



Compositionality Seminar

Bisimulation Can't Be Traced



Bisimulation Can't Be Traced

- Bard Bloom
- Sorin Istrail
- Albert R. Meyer
- July 1990



Compositionality and Formal Methods

- Process Description
- Specification and Verification
- Exchangeability of Processes



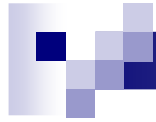
Key Contributions

- Checkable condition for exchangeability
 - Necessary, but not sufficient
- General rule-format for process algebras
- Sufficient condition for specific extensions
- Seminal contribution
- Defines today's status quo



Overview

- Process Description
- Specification & Verification
- Process Equivalencies
- Testing
- Formats & Rules



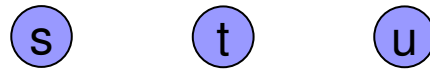
Process Description

- Labelled Transition Systems
- Calculus of Communicating Systems



Labeled Transition System

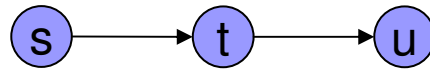
states



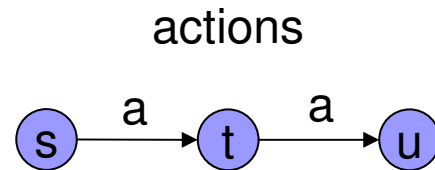


Labeled Transition System

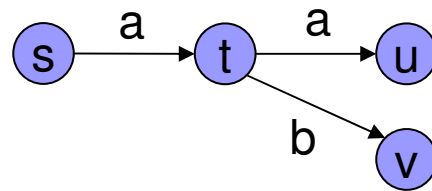
transitions



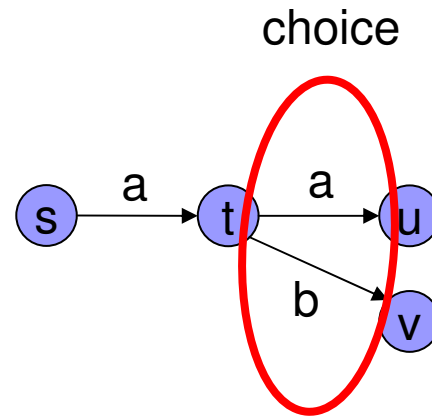
Labeled Transition System



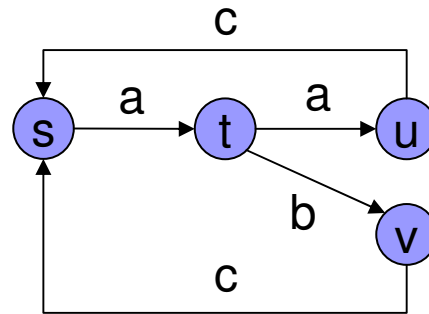
Labeled Transition System



Labeled Transition System



Labeled Transition System





Labeled Transition System

- Formally...

$$(S, Act, (\xrightarrow{a} \mid a \in Act)),$$
$$s \xrightarrow{a} s' \in (\xrightarrow{a} \mid a \in Act).$$



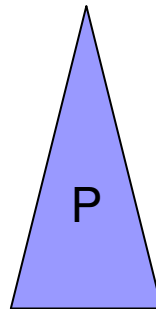
Calculus of Communication Systems

- Robin Milner
 - A Calculus of Communicating Systems, 1980
- Minus: communicating
- ...explained using LTSes



