Weak Bisimulation Finiteness of Pushdown Systems With Deterministic ε-Transitions Is 2-EXPTIME-Complete

Stefan Göller
University of Kassel

Paweł Parys
University of Warsaw

RP 2022 (based on SODA 2023 paper)

Pushdown systems

are given by a tuple (Q,Γ,A,R), where

- $Q=\{p,q,r\}$ is a finite set of control states
- $\Gamma=\{X,Y,Z\}$ is a finite set of stack symbols
- $A=\{a,b,c\}$ is a finite set of input symbols and
- R is a finite set of **rewrite rules** of either form:

$\overset{p}{\boxed{X}} \overset{a}{\rightarrowtail} q$ (pop rule) or $\overset{p}{\boxed{X}} \overset{a}{\rightarrowtail} \overset{q}{\begin{array}{|c|}\hline Y \\ \hline Z \\ \hline\end{array}}$ (push rule)

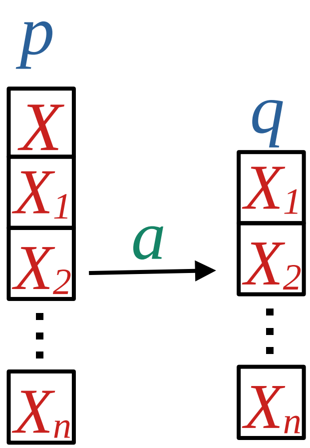
induce an infinite A-edge-labeled transition system…

Induced transition system (infinite)

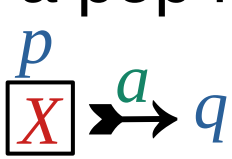
Each pushdown system (Q,Γ,A,R) induces an infinite transition system:

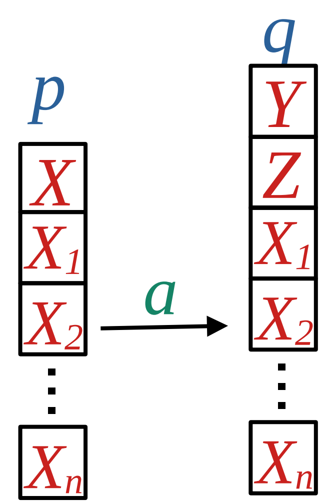
- nodes = state & stack  $\in Q\times\Gamma^*$

- transitions (labeled by A):

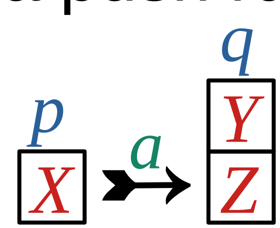


for a pop rule:



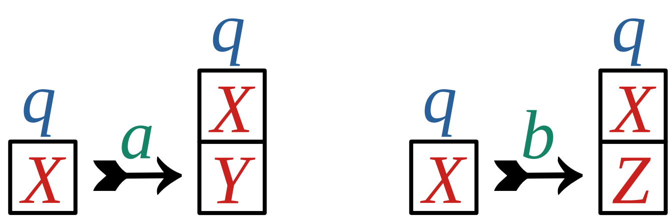


for a push rule:

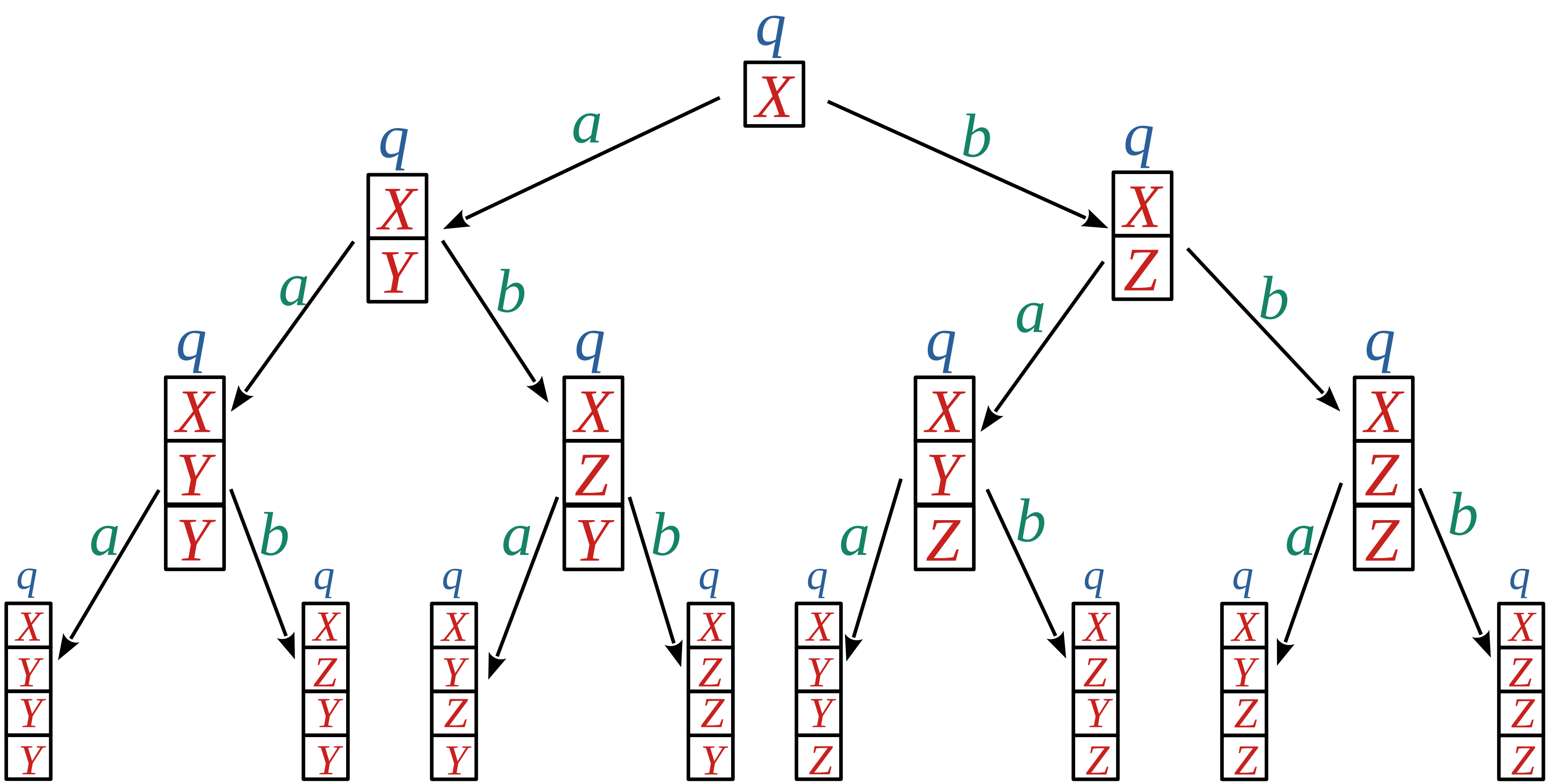


Example pushdown system

The two rules

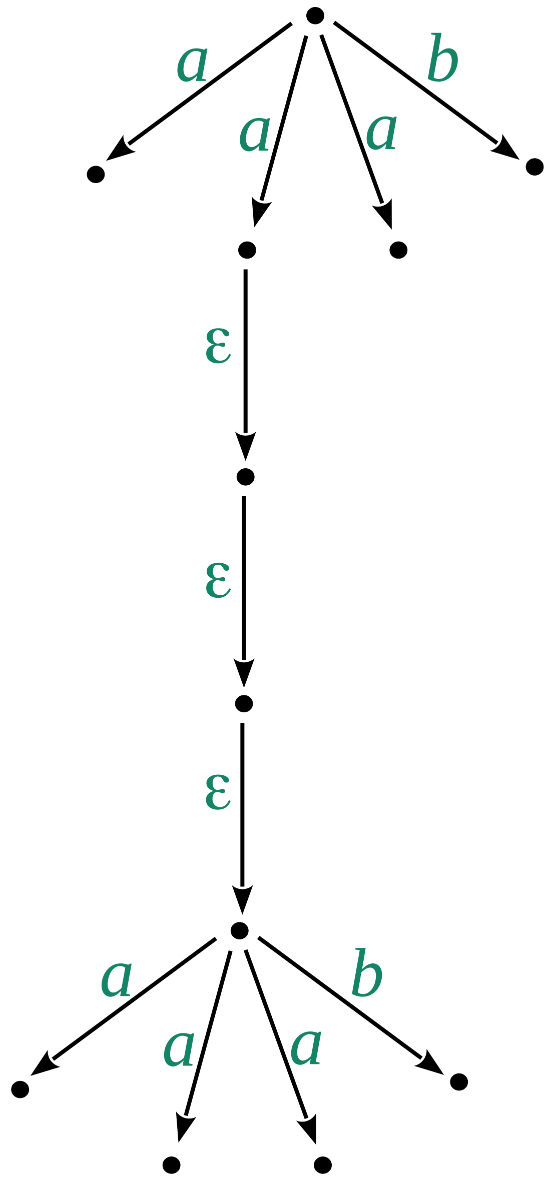


induce the infinite binary tree

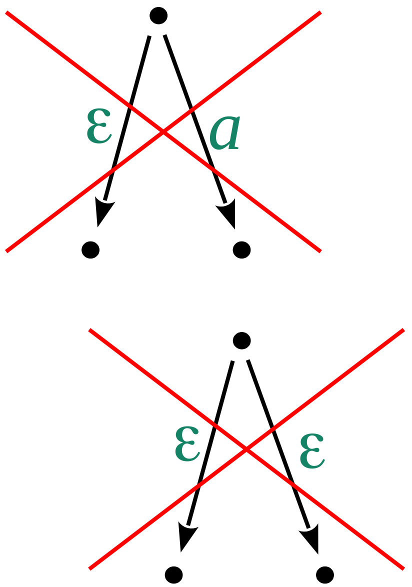


We allow deterministic ε-transitions

allowed:



forbidden:



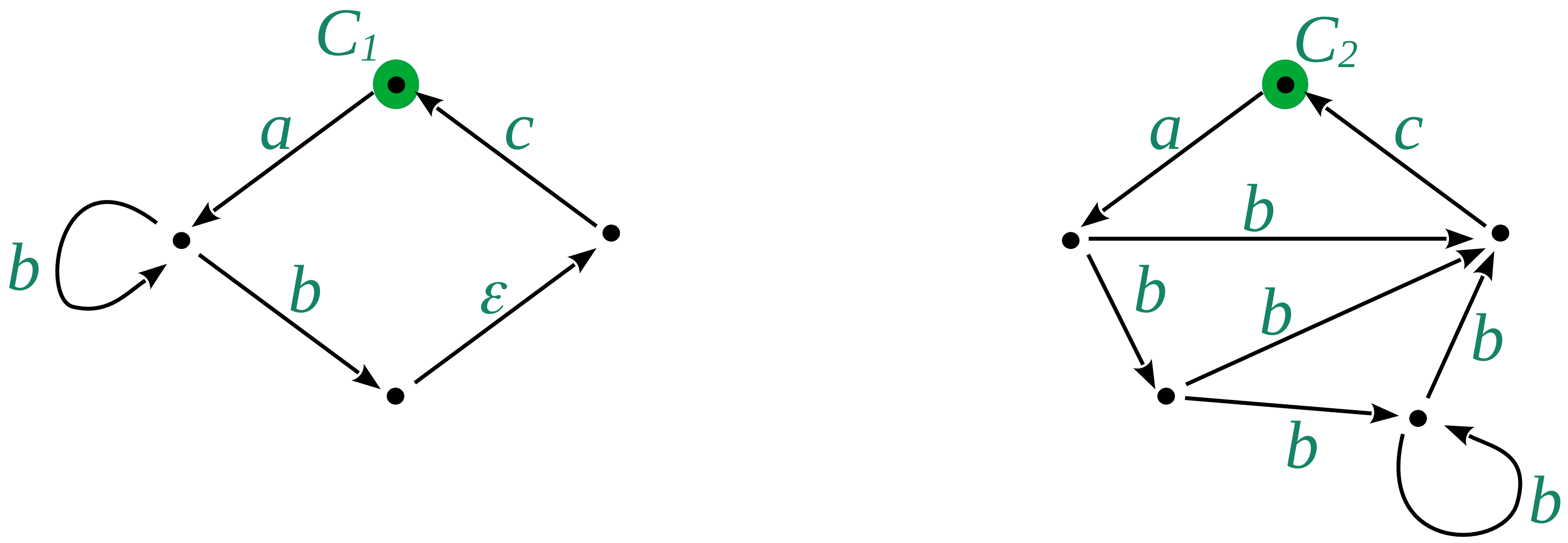
Why study pushdown systems?

Pushdown systems…

- can be used to model the call and return behavior of recursive programs
- have been used to find bugs in Java programs [Suwimontherabuth/Berger/Schwoon/Esparza 1997]
- equivalence checking (in the deterministic case) has been used to verify security protocols [Chrétien, Cortier, Delaune 2015]
- reachability can be checked in polynomial time [Caucal 1990, Bouajjani/Esparza/Maler 1997]
- have a decidable MSO-theory [Muller/Schupp 1985]
- can be model checked against μ-calculus formulas in exponential time [Walukiewicz 1996]

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.

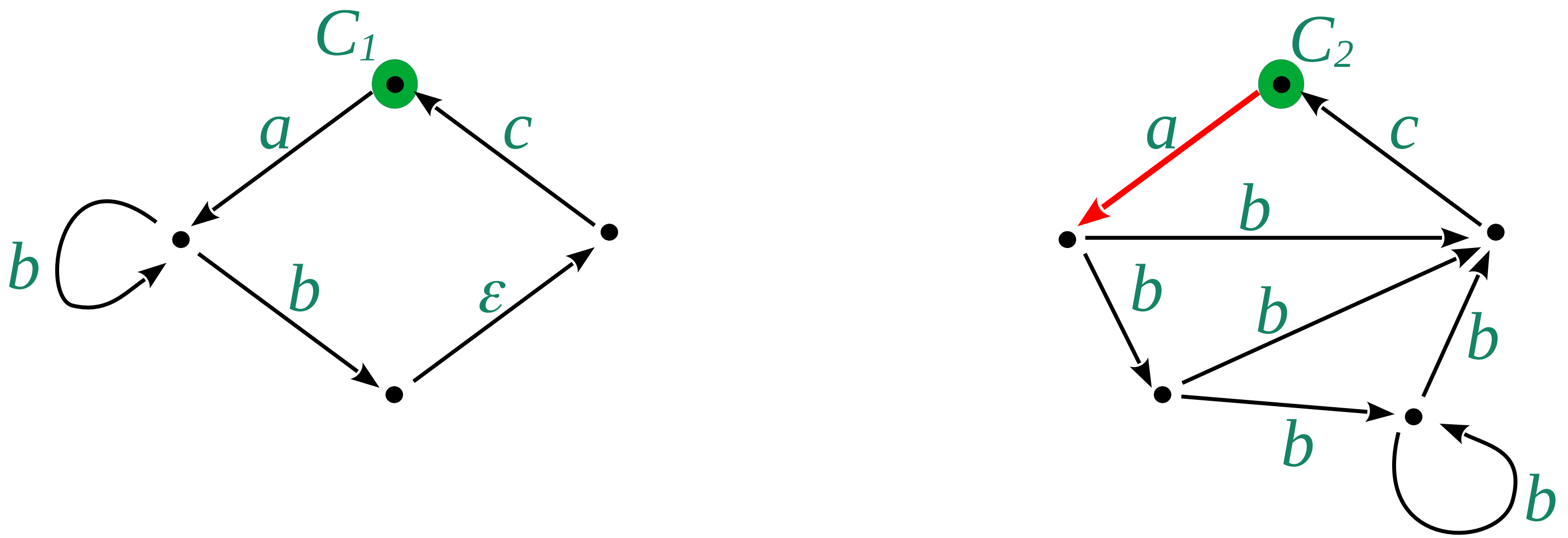


Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.

