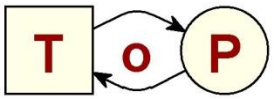
SUMMERSOC 2014
Wed July 3rd 10:30 - 12
Wed July 3rd 15 – 16.30

Tutorial
Formal Methods for SOC
3. Simulation and equivalence

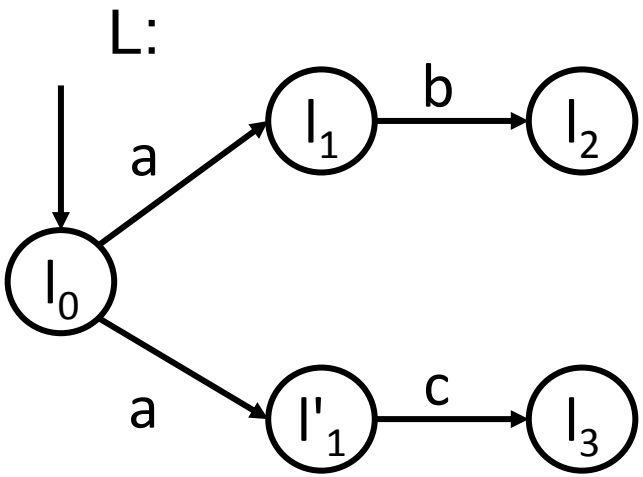
Wolfgang Reisig

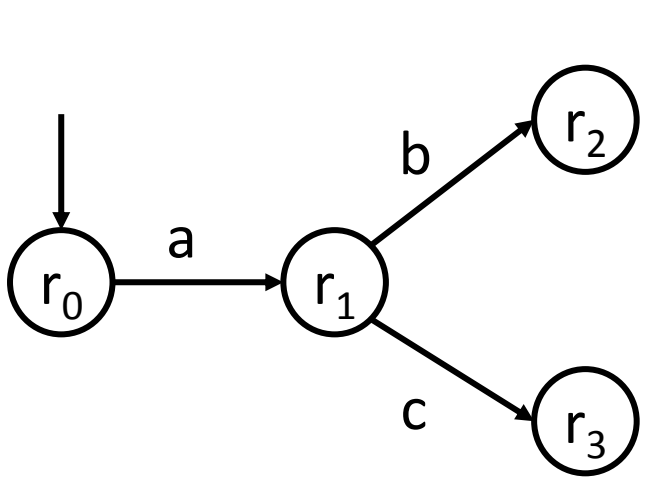


Theory of Programming

Prof. Dr. W. Reisig

.1 Process Simulation





Process L has two traces: a.b, a.c

„Systems with same traces are equivalent!“

Process R has same traces.

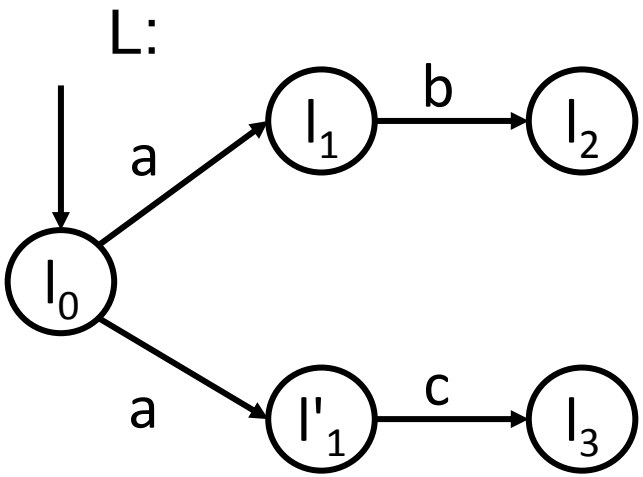
L and R are *not* equivalent, … by no means!

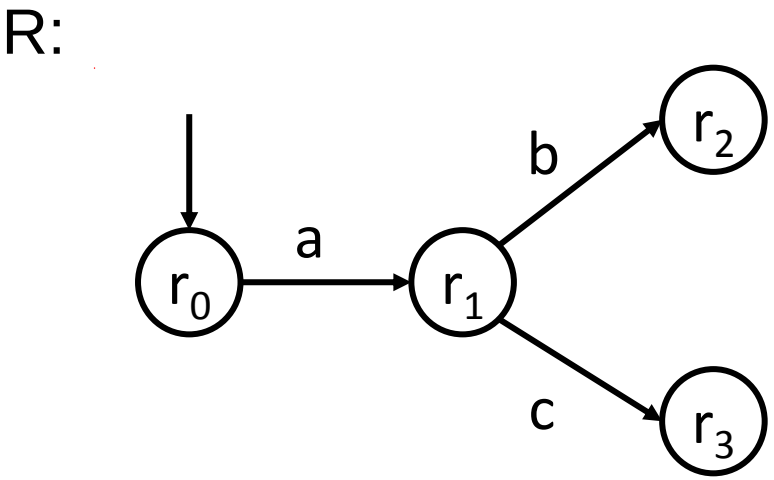
R is „more liberal“ than L.

