

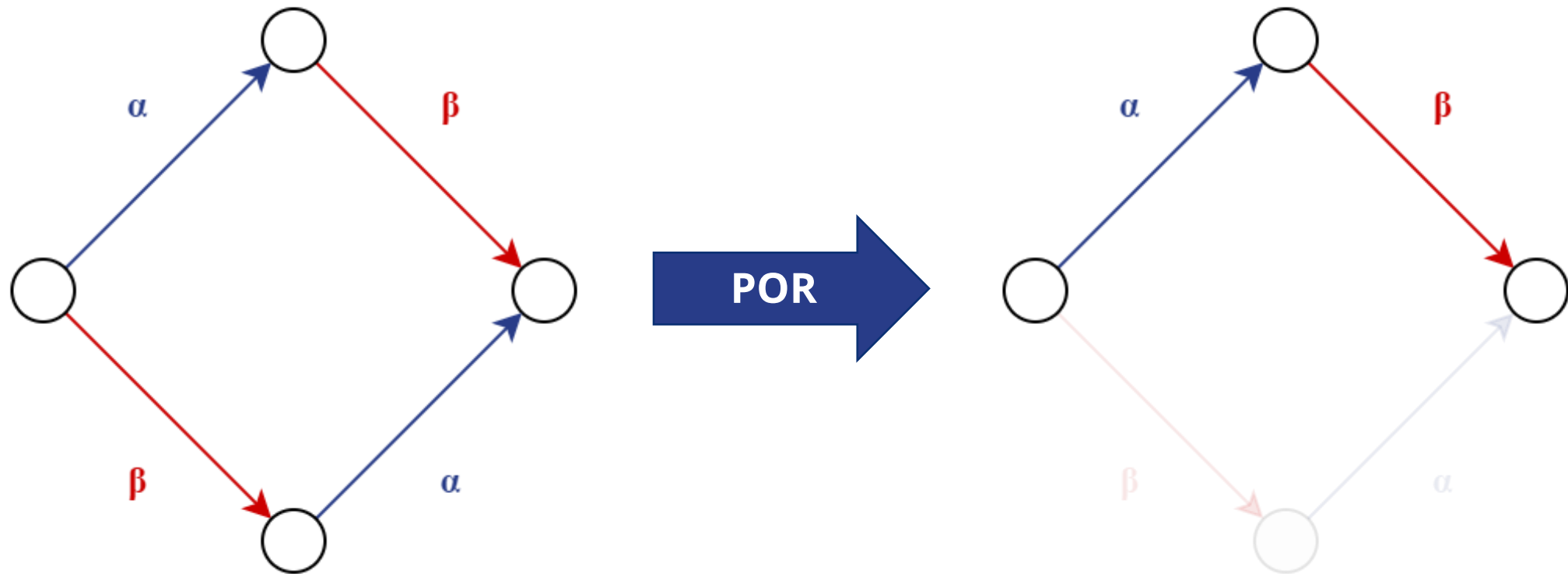
Partial Order Reduction for Abstraction- Based Verification of Concurrent Software

