Branching Bisimilarity on Normed BPA Processes is in \text{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

\[ \text{a} \]
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

\[ X \xrightarrow{a} \alpha \text{ in the automaton} \]
BPA Processes

BPA = stateless pushdown automaton

\[ X \xrightarrow{a} \alpha \text{ in the automaton} \]

\[ X\beta \xrightarrow{a} \alpha\beta \text{ in the multigraph} \]
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 $\alpha$ and $\beta$
Problem

Given: normed BPA, configurations $\alpha$ and $\beta$

Question: is $\alpha$ branching bisimilar to $\beta$?
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 \((\alpha, \beta)\) 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!

\[ A \xrightarrow{a} A \quad \text{and} \quad A \xrightarrow{a} \varepsilon \]
For branching bisimulation

unique decomposition does not work!

\[
\begin{align*}
A \xrightarrow{a} A & \quad & A \xrightarrow{a} \varepsilon & \quad & A \xrightarrow{} \varepsilon
\end{align*}
\]
For branching bisimulation

unique decomposition does not work!

$A \xrightarrow{a} A \quad A \xrightarrow{a} \epsilon \quad A \rightarrow \epsilon$

$A \sim AA$
For branching bisimulation

unique decomposition does not work!

\[
\begin{align*}
A & \xrightarrow{a} A \\
A & \xrightarrow{a} \varepsilon \\
A & \xrightarrow{} \varepsilon \\
A & \sim AA \\
A & \not\sim \varepsilon
\end{align*}
\]
For branching bisimulation

unique decomposition does not work!

\[ A \xrightarrow{a} A \quad A \xrightarrow{a} \varepsilon \quad A \xrightarrow{} \varepsilon \]

\[ A \sim AA \quad A \not\sim \varepsilon \]

\[ \text{pf}(A) = \text{pf}(A) \quad \text{pf}(A) \Rightarrow \text{pf}(A) = \varepsilon \]
For branching bisimulation

unique decomposition does not work!

\[ A \xrightarrow{a} A \quad A \xrightarrow{a} \varepsilon \quad A \rightarrow \varepsilon \]

\[ A \sim AA \quad A \not\sim \varepsilon \]

\[ \text{pf}(A) = \text{pf}(A) \quad \text{pf}(A) \Rightarrow \text{pf}(A) = \varepsilon \]

some more ideas needed!
Redundant variables
Redundant variables

\[ \text{RED}(\alpha) = \{X : X\alpha \sim \alpha\} \]
Redundant variables

\[ \text{RED}(\alpha) = \{X: X\alpha \sim \alpha\} \]

redundant variables provide full information about the suffix:
Redundant variables

\[ RED(\alpha) = \{ X : X\alpha \sim \alpha \} \]

redundant variables provide full information about the suffix:

\[ RED(\alpha) = RED(\beta) \Rightarrow (\gamma\alpha \sim \delta\alpha \Leftrightarrow \gamma\beta \sim \delta\beta) \]
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 \( R \subseteq \text{Var} \) is enough
Main technical contribution

- relative decomposition works!
- decomposition depends on the suffix
- concretely: on the RED(suffix)
- one decomposition system for every $R \subseteq \text{Var}$ 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 \( R \subseteq \text{Var} \):
Base

• an exponential description of bisimilarity

• for every $R \subseteq \text{Var}$:

  • set of $R$-primes, $R$-decomposables and decompositions for them
Base

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

• how many class changes needed to reach $\varepsilon$

• $\alpha \sim \beta \Rightarrow \text{cc-norm}(\alpha) = \text{cc-norm}(\beta)$
Class change norm

- how many class changes needed to reach $\varepsilon$
- $\alpha \sim \beta \Rightarrow \text{cc-norm}(\alpha) = \text{cc-norm}(\beta)$
- cc-norm does not change too much $\Rightarrow$ short responses are enough
Class change norm

- how many class changes needed to reach $\varepsilon$
- $\alpha \sim \beta \Rightarrow \text{cc-norm}(\alpha) = \text{cc-norm}(\beta)$
- cc-norm does not change too much $\Rightarrow$
  short responses are enough
- correctness possible to verify
Thank you!