intuitively : R *simulates* L

L does *not* simulate R

Simulation: a relation



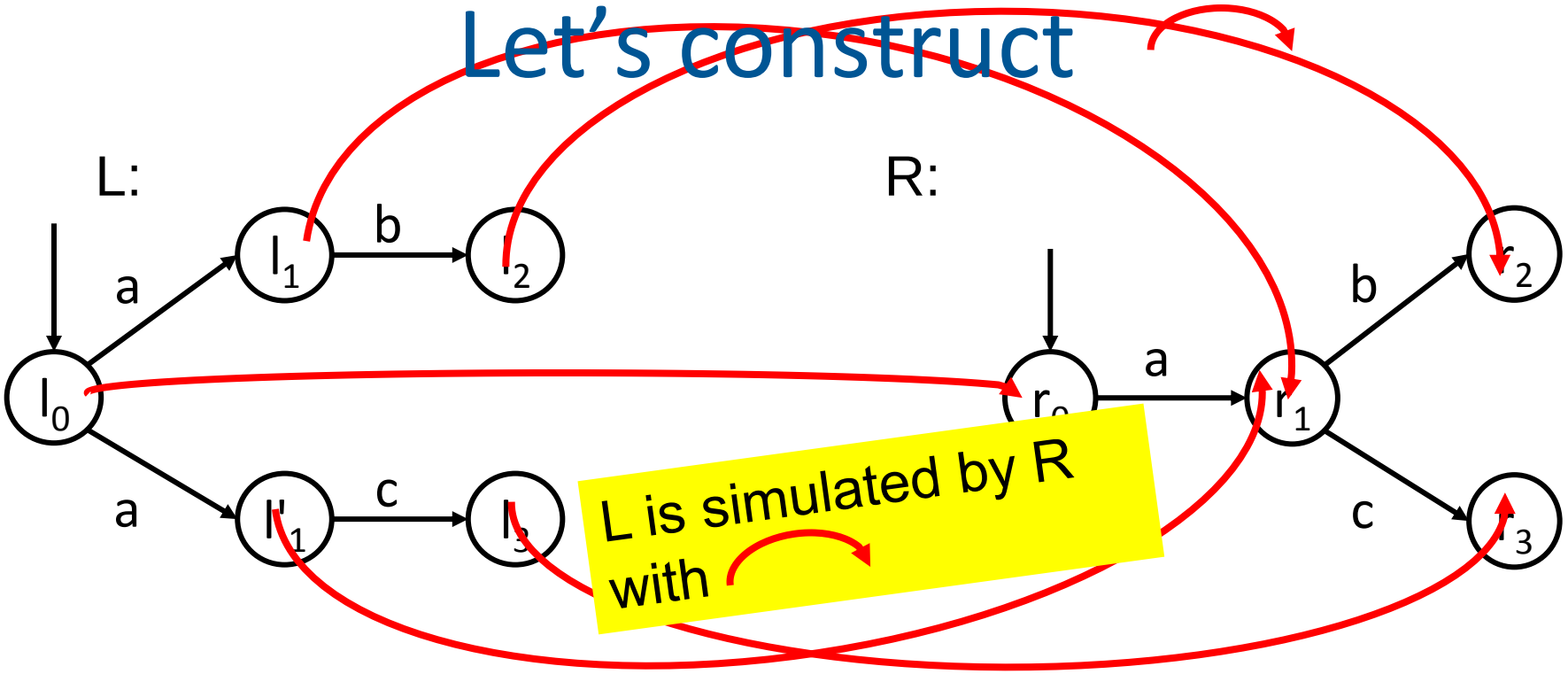


Def. L is simulated by R with ⌒ iff

a) l_0 ⌒ r_0 „ l_0 is simulated by r_0 “

b) If $l \xrightarrow{\alpha} l'$ then there exists $r \xrightarrow{\alpha} r'$ and l' is simulated by r'

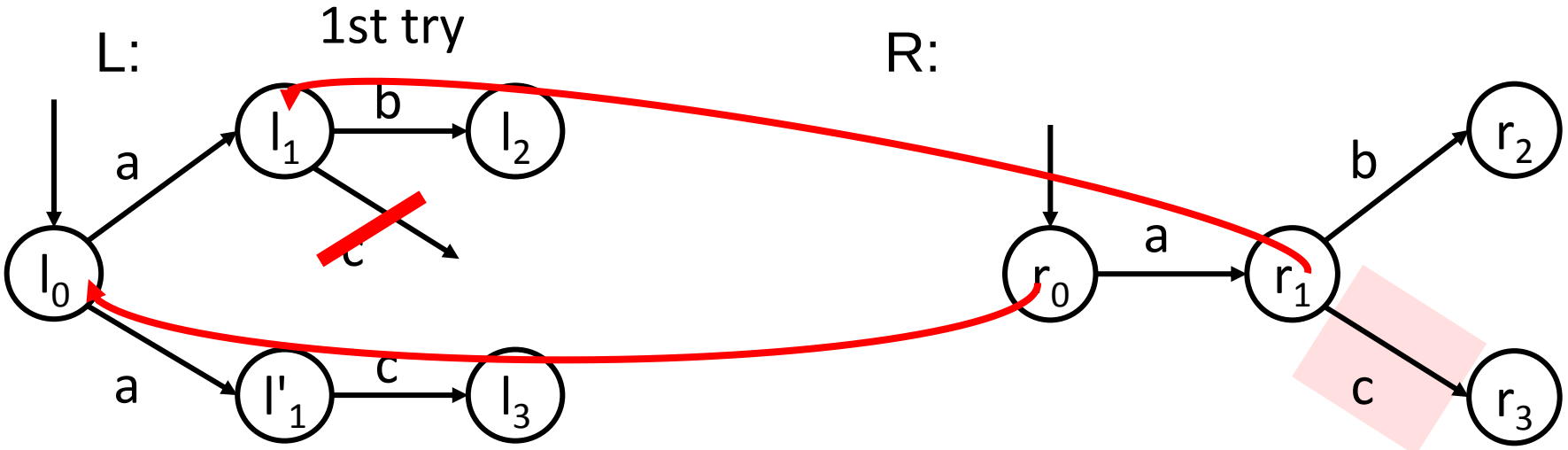
Let's construct



a) l_0 r_0 „ l_0 is simulated by r_0 "

b) If $l \xrightarrow{\alpha} l'$ then there exists $r \xrightarrow{\alpha} r'$ and l' is simulated by r'

