

a ranked alphabet

arity 2



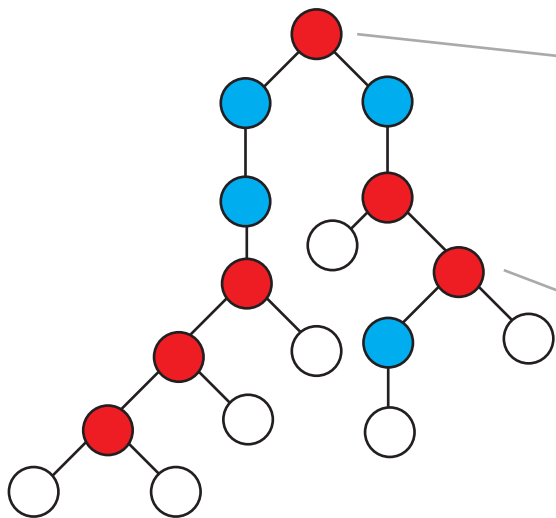
arity 1



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 $\Sigma^{[2]}$



$\text{unfold}_1(t)$



$\text{unfold}_2(t)$





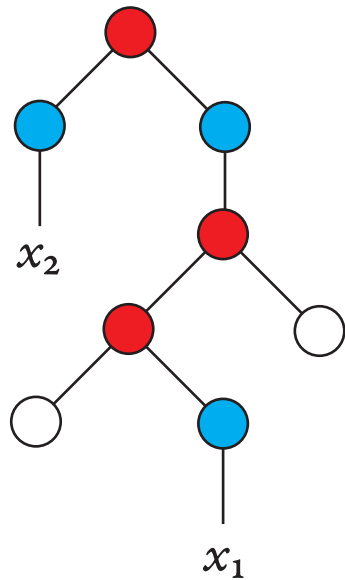
t



substitute(t)