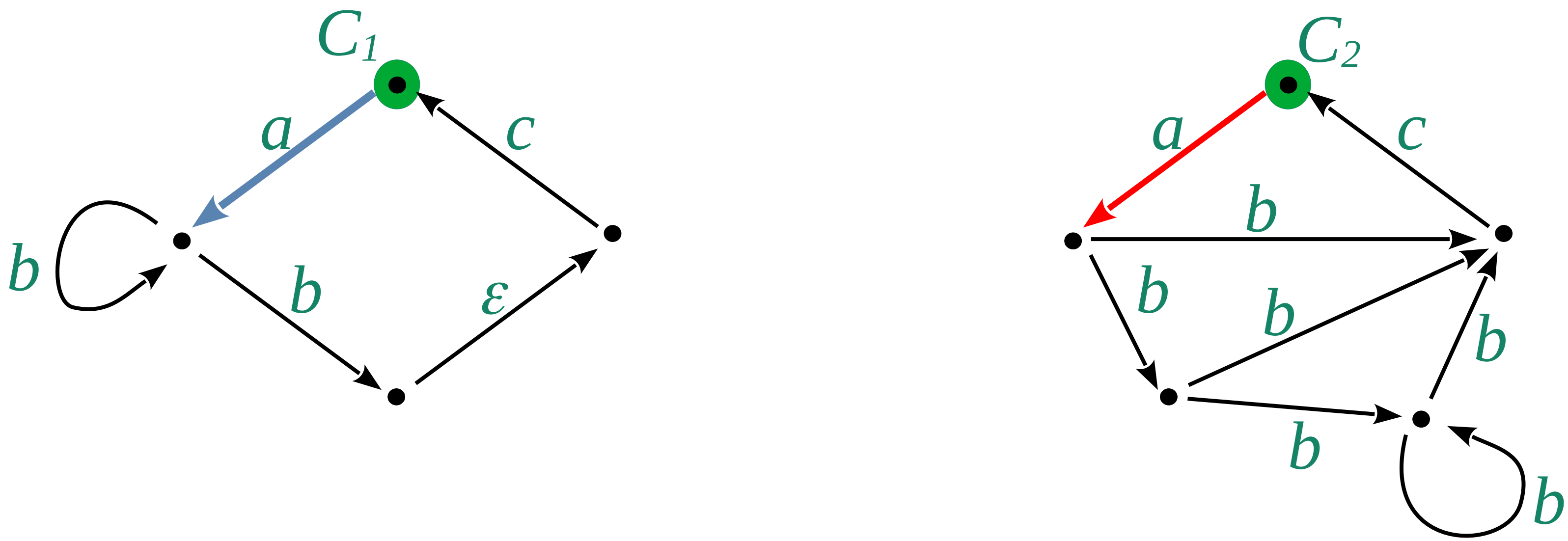


Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.

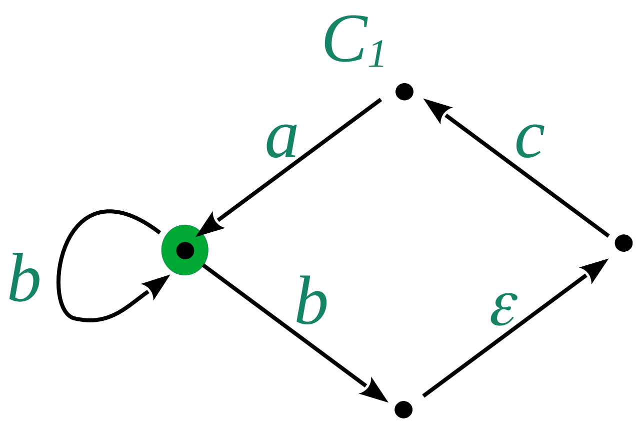


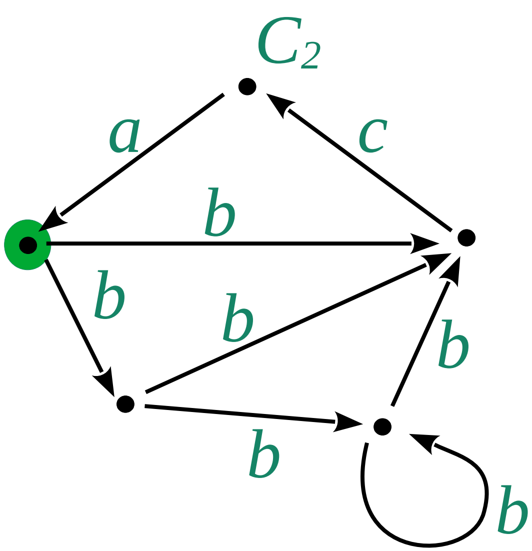
Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.



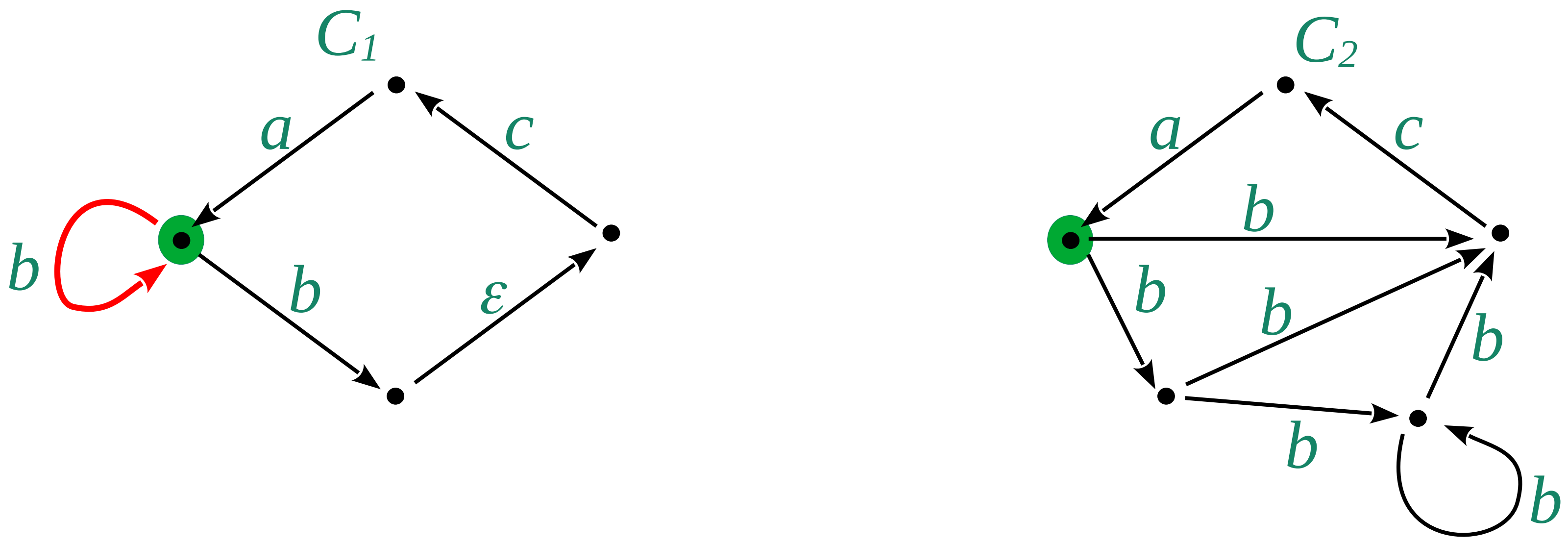


Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.

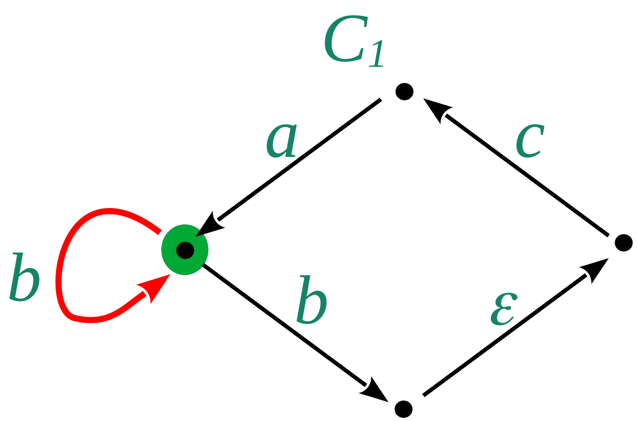


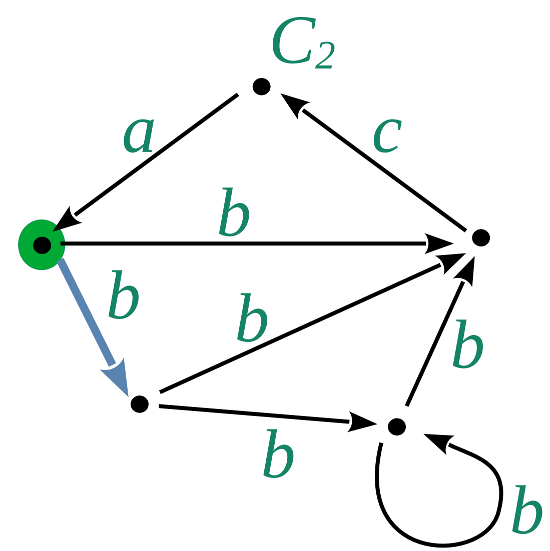
Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.