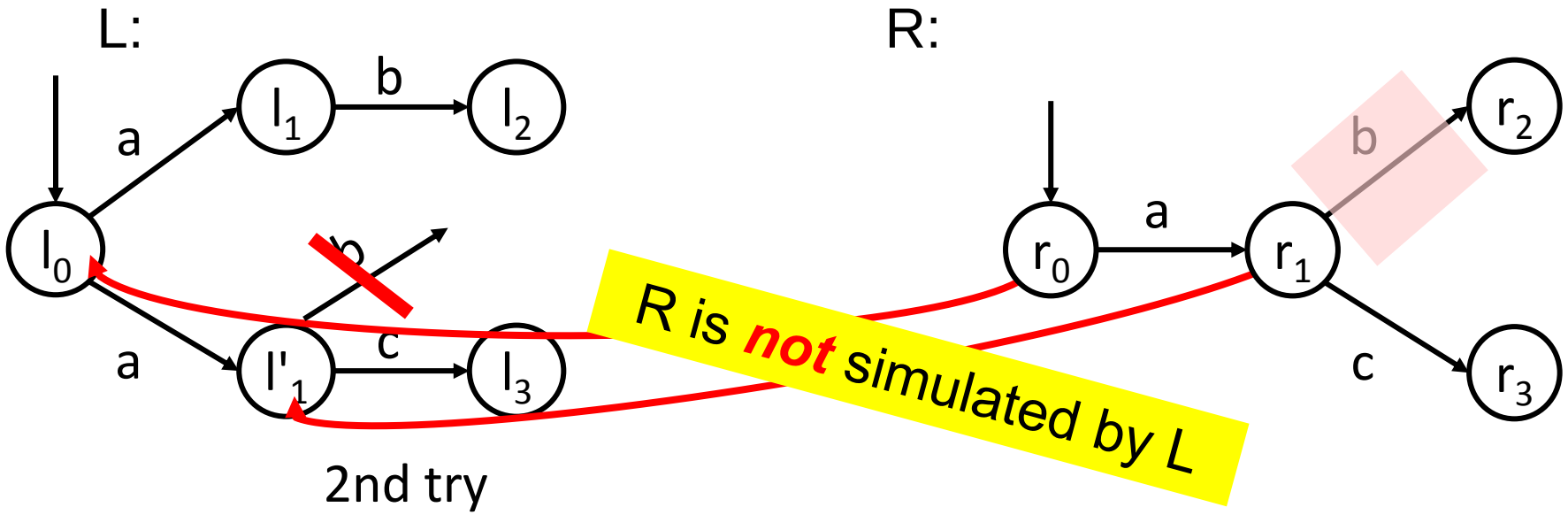
Vice versa



a) l_0 r_0 „ l_0 is simulated by r_0 “

b) If $l \xrightarrow{\alpha} l'$ then there exists $r \xrightarrow{\alpha} r'$ and l' is simulated by r'

