Fast bisimulation-checking for normed context-free processes

BASICS summer school, 19-23.08.2013

Fast bisimulation-checking for normed context-free processes

Sławomir Lasota
University of Warsaw

BASICS summer school, 19-23.08.2013

Fast bisimulation-checking for normed context-free processes

Sławomir Lasota
University of Warsaw

joint work with Wojciech Czerwiński and Sibylle Froeschle

BASICS summer school, 19-23.08.2013

- W. Czerwiński, *Partially-commutative context-free graphs*, PhD Thesis, University of Warsaw, 2012.

- W. Czerwiński, S. Froeschle, S. Lasota, *Partially-commutative context-free processes: expressibility and tractability,* Information and Computation, 2011.

- W. Czerwiński, S. Lasota, *Fast equivalence-checking for normed context-free processes,* FSTTCS 2010.

- W. Czerwiński, S. Froeschle, S. Lasota, *Partially-commutative context-free processes,* CONCUR 2009.

Outline

- Background
    - Context-free graphs and commutative context-free graphs
    - Bisimulation equivalence problem
    - Norm
    - History of the problem
    - Unique decomposition
    - Naive algorithm
- Efficient algorithm for BPA and BPP
    - Outline of the algorithm
    - Refinement
    - Efficient computation of refinement for BPA
    - Time-cost analysis
    - Partially-commutative context-free graphs

Outline

- Background
  - **Context-free graphs and commutative context-free graphs**
  - Bisimulation equivalence problem
  - Norm
  - History of the problem
  - Unique decomposition
  - Naive algorithm
- Efficient algorithm for BPA and BPP
  - Outline of the algorithm
  - Refinement
  - Efficient computation of refinement for BPA
  - Time-cost analysis
  - Partially-commutative context-free graphs

Context-free grammars (CFG)

Context-free grammars (CFG)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

Context-free grammars (CFG)

alphabet letters (terminal symbols)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

Context-free grammars (CFG)

alphabet letters (terminal symbols)

variables (non-terminal symbols)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

Context-free grammars (CFG)

alphabet letters (terminal symbols)

variables (non-terminal symbols)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

transition rules (productions)

$$S \longrightarrow a\,S\,b \qquad S \longrightarrow s$$

Context-free grammars (CFG)

alphabet letters (terminal symbols)

variables (non-terminal symbols)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

transition rules (productions)

$$S \longrightarrow a\,S\,b \qquad S \longrightarrow s$$

rewrite rules

Context-free grammars (CFG)

alphabet letters (terminal symbols)

variables (non-terminal symbols)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

transition rules (productions)

$$S \longrightarrow a\,S\,b \qquad S \longrightarrow s$$

rewrite rules

$$L = \{a^n\, s\, b^n : n \in \mathbb{N}\}$$

Context-free grammars (CFG)

alphabet letters (terminal symbols)

variables (non-terminal symbols)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

transition rules (productions)

$$S \longrightarrow a\,S\,b \qquad S \longrightarrow s$$

rewrite rules

$$L = \{a^n\, s\, b^n : n \in \mathbb{N}\}$$

Graibach normal form:

Context-free grammars (CFG)

alphabet letters (terminal symbols)

variables (non-terminal symbols)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

transition rules (productions)

$$S \longrightarrow a\,S\,b \qquad S \longrightarrow s$$

rewrite rules

$$L = \{a^n\, s\, b^n : n \in \mathbb{N}\}$$

Graibach normal form:

$$S \longrightarrow a\,S\,B \qquad S \longrightarrow s \qquad B \longrightarrow b$$

Context-free grammars (CFG)

alphabet letters (terminal symbols)

variables (non-terminal symbols)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

transition rules (productions)

$$S \longrightarrow a\,S\,b \qquad S \longrightarrow s$$

rewrite rules

$$L = \{a^n\, s\, b^n : n \in \mathbb{N}\}$$

Graibach normal form:

$$S \longrightarrow a\,S\,B \qquad S \longrightarrow s \qquad B \longrightarrow b$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

Context-free grammars (CFG)

alphabet letters (terminal symbols)

variables (non-terminal symbols)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

transition rules (productions)

$$S \longrightarrow a\, S\, b \qquad S \longrightarrow s$$

rewrite rules

$$L = \{a^n\, s\, b^n : n \in \mathbb{N}\}$$

stateless pushdown automata
without ε-transitions

Graibach normal form:

$$S \longrightarrow a\, S\, B \qquad S \longrightarrow s \qquad B \longrightarrow b$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

Context-free graphs

Basic Process Algebra (BPA)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

Context-free graphs

Basic Process Algebra (BPA)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

- processes (configurations) = words over V

Context-free graphs

Basic Process Algebra (BPA)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

- processes (configurations) = words over V $\qquad \alpha \in V^*$

Context-free graphs

Basic Process Algebra (BPA)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

- processes (configurations) = words over V $\qquad \alpha \in V^*$
- prefix rewriting

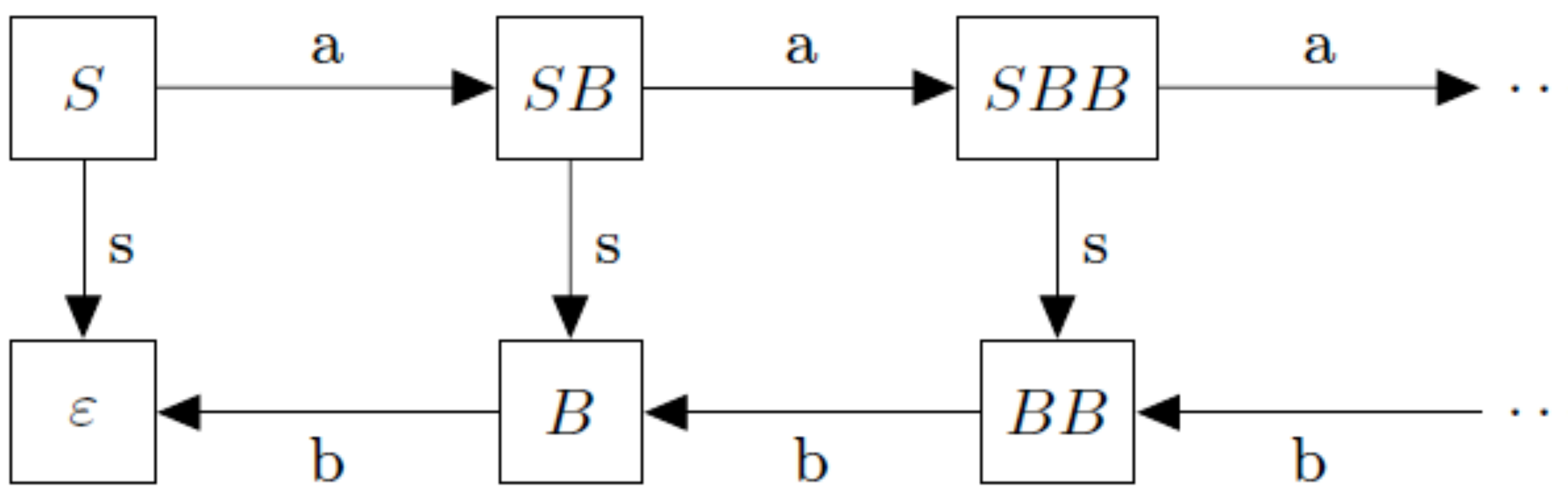
Context-free graphs

Basic Process Algebra (BPA)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

- processes (configurations) = words over V $\qquad \alpha \in V^*$
- prefix rewriting



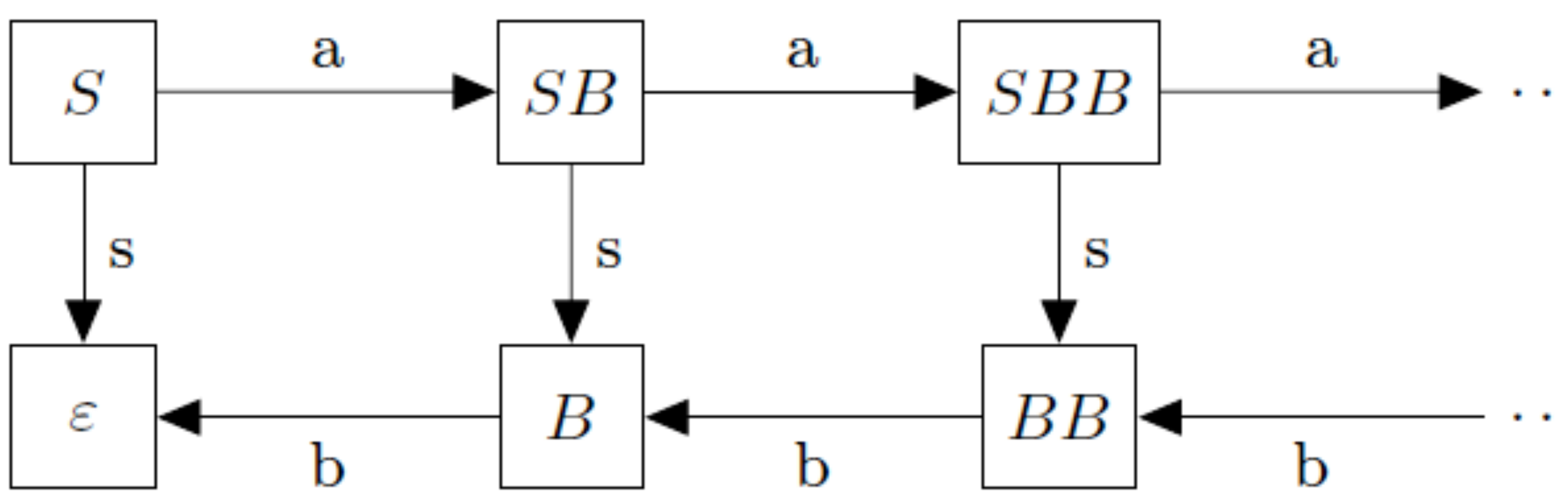
Context-free graphs

Basic Process Algebra (BPA)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

- processes (configurations) = words over V $\quad \alpha \in V^*$
- prefix rewriting



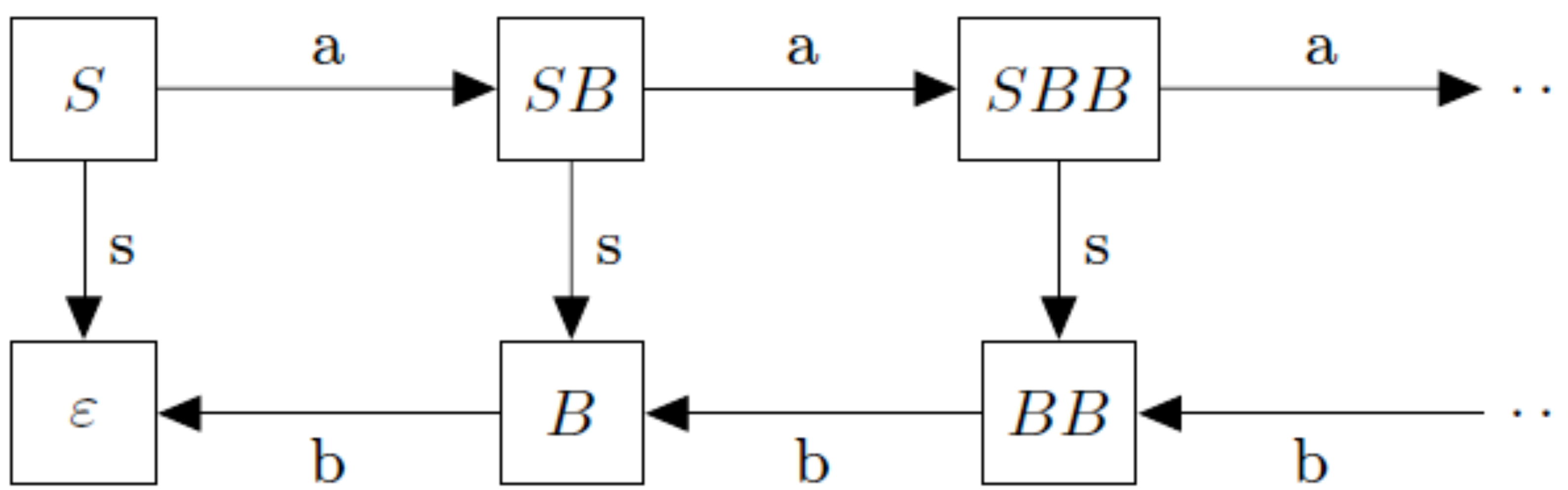
the simplest model of recursive programs

Deterministic context-free graphs

Simple grammars

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$



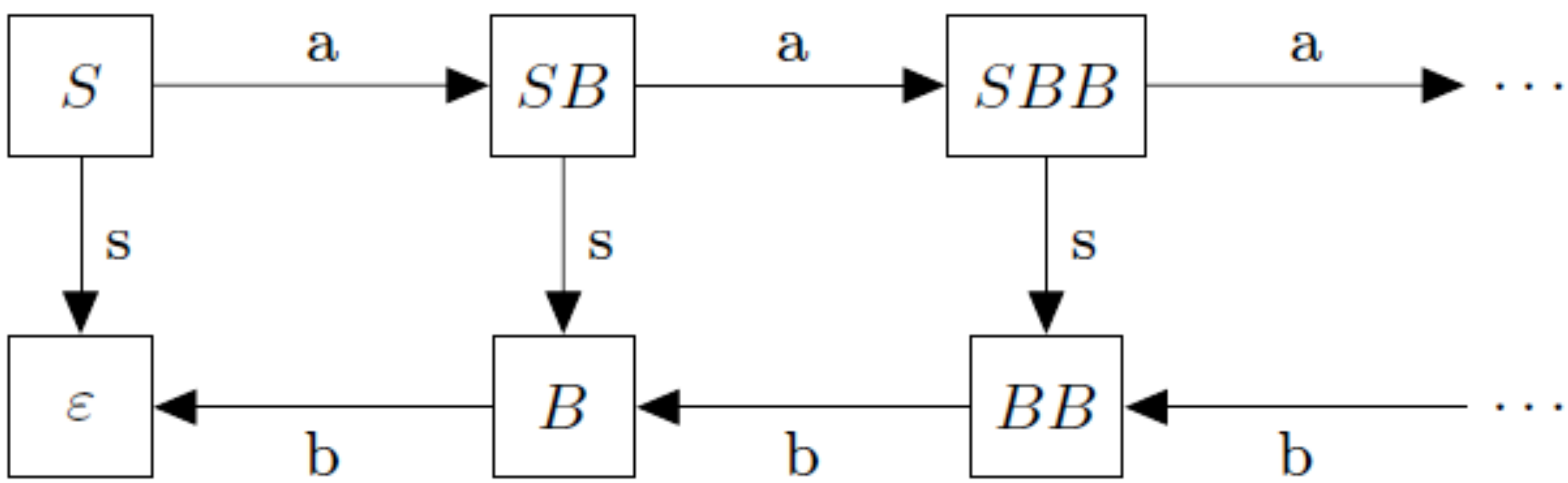
Deterministic context-free graphs

Simple grammars

$\Sigma = \{a, b\} \qquad V = \{B, S\}$

for every alphabet letter *a*,
every variable has at most one *a*-rule

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$



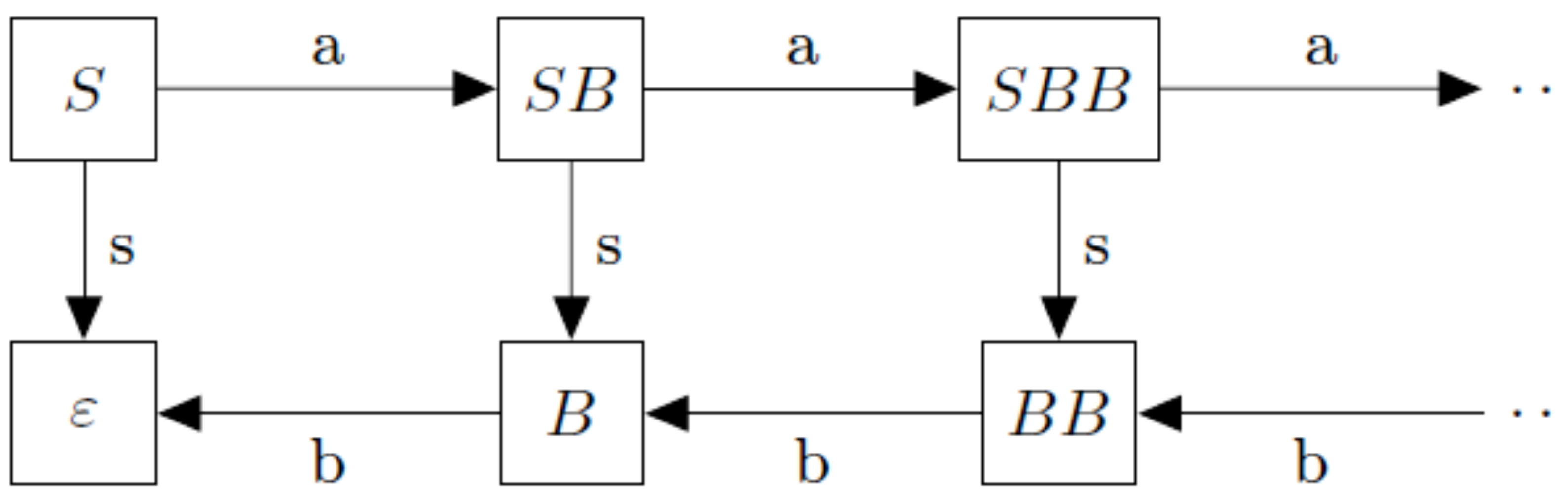
Deterministic context-free graphs

Simple grammars

$\Sigma = \{a, b\}$ $\quad V = \{B, S\}$

for every alphabet letter *a*, every variable has at most one *a*-rule

$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$



determinism: for every alphabet letter *a*, every process has at most one *a*-successor

Commutative context-free graphs

Basic Parallel Processes (BPP)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

Commutative context-free graphs

Basic Parallel Processes (BPP)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

- processes (configurations) = multisets over V

Commutative context-free graphs

Basic Parallel Processes (BPP)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

- processes (configurations) = multisets over V $\quad \alpha \in V^*$

Commutative context-free graphs

Basic Parallel Processes (BPP)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

- processes (configurations) = multisets over V $\qquad \alpha \in V^*$
- unrestricted rewriting

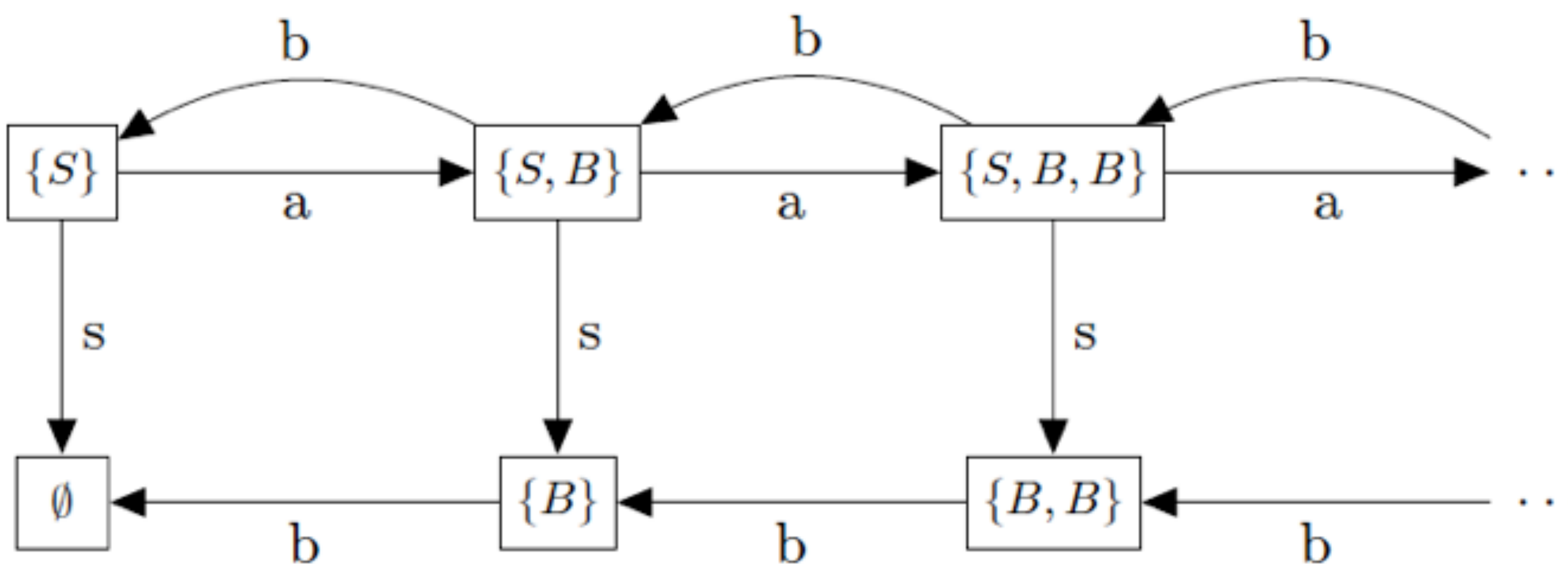
Commutative context-free graphs

Basic Parallel Processes (BPP)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

- processes (configurations) = multisets over V $\qquad \alpha \in V^*$
- unrestricted rewriting



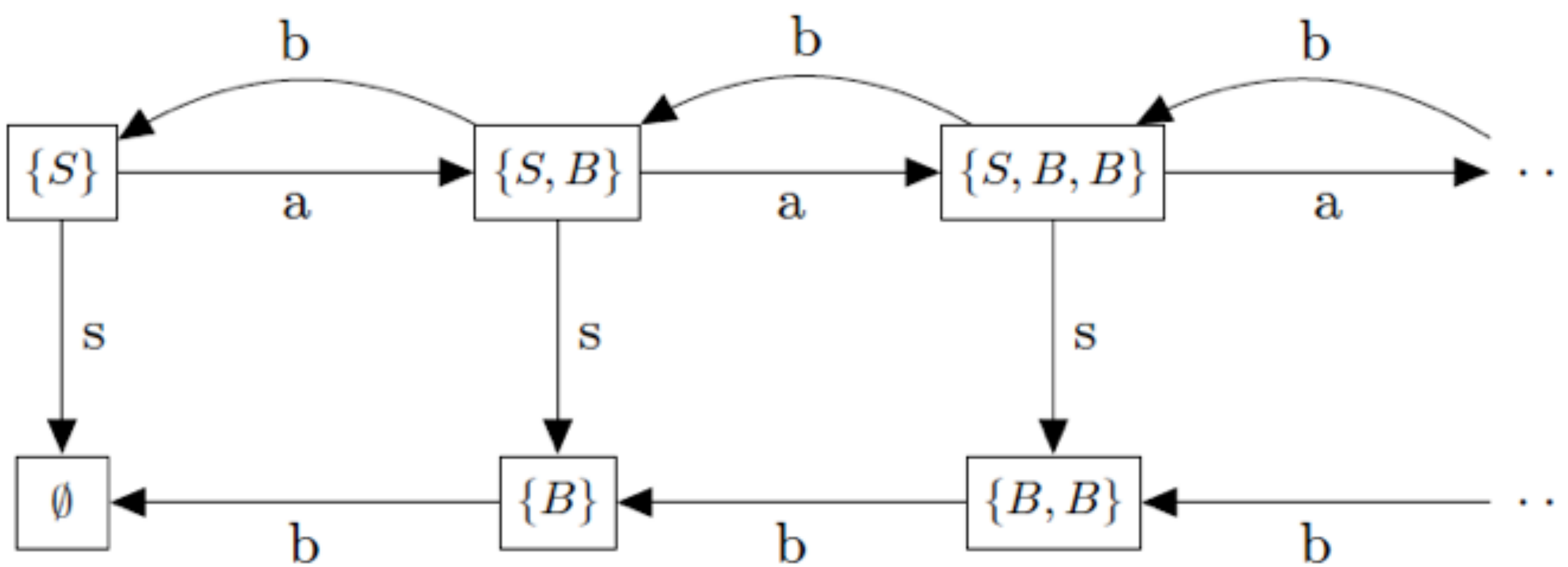
Commutative context-free graphs

Basic Parallel Processes (BPP)

$$\Sigma = \{a, b\} \qquad V = \{B, S\}$$

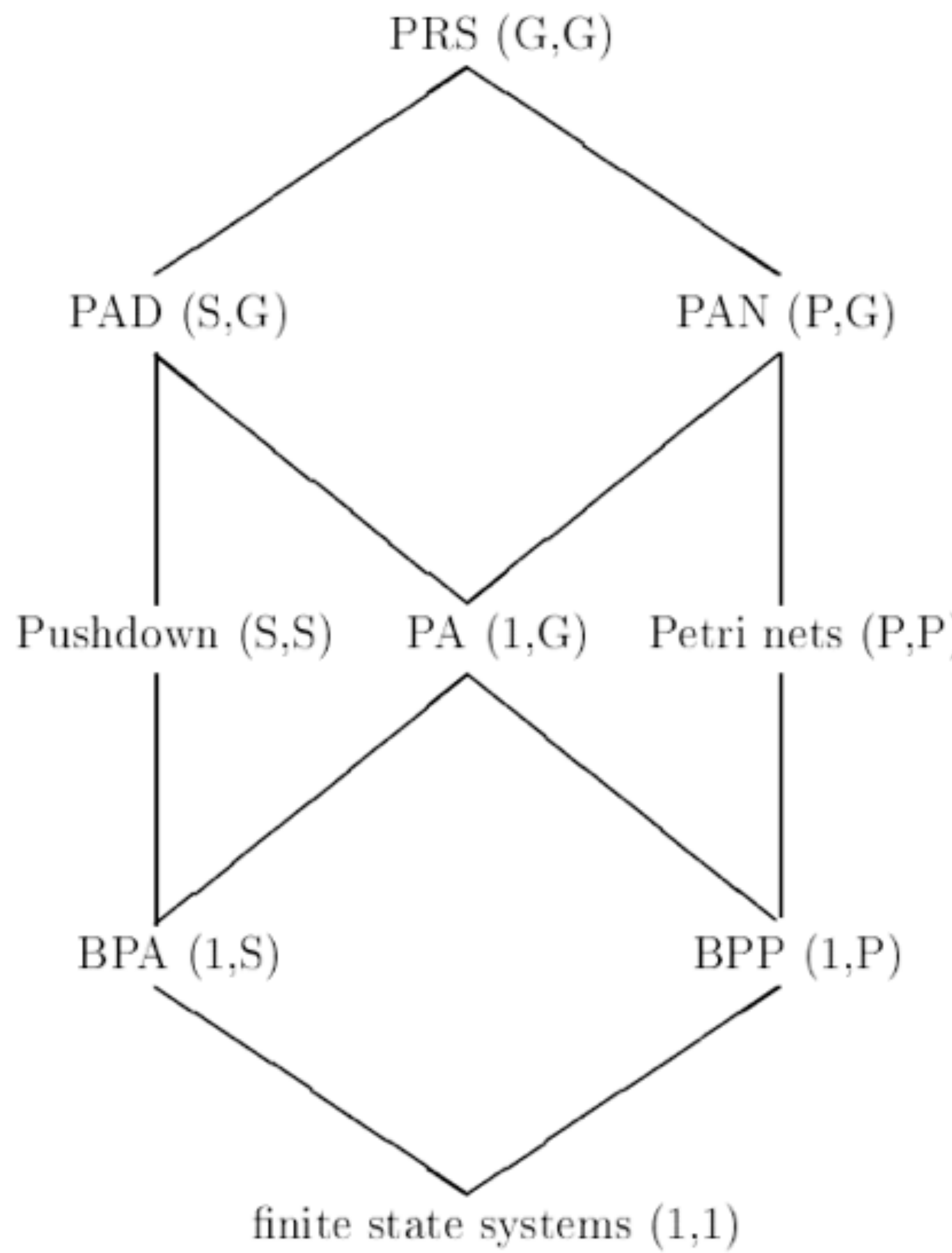
$$S \xrightarrow{a} SB \qquad S \xrightarrow{s} \varepsilon \qquad B \xrightarrow{b} \varepsilon.$$

- processes (configurations) = multisets over V $\qquad \alpha \in V^*$
- unrestricted rewriting