Csanád Telbisz
Vince Molnár



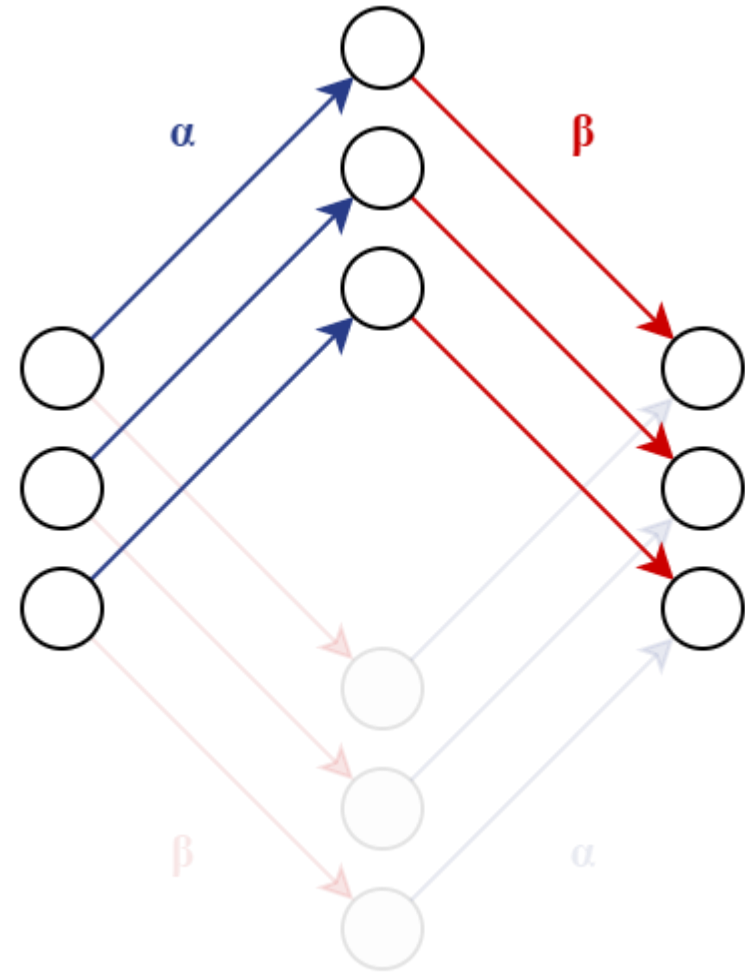
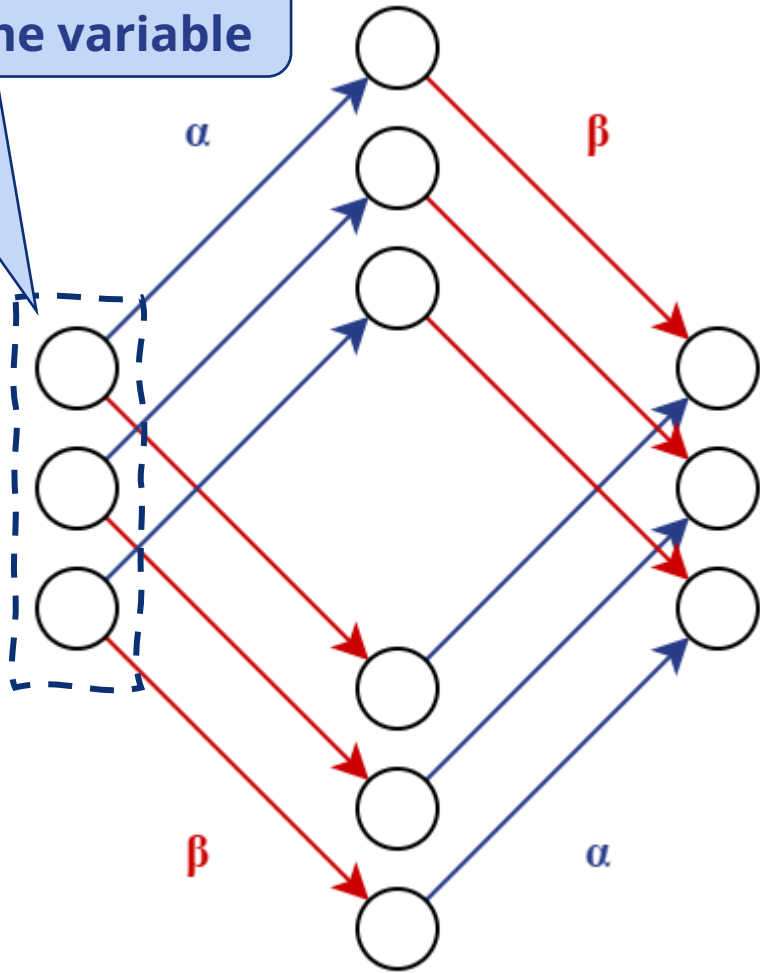
**Critical Systems
Research Group**

Partial Order Reduction (POR)

