

Investigation on Soundness Regarding Lazy Activities

Frank Puhlmann and Mathias Weske

Business Process Technology Group
Hasso-Plattner-Institute
Potsdam, Germany

Outline

- Motivation (Problem Statement)
- Soundness Classification (Related Work)
- Lazy Soundness (Solution)
- Conclusion

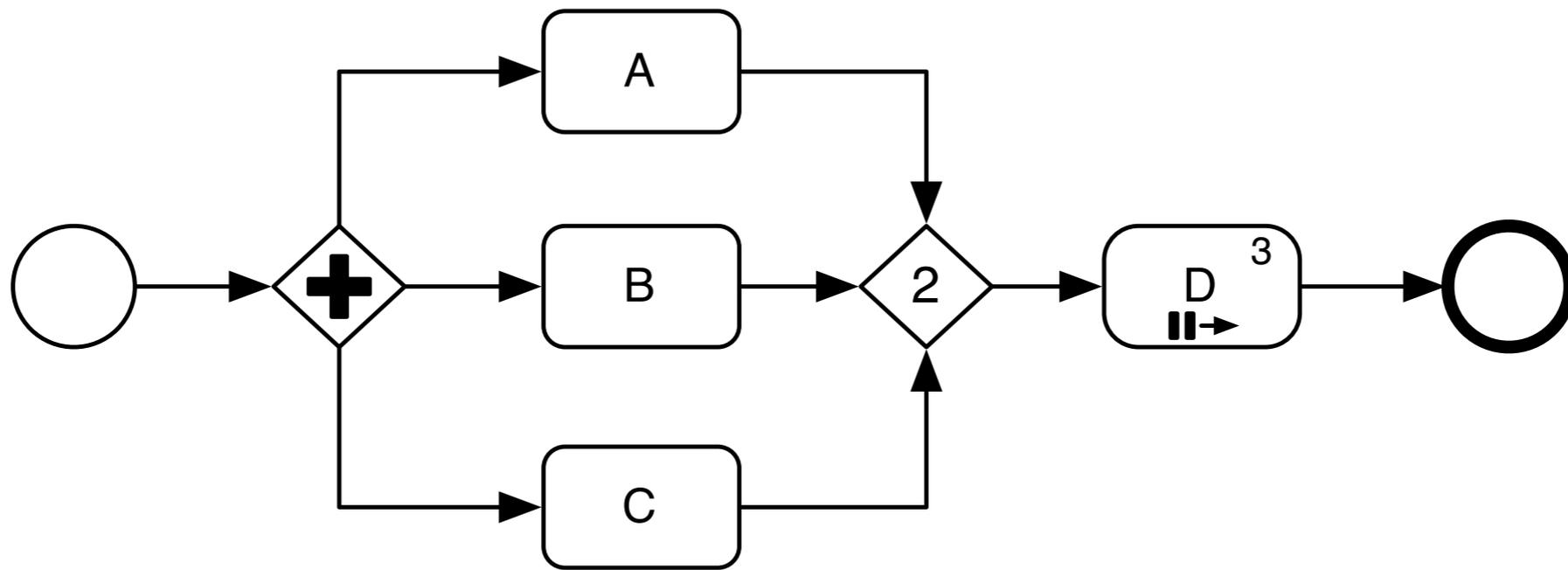
Motivation

Soundness (Informal)

- From each activity reachable from the initial activity, the final activity is reachable (i.e. the process is free of deadlocks and livelocks)
- After the final activity has been reached no other activities are active
- There are no unreachable activities (i.e. each activity participates in the process)