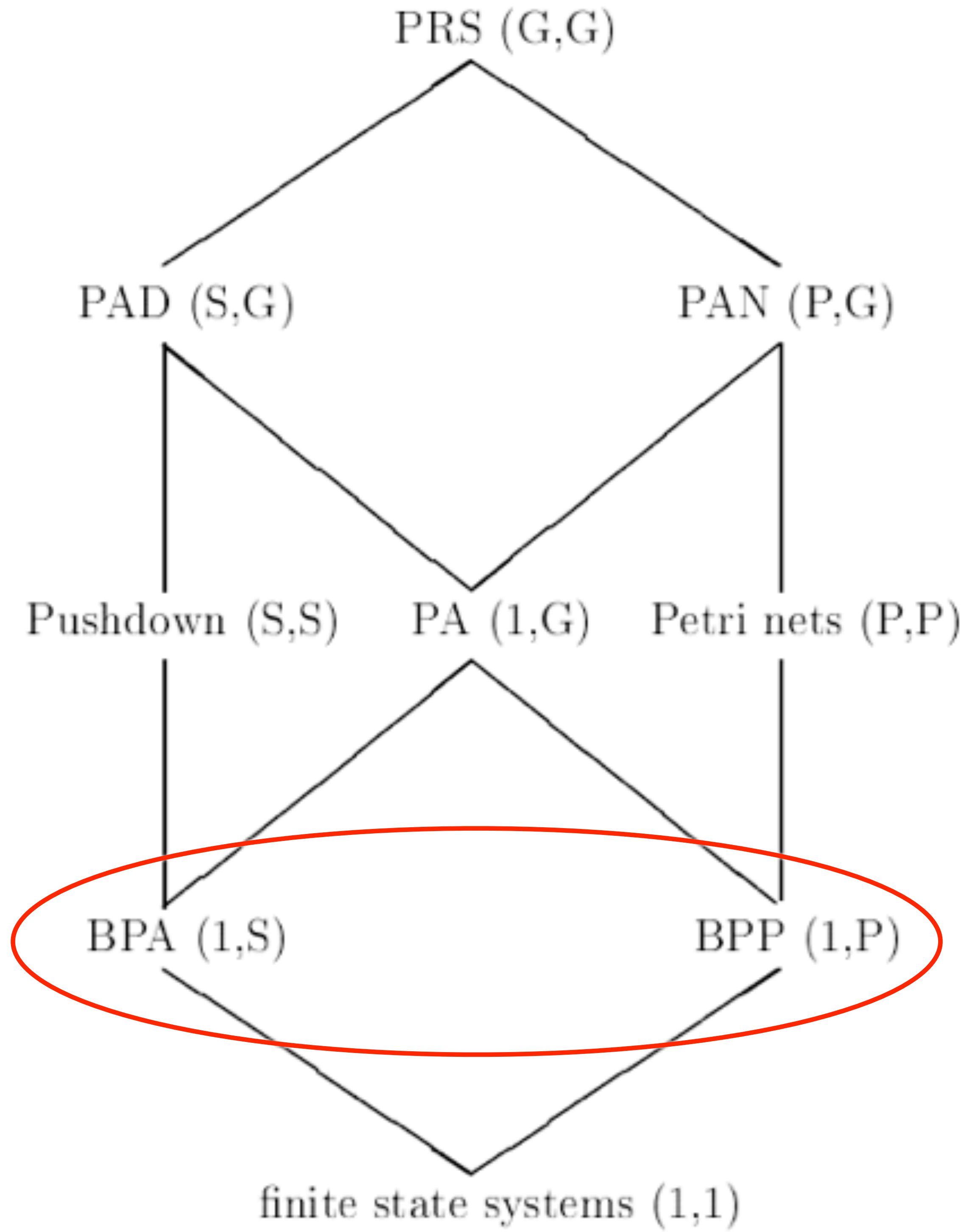


the simplest model of parallel programs

PRS hierarchy



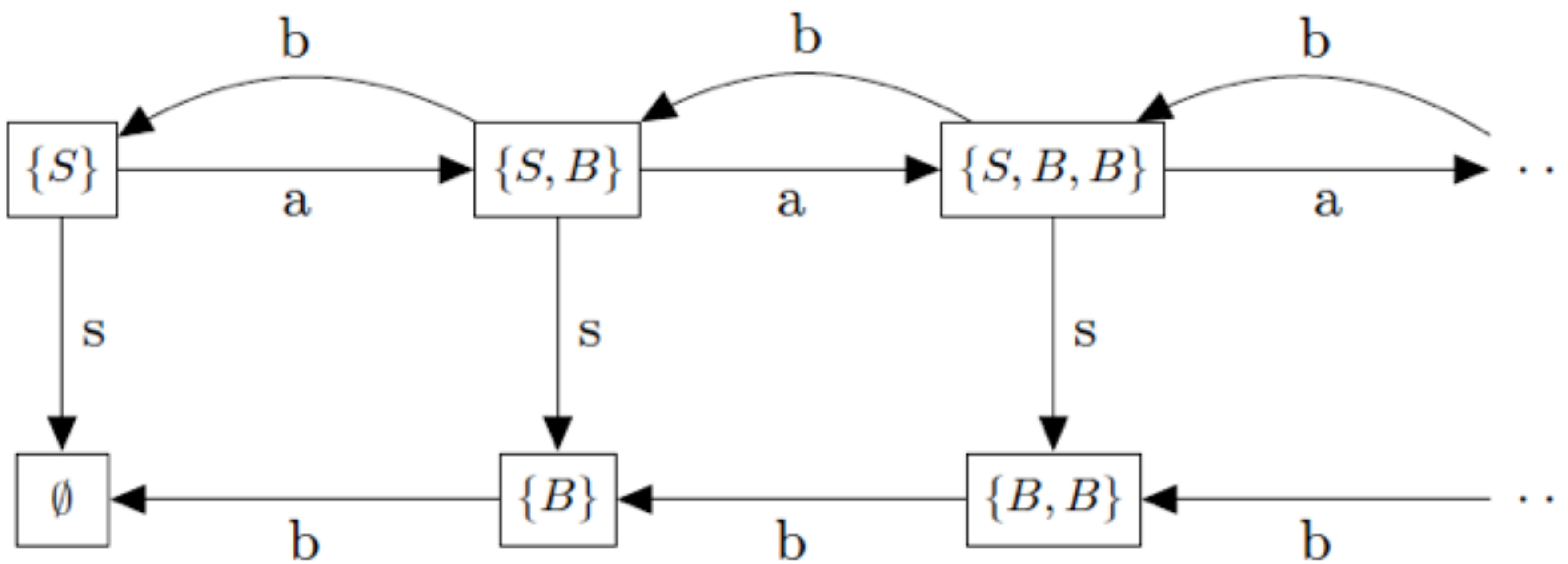
PRS hierarchy



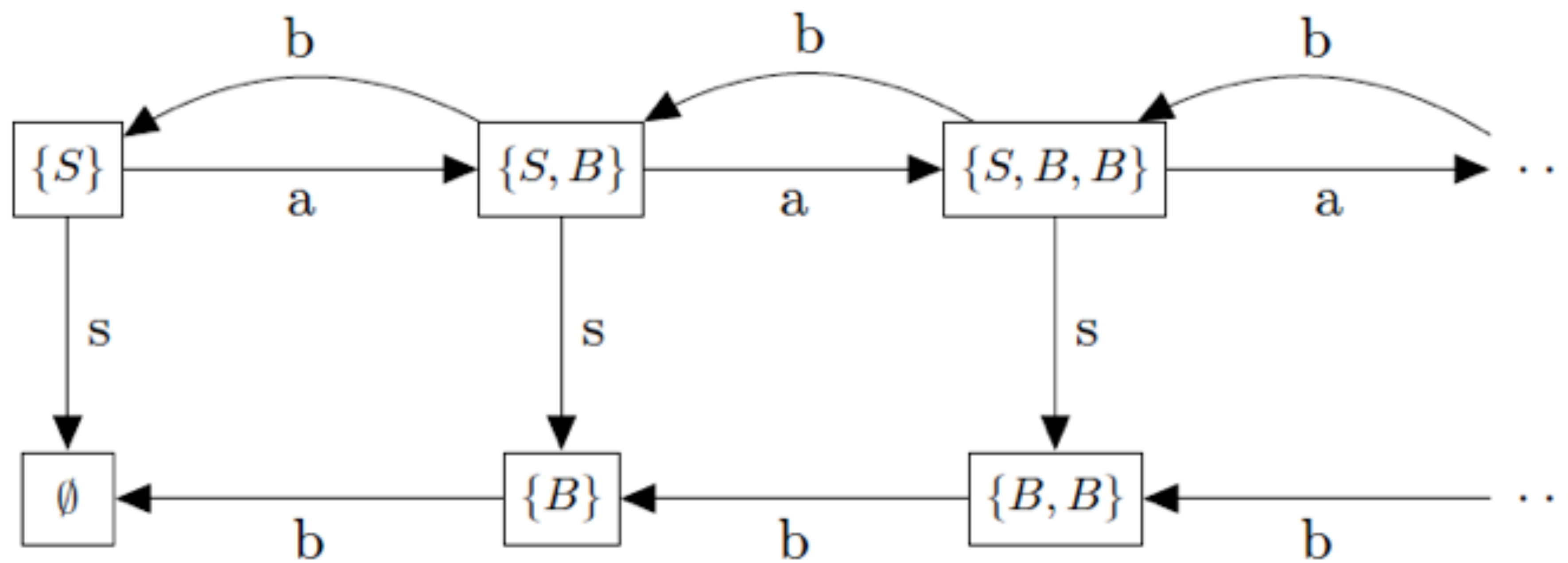
Outline

- Background
    - Context-free graphs and commutative context-free graphs
    - **Bisimulation equivalence problem**
    - Norm
    - History of the problem
    - Unique decomposition
    - Naive algorithm
- Efficient algorithm for BPA and BPP
    - Outline of the algorithm
    - Refinement
    - Efficient computation of refinement for BPA
    - Time-cost analysis
    - Partially-commutative context-free graphs