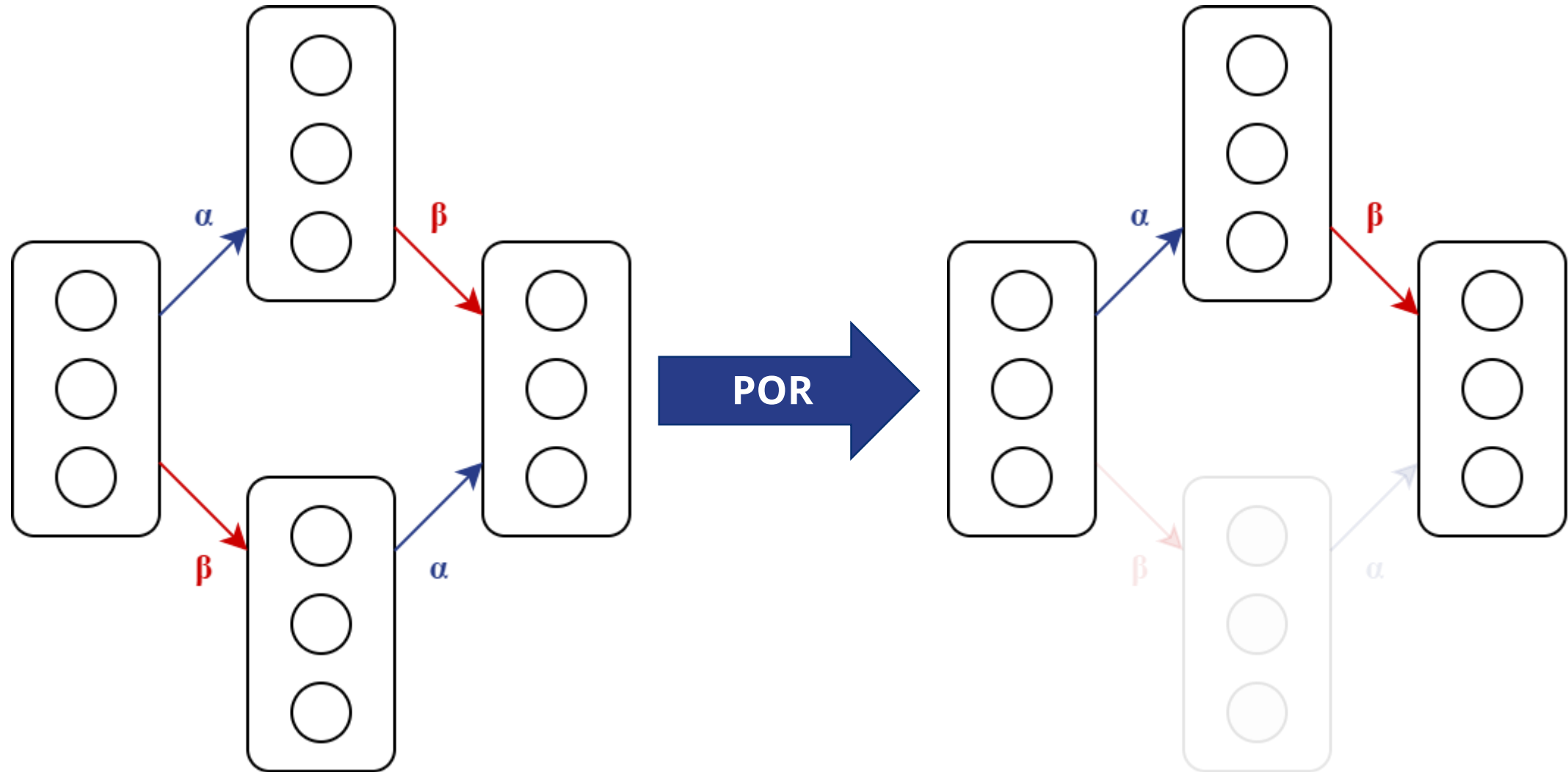


Partial Order Reduction (POR)

Different values of
the same variable



Abstraction + POR



Dependency Relation

Two operations α and β are independent if:

- α and β are in different processes
- there is no object that both α and β accesses

Abstraction-aware dependency relation

Two operations α and β are independent if:

- α and β are in different processes
- *all objects that both α and β accesses are abstracted away*