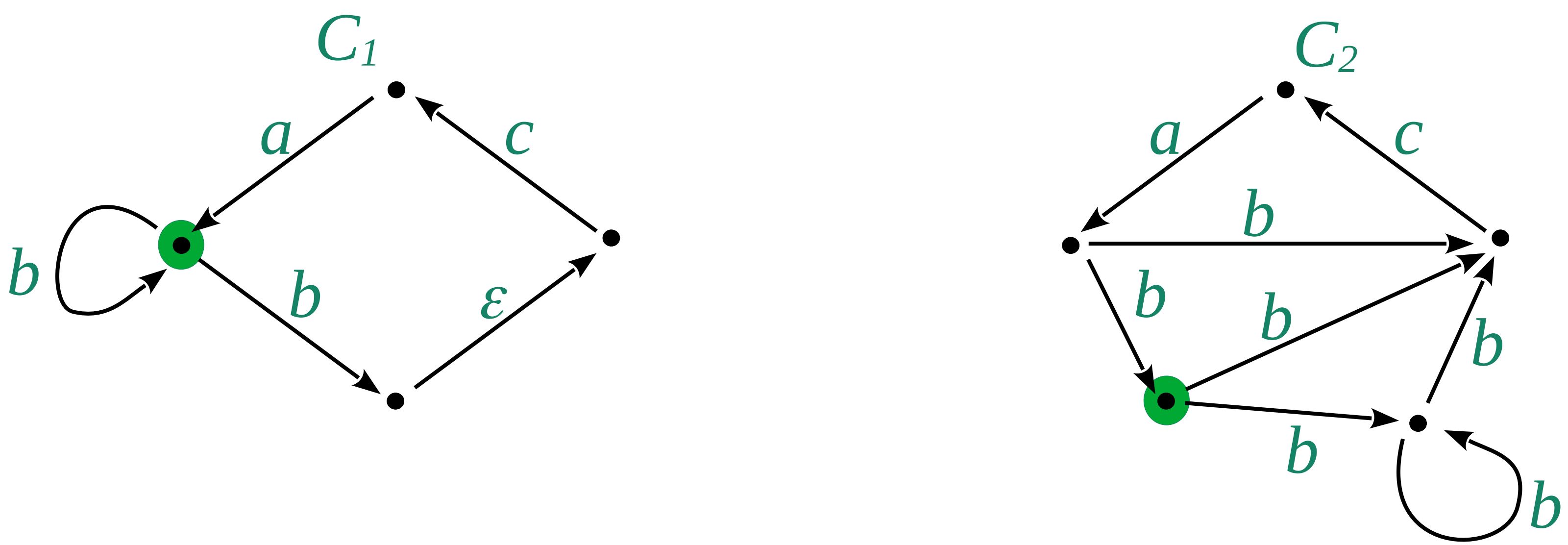


Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.

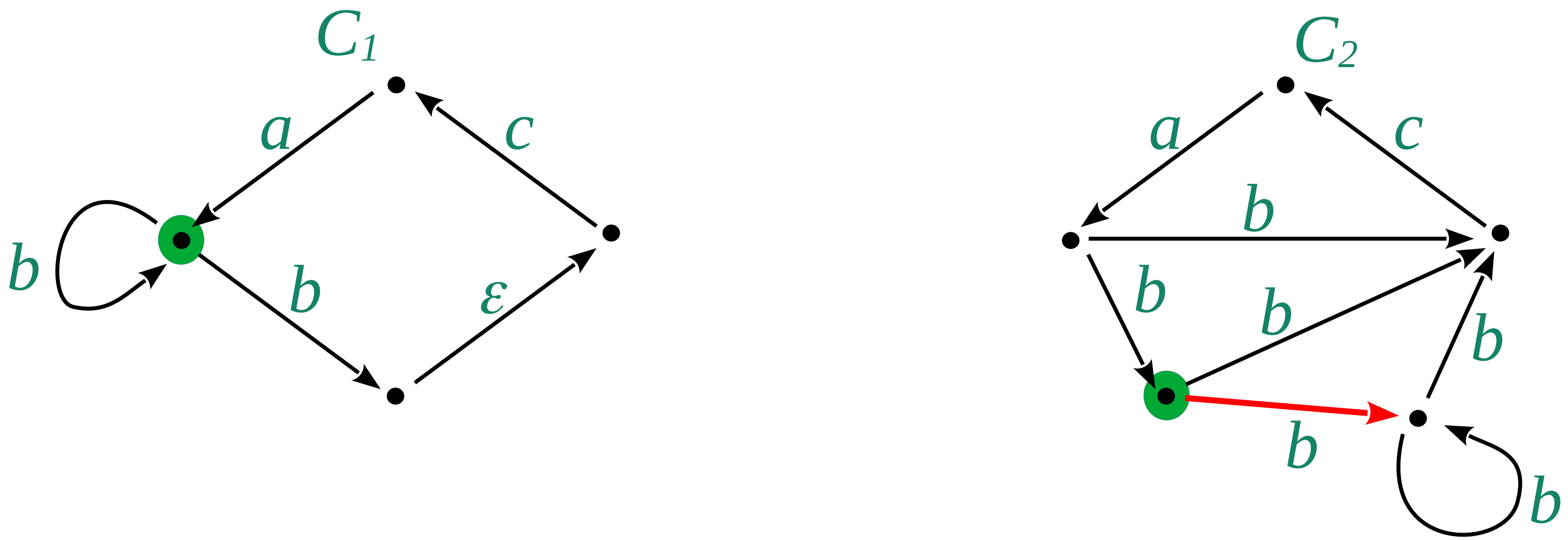


Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.

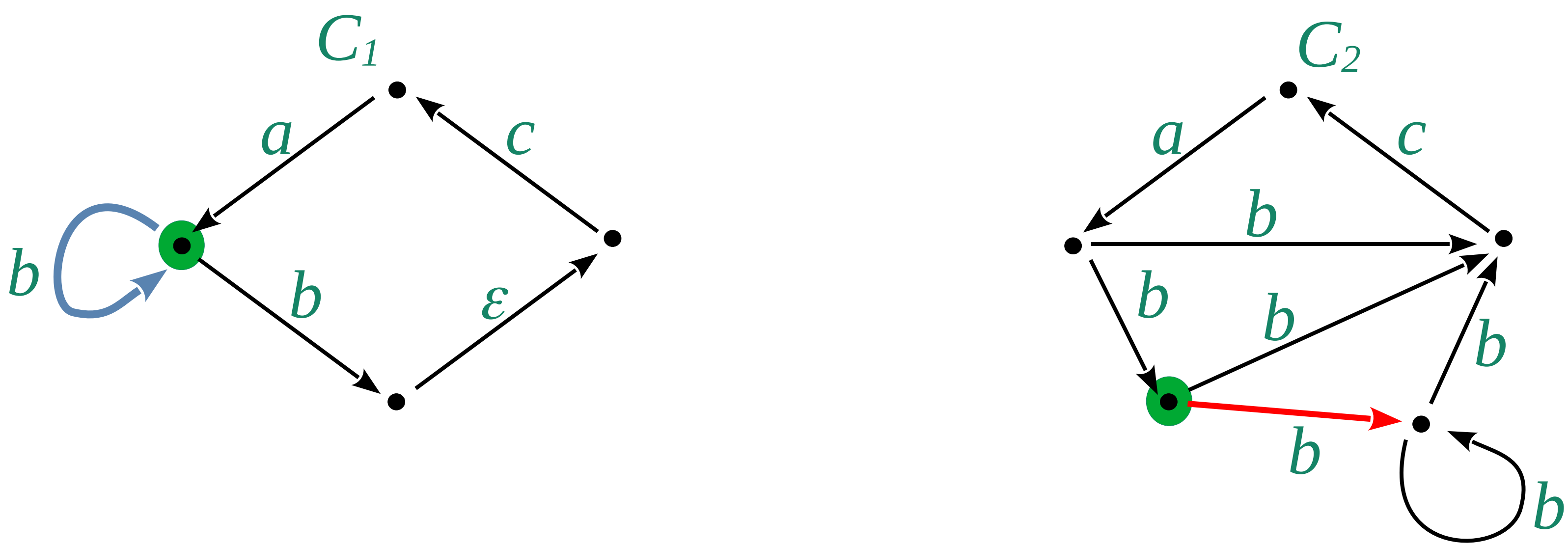


Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.