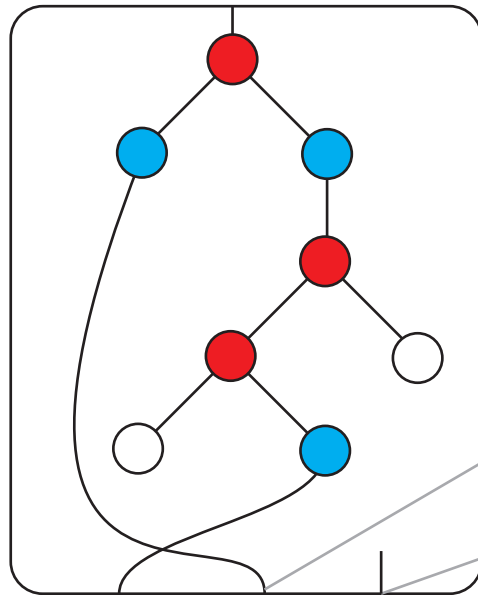






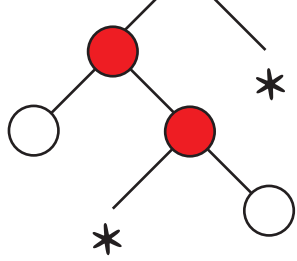
=

a term of arity 3

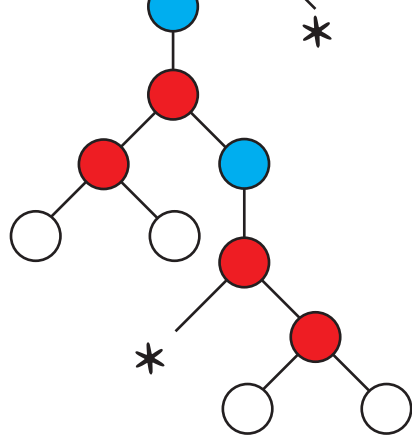


lines leaving at the bottom of the box
represent variables

dangling edges represent unused variables

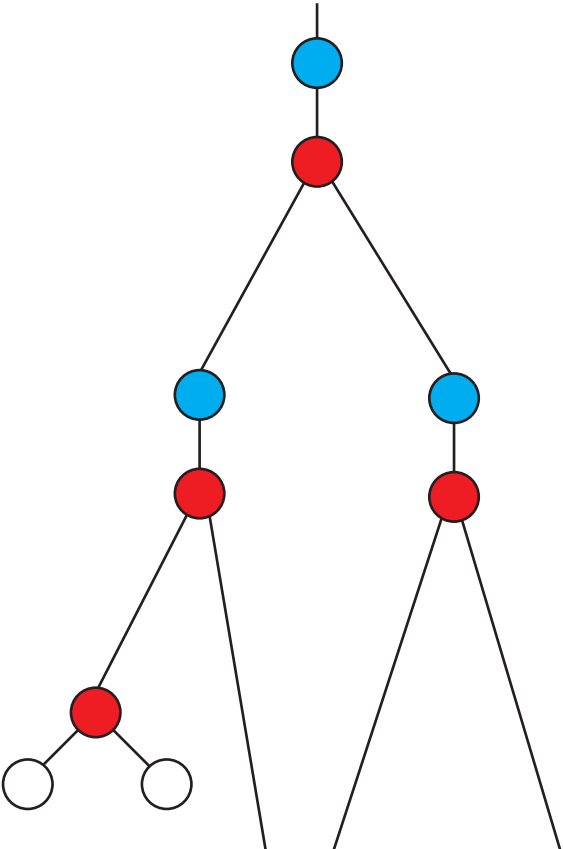


$\mathsf{T}f$
 \mapsto





\mapsto





a term



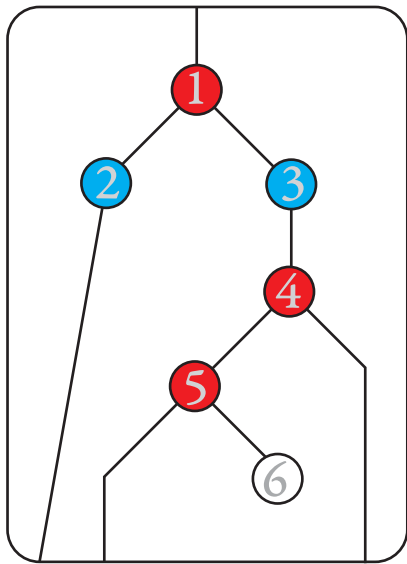
ancestor equivalence



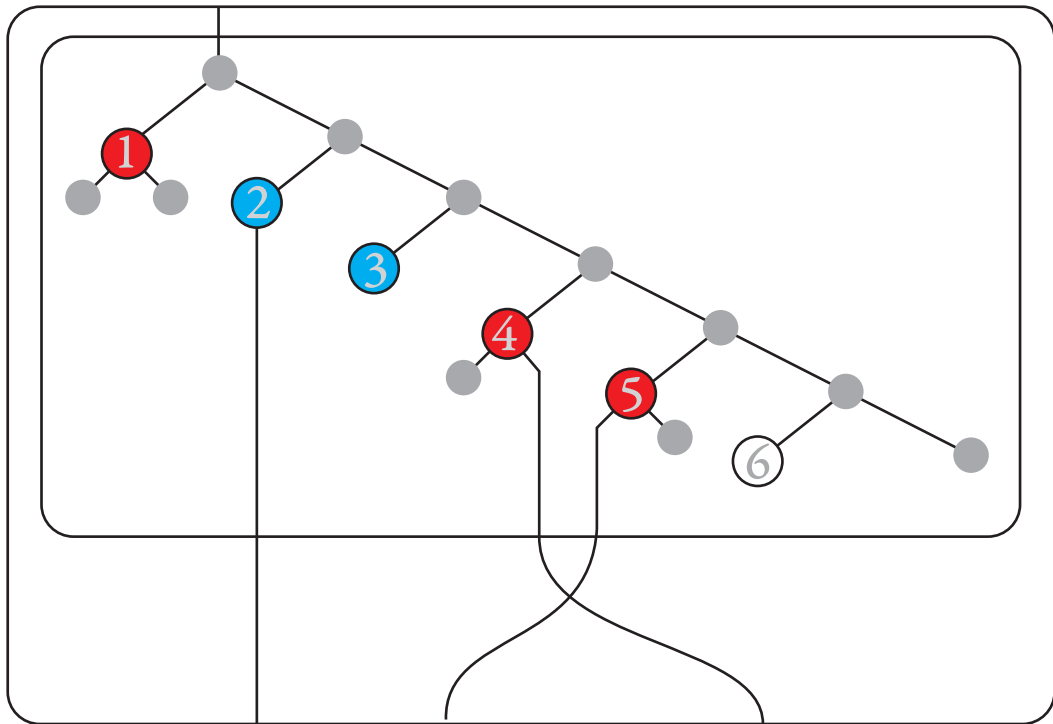
descendant equivalence



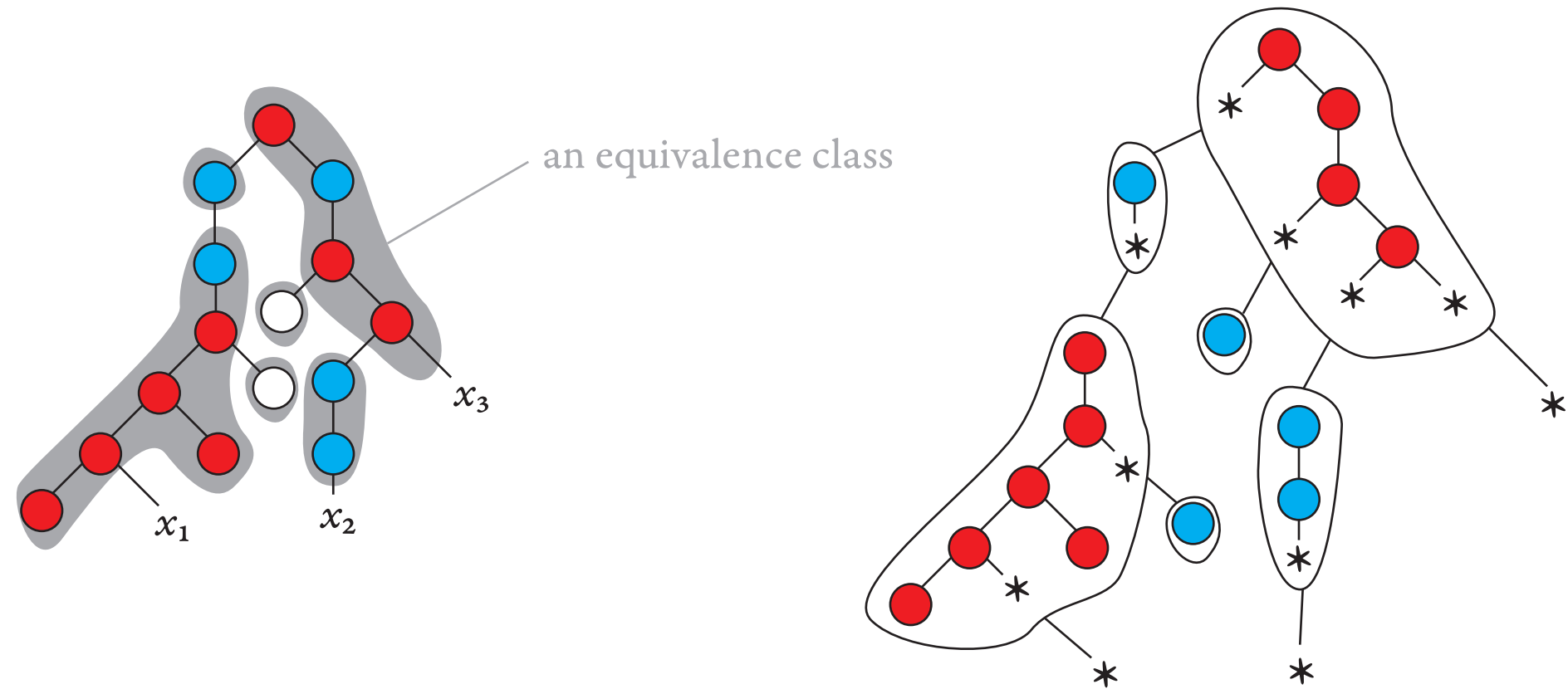




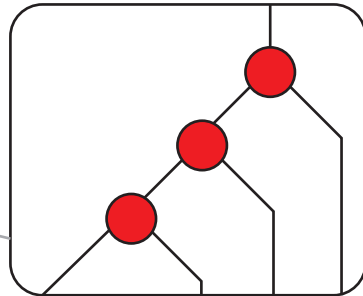
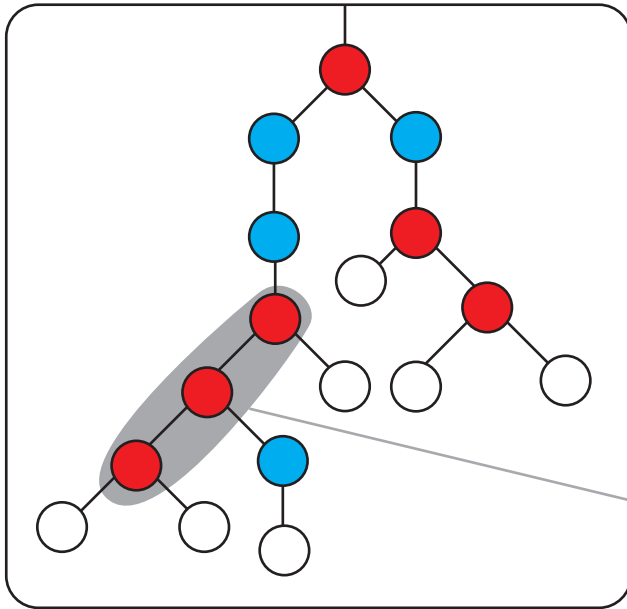
\mapsto



a factorisation equivalence



a tree



a term that
represents a
part of the tree



input alphabet

arity 2



arity 1



arity 0



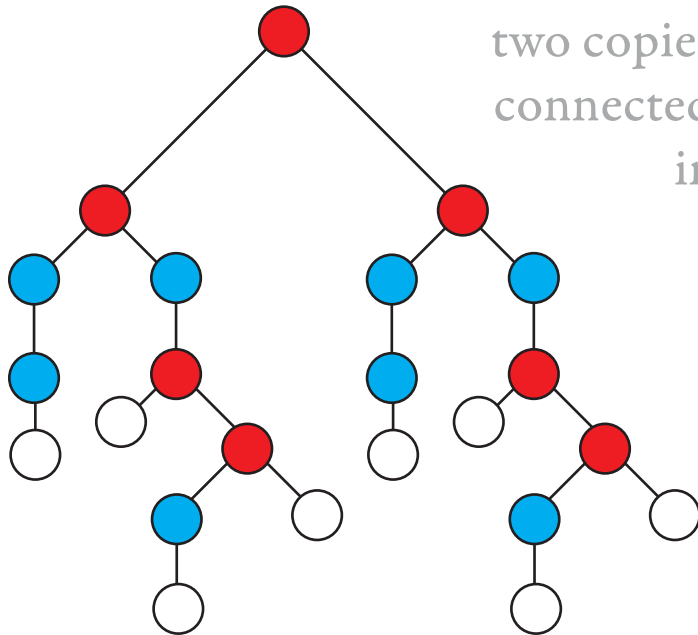
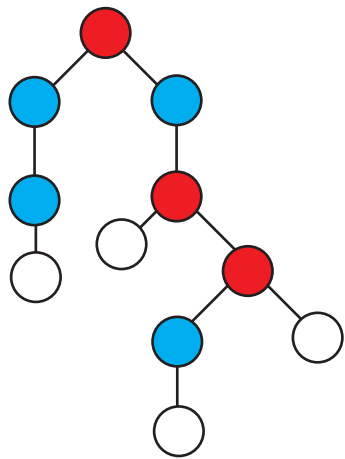
output alphabet