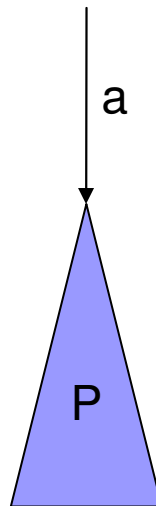
Calculus of Communication Systems

■ Process P



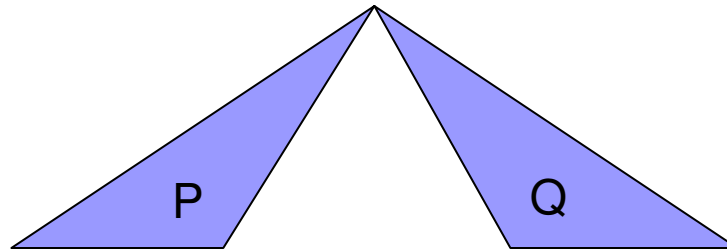
Calculus of Communication Systems

■ Process $a.P$



Calculus of Communication Systems

- Process $P + Q$





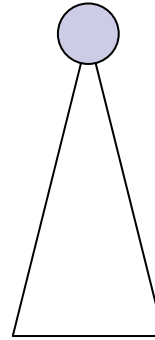
Specification & Verification

- Specify desired behavior
- Model your implementation
- Verify that your implementation meets the specification



Verification using LTS

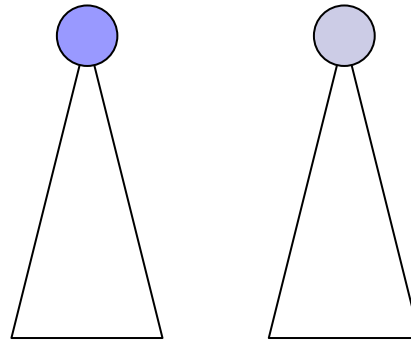
specification



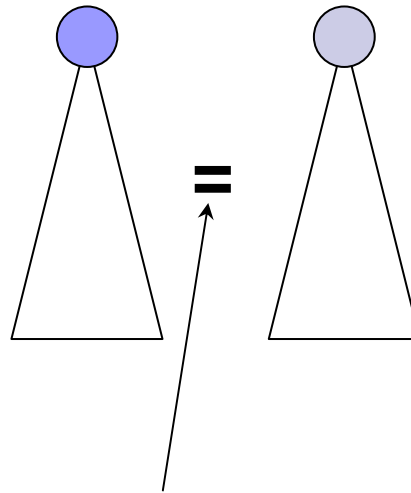


Verification using LTS

implementation

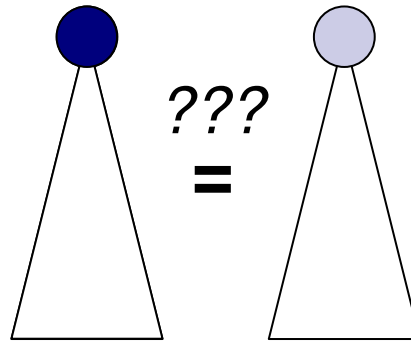


Verification using LTS



process equivalence

Compositionality on LTS





Compositionality on LTS

- What is the catch?
 - Process equivalence



Compositionality on LTS

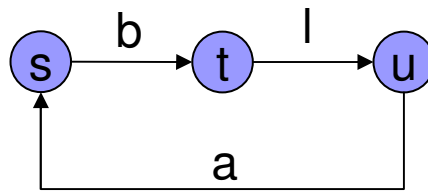
- What is the catch?
 - Process equivalence
 - Appropriate modeling
 - Specification
 - Implementation



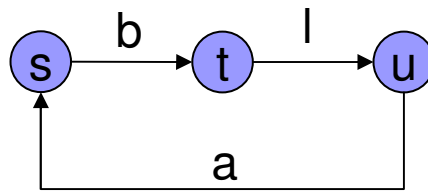
Trace Equivalence

- LTS as „talking boxes“
- Compare what they are saying

Traces

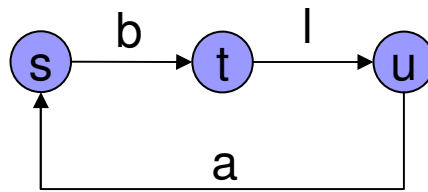


Traces



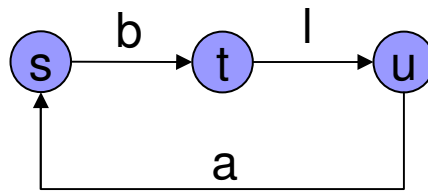
$\text{tr}(s) =$

Traces



$\text{tr}(s) = \text{blablabla} \dots$

Traces



$$\text{tr}(s) = \text{blablabla}\dots = (\text{bla})^*[\text{b}[\text{l}[\text{a}]]]$$