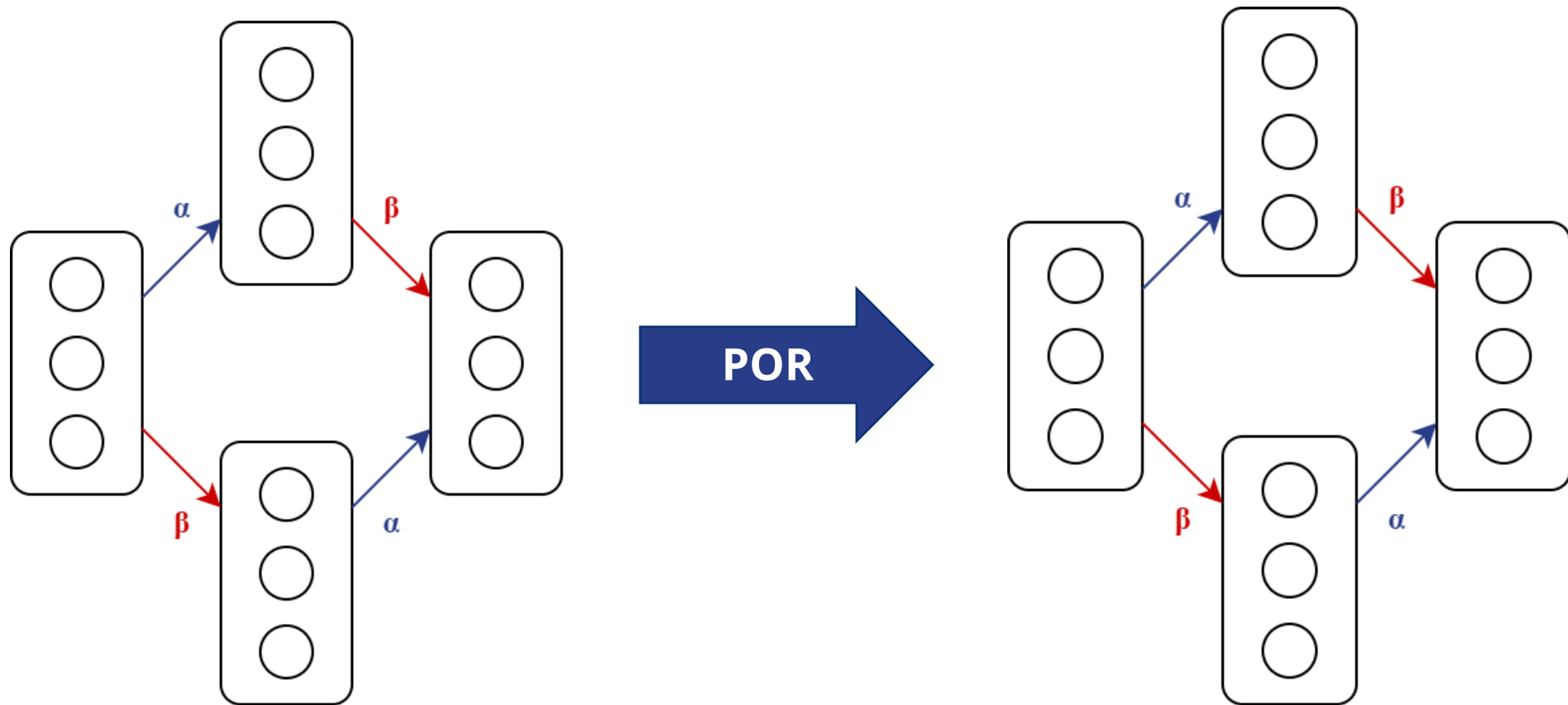
Example – Traditional POR

$\alpha: (x = 1)$

$\beta: (y = 2 * x)$

tracked objects: $\{y, z\}$

