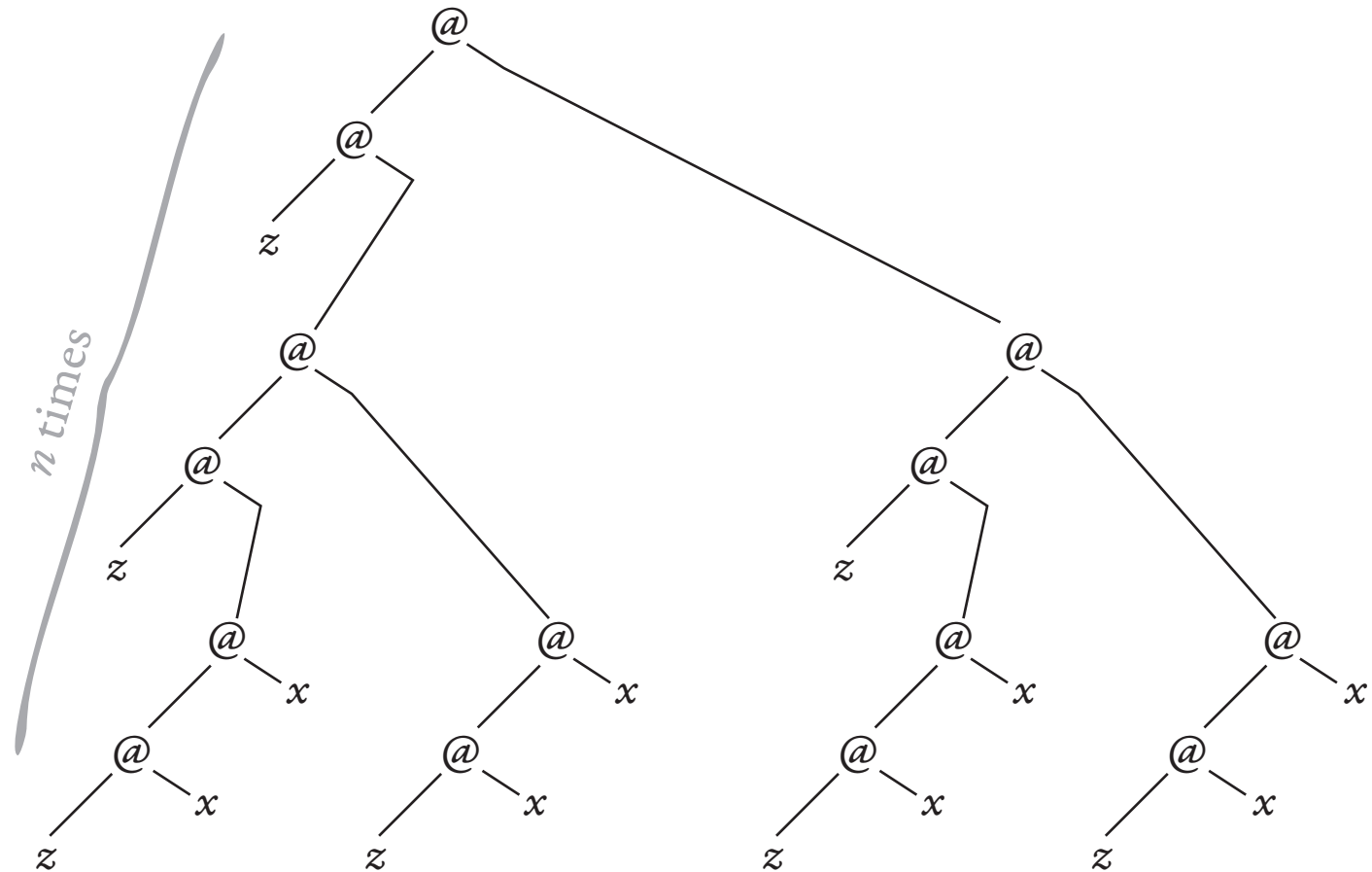


not linear
because
variable z
is used twice

a λ -term of size $O(n)$



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

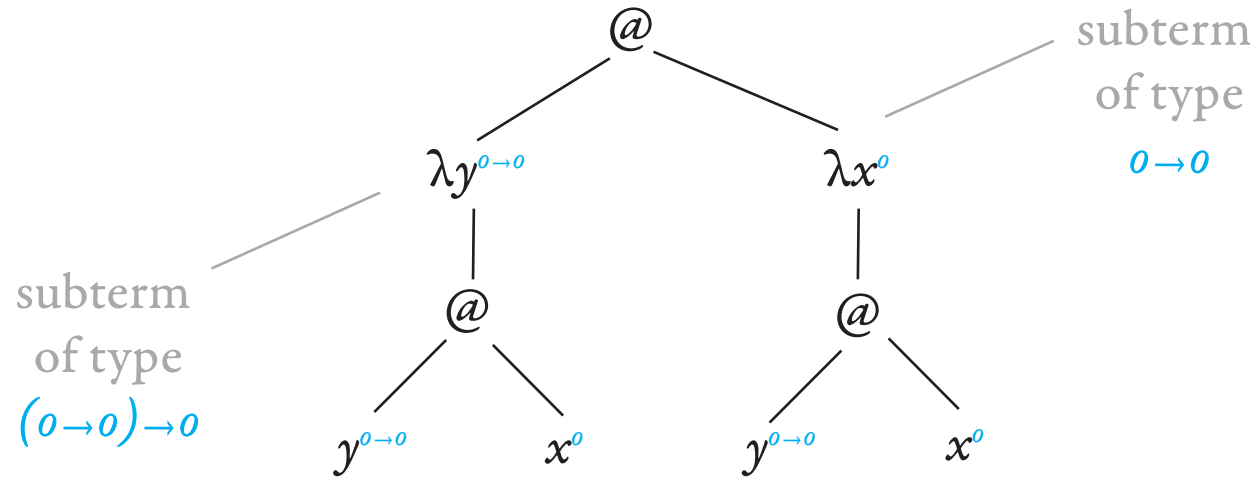


variables

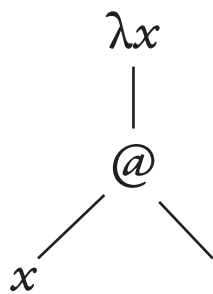
types of variables in superscript

x^o $y^{o \rightarrow o}$

λ -term of type o