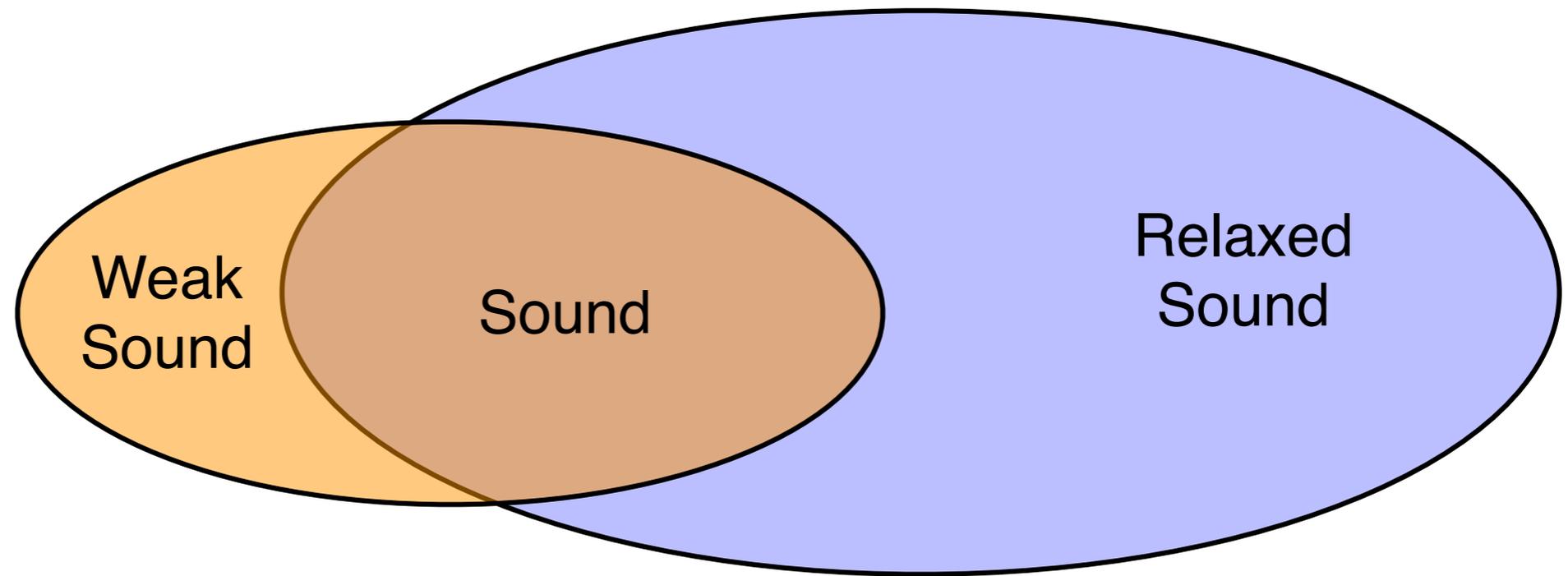
Motivation

- Some workflow patterns cause problems regarding soundness:
 - Discriminator
 - N-out-of-M-Join
 - Multiple Instances without Synchronization
- All these patterns can leave running (lazy) activities behind



