Bisimulation and Simulation

Bisimilarity

Intuition:

Two (states of two) transition systems are bisimilar if they have the same behavior.

In the sense that:

- *Locally they (the two **states**) look indistinguishable*
- *Every **action** that can be done on one of them can also be done on the other remaining indistinguishable*

Bisimilarity

- A binary relation R is a **bisimulation** iff:

  $(s,t) \in R$ implies that
  - s is *final* iff t is *final*
  - for all actions a
    - if $s \rightarrow_a s'$ then $\exists t' . t \rightarrow_a t'$ and $(s',t') \in R$
    - if $t \rightarrow_a t'$ then $\exists s' . s \rightarrow_a s'$ and $(s',t') \in R$

- A state s_0 of transition system S is **bisimilar**, or simply **equivalent,** to a state t_0 of transition system T iff there **exists** a **bisimulation** between the initial states s_0 and t_0.

- Notably
  - **bisimilarity** is a bisimulation
  - **bisimilarity** is the largest bisimulation

Note it is a co-inductive definition!

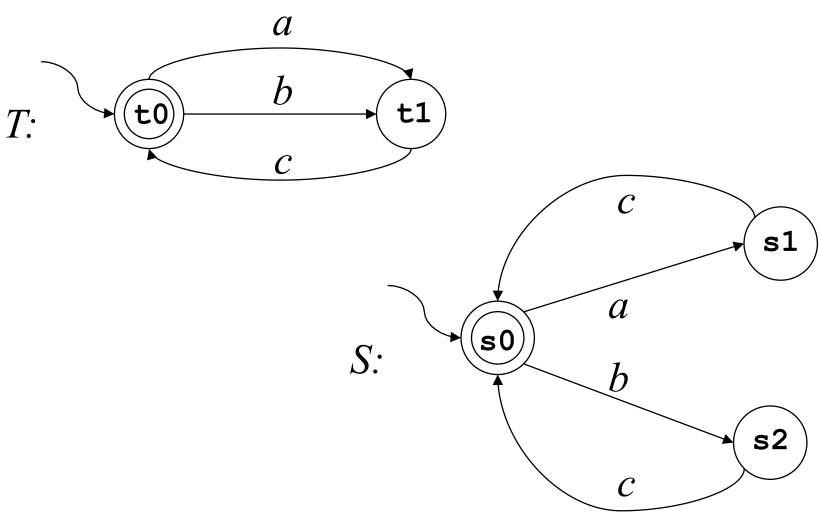
Computing Bisimilarity on Finite Transition Systems

Algorithm ComputingBisimulation
Input: transition system $TS_S = < A, S, S^0, \delta_S, F_S>$ and
transition system $TS_T = < A, T, T^0, \delta_T, F_T>$
Output: the **bisimilarity** relation (the largest bisimulation)

Body
$R = S \times T$
$R' = R - \{(s,t) \mid \neg(s \in F_S \equiv t \in F_T)\}$
while $(R \neq R')$ {
$R := R'$
$R' := R' - (\{(s,t) \mid \exists s',a.\ s \rightarrow_a s' \wedge \neg\exists t' . t \rightarrow_a t' \wedge (s',t') \in R' \}$
$\{(s,t) \mid \exists t',a.\ t \rightarrow_a t' \wedge \neg\exists s' . s \rightarrow_a s' \wedge (s',t') \in R' \})$
}
return R'
Ydob

Example of Bisimilarity

Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
SAPIENZA Università di Roma