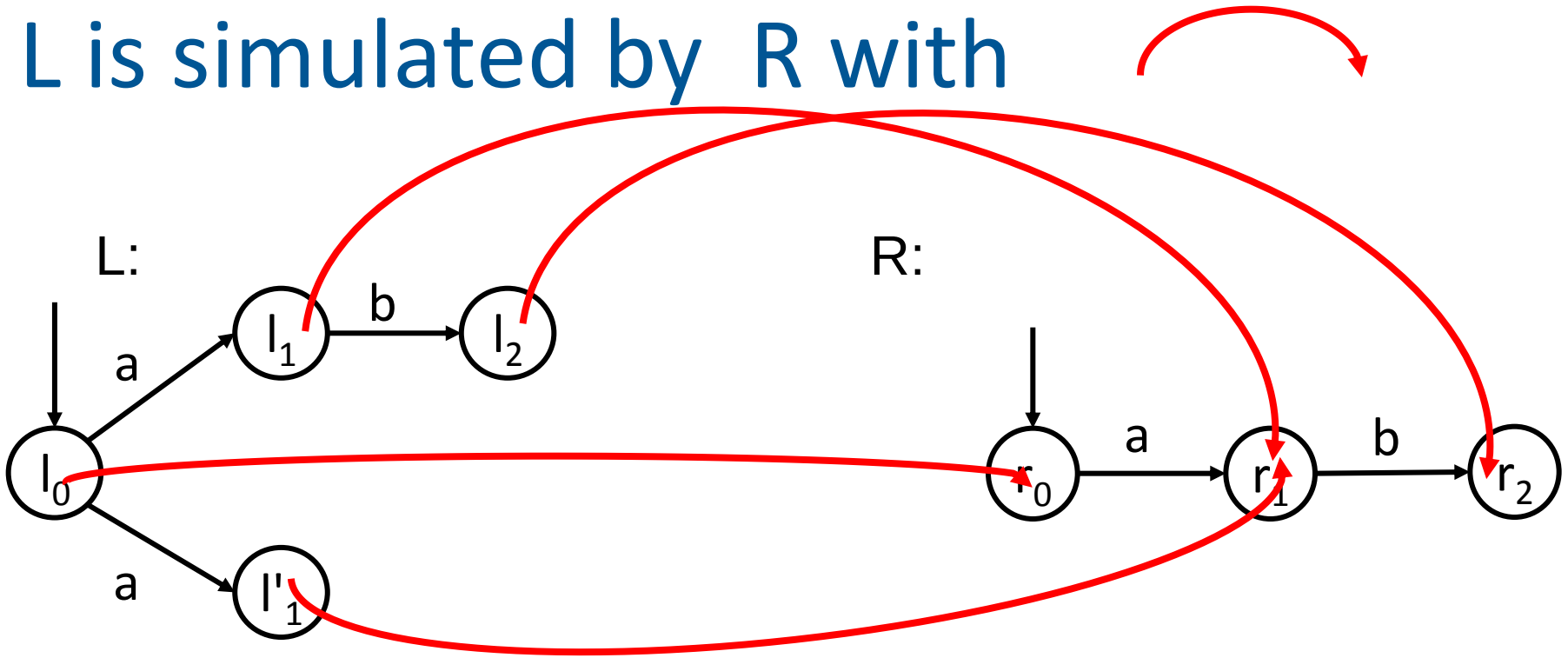
Vice versa



a) l_0 r_0 „ l_0 is simulated by r_0 “

b) If $l \xrightarrow{\alpha} l'$ then there exists $r \xrightarrow{\alpha} r'$ and l' is simulated by r'

L is simulated by R with

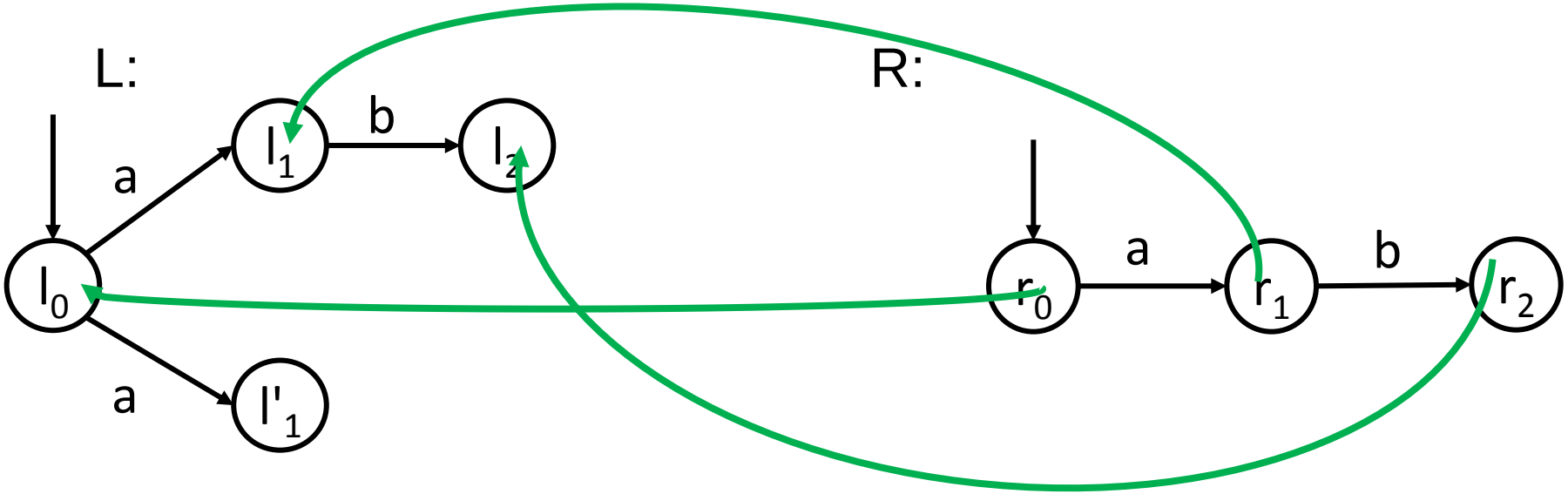


a) l_0 r_0 „ l_0 is simulated by r_0 “

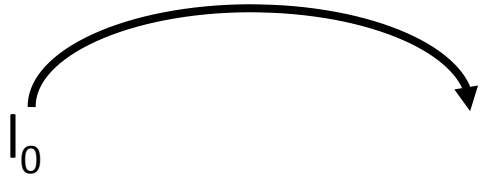
b) If $l \xrightarrow{\alpha} l'$ then there exists $r \xrightarrow{\alpha} r'$ and l' is simulated by r'

7

R is simulated by L with

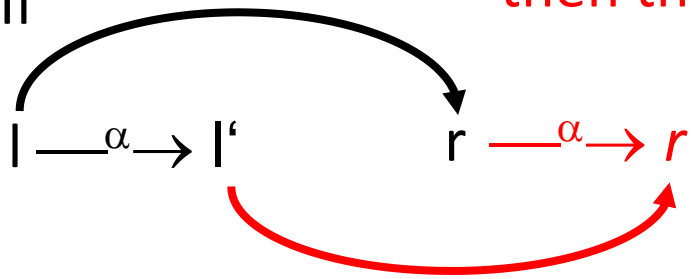


a)



„ l_0 is simulated by r_0 “

b) If



then there exists

and l' is simulated by r'

Simulation equivalence

Def.: P and Q are *simulation equivalent*
iff P simulates Q and Q simulates P.