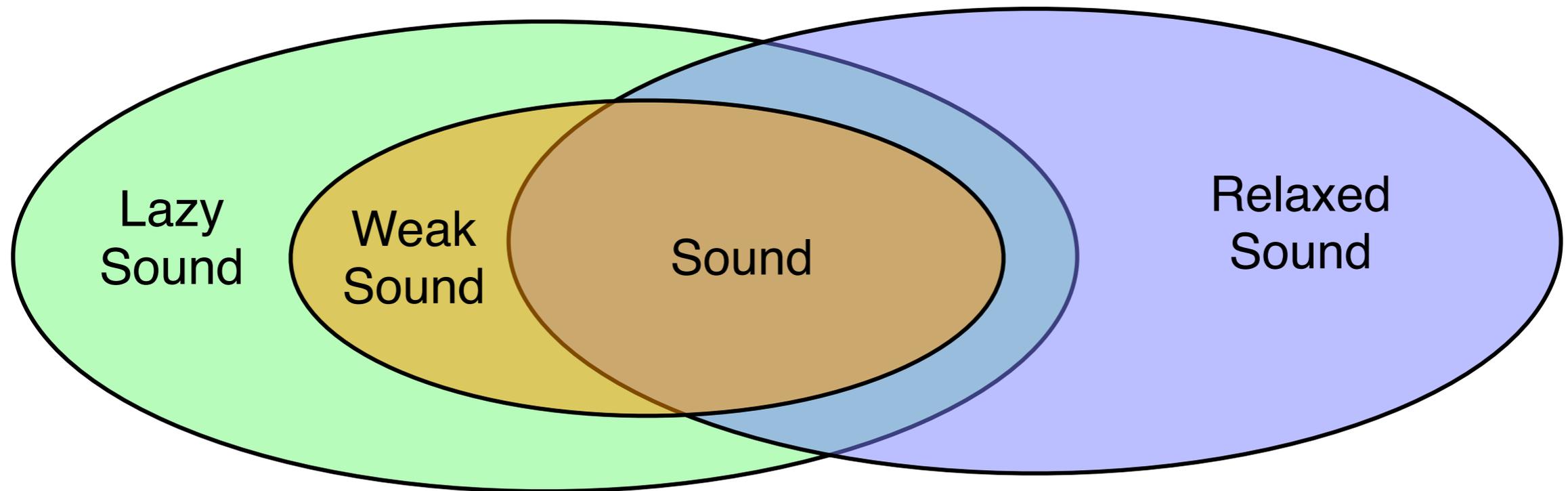
Example

Soundness Classification



[according to van der Aalst, Dehnert, Martens]

Soundness Classification



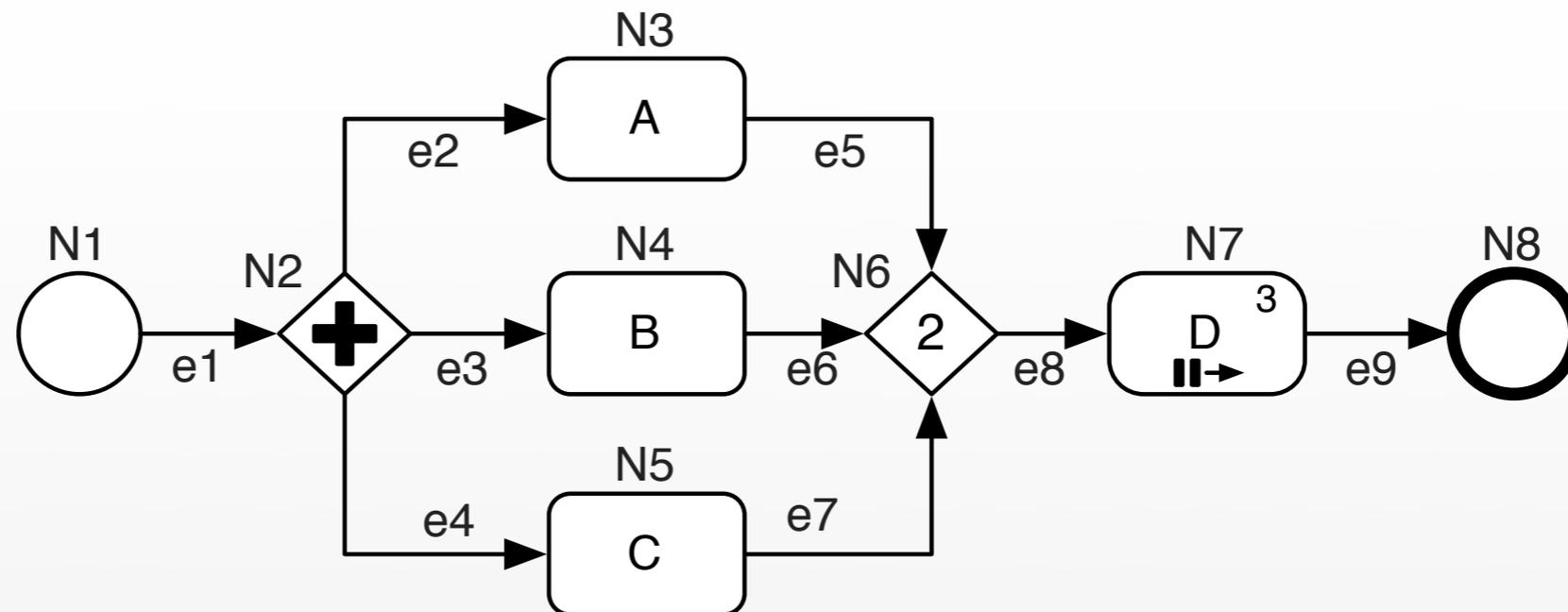
[according to van der Aalst, Dehnert, Martens]