Are S and T **bisimilar?**

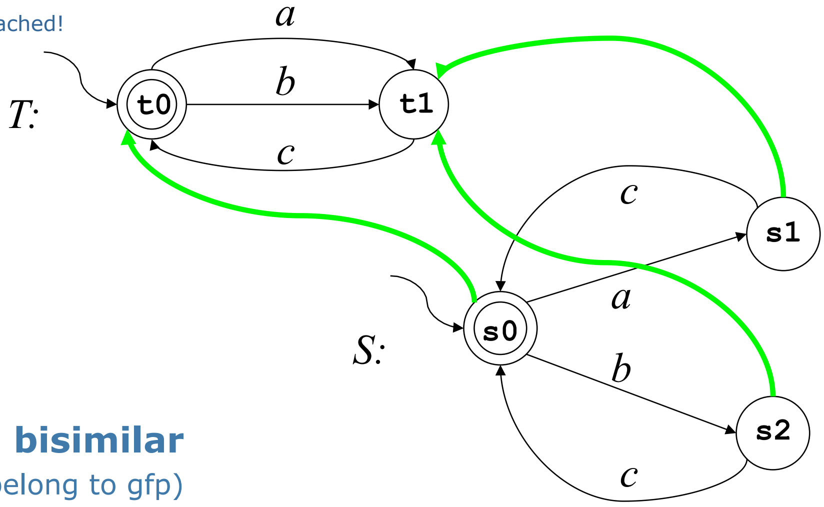
Computing Bisimilarity

Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
SAPIENZA Università di Roma

We need to compute the greatest fixpoint (gfp): we do it by computing approximates starting from the Cartesian product:

- R0={(t0,s0), (t0,s1), (t0,s2), (t1,s0), (t1,s1),(t1,s2)} – Cartesian product
- R1={(t0,s0),(t1,s1),(t1,s2)} – removed those pairs that violate local condition on final (final iff final)
- R2={(t0,s0),(t1,s1),(t1,s2)} – removed those pairs where one can do action and other cannot copy remaining in the relation.

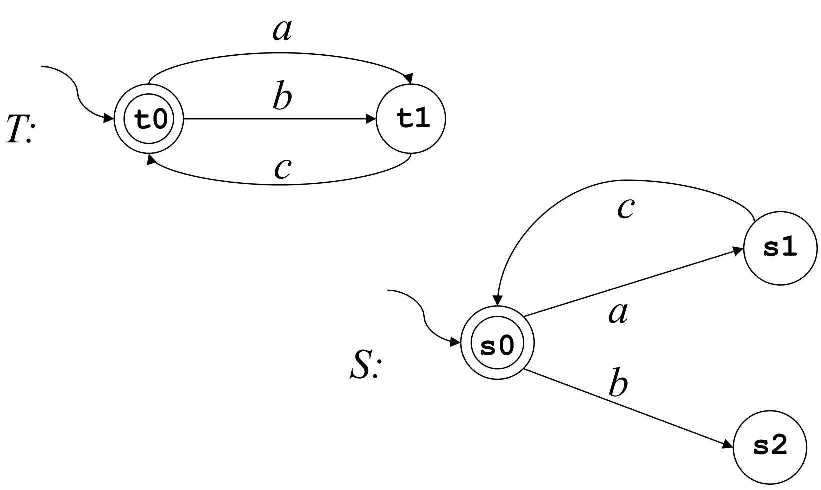
R1=R2 greatest fixpoint reached!



S and T are **bisimilar**

((t0,s0) do belong to gfp)

Example of NON Bisimilarity



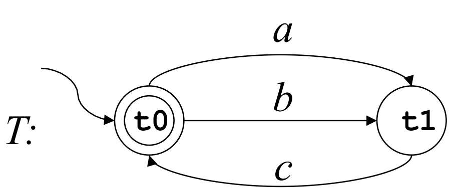
Are S and T **bisimilar?**

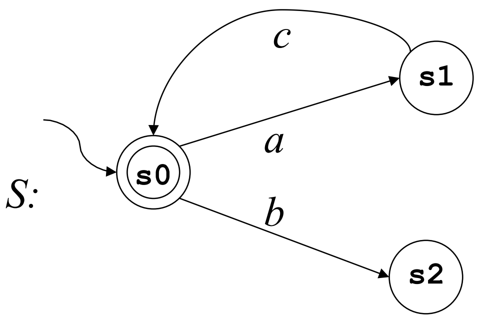
Computing Bisimilarity

We need to compute the greatest fixpoint: we do it by computing approximates starting from the cartesian product:

- R0={(t0,s0), (t0,s1), (t0,s2), (t1,s0), (t1,s1),(t1,s2)} – cartesian product
- R1={(t0,s0),(t1,s1),(t1,s2)} – removed those pairs that violate local condition on final (final iff final)
- R2={(t0,s0),(t1,s1)} – removed (t1,s2) since t1 can do c but s2 cannot.
- R3={(t1,s1)} – removed (t0,s0) since t0 can do b, s2 can do b as well, but then the resulting states (t1,s2) are NOT in R2.
- R4 = {} – removed (t1,s1) since t1 can do c, s1 can do c as well, but then the resulting states (t0,s0) are NOT in R3.
- R5 = {}

R4=R5 greatest fixpoint reached!





S and T are NOT **bisimilar**

((t0,s0) do not belong to gfp)

Simulation

Intuition:

*One (state of a) transition system can be **mimicked** (or **copied**) by another (state of another) transition system.*

In the sense that:

- *Locally the property that hold on the **state** of the "to be copied" transitions systems, holds also in the state of the "coping" transition*
- *Every **action** that the "to be copied" transition system can do in the current state, can be copied by the "coping" transition system (in the current state) and the same thing holds in the resulting states.*

Simulation

- A binary relation R is a **simulation** iff:

  (s,t) $\in R$ implies that
  - s is *final* implies that t is *final*
  - for all actions a
    - if $s \rightarrow_a s'$ then $\exists t' . t \rightarrow_a t'$ and $(s',t') \in R$

- A state s_0 of transition system S is **simulated by** a state t_0 of transition system T iff there **exists** a **simulation** between the initial states s_0 and t_0.

- Notably
  - **simulated-by** is a simulation
  - **simulated-by** is the largest simulation

Note it is a co-inductive definition!

- NB: A simulation is just one of the two directions of a bisimulation

Computing Simulation on Finite Transition Systems

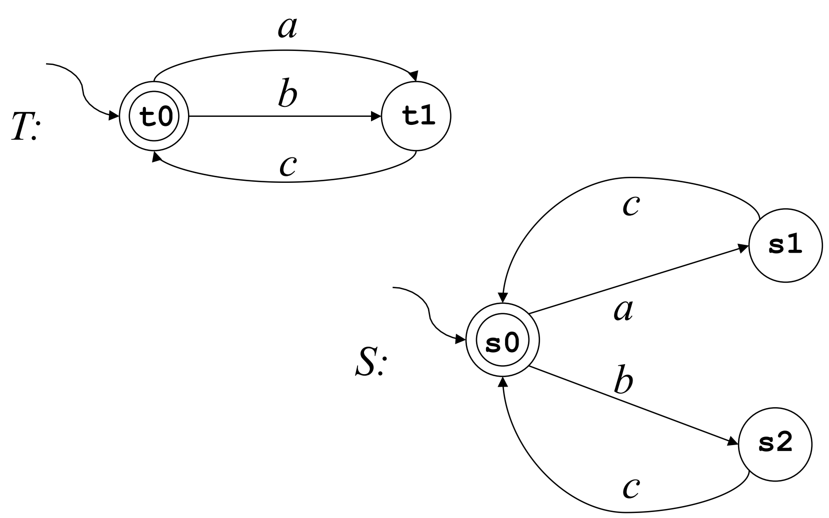
Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
SAPIENZA Università di Roma

Algorithm ComputingSimulation
Input: transition system $TS_S = < A, S, S^0, \delta_S, F_S>$ and
transition system $TS_T = < A, T, T^0, \delta_T, F_T>$
Output: the **simulated-by** relation (the largest simulation)

Body
$R = S \times T$
$R' = S \times T - \{(s,t) \mid s \in F_S \wedge \neg(t \in F_T)\}$
while $(R \neq R')$ {
$R := R'$
$R' := R' - \{(s,t) \mid \exists s',a.\ s \to_a s' \wedge \neg\exists t'.\ t \to_a t' \wedge (s',t') \in R'\}$
}
return R'
Ydob

Example of Simulation

Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
SAPIENZA Università di Roma

