Branching Bisimilarity on Normed BPA Processes is in NEXPTIME

Wojciech Czerwiński

Petr Jančar

Bisimulation

Bisimulation

Given labelled (multi)graph

Bisimulation

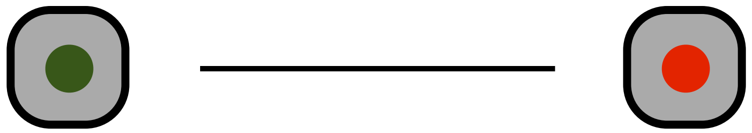
Given labelled (multi)graph

Bisimulation - equivalence on the set of nodes

Bisimulation

Given labelled (multi)graph

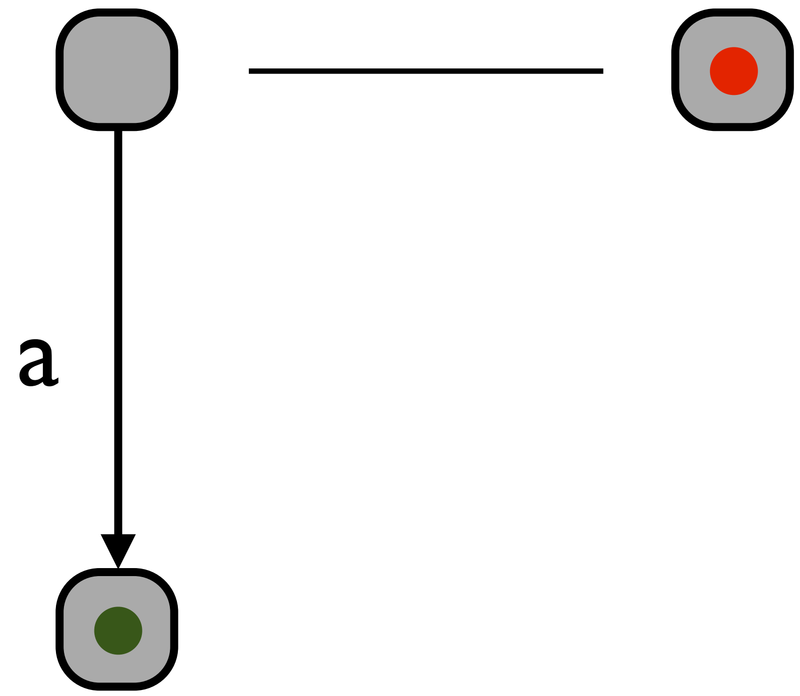
Bisimulation - equivalence on the set of nodes



Bisimulation

Given labelled (multi)graph

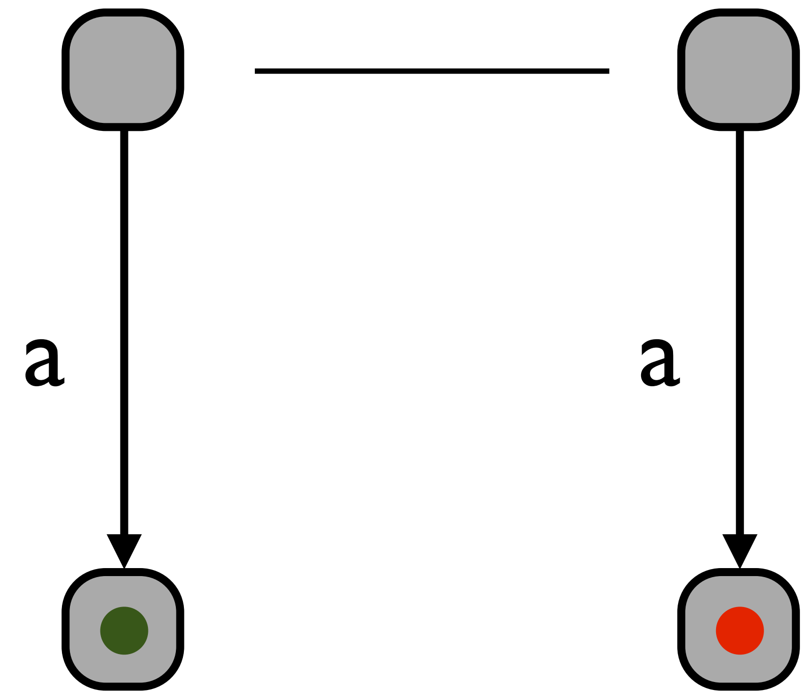
Bisimulation - equivalence on the set of nodes



Bisimulation

Given labelled (multi)graph

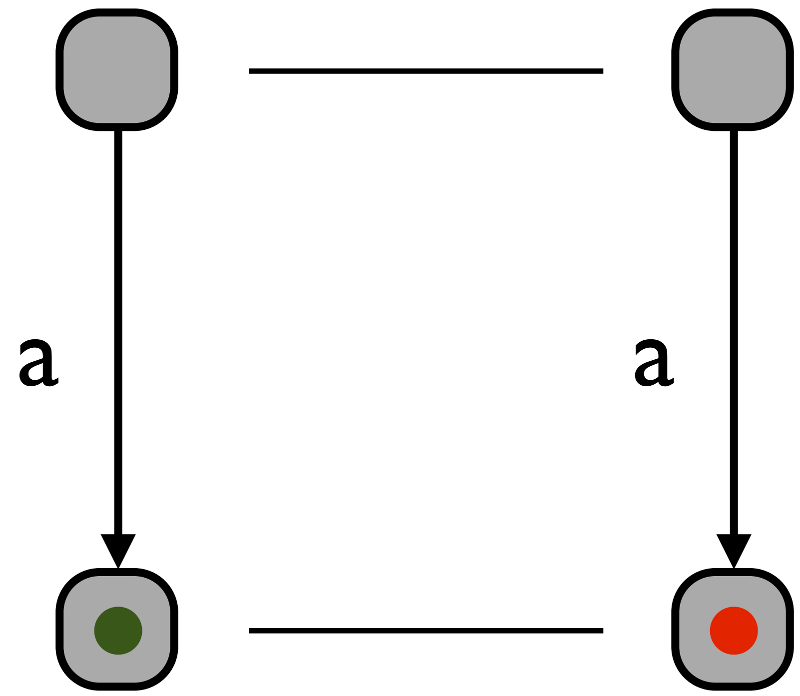
Bisimulation - equivalence on the set of nodes