Bisimulation Game

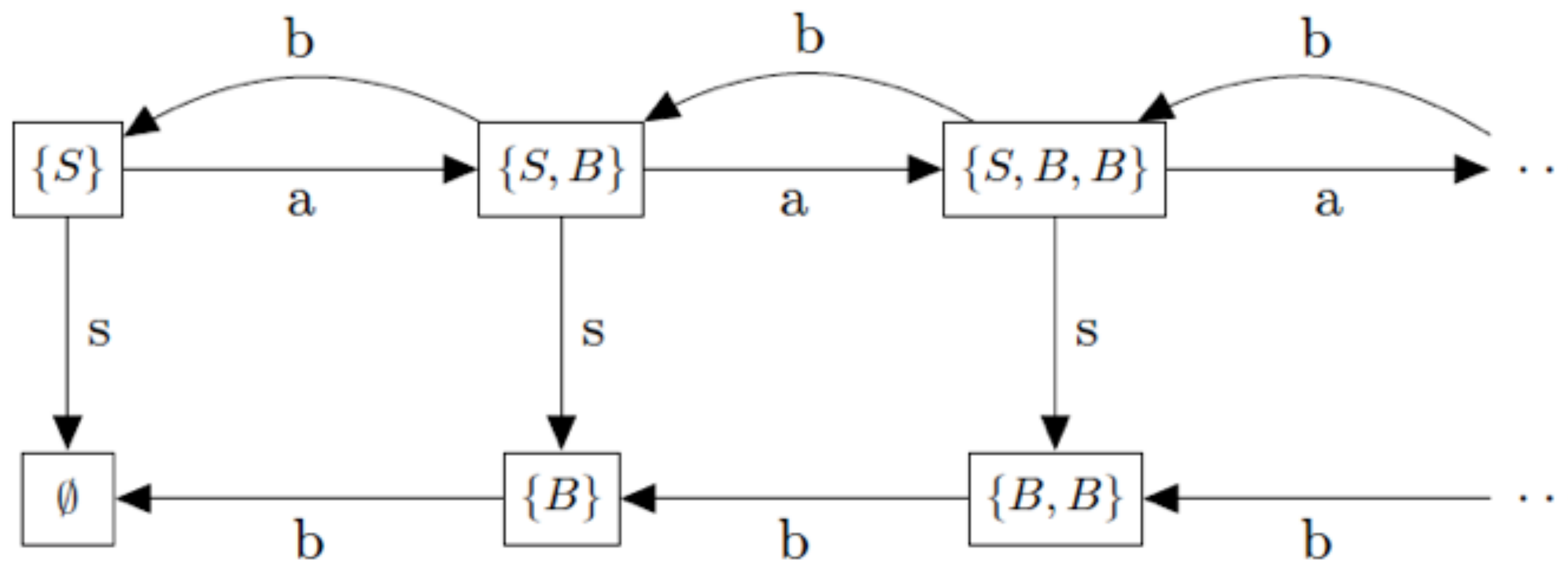


Bisimulation Game



Two-player game (between Spoiler and Duplicator) proceeding in rounds

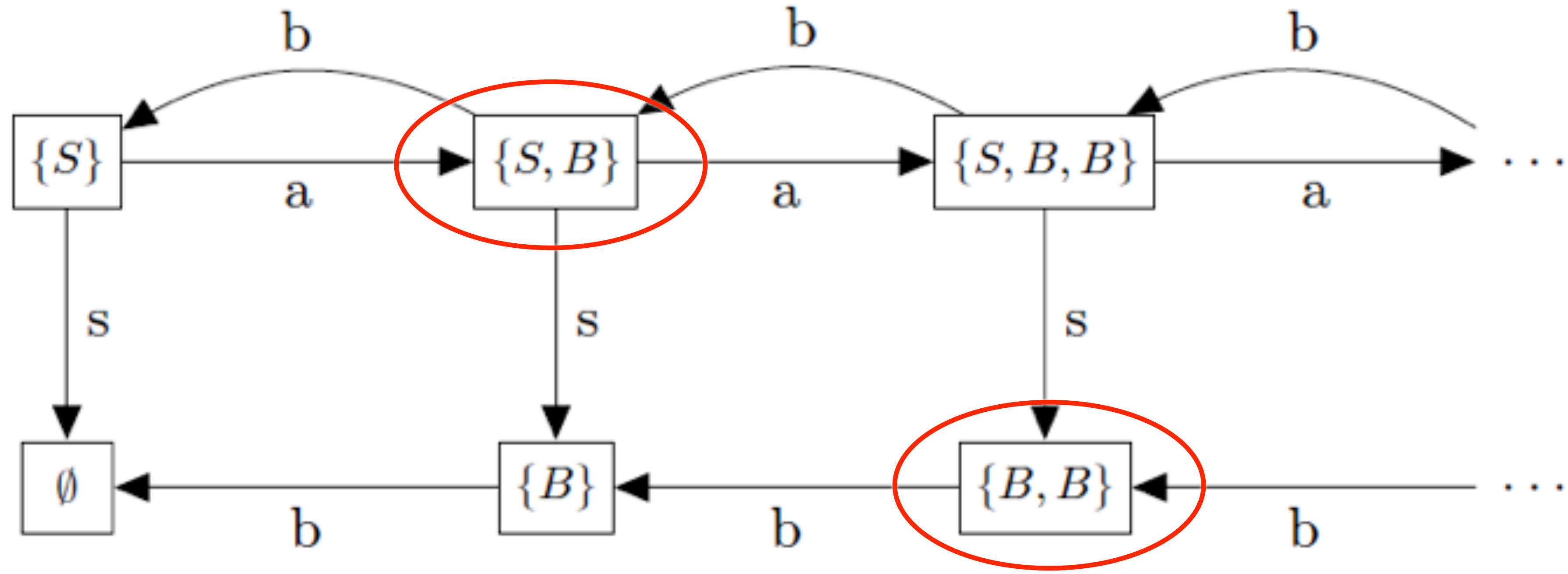
Bisimulation Game



Two-player game (between Spoiler and Duplicator) proceeding in rounds

Configurations = pairs of processes $(\alpha, \beta) \in V^* \times V^*$

Bisimulation Game

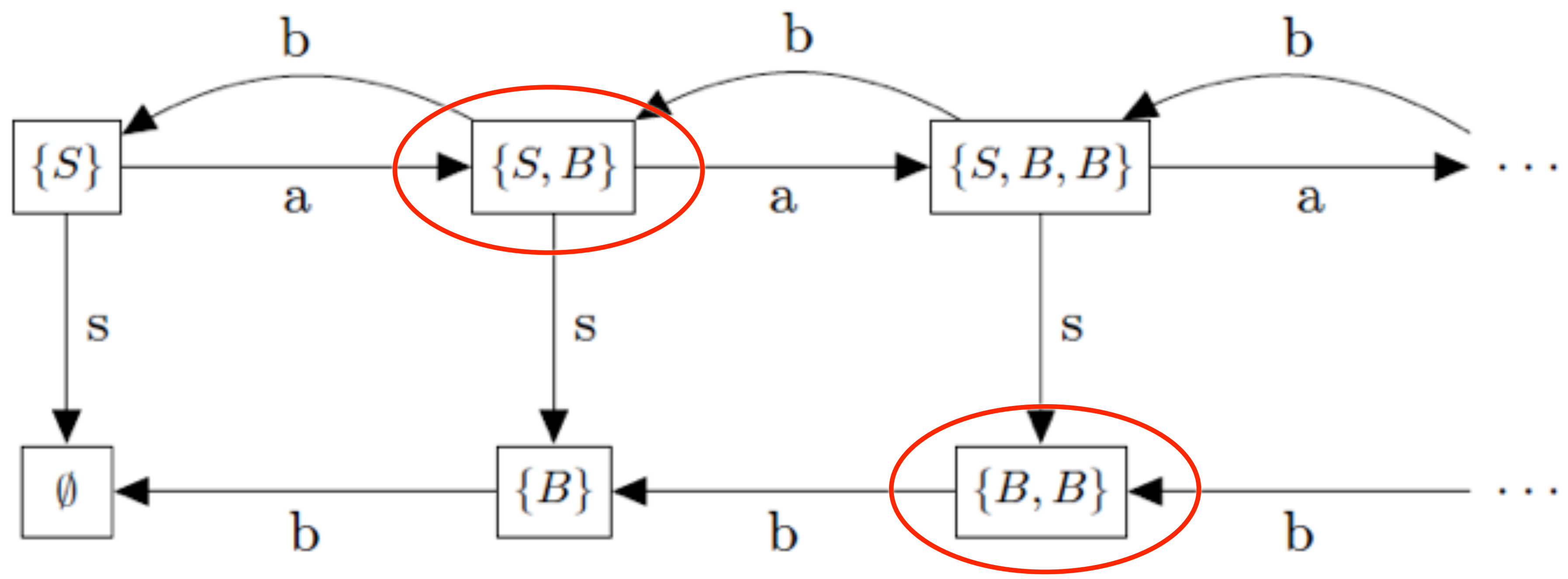


Two-player game (between Spoiler and Duplicator) proceeding in rounds

Configurations = pairs of processes $(\alpha, \beta) \in V^* \times V^*$

Bisimulation Game

Spoiler's move

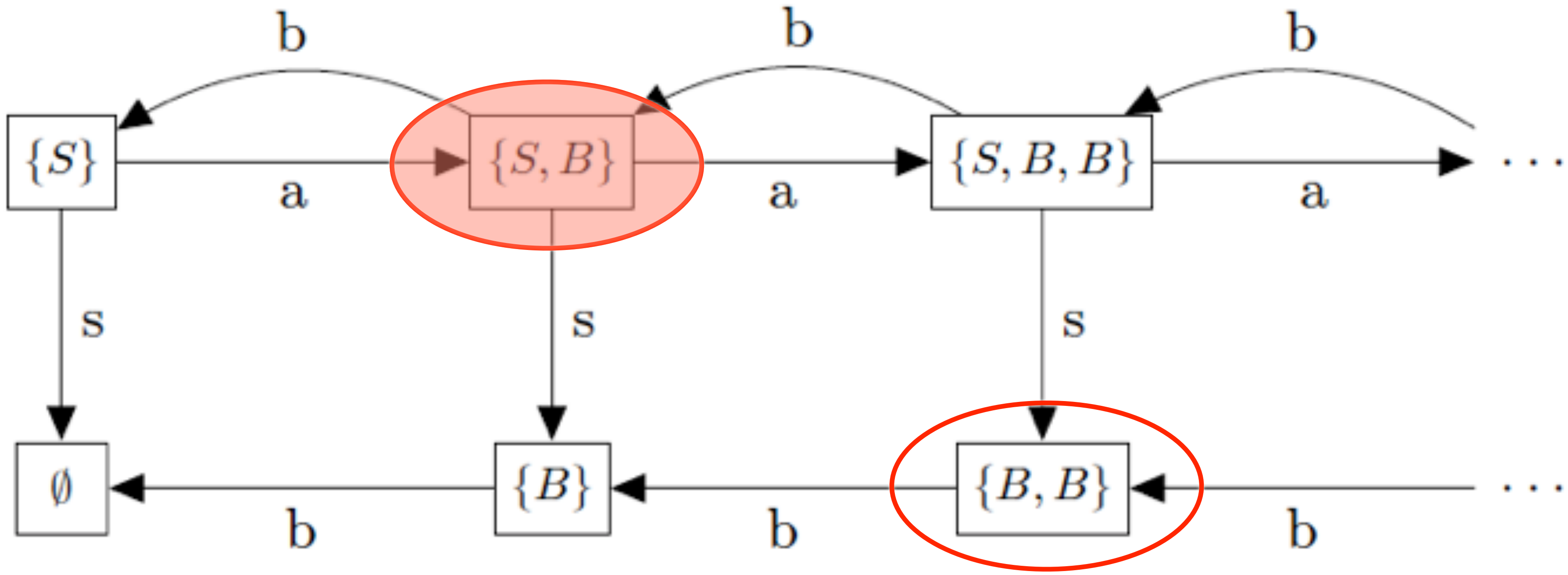


Two-player game (between Spoiler and Duplicator) proceeding in rounds

Configurations = pairs of processes $(\alpha, \beta) \in V^* \times V^*$

Bisimulation Game

Spoiler's move

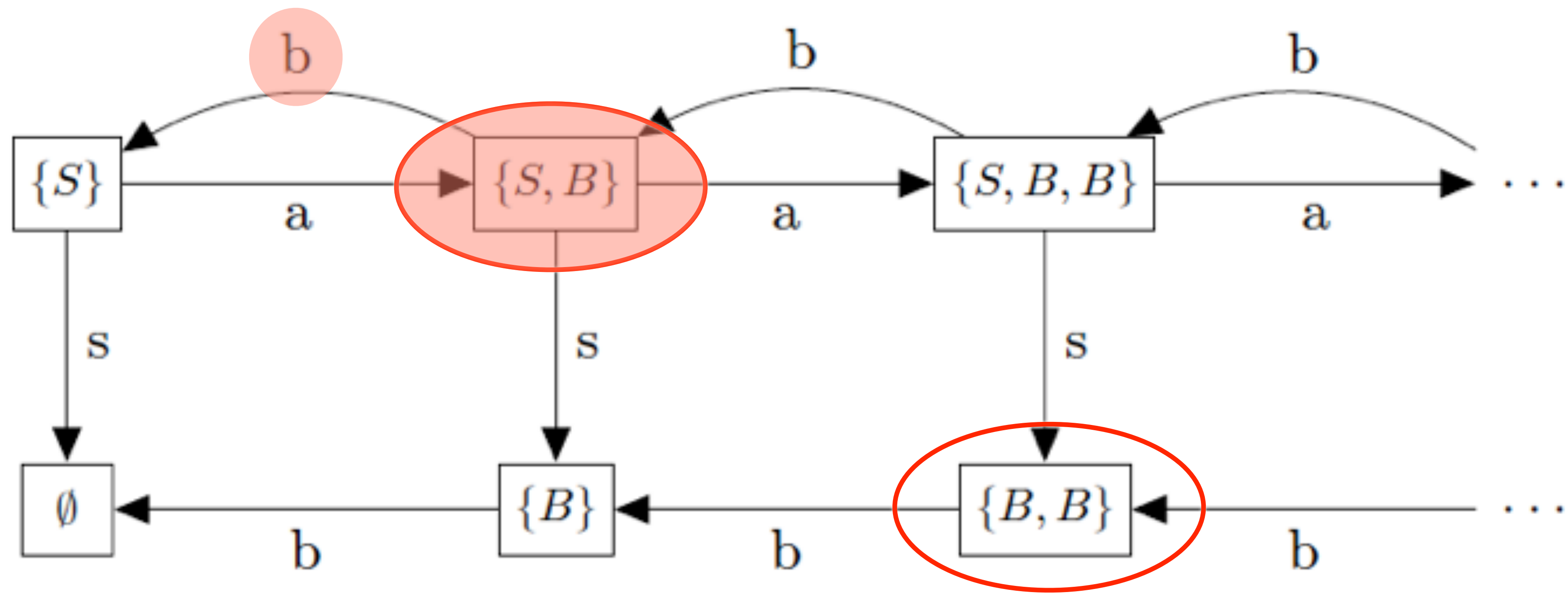


Two-player game (between Spoiler and Duplicator) proceeding in rounds

Configurations = pairs of processes $(\alpha, \beta) \in V^* \times V^*$

Bisimulation Game

Spoiler's move

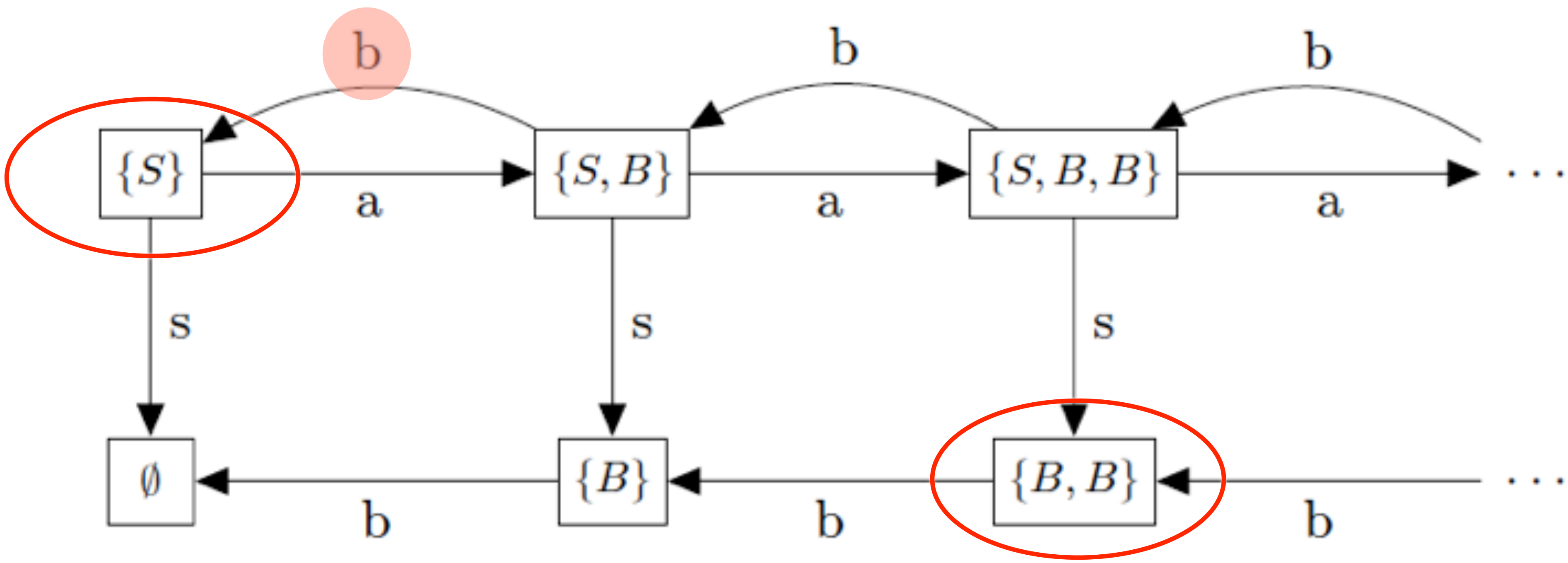


Two-player game (between Spoiler and Duplicator) proceeding in rounds

Configurations = pairs of processes $(\alpha, \beta) \in V^* \times V^*$

Bisimulation Game

Spoiler's move

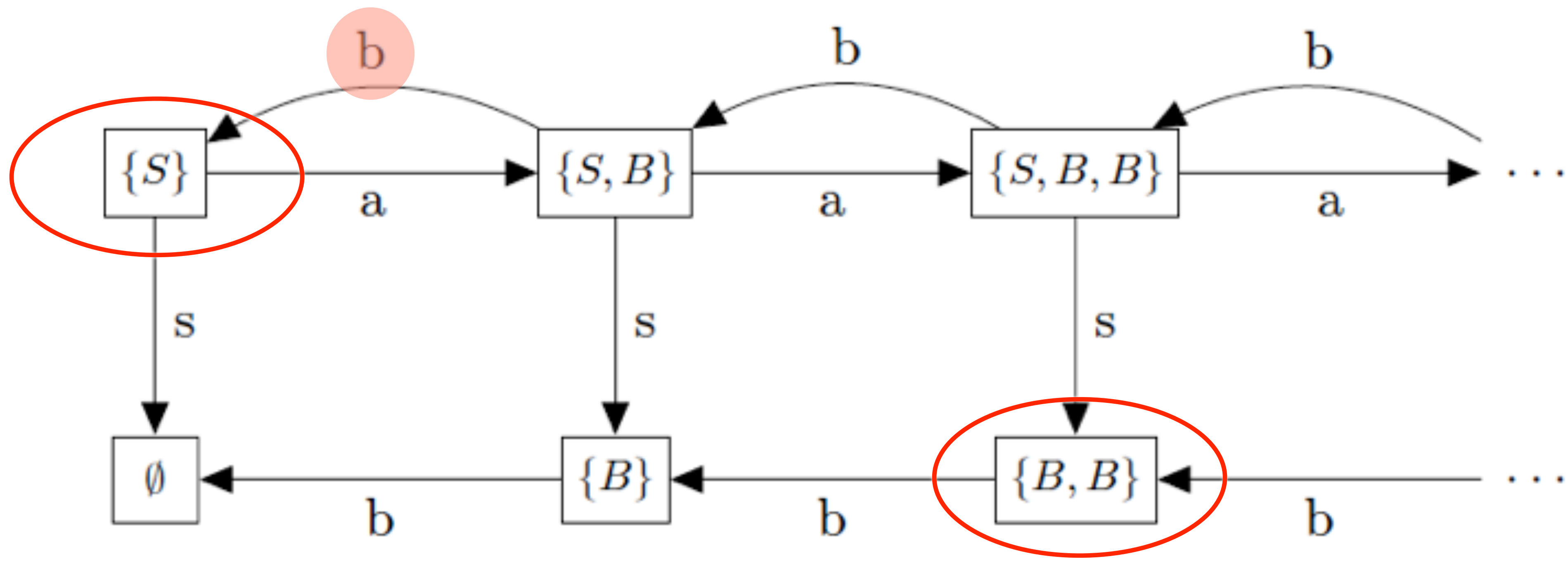


Two-player game (between Spoiler and Duplicator) proceeding in rounds

Configurations = pairs of processes $(\alpha, \beta) \in V^* \times V^*$

Bisimulation Game

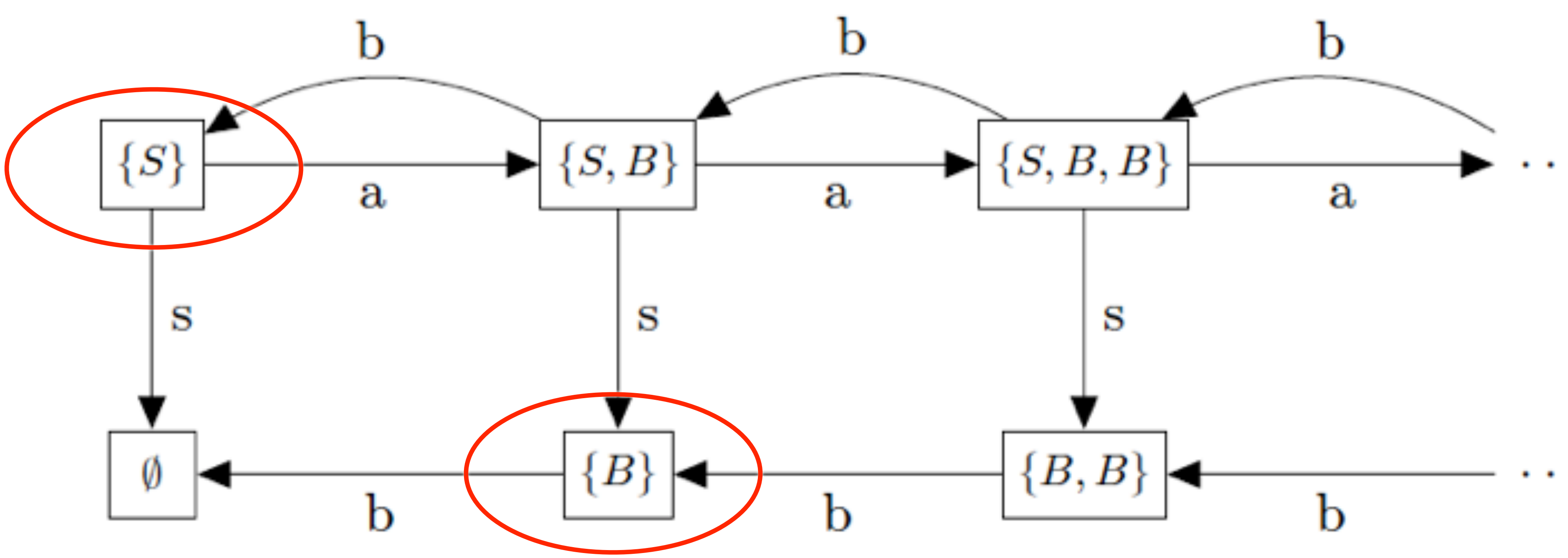
Duplicator's move



Two-player game (between Spoiler and Duplicator) proceeding in rounds

Configurations = pairs of processes $(\alpha, \beta) \in V^* \times V^*$

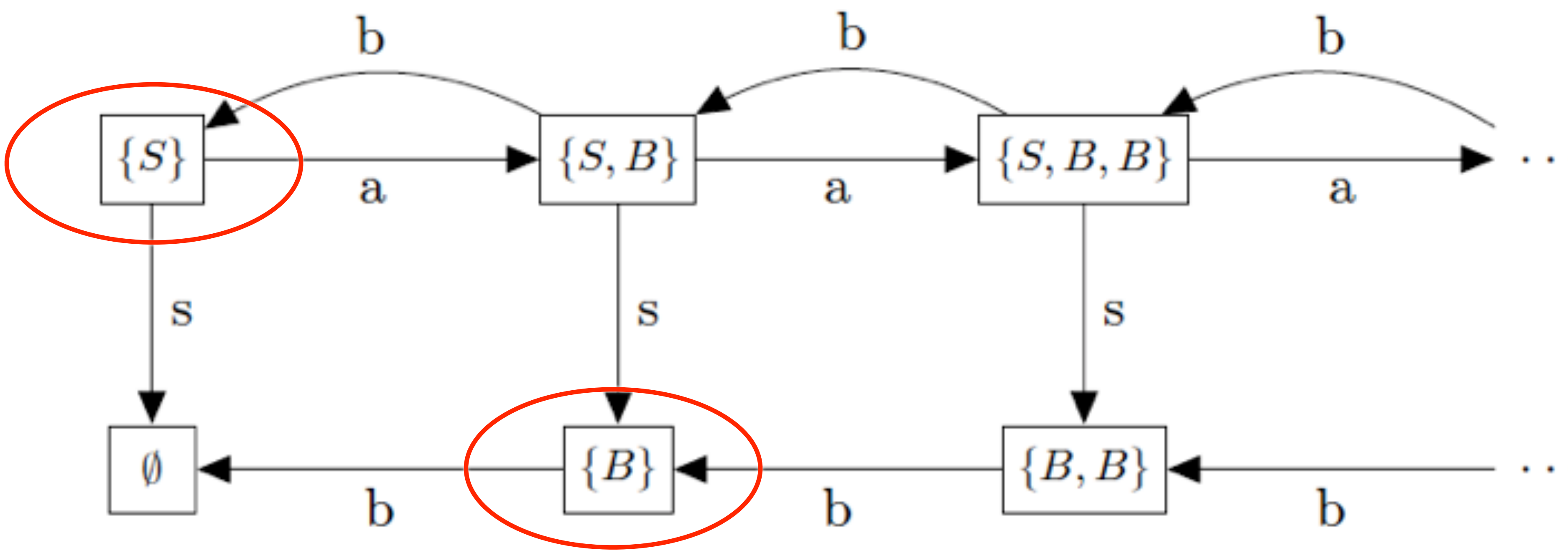
Bisimulation Game



Two-player game (between Spoiler and Duplicator) proceeding in rounds

Configurations = pairs of processes $(\alpha, \beta) \in V^* \times V^*$

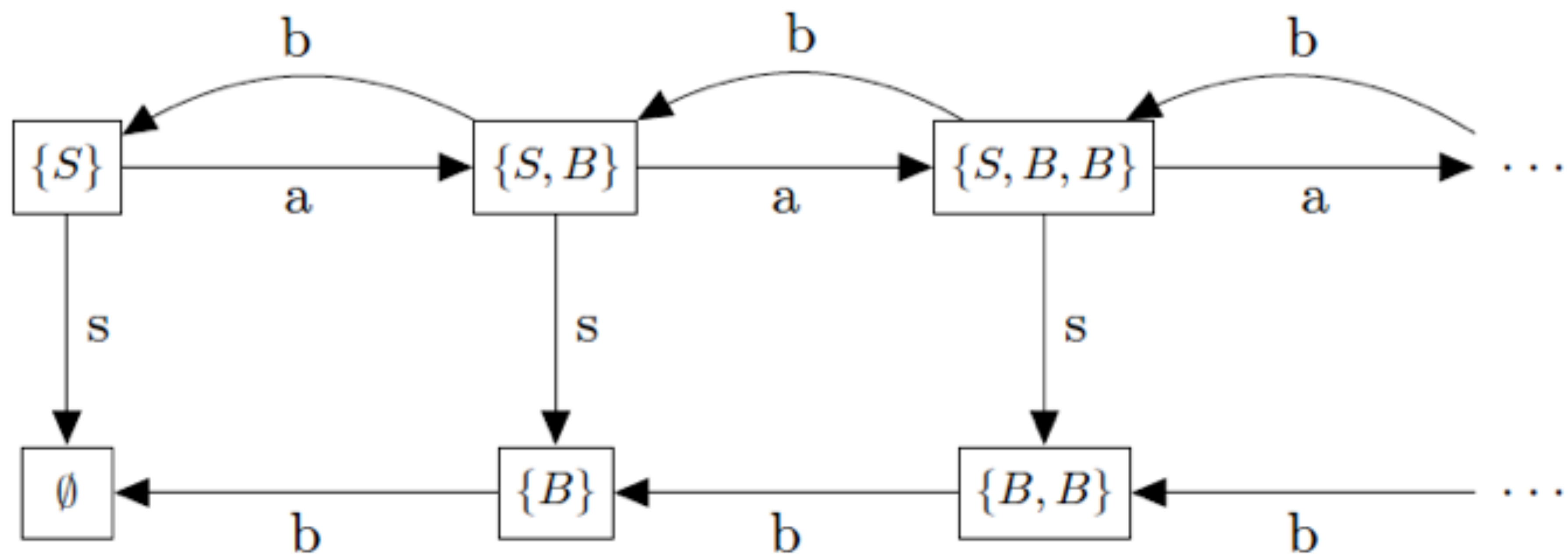
Bisimulation Game



Two-player game (between Spoiler and Duplicator) proceeding in rounds

Configurations = pairs of processes $(\alpha, \beta) \in V^* \times V^*$

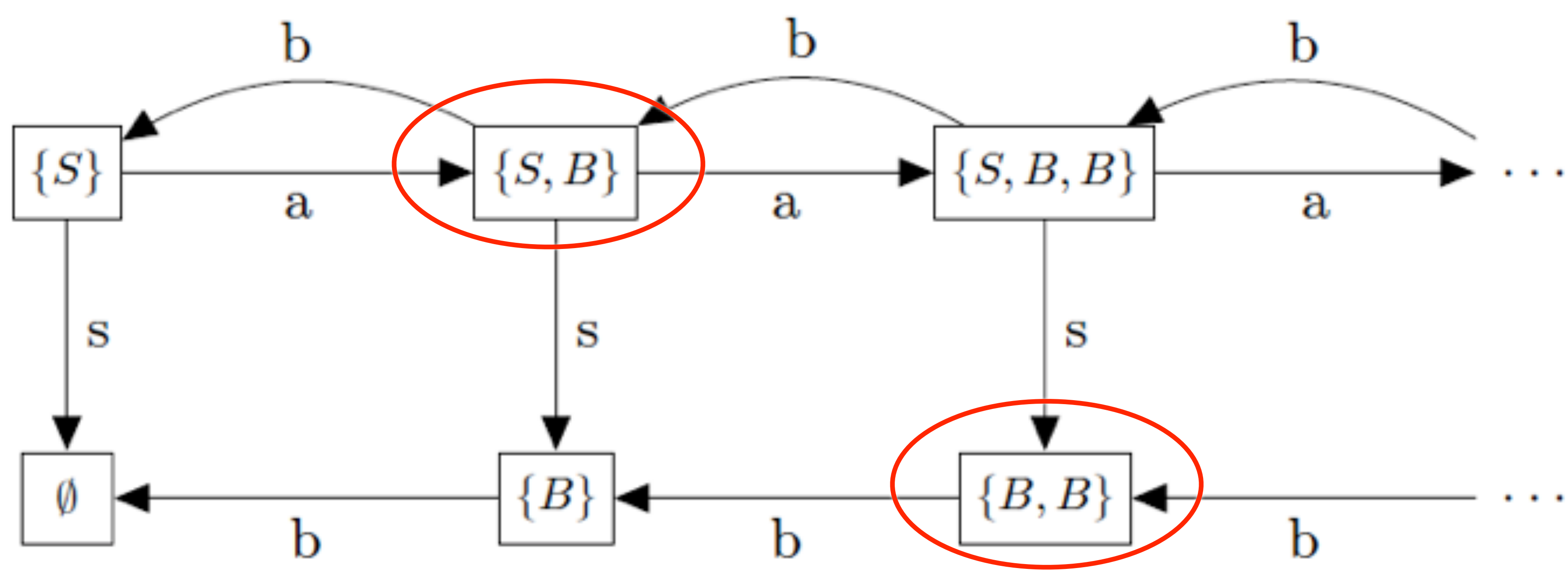
Bisimulation Game



Two-player game (between Spoiler and Duplicator) proceeding in rounds

Configurations = pairs of processes $(\alpha, \beta) \in V^* \times V^*$

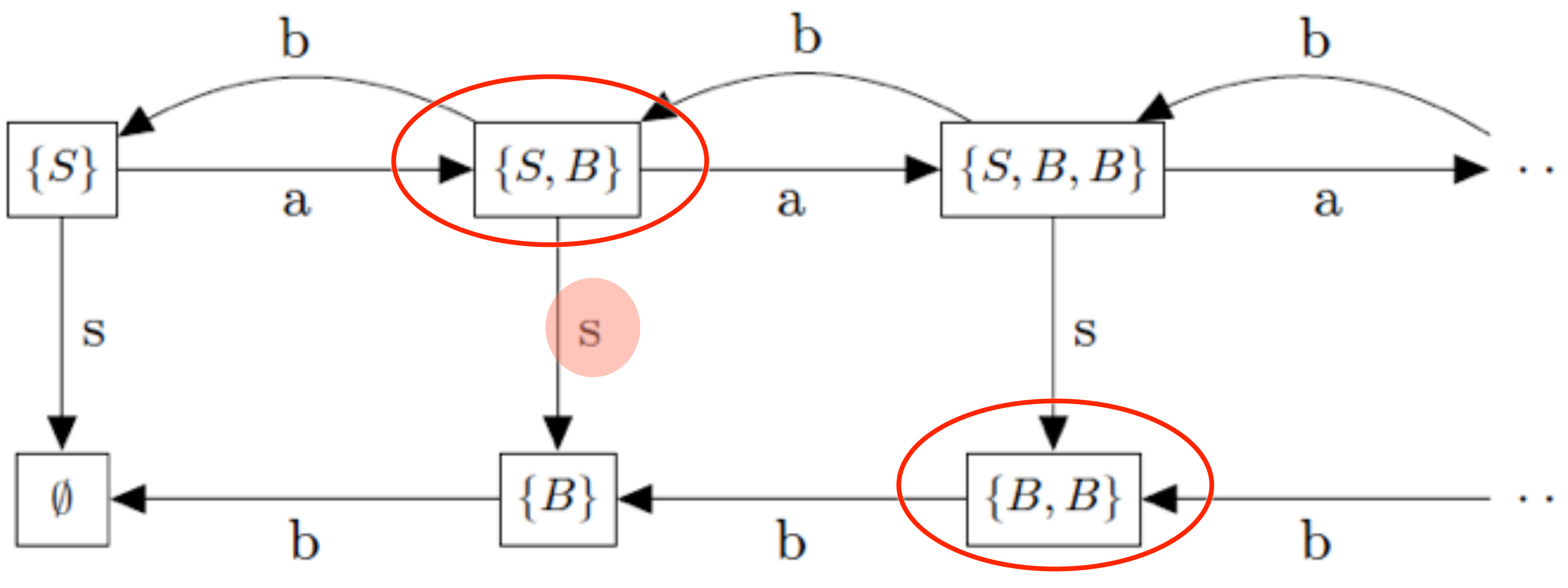
Bisimulation Game



Two-player game (between Spoiler and Duplicator) proceeding in rounds

Configurations = pairs of processes $(\alpha, \beta) \in V^* \times V^*$

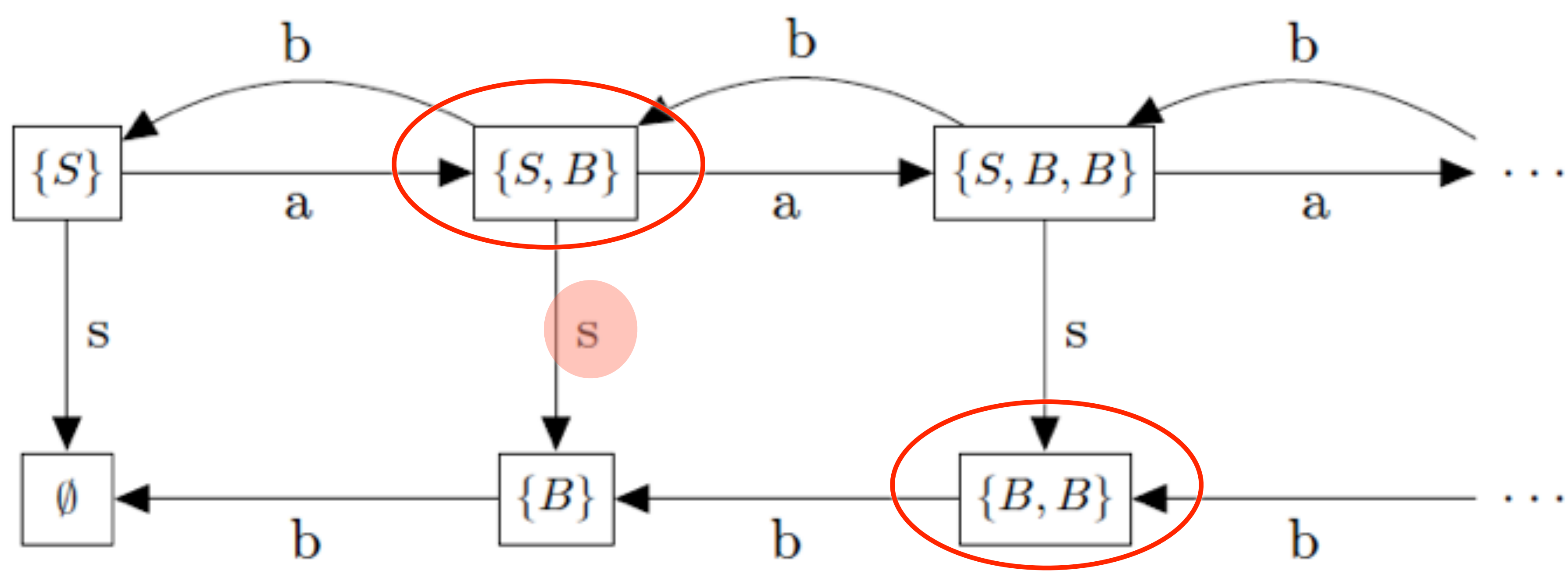
Bisimulation Game



Two-player game (between Spoiler and Duplicator) proceeding in rounds

Configurations = pairs of processes $(\alpha, \beta) \in V^* \times V^*$

Bisimulation Game



Two-player game (between Spoiler and Duplicator) proceeding in rounds

Configurations = pairs of processes $(\alpha, \beta) \in V^* \times V^*$

$\alpha \sim \beta$ iff Duplicator has a winning strategy from (α, β)

Bisimulation equivalence

Bisimulation equivalence

$$R \ \subseteq \ V^* \times V^*$$

Bisimulation equivalence

$R \subseteq V^* \times V^*$

R is a bisimulation if $R \subseteq \exp(R)$

Bisimulation equivalence

$R \subseteq V^* \times V^*$

R **is a** bisimulation **if** $R \subseteq \exp(R)$

$(\alpha, \beta) \in \exp(R) \iff$

Bisimulation equivalence

$R \subseteq V^* \times V^*$

R is a bisimulation if $R \subseteq \exp(R)$

$(\alpha, \beta) \in \exp(R) \iff$

$$\forall a \quad \alpha \xrightarrow{a} \alpha' \implies \beta \xrightarrow{a} \beta' \text{ and } (\alpha', \beta') \in R$$
$$\beta \xrightarrow{a} \beta' \implies \alpha \xrightarrow{a} \alpha' \text{ and } (\alpha', \beta') \in R$$

Bisimulation equivalence

$$R \subseteq V^* \times V^*$$

R is a bisimulation if $R \subseteq \exp(R)$

$$(\alpha, \beta) \in \exp(R) \iff$$

$$\forall a \quad \alpha \xrightarrow{a} \alpha' \implies \beta \xrightarrow{a} \beta' \text{ and } (\alpha', \beta') \in R$$
$$\beta \xrightarrow{a} \beta' \implies \alpha \xrightarrow{a} \alpha' \text{ and } (\alpha', \beta') \in R$$

Bisimulation equivalence is the union of all bisimulations

$$\sim \stackrel{\text{def}}{=} \bigcup \{R : R \text{ is a bisimulation}\}$$

Bisimulation equivalence

$R \subseteq V^* \times V^*$

R is a bisimulation if $R \subseteq \exp(R)$

$$(\alpha, \beta) \in \exp(R) \iff$$

$$\forall a \quad \alpha \xrightarrow{a} \alpha' \implies \beta \xrightarrow{a} \beta' \text{ and } (\alpha', \beta') \in R$$

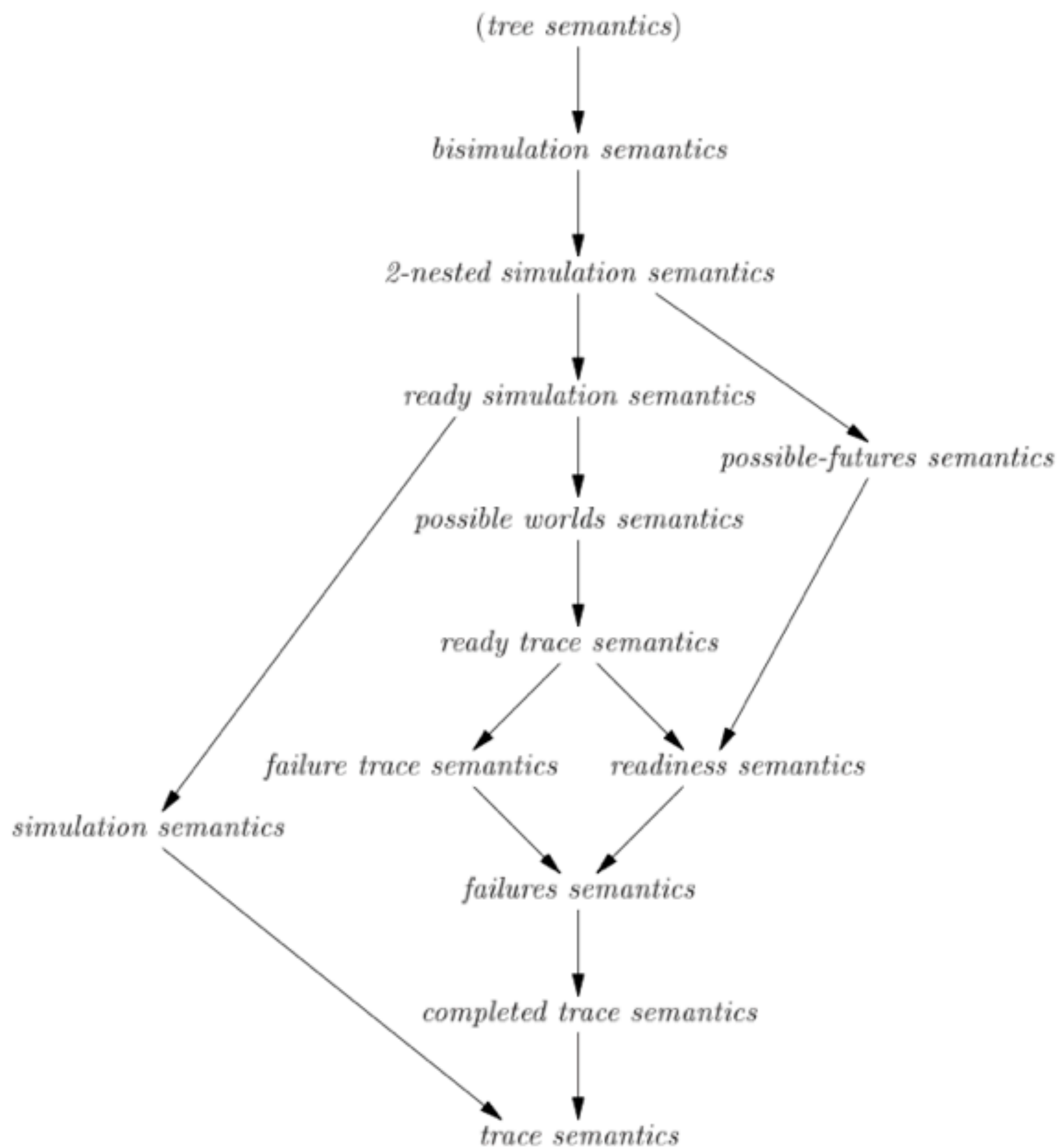
$$\beta \xrightarrow{a} \beta' \implies \alpha \xrightarrow{a} \alpha' \text{ and } (\alpha', \beta') \in R$$

Bisimulation equivalence is the union of all bisimulations

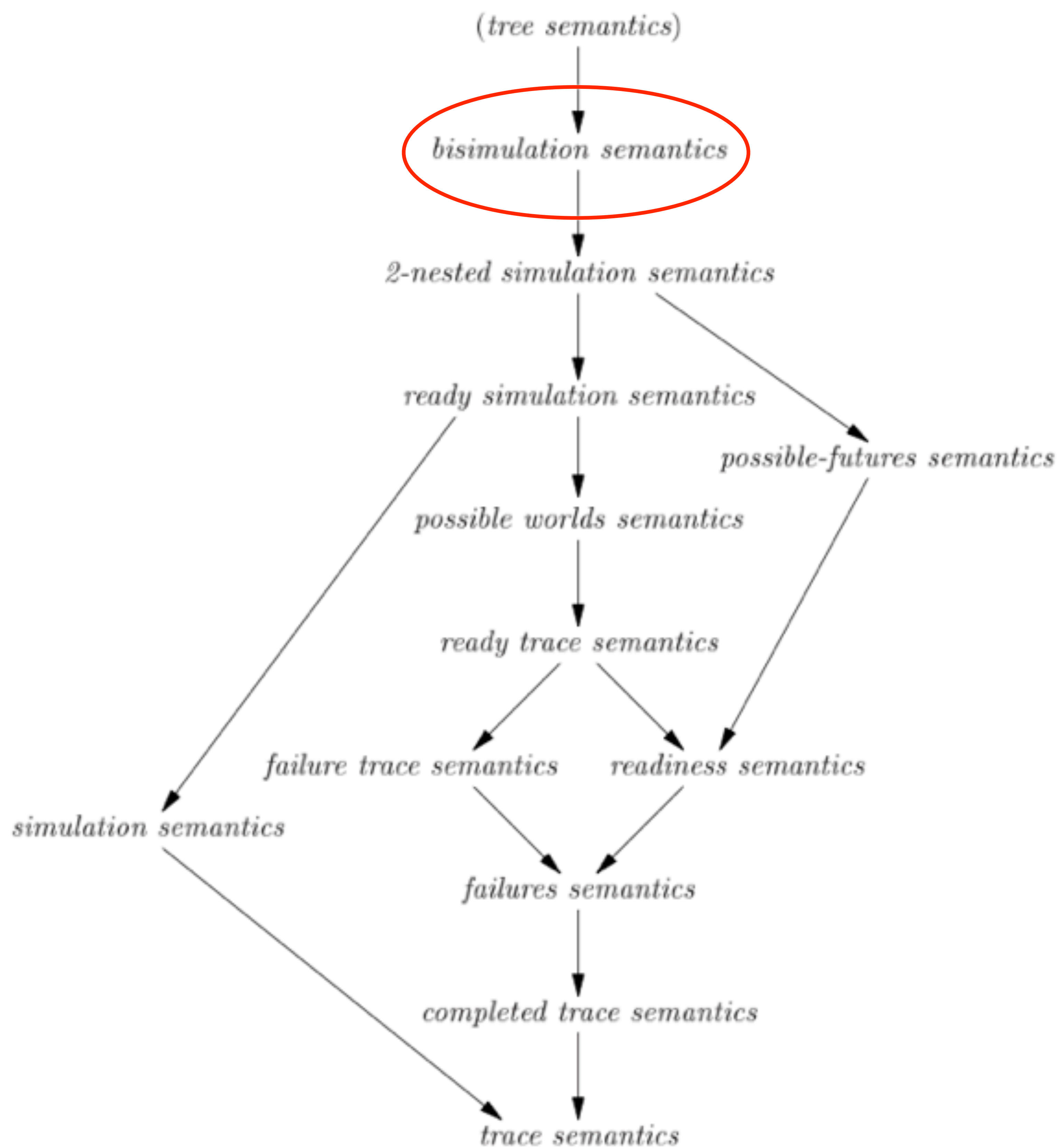
$$\sim \stackrel{\text{def}}{=} \bigcup \{R : R \text{ is a bisimulation}\}$$

Bisimulation equivalence is the greatest bisimulation.