Observation. Simulation equivalence is
an equivalence relation on processes.

Def.: Let ~ be an equivalence relation,
and let + be any reasonable composition of processes.
Then ~ is a *congruence* (w.r.t. +) iff
for all processes P, Q, R holds:
If P ~ Q then P+R ~ Q+R.

Observation. Simulation equivalence is no congruence! ☹

2. Bisimulation

Observation:

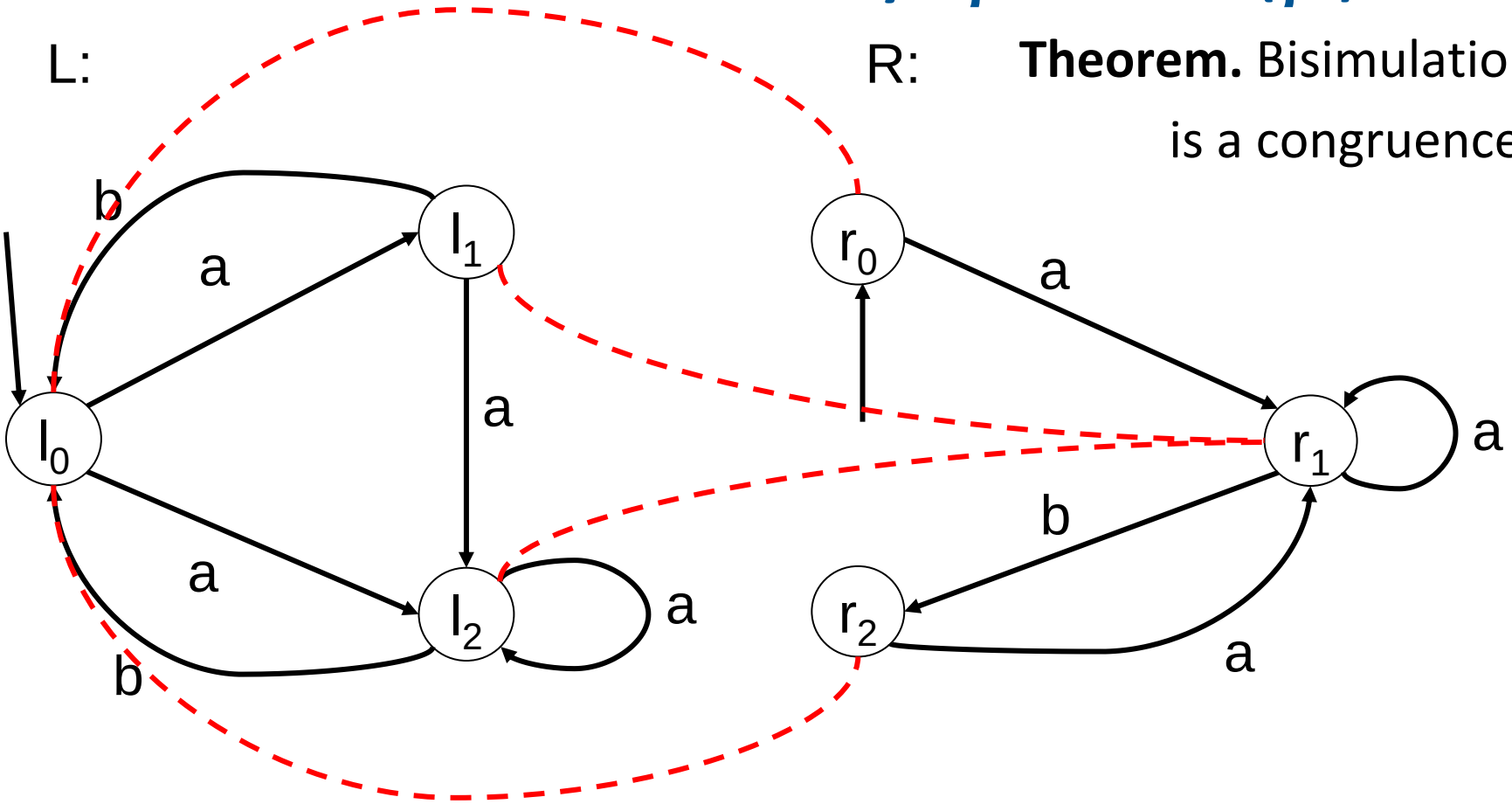
A slightly more tight relation

makes simulation equivalence a congruence:

R is simulated by L with ρ and

L is simulated by R with $(\rho)^{-1}$.

mutual simulation by ρ and $(\rho)^{-1}$



Theorem. Bisimulation is a congruence.

Def. ρ is a *bisimulation from L to R* iff R is simulated by L with ρ and L is simulated by R with $(\rho)^{-1}$.

Def. L and R are *bisimular* iff there exists a bisimulation from L to R.

sim $=_{def}$ $\{(l_0,r_0), (l_0,r_2), (l_1,r_1), (l_2, r_1)\}$ is a bisimulation from L to R

Bisimulation harmonizes with CTL*

Theorem.