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



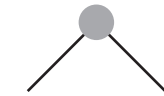
arity 1



arity 0

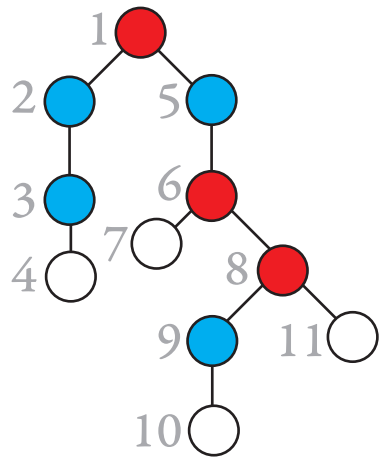


arity 2

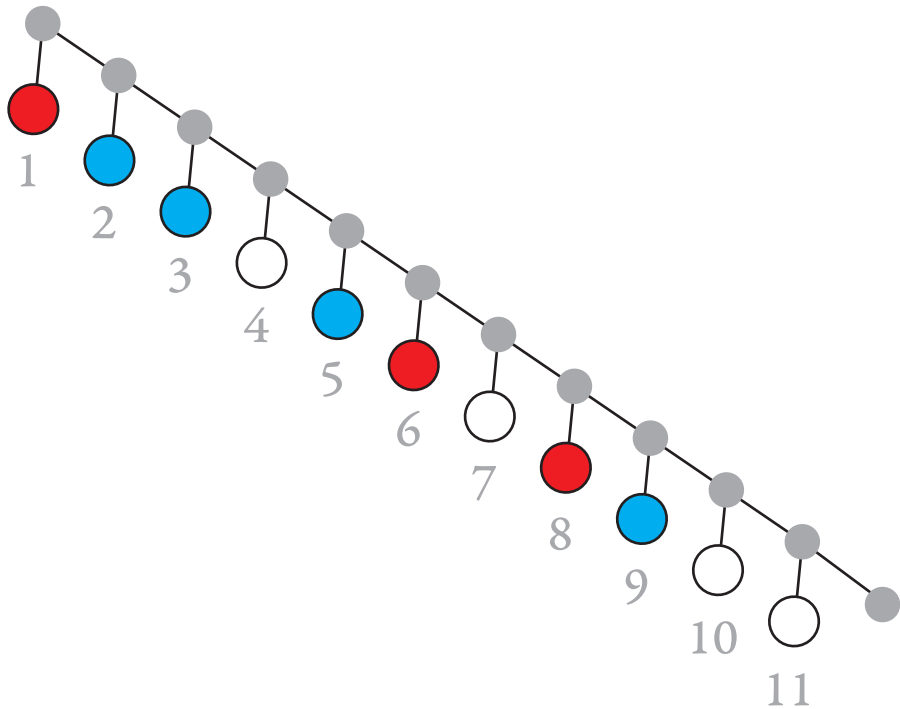