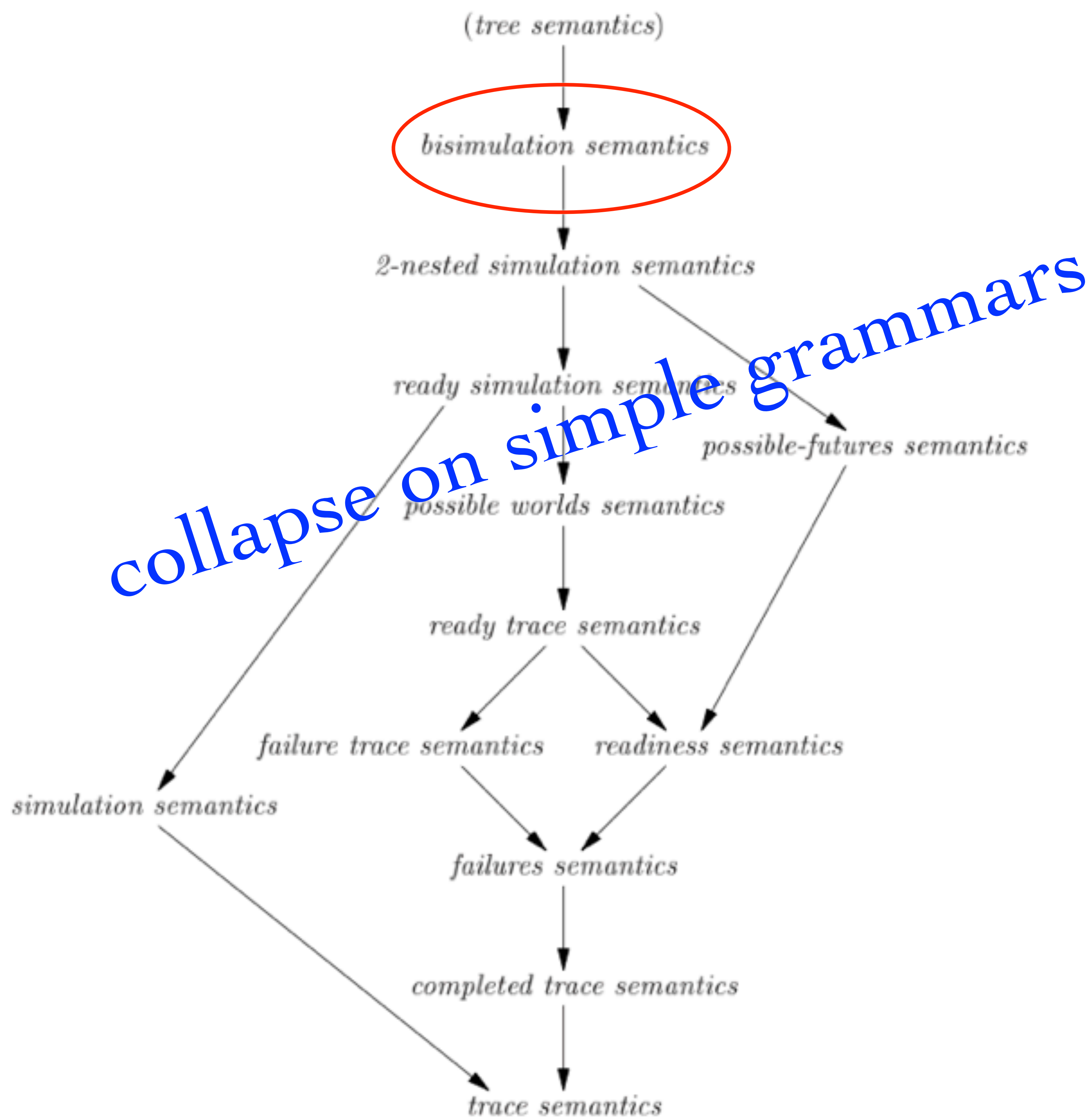
Other equivalences



Other equivalences



Other equivalences



Decision problem

Given a plain/commutative CFG and two processes

$$\alpha, \beta \in V^*$$

Decision problem

Given a plain/commutative CFG and two processes

$$\alpha, \beta \in V^*$$

decide whether they are equivalent

$$\alpha \sim \beta \quad ?$$

Outline

- Background
    - Context-free graphs and commutative context-free graphs
    - Bisimulation equivalence problem
    - **Norm**
    - History of the problem
    - Unique decomposition
    - Naive algorithm
- Efficient algorithm for BPA and BPP
    - Outline of the algorithm
    - Refinement
    - Efficient computation of refinement for BPA
    - Time-cost analysis
    - Partially-commutative context-free graphs

Norm

Norm of a variable X is the length of the shortest path to ε

$$X \xrightarrow{a_1} \ldots \xrightarrow{a_1} \varepsilon$$

(if such path exists)

Norm

the empty process

Norm of a variable X is the length of the shortest path to ε

$$X \xrightarrow{a_1} \ldots \xrightarrow{a_1} \varepsilon$$

(if such path exists)

Norm

the empty process

Norm of a variable X is the length of the shortest path to ε

$$X \xrightarrow{a_1} \ldots \xrightarrow{a_1} \varepsilon$$

(if such path exists)

Example:

$$\begin{array}{cccccccccccc}
X_1 & \xrightarrow{a} & \varepsilon & X_2 & \xrightarrow{a} & X_1 & X_2 & \xrightarrow{b} & X_1X_1 & X_3 & \xrightarrow{a} & X_2 \\
X_4 & \xrightarrow{a} & X_3X_2 & X_4 & \xrightarrow{b} & X_4 & X_5 & \xrightarrow{a} & X_4 & X_5 & \xrightarrow{a} & X_3X_3
\end{array}$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

Norm

the empty process

Norm of a variable X is the length of the shortest path to ε

$$X \xrightarrow{a_1} \dots \xrightarrow{a_1} \varepsilon$$

(if such path exists)

Example:

$$\begin{array}{cccccccccccc}
X_1 & \xrightarrow{a} & \varepsilon & X_2 & \xrightarrow{a} & X_1 & X_2 & \xrightarrow{b} & X_1X_1 & X_3 & \xrightarrow{a} & X_2 \\
X_4 & \xrightarrow{a} & X_3X_2 & X_4 & \xrightarrow{b} & X_4 & X_5 & \xrightarrow{a} & X_4 & X_5 & \xrightarrow{a} & X_3X_3
\end{array}$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

Norms can be easily computed

Norm

the empty process

Norm of a variable X is the length of the shortest path to ε

$$X \xrightarrow{a_1} \ldots \xrightarrow{a_1} \varepsilon$$

(if such path exists)

Example:

$$\begin{array}{ccccccccccccccc} X_1 & \xrightarrow{a} & \varepsilon & & X_2 & \xrightarrow{a} & X_1 & & X_2 & \xrightarrow{b} & X_1X_1 & & X_3 & \xrightarrow{a} & X_2 \\ X_4 & \xrightarrow{a} & X_3X_2 & & X_4 & \xrightarrow{b} & X_4 & & X_5 & \xrightarrow{a} & X_4 & & X_5 & \xrightarrow{a} & X_3X_3 \end{array}$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

Norms can be easily computed

$$X \xrightarrow{a} Y\,Z \qquad |X| = |Y| + |Z| + 1$$

Norm

A variable X is normed if X has finite norm.

Norm

A variable X is normed if X has finite norm.
if X generates a word.

Norm

A variable X is normed if X has finite norm.
if X generates a word.

A grammar is normed if every its variable is normed.

Norm

Fact: Norm of a variable is at most exponential:

Norm

Fact: Norm of a variable is at most exponential:

$$X \xrightarrow{a} Y\,Z \qquad |X| = |Y| + |Z| + 1$$

Norm

Fact: Norm of a variable is at most exponential:

$$X \xrightarrow{a} Y\,Z \qquad |X| = |Y| + |Z| + 1$$

Fact: Bisimulation equivalence is norm-preserving:

$$X \sim Y \implies |X| = |Y|$$

Norm

Fact: Norm of a variable is at most exponential:

$$X \xrightarrow{a} Y\,Z \qquad |X| = |Y| + |Z| + 1$$

Fact: Bisimulation equivalence is norm-preserving:

$$X \sim Y \implies |X| = |Y|$$

Normedness simplifies the bisimulation equivalence problem

Norm

Fact: Norm of a variable is at most exponential:

$$X \xrightarrow{a} Y\,Z \qquad |X| = |Y| + |Z| + 1$$

Fact: Bisimulation equivalence is norm-preserving:

$$X \sim Y \implies |X| = |Y|$$

Normedness simplifies the bisimulation equivalence problem but does not simplify the language equivalence problem

Outline

- Background
  - Context-free graphs and commutative context-free graphs
  - Bisimulation equivalence problem
  - Norm
  - **History of the problem**
  - Unique decomposition
  - Naive algorithm
- Efficient algorithm for BPA and BPP
  - Outline of the algorithm
  - Refinement
  - Efficient computation of refinement for BPA
  - Time-cost analysis
  - Partially-commutative context-free graphs

Complexity of bisimulation equivalence

	normed	unnormed
BPA	P-complete	EXPTIME-hard in 2-EXPTIME
BPP	P-complete	PSPACE-complete

Complexity of bisimulation equivalence

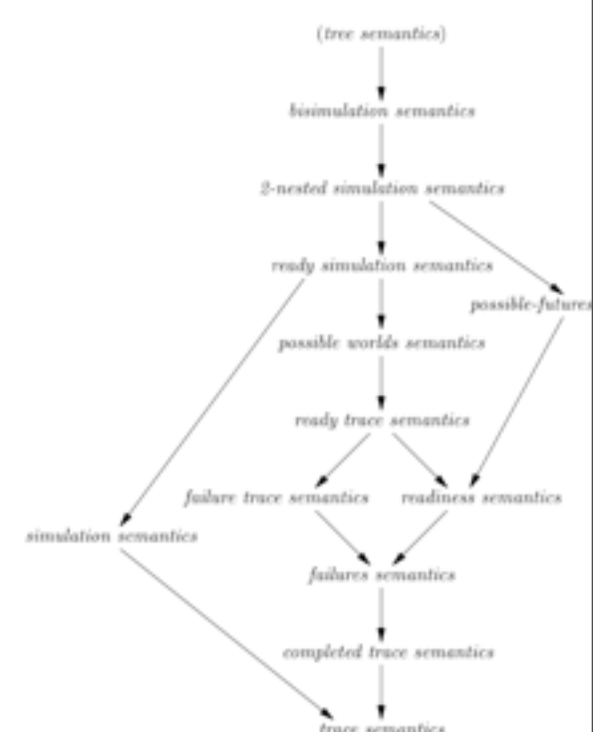
even reachability is NP-complete

	normed	unnormed
BPA	P-complete	EXPTIME-hard in 2-EXPTIME
BPP	P-complete	PSPACE-complete

Complexity of bisimulation equivalence

even reachability is NP-complete

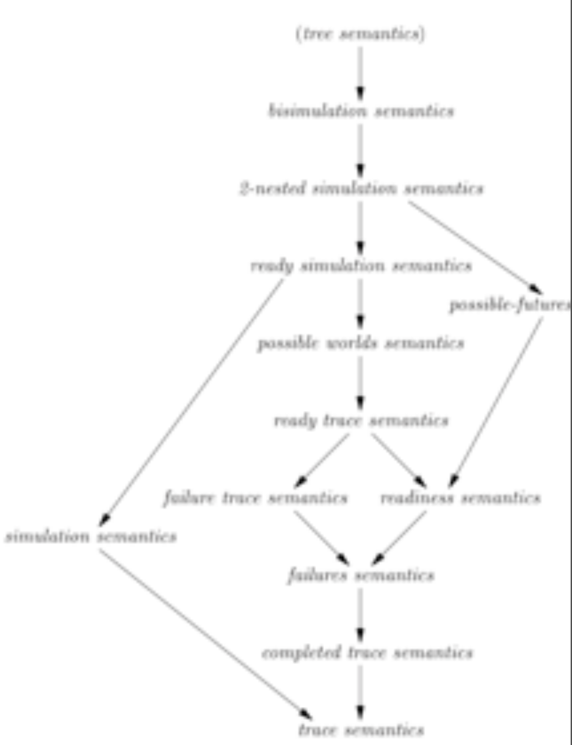
	rmed	unnormed
BPA	P-c mplete	EXPTIME-hard in 2-EXPTIME
BPP	P-complete	PSPACE-complete



essentially all other equivalences are undecidable

Complexity of bisimulation equivalence

	normed	unnormed
BPA	P-complete	EXPTIME-hard in 2-EXPTIME
BPP	P-complete	PSPACE-complete



This motivates searching for efficient polynomial-time algorithms, aiming at understanding better the structure of bisimulation over BPA and BPP

History (upper bound)

	normed	unnormed
language equiv. simple grammars	•n^7 [Bastien, Czyżowicz, Frączak, Rytter 2005] •n^6 [L., Rytter 2006] **•n^3 [Czerwiński's PhD thesis 2012]**	•2-EXPTIME [Korenjak, Hopcroft 1966] •EXPTIME [Caucal 1990]
~ BPA	•decidable [Beaten, Bergstra, Klop 1987] •EXPTIME [Huettel, Stirling 1991] •NP^NP [Huynh, Tian 1994] •P [Hirshfeld, Jerrum, Moller 1996] •n^8 [L., Rytter 2006] **•n^5 [Czerwiński, L. 2010]** **•n^4 [Czerwiński's PhD thesis 2012]**	•decidable [Christensen, Hirshfeld, Stirling 1995] • 2-EXPTIME [Burkart, Caucal, Steffen 1995] •2-EXPTIME [Jancar 2013]
~ BPP	•P [Hirshfeld, Jerrum, Moller 1996] •n^3 [Jancar, Kot 2004]	•decidable [Christensen, Hirshfeld, Moller 1993] •PSPACE [Jancar 2003]

History (upper bound)

- bisimulation equivalence in presence of silent steps (weak bisimulation, branching bisimulation)

	normed	unnormed
language equiv. simple grammars	•n^7 [Bastien, Czyżowicz, Frączak, Rytter 2005] •n^6 [L., Rytter 2006] **•n^3 [Czerwiński's PhD thesis 2012]**	•2-EXPTIME [Korenjak, Hopcroft 1966] •EXPTIME [Caucal 1990]
~ BPA	•decidable [Beaten, Bergstra, Klop 1987] •EXPTIME [Huettel, Stirling 1991] •NP^NP [Huynh, Tian 1994] •P [Hirshfeld, Jerrum, Moller 1996] •n^8 [L., Rytter 2006] **•n^5 [Czerwiński, L. 2010]** **•n^4 [Czerwiński's PhD thesis 2012]**	•decidable [Christensen, Hirshfeld, Stirling 1995] • 2-EXPTIME [Burkart, Caucal, Steffen 1995] •2-EXPTIME [Jancar 2013]
~ BPP	•P [Hirshfeld, Jerrum, Moller 1996] •n^3 [Jancar, Kot 2004]	•decidable [Christensen, Hirshfeld, Moller 1993] •PSPACE [Jancar 2003]

History (upper bound)

- bisimulation equivalence in presence of silent steps (weak bisimulation, branching bisimulation)

- state-extended BPA and BPP (pushdown systems, Petri nets)

	normed	unnormed
language equiv. simple grammars	•n^7 [Bastien, Czyżowicz, Frączak, Rytter 2005] •n^6 [L., Rytter 2006] **•n^3 [Czerwiński's PhD thesis 2012]**	•2-EXPTIME [Korenjak, Hopcroft 1966] •EXPTIME [Caucal 1990]
~ BPA	•decidable [Beaten, Bergstra, Klop 1987] •EXPTIME [Huettel, Stirling 1991] •NP^NP [Huynh, Tian 1994] •P [Hirshfeld, Jerrum, Moller 1996] •n^8 [L., Rytter 2006] **•n^5 [Czerwiński, L. 2010]** **•n^4 [Czerwiński's PhD thesis 2012]**	•decidable [Christensen, Hirshfeld, Stirling 1995] •2-EXPTIME [Burkart, Caucal, Steffen 1995] •2-EXPTIME [Jancar 2013]
~ BPP	•P [Hirshfeld, Jerrum, Moller 1996] •n^3 [Jancar, Kot 2004]	•decidable [Christensen, Hirshfeld, Moller 1993] •PSPACE [Jancar 2003]

History (upper bound)

	normed	unnormed
language equiv. simple grammars	•n^7 [Bastien, Czyżowicz, Frączak, Rytter 2005] •n^6 [L., Rytter 2006] **•n^3 [Czerwiński's PhD thesis 2012]**	•2-EXPTIME [Korenjak, Hopcroft 1966] •EXPTIME [Caucal 1990]
~ BPA	•decidable [Beaten, Bergstra, Klop 1987] •EXPTIME [Huettel, Stirling 1991] •NP^NP [Huynh, Tian 1994] •P [Hirshfeld, Jerrum, Moller 1996] •n^8 [L., Rytter 2006] **•n^5 [Czerwiński, L. 2010]** **•n^4 [Czerwiński's PhD thesis 2012]**	•decidable [Christensen, Hirshfeld, Stirling 1995] •2-EXPTIME [Burkart, Caucal, Steffen 1995] •2-EXPTIME [Jancar 2013]
~ BPP	•P [Hirshfeld, Jerrum, Moller 1996] •n^3 [Jancar, Kot 2004]	•decidable [Christensen, Hirshfeld, Moller 1993] •PSPACE [Jancar 2003]

- bisimulation equivalence in presence of silent steps (weak bisimulation, branching bisimulation)
- state-extended BPA and BPP (pushdown systems, Petri nets)
- lower bounds EPXTIME/PSPACE

History (upper bound)

	normed	unnormed
language equiv. simple grammars	•n^7 [Bastien, Czyżowicz, Frączak, Rytter 2005] •n^6 [L., Rytter 2006] **•n^3 [Czerwiński's PhD thesis 2012]**	•2-EXPTIME [Korenjak, Hopcroft 1966] •EXPTIME [Caucal 1990]
~ BPA	•decidable [Beaten, Bergstra, Klop 1987] •EXPTIME [Huettel, Stirling 1991] •NP^NP [Huynh, Tian 1994] •P [Hirshfeld, Jerrum, Moller 1996] •n^8 [L., Rytter 2006] **•n^5 [Czerwiński, L. 2010]** **•n^4 [Czerwiński's PhD thesis 2012]**	•decidable [Christensen, Hirshfeld, Stirling 1995] • 2-EXPTIME [Burkart, Caucal, Steffen 1995] •2-EXPTIME [Jancar 2013]
~ BPP	•P [Hirshfeld, Jerrum, Moller 1996] •n^3 [Jancar, Kot 2004]	•decidable [Christensen, Hirshfeld, Moller 1993] •PSPACE [Jancar 2003]

- bisimulation equivalence in presence of silent steps (weak bisimulation, branching bisimulation)
- state-extended BPA and BPP (pushdown systems, Petri nets)
- lower bounds EPXTIME/PSPACE
- equivalence with a finite state system

History (upper bound)

	normed	unnormed
language equiv. simple grammars	•n^7 [Bastien, Czyżowicz, Frączak, Rytter 2005] •n^6 [L., Rytter 2006] **•n^3 [Czerwiński's PhD thesis 2012]**	•2-EXPTIME [Korenjak, Hopcroft 1966] •EXPTIME [Caucal 1990]
~ BPA	•decidable [Beaten, Bergstra, Klop 1987] •EXPTIME [Huettel, Stirling 1991] •NP^NP [Huynh, Tian 1994] •P [Hirshfeld, Jerrum, Moller 1996] •n^8 [L., Rytter 2006] **•n^5 [Czerwiński, L. 2010]** **•n^4 [Czerwiński's PhD thesis 2012]**	•decidable [Christensen, Hirshfeld, Stirling 1995] •2-EXPTIME [Burkart, Caucal, Steffen 1995] •2-EXPTIME [Jancar 2013]
~ BPP	•P [Hirshfeld, Jerrum, Moller 1996] •n^3 [Jancar, Kot 2004]	•decidable [Christensen, Hirshfeld, Moller 1993] •PSPACE [Jancar 2003]

- bisimulation equivalence in presence of silent steps (weak bisimulation, branching bisimulation)
- state-extended BPA and BPP (pushdown systems, Petri nets)
- lower bounds EPXTIME/PSPACE
- equivalence with a finite state system
- finiteness up to bisimulation
- BPA vs. BPP
- ...

Outline

- Background
  - Context-free graphs and commutative context-free graphs
  - Bisimulation equivalence problem
  - Norm
  - History of the problem
  - **Unique decomposition**
  - Naive algorithm
- Efficient algorithm for BPA and BPP
  - Outline of the algorithm
  - Refinement
  - Efficient computation of refinement for BPA
  - Time-cost analysis
  - Partially-commutative context-free graphs

Congruence

$$\alpha \;\sim\; \alpha' \;\; \text{and} \;\; \beta \;\sim\; \beta' \implies \alpha\,\beta \;\sim\; \alpha'\,\beta'$$

Fact: Bisimulation equivalence is a congruence, both over BPA and BPP.

From now on we restrict ourselves to

normed

BPA and BPP

Unique decomposition
(both for BPA and BPP)

Unique decomposition
(both for BPA and BPP)

order variables according to norm: $|X_1| \leq |X_2| \leq \ldots \leq |X_n|$

Unique decomposition
(both for BPA and BPP)

order variables according to norm: $|X_1| \leq |X_2| \leq \ldots \leq |X_n|$

choose a congruence $\equiv$

Unique decomposition
(both for BPA and BPP)

order variables according to norm: $|X_1| \;\leq\; |X_2| \;\leq\; \ldots \;\leq\; |X_n|$

choose a congruence $\equiv$

decomposition: $X_i \;\equiv\; \alpha, \qquad \alpha \in \{X_1 \ldots X_{i-1}\}^*$

Unique decomposition
(both for BPA and BPP)

order variables according to norm: $|X_1| \;\leq\; |X_2| \;\leq\; \ldots \;\leq\; |X_n|$

choose a congruence $\equiv$

decomposition: $X_i \;\equiv\; \alpha, \qquad \alpha \in \{X_1 \ldots X_{i-1}\}^*$

Def: A variable is decomposable if it admits a non-trivial decomposition; otherwise it is prime.

Unique decomposition
(both for BPA and BPP)

order variables according to norm: $|X_1| \leq |X_2| \leq \ldots \leq |X_n|$

choose a congruence $\equiv$

decomposition: $X_i \equiv \alpha, \qquad \alpha \in \{X_1 \ldots X_{i-1}\}^*$

Def: A variable is decomposable if it admits a non-trivial decomposition; otherwise it is prime.

Def: A congruence has unique decomposition property if every variable has a unique decomposition into primes.

Unique decomposition
(both for BPA and BPP)

order variables according to norm: $|X_1| \leq |X_2| \leq \ldots \leq |X_n|$

choose a congruence $\equiv$

decomposition: $X_i \equiv \alpha, \qquad \alpha \in \{X_1 \ldots X_{i-1}\}^*$

Def: A variable is decomposable if it admits a non-trivial decomposition; otherwise it is prime.

Def: A congruence has unique decomposition property if every variable has a unique decomposition into primes.

Fact: Every variable has a decomposition into primes.

Unique decomposition
(both for BPA and BPP)

order variables according to norm: $|X_1| \;\leq\; |X_2| \;\leq\; \ldots \;\leq\; |X_n|$

choose a congruence $\equiv$

decomposition: $X_i \;\equiv\; \alpha, \qquad \alpha \in \{X_1 \ldots X_{i-1}\}^*$

Def: A variable is decomposable if it admits a non-trivial decomposition; otherwise it is prime.

Def: A congruence has unique decomposition property if every variable has a unique decomposition into primes.

Fact: Every variable has a decomposition into primes.

Thm: Bisimulation equivalence has unique decomposition property, both over BPA and BPP.

Unique decomposition

Example:

$$\begin{array}{llllllllllll}
X_1 & \xrightarrow{a} & \varepsilon & X_2 & \xrightarrow{a} & X_1 & X_2 & \xrightarrow{b} & X_1X_1 & X_3 & \xrightarrow{a} & X_2 \\
X_4 & \xrightarrow{a} & X_3X_2 & X_4 & \xrightarrow{b} & X_4 & X_5 & \xrightarrow{a} & X_4 & X_5 & \xrightarrow{a} & X_3X_3
\end{array}$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

Unique decomposition

Example:

is it BPA or BPP?

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

Unique decomposition

is it BPA or BPP?

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

prime variables

decomposable variables

Unique decomposition

is it BPA or BPP?

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

prime variables

X_1

decomposable variables

Unique decomposition

is it BPA or BPP?

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

prime variables

X_1

$X_2 \not\sim X_1 X_1$

decomposable variables

Unique decomposition

is it BPA or BPP?

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1 X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3 X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3 X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

prime variables

X_1

$X_2 \not\sim X_1 X_1$

decomposable variables

$X_3 \sim X_1 X_2$

Unique decomposition

is it BPA or BPP?

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

prime variables

X_1

$X_2 \not\sim X_1\, X_1$

decomposable variables

$X_3 \sim X_1\, X_2$

$\mathrm{dec}(X_3)$

Unique decomposition

is it BPA or BPP?

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

prime variables

X_1

$X_2 \not\sim X_1\,X_1$

$X_4 \not\sim X_1\,\alpha$

decomposable variables

$X_3 \sim X_1\,X_2$

$\mathrm{dec}(X_3)$

Unique decomposition

is it BPA or BPP?

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

prime variables

X_1

$X_2 \not\sim X_1\,X_1$

$X_4 \not\sim X_1\,\alpha \qquad X_4 \not\sim X_2\,\alpha$

decomposable variables

$X_3 \sim X_1\,X_2$

$\mathrm{dec}(X_3)$

Unique decomposition

is it BPA or BPP?

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1 X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3 X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3 X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

prime variables

X_1

$X_2 \not\sim X_1 X_1$

$X_4 \not\sim X_1 \alpha \qquad X_4 \not\sim X_2 \alpha$

X_5

decomposable variables

$X_3 \sim X_1 X_2$

$\mathrm{dec}(X_3)$

Algebraic perspective

unique decomposition property says that

Algebraic perspective

unique decomposition property says that

quotient of V^* by bisimulation equivalence $\sim$ is a free semigroup (resp. free commutative semigroup).