Traces

- Trace equivalence

$$\equiv_{tr}$$

- Trace congruence

$$\subseteq_{tr}$$



Compositionality on LTS

- Exchangeability

$$\underline{\subseteq}_{tr}$$



Compositionality on LTS

- Exchangeability

$$\subseteq_{tr}$$

- High complexity!



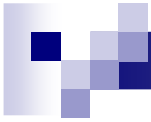
Compositionality on LTS

- We need something we can check automatically!

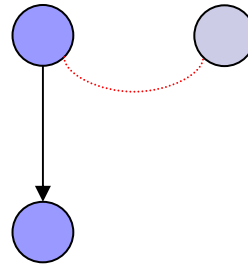


Bisimulation

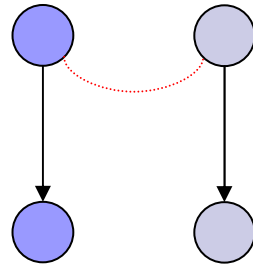




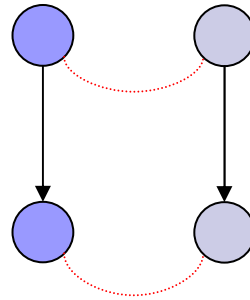
Bisimulation



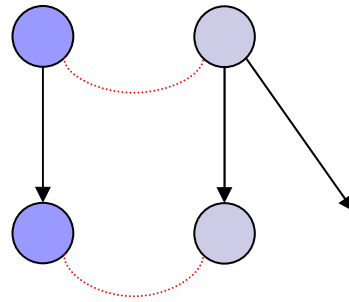
Bisimulation



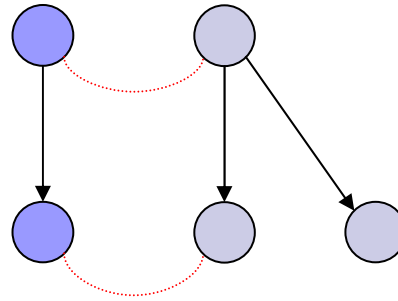
Bisimulation



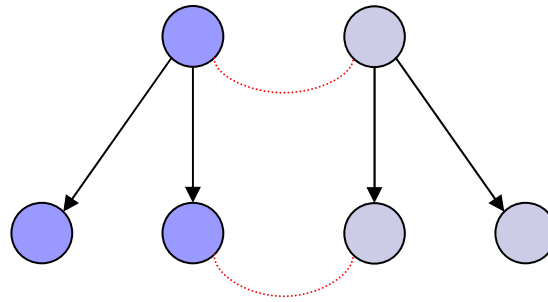
Bisimulation



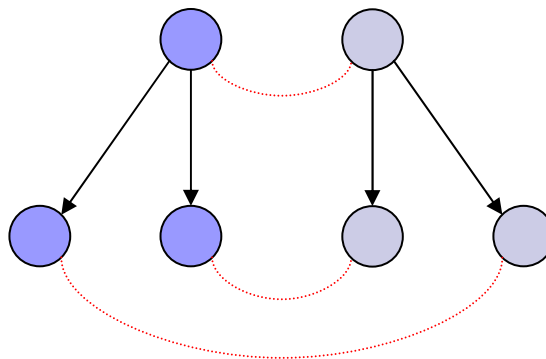
Bisimulation



Bisimulation

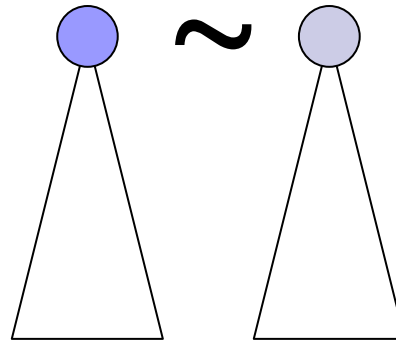


Bisimulation

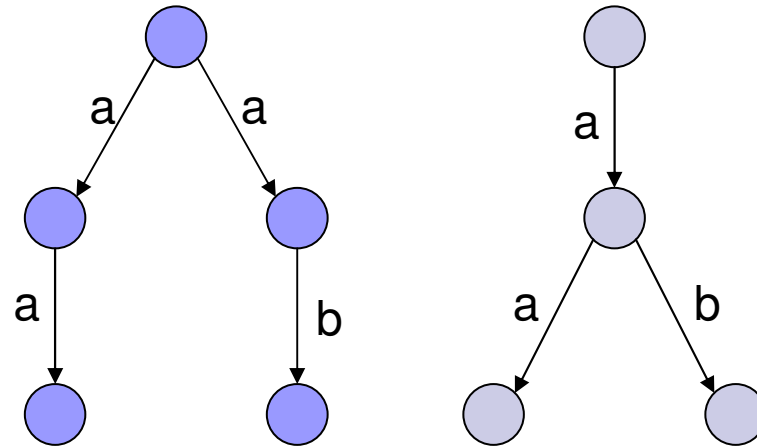




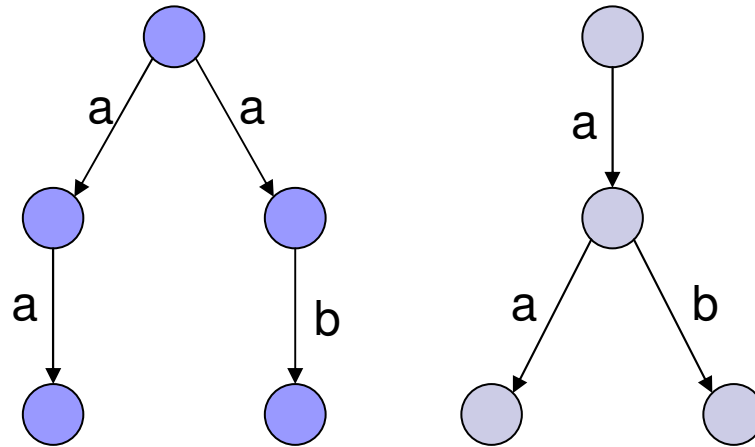
Bisimulation



Not Bisimilar



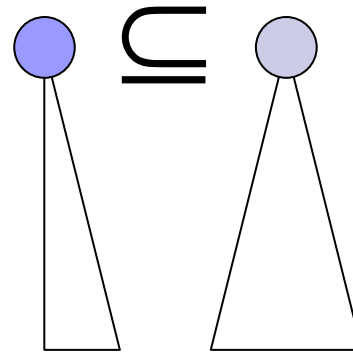
Not Bisimilar