arity 0





\mapsto





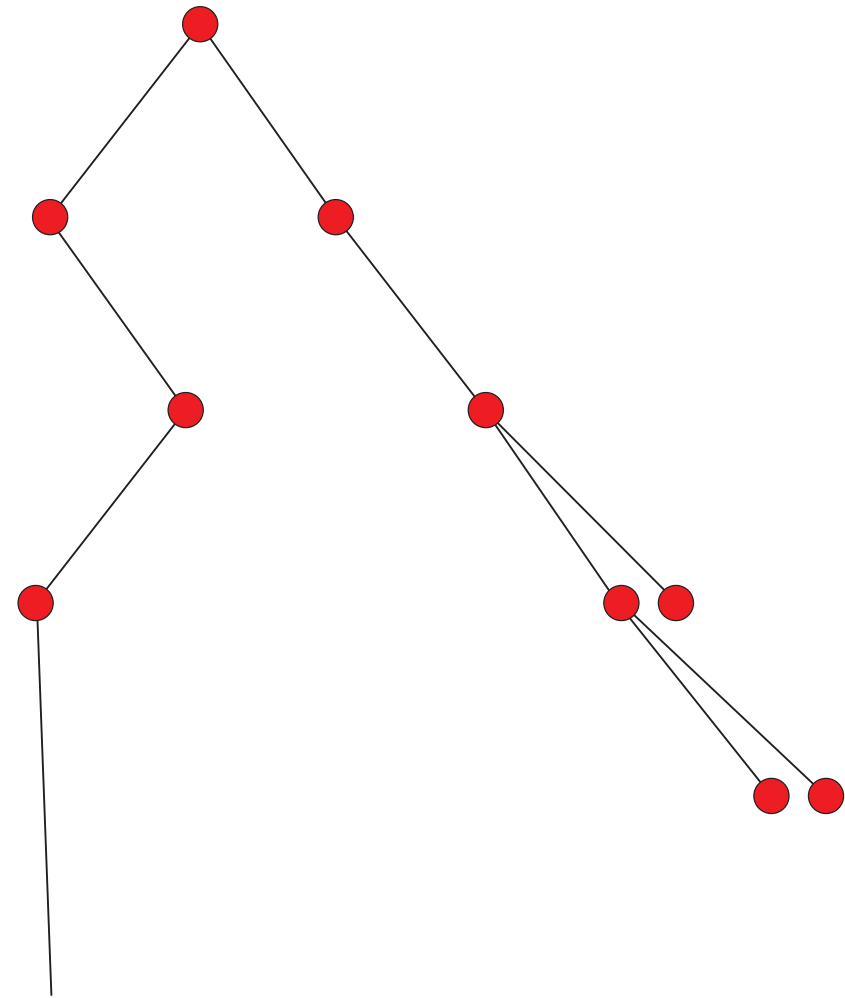
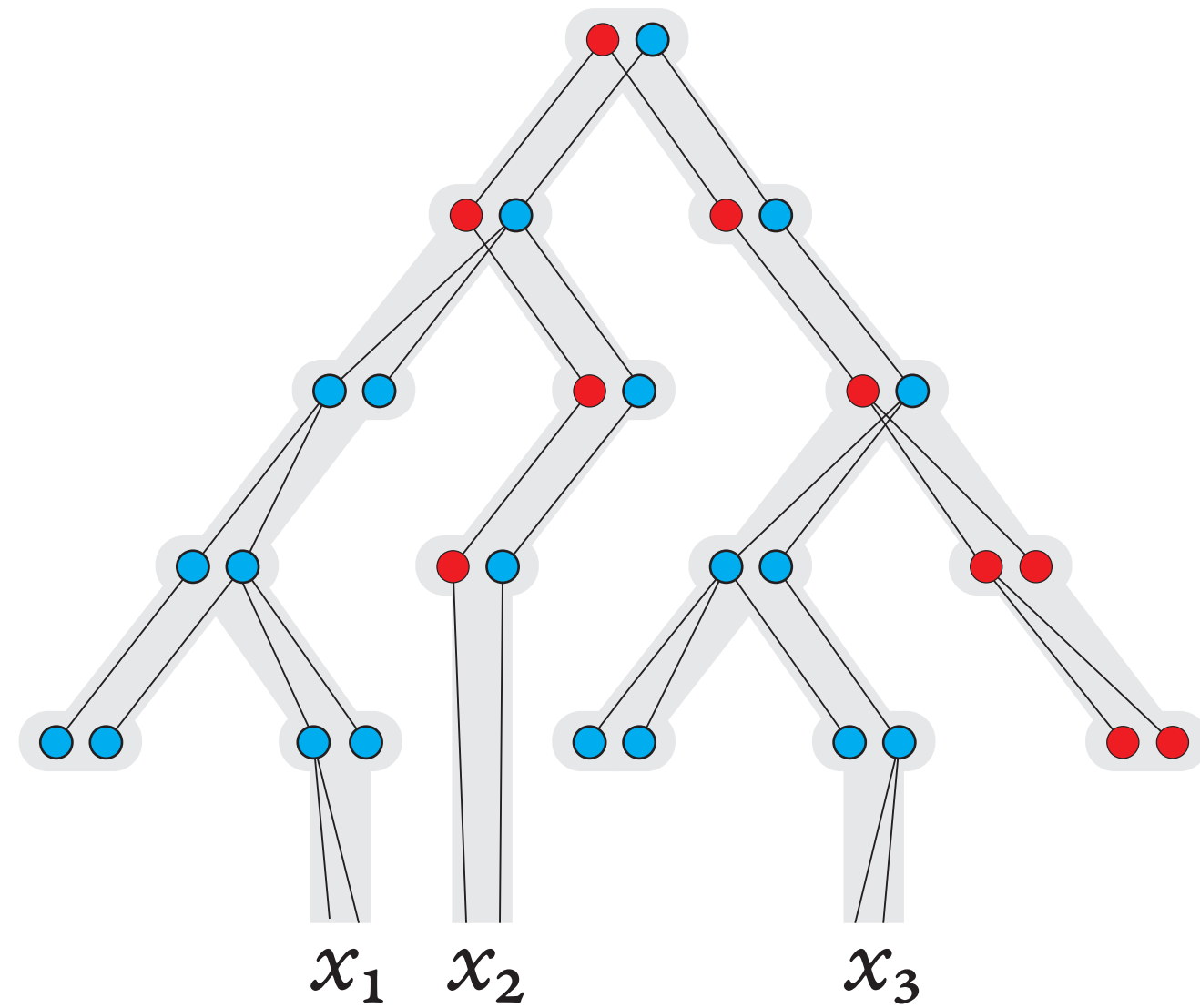


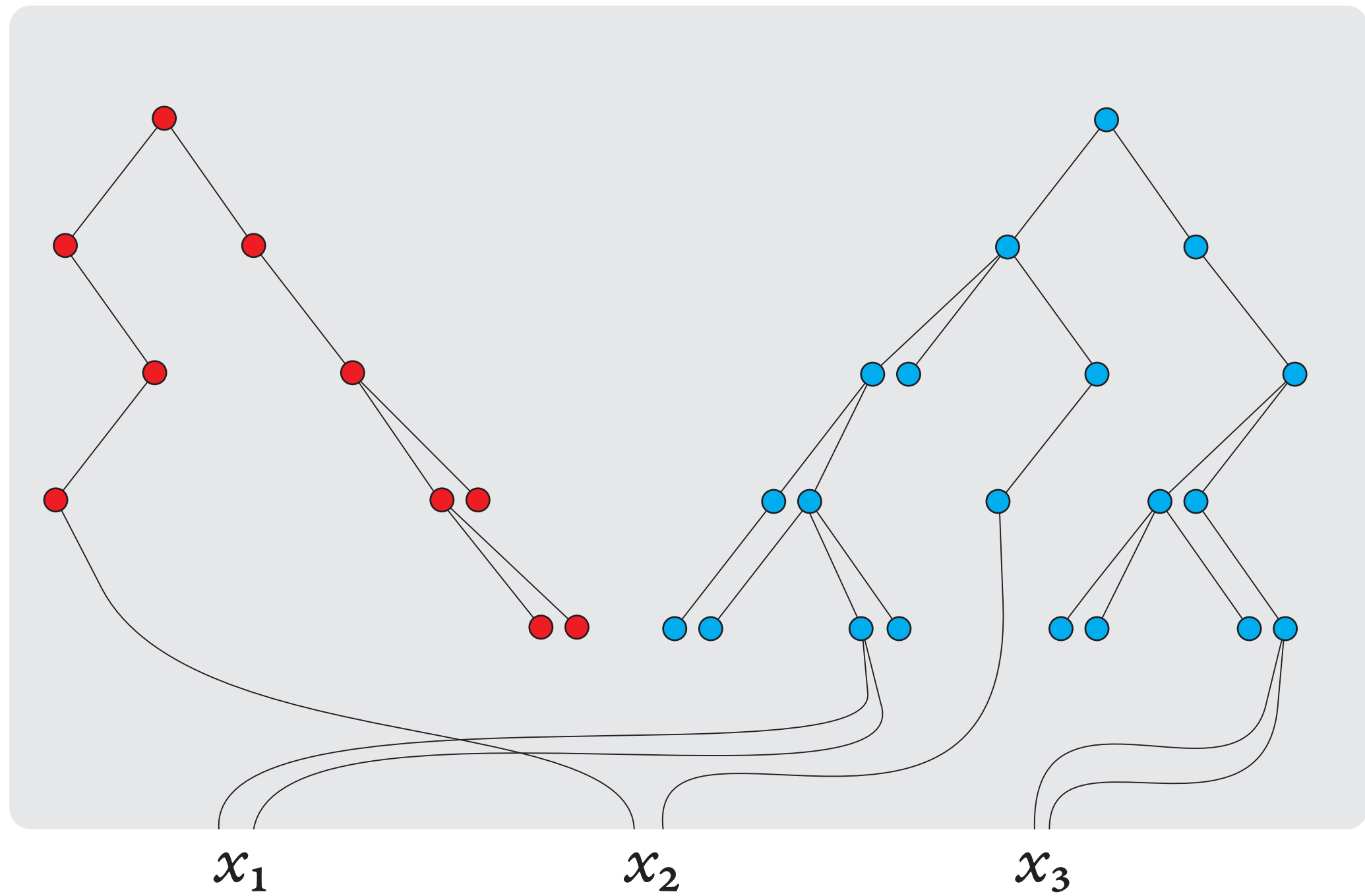
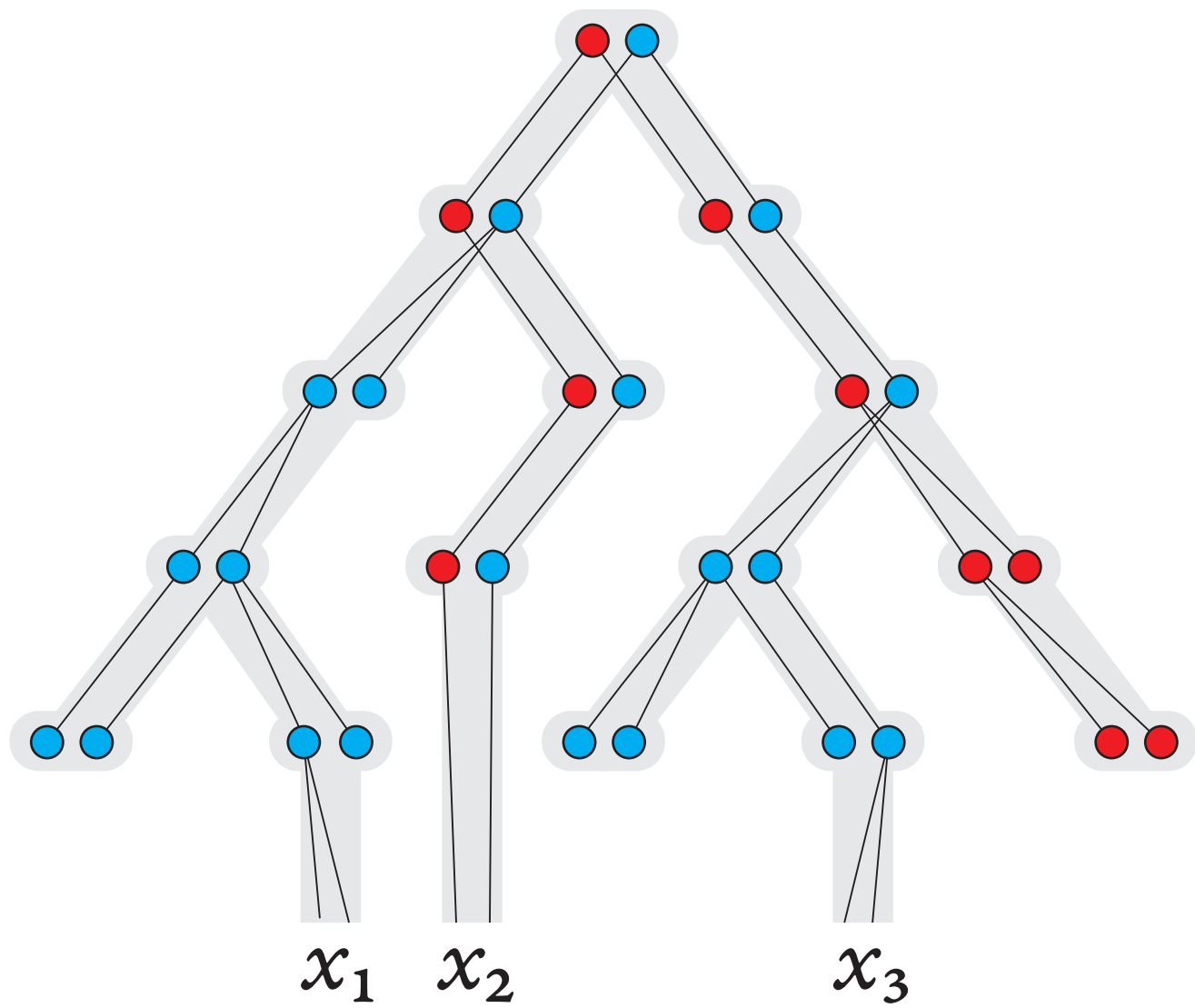
a term of arity 4