α and β are **dependent**



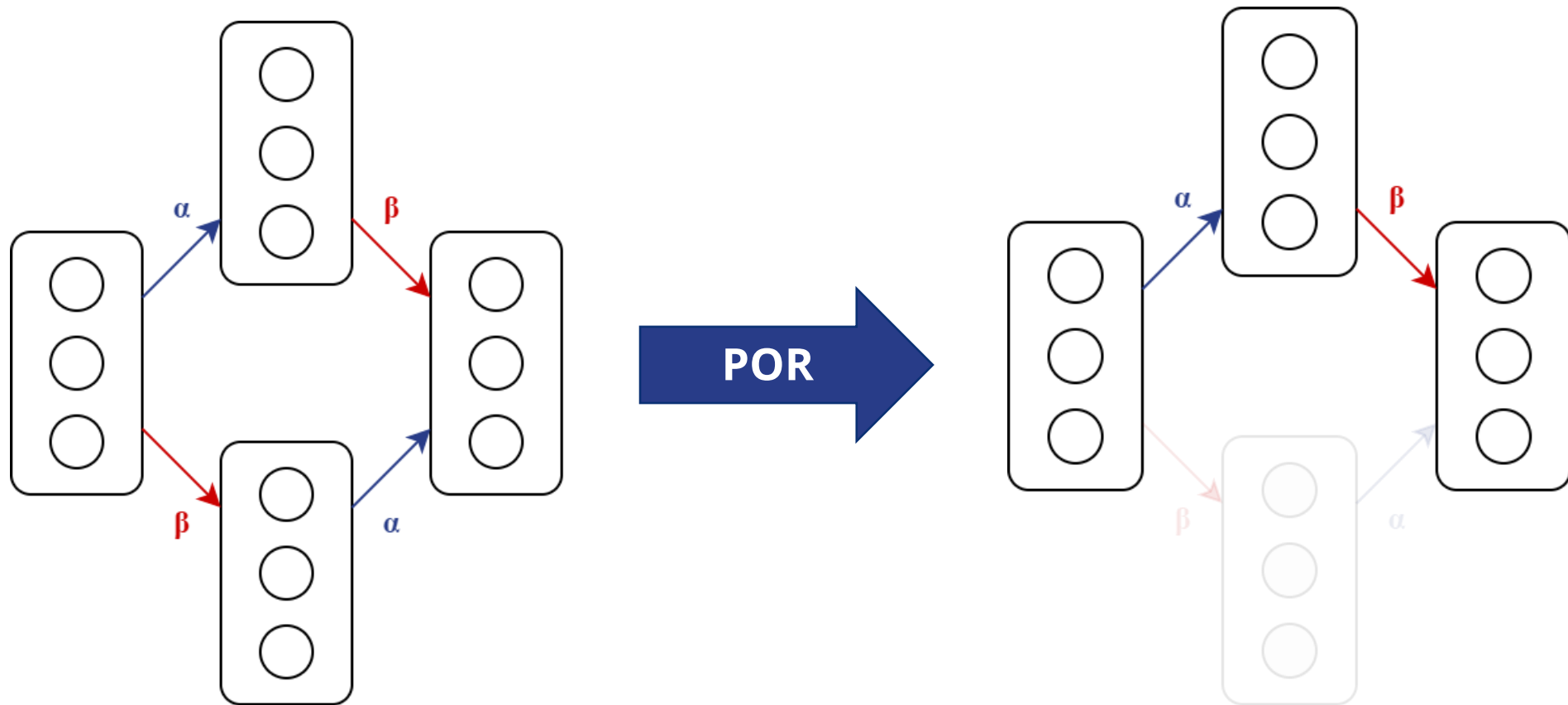
Example – Abstraction-Aware POR

α : ($x = 1$)

β : ($y = 2 * x$)

