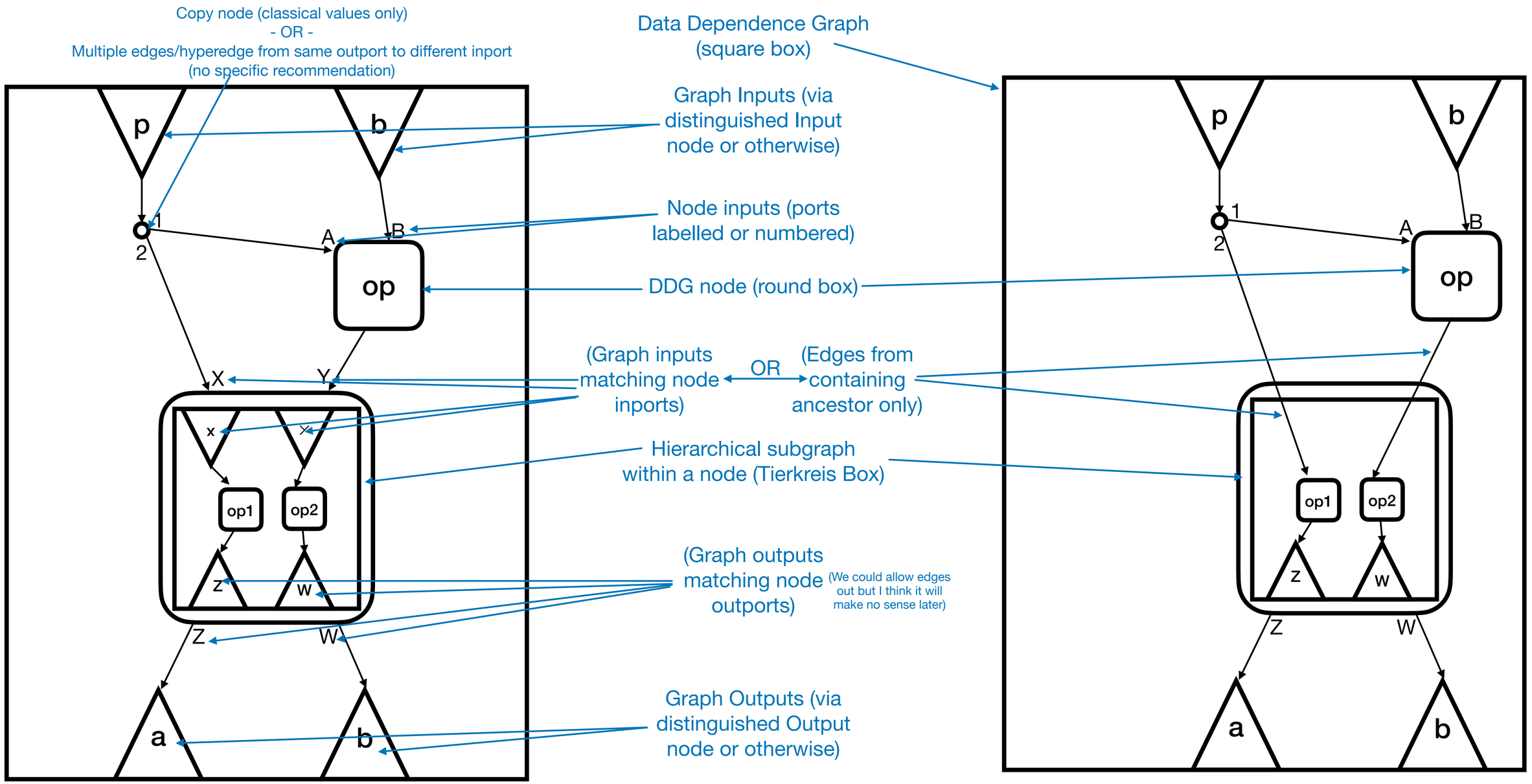
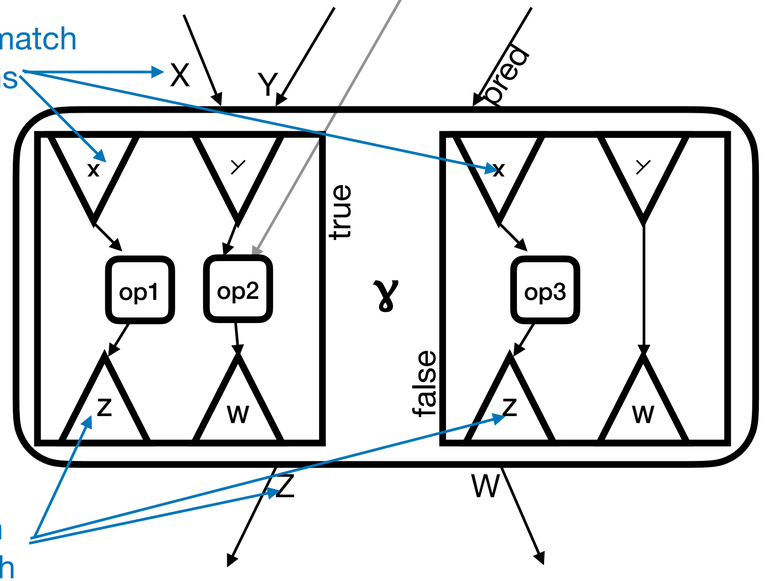
Basic Hierarchical Graph IR (For anyone familiar with Tierkreis, there is nothing new here!)