Soundness Classification

Lazy Soundness

Process Graph

- A *process graph* formally defines the static structure of a business process as a four-tuple $P=(N,E,T,A)$:
 - N is a finite, non-empty set of nodes.
 - E is a set of directed edges between nodes.
 - T is a function mapping types to nodes.
 - A is a function mapping key/value pairs to nodes.



$N = \{N1, N2, N3, N4, N5, N6, N7, N8\}$

$E = \{ (N1, N2), (N2, N3), (N2, N4), (N2, N5), (N3, N6), (N4, N6), (N5, N6), (N6, N7), (N7, N8) \}$

$T = \{(N1, StartEvent), (N2, ANDGateway), (N3, Task), (N4, Task), (N5, Task), (N6, N-out-of-M-Join), (N7, MIwithoutSync), (N8, EndEvent)\}$

$A = \{(N6, (continue, 2)), (N7, (count, 3))\}$

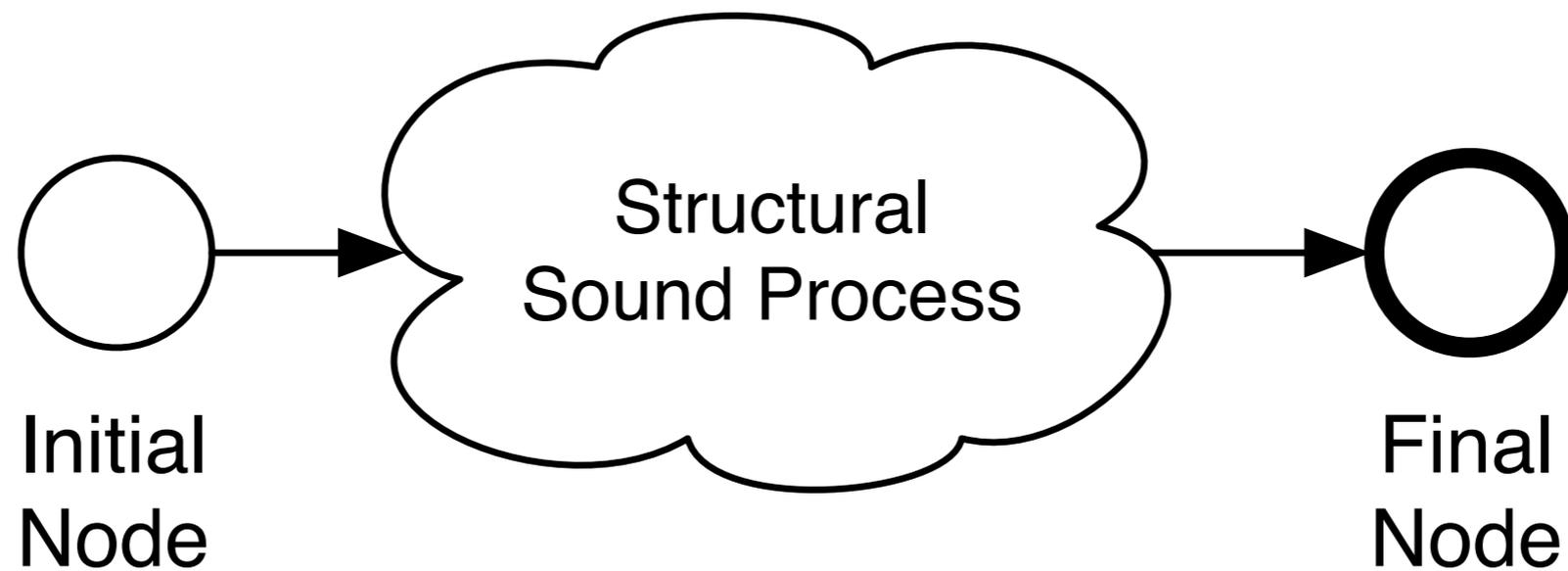
Process Graph Example

Structural Soundness

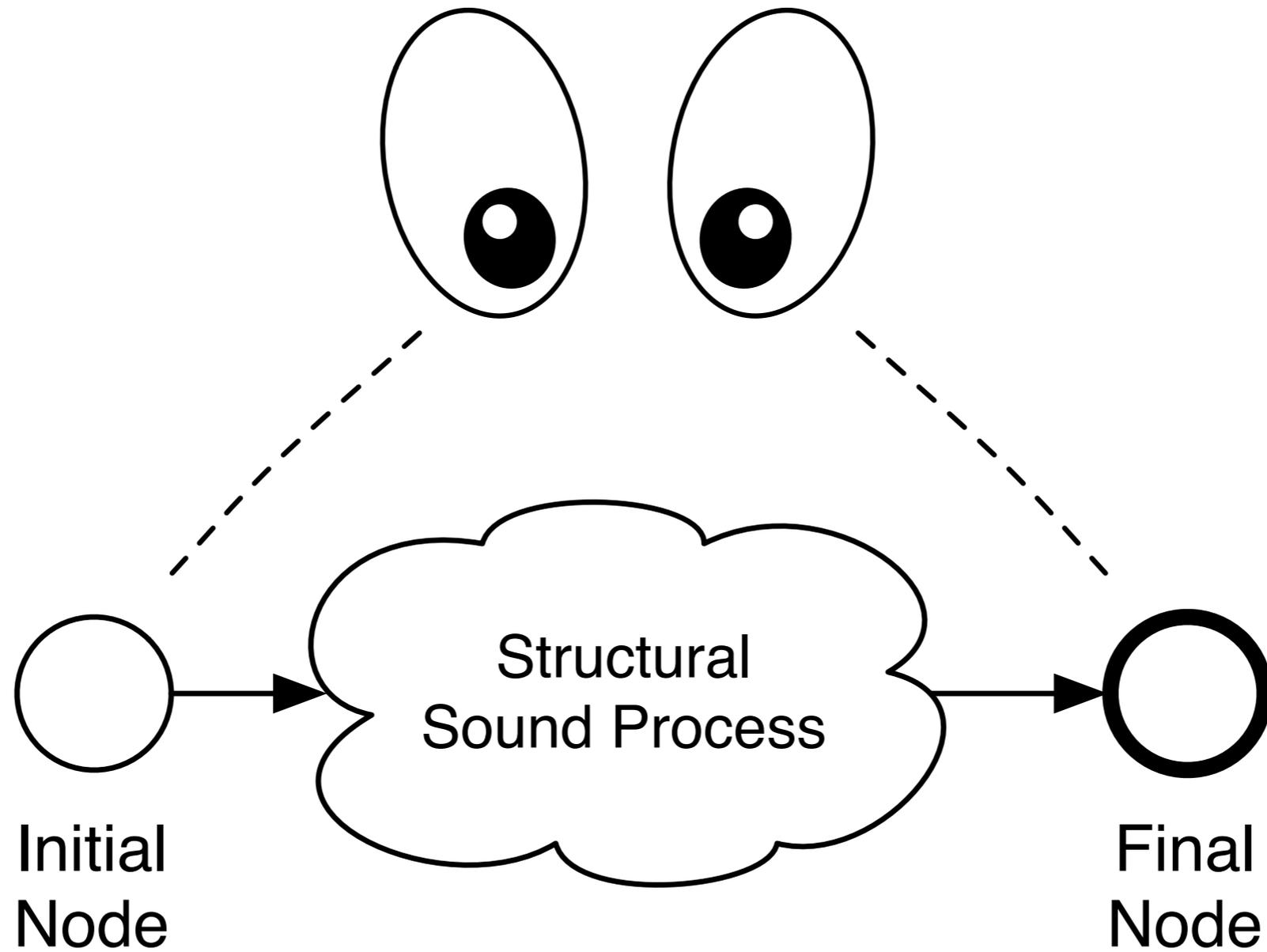
- A process graphs is structural sound iff:
 - There is exactly one initial node.
 - There is exactly one final node.
 - Every node is on a path from the initial node to the final node.
- Easy to show