Bisimulation

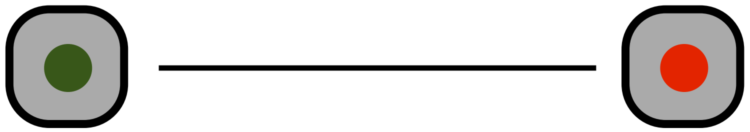
Given labelled (multi)graph

Bisimulation - equivalence on the set of nodes

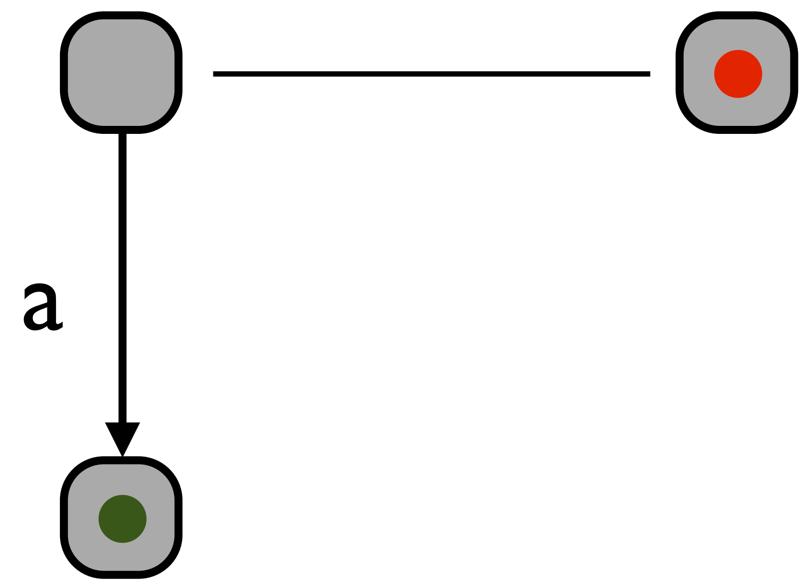


Weak bisimulation

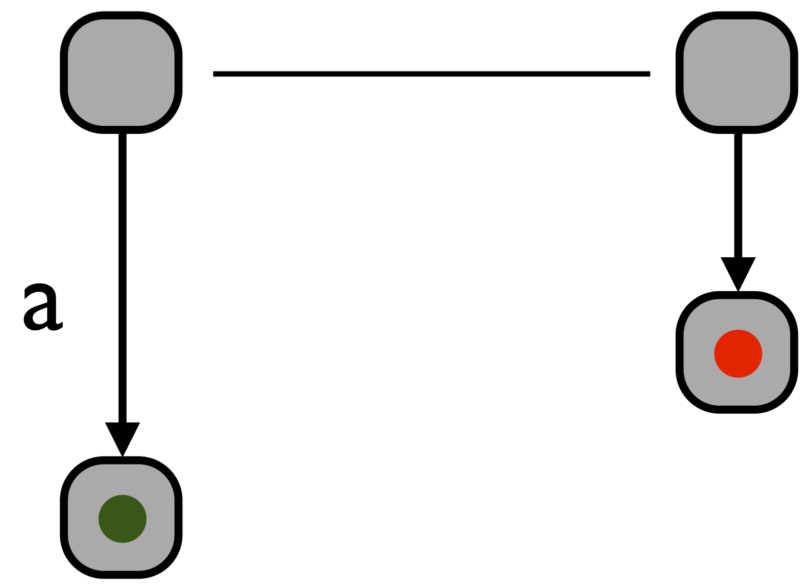
Weak bisimulation



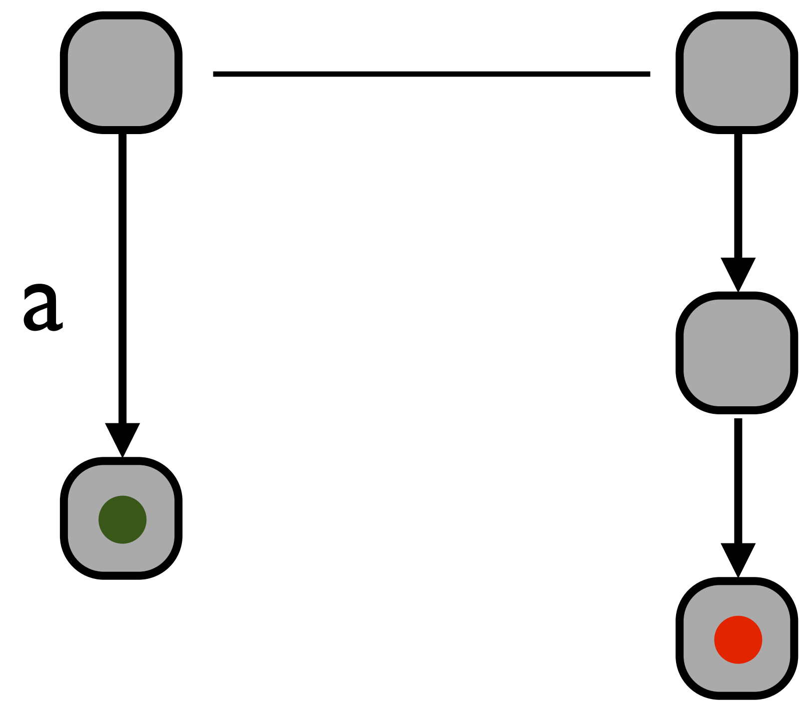
Weak bisimulation



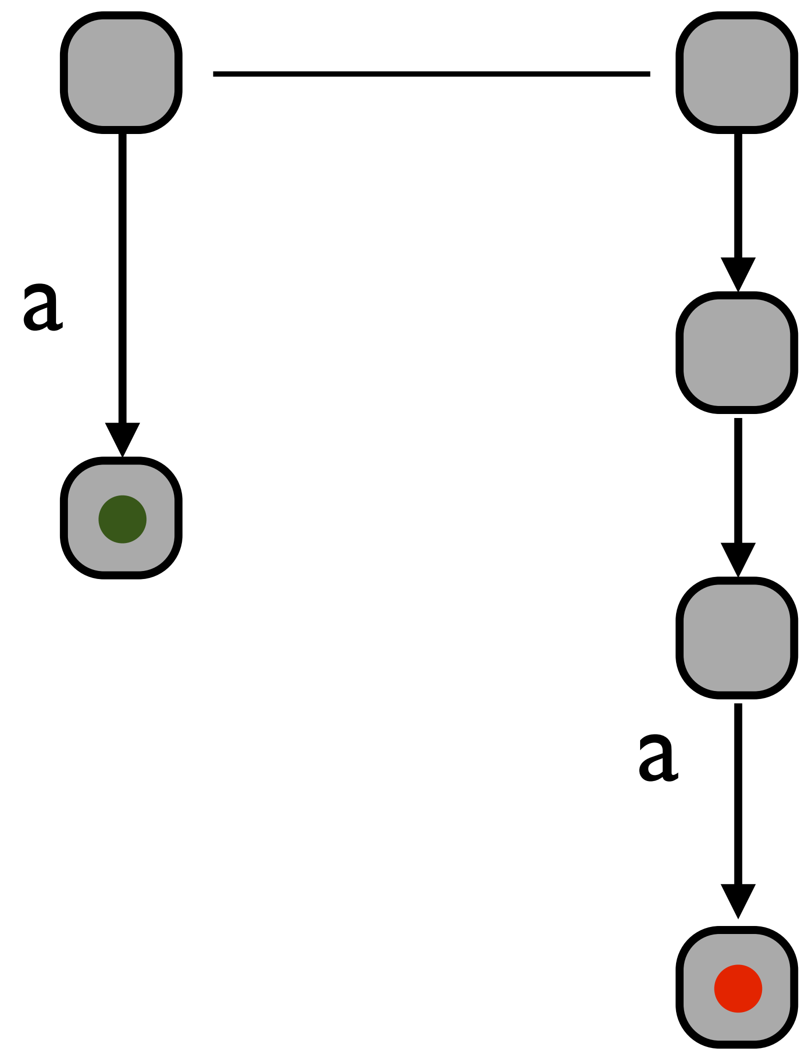