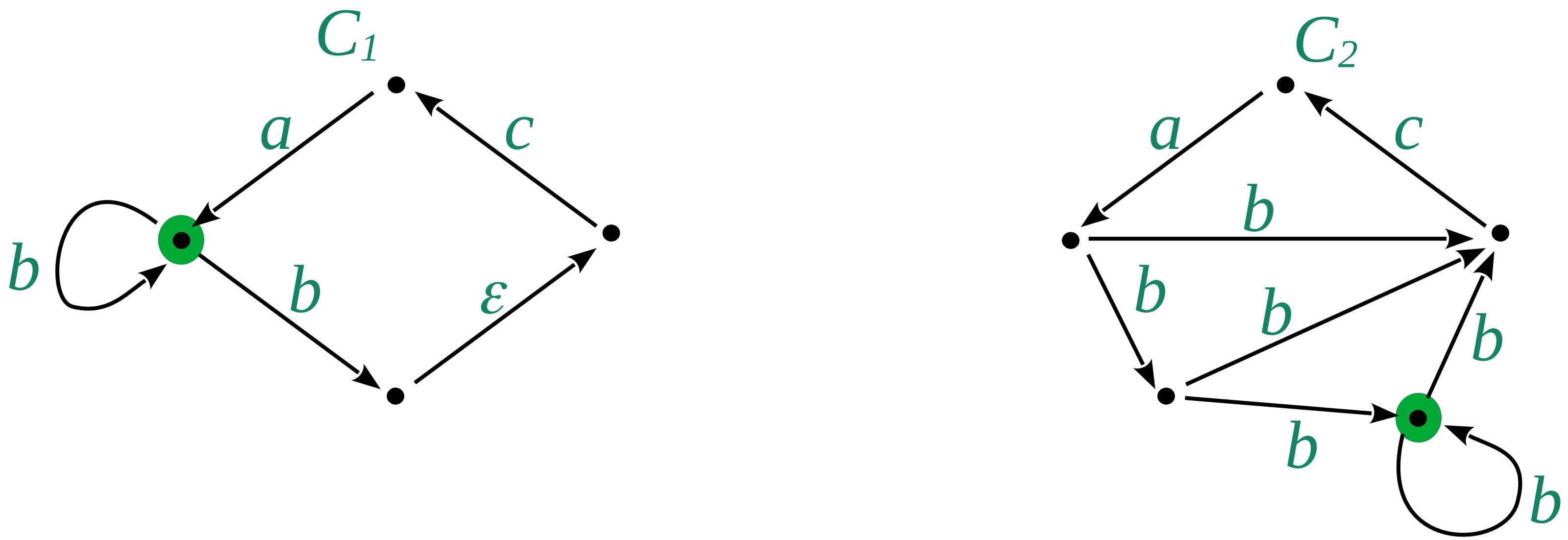


Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.

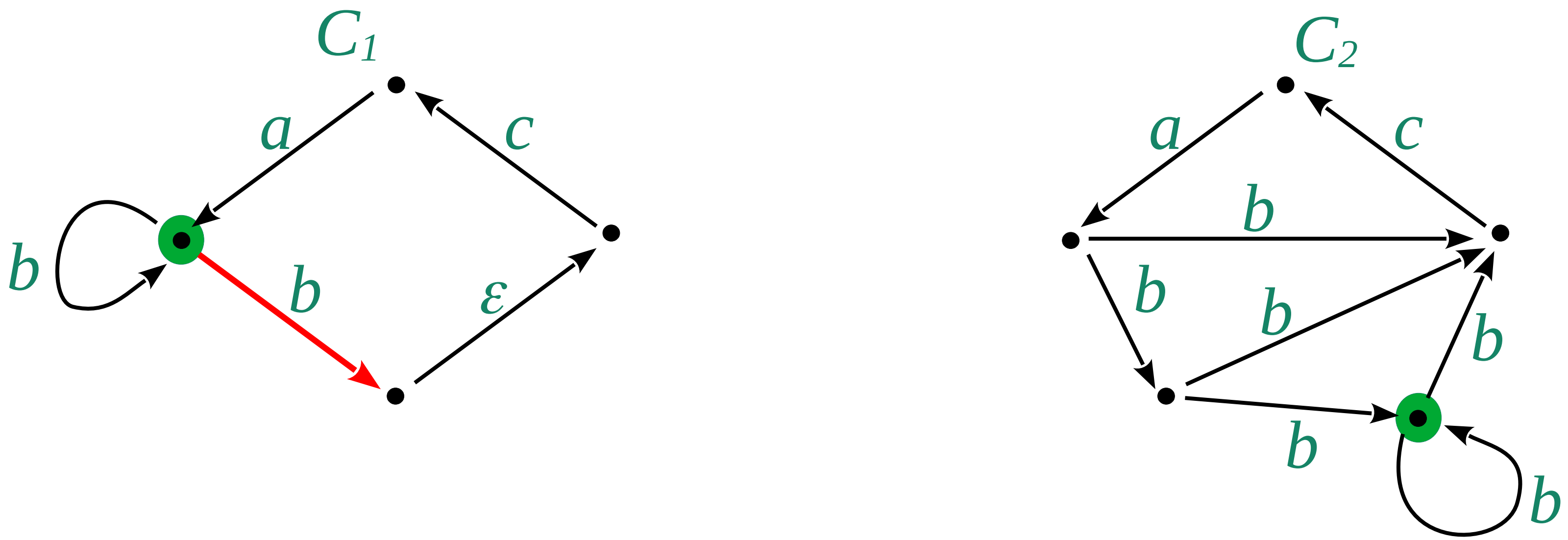


Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.

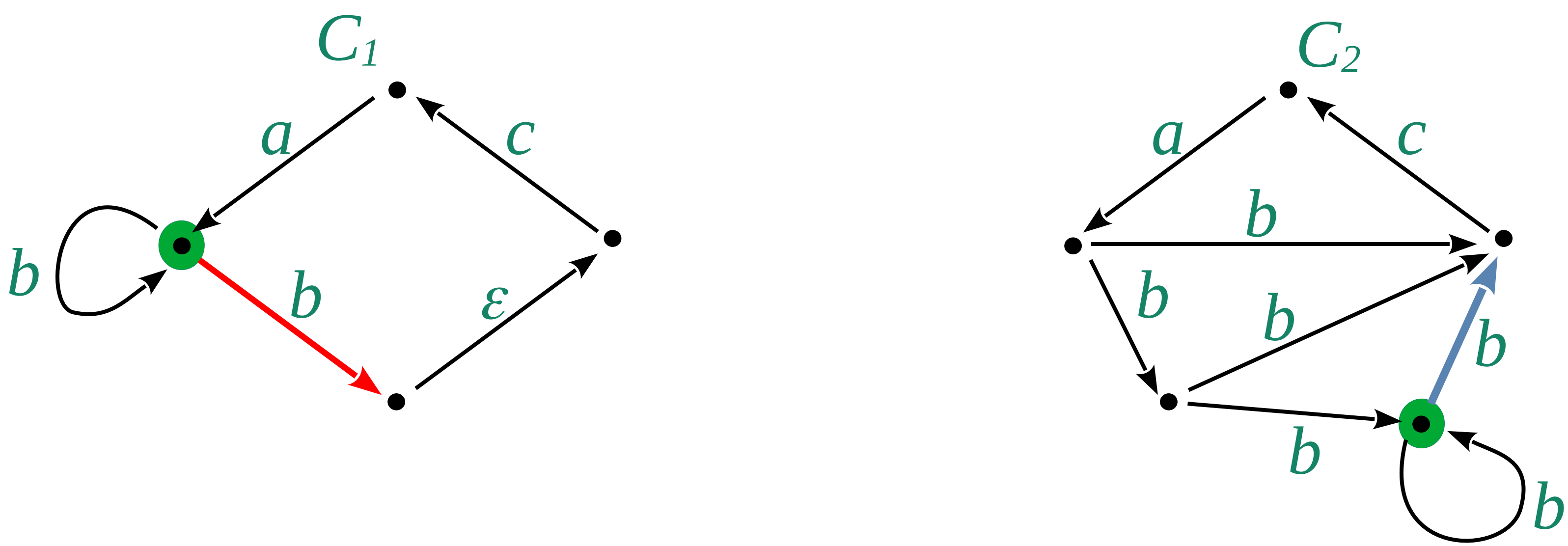


Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.