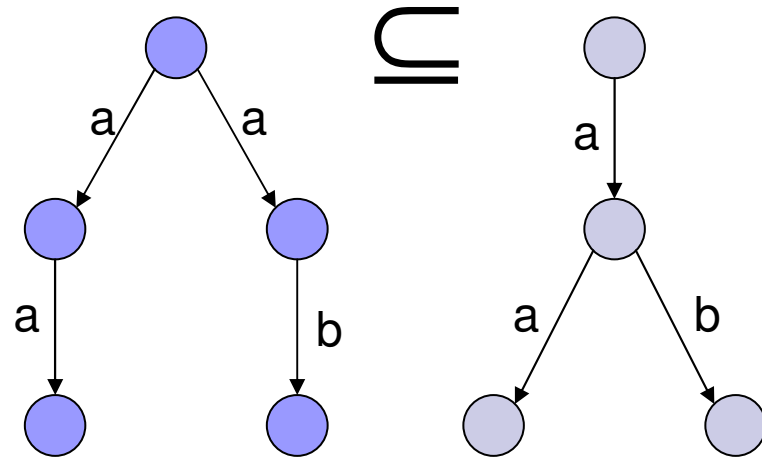
choice matters



Simulation

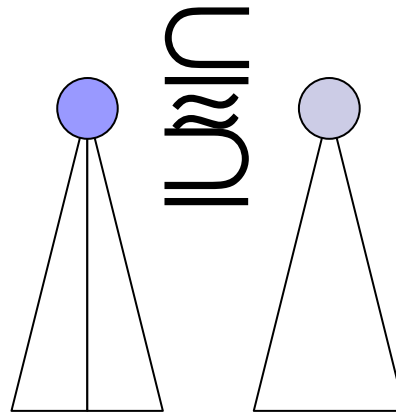


Simulation

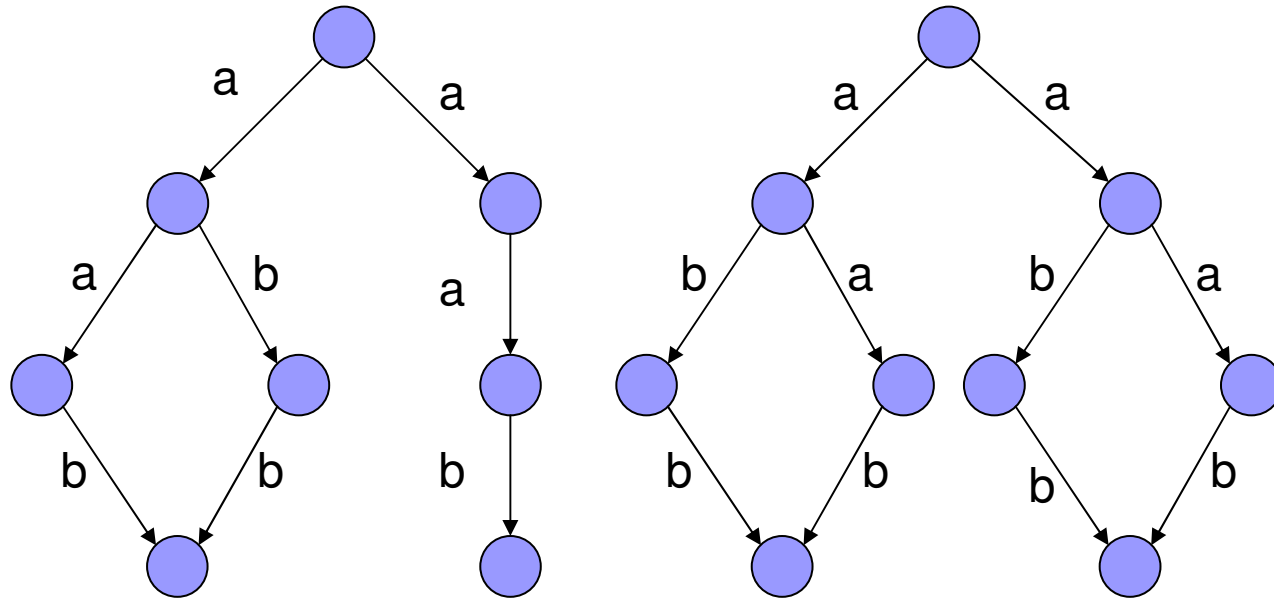




Simulation Equivalence



Bisimulation vs. Simulation Equivalence

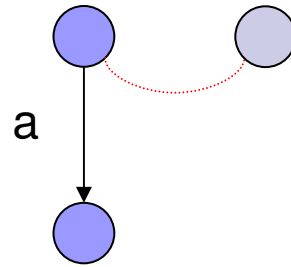




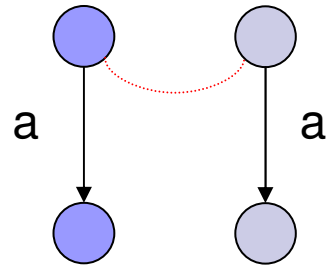
Ready Simulation



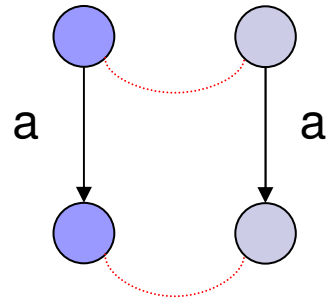
Ready Simulation



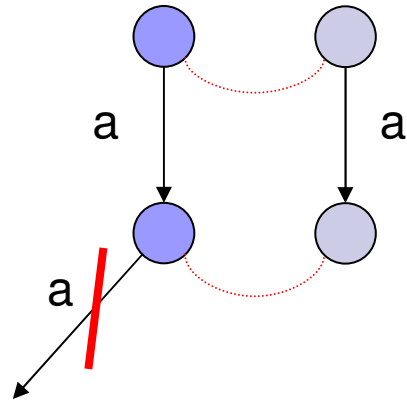
Ready Simulation



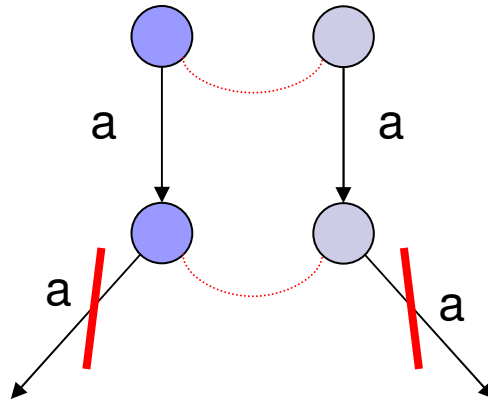