Weak bisimulation



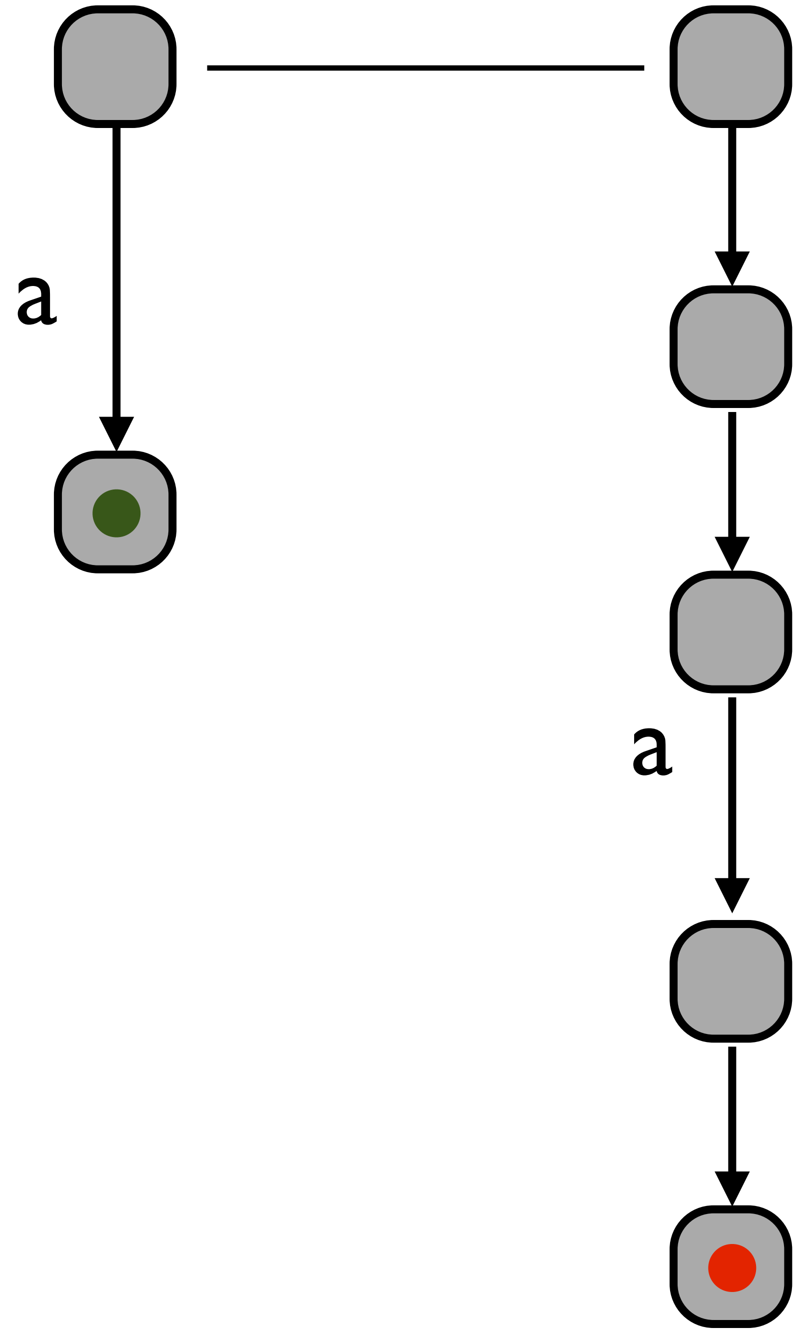
Weak bisimulation



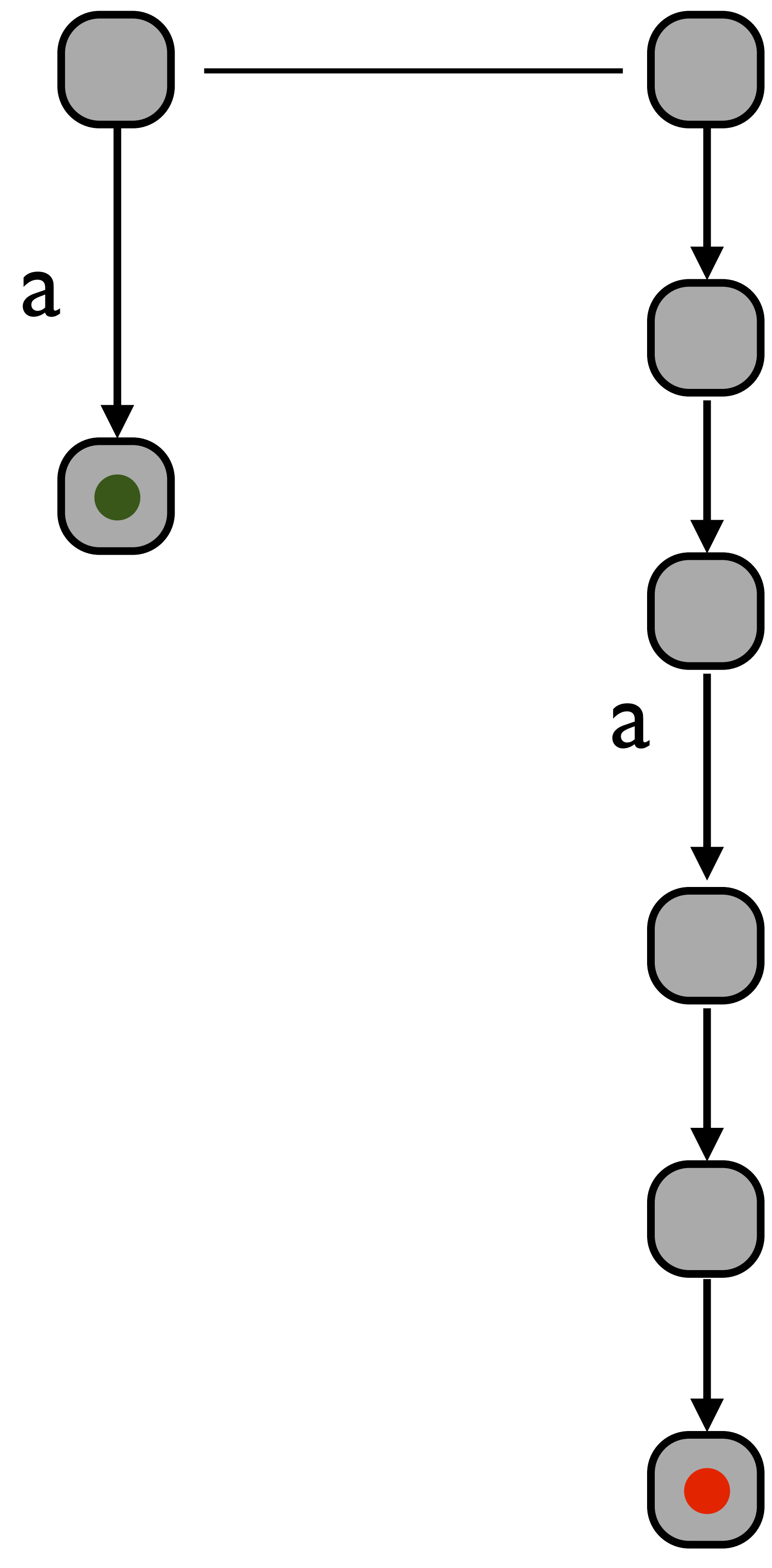
Weak bisimulation



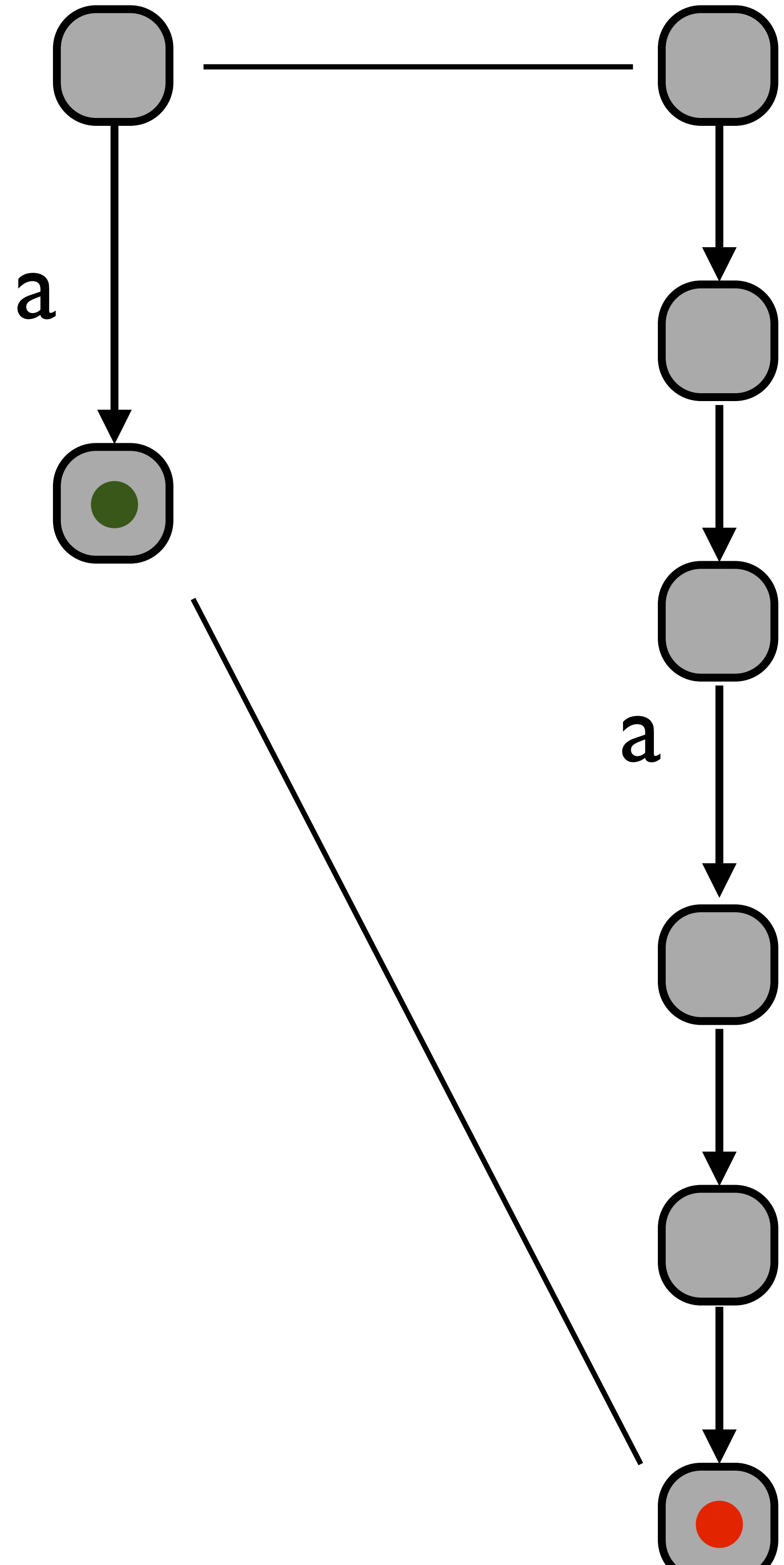
Weak bisimulation



Weak bisimulation

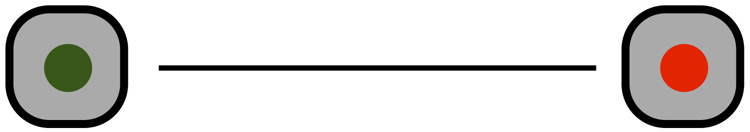


Weak bisimulation

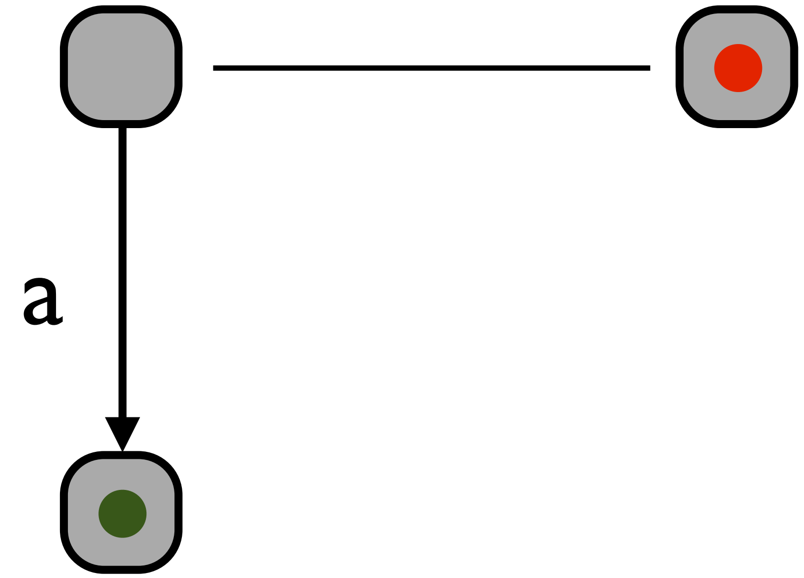