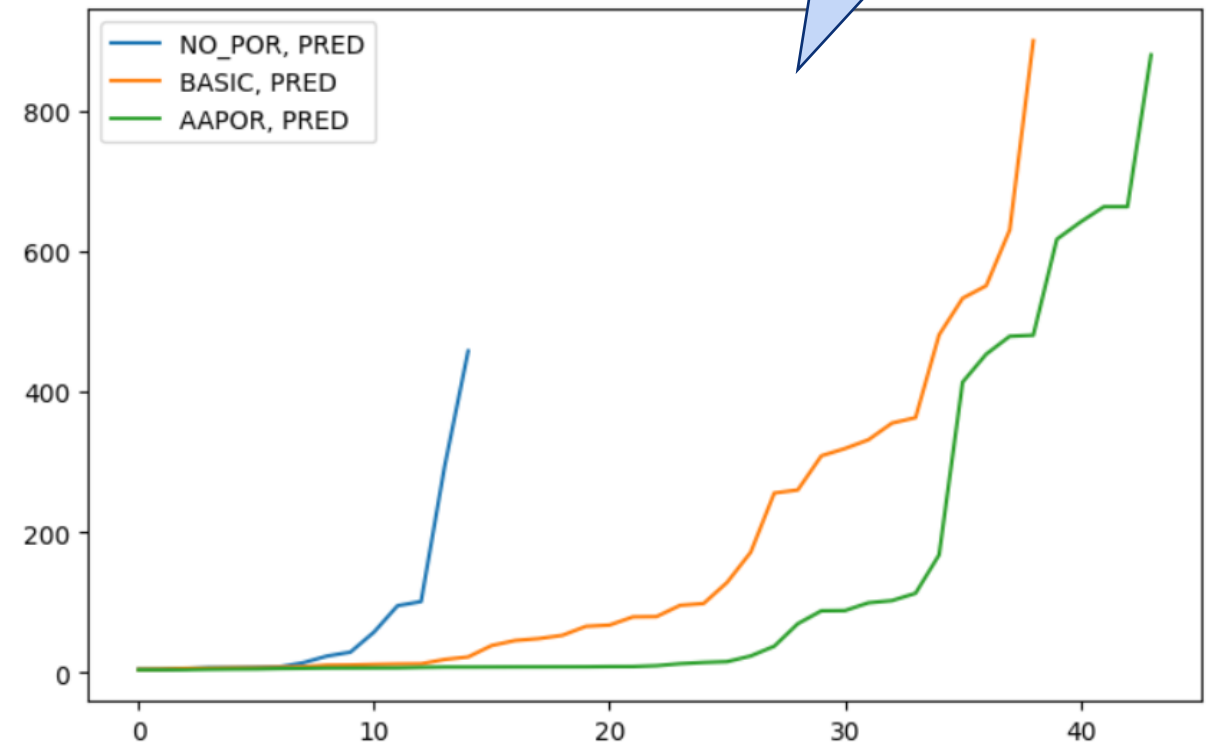
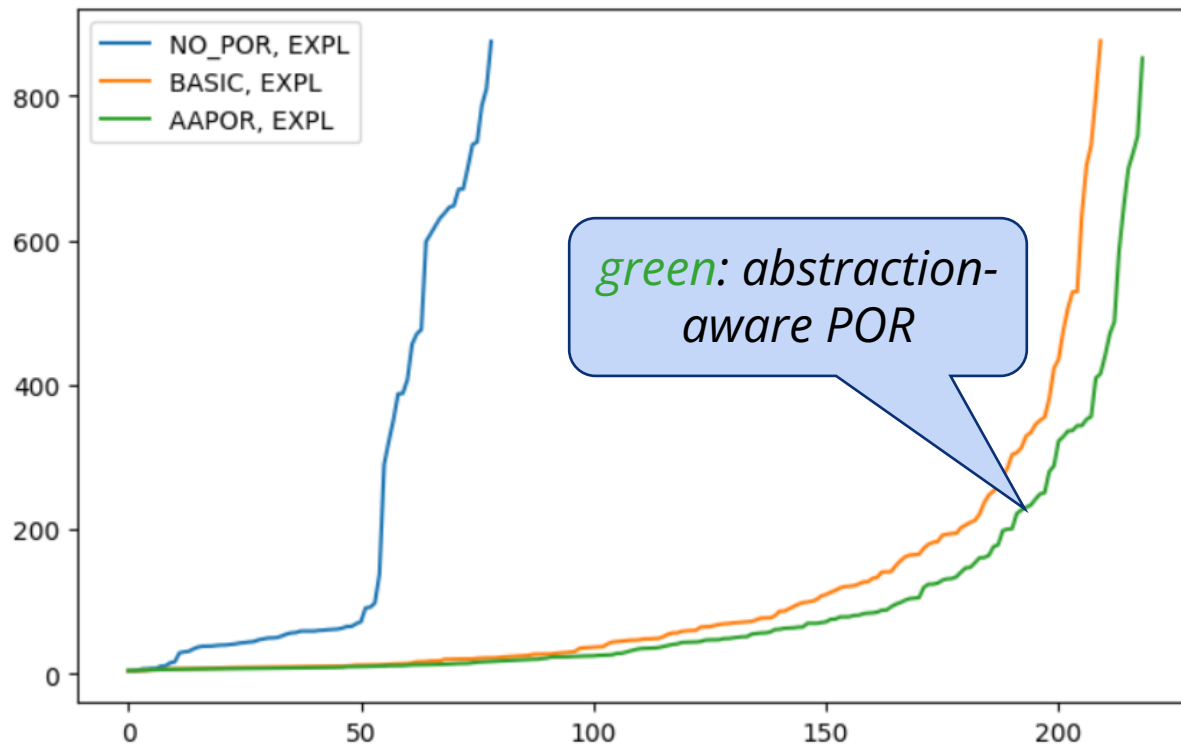
tracked objects: $\{y, z\}$

α and β are independent



Experiments

- Abstraction-aware POR is integrated into THETA
- 763 C benchmark programs from SV-COMP



Source: gitlab.com/sosy-lab/benchmarking/sv-benchmarks/-/tree/main/c
github.com/ftsrg/theta