Cancellation

$$\alpha\gamma \sim \beta\gamma \implies \alpha \sim \beta$$

Cancellation

$$\alpha\gamma \ \sim \ \beta\gamma \quad \Longrightarrow \quad \alpha \ \sim \ \beta$$

Fact: unique decomposition $\quad \Longrightarrow \quad$ cancellation

Cancellation

$$\alpha\gamma \sim \beta\gamma \implies \alpha \sim \beta$$

Fact: unique decomposition $\implies$ cancellation

Proof: (both for BPA and BPP)

Cancellation

$$\alpha\gamma \sim \beta\gamma \implies \alpha \sim \beta$$

Fact: unique decomposition $\implies$ cancellation

Proof: (both for BPA and BPP)

extend dec to all processes:

$$\mathrm{dec}(\alpha\,\beta) = \mathrm{dec}(\alpha)\,\mathrm{dec}(\beta)$$

Cancellation

$$\alpha\,\gamma \;\sim\; \beta\,\gamma \quad\Longrightarrow\quad \alpha \;\sim\; \beta$$

Fact: unique decomposition $\Longrightarrow$ cancellation

Proof: (both for BPA and BPP)

extend dec to all processes:

$$\mathrm{dec}(\alpha\,\beta) \;=\; \mathrm{dec}(\alpha)\,\mathrm{dec}(\beta)$$

$$\alpha \sim \beta \quad\Longleftrightarrow\quad \mathrm{dec}(\alpha) = \mathrm{dec}(\beta)$$

Cancellation

$$\alpha\gamma \sim \beta\gamma \implies \alpha \sim \beta$$

Fact: unique decomposition $\implies$ cancellation

Proof: (both for BPA and BPP)

extend dec to all processes:

$$\mathrm{dec}(\alpha\,\beta) = \mathrm{dec}(\alpha)\,\mathrm{dec}(\beta)$$

$$\alpha \sim \beta \iff \mathrm{dec}(\alpha) = \mathrm{dec}(\beta)$$

and observe:

$$\mathrm{dec}(\alpha)\,\mathrm{dec}(\gamma) = \mathrm{dec}(\beta)\,\mathrm{dec}(\gamma) \implies \mathrm{dec}(\alpha) = \mathrm{dec}(\beta)$$

BPA and BPP differ

proofs of unique decomposition

$$\begin{array}{c} \text{unique decomposition} \\ \text{BPA} \Uparrow \quad \Downarrow \text{BPP} \\ \text{cancellation} \end{array}$$

[Hirshfeld, Jerrum, Moller 1996]

Base

Base

Base = prime variables + decompositions of other variables

Base

Base = prime variables + decompositions of other variables

Primes and decompositions may be chosen arbitrarily

Base

Base = prime variables + decompositions of other variables

Primes and decompositions may be chosen arbitrarily

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

Base

Base = prime variables + decompositions of other variables

Primes and decompositions may be chosen arbitrarily

Example:

$$
\begin{array}{llllllllllll}
X_1 & \xrightarrow{a} & \varepsilon & X_2 & \xrightarrow{a} & X_1 & X_2 & \xrightarrow{b} & X_1X_1 & X_3 & \xrightarrow{a} & X_2 \\
X_4 & \xrightarrow{a} & X_3X_2 & X_4 & \xrightarrow{b} & X_4 & X_5 & \xrightarrow{a} & X_4 & X_5 & \xrightarrow{a} & X_3X_3
\end{array}
$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$\mathrm{dec}_B(X_3)$

$$B \quad = \quad (\{X_1, X_2, X_4, X_5\}, \quad \{X_3 = X_1\, X_2\})$$

Base

Base = prime variables + decompositions of other variables

Primes and decompositions may be chosen arbitrarily

Example:

$$
\begin{array}{llllllllllll}
X_1 & \xrightarrow{a} & \varepsilon & X_2 & \xrightarrow{a} & X_1 & X_2 & \xrightarrow{b} & X_1X_1 & X_3 & \xrightarrow{a} & X_2 \\
X_4 & \xrightarrow{a} & X_3X_2 & X_4 & \xrightarrow{b} & X_4 & X_5 & \xrightarrow{a} & X_4 & X_5 & \xrightarrow{a} & X_3X_3
\end{array}
$$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

$\mathrm{dec}_B(X_3)$

$$B = (\{X_1, X_2, X_4, X_5\}, \ \{X_3 = X_1\, X_2\})$$

$$B = (\{X_1, X_4\}, \ \{X_2 = X_1, X_3 = X_1\, X_1, \ X_5 = X_4\})$$

It is sufficient to restrict to

norm-preserving

bases

$$|X| = |\mathrm{dec}_B(X)|$$

It is sufficient to restrict to norm-preserving bases

$$|X| \; = \; |\mathrm{dec}_B(X)|$$

Example:

$$\begin{array}{llllllllllll}
X_1 & \xrightarrow{a} & \varepsilon & X_2 & \xrightarrow{a} & X_1 & X_2 & \xrightarrow{b} & X_1 X_1 & X_3 & \xrightarrow{a} & X_2 \\
X_4 & \xrightarrow{a} & X_3 X_2 & X_4 & \xrightarrow{b} & X_4 & X_5 & \xrightarrow{a} & X_4 & X_5 & \xrightarrow{a} & X_3 X_3
\end{array}$$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

$\mathrm{dec}_B(X_3)$

$$B \;=\; (\{X_1, X_2, X_4, X_5\}, \;\; \{X_3 = X_1\, X_2\})$$

$$B \;=\; (\{X_1, X_4\}, \;\; \{X_2 = X_1, X_3 = X_1\, X_1, \;\; X_5 = X_4\})$$

It is sufficient to restrict to **norm-preserving** bases

$$|X| = |\mathrm{dec}_B(X)|$$

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1 X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3 X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3 X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$\mathrm{dec}_B(X_3)$

$$B = (\{X_1, X_2, X_4, X_5\}, \ \{X_3 = X_1 X_2\})$$ yes

$$B = (\{X_1, X_4\}, \ \{X_2 = X_1, X_3 = X_1 X_1, \ X_5 = X_4\})$$

It is sufficient to restrict to norm-preserving bases

$$|X| = |\mathrm{dec}_B(X)|$$

Example:

$$\begin{array}{cccccccccccc} X_1 & \xrightarrow{a} & \varepsilon & X_2 & \xrightarrow{a} & X_1 & X_2 & \xrightarrow{b} & X_1X_1 & X_3 & \xrightarrow{a} & X_2 \\ X_4 & \xrightarrow{a} & X_3X_2 & X_4 & \xrightarrow{b} & X_4 & X_5 & \xrightarrow{a} & X_4 & X_5 & \xrightarrow{a} & X_3X_3 \end{array}$$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

$\mathrm{dec}_B(X_3)$

$B = (\{X_1, X_2, X_4, X_5\}, \ \{X_3 = X_1 X_2\})$ yes

$B = (\{X_1, X_4\}, \ \{X_2 = X_1, X_3 = X_1 X_1, \ X_5 = X_4\})$ no

Base as a representation

Base as a representation

extend dec_B to all processes :

$$\mathrm{dec}_B(\alpha\,\beta) = \mathrm{dec}_B(\alpha)\,\mathrm{dec}_B(\beta)$$

Base as a representation

extend dec_B to all processes :

$$\mathrm{dec}_B(\alpha\,\beta) = \mathrm{dec}_B(\alpha)\,\mathrm{dec}_B(\beta)$$

congruence represented by a base B:

$$\alpha \;\equiv_B\; \beta \quad\Longleftrightarrow\quad \mathrm{dec}_B(\alpha) \;=\; \mathrm{dec}_B(\beta)$$

Base as a representation

extend dec_B to all processes :

$$\mathrm{dec}_B(\alpha\,\beta) = \mathrm{dec}_B(\alpha)\,\mathrm{dec}_B(\beta)$$

congruence represented by a base B:

$$\alpha \;\equiv_B\; \beta \quad\Longleftrightarrow\quad \mathrm{dec}_B(\alpha) \;=\; \mathrm{dec}_B(\beta)$$

Fact: A congruence has unique decomposition property
iff
it is represented by a base

Outline

- Background
  - Context-free graphs and commutative context-free graphs
  - Bisimulation equivalence problem
  - Norm
  - History of the problem
  - Unique decomposition
  - **Naive algorithm**
- Efficient algorithm for BPA and BPP
  - Outline of the algorithm
  - Refinement
  - Efficient computation of refinement for BPA
  - Time-cost analysis
  - Partially-commutative context-free graphs

Bisimulation base

Bisimulation equivalence problem:

Given a plain/commutative CFG and two processes

$$\alpha, \beta \in V^*$$

decide whether they are equivalent:

$$\alpha \ \sim \ \beta$$

Bisimulation base

Bisimulation equivalence problem:

Given a plain/commutative CFG and two processes

$$\alpha, \beta \in V^*$$

decide whether they are equivalent:

$$\alpha \ \sim \ \beta$$

$$X \xrightarrow{a} \alpha$$
$$Y \xrightarrow{a} \beta$$

$$X \ \sim \ Y$$

Bisimulation base

Bisimulation equivalence problem:

Given a plain/commutative CFG and two processes

$$\alpha, \beta \in V^*$$

decide whether they are equivalent:

$$\alpha \ \sim \ \beta$$

$$X \xrightarrow{a} \alpha$$
$$Y \xrightarrow{a} \beta$$

$$X \sim Y$$

bisimulation base = base representing a bisimulation

Bisimulation base

Bisimulation equivalence problem:

Given a plain/commutative CFG and two processes

$$\alpha, \beta \in V^*$$

decide whether they are equivalent:

$$\alpha \ \sim \ \beta$$

$$X \xrightarrow{a} \alpha$$
$$Y \xrightarrow{a} \beta$$

$$X \ \sim \ Y$$

bisimulation base = base representing a bisimulation

For solving the bisimulation equivalence problem, it is sufficient to compute a bisimulation base containing the decomposition $X = Y$ (if any such exists)

Naive algorithm

Naive algorithm

- Guess a base B

Naive algorithm

- Guess a base B

the size of B is at most exponential, as it has to be norm-preserving

Naive algorithm

in BPA, we need a compressed representation of decompositions

- Guess a base B

the size of B is at most exponential, as it has to be norm-preserving

Naive algorithm

- Guess a base B

the size of B is at most exponential, as it has to be norm-preserving

- For every decomposition

$$X \;=\; \alpha$$

in B, check whether

$$(X, \alpha) \;\in\; \exp(\equiv_B)$$

Naive algorithm

- Guess a base B

the size of B is at most exponential, as it has to be norm-preserving

- For every decomposition

$$X \;=\; \alpha$$

in B, check whether

$$(X, \alpha) \;\in\; \exp(\equiv_B)$$

polynomially many equality tests

$$\mathrm{dec}_B(\beta) \;=\; \mathrm{dec}_B(\gamma)$$

$$X \xrightarrow{a} \beta \qquad \alpha \xrightarrow{a} \gamma$$

Naive algorithm

- Guess a base B

the size of B is at most exponential, as it has to be norm-preserving

- For every decomposition

$$X \;=\; \alpha$$

in B, check whether

$$(X, \alpha) \;\in\; \exp(\equiv_B)$$

polynomially many equality tests

$$\mathrm{dec}_B(\beta) \;=\; \mathrm{dec}_B(\gamma)$$

$$X \xrightarrow{a} \beta \qquad \alpha \xrightarrow{a} \gamma$$

$$B \;\subseteq\; \exp(\equiv_B) \quad\Longrightarrow\quad \equiv_B \subseteq\; \exp(\equiv_B)$$

[Caucal 1990]

Compressed representation of strings

Compressed representation of strings

- straight-line programs

Compressed representation of strings

- straight-line programs
- acyclic morphisms

Compressed representation of strings

- straight-line programs
- acyclic morphisms
- grammars generating a single word

Compressed representation of strings

- straight-line programs
- acyclic morphisms
- grammars generating a single word
- ...

Compressed representation of strings

- straight-line programs
- acyclic morphisms
- grammars generating a single word
- ...

Example:

$$\begin{array}{lcc} X_0 & \longrightarrow & a \\ X_1 & \longrightarrow & a\,b \\ X_2 & \longrightarrow & X_1\,X_0 \\ X_3 & \longrightarrow & X_2\,X_1 \\ & \dots & \end{array}$$

Compressed representation of strings

- straight-line programs
- acyclic morphisms
- grammars generating a single word
- ...

Example:

$$\begin{aligned} X_0 &\longrightarrow a \\ X_1 &\longrightarrow a\,b \\ X_2 &\longrightarrow X_1\,X_0 \\ X_3 &\longrightarrow X_2\,X_1 \\ &\dots \end{aligned}$$

aba

Compressed representation of strings

- straight-line programs
- acyclic morphisms
- grammars generating a single word
- ...

Example:

$$\begin{array}{lcll} X_0 & \longrightarrow & a & \\ X_1 & \longrightarrow & a\,b & \\ X_2 & \longrightarrow & X_1\,X_0 & \text{aba} \\ X_3 & \longrightarrow & X_2\,X_1 & \text{abaab} \\ & \dots & & \end{array}$$

Compressed representation of strings

- straight-line programs
- acyclic morphisms
- grammars generating a single word
- ...

Example:

$$\begin{array}{lcll} X_0 & \longrightarrow & a & \\ X_1 & \longrightarrow & a\,b & \\ X_2 & \longrightarrow & X_1\,X_0 & \text{aba} \\ X_3 & \longrightarrow & X_2\,X_1 & \text{abaab} \\ & \ldots & & \text{abaababa} \end{array}$$

Compressed representation of strings

- straight-line programs
- acyclic morphisms
- grammars generating a single word
- ...

Example:

$$\begin{array}{lcl} X_0 & \longrightarrow & a \\ X_1 & \longrightarrow & a\, b \\ X_2 & \longrightarrow & X_1\, X_0 \\ X_3 & \longrightarrow & X_2\, X_1 \\ & \ldots & \end{array}$$

aba
abaab
abaababa
abaababaabaab

Compressed representation of strings

- straight-line programs
- acyclic morphisms
- grammars generating a single word
- ...

Example:

$$
\begin{array}{rcl}
X_0 & \longrightarrow & a \\
X_1 & \longrightarrow & a\, b \\
X_2 & \longrightarrow & X_1\, X_0 \\
X_3 & \longrightarrow & X_2\, X_1 \\
 & \dots &
\end{array}
$$

aba
abaab
abaababa
abaababaabaab
...

Compressed representation of strings

- straight-line programs
- acyclic morphisms
- grammars generating a single word
- ...

Example:

$$\begin{array}{lcc} X_0 & \longrightarrow & a \\ X_1 & \longrightarrow & a\,b \\ X_2 & \longrightarrow & X_1\,X_0 \\ X_3 & \longrightarrow & X_2\,X_1 \\ & \dots & \end{array}$$

aba
abaab
abaababa
abaababaabaab
...

- exponential compression

Compressed representation of strings

- straight-line programs
- acyclic morphisms
- grammars generating a single word
- ...

Example:

$$\begin{array}{lcc} X_0 & \longrightarrow & a \\ X_1 & \longrightarrow & a\,b \\ X_2 & \longrightarrow & X_1\,X_0 \\ X_3 & \longrightarrow & X_2\,X_1 \\ & \dots & \end{array}$$

aba
abaab
abaababa
abaababaabaab
...

- exponential compression
- efficient manipulation without decompression

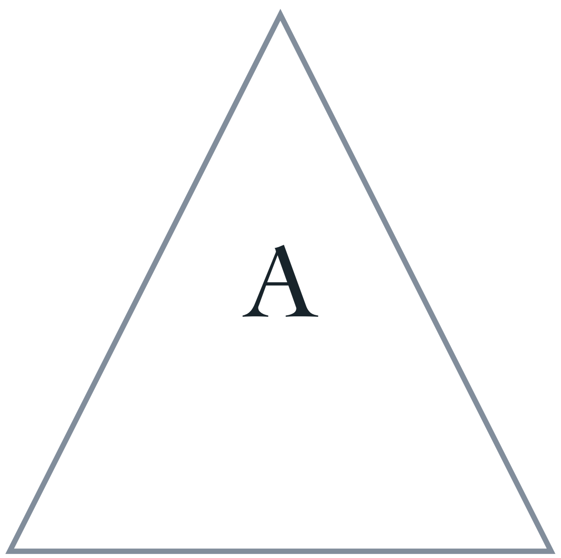
Operations on compressed strings

Operations on compressed strings

$$A \longrightarrow B\,C$$

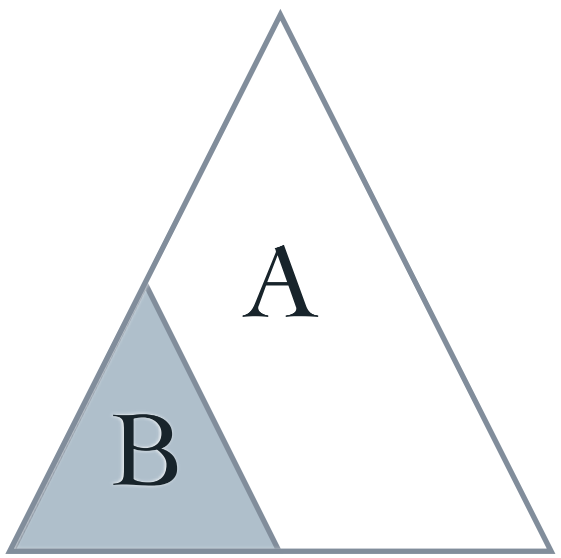
Operations on compressed strings

$$A \longrightarrow B\,C$$



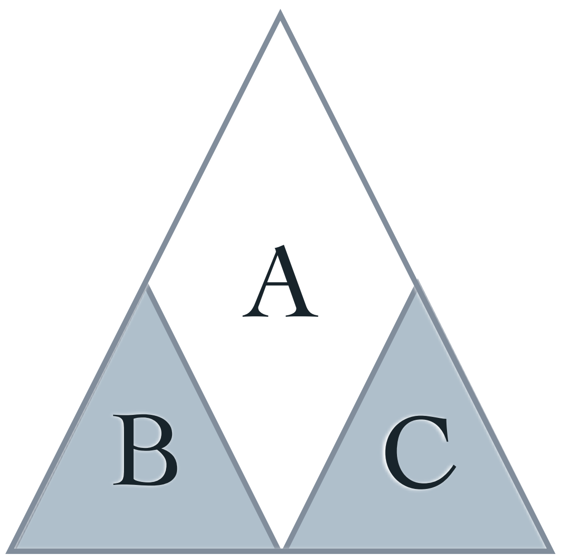
Operations on compressed strings

$$A \longrightarrow B\,C$$



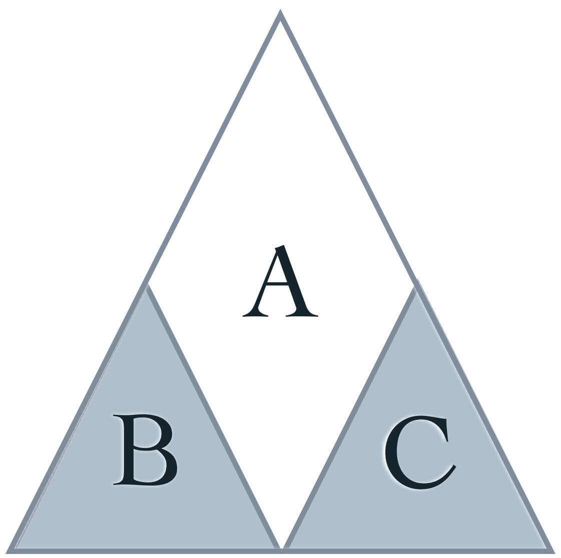
Operations on compressed strings

$$A \longrightarrow B\,C$$



Operations on compressed strings

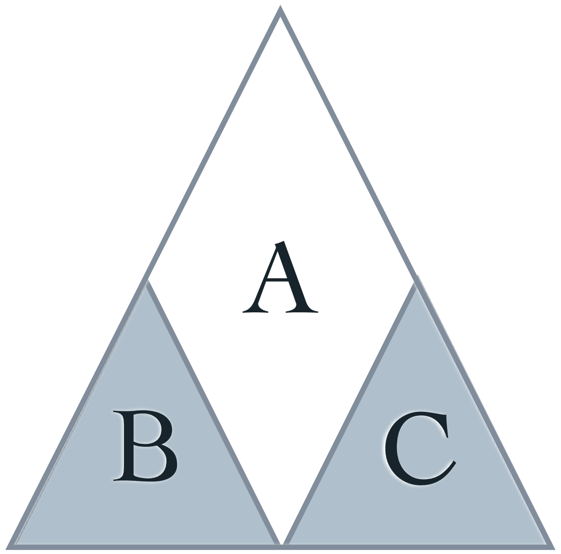
$$A \longrightarrow B\,C$$



$$A \longrightarrow B\,C$$
$$X \longrightarrow Y\,Z$$

Operations on compressed strings

$$A \longrightarrow B\,C$$

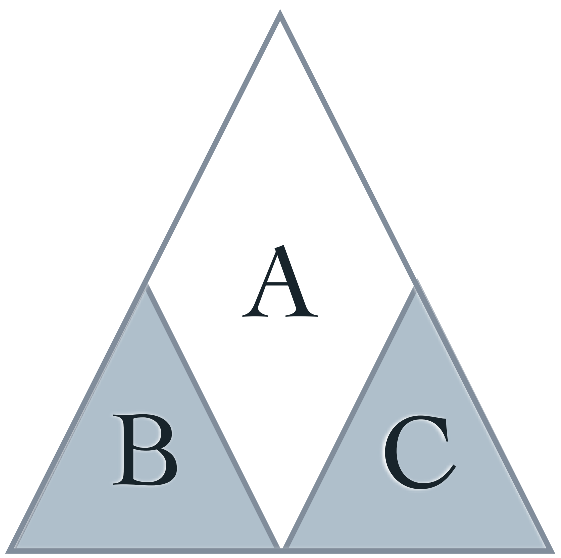


$$A \longrightarrow B\,C$$
$$X \longrightarrow Y\,Z$$

equality test A = X ?

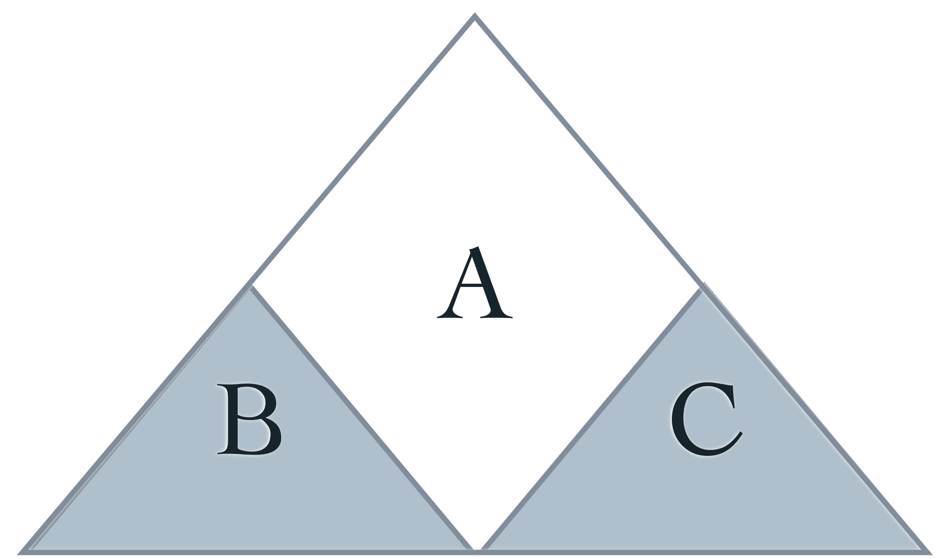
Operations on compressed strings

$$A \longrightarrow B\,C$$



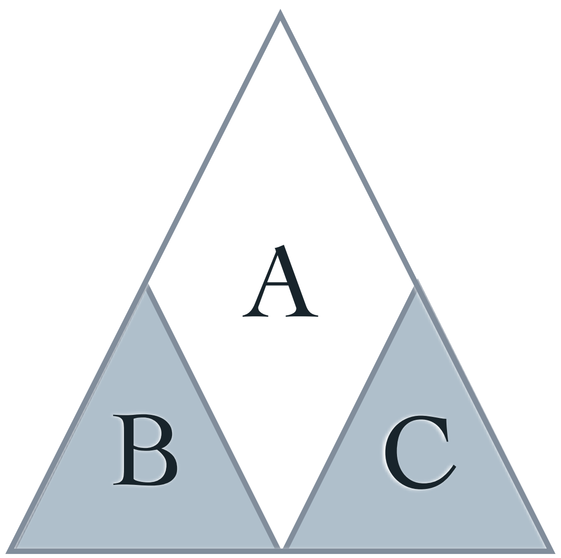
$$A \longrightarrow B\,C$$
$$X \longrightarrow Y\,Z$$

equality test A = X ?



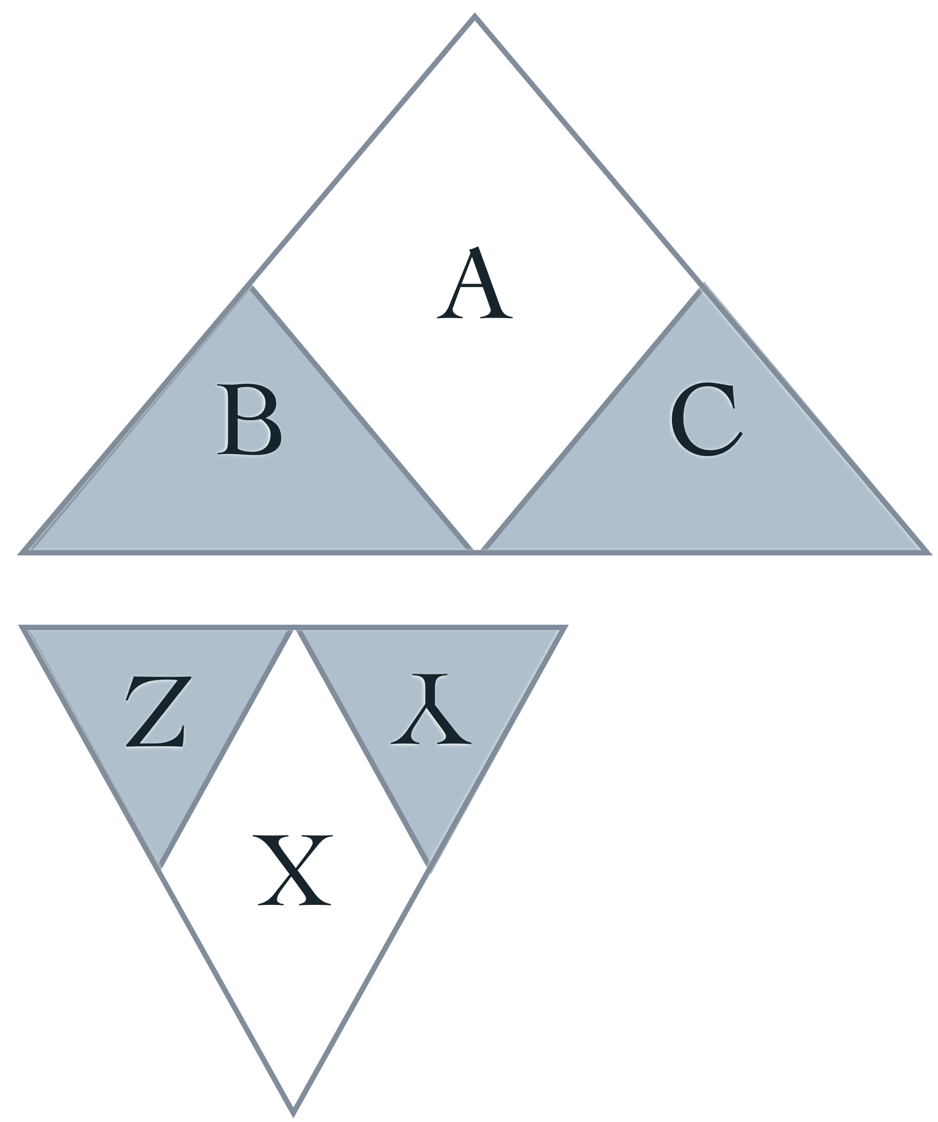
Operations on compressed strings

$$A \longrightarrow B\,C$$



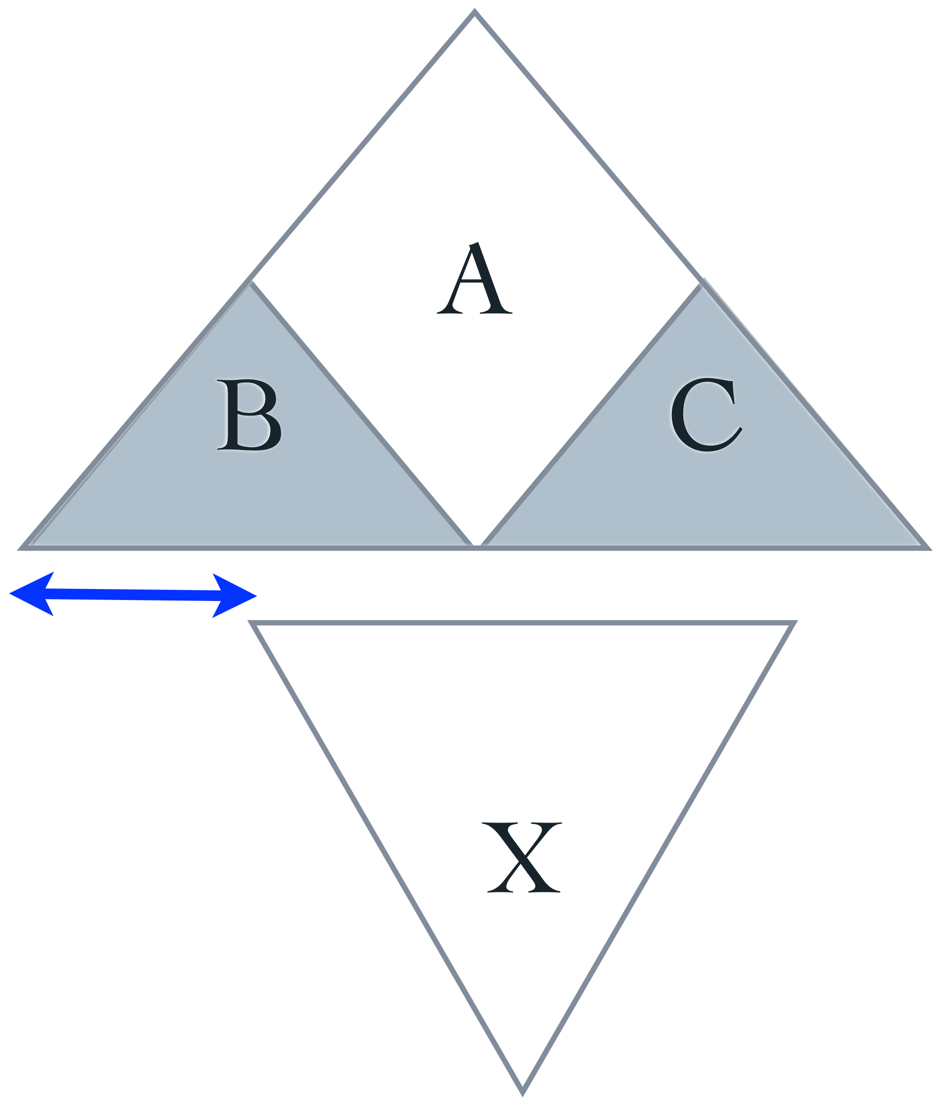
$$A \longrightarrow B\,C$$
$$X \longrightarrow Y\,Z$$

equality test A = X ?