Branching bisimulation

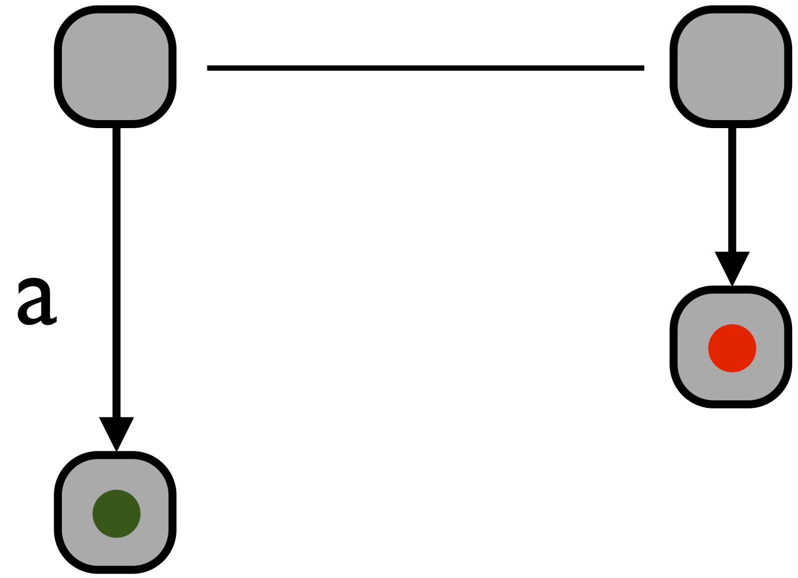
Branching bisimulation



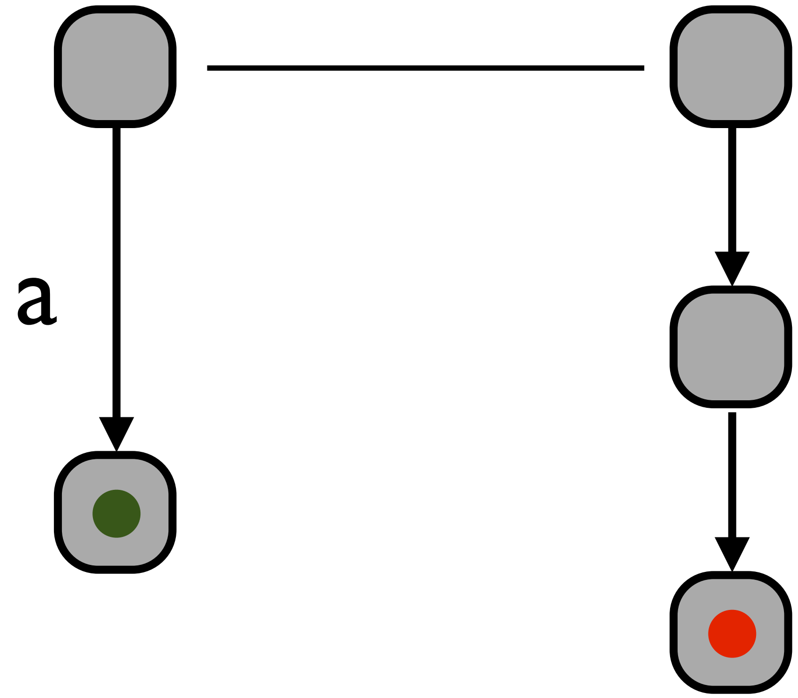
Branching bisimulation



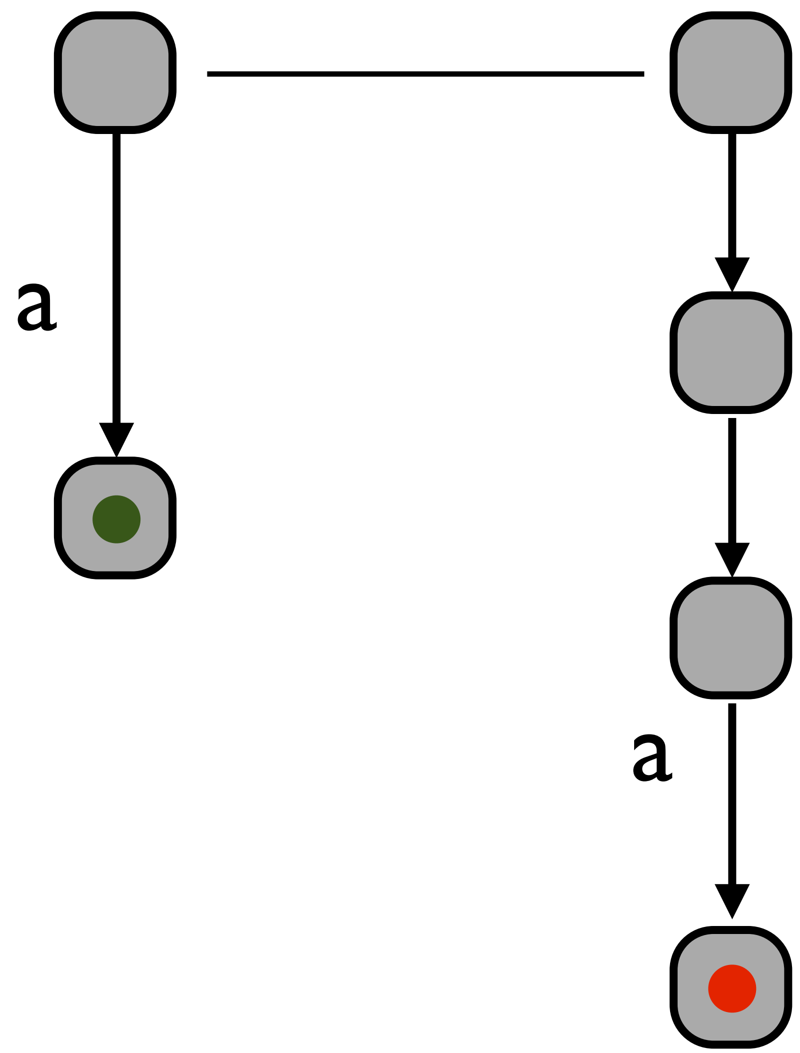
Branching bisimulation



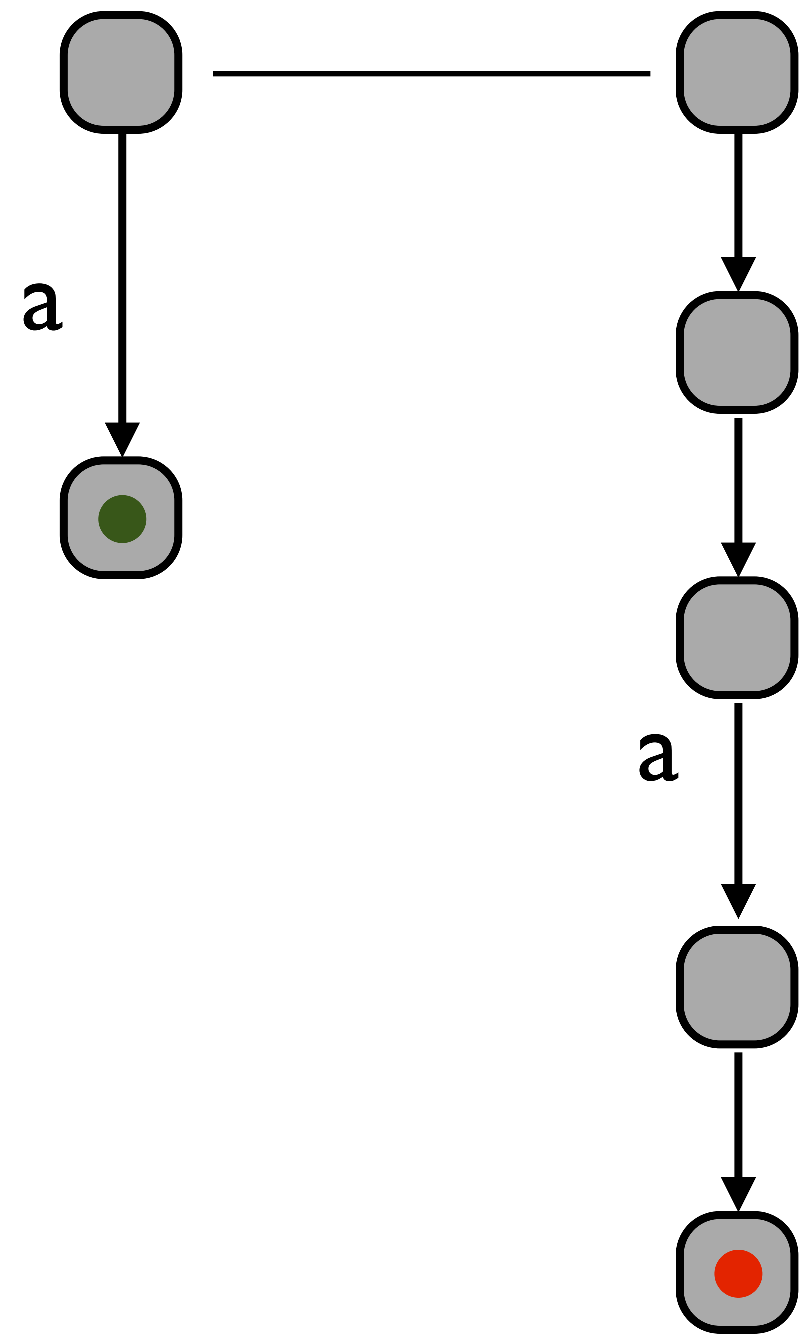
Branching bisimulation



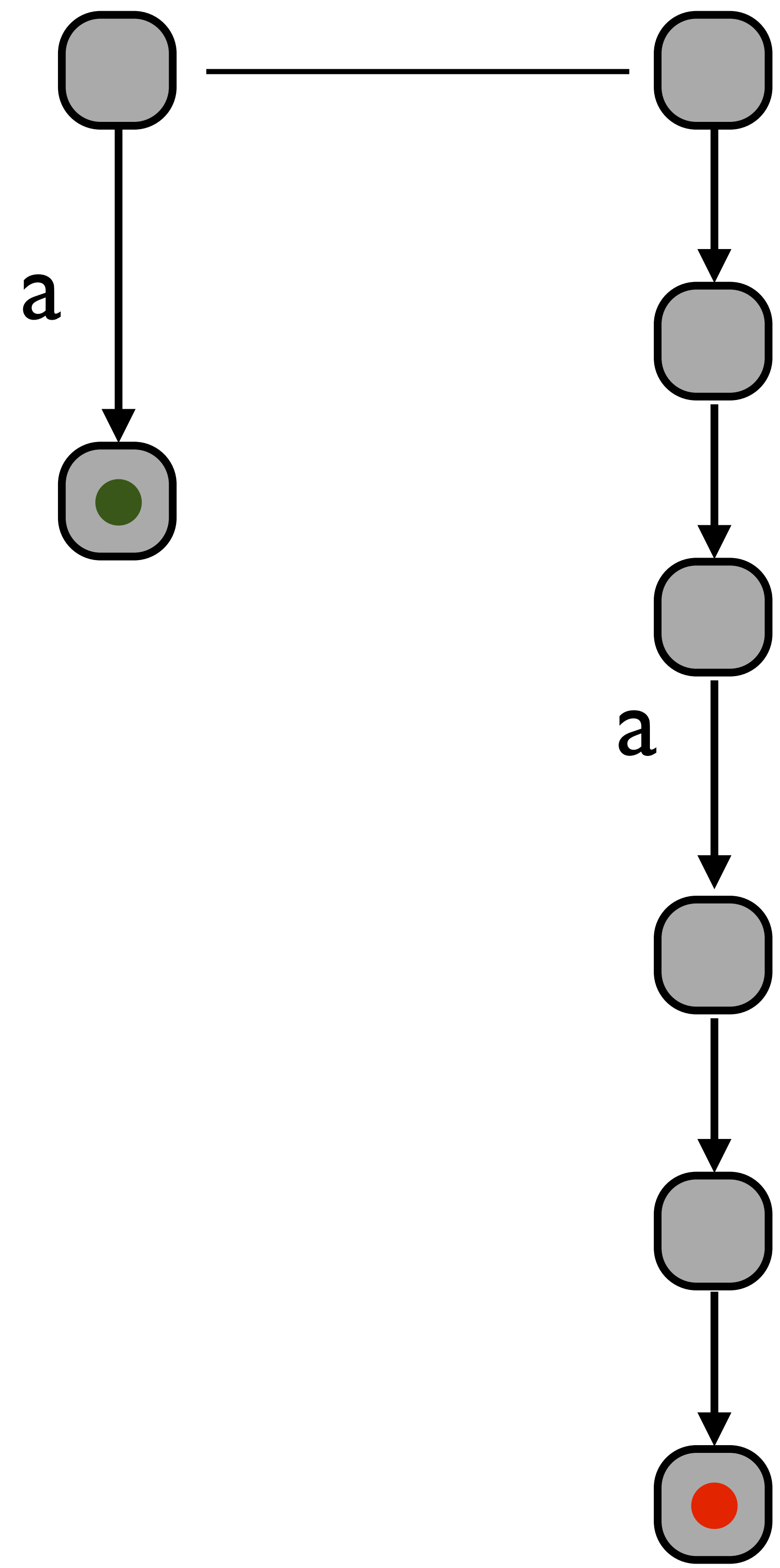
Branching bisimulation