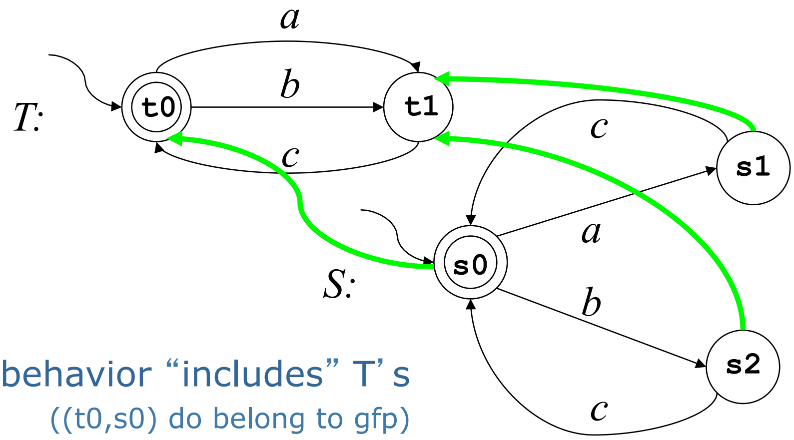


Are S and T **similar?**

Computing Simulation

We need to compute the greatest fixpoint (gfp): we do it by computing approximates starting from the Cartesian product:

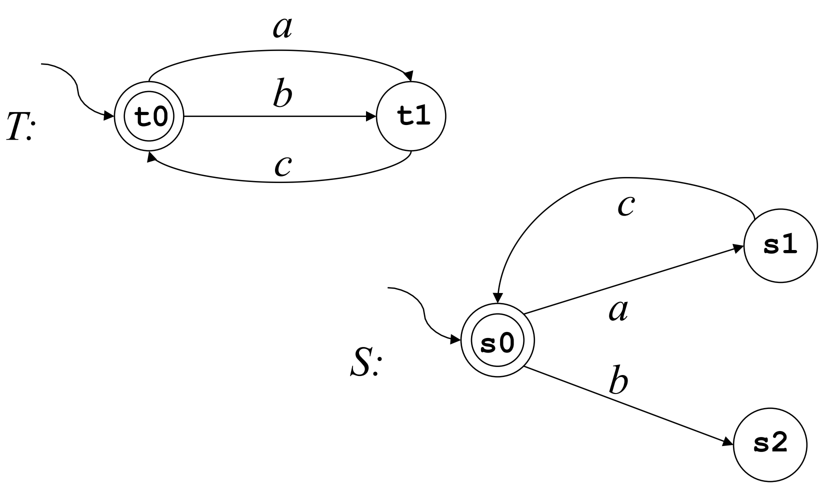
- R0={(t0,s0), (t0,s1), (t0,s2), (t1,s0), (t1,s1),(t1,s2)} – Cartesian product
- R1={(t0,s0),(t1,s0),(t1,s1),(t1,s2)} – removed those pairs that violate local condition on final (if T final then S final)
- R2={(t0,s0),(t1,s1),(t1,s2)} – removed (t1,s0), since t1 can do c and s0 cannot
- R3={(t0,s0),(t1,s1),(t1,s2)} – nothing is further removed
- R2=R3 greatest fixpoint reached!



S **simulates** T: S' s behavior "includes" T' s

((t0,s0) do belong to gfp)

Example of Simulation



Does S **simulate** T?

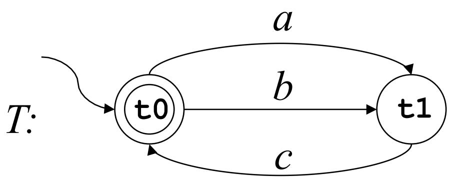
Computing Simulation

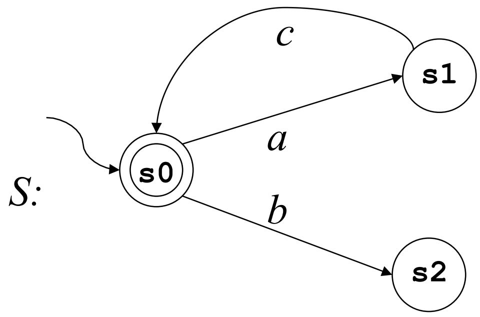
Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
SAPIENZA UNIVERSITÀ DI ROMA