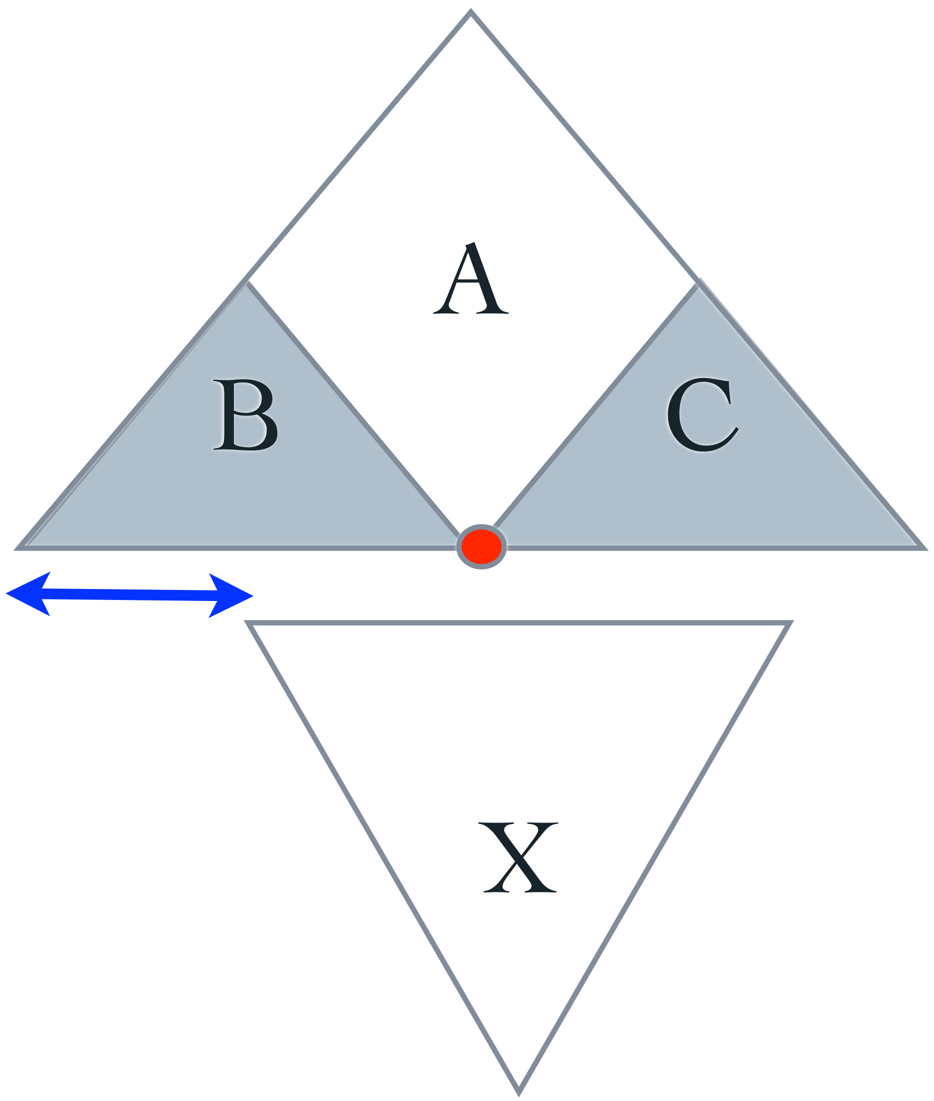
Operations on compressed strings

occurence test



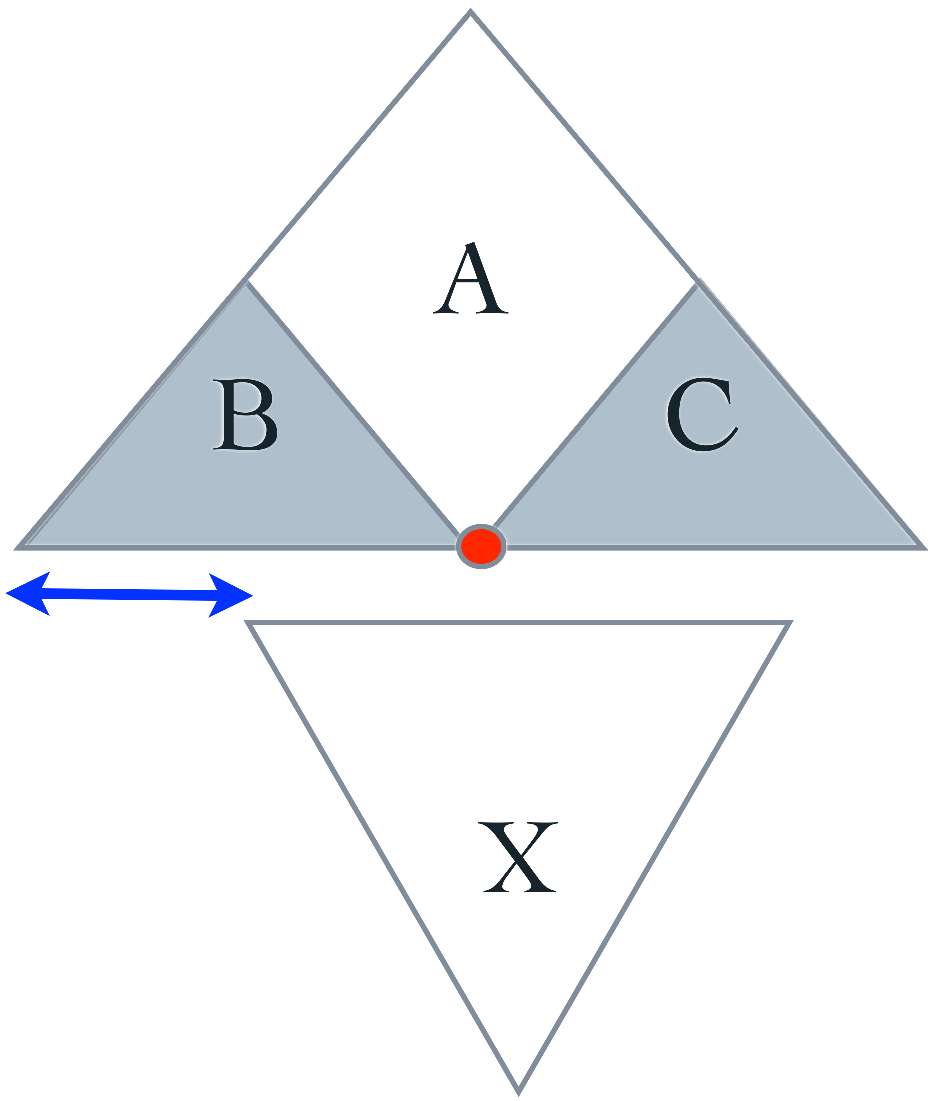
Operations on compressed strings

occurence test



Operations on compressed strings

occurence test

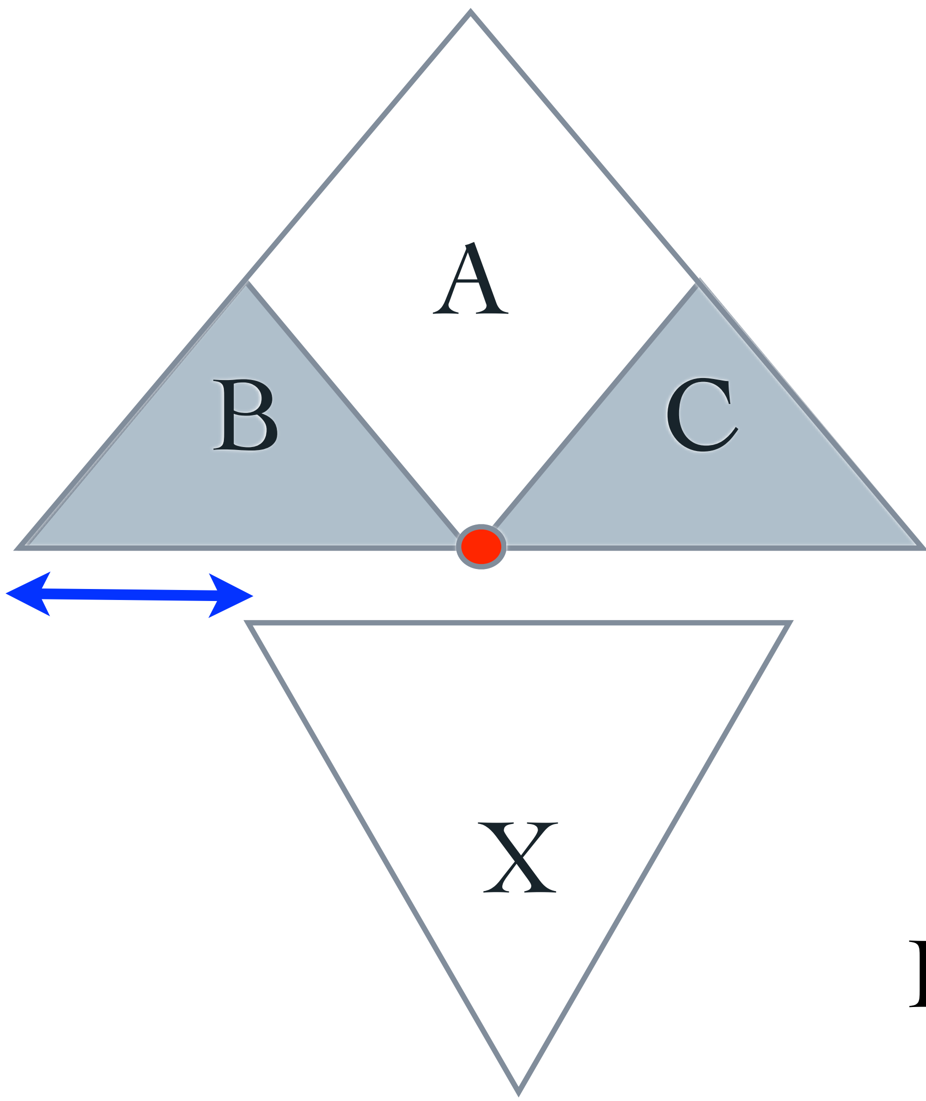


Fact:

The set of occurrences of X in A touching the cutting point is an arithmetic progression

Operations on compressed strings

occurence test

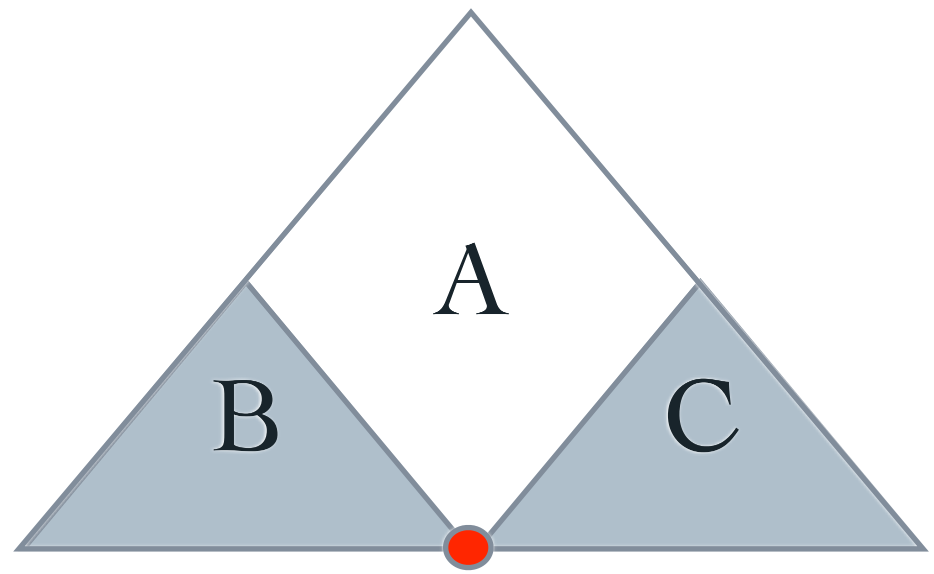


Fact:

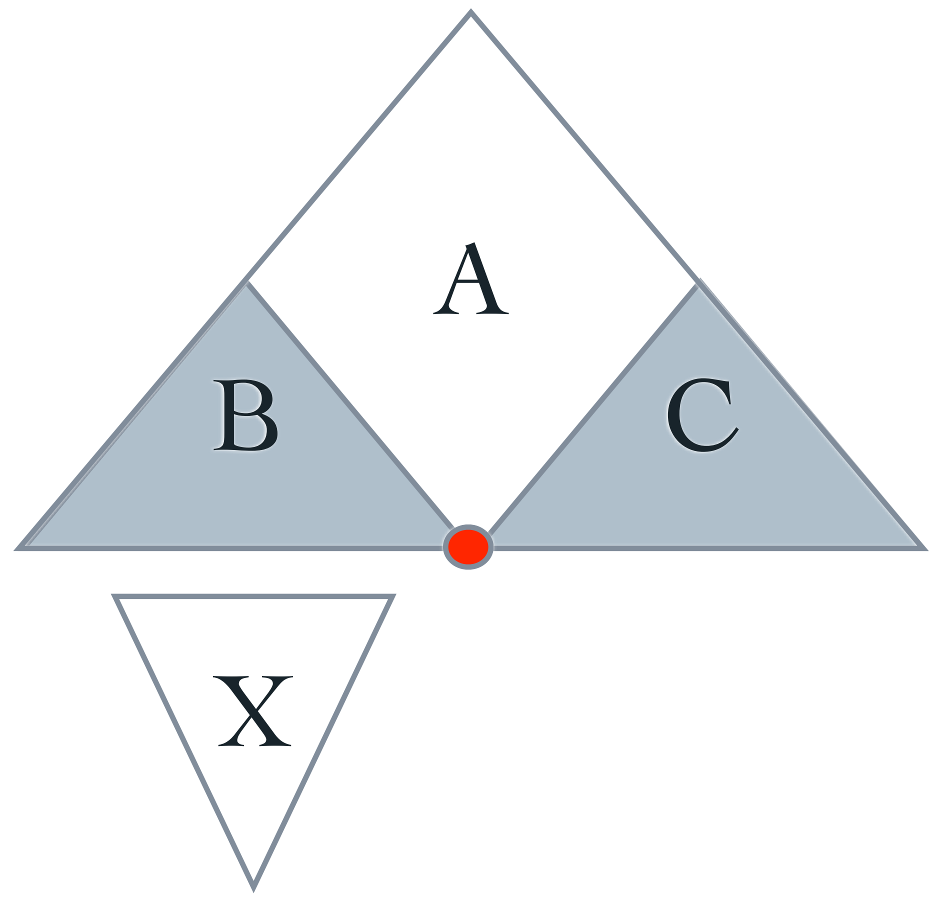
The set of occurrences of X in A touching the cutting point is an arithmetic progression

How to compute the occurrence table?

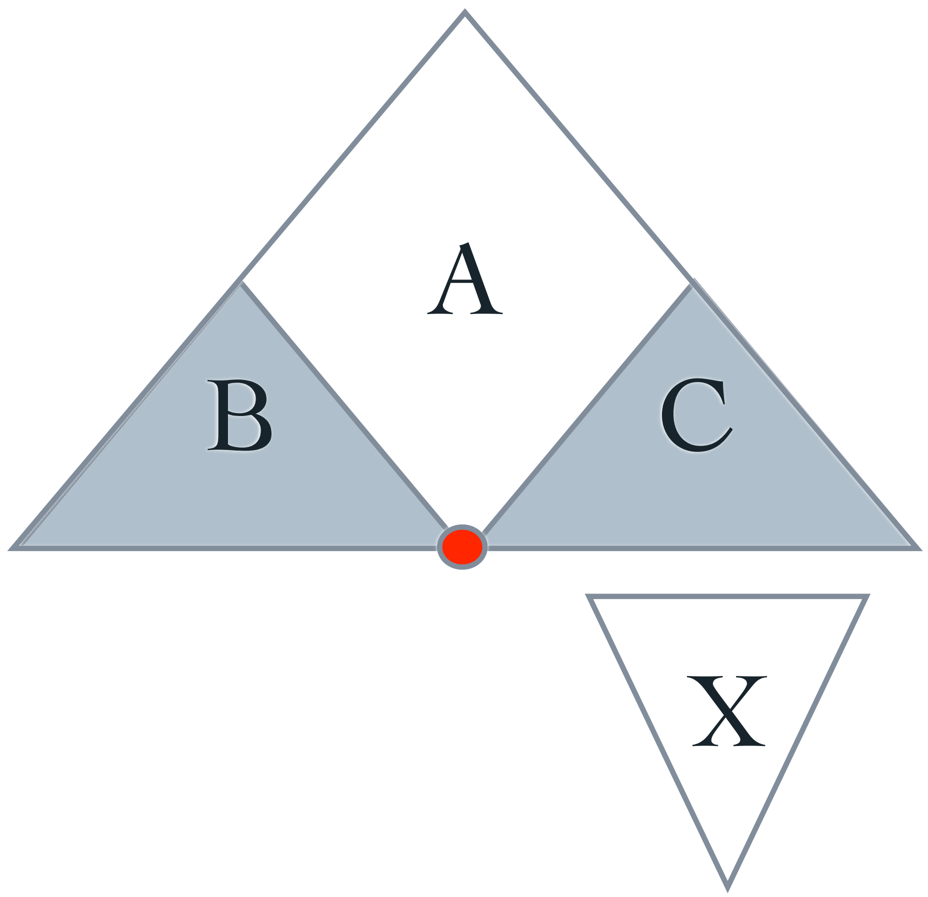
Computing occurrence table



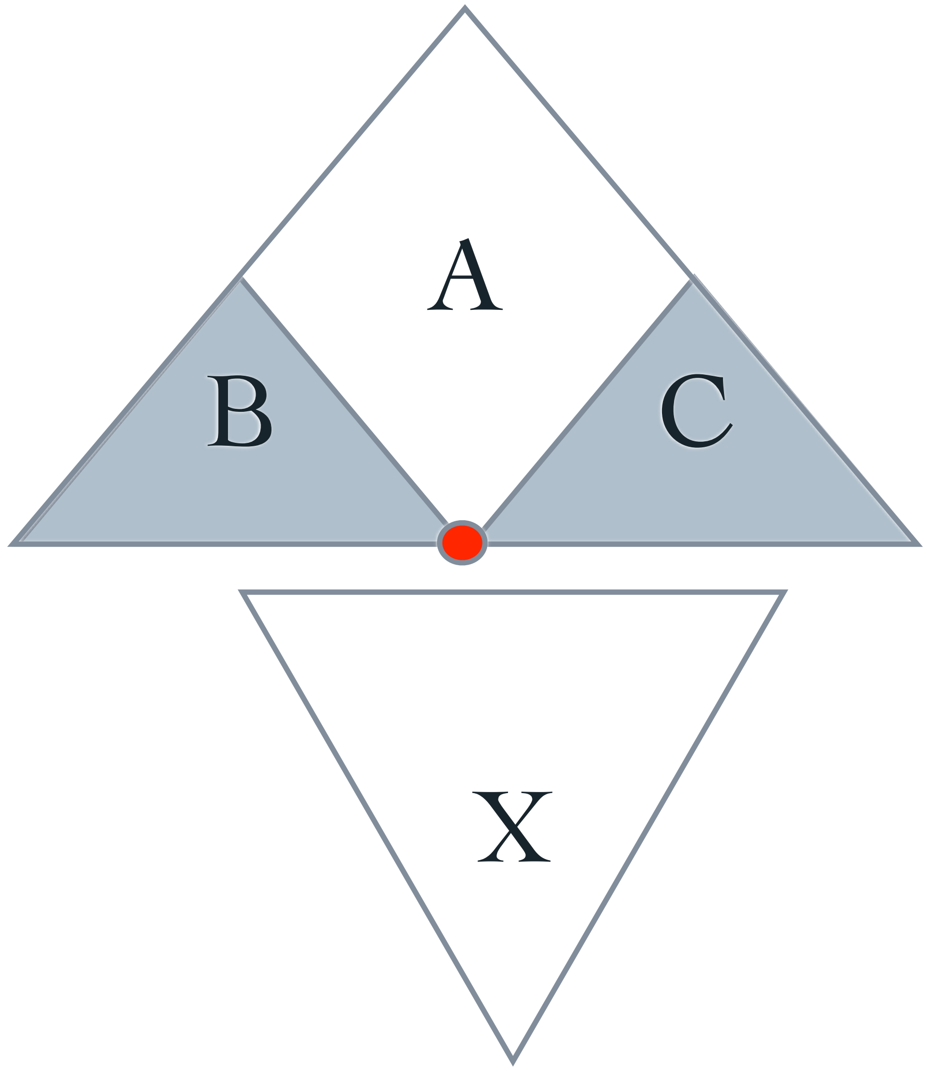
Computing occurrence table



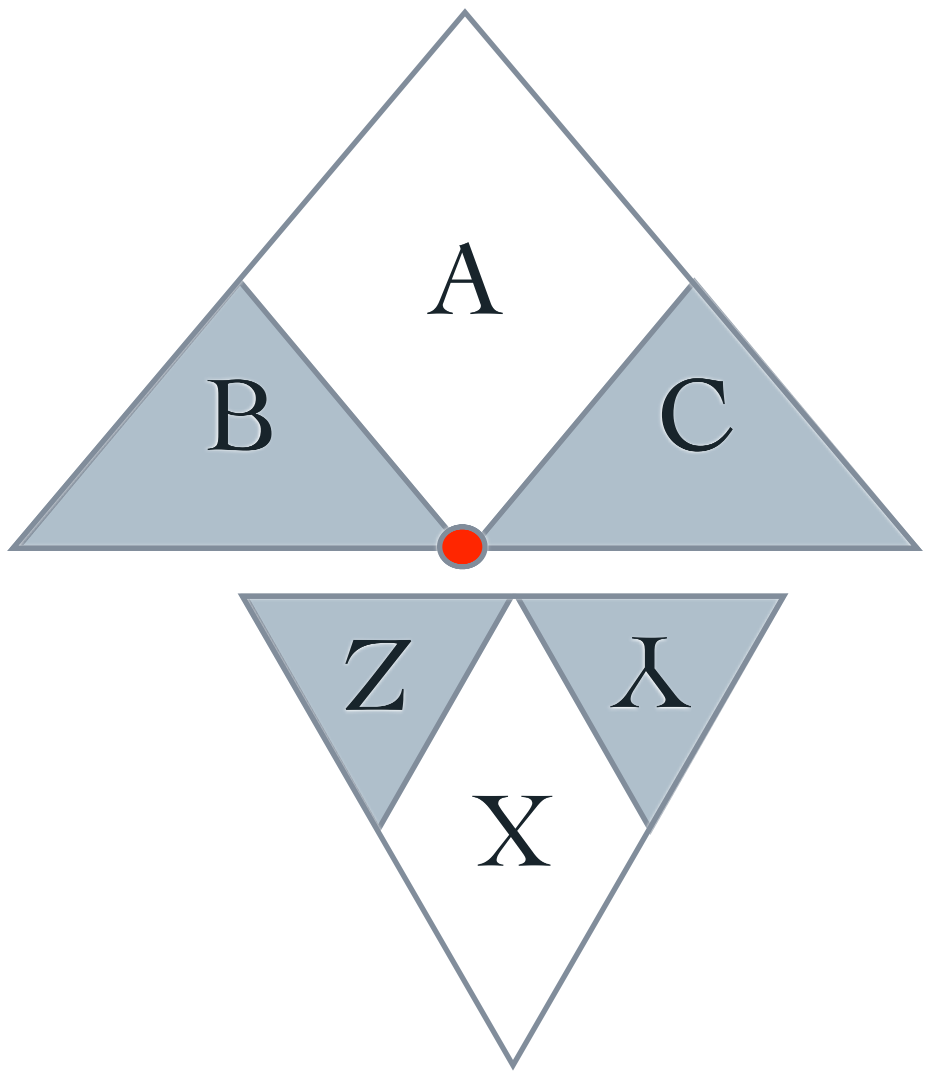
Computing occurrence table



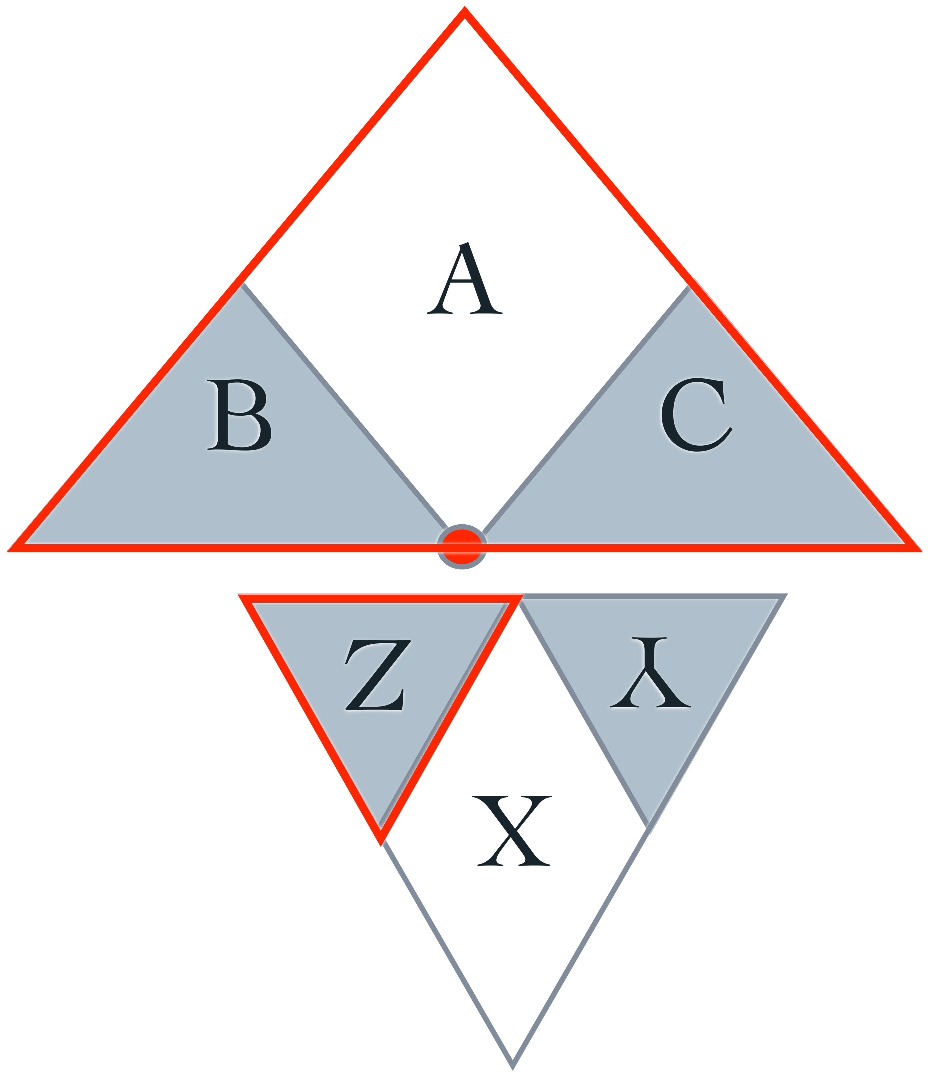
Computing occurrence table



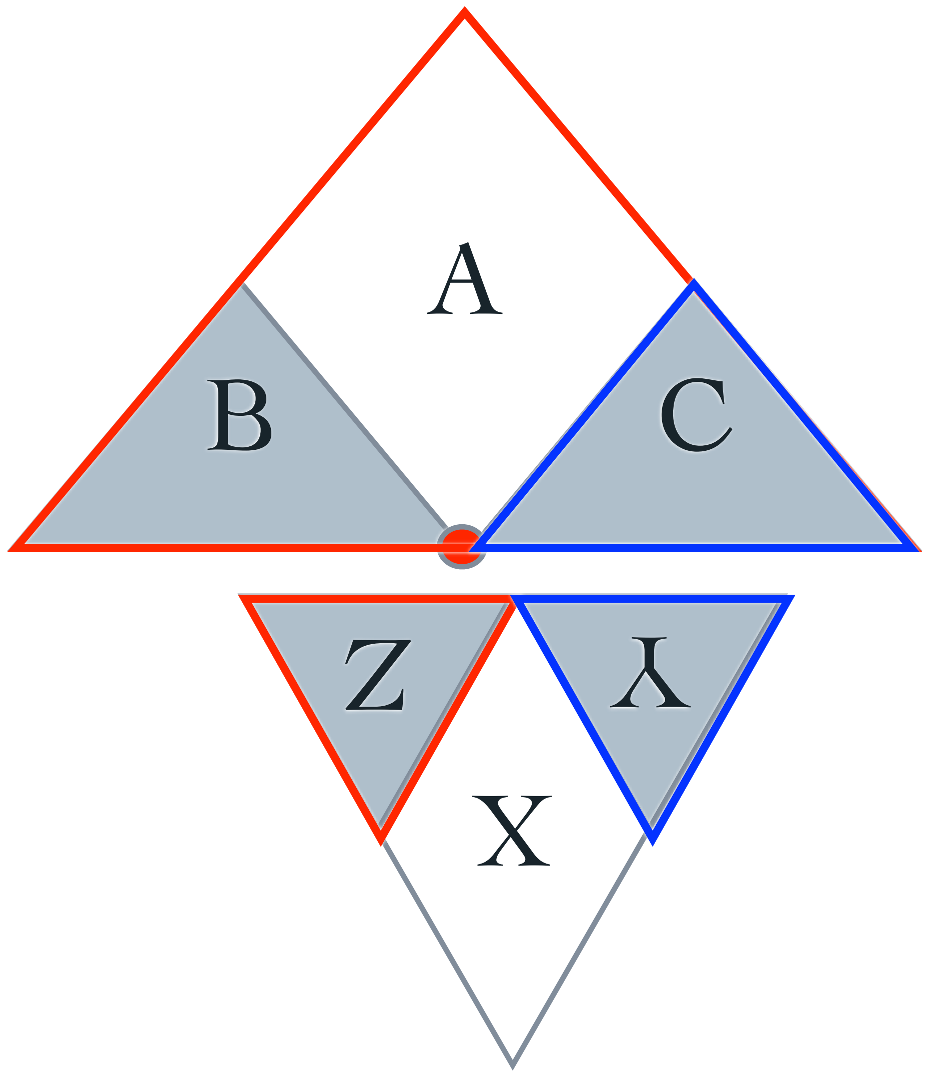
Computing occurrence table



Computing occurrence table



Computing occurrence table



BPA and BPP differ
polynomial-time algorithms

A sequence of over-approximating congruences:

$$\equiv_{B_0} \supseteq \equiv_{B_1} \supseteq \equiv_{B_2} \supseteq \ldots \supseteq \equiv_{B_n} = \sim$$

yielded by an efficient refinement step:

$$B_i \mapsto \mathrm{ref}(B_i) = B_{i+1}$$

BPA and BPP differ
polynomial-time algorithms

A sequence of over-approximating congruences:

$$\equiv_{B_0} \supseteq \equiv_{B_1} \supseteq \equiv_{B_2} \supseteq \ldots \supseteq \equiv_{B_n} = \sim$$

yielded by an efficient refinement step:

$$B_i \mapsto \mathrm{ref}(B_i) = B_{i+1}$$

- different notions of base

BPA and BPP differ
polynomial-time algorithms

A sequence of over-approximating congruences:

$$\equiv_{B_0} \supseteq \equiv_{B_1} \supseteq \equiv_{B_2} \supseteq \ldots \supseteq \equiv_{B_n} = \sim$$

yielded by an efficient refinement step:

$$B_i \mapsto \mathrm{ref}(B_i) = B_{i+1}$$

- different notions of base

$$X = Y\alpha$$
$$X = Z\beta$$
$$\ldots$$

BPA and BPP differ
polynomial-time algorithms

A sequence of over-approximating congruences:

$$\equiv_{B_0} \;\supseteq\; \equiv_{B_1} \;\supseteq\; \equiv_{B_2} \;\supseteq\; \ldots \;\supseteq\; \equiv_{B_n} \;=\; \sim$$

yielded by an efficient refinement step:

$$B_i \;\mapsto\; \mathrm{ref}(B_i) \;=\; B_{i+1}$$

$$\begin{aligned} X &= Y\alpha \\ X &= Z\beta \\ &\ldots \end{aligned}$$

- different notions of base
- different refinement steps

Outline

- Background
    - Context-free graphs and commutative context-free graphs
    - Bisimulation equivalence problem
    - Norm
    - History of the problem
    - Unique decomposition
    - Naive algorithm
- **Efficient algorithm for BPA and BPP**
    - Outline of the algorithm
    - Refinement
    - Efficient computation of refinement for BPA
    - Time-cost analysis
    - Partially-commutative context-free graphs

Outline

- Background
  - Context-free graphs and commutative context-free graphs
  - Bisimulation equivalence problem
  - Norm
  - History of the problem
  - Unique decomposition
  - Naive algorithm
- Efficient algorithm for BPA and BPP
  - **Outline of the algorithm**
  - Refinement
  - Efficient computation of refinement for BPA
  - Time-cost analysis
  - Partially-commutative context-free graphs

Outline of the algorithm

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Novel ideas

Novel ideas

- application of BPP approximation scheme to BPA

Novel ideas

- application of BPP approximation scheme to BPA

In consequence, there is essentially one algorithm for both BPA and BPP

Novel ideas

- application of BPP approximation scheme to BPA
- careful exploitation of unique decomposition

In consequence, there is essentially one algorithm for both BPA and BPP

Novel ideas

- application of BPP approximation scheme to BPA
- careful exploitation of unique decomposition
- efficient operations on compressed strings

In consequence, there is essentially one algorithm for both BPA and BPP

Number of iterations

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Number of iterations

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Fact: The algorithm halts after n iterations.