@



$\lambda x.$



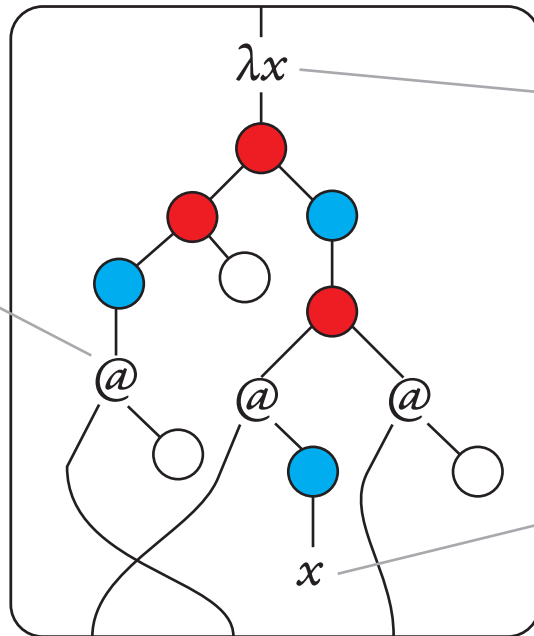
r



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



register i is replaced by $@$
with the left child leading
to the i -th port and the
right child leading to the
previous subtree of register i



variable x is bound in the root

the original port is replaced by x