a term of arity 0





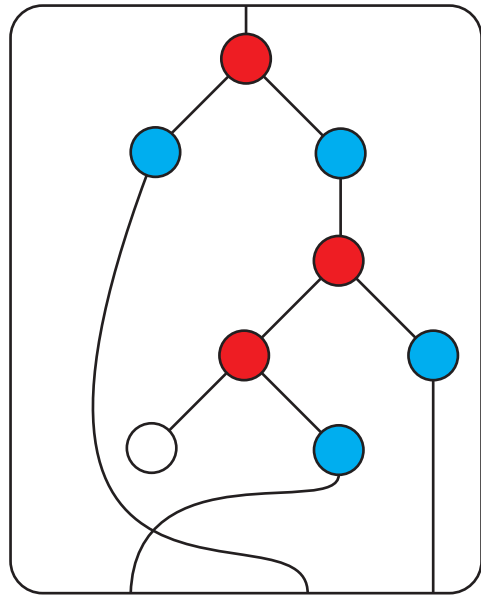




satisfies (*)

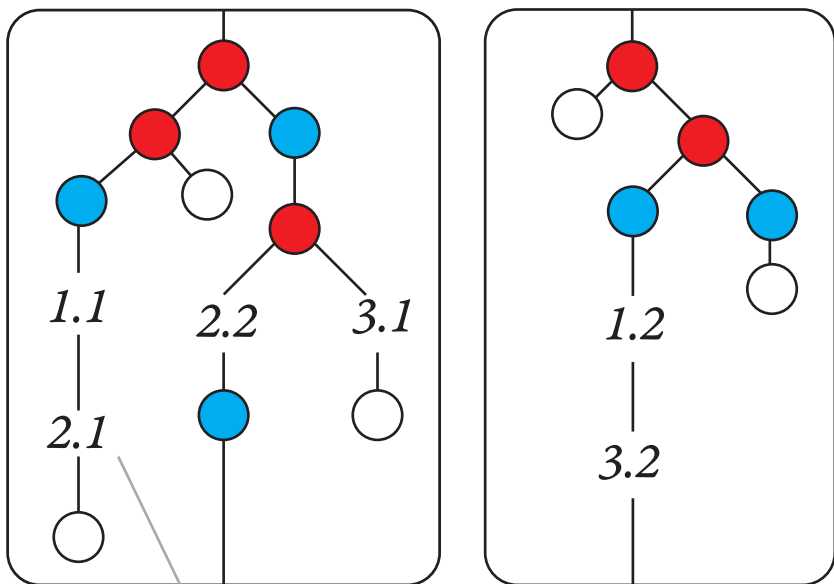
(*)

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



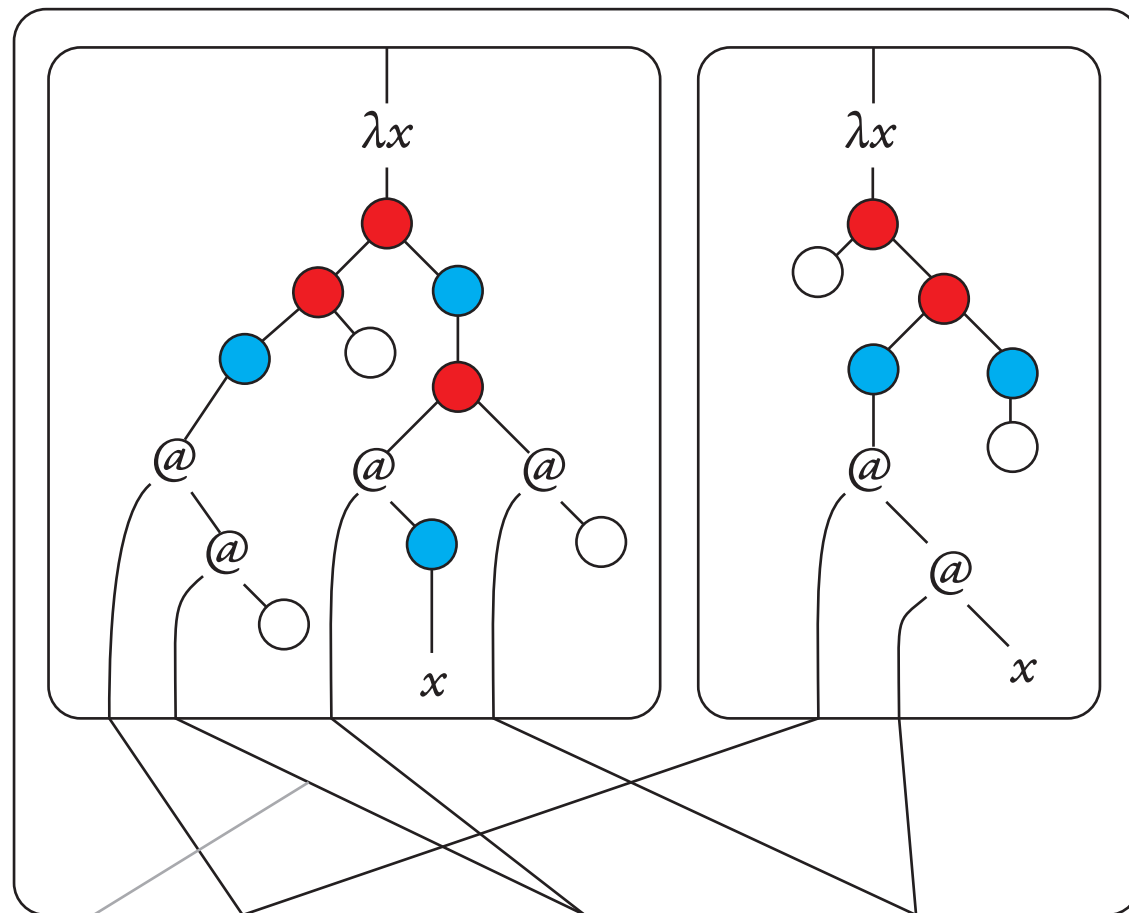
violates (*)