Number of iterations

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Fact: The algorithm halts after n iterations.

Proof:

Number of iterations

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Fact: The algorithm halts after n iterations.

Proof: Every iteration increases the number of primes.

Number of iterations

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Fact: The algorithm halts after n iterations.

Proof: Every iteration increases the number of primes.

$$B = (P, E) \qquad \mathrm{ref}(B) = (P', E')$$

$$\equiv_{\mathrm{ref}(B)} \subseteq \equiv_B \implies P \subseteq P'$$

Initial base

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Initial base

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Initial base

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

$$|X_1| \;\leq\; |X_2| \;\leq\; \ldots \;\leq\; |X_n|$$

Initial base

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

$$|X_1| \leq |X_2| \leq \ldots \leq |X_n|$$

primes: $P = \{X_1\}$

Initial base

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

$$|X_1| \leq |X_2| \leq \ldots \leq |X_n|$$

primes: $P = \{X_1\}$

decompositions: $E = \{X_i = X_1{}^{|X_i|} \; : \; i \geq 2\}$

Initial base

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

$$|X_1| \;\leq\; |X_2| \;\leq\; \ldots \;\leq\; |X_n|$$

primes: $P \;=\; \{X_1\}$

decompositions: $E \;=\; \{X_i = {X_1}^{|X_i|} \;:\; i \geq 2\}$

correctness: $\sim \;\subseteq\; \equiv_B$

Refinement

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Refinement

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Refinement

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

correctness:

Refinement

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

correctness: $\equiv_{\mathrm{ref}(B)} \;\subseteq\; \equiv_B$

Refinement

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

correctness: $\equiv_{\mathrm{ref}(B)} \;\subseteq\; \equiv_B$

$$\sim \;\subseteq\; \equiv_B \quad\Longrightarrow\quad \sim \;\subseteq\; \equiv_{\mathrm{ref}(B)}$$

Refinement

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

correctness: $\equiv_{\mathrm{ref}(B)} \ \subseteq \ \equiv_B$

$$\sim \ \subseteq \ \equiv_B \quad \Longrightarrow \quad \sim \ \subseteq \ \equiv_{\mathrm{ref}(B)}$$

$$\equiv_{\mathrm{ref}(B)} \ = \ \equiv_B \quad \Longrightarrow \quad \equiv_B = \sim$$

Refinement

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

correctness: $\equiv_{\mathrm{ref}(B)} \subseteq \equiv_B$

$$\sim \subseteq \equiv_B \implies \sim \subseteq \equiv_{\mathrm{ref}(B)}$$

$$\equiv_{\mathrm{ref}(B)} = \equiv_B \implies \equiv_B = \sim$$

Halting condition

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Halting condition

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Halting condition

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Fact: $\mathrm{ref}(B) = B \iff \equiv_{\mathrm{ref}(B)} = \equiv_B$

Halting condition

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Fact: $\mathrm{ref}(B) = B \iff \equiv_{\mathrm{ref}(B)} \; = \; \equiv_B$

Proof:

Halting condition

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Fact: $\mathrm{ref}(B) = B \iff \equiv_{\mathrm{ref}(B)} \; = \; \equiv_B$

Proof: $\equiv_{\mathrm{ref}(B)} \;\subseteq\; \equiv_B$

Halting condition

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Fact: $\mathrm{ref}(B) = B \iff \equiv_{\mathrm{ref}(B)} \; = \; \equiv_B$

Proof: $\equiv_{\mathrm{ref}(B)} \subseteq \equiv_B$

unique decomposition of $\equiv_B$

Outline

- Background
  - Context-free graphs and commutative context-free graphs
  - Bisimulation equivalence problem
  - Norm
  - History of the problem
  - Unique decomposition
  - Naive algorithm

- Efficient algorithm for BPA and BPP
  - Outline of the algorithm
  - **Refinement**
  - Efficient computation of refinement for BPA
  - Time-cost analysis
  - Partially-commutative context-free graphs

What refinement?

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

$$B \mapsto \mathrm{ref}(B)$$

What refinement?

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

$$B \mapsto \mathrm{ref}(B)$$

What refinement?

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

$$\equiv \;\mapsto\; \mathrm{ref}(\equiv)$$

What refinement?

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

$$\equiv \;\mapsto\; \mathrm{ref}(\equiv)$$

$\equiv$ is a congruence with unique decomposition property

$$\Downarrow$$

$\mathrm{ref}(\equiv)$ is a congruence with unique decomposition property

What refinement?

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

$$\equiv \; \mapsto \; \mathrm{ref}(\equiv)$$

$\equiv$ is a congruence with unique decomposition property

$$\Downarrow$$

$\mathrm{ref}(\equiv)$ is a congruence with unique decomposition property

$$\equiv \; \mapsto \; \equiv \cap \exp(\equiv)$$

What refinement?

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

$$\equiv \ \mapsto \ \mathrm{ref}(\equiv)$$

$\equiv$ is a congruence with unique decomposition property

$$\Downarrow$$

$\mathrm{ref}(\equiv)$ is a congruence with unique decomposition property

$$\cancel{\equiv \ \mapsto \ \equiv} \cap \exp(\equiv)$$

$$\equiv \; \xcancel{\mapsto \; \equiv} \; \cap \exp(\equiv)$$

$$\equiv \;\; \cancel{\mapsto \;\; \equiv} \;\; \cap \; \exp(\equiv)$$

Example:

$$
\begin{array}{cccccccccccc}
X_1 & \xrightarrow{a} & \varepsilon & X_2 & \xrightarrow{a} & X_1 & X_2 & \xrightarrow{b} & X_1X_1 & X_3 & \xrightarrow{a} & X_2 \\
X_4 & \xrightarrow{a} & X_3X_2 & X_4 & \xrightarrow{b} & X_4 & X_5 & \xrightarrow{a} & X_4 & X_5 & \xrightarrow{a} & X_3X_3
\end{array}
$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\cancel{\equiv \;\mapsto\; \equiv \cap \exp(\equiv)}$$

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\text{let } \alpha \equiv \beta \iff |\alpha| = |\beta|$$

$$\cancel{\equiv \;\; \mapsto \;\; \equiv \;\cap\; \exp(\equiv)}$$

Example:

$$\begin{array}{cccccccccccc}
X_1 & \xrightarrow{a} & \varepsilon & X_2 & \xrightarrow{a} & X_1 & X_2 & \xrightarrow{b} & X_1X_1 & X_3 & \xrightarrow{a} & X_2 \\
X_4 & \xrightarrow{a} & X_3X_2 & X_4 & \xrightarrow{b} & X_4 & X_5 & \xrightarrow{a} & X_4 & X_5 & \xrightarrow{a} & X_3X_3
\end{array}$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\text{let } \alpha \;\equiv\; \beta \quad\Longleftrightarrow\quad |\alpha| \;=\; |\beta|$$

$$X_1 \text{ and } X_2 \text{ are primes wrt. } \;\equiv \cap \exp(\equiv)$$

$$\equiv \;\cancel{\mapsto\; \equiv}\; \cap \exp(\equiv)$$

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

let $\alpha \equiv \beta \iff |\alpha| = |\beta|$

X_1 and X_2 are primes wrt. $\equiv \cap \exp(\equiv)$ $\qquad (X_2,\, X_1 X_1) \notin \exp(\equiv)$

$$\cancel{\equiv \;\; \mapsto \;\; \equiv \; \cap \; \exp(\equiv)}$$

Example:

$$\begin{array}{cccccccccccc}
X_1 & \xrightarrow{a} & \varepsilon & X_2 & \xrightarrow{a} & X_1 & X_2 & \xrightarrow{b} & X_1X_1 & X_3 & \xrightarrow{a} & X_2 \\
X_4 & \xrightarrow{a} & X_3X_2 & X_4 & \xrightarrow{b} & X_4 & X_5 & \xrightarrow{a} & X_4 & X_5 & \xrightarrow{a} & X_3X_3
\end{array}$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\text{let } \alpha \;\equiv\; \beta \;\iff\; |\alpha| \;=\; |\beta|$$

X_1 and X_2 are primes wrt. $\equiv \cap \exp(\equiv)$

$$\cancel{\equiv \;\mapsto\; \equiv} \cap \exp(\equiv)$$

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

let $\alpha \equiv \beta \iff |\alpha| = |\beta|$

X_1 and X_2 are primes wrt. $\equiv \cap \exp(\equiv)$

$$(X_1\, X_1\, X_1,\ X_1\, X_2) \in \equiv \cap \exp(\equiv)$$

Right refinement

$$\boxed{\equiv \;\mapsto\; \mathbf{gnrb}(\equiv \cap \exp(\equiv))}$$

Right refinement

$$\boxed{\equiv \;\mapsto\; \mathbf{gnrb}(\equiv \cap \exp(\equiv))}$$

greatest
norm-reducing
bisimulation
contained in
$\equiv \cap \exp(\equiv)$

Norm-reducing bisimulation

Norm-reducing bisimulation

norm-reducing transitions:

$$\alpha \xrightarrow{a} \beta \qquad |\beta| < |\alpha| \qquad\qquad |\beta| = |\alpha| - 1$$

Norm-reducing bisimulation

norm-reducing transitions:

$$\alpha \xrightarrow{a} \beta \qquad |\beta| < |\alpha| \qquad\qquad |\beta| = |\alpha| - 1$$

norm-reducing bisimulation expansion n-r-exp(_)

Norm-reducing bisimulation

norm-reducing transitions:

$$\alpha \xrightarrow{a} \beta \qquad |\beta| < |\alpha| \qquad\qquad |\beta| = |\alpha| - 1$$

norm-reducing bisimulation expansion n-r-exp(_)

R is a n-r-bisimulation if $R \subseteq$ n-r-exp(R)

Norm-reducing bisimulation

norm-reducing transitions:

$$\alpha \xrightarrow{a} \beta \qquad |\beta| < |\alpha| \qquad\qquad |\beta| = |\alpha| - 1$$

norm-reducing bisimulation expansion n-r-exp(_)

R is a n-r-bisimulation if $R \subseteq$ n-r-exp(R)

Greatest norm-reducing bisimulation contained in $\equiv$:

$$\mathbf{gnrb}(\equiv) \stackrel{\text{def}}{=} \bigcup \{R : R \text{ is a norm-reducing bisimulation, } R \subseteq \equiv\}$$

Right refinement

$$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$$

Right refinement

$$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$$

Adaptation of Bisimulation Game:

Right refinement

$$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$$

Adaptation of Bisimulation Game:

- positions always belong to $\equiv$

Right refinement

$$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$$

Adaptation of Bisimulation Game:

- positions always belong to $\equiv$ $\qquad \alpha \ \equiv \ \beta$

Right refinement

$$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$$

Adaptation of Bisimulation Game:

- positions always belong to $\equiv$ (otherwise Spoiler wins)

$$\alpha \equiv \beta$$

Right refinement

$$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$$

Adaptation of Bisimulation Game:

- positions always belong to $\equiv$ $\qquad \alpha \equiv \beta$

  (otherwise Spoiler wins)
- Spoiler may freely use norm-reducing transitions

Right refinement

$$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$$

Adaptation of Bisimulation Game:

- positions always belong to $\equiv$ $\qquad\qquad \alpha \equiv \beta$

  (otherwise Spoiler wins)

- Spoiler may freely use norm-reducing transitions
- Spoiler may use an arbitrary transition once,

Right refinement

$$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$$

Adaptation of Bisimulation Game:

- positions always belong to $\equiv$ $\qquad \alpha \equiv \beta$

  (otherwise Spoiler wins)
- Spoiler may freely use norm-reducing transitions
- Spoiler may use an arbitrary transition once,

  thus declaring the move to be the last one

Correctness of refinement

$$\boxed{\equiv \;\mapsto\; \mathbf{gnrb}(\equiv \cap \exp(\equiv))}$$

Correctness of refinement

$$\boxed{\equiv \;\mapsto\; \mathbf{gnrb}(\equiv \cap \exp(\equiv))}$$

Theorem:

$\equiv$ is a congruence with unique decomposition property

$$\Downarrow$$

$\mathbf{gnrb}(\equiv \cap \exp(\equiv))$ is a congruence
with unique decomposition property

Proof (both for BPA and BPP)

Theorem:

$\equiv$ is a congruence with unique decomposition property

$$\Downarrow$$

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ is a congruence with unique decomposition property

Proof (both for BPA and BPP)

Theorem:

$\equiv$ is a congruence with unique decomposition property
is norm-preserving

$\Downarrow$

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ is a congruence
with unique decomposition property

Proof (both for BPA and BPP)

Theorem:

$\equiv$ is a congruence with unique decomposition property

$$\Downarrow$$

$\mathbf{gnrb}(\equiv \ \cap \ \exp(\equiv))$ is a congruence with unique decomposition property

Proof (both for BPA and BPP)

Theorem:

$\equiv$ is a congruence with unique decomposition property

$$\Downarrow$$

$\mathbf{gnrb}(\equiv \ \cap \ \mathrm{exp}(\equiv))$ is a congruence
with unique decomposition property

is a congruence

Proof (both for BPA and BPP)

Theorem:

$\equiv$ is a congruence with unique decomposition property

$$\Downarrow$$

$\mathbf{gnrb}(\equiv \cap \exp(\equiv))$ is a congruence with unique decomposition property

Proof (both for BPA and BPP)

Theorem:

$\equiv$ is a congruence with unique decomposition property

$$\Downarrow$$

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ is a congruence
with unique decomposition property

Proof (both for BPA and BPP)

Theorem:

$\equiv$ is a congruence with unique decomposition property

$$\Downarrow$$

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ is a congruence
with unique decomposition property

Lemma:

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ has unique decomposition property

Proof (both for BPA and BPP)

Theorem:

$\equiv$ is a congruence with unique decomposition property

$$\Downarrow$$

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ is a congruence with unique decomposition property

Lemma:

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ has unique decomposition property

a crucial technical observation

Lemma:

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPA:

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPA:

Existence of prime decomposition by induction on norm

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPA:

Existence of prime decomposition by induction on norm

Cancellation: $\alpha\gamma \sim \beta\gamma \implies \alpha \sim \beta$

Lemma:

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPA:

Existence of prime decomposition by induction on norm

Cancellation: $\alpha\gamma \ \sim \ \beta\gamma \quad \Longrightarrow \quad \alpha \ \sim \ \beta$

$\{(\alpha, \beta) \ : \ (\alpha\gamma, \beta\gamma) \ \in \ \mathrm{gnrb}(\equiv \ \cap \exp(\equiv))\}$

is a norm-reducing bisimulation contained in $\equiv \ \cap \exp(\equiv)$

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPA:

Existence of prime decomposition by induction on norm

Cancellation: $\alpha\gamma \sim \beta\gamma \implies \alpha \sim \beta$

$\{(\alpha, \beta) : (\alpha\gamma, \beta\gamma) \in \text{gnrb}(\equiv\ \cap \exp(\equiv))\}$

is a norm-reducing bisimulation contained in $\equiv\ \cap \exp(\equiv)$

Uniqueness of prime decomposition by induction on norm

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPA:

Existence of prime decomposition by induction on norm

Cancellation: $\alpha\gamma \sim \beta\gamma \implies \alpha \sim \beta$

$\{(\alpha, \beta) : (\alpha\gamma, \beta\gamma) \in \mathrm{gnrb}(\equiv\ \cap \exp(\equiv))\}$

is a norm-reducing bisimulation contained in $\equiv\ \cap \exp(\equiv)$

Uniqueness of prime decomposition by induction on norm

$(P_1 \ldots P_n, Q_1 \ldots Q_m) \in \mathrm{gnrb}(\equiv\ \cap \exp(\equiv))$

Lemma:

$\mathbf{gnrb}(\equiv \; \cap \exp(\equiv))$ has unique decomposition property

Proof for BPA:

Existence of prime decomposition by induction on norm

Cancellation: $\alpha\,\gamma \;\sim\; \beta\,\gamma \quad\Longrightarrow\quad \alpha \;\sim\; \beta$

$\{(\alpha, \beta) \;:\; (\alpha\,\gamma, \beta\,\gamma) \;\in\; \mathrm{gnrb}(\equiv \; \cap \exp(\equiv))\}$

is a norm-reducing bisimulation contained in $\equiv \; \cap \exp(\equiv)$

Uniqueness of prime decomposition by induction on norm

$(P_1 \ldots P_n, Q_1 \ldots Q_m) \;\in\; \mathrm{gnrb}(\equiv \; \cap \exp(\equiv))$

$n, m \neq 1$

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPA:

Existence of prime decomposition by induction on norm

Cancellation: $\alpha\,\gamma\ \sim\ \beta\,\gamma \quad \Longrightarrow \quad \alpha\ \sim\ \beta$

$\{(\alpha, \beta)\ :\ (\alpha\,\gamma, \beta\,\gamma)\ \in\ \mathrm{gnrb}(\equiv\ \cap \exp(\equiv))\}$

is a norm-reducing bisimulation contained in $\equiv\ \cap \exp(\equiv)$

Uniqueness of prime decomposition by induction on norm

$(P_1\ \ldots\ P_n, Q_1\ \ldots\ Q_m)\ \in\ \mathrm{gnrb}(\equiv\ \cap \exp(\equiv))$

$n, m \neq 1$

$P_n = Q_m$ by induction assumption

Lemma:

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPA:

Existence of prime decomposition by induction on norm

Cancellation: $\alpha\,\gamma \ \sim \ \beta\,\gamma \quad \Longrightarrow \quad \alpha \ \sim \ \beta$

$\{(\alpha, \beta) \ : \ (\alpha\,\gamma, \beta\,\gamma) \ \in \ \mathrm{gnrb}(\equiv \ \cap \exp(\equiv))\}$

is a norm-reducing bisimulation contained in $\equiv \ \cap \exp(\equiv)$

Uniqueness of prime decomposition by induction on norm

$(P_1 \ \ldots \ P_n, Q_1 \ \ldots \ Q_m) \ \in \ \mathrm{gnrb}(\equiv \ \cap \exp(\equiv))$

$n, m \neq 1$

$P_n = Q_m$ by induction assumption

the lemma follows by cancellation and induction assumption

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

Lemma:

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

$$(X_1^{a_1} \ldots X_n^{a_n}, X_1^{b_1} \ldots X_n^{b_n}) \ \in \ \mathrm{gnrb}(\equiv \ \cap \exp(\equiv))$$

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

$(X_1^{a_1} \ldots X_n^{a_n}, X_1^{b_1} \ldots X_n^{b_n}) \in \text{gnrb}(\equiv\ \cap \exp(\equiv))$ $\quad |X_1| \leq |X_2| \leq \ldots \leq |X_n|$

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

$(X_1^{a_1} \dots X_n^{a_n}, X_1^{b_1} \dots X_n^{b_n}) \ \in \ \mathrm{gnrb}(\equiv\ \cap \exp(\equiv))$ $\qquad |X_1| \ \leq \ |X_2| \ \leq \ \dots \ \leq \ |X_n|$

let $m = \max\{i \ : \ a_i \neq b_i\}$

Lemma:

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

$(X_1^{a_1} \ \ldots \ X_n^{a_n}, X_1^{b_1} \ \ldots \ X_n^{b_n}) \ \in \ \mathrm{gnrb}(\equiv \ \cap \exp(\equiv))$ $\quad |X_1| \ \leq \ |X_2| \ \leq \ \ldots \ \leq \ |X_n|$

let $m = \max\{i \ : \ a_i \neq b_i\}$ $\quad a_m > b_m$

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

$(X_1^{a_1}\ \ldots X_n^{a_n}, X_1^{b_1}\ \ldots X_n^{b_n}) \in \text{gnrb}(\equiv\ \cap \exp(\equiv))$ $\quad |X_1| \leq |X_2| \leq \ldots \leq |X_n|$

let $m = \max\{i\ :\ a_i \neq b_i\}$ $\quad a_m > b_m$

Case I: $\quad a_i > 0$ for some $i < m$

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

$(X_1^{a_1} \ldots X_n^{a_n}, X_1^{b_1} \ldots X_n^{b_n}) \in \mathrm{gnrb}(\equiv\ \cap \exp(\equiv))$ $\quad |X_1| \leq |X_2| \leq \ldots \leq |X_n|$

let $m = \max\{i : a_i \neq b_i\}$ $\quad a_m > b_m$

Case I: $\quad a_i > 0$ for some $i < m$ $\quad$ Spoiler moves on the left

Lemma:

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

$(X_1^{a_1} \ \ldots X_n^{a_n}, X_1^{b_1} \ \ldots X_n^{b_n}) \ \in \ \mathrm{gnrb}(\equiv \ \cap \exp(\equiv))$ $\quad |X_1| \ \leq \ |X_2| \ \leq \ \ldots \ \leq \ |X_n|$

let $m = \max\{i \ : \ a_i \neq b_i\}$ $\quad a_m > b_m$

Case I: $\quad a_i > 0$ for some $i < m$ $\quad$ Spoiler moves on the left

Case II: $\quad a_i > 0$ for some $i > m$

Lemma:

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

$(X_1^{a_1} \ldots X_n^{a_n}, X_1^{b_1} \ldots X_n^{b_n}) \in \text{gnrb}(\equiv \ \cap \exp(\equiv))$ $\quad |X_1| \leq |X_2| \leq \ldots \leq |X_n|$

let $m = \max\{i \ : \ a_i \neq b_i\}$ $\qquad a_m > b_m$

Case I: $\quad a_i > 0$ for some $i < m$ $\quad$ Spoiler moves on the left

Case II: $\quad a_i > 0$ for some $i > m$ $\quad$ Spoiler moves on the left

Lemma:

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

$(X_1^{a_1} \ldots X_n^{a_n}, X_1^{b_1} \ldots X_n^{b_n}) \ \in \ \mathrm{gnrb}(\equiv \ \cap \exp(\equiv))$ $\quad |X_1| \ \leq \ |X_2| \ \leq \ \ldots \ \leq \ |X_n|$

let $m = \max\{i \ : \ a_i \neq b_i\}$ $\quad a_m > b_m$

Case I: $a_i > 0$ for some $i < m$ Spoiler moves on the left

Case II: $a_i > 0$ for some $i > m$ Spoiler moves on the left

Case III: $a_m \geq 2$

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

$(X_1^{a_1} \ldots X_n^{a_n}, X_1^{b_1} \ldots X_n^{b_n}) \in \text{gnrb}(\equiv\ \cap \exp(\equiv))$ $\quad |X_1| \leq |X_2| \leq \ldots \leq |X_n|$

let $m = \max\{i\ :\ a_i \neq b_i\}$ $\quad a_m > b_m$

Case I: $a_i > 0$ for some $i < m$ Spoiler moves on the left

Case II: $a_i > 0$ for some $i > m$ Spoiler moves on the left

Case III: $a_m \geq 2$

$b_m = 0$

Lemma:

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

$(X_1^{a_1} \ldots X_n^{a_n}, X_1^{b_1} \ldots X_n^{b_n}) \in \mathrm{gnrb}(\equiv \ \cap \exp(\equiv))$ $\quad |X_1| \leq |X_2| \leq \ldots \leq |X_n|$

let $m = \max\{i \ : \ a_i \neq b_i\}$ $\quad a_m > b_m$

Case I: $a_i > 0$ for some $i < m$ Spoiler moves on the left

Case II: $a_i > 0$ for some $i > m$ Spoiler moves on the left

Case III: $a_m \geq 2$

$b_m = 0$ Spoiler moves on the left

Lemma:

$\mathbf{gnrb}(\equiv\ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

$(X_1^{a_1} \ldots X_n^{a_n}, X_1^{b_1} \ldots X_n^{b_n}) \in \mathrm{gnrb}(\equiv\ \cap \exp(\equiv))$ $\quad |X_1| \leq |X_2| \leq \ldots \leq |X_n|$

let $m = \max\{i : a_i \neq b_i\}$ $\quad a_m > b_m$

Case I: $a_i > 0$ for some $i < m$ Spoiler moves on the left

Case II: $a_i > 0$ for some $i > m$ Spoiler moves on the left

Case III: $a_m \geq 2$

$b_m = 0$ Spoiler moves on the left

$b_m = 1$

Lemma:

$\mathbf{gnrb}(\equiv \ \cap \exp(\equiv))$ has unique decomposition property

Proof for BPP:

Existence of prime decomposition by induction on norm

Uniqueness of prime decomposition by induction on norm

$(X_1^{a_1} \ \ldots \ X_n^{a_n}, X_1^{b_1} \ \ldots \ X_n^{b_n}) \ \in \ \mathrm{gnrb}(\equiv \ \cap \exp(\equiv))$ $\quad |X_1| \ \leq \ |X_2| \ \leq \ \ldots \ \leq \ |X_n|$

let $m = \max\{i \ : \ a_i \neq b_i\}$ $\quad a_m > b_m$

Case I: $a_i > 0$ for some $i < m$ Spoiler moves on the left

Case II: $a_i > 0$ for some $i > m$ Spoiler moves on the left

Case III: $a_m \geq 2$

$b_m = 0$ Spoiler moves on the left

$b_m = 1$ Spoiler moves on the right

One common algorithm.

What about one common proof?

$$\begin{tikzcd} \text{unique decomposition} \arrow[d, Rightarrow, shift left=3, "\text{BPP}"] \\ \text{cancellation} \arrow[u, Rightarrow, shift left=3, "\text{BPA}"] \end{tikzcd}$$

One common algorithm.