Two states are bisimilar

iff they share the same *CTL** properties.

Consequence:

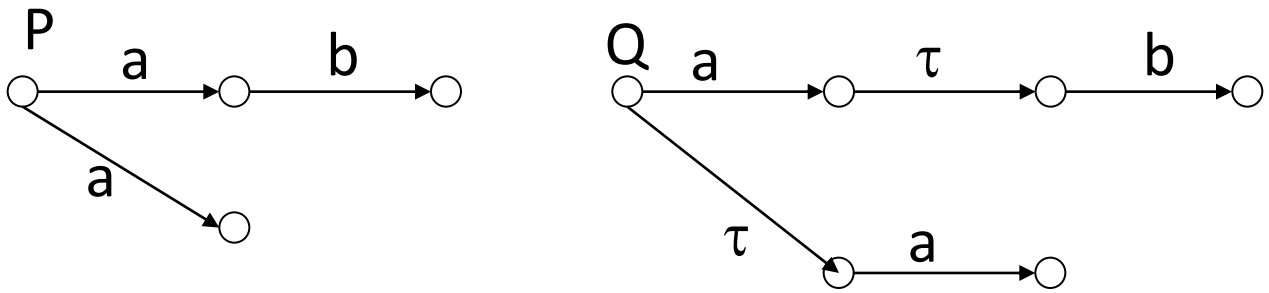
Specify a system in terms of CTL*.

This may yield various different implementations.

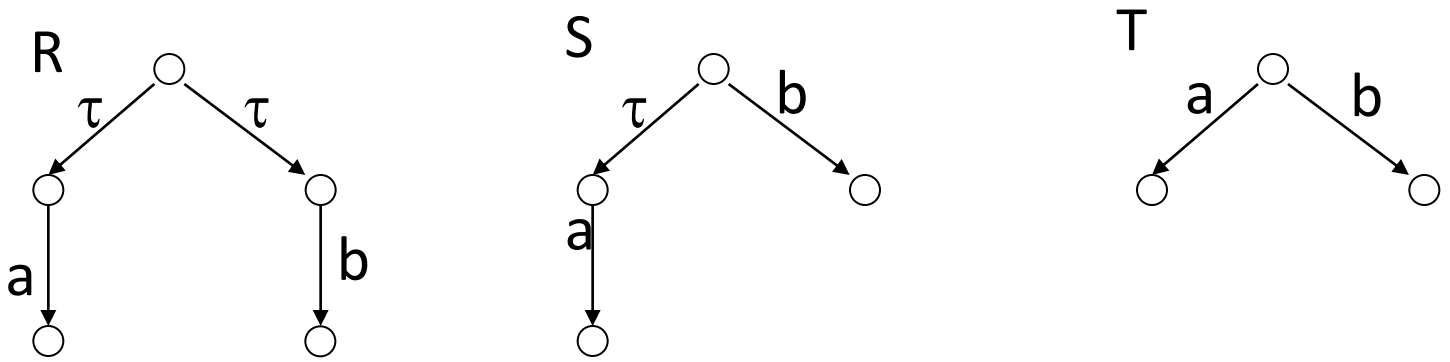
They all are bisimular.

Examples for Bisimularity

P and Q are bisimilar:

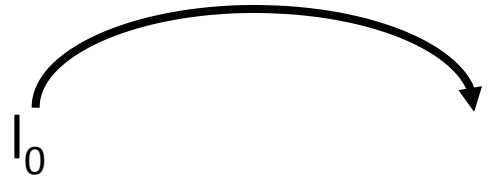


R, S and T are pairwise *not* bisimilar:



Variant: L is *wekly simulated* by R

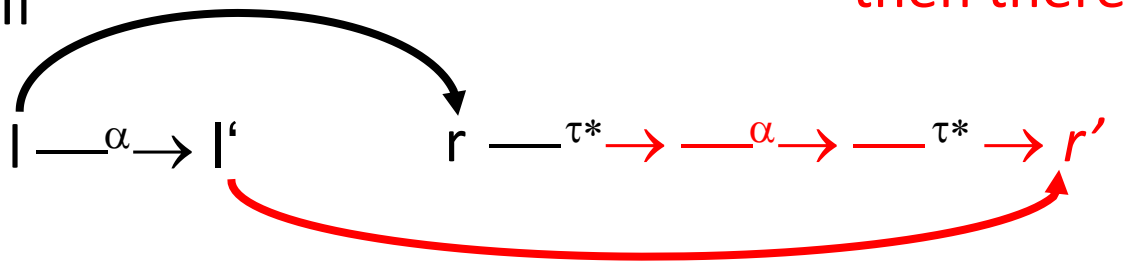
a)



„ l_0 is simulated by r_0 “

b) If

then there exists



Caution!