a register update



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

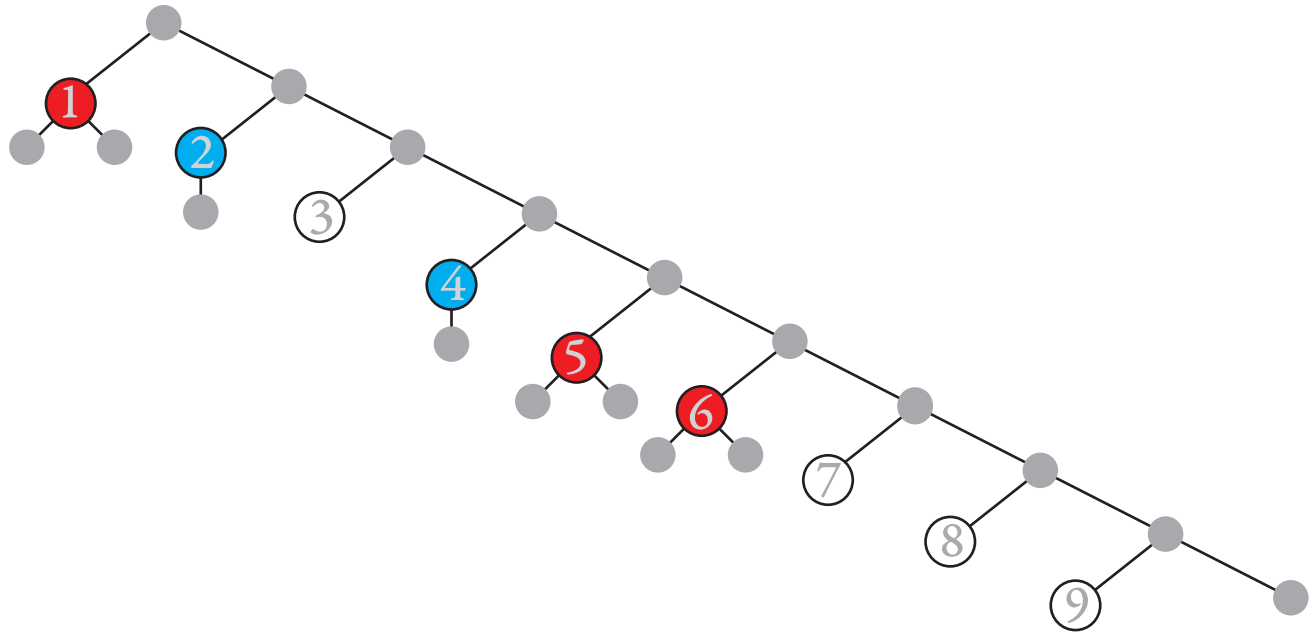
its dual



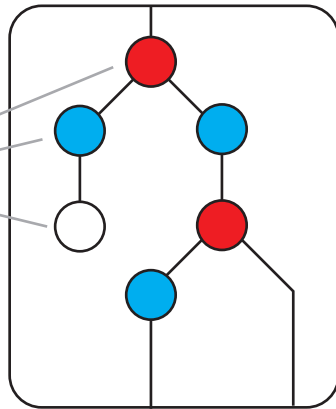
input



output

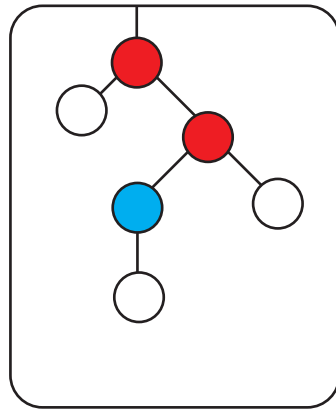


register r

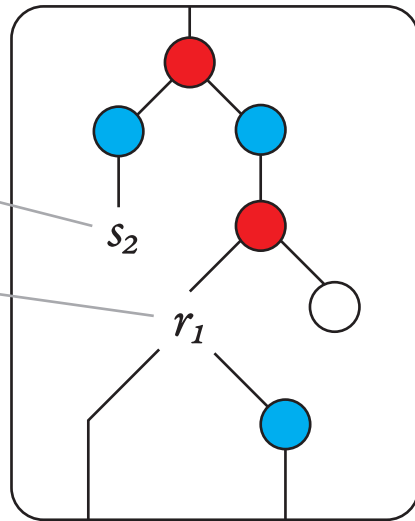


letters of the output alphabet

register s



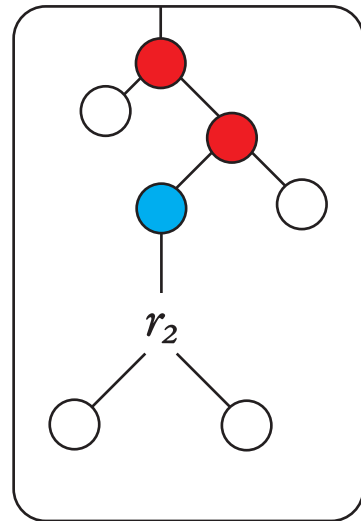
register r



copy 2 of register s

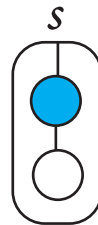
copy 1 of register r

register s









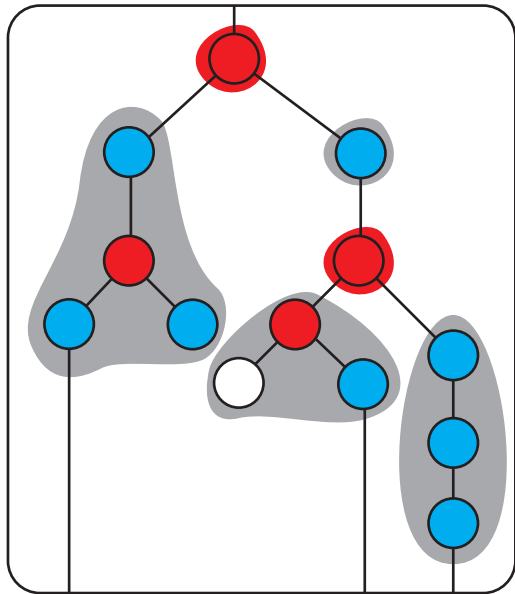




factors without
branching nodes

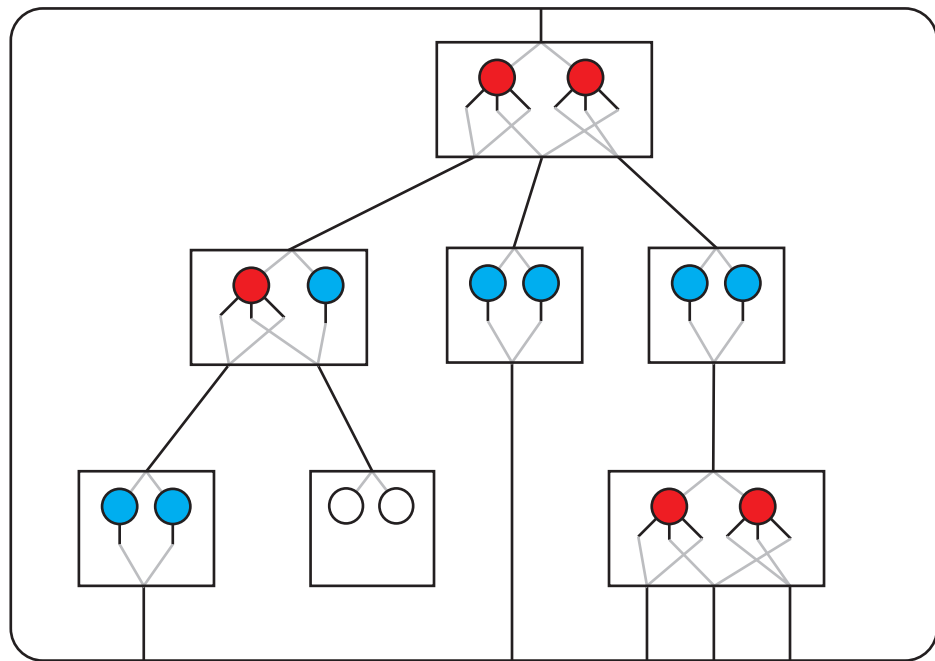


factors with
branching nodes

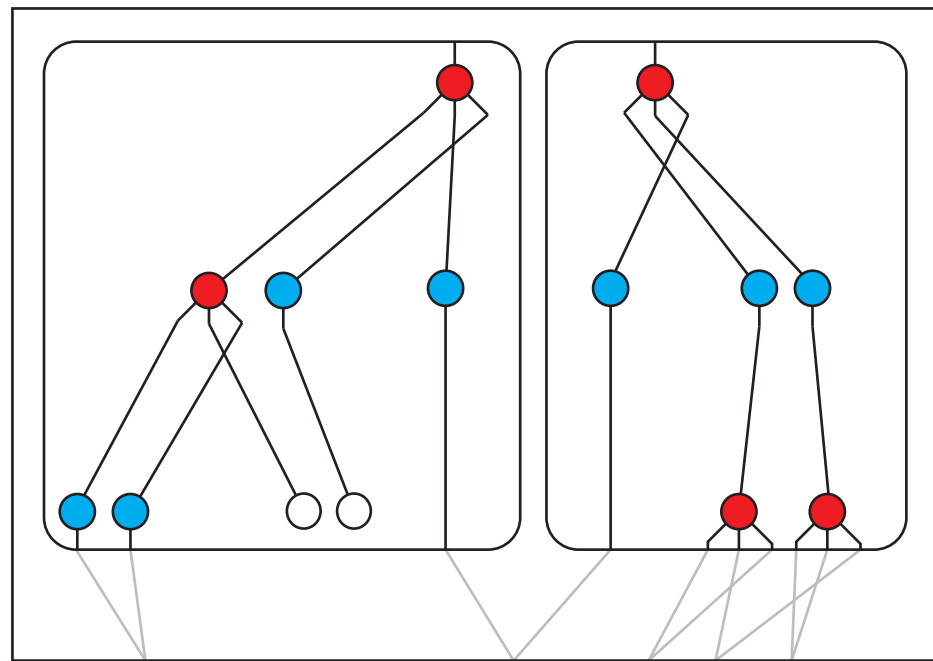




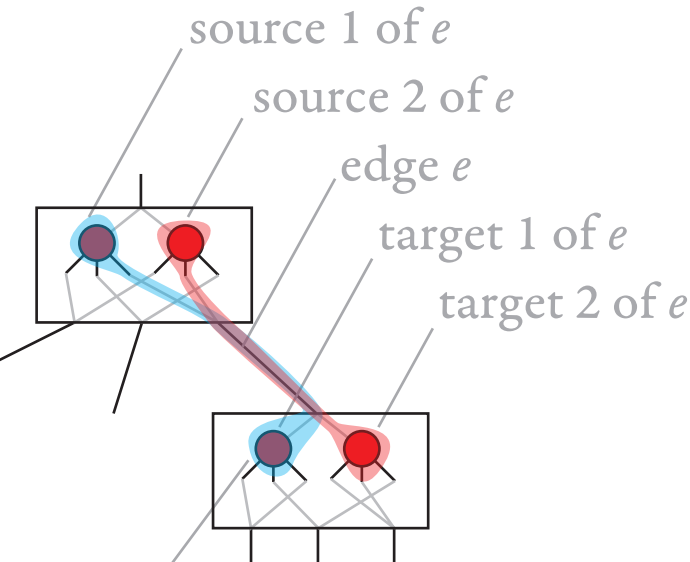
input



output







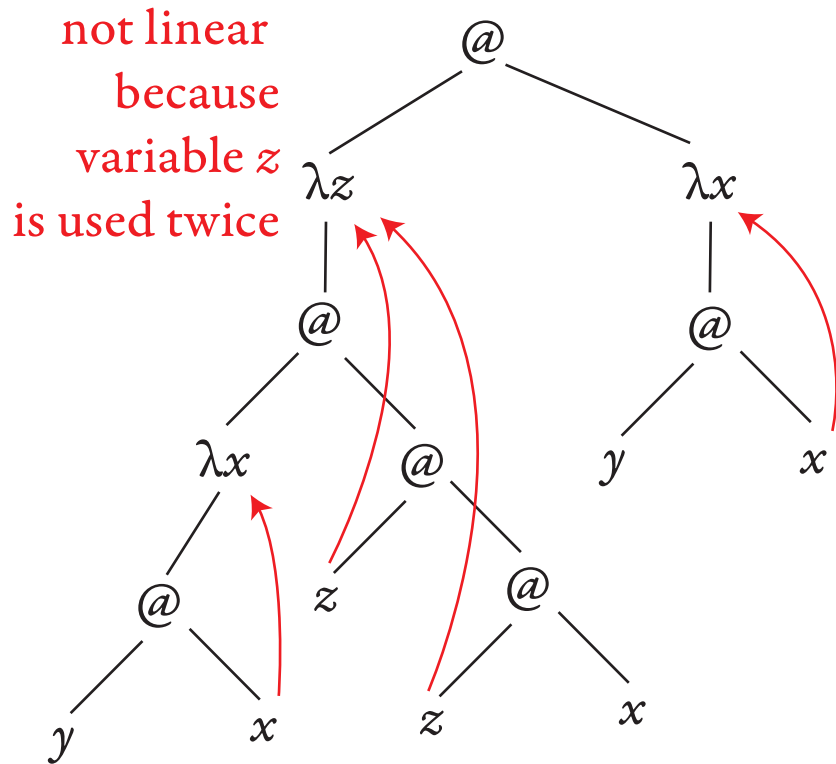
linear



we only count
variables used
in their scope

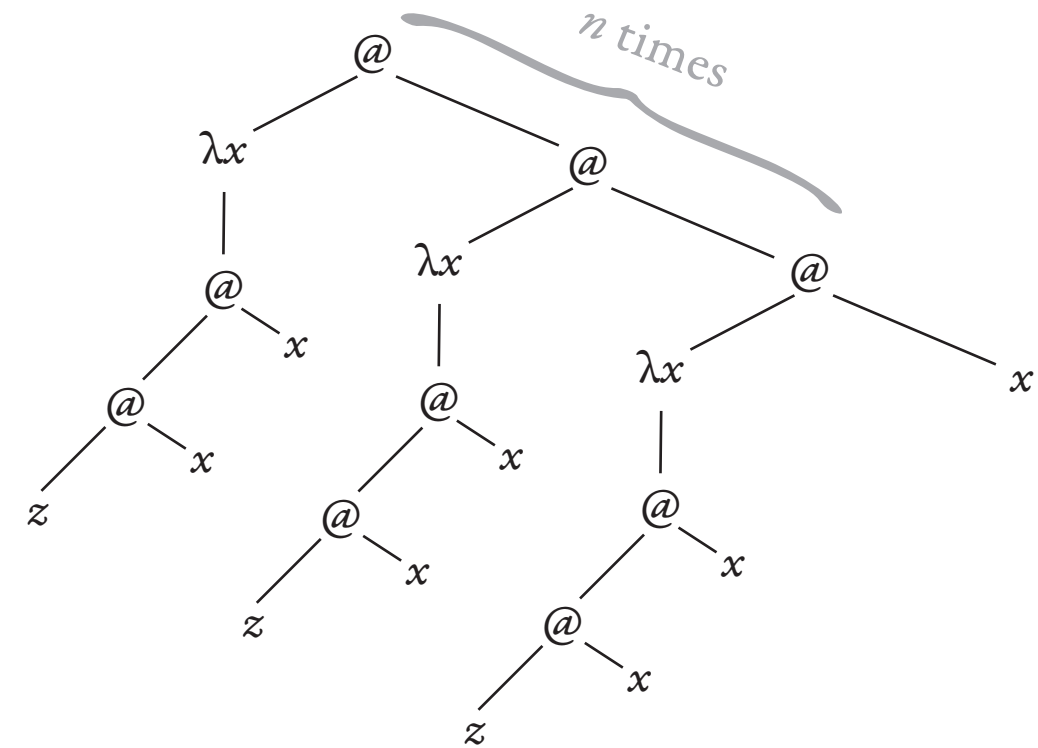
variable z can be used twice because it is free