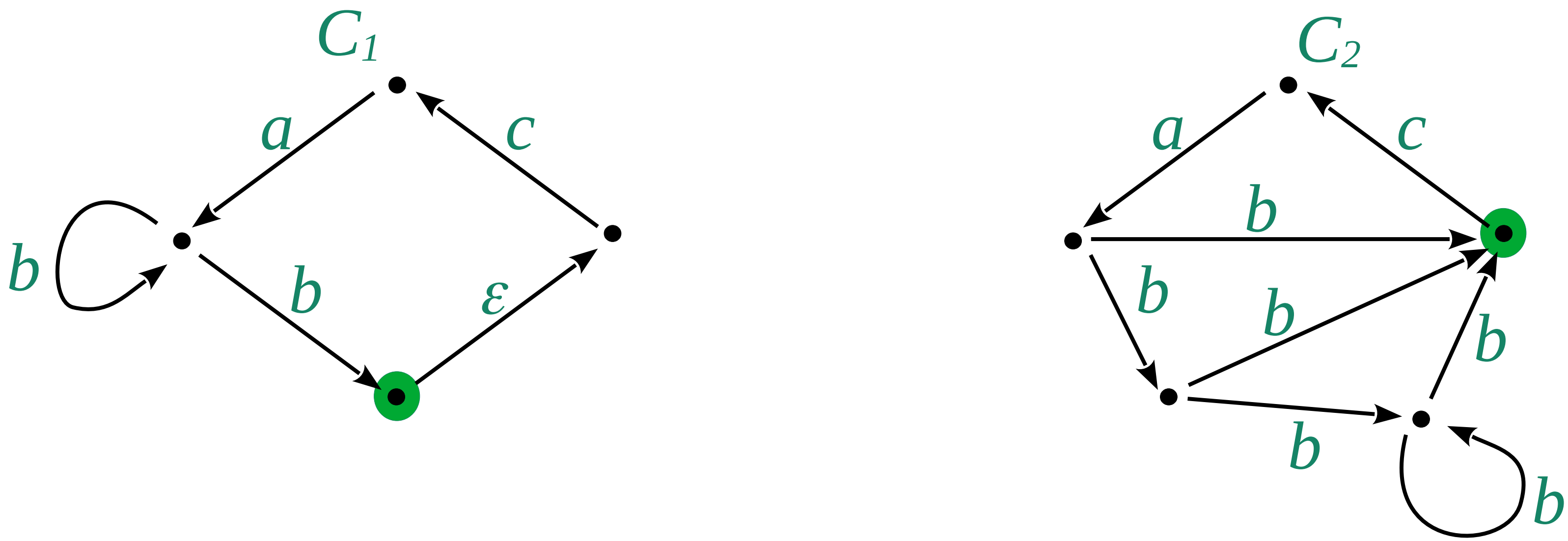


Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.

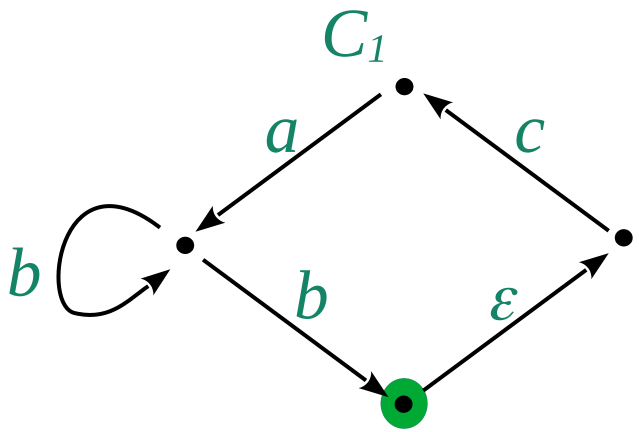


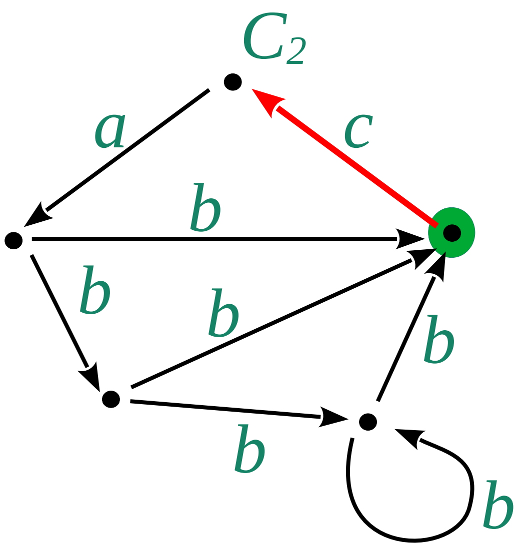
Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.



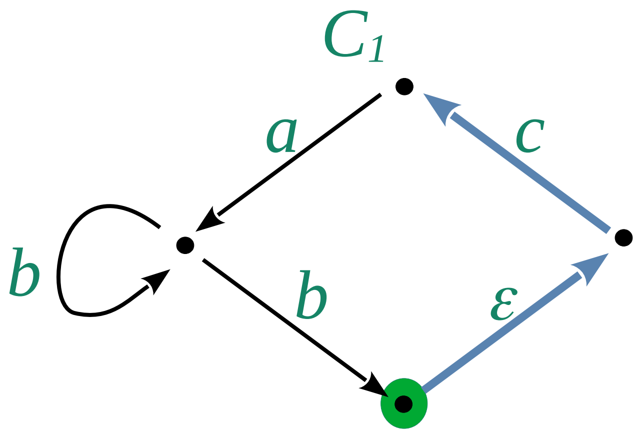


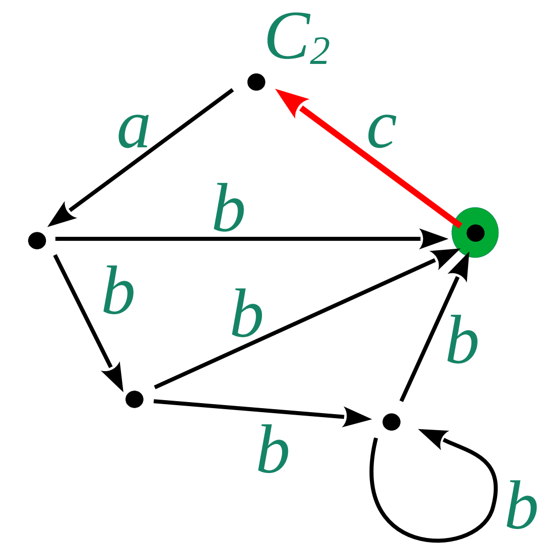
Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

Bisimulation equivalence

can be seen as a two player game between Spoiler and Duplicator.