Ready Simulation



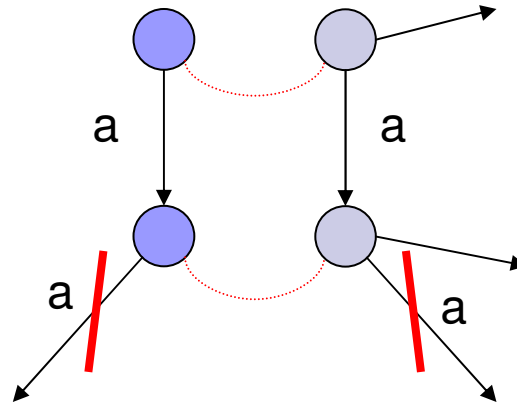
Ready Simulation



Ready Simulation



Ready Simulation





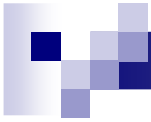
Key Contribution

- Ready simulation implies trace congruence...
- ...and trace congruence means
exchangability

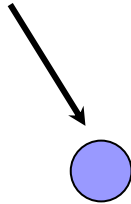


Testing

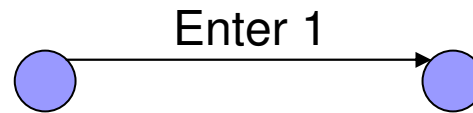
- Testing a process using another process
- Not possible with bisimulation
 - hence the title...
- A brief example...