not linear

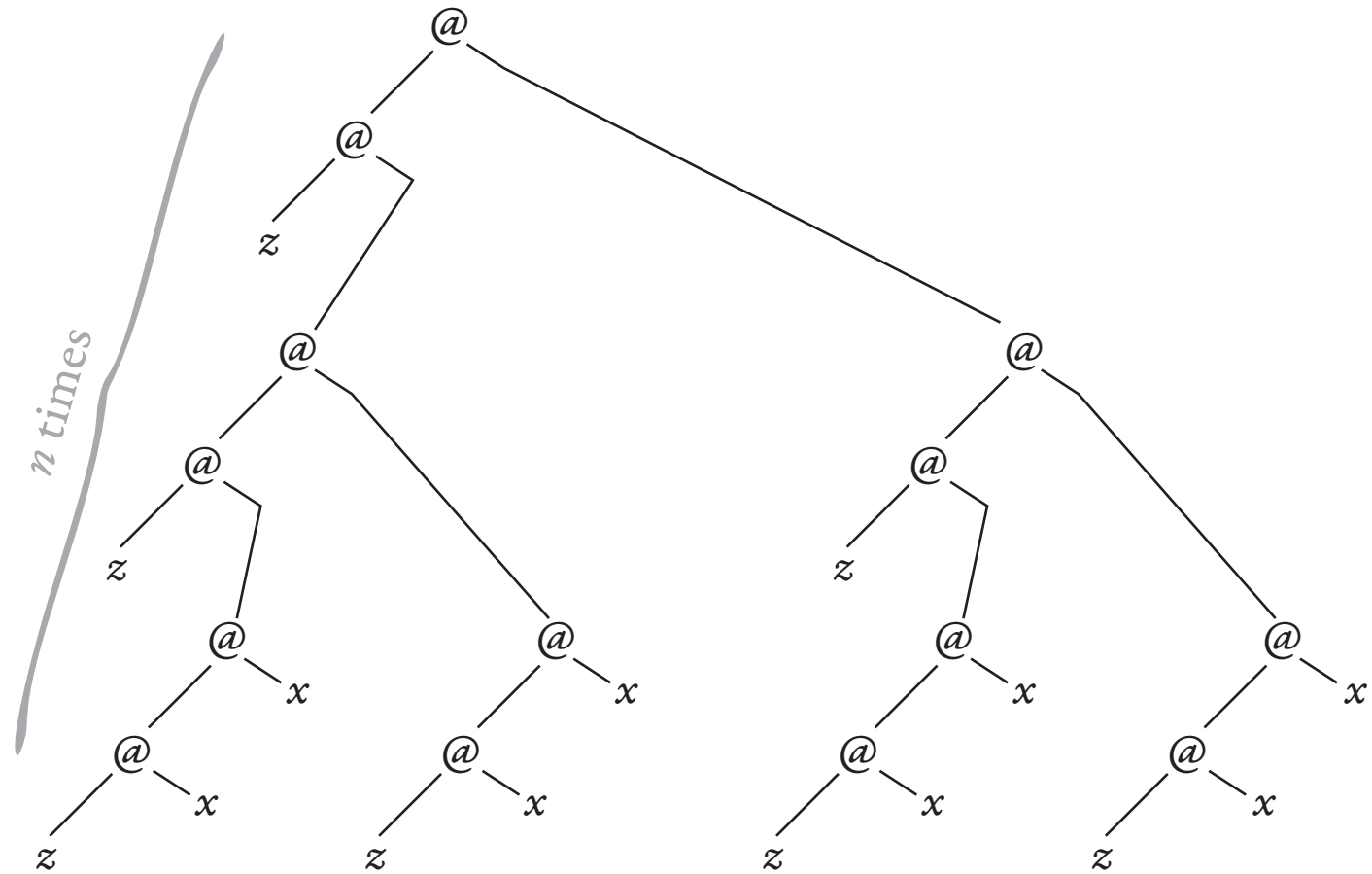


not linear
because
variable z
is used twice

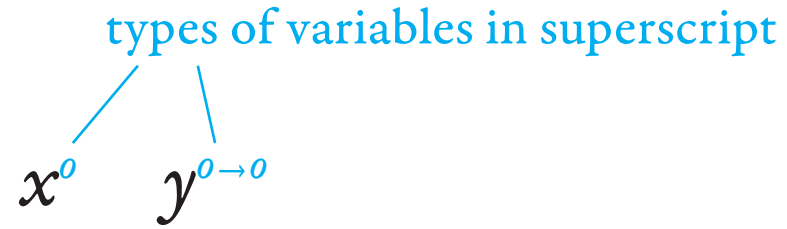
a λ -term of size $O(n)$



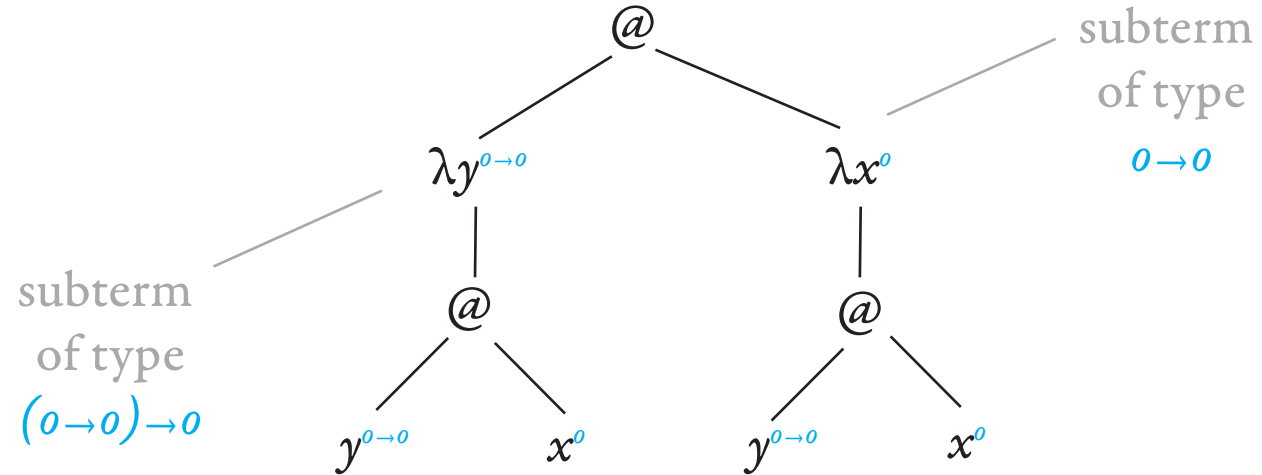
its normal form of size $O(2^n)$



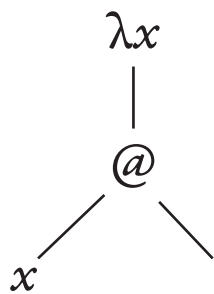
variables



λ -term of type o



@



$\lambda x.$

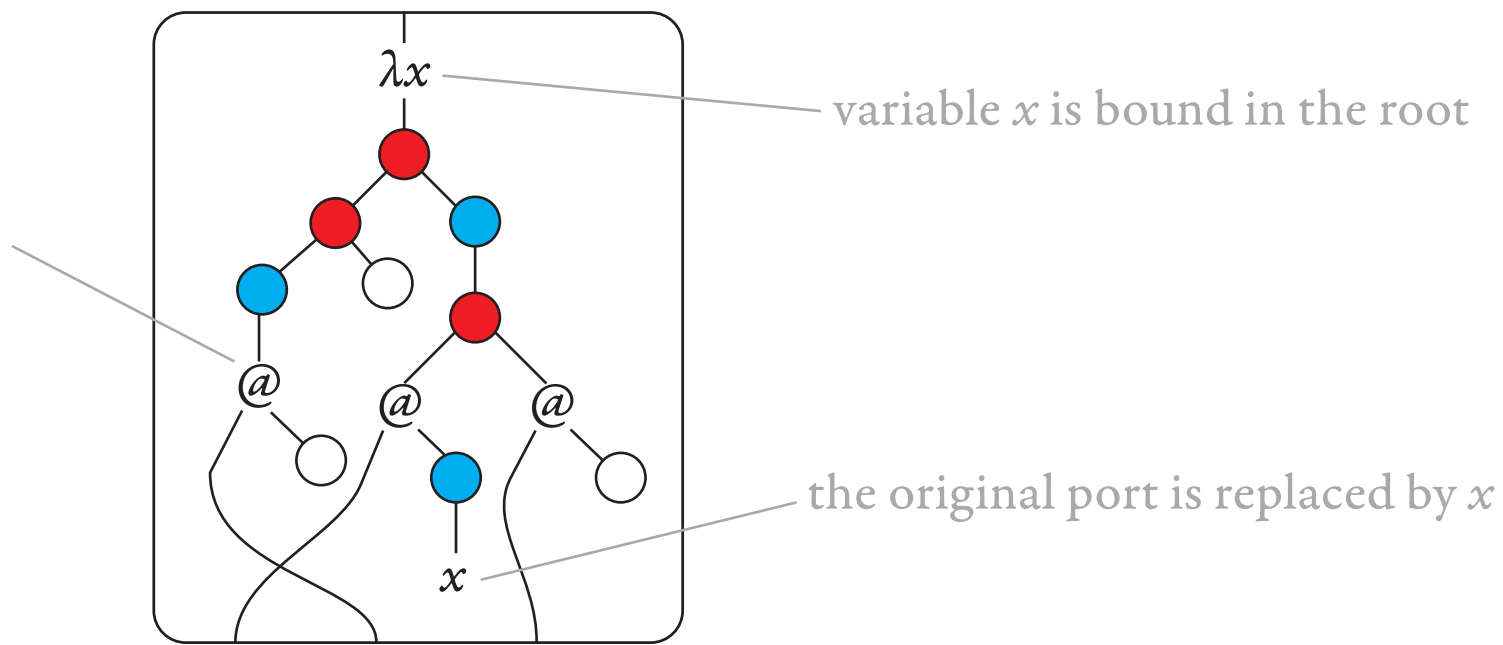


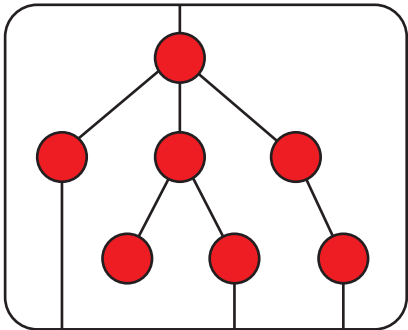
r

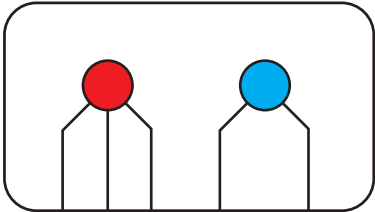


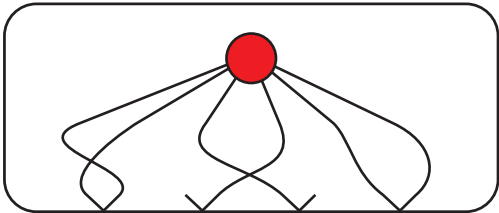
placeholder for the term
stored in the unique register
of the 2nd child