We need to compute a greatest fixpoint: we do it by computing approximates starting from the Cartesian product:

- R0={(t0,s0), (t0,s1), (t0,s2), (t1,s0), (t1,s1),(t1,s2)} – Cartesian product
- R1={(t0,s0), (t1,s0), (t1,s1),(t1,s2)} – removed those pairs that violate local condition on final (if T final then S final)
- R2={(t0,s0),(t1,s1)} – removed (t1,s0) since t1 can do c but s0 cannot; removed (t1,s2) since t1 can do c but s2 cannot
- R3={(t1,s1)} – removed (t0,s0) since t0 can do b, s2 can do b as well, but then the resulting states (t1,s2) are NOT in R2.
- R4 = {} – removed (t1,s1) since t1 can do c, s1 can do c as well, but then the resulting states (t0,s0) are NOT in R3.
- R5 = {}

R4=R5 greatest fixpoint reached!

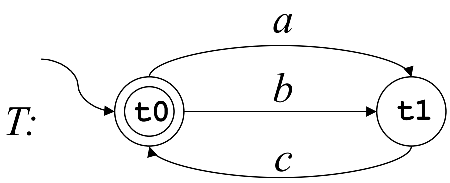


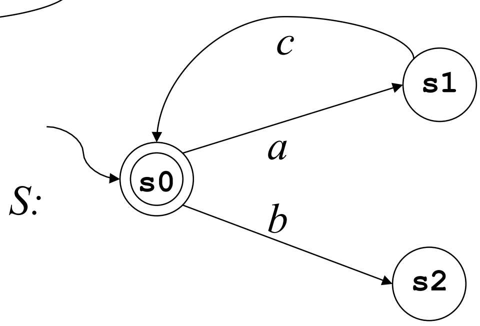


S does **not simulate** T
((t0,s0) do not belong to gfp)

Example of Simulation

Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
SAPIENZA UNIVERSITÀ DI ROMA





Does T **simulate** S?

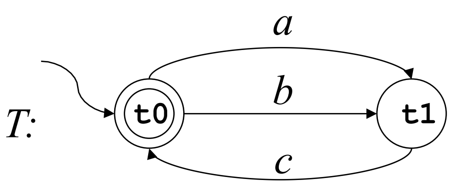
Computing Simulation

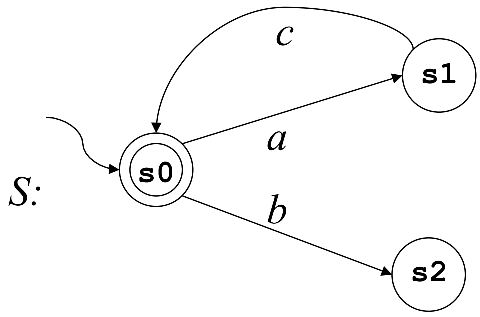
We need to compute the greatest fixpoint: we do it by computing approximates starting from the Cartesian product:

- R0={(s0,t0), (s0,t1), (s1,t0), (s1,t1), (s2,t0),(s2,t1)} – Cartesian product
- R1={(s0,t0), (s1,t0), (s1,t1), (s2,t0),(s2,t1)} – removed those pairs that violate local condition on final (if S final then T final)
- R2={(s0,t0), (s1,t1), (s2,t0), (s2,t1)} – removed (s1,t0), since s1 can do c but t0 cannot; removed (t1,s2) since t1 can do c but s2 cannot
- R3={(s0,t0), (s1,t1), (s2,t0), (s2,t1)} – nothing is removed

R2=R3 greatest fixpoint reached!