Lazy Soundness

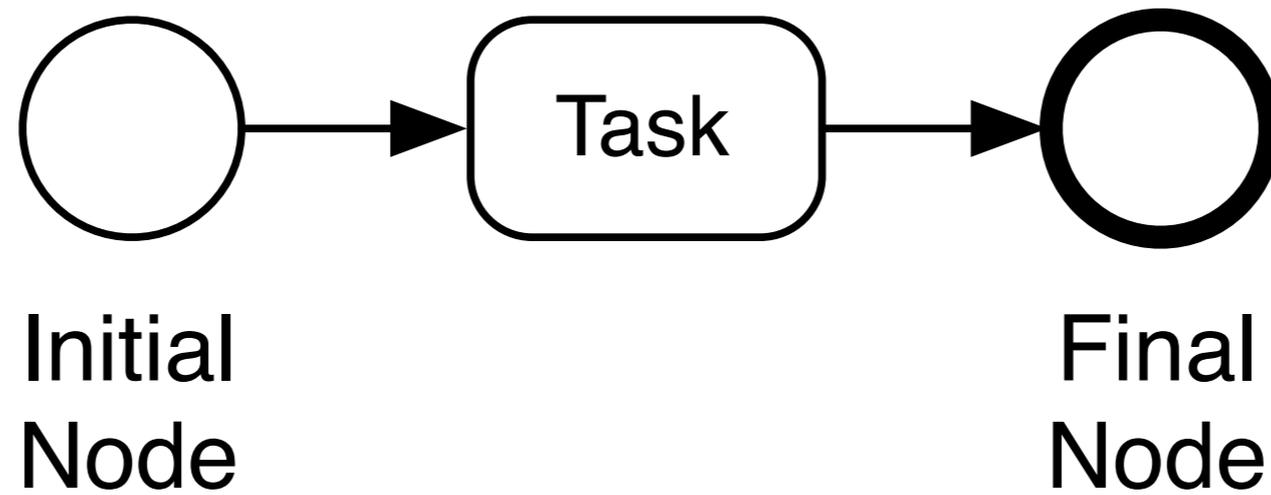
- Semantic Reachability:
 - A node of a process graph is semantically reachable from another node iff there exists a path leading from the first to the second node according to the semantics of all nodes.
- Lazy Soundness:
 1. The final node must be semantically reachable from every node semantically reachable from the initial node until the final node has been executed.
 2. The final node is executed exactly once.



Lazy Soundness Observation



Lazy Soundness Observation



Trivial Lazy Sound Process

Lazy Soundness in Pi-Calculus

- We observe the initial and the final activity by annotating the pi-calculus mapping of a process graph with i and o (initial, final activity)
- If we observe i and o exactly one time, the mapping is lazy sound
- Done by deciding $D \sim_{i,o}^o S_{LAZY}$
 - with $D = \text{pi-mapping}$, $S_{LAZY} = i.\tau.\bar{o}.0$

Conclusion

Conclusion

- New kind of soundness supporting „lazy“ activities
- Algorithms already implemented in prototypic tool chain
- First approach utilizing pi-calculus for soundness

Thank you!