Spoiler claims that $C_1 \not\sim C_2$

Duplicator claims that $C_1 \sim C_2$

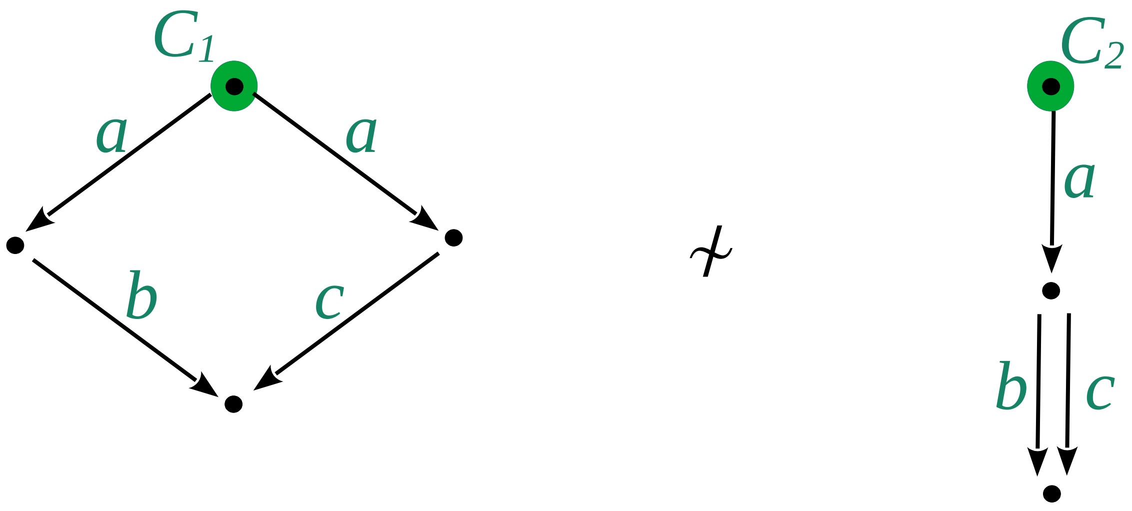
infinite play =
Duplicator wins

Moves = paths $\varepsilon^* a \varepsilon^*$

A.k.a. weak bisimulation
A.k.a. bisimulation after contracting ε-transitions

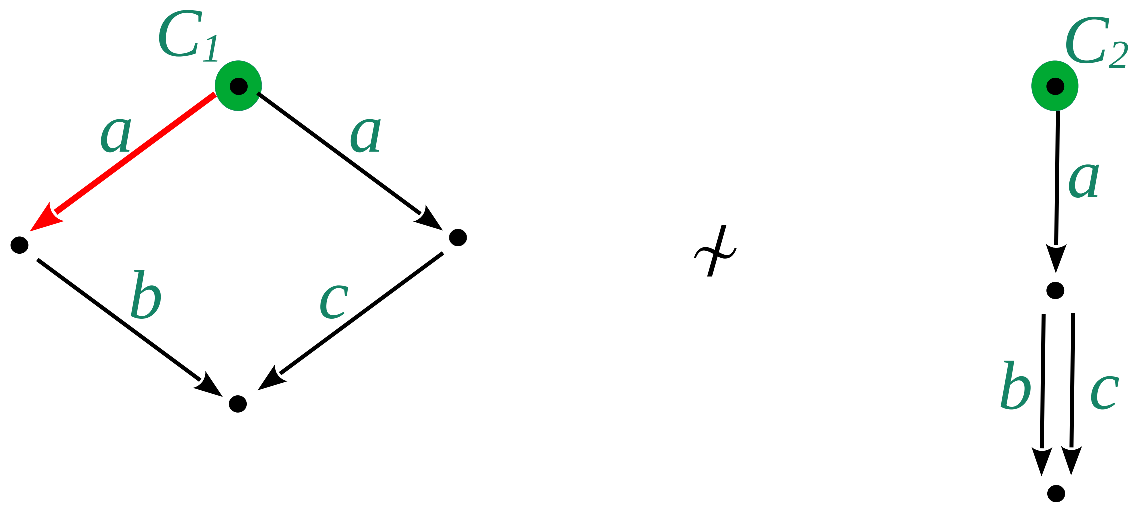
Bisimulation equivalence

Negative example:



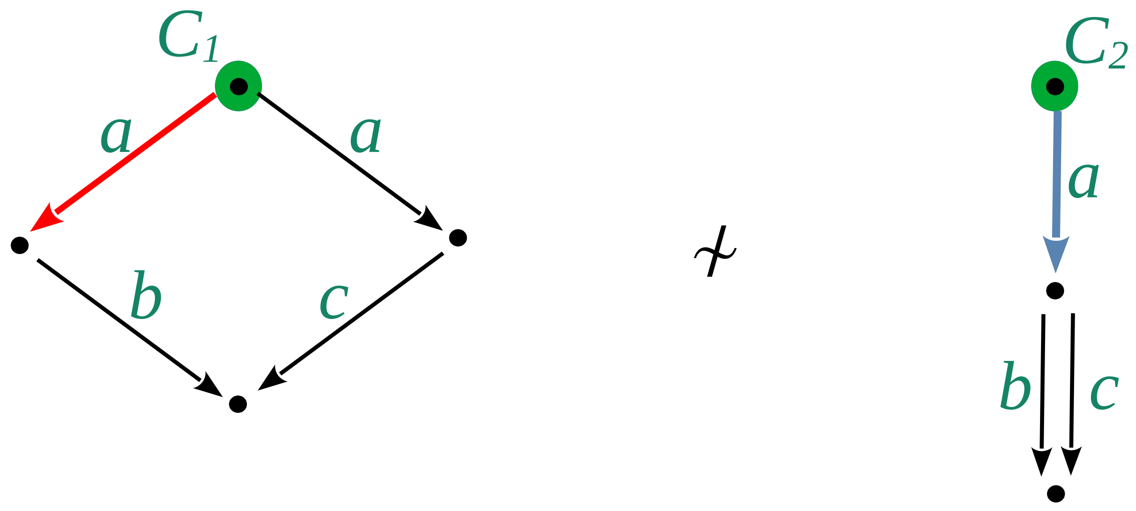
Bisimulation equivalence

Negative example:



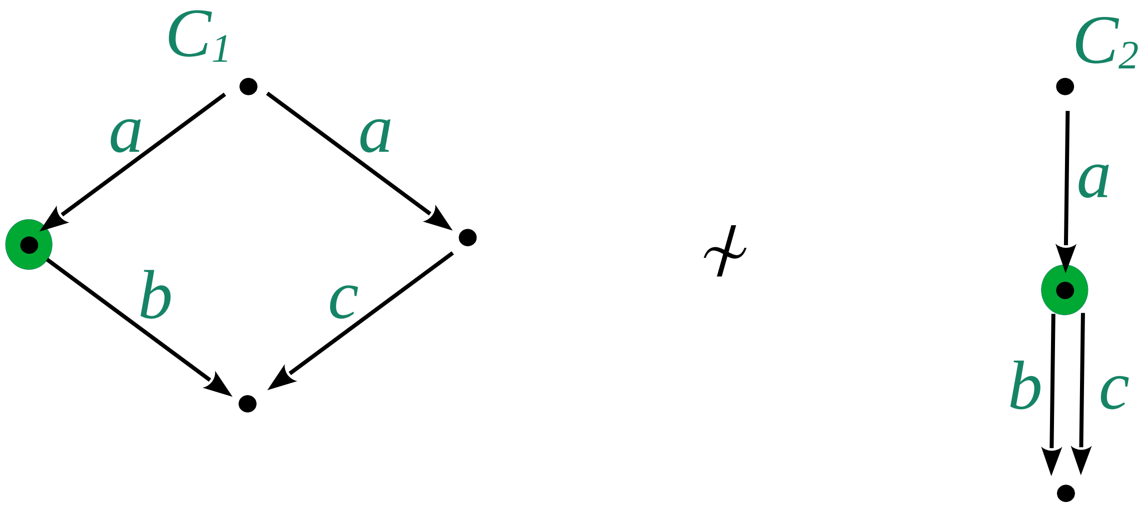
Bisimulation equivalence

Negative example:



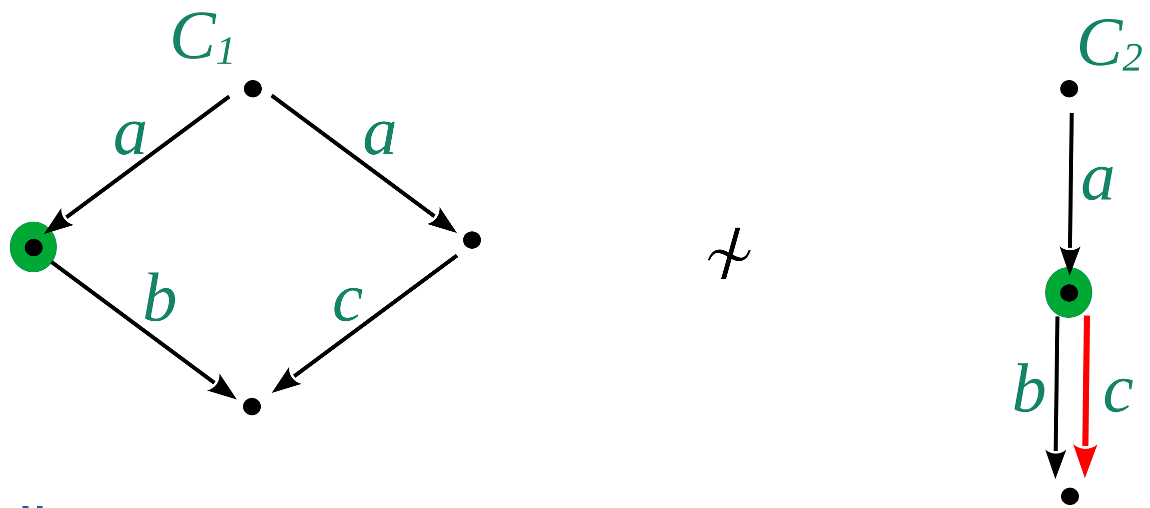
Bisimulation equivalence

Negative example:



Bisimulation equivalence

Negative example:



Duplicator cannot answer

Why bisimulation equivalence?