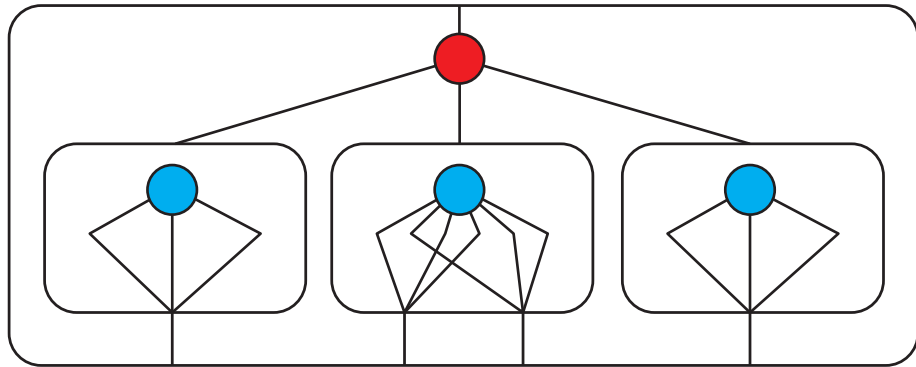


the root is from Σ

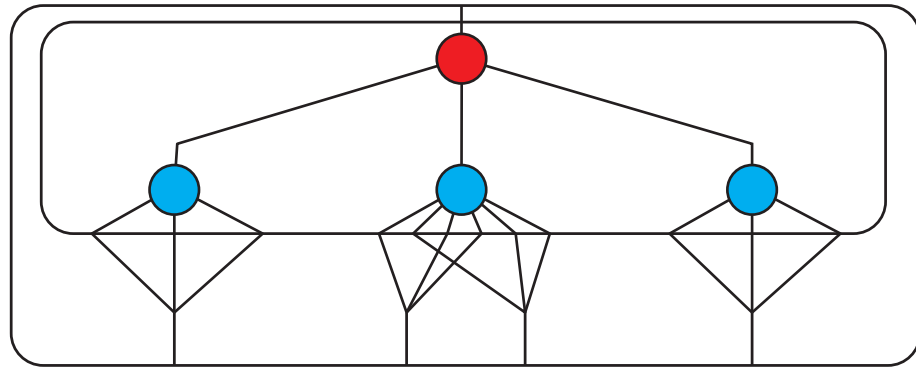
all children are from Γ

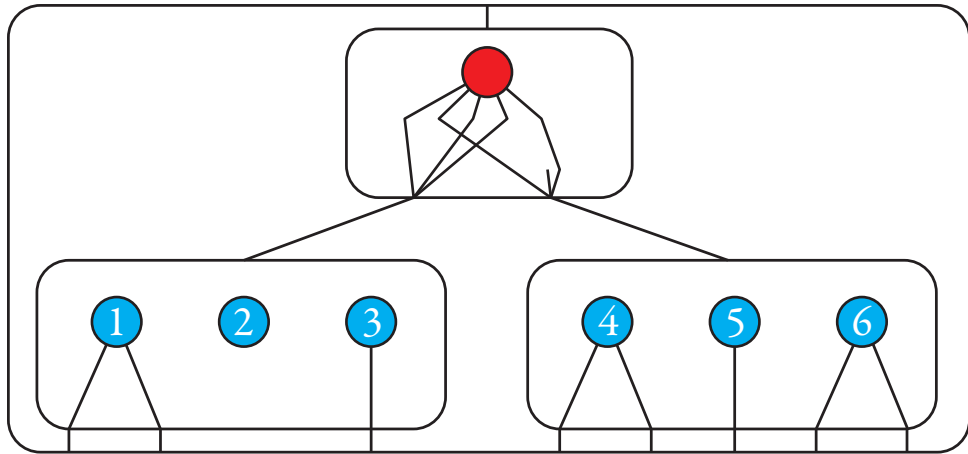


input

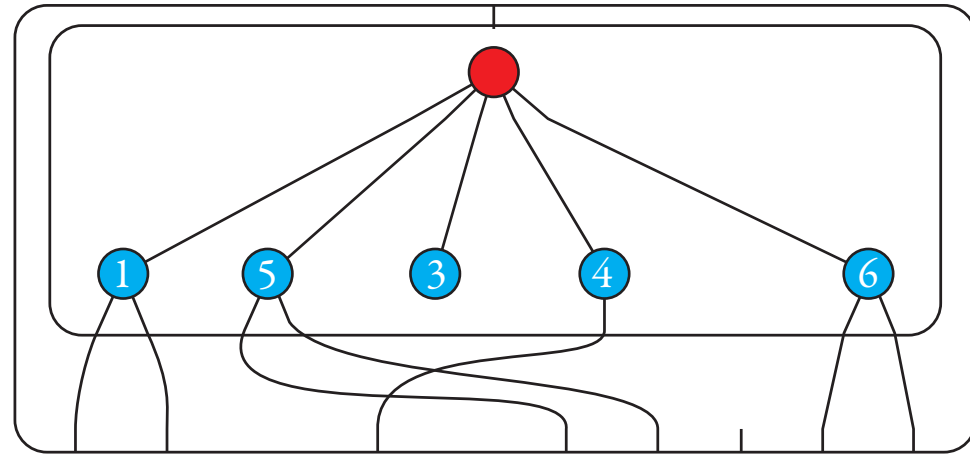


output

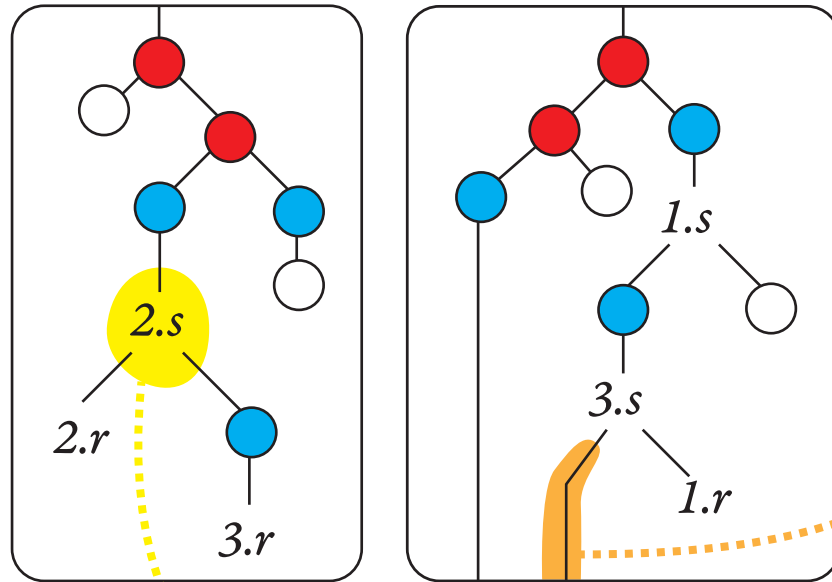




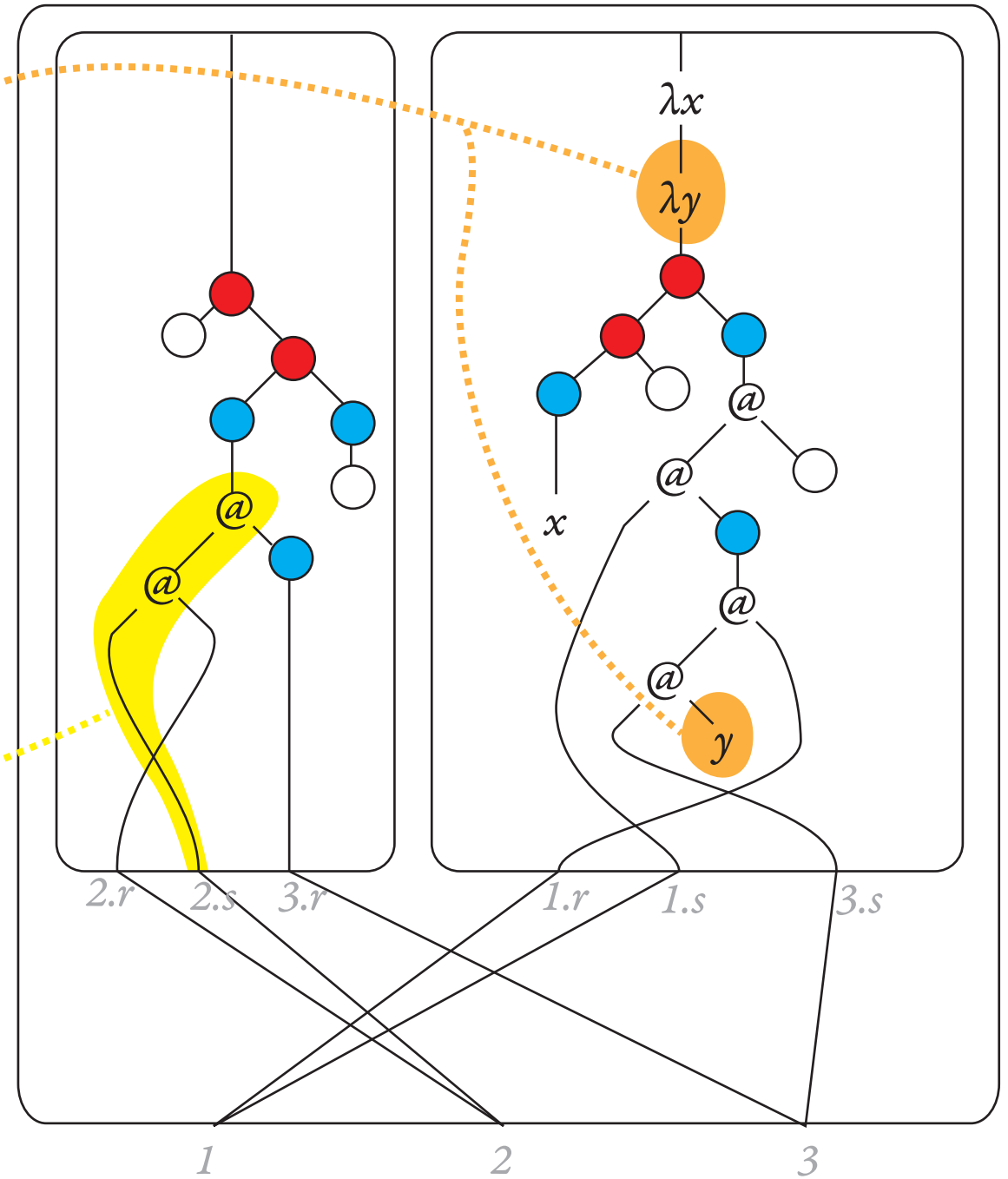
\mapsto



a register update



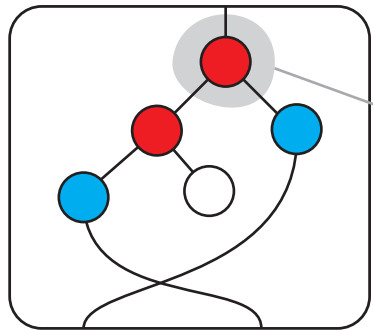
its dual



The diagram shows a binary tree structure. The root node is red. Its left child is red, and its right child is blue. The red node's left child is blue, and its right child is white. The blue node's left child is blue, and its right child is white. A yellow circle labeled r_1 highlights the blue node that is the right child of the root. An orange shape labeled r_2 highlights the subtree rooted at the blue node that is the left child of the root. A dashed orange line labeled r_3 indicates a path from the root to the right child of the root, and then to the right child of that node.

[illegible]

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