-



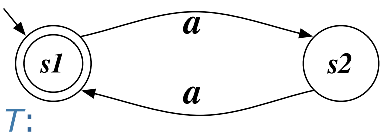


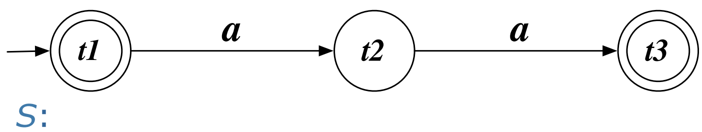
T does **simulate** S
((s0,t0) do belong to gfp)

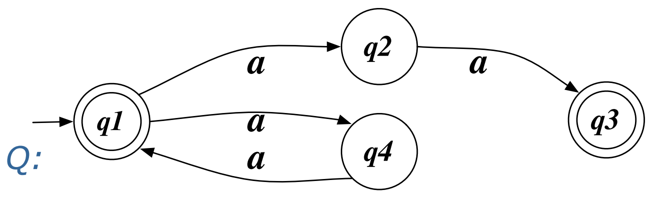
Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
SAPIENZA Università di Roma

Exercises

Consider the following transition systems.





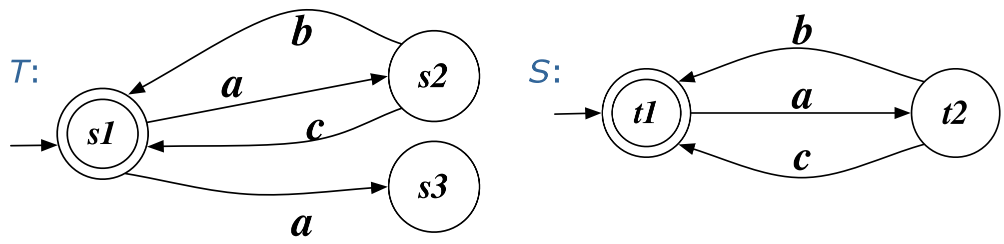


- *Which **simulations** hold between the three?* If simulation holds, write a simulation relation, otherwise show where simulation breaks.
- *Which **bisimulations** hold between the three?* If bisimulation holds, write a bisimulation relation, otherwise show where simulation breaks.

Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
SAPIENZA Università di Roma

Exercises

Consider the following transition systems.

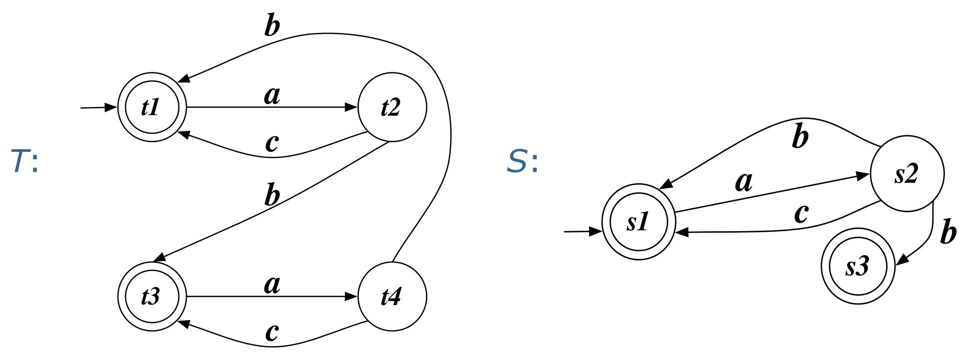


- *Does T* ***simulates*** *S?* If so, write a simulation relation. If not, show where simulation breaks.
- *Does S* ***simulates*** *T?* If so, write a simulation relation. If not, show where simulation breaks.
- *Are they* ***bisimilar****?* If so, write a bisimulation relation. If not, show where bisimulation breaks.

Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
SAPIENZA Università di Roma

Exercises

Consider the following transition systems.



- *Does T* ***simulates*** *S?* If so, write a simulation relation. If not, show where simulation breaks.
- *Does S* ***simulates*** *T?* If so, write a simulation relation. If not, show where simulation breaks.
- *Are they* ***bisimilar****?* If so, write a bisimulation relation. If not, show where bisimulation breaks.