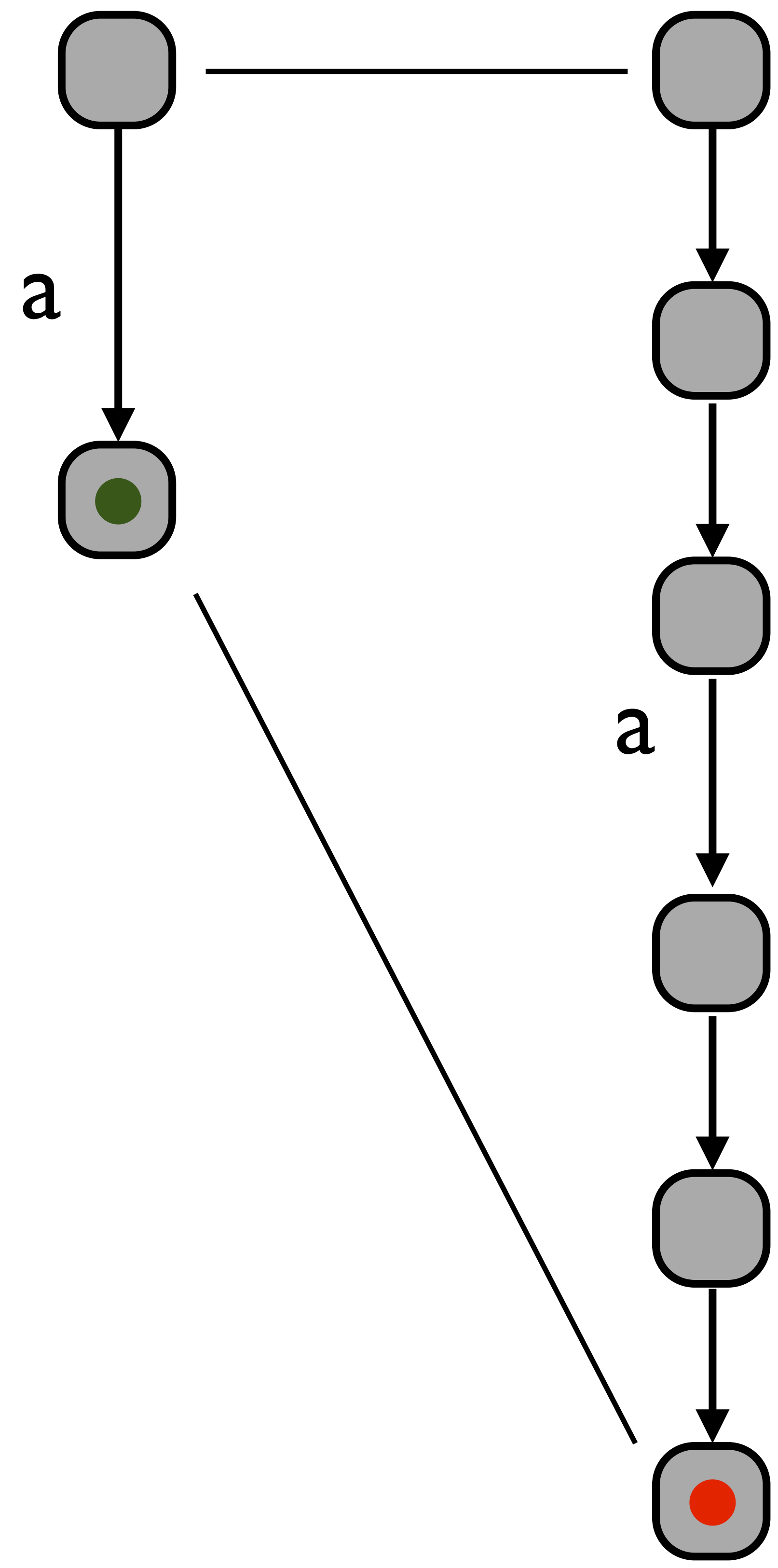
Branching bisimulation



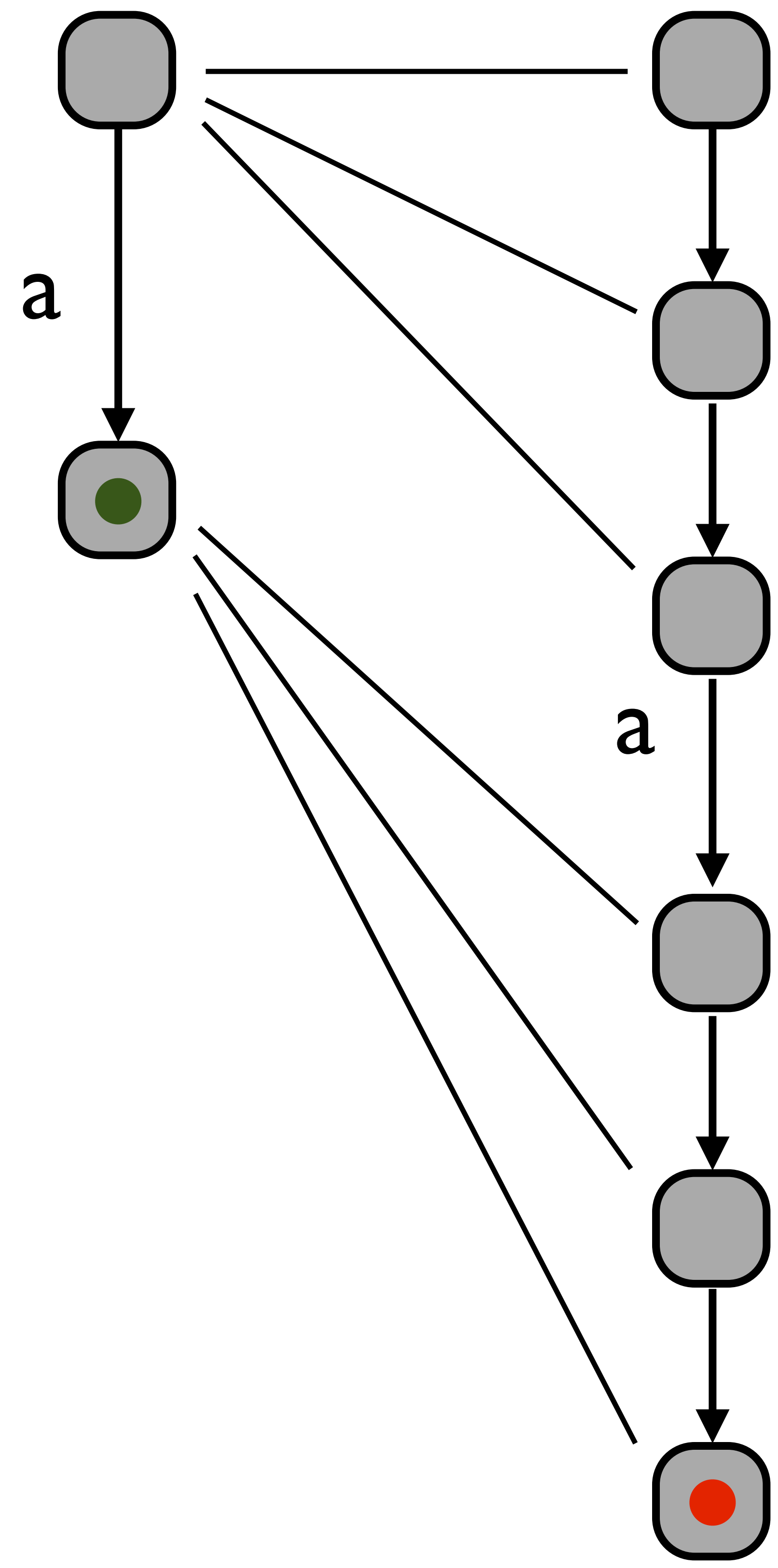
Branching bisimulation



Branching bisimulation



Branching bisimulation



BPA Processes

BPA Processes

BPA = stateless pushdown automaton

BPA Processes

BPA = stateless pushdown automaton

$X \xrightarrow{a} \alpha$ in the automaton

BPA Processes

BPA = stateless pushdown automaton

$X \xrightarrow{a} \alpha$ in the automaton

$\downarrow$

BPA Processes

BPA = stateless pushdown automaton

$$X \xrightarrow{a} \alpha \text{ in the automaton}$$

$$\longmapsto$$

$$X\beta \xrightarrow{a} \alpha\beta \text{ in the multigraph}$$

BPA Processes

BPA = stateless pushdown automaton

$$X \xrightarrow{a} \alpha \text{ in the automaton}$$

$$\downarrow$$

$$X\beta \xrightarrow{a} \alpha\beta \text{ in the multigraph}$$

Normedness: every variable has a path to the empty configuration

Problem

Problem

Given: normed BPA, configurations α and β

Problem

Given: normed BPA, configurations α and β

Question: is α branching bisimilar to β?