Weak bisimulation is no congruence

4. Further Congruences

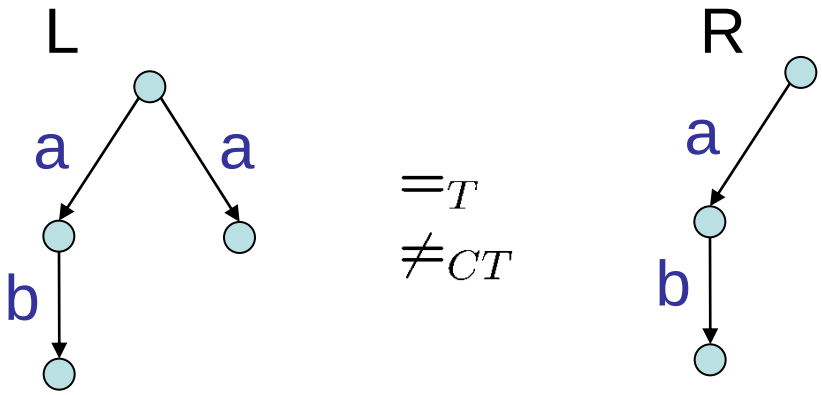
K ! U ! B ! RT ! FT ! F ! CT ! T

less ⟵ identification ⟶ more

We conside them from right to left

Complete Trace Equivalence

Combining termination and choice …



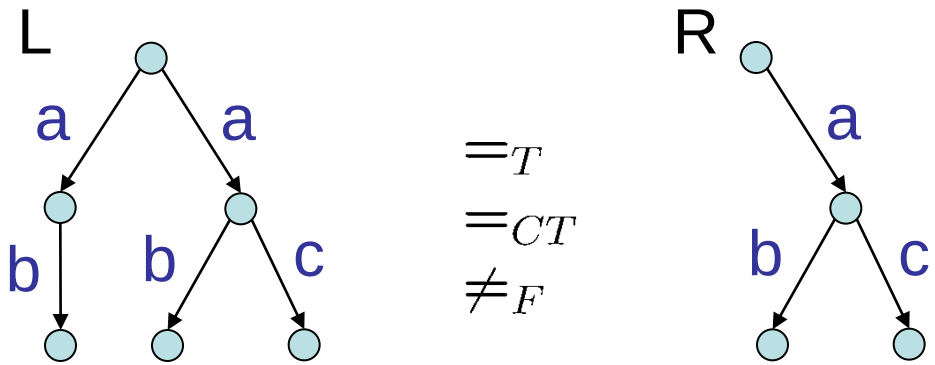
a is a complete trace of L but not of R

K ! U ! B ! RT ! FT ! F ! CT ! T

Failure Equivalence of a set M of actions

Def.: For an action w and a set of actions M:
[w,M] is a *failure pair* of P iff P may do a step

P –w $\rightarrow$ Q and no action of M is enabled in Q.



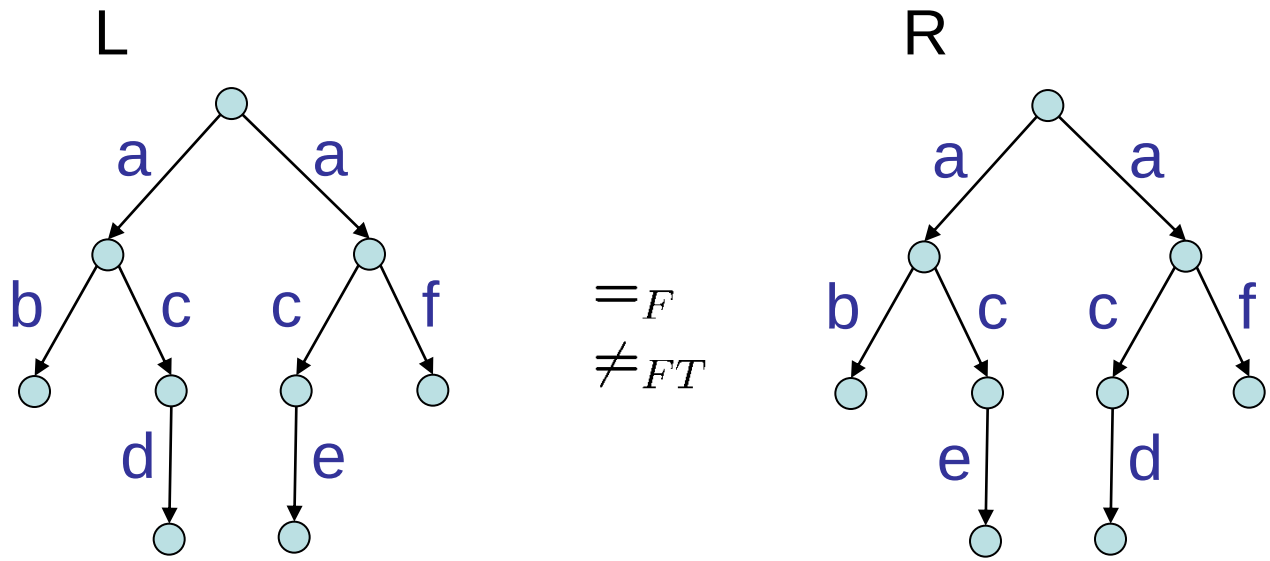
[a,{c}] is a failure pair of L but not of R

K ! U ! B ! RT ! FT ! F ! CT ! T

Failure Trace Equivalence

… like Failure equivalence.