DDG must be acyclic; semantics: all nodes are evaluated, in some order respecting dependence edges

Simple conditionals via ɣ-nodes

A DDG node containing two, disjoint, sub-DDGs



Edges from containing graph??
(Classical values used on one side only)

(Restricts ordering just as if target was the ɣ-node itself)

Non-pred inports match both subgraphs

(Note: these are effectively σ-functions as per next slide)

Semantics: evaluating the ɣ-node evaluates exactly one subgraph (according to pred)

(Edges from subgraphs make no sense as the source nodes may not be evaluated at runtime.)

Outputs of both subgraphs match outports

…and something similar for simple loops
(e.g. while loop where output types==inputs)

Complex Control Flow via explicit CFGs

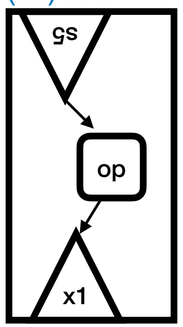
CFG instead of DDG
(e.g. inside Box- or ɣ-node)

Explicit control flow edges

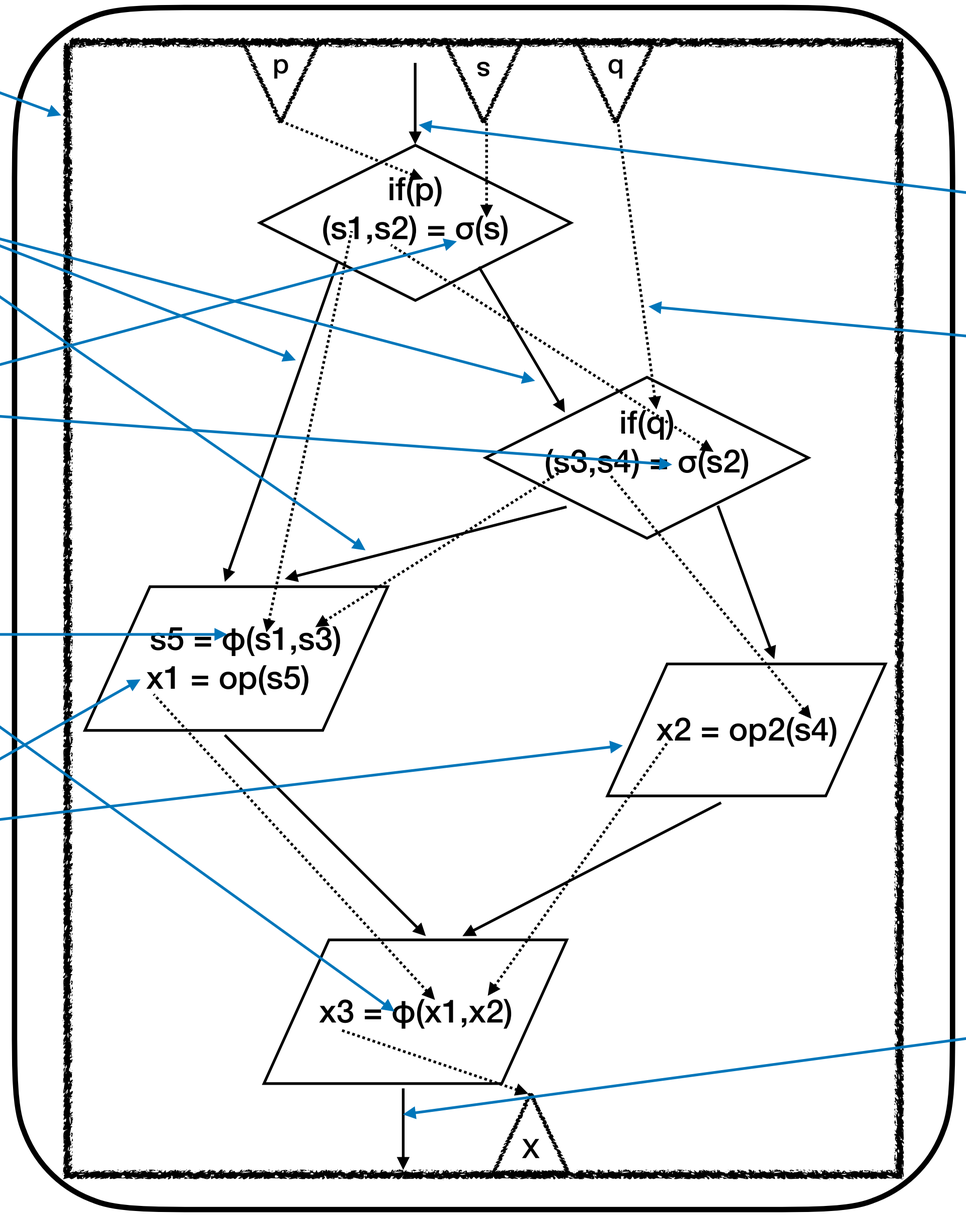
Sigma (σ) function - SSI form
Dual of phi (ϕ):
outputs to one or other of s1 or s2

Phi (ϕ) function - SSA form
(Maybe an explicit CF-merge node?)

Body of CFG node is actually a DDG
x1=op(s5) shorthand for:



CFGnode ::=
ϕ*<body_ddg>(<ifcond>σ*)?



Unique entry edge / node

Dataflow edges (use/def)
(presumably?)

from dominator only

Unique exit edge / node

σ perhaps unnecessary for some
classical values (if used on one side only).

Can be elided:
(a1,a2)=σ(a) …. a3=ϕ(a1,a2) ==> a3=a