Verification logics		Classical logics	
Modal logic	=	$\mathrm{FO}_{\sim}$	[van Benthem 1976]
μ-calculus	=	$\mathrm{MSO}_{\sim}$	[Janin/Walukiewicz 1996]
CTL^*	=	$\mathrm{MPL}_{\sim}$	[Moller/Rabinovich 2003]
	$\vdots$		

Bisimulation equivalence is the central notion of equivalence in formal verification!

Bisimulation finiteness

is the following decision problem:

INPUT: a pushdown system P

QUESTION: is P bisimilar to some finite system?

(the finite system is NOT part of the input)

Bisimulation finiteness

is the following decision problem:

INPUT: a pushdown system P

QUESTION: is P bisimilar to some finite system?

(the finite system is NOT part of the input)

Theorem [Jančar 2016]
This problem is decidable.

Proof: two semi-decision procedure;
oracle calls to the bisimulation equivalence problem

Bisimulation equivalence

is the following decision problem:

INPUT: two pushdown systems P_1, P_2

QUESTION: does $P_1 \sim P_2$?

Theorem

This problem is decidable [Sénizergues 1998]

and ACKERMANN-complete [Zhang/Yin/Long/Xu 2020, Schmitz/Jancar 2019]

Bisimulation equivalence

is the following decision problem:

INPUT: two pushdown systems P_1, P_2

QUESTION: does $P_1 \sim P_2$?

Theorem
This problem is decidable [Sénizergues 1998]
and ACKERMANN-complete [Zhang/Yin/Long/Xu 2020, Schmitz/Jancar 2019]

Bisimulation equivalence with a finite system

INPUT: a pushdown system P, a finite system F

QUESTION: does $P \sim F$?

Theorem [Kučera/Mayr 2010]
This problem is PSPACE-complete.

Bisimulation finiteness

INPUT: a pushdown system P

QUESTION: is P bisimilar to some finite system?

(the finite system is NOT part of the input)

- This problem is decidable (in ACKERMANN) [Jančar 2016]
- For P without ε-transitions, it is in 6-EXPSPACE [Göller/Parys 2020]
- **This paper: the problem is 2-EXPTIME-complete**

Our main result

Bisimulation finiteness is 2-EXPTIME-complete

Proof strategy (upper bound)

Step 1: If $P \sim F$ for some F then $P \sim F'$ for some F' of size $< 2^{2^{|P|^c}}$

Step 2: Try to generate minimal F bisimilar to P;
stop when F too large.

Our main result

Bisimulation finiteness is 2-EXPTIME-complete

Proof strategy (upper bound)

Step 1: If $P \sim F$ for some F then $P \sim F'$ for some F' of size $< 2^{2^{|P|^c}}$

Step 1.1: Suppose that $q\alpha\beta^i\gamma$ are reachable for all $i \in \mathbb{N}$, and $P \sim F$.

Then configurations $q\alpha\beta^i\gamma$ for $i > 2^{2^{|P|^c}}$ are all bisimilar.

Step 2: Try to generate minimal F bisimilar to P;

stop when F too large.

Our main result

Bisimulation finiteness is 2-EXPTIME-complete

Proof strategy (lower bound)

- Suppose that P_1, P_2 are bisimulation finite systems.
  Then we can construct $P(P_1,P_2)$ that is bisimulation finite iff $P_1 \sim P_2$

Our main result

Bisimulation finiteness is 2-EXPTIME-complete

Proof strategy (lower bound)

- Suppose that P_1, P_2 are bisimulation finite systems.
  Then we can construct $P(P_1,P_2)$ that is bisimulation finite iff $P_1 \sim P_2$
- We reduce from alternating EXPSPACE Turing machines.
  We have to construct bisimulation finite systems P_1, P_2 such that $P_1 \sim P_2$ iff M accepts.

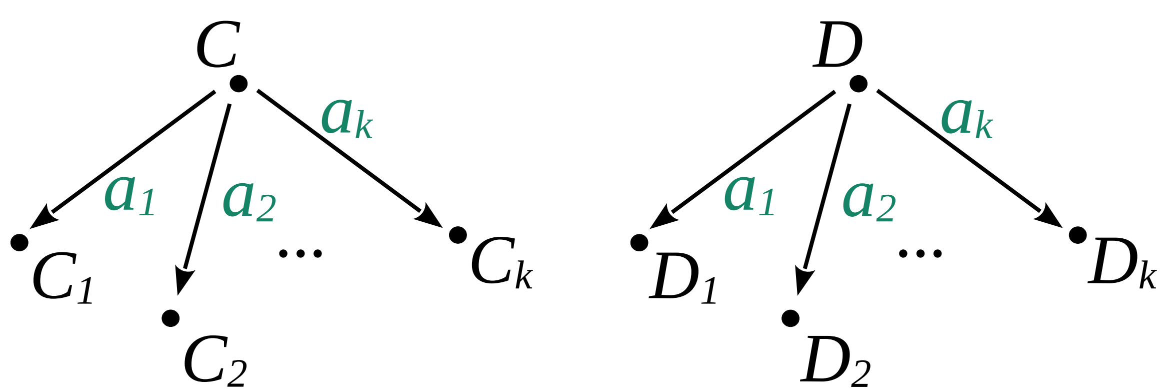
Our main result

Bisimulation finiteness is 2-EXPTIME-complete

Proof strategy (lower bound)

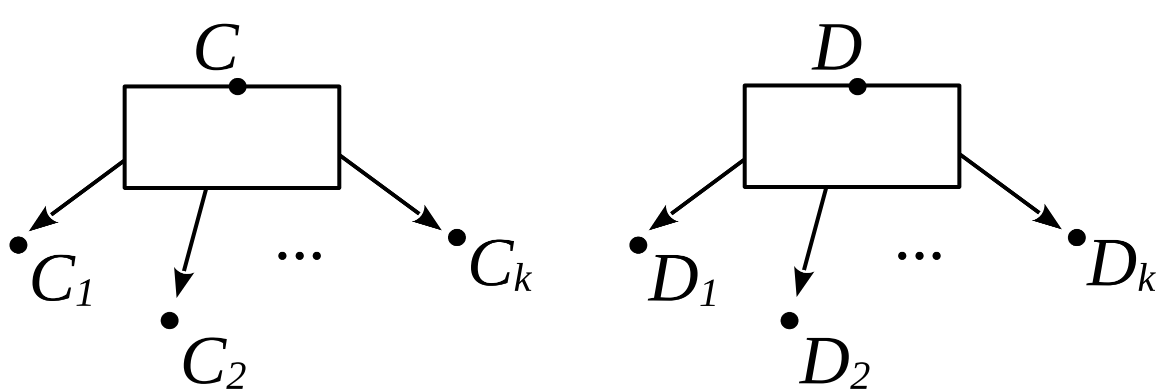
- We have to construct bisimulation finite systems P_1, P_2 such that $P_1 \sim P_2$ iff an alternating EXPSPACE Turing machine M accepts.
- AND realized directly:

$C \sim D$ iff $\forall i.\ C_i \sim D_i$



- OR realized by „Defender's forcing" gadget [Jančar/Srba 2008]:

$C \sim D$ iff $\exists i.\ C_i \sim D_i$



Conclusion

- Bisimulation finiteness of pushdown systems with deterministic ε-transitions is 2-EXPTIME-complete
  (thus much easier than bisimulation equivalence)
- Open problem: complexity for systems without ε-transitions
  - upper bound: 2-EXPTIME
  - lower bound: EXPTIME [Kučera/Mayr 02, Srba 02]
- Generalize the proof to other classes of infinite systems