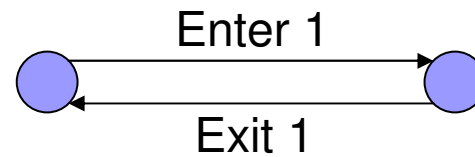
Testing MUTEX using LTS



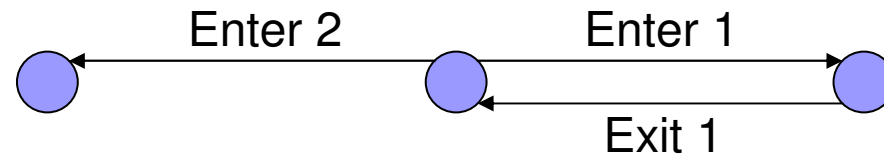
Testing MUTEX using LTS



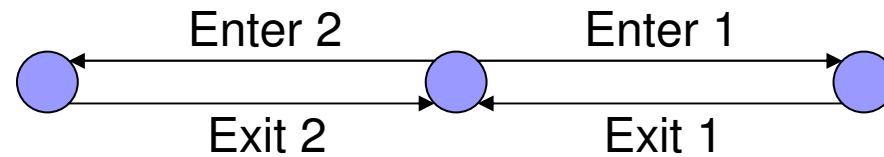
Testing MUTEX using LTS



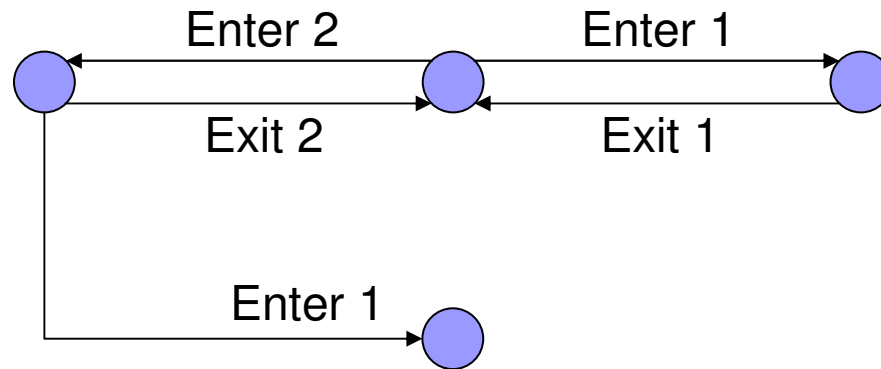
Testing MUTEX using LTS



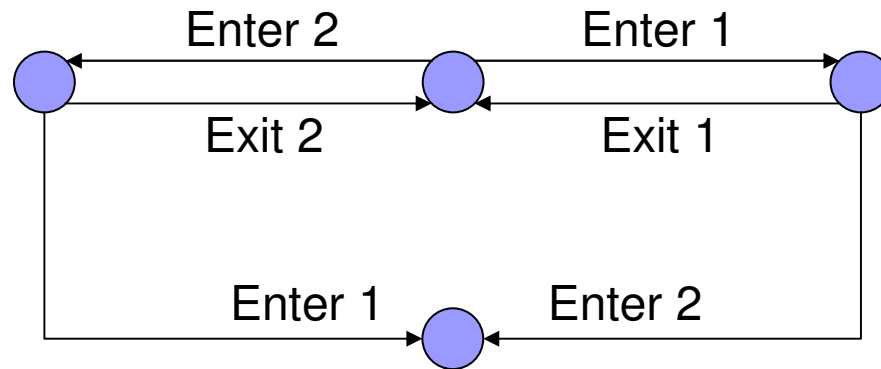
Testing MUTEX using LTS



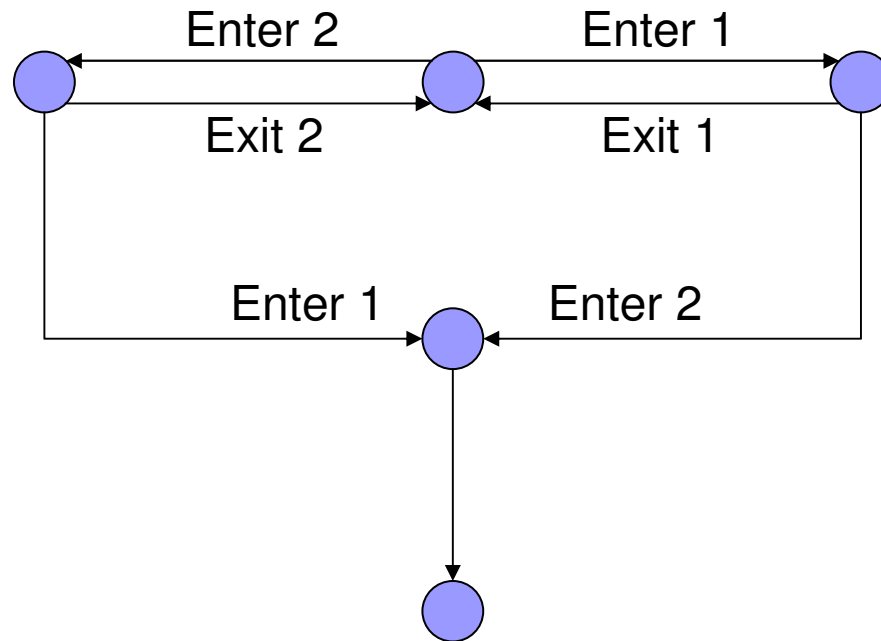
Testing MUTEX using LTS



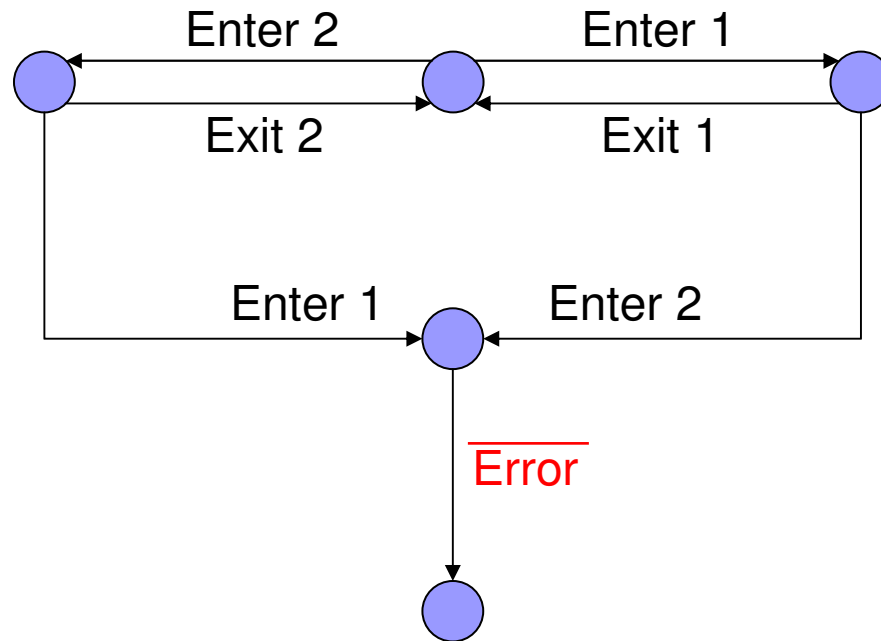
Testing MUTEX using LTS



Testing MUTEX using LTS



Testing MUTEX using LTS





Bisimulation Can't Be Traced

- No testing for bisimilarity using traces in CCS
- Game characterization
- Distinguishing bisimilarity is a meta-process



Rules & Formats

- General structured operational semantics
- GSOS Rules

premises in some format

conclusions

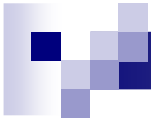


Rules & Formats

- Only requires + and .

$$\frac{}{a.X \xrightarrow{a} X}$$

$$\frac{X \xrightarrow{a} Y}{X'+X \xrightarrow{a} X'+Y}$$



Rules & Formats

- No single specific language for process description
- Sparked research on formats



Key Contributions

- GSOS Rules
- Formats
- Ready simulation equivalent to trace congruence for a specific extension
 - Fork (copy) operator
 - Controlled communication



Summary

- Trace congruence \leftrightarrow Exchangability



Summary

- Trace congruence \leftrightarrow Exchangability
- Ready simulation \rightarrow Trace congruence



Summary

- Trace congruence \leftrightarrow Exchangability
- Ready simulation \rightarrow Trace congruence
- Formats



Summary

- Trace congruence \leftrightarrow Exchangability
- Ready simulation \rightarrow Trace congruence
- Formats
- GSOS



Bisimulation Can't be Traced

Thank you!