Main result

Main result

Theorem

Branching bisimilarity on normed BPA
is in NExpTime

History (personal view)

History (personal view)

- a lot of research on bisimilarity checking (since 90-ties)

History (personal view)

- a lot of research on bisimilarity checking (since 90-ties)
- branching bisimilarity on normed BPA is EXPTIME-hard (Mayr `04)

History (personal view)

- a lot of research on bisimilarity checking (since 90-ties)
- branching bisimilarity on normed BPA is EXPTIME-hard (Mayr `04)
- br. bis. on normed BPP is decidable (C., Hofman, Lasota `11)

History (personal view)

- a lot of research on bisimilarity checking (since 90-ties)
- branching bisimilarity on normed BPA is EXPTIME-hard (Mayr `04)
- br. bis. on normed BPP is decidable (C., Hofman, Lasota `11)
- br. bis. on normed BPA is decidable (Fu `13)

History (personal view)

- a lot of research on bisimilarity checking (since 90-ties)
- branching bisimilarity on normed BPA is EXPTIME-hard (Mayr `04)
- br. bis. on normed BPP is decidable (C., Hofman, Lasota `11)
- br. bis. on normed BPA is decidable (Fu `13)
- br. bis. on normed BPA is in NEXPTIME (now)

History (personal view)

- a lot of research on bisimilarity checking (since 90-ties)
- branching bisimilarity on normed BPA is EXPTIME-hard (Mayr `04)
- br. bis. on normed BPP is decidable (C., Hofman, Lasota `11)
- br. bis. on normed BPA is decidable (Fu `13)
- br. bis. on normed BPA is in NEXPTIME (now)
- br. bis. on normed BPA is EXPTIME-comp. (He, Huang LICS`15)

Idea

Idea

- show that branching bisimilarity can be represented by an exponential *base*

Idea

- show that branching bisimilarity can be represented by an exponential *base*
- guess an exponential base

Idea

- show that branching bisimilarity can be represented by an exponential *base*
- guess an exponential base
- verify its correctness

Idea

- show that branching bisimilarity can be represented by an exponential *base*
- guess an exponential base
- verify its correctness
- check whether (α, β) belongs to guessed relation

Unique prime decomposition

Unique prime decomposition

- idea - from the strong bisimulation

Unique prime decomposition

- idea - from the strong bisimulation
- variables: decomposable or prime

Unique prime decomposition

- idea - from the strong bisimulation
- variables: decomposable or prime
- every decomposable has a decomposition into primes

Unique prime decomposition

- idea - from the strong bisimulation
- variables: decomposable or prime
- every decomposable has a decomposition into primes
- every configuration - exactly one equivalent configuration using only primes

Unique prime decomposition

- idea - from the strong bisimulation
- variables: decomposable or prime
- every decomposable has a decomposition into primes
- every configuration - exactly one equivalent configuration using only primes
- bisimilarity = equality of prime forms

For branching bisimulation

For branching bisimulation

unique decomposition does not work!

For branching bisimulation

unique decomposition does not work!

$$A \xrightarrow{a} A$$

For branching bisimulation

unique decomposition does not work!

$$A \xrightarrow{a} A \qquad A \xrightarrow{a} \varepsilon$$

For branching bisimulation

unique decomposition does not work!

$$A \xrightarrow{a} A \qquad A \xrightarrow{a} \varepsilon \qquad A \longrightarrow \varepsilon$$

For branching bisimulation

unique decomposition does not work!

$A \xrightarrow{a} A \qquad A \xrightarrow{a} \varepsilon \qquad A \longrightarrow \varepsilon$

$A \sim AA$

For branching bisimulation

unique decomposition does not work!

$$A \xrightarrow{a} A \qquad A \xrightarrow{a} \varepsilon \qquad A \longrightarrow \varepsilon$$

$$A \sim AA \qquad A \nsim \varepsilon$$

For branching bisimulation

unique decomposition does not work!

$$A \xrightarrow{a} A \qquad A \xrightarrow{a} \varepsilon \qquad A \longrightarrow \varepsilon$$

$$A \sim AA \qquad A \not\sim \varepsilon$$

$$pf(A) = pf(A)\, pf(A) \Rightarrow pf(A) = \varepsilon$$

For branching bisimulation

unique decomposition does not work!

$$A \xrightarrow{a} A \qquad A \xrightarrow{a} \varepsilon \qquad A \longrightarrow \varepsilon$$

$$A \sim AA \qquad A \not\sim \varepsilon$$

$$pf(A) = pf(A)\, pf(A) \Rightarrow pf(A) = \varepsilon$$

some more ideas needed!

Redundant variables

Redundant variables

$$\mathrm{RED}(\alpha) = \{X : X\alpha \sim \alpha\}$$

Redundant variables

$$\mathrm{RED}(\alpha) = \{X : X\alpha \sim \alpha\}$$

redundant variables provide full information about the suffix:

Redundant variables

$$\mathrm{RED}(\alpha) = \{X: X\alpha \sim \alpha\}$$

redundant variables provide full information about the suffix:

$$\mathrm{RED}(\alpha) = \mathrm{RED}(\beta) \Rightarrow (\gamma\alpha \sim \delta\alpha \Leftrightarrow \gamma\beta \sim \delta\beta)$$

Main technical contribution

Main technical contribution

- relative decomposition works!

Main technical contribution

- relative decomposition works!
- decomposition depends on the suffix

Main technical contribution

- relative decomposition works!
- decomposition depends on the suffix
- concretely: on the RED(suffix)

Main technical contribution

- relative decomposition works!
- decomposition depends on the suffix
- concretely: on the RED(suffix)
- one decomposition system for every $R \subseteq Var$ is enough

Main technical contribution

- relative decomposition works!
- decomposition depends on the suffix
- concretely: on the RED(suffix)
- one decomposition system for every $R \subseteq Var$ is enough
- unique decomposition: still exactly one fully decomposed form!

Base

Base

- an exponential description of bisimilarity

Base

- an exponential description of bisimilarity
- for every $R \subseteq \mathrm{Var}$:

Base

- an exponential description of bisimilarity
- for every R ⊆ Var:
  - set of R-primes, R-decomposables and decompositions for them

Base

- an exponential description of bisimilarity
- for every $R \subseteq Var$:
  - set of R-primes, R-decomposables and decompositions for them
  - for every $X \in Var$ a rule $R \xrightarrow{X} R'$ $(RED(\alpha) = R \Rightarrow RED(X\alpha) = R')$

Algorithm

Algorithm

- guess a base (exponential)

Algorithm

- guess a base (exponential)
- verify it (nontrivial):

Algorithm

- guess a base (exponential)
- verify it (nontrivial):
  - check that defined relation is indeed a branching bisimulation

Algorithm

- guess a base (exponential)
- verify it (nontrivial):
  - check that defined relation is indeed a branching bisimulation
  - possible due to a new, elegant class change norm (due to Fu)

Algorithm

- guess a base (exponential)
- verify it (nontrivial):
  - check that defined relation is indeed a branching bisimulation
  - possible due to a new, elegant class change norm (due to Fu)
  - check only not too big responses

Class change norm

Class change norm

- how many class changes needed to reach ε

Class change norm

- how many class changes needed to reach ε
- $\alpha \sim \beta \Rightarrow$ cc-norm(α) = cc-norm(β)

Class change norm

- how many class changes needed to reach ε
- $\alpha \sim \beta \Rightarrow$ cc-norm(α) = cc-norm(β)
- cc-norm does not change too much $\Rightarrow$ short responses are enough

Class change norm

- how many class changes needed to reach ε
- $\alpha \sim \beta \Rightarrow$ cc-norm(α) = cc-norm(β)
- cc-norm does not change too much ⇒ short responses are enough
- correctness possible to verify

Thank you!