But now you continue along a trace

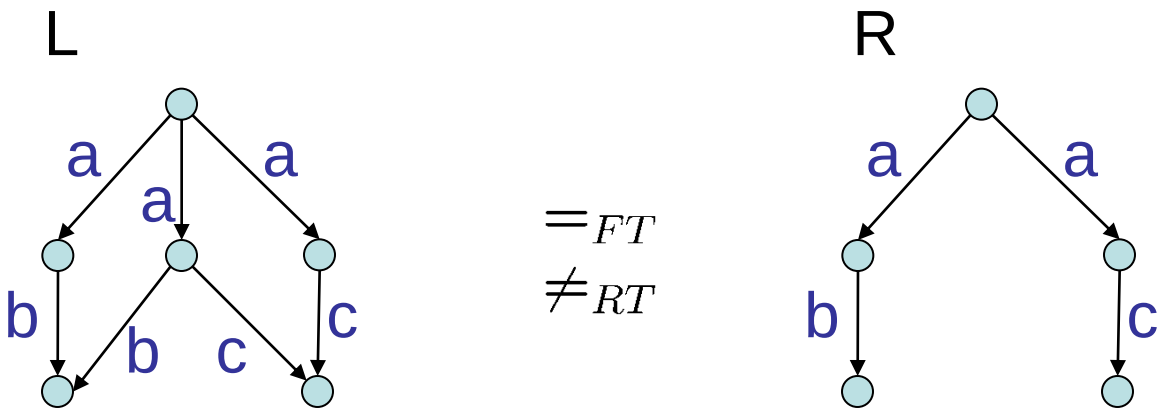


a {f} c {e} d is a failure trace of L but not of R

K ! U ! B ! RT ! FT ! F ! CT ! T

Ready Trace Equivalence

In a trace, between each two actions,
present the alternative actions.



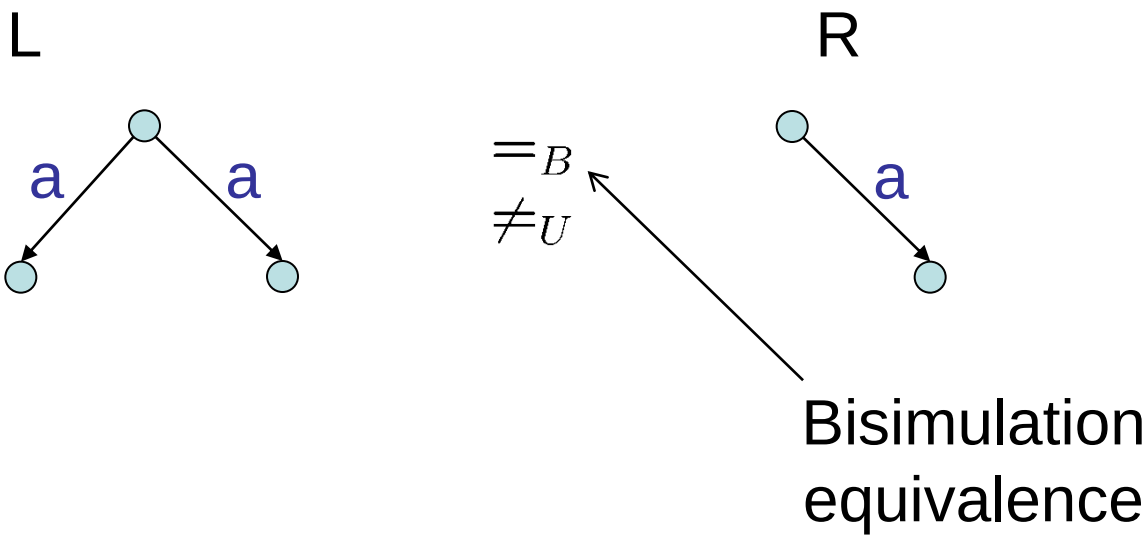
[a,{c},b] is a ready trace of L but not of R

K ! U ! B ! RT ! FT ! F ! CT ! T

Tree Equivalence

Unfold the transition systems as trees

L $=_U$ R iff both trees are isomorpic

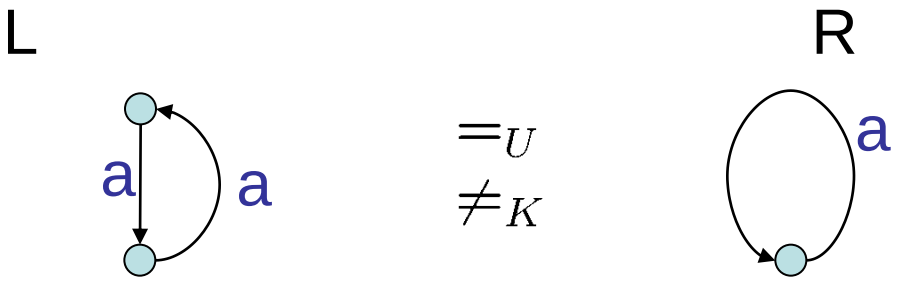


K ! U ! B ! RT ! FT ! F ! CT ! T

Structural Equivalence

Equivalence:

L $=_K$ R iff the transtion systems are isomorphic



K ! U ! B ! RT ! FT ! F ! CT ! T

Further equivalences

Ready equivalence

Ready Simulation equivalence

Ready Trace Simulation equivalence

Completed Simulation equivalence

Failure Simulation equivalence

Failure Trace Simulation equivalence

Simulation equivalence

…

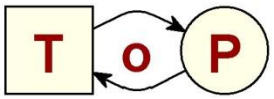
152 ones

K ! U ! B ! RT ! FT ! F ! CT ! T

SUMMERSOC 2014
Wed July 3rd 10:30 - 12
Wed July 3rd 15 – 16.30

Tutorial
Formal Methods for SOC
3. Simulation and equivalence

Wolfgang Reisig



Theory of Programming

Prof. Dr. W. Reisig

the end