What about one common proof?

possible in the setting of
partially-commmutative
context-free graphs

unique decomposition

BPA $\Uparrow$ $\Downarrow$ BPP

cancellation

Outline

- Background
    - Context-free graphs and commutative context-free graphs
    - Bisimulation equivalence problem
    - Norm
    - History of the problem
    - Unique decomposition
    - Naive algorithm
- Efficient algorithm for BPA and BPP
    - Outline of the algorithm
    - Refinement
    - **Efficient computation of refinement for BPA**
    - Time-cost analysis
    - Partially-commutative context-free graphs

Computation of refinement

$$\boxed{\equiv \;\mapsto\; \mathbf{gnrb}(\equiv \,\cap\, \exp(\equiv))}$$

Fact: $(\alpha, \beta) \in \mathbf{gnrb}(\equiv) \iff \alpha \equiv \beta$ and $(\alpha, \beta) \in \text{n-r-exp}(\mathbf{gnrb}(\equiv))$

Computation of refinement

$$\boxed{\equiv \;\mapsto\; \mathbf{gnrb}(\equiv \;\cap\; \exp(\equiv))}$$

Fact: $(\alpha, \beta) \in \mathbf{gnrb}(\equiv) \iff \alpha \equiv \beta$ and $(\alpha, \beta) \in \text{n-r-exp}(\mathbf{gnrb}(\equiv))$

fixpoint characterization

Computation of refinement

$$\boxed{\equiv \;\mapsto\; \mathbf{gnrb}(\equiv \,\cap\, \exp(\equiv))}$$

Fact: $(\alpha, \beta) \in \mathbf{gnrb}(\equiv) \iff \alpha \equiv \beta$ and $(\alpha, \beta) \in \text{n-r-exp}(\mathbf{gnrb}(\equiv))$

Proof:

Computation of refinement

$$\boxed{\equiv \;\mapsto\; \mathbf{gnrb}(\equiv \,\cap\, \exp(\equiv))}$$

Fact: $(\alpha, \beta) \in \mathbf{gnrb}(\equiv) \iff \alpha \equiv \beta$ and $(\alpha, \beta) \in \text{n-r-exp}(\mathbf{gnrb}(\equiv))$

Proof: The following relation is an n-r-bisimulation:

Computation of refinement

$$\boxed{\equiv \ \mapsto \ \ \mathbf{gnrb}(\equiv \ \cap \exp(\equiv))}$$

Fact: $(\alpha, \beta) \in \mathbf{gnrb}(\equiv) \quad \Longleftrightarrow \quad \alpha \ \equiv \beta \ \text{ and } \ (\alpha, \beta) \in \text{n-r-exp}(\mathbf{gnrb}(\equiv))$

Proof: The following relation is an n-r-bisimulation:

$$\{(\alpha, \beta) : \alpha \ \equiv \ \beta \ \text{ and } \ (\alpha, \beta) \in \text{n-r-exp}(\mathbf{gnrb}(\equiv))\}$$

Computation of refinement

$$\boxed{\equiv \ \mapsto \quad \mathbf{gnrb}(\equiv \ \cap \exp(\equiv))}$$

Fact: $(\alpha, \beta) \in \mathbf{gnrb}(\equiv) \quad \Longleftrightarrow \quad \alpha \ \equiv \beta \ \text{ and } \ (\alpha, \beta) \in \text{n-r-exp}(\mathbf{gnrb}(\equiv))$

Proof: The following relation is an n-r-bisimulation:

$$\{(\alpha, \beta) : \alpha \ \equiv \ \beta \ \text{ and } \ (\alpha, \beta) \in \text{n-r-exp}(\mathbf{gnrb}(\equiv))\}$$

Corollary:

Computation of refinement

$$\boxed{\equiv \;\mapsto\; \mathbf{gnrb}(\equiv \;\cap \exp(\equiv))}$$

Fact: $(\alpha, \beta) \in \mathbf{gnrb}(\equiv) \iff \alpha \equiv \beta$ and $(\alpha, \beta) \in$ n-r-exp$(\mathbf{gnrb}(\equiv))$

Proof: The following relation is an n-r-bisimulation:

$$\{(\alpha, \beta) : \alpha \equiv \beta \text{ and } (\alpha, \beta) \in \text{n-r-exp}(\mathbf{gnrb}(\equiv))\}$$

Corollary: Let $B' = \mathrm{ref}(B)$. Then $(X_i, \alpha) \in \equiv_{B'}$ iff

Computation of refinement

$$\boxed{\equiv \;\mapsto\; \mathbf{gnrb}(\equiv \;\cap\; \exp(\equiv))}$$

Fact: $(\alpha, \beta) \in \mathbf{gnrb}(\equiv) \iff \alpha \equiv \beta$ and $(\alpha, \beta) \in \text{n-r-exp}(\mathbf{gnrb}(\equiv))$

Proof: The following relation is an n-r-bisimulation:

$$\{(\alpha, \beta) : \alpha \equiv \beta \text{ and } (\alpha, \beta) \in \text{n-r-exp}(\mathbf{gnrb}(\equiv))\}$$

Corollary: Let $B' = \text{ref}(B)$. Then $(X_i, \alpha) \in \equiv_{B'}$ iff

$(a) \quad (X_i, \alpha) \in \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

Algorithm 1 Naive algorithm

initialise $B = (P, E)$ as the base of $\equiv_0$;
$P' := P$;
repeat
  $E' := \emptyset$;
  for all $X_i \in (\{X_2, \ldots, X_n\} \setminus P)$ **do**
    for all α such that $|\alpha| = |X_i|$ **do**
      if $\texttt{test}_{(P',E')}(X_i, \alpha)$ **then**
        $E' := E' \cup \{(X_i = \alpha)\}$;
        break the inner for loop;
      end if
    end for
    if the inner for loop not broken **then**
      $P' := P' \cup \{X_i\}$;
    end if
  end for
  $P := P'$; $E := E'$;
until P does not change

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Algorithm 1 Naive algorithm

initialise $B = (P, E)$ as the base of $\equiv_0$;
$P' := P$;
repeat
  $E' := \emptyset$;
  for all $X_i \in (\{X_2, \ldots, X_n\} \setminus P)$ **do**
    for all α such that $|\alpha| = |X_i|$ **do**
      if $\mathtt{test}_{(P',E')}(X_i, \alpha)$ **then**
        $E' := E' \cup \{(X_i = \alpha)\}$;
        break the inner for loop;
      end if
    end for
    if the inner for loop not broken **then**
      $P' := P' \cup \{X_i\}$;
    end if
  end for
  $P := P'$; $E := E'$;
until P does not change

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr\text{-}exp}(\equiv_{B'})$

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Algorithm 1 Naive algorithm

initialise $B = (P, E)$ as the base of $\equiv_0$;
$P' := P$;
repeat
  $E' := \emptyset$;
  for all $X_i \in (\{X_2, \ldots, X_n\} \setminus P)$ **do**
    for all α such that $|\alpha| = |X_i|$ **do**
      if $\texttt{test}_{(P',E')}(X_i, \alpha)$ **then**
        $E' := E' \cup \{(X_i = \alpha)\}$;
        break the inner for loop;
      end if
    end for
    if the inner for loop not broken **then**
      $P' := P' \cup \{X_i\}$;
    end if
  end for
  $P := P'$; $E := E'$;
until P does not change

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Algorithm 1 Naive algorithm

initialise $B = (P, E)$ as the base of $\equiv_0$;
$P' := P$;
repeat
  $E' := \emptyset$;
  for all $X_i \in (\{X_2, \ldots, X_n\} \setminus P)$ **do**
    for all α such that $|\alpha| = |X_i|$ **do**
      if $\texttt{test}_{(P',E')}(X_i, \alpha)$ **then**
        $E' := E' \cup \{(X_i = \alpha)\}$;
        break the inner for loop;
      end if
    end for
    if the inner for loop not broken **then**
      $P' := P' \cup \{X_i\}$;
    end if
  end for
  $P := P'$; $E := E'$;
until P does not change

- small number of candidates α

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Algorithm 1 Naive algorithm

initialise $B = (P, E)$ as the base of $\equiv_0$;
$P' := P$;
repeat
    $E' := \emptyset$;
    for all $X_i \in (\{X_2, \ldots, X_n\} \setminus P)$ **do**
        for all α such that $|\alpha| = |X_i|$ **do**
            if $\texttt{test}_{(P',E')}(X_i, \alpha)$ **then**
                $E' := E' \cup \{(X_i = \alpha)\}$;
                break the inner for loop;
            end if
        end for
        if the inner for loop not broken **then**
            $P' := P' \cup \{X_i\}$;
        end if
    end for
    $P := P'$; $E := E'$;
until P does not change

- small number of candidates α
- compressed representation of α's

Outline of the algorithm:

1. Compute the initial base B
2. Compute the refinement ref(B)
3. If ref(B) = B then halt and return B
4. Assign B := ref(B) and go to step 2.

Until now we dealt with both BPA and BPP

Until now we dealt with both BPA and BPP

From now on we restrict ourselves to BPA

Left-most prime factor (lpf)

Left-most prime factor (lpf)

Def: $X_j \in P$ is left-most prime factor of X_i wrt. B

iff

$$X_i \;\equiv_B\; X_j\,\alpha, \quad \text{for some } \alpha \in (P \;\cap\; \{X_1 \ldots X_{i-1}\})^*$$

Left-most prime factor (lpf)

Def: $X_j \in P$ is left-most prime factor of X_i wrt. B

iff $\qquad X_j = \mathrm{lpf}_B(X_i)$

$$X_i \;\equiv_B\; X_j\,\alpha, \quad \text{for some } \alpha \in (P \;\cap\; \{X_1 \ldots X_{i-1}\})^*$$

Left-most prime factor (lpf)

Def: $X_j \in P$ is left-most prime factor of X_i wrt. B

iff $\quad X_j = \mathrm{lpf}_B(X_i)$

$$X_i \equiv_B X_j\,\alpha, \quad \text{for some } \alpha \in (P \cap \{X_1 \ldots X_{i-1}\})^*$$

Fix norm-reducing transitions $X_i \xrightarrow{a} \alpha_i \qquad \alpha_i \in \{X_1 \ldots X_{i-1}\}^*$

Left-most prime factor (lpf)

Def: $X_j \in P$ is left-most prime factor of X_i wrt. B

iff $\qquad X_j = \mathrm{lpf}_B(X_i)$

$$X_i \equiv_B X_j\,\alpha, \quad \text{for some } \alpha \in (P \cap \{X_1 \ldots X_{i-1}\})^*$$

Fix norm-reducing transitions $X_i \xrightarrow{a} \alpha_i \qquad \alpha_i \in \{X_1 \ldots X_{i-1}\}^*$

Fact: If $\equiv_B$ is norm-reducing bisimulation and $X_j = \mathrm{lpf}_B(X_i)$ then

$$X_i \equiv_B X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_B(\alpha_i))$$

Left-most prime factor (lpf)

Def: $X_j \in P$ is left-most prime factor of X_i wrt. B

iff $\quad X_j = \mathrm{lpf}_B(X_i)$

$$X_i \equiv_B X_j\,\alpha, \quad \text{for some } \alpha \in (P \cap \{X_1 \dots X_{i-1}\})^*$$

Fix norm-reducing transitions $X_i \xrightarrow{a} \alpha_i \qquad \alpha_i \in \{X_1 \dots X_{i-1}\}^*$

Fact: If $\equiv_B$ is norm-reducing bisimulation and $X_j = \mathrm{lpf}_B(X_i)$ then

$$X_i \equiv_B X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_B(\alpha_i))$$

suffix wrt. norm

Left-most prime factor (lpf)

Def: $X_j \in P$ is left-most prime factor of X_i wrt. B

iff $\qquad X_j = \mathrm{lpf}_B(X_i)$

$$X_i \equiv_B X_j\,\alpha, \quad \text{for some } \alpha \in (P \cap \{X_1 \ldots X_{i-1}\})^*$$

Fix norm-reducing transitions $X_i \xrightarrow{a} \alpha_i \qquad \alpha_i \in \{X_1 \ldots X_{i-1}\}^*$

Fact: If $\equiv_B$ is norm-reducing bisimulation and $X_j = \mathrm{lpf}_B(X_i)$ then

$$X_i \equiv_B X_j\, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_B(\alpha_i))$$

suffix wrt. norm may be undefined

Left-most prime factor (lpf)

Def: $X_j \in P$ is left-most prime factor of X_i wrt. B

iff

$X_j = \mathrm{lpf}_B(X_i)$

$$X_i \equiv_B X_j \boxed{\alpha,} \text{ for some } \alpha \in (P \cap \{X_1 \ldots X_{i-1}\})^*$$

Fix norm-reducing transitions $X_i \xrightarrow{a} \alpha_i$ $\alpha_i \in \{X_1 \ldots X_{i-1}\}^*$

Fact: If $\equiv_B$ is norm-reducing bisimulation and $X_j = \mathrm{lpf}_B(X_i)$ then

$$X_i \equiv_B X_j \boxed{\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_B(\alpha_i))}$$

suffix wrt. norm
may be undefined

Left-most prime factor (lpf)

Def: $X_j \in P$ is left-most prime factor of X_i wrt. B

iff $\qquad X_j = \mathrm{lpf}_B(X_i)$

$$X_i \;\equiv_B\; X_j \boxed{\alpha,} \;\text{ for some } \alpha \in (P \;\cap\; \{X_1 \ldots X_{i-1}\})^*$$

Fix norm-reducing transitions $\quad X_i \xrightarrow{a} \alpha_i \qquad \alpha_i \in \{X_1 \ldots X_{i-1}\}^*$

Fact: If $\equiv_B$ is norm-reducing bisimulation and $X_j = \mathrm{lpf}_B(X_i)$ then

$$X_i \;\equiv_B\; X_j \boxed{\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_B(\alpha_i))}$$

suffix wrt. norm
may be undefined

Proof:

Left-most prime factor (lpf)

Def: $X_j \in P$ is left-most prime factor of X_i wrt. B

iff $\qquad X_j = \mathrm{lpf}_B(X_i)$

$$X_i \equiv_B X_j \boxed{\alpha,} \text{ for some } \alpha \in (P \cap \{X_1 \ldots X_{i-1}\})^*$$

Fix norm-reducing transitions $X_i \xrightarrow{a} \alpha_i \qquad \alpha_i \in \{X_1 \ldots X_{i-1}\}^*$

Fact: If $\equiv_B$ is norm-reducing bisimulation and $X_j = \mathrm{lpf}_B(X_i)$ then

$$X_i \equiv_B X_j \boxed{\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_B(\alpha_i))}$$

suffix wrt. norm
may be undefined

Proof: Bisimulation Game for $X_i \equiv_B X_j \, \alpha$

Left-most prime factor (lpf)

Def: $X_j \in P$ is left-most prime factor of X_i wrt. B

iff

$X_j = \mathrm{lpf}_B(X_i)$

$X_i \equiv_B X_j \boxed{\alpha,}$ for some $\alpha \in (P \cap \{X_1 \ldots X_{i-1}\})^*$

Fix norm-reducing transitions $X_i \xrightarrow{a} \alpha_i$ $\quad \alpha_i \in \{X_1 \ldots X_{i-1}\}^*$

Fact: If $\equiv_B$ is norm-reducing bisimulation and $X_j = \mathrm{lpf}_B(X_i)$ then

$X_i \equiv_B X_j \boxed{\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_B(\alpha_i))}$

suffix wrt. norm
may be undefined

Proof: Bisimulation Game for $X_i \equiv_B X_j \alpha$

Spoiler's move: $X_i \xrightarrow{a} \alpha_i$

Left-most prime factor (lpf)

Def: $X_j \in P$ is left-most prime factor of X_i wrt. B

iff $\qquad X_j = \mathrm{lpf}_B(X_i)$

$$X_i \ \equiv_B \ X_j \boxed{\alpha,} \ \text{ for some } \alpha \in (P \ \cap \ \{X_1 \ldots X_{i-1}\})^*$$

Fix norm-reducing transitions $X_i \xrightarrow{a} \alpha_i \qquad \alpha_i \in \{X_1 \ldots X_{i-1}\}^*$

Fact: If $\equiv_B$ is norm-reducing bisimulation and $X_j = \mathrm{lpf}_B(X_i)$ then

$$X_i \ \equiv_B \ X_j \boxed{\mathrm{suffix}_{|X_i| - |X_j|}(\mathrm{dec}_B(\alpha_i))}$$

suffix wrt. norm
may be undefined

Proof: Bisimulation Game for $X_i \ \equiv_B \ X_j \, \alpha$

Spoiler's move: $X_i \xrightarrow{a} \alpha_i$

Duplicator's response: $X_j \xrightarrow{a} \gamma$

Left-most prime factor (lpf)

Def: $X_j \in P$ is left-most prime factor of X_i wrt. B

iff

$X_j = \mathrm{lpf}_B(X_i)$

$X_i \ \equiv_B \ X_j \boxed{\alpha,}$ for some $\alpha \in (P \ \cap \ \{X_1 \ldots X_{i-1}\})^*$

Fix norm-reducing transitions $X_i \ \xrightarrow{a} \ \alpha_i$ $\qquad \alpha_i \in \{X_1 \ldots X_{i-1}\}^*$

Fact: If $\equiv_B$ is norm-reducing bisimulation and $X_j = \mathrm{lpf}_B(X_i)$ then

$X_i \ \equiv_B \ X_j \boxed{\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_B(\alpha_i))}$

suffix wrt. norm
may be undefined

Proof: Bisimulation Game for $X_i \ \equiv_B \ X_j \, \alpha$

Spoiler's move: $X_i \ \xrightarrow{a} \ \alpha_i$

Duplicator's response: $X_j \ \xrightarrow{a} \ \gamma$ $\qquad \alpha_i \ \equiv_B \ \gamma \, \alpha$

Left-most prime factor (lpf)

Def: $X_j \in P$ is left-most prime factor of X_i wrt. B

iff $\qquad X_j = \mathrm{lpf}_B(X_i)$

$$X_i \equiv_B X_j \boxed{\alpha,} \text{ for some } \alpha \in (P \cap \{X_1 \dots X_{i-1}\})^*$$

Fix norm-reducing transitions $X_i \xrightarrow{a} \alpha_i \qquad \alpha_i \in \{X_1 \dots X_{i-1}\}^*$

Fact: If $\equiv_B$ is norm-reducing bisimulation and $X_j = \mathrm{lpf}_B(X_i)$ then

$$X_i \equiv_B X_j \boxed{\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_B(\alpha_i))}$$

suffix wrt. norm
may be undefined

Proof: Bisimulation Game for $X_i \equiv_B X_j \alpha$

Spoiler's move: $X_i \xrightarrow{a} \alpha_i$

Duplicator's response: $X_j \xrightarrow{a} \gamma \qquad \alpha_i \equiv_B \gamma \alpha$

$$\mathrm{dec}_B(\alpha_i) = \mathrm{dec}_B(\gamma)\,\mathrm{dec}_B(\alpha)$$

Algorithm 1 Naive algorithm

initialise $B = (P, E)$ as the base of $\equiv_0$;
$P' := P$;
repeat
    $E' := \emptyset$;
    for all $X_i \in (\{X_2, \ldots, X_n\} \setminus P)$ **do**
        for all α such that $|\alpha| = |X_i|$ **do**
            if $\mathtt{test}_{(P', E')}(X_i, \alpha)$ **then**
                $E' := E' \cup \{(X_i = \alpha)\}$;
                break the inner for loop;
            end if
        end for
        if the inner for loop not broken **then**
            $P' := P' \cup \{X_i\}$;
        end if
    end for
    $P := P'$; $E := E'$;
until P does not change

- small number of candidates α

Small number of candidates

Small number of candidates

When computing *B'*, restrict to candidate prime decompositions of the following form:

Small number of candidates

When computing *B'*, restrict to candidate prime decompositions of the following form:

$$X_i \; = \; X_j \, \alpha \qquad \text{where} \;\; \alpha \; = \; \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

Small number of candidates

When computing *B'*, restrict to candidate prime decompositions of the following form:

$$X_i \;=\; X_j\,\alpha \qquad \text{where} \;\; \alpha \;=\; \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\alpha_i \in \{X_1 \ldots X_{i-1}\}^*$$

Small number of candidates

When computing *B'*, restrict to candidate prime decompositions of the following form:

$$X_i \;=\; X_j\,\alpha \qquad \text{where} \;\; \alpha \;=\; \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$\alpha_i \in \{X_1 \ldots X_{i-1}\}^*$
$\mathrm{dec}_{B'}(\alpha_i)$ defined

Small number of candidates

When computing *B'*, restrict to candidate prime decompositions of the following form:

$$\begin{aligned} X_i &= X_j\,\alpha \qquad \text{where } \alpha = \text{suffix}_{|X_i|-|X_j|}(\text{dec}_{B'}(\alpha_i)) \\ \text{dec}_{B'}(X_i) &= X_j\,\alpha \end{aligned}$$

$\alpha_i \in \{X_1 \ldots X_{i-1}\}^*$
$\text{dec}_{B'}(\alpha_i)$ defined

Small number of candidates

When computing *B'*, restrict to candidate prime decompositions of the following form:

$$X_i = X_j\,\alpha \qquad \text{where} \quad \alpha = \text{suffix}_{|X_i|-|X_j|}(\text{dec}_{B'}(\alpha_i))$$
$$\text{dec}_{B'}(X_i) = X_j\,\alpha$$

$\alpha_i \in \{X_1 \ldots X_{i-1}\}^*$
$\text{dec}_{B'}(\alpha_i)$ defined

In other words, we check whether

$$X_j = \text{lpf}_{B'}(X_i)$$

Small number of candidates

When computing B', restrict to candidate prime decompositions of the following form:

$$\begin{aligned} X_i &= X_j\,\alpha \qquad \text{where } \alpha = \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i)) \\ \mathrm{dec}_{B'}(X_i) &= X_j\,\alpha \end{aligned}$$

$\alpha_i \in \{X_1 \ldots X_{i-1}\}^*$
$\mathrm{dec}_{B'}(\alpha_i)$ defined

In other words, we check whether

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$\texttt{lpftest}_{B'}(X_i, X_j)$ subroutine on the next slide

Small number of candidates

When computing *B'*, restrict to candidate prime decompositions of the following form:

$$X_i = X_j\,\alpha \qquad \text{where} \quad \alpha = \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$
$$\mathrm{dec}_{B'}(X_i) = X_j\,\alpha$$

$\alpha_i \in \{X_1 \ldots X_{i-1}\}^*$
$\mathrm{dec}_{B'}(\alpha_i)$ defined

In other words, we check whether

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$\texttt{lpftest}_{B'}(X_i, X_j)$ subroutine on the next slide

Fact: $\mathrm{lpf}_B(X_i) \leq \mathrm{lpf}_{B'}(X_i) < X_i$

Small number of candidates

When computing *B'*, restrict to candidate prime decompositions of the following form:

$$X_i = X_j\,\alpha \qquad \text{where} \quad \alpha = \text{suffix}_{|X_i|-|X_j|}(\text{dec}_{B'}(\alpha_i))$$
$$\text{dec}_{B'}(X_i) = X_j\,\alpha$$

$\alpha_i \in \{X_1 \ldots X_{i-1}\}^*$
$\text{dec}_{B'}(\alpha_i)$ defined

In other words, we check whether

$$X_j = \text{lpf}_{B'}(X_i)$$

$\texttt{lpftest}_{B'}(X_i, X_j)$ subroutine on the next slide

Fact: $\text{lpf}_B(X_i) \leq \text{lpf}_{B'}(X_i) < X_i$

Proof:

Small number of candidates

When computing *B'*, restrict to candidate prime decompositions of the following form:

$$\begin{aligned} X_i &= X_j\,\alpha \qquad \text{where} \quad \alpha = \text{suffix}_{|X_i|-|X_j|}(\text{dec}_{B'}(\alpha_i)) \\ \text{dec}_{B'}(X_i) &= X_j\,\alpha \end{aligned}$$

$\alpha_i \in \{X_1 \ldots X_{i-1}\}^*$
$\text{dec}_{B'}(\alpha_i)$ defined

In other words, we check whether

$$X_j = \text{lpf}_{B'}(X_i)$$

$\texttt{lpftest}_{B'}(X_i, X_j)$ subroutine on the next slide

Fact: $\text{lpf}_B(X_i) \leq \text{lpf}_{B'}(X_i) < X_i$

Proof: Again, by unique decomposition property of $\equiv_B$

Small number of candidates

When computing *B'*, restrict to candidate prime decompositions of the following form:

$$\begin{aligned} X_i &= X_j\,\alpha \qquad \text{where} \quad \alpha = \text{suffix}_{|X_i|-|X_j|}(\text{dec}_{B'}(\alpha_i)) \\ \text{dec}_{B'}(X_i) &= X_j\,\alpha \end{aligned}$$

$\alpha_i \in \{X_1 \ldots X_{i-1}\}^*$
$\text{dec}_{B'}(\alpha_i)$ defined

In other words, we check whether

$$X_j = \text{lpf}_{B'}(X_i)$$

$\texttt{lpftest}_{B'}(X_i, X_j)$ subroutine on the next slide

Fact: $\text{lpf}_B(X_i) \leq \text{lpf}_{B'}(X_i) < X_i$

Proof: Again, by unique decomposition property of $\equiv_B$

$\text{lpf}_B(\text{lpf}_{B'}(X_i)) = \text{lpf}_B(X_i)$

Algorithm 2 Efficient algorithm for CFG

initialise $B = (P, E)$ as the base of $\equiv_0$;
$P' := P$;
repeat
  $E' := \emptyset$;
  for all $X_i \in (\{X_2, \ldots, X_n\} \setminus P)$ **do**
    if $\neg \texttt{lpftest}_{(P',E')}(X_i, \texttt{lpf}_{(P,E)}(X_i))$ **then**
      for all $X_j \in \{X_{\texttt{lpfindex}_{(P,E)}+1}, \ldots, X_{i-1}\} \cap (P' \setminus P)$ **do**
        if $\texttt{lpftest}_{(P',E')}(X_i, X_j)$ **then**
          $s := \texttt{dec}_{(P',E')}(\alpha_i)$;
          $E' := E' \cup \{(X_i, X_j\ \texttt{suffix}_{|X_i|-|X_j|}(s))\}$;
          break the inner for loop;
        end if
      end for
      if the inner for loop not broken **then**
        $P' := P' \cup \{X_i\}$;
      end if
    else
      $X := \texttt{lpf}_{(P,E)}(X_i)$;
      $s := \texttt{dec}_{(P',E')}(\alpha_i)$;
      $E' := E' \cup \{(X_i, X\ \texttt{suffix}_{|X_i|-|X|}(s))\}$;
    end if
  end for
  $P := P'$; $E := E'$;
until P does not change

$\mathrm{lpf}_B(X_i) \leq \mathrm{lpf}_{B'}(X_i) < X_i$

Algorithm 2 Efficient algorithm for CFG

initialise $B = (P, E)$ as the base of $\equiv_0$;
$P' := P$;
repeat
  $E' := \emptyset$;
  for all $X_i \in (\{X_2, \ldots, X_n\} \setminus P)$ **do**
    if $\neg \texttt{lpftest}_{(P',E')}(X_i, \texttt{lpf}_{(P,E)}(X_i))$ **then**
      for all $X_j \in \{X_{\texttt{lpfindex}_{(P,E)}+1}, \ldots, X_{i-1}\} \cap (P' \setminus P)$ **do**
        if $\texttt{lpftest}_{(P',E')}(X_i, X_j)$ **then**
          $s := \texttt{dec}_{(P',E')}(\alpha_i)$;
          $E' := E' \cup \{(X_i, X_j\ \texttt{suffix}_{|X_i|-|X_j|}(s))\}$;
          break the inner for loop;
        end if
      end for
      if the inner for loop not broken **then**
        $P' := P' \cup \{X_i\}$;
      end if
    else
      $X := \texttt{lpf}_{(P,E)}(X_i)$;
      $s := \texttt{dec}_{(P',E')}(\alpha_i)$;
      $E' := E' \cup \{(X_i, X\ \texttt{suffix}_{|X_i|-|X|}(s))\}$;
    end if
  end for
  $P := P'$; $E := E'$;
until P does not change

$\mathrm{lpf}_B(X_i) \leq \mathrm{lpf}_{B'}(X_i) < X_i$

Algorithm 2 Efficient algorithm for CFG

initialise $B = (P, E)$ as the base of $\equiv_0$;

$P' := P$;

repeat

  $E' := \emptyset$;

  for all $X_i \in (\{X_2, \dots, X_n\} \setminus P)$ **do**

    if $\neg\texttt{lpftest}_{(P',E')}(X_i, \texttt{lpf}_{(P,E)}(X_i))$ **then**

      for all $X_j \in \{X_{\texttt{lpfindex}_{(P,E)}+1}, \dots, X_{i-1}\} \cap (P' \setminus P)$ **do**

        if $\texttt{lpftest}_{(P',E')}(X_i, X_j)$ **then**

          $s := \texttt{dec}_{(P',E')}(\alpha_i)$;

          $E' := E' \cup \{(X_i, X_j\ \texttt{suffix}_{|X_i|-|X_j|}(s))\}$;

          break the inner for loop;

        end if

      end for

      if the inner for loop not broken **then**

        $P' := P' \cup \{X_i\}$;

      end if

    else

      $X := \texttt{lpf}_{(P,E)}(X_i)$;

      $s := \texttt{dec}_{(P',E')}(\alpha_i)$;

      $E' := E' \cup \{(X_i, X\ \texttt{suffix}_{|X_i|-|X|}(s))\}$;

    end if

  end for

  $P := P'$; $E := E'$;

until P does not change

Algorithm 2 Efficient algorithm for CFG

initialise $B = (P, E)$ as the base of $\equiv_0$;
$P' := P$;
repeat
    $E' := \emptyset$;
    for all $X_i \in (\{X_2, \ldots, X_n\} \setminus P)$ **do**
        if $\neg \texttt{lpftest}_{(P',E')}(X_i, \texttt{lpf}_{(P,E)}(X_i))$ **then**
            for all $X_j \in \{X_{\texttt{lpfindex}_{(P,E)}+1}, \ldots, X_{i-1}\} \cap (P' \setminus P)$ **do**
                if $\texttt{lpftest}_{(P',E')}(X_i, X_j)$ **then**
                    $s := \texttt{dec}_{(P',E')}(\alpha_i)$;
                    $E' := E' \cup \{(X_i, X_j\ \texttt{suffix}_{|X_i|-|X_j|}(s))\}$;
                    break the inner for loop;
                end if
            end for
            if the inner for loop not broken **then**
                $P' := P' \cup \{X_i\}$;
            end if
        else
            $X := \texttt{lpf}_{(P,E)}(X_i)$;
            $s := \texttt{dec}_{(P',E')}(\alpha_i)$;
            $E' := E' \cup \{(X_i, X\ \texttt{suffix}_{|X_i|-|X|}(s))\}$;
        end if
    end for
    $P := P'$; $E := E'$;
until P does not change

?

before explaining lpftest,
let's look at an example

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

Fix $\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

Fix $\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$

$$X_i \xrightarrow{a_i} \alpha_i$$

Example:

$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$

$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

Fix $\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$

$X_i \xrightarrow{a_i} \alpha_i$

Initial base:

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

Fix $\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$

$$X_i \xrightarrow{a_i} \alpha_i$$

Initial base:

$$P = \{ X_1 \}$$

$$\begin{aligned} E = \{ X_2 &= X_1 X_1 \\ X_3 &= X_1 X_1 X_1 \\ X_4 &= X_1 X_1 X_1 X_1 X_1 X_1 \\ X_5 &= X_1 X_1 X_1 X_1 X_1 X_1 X_1 \} \end{aligned}$$

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

first iteration

Example: first iteration

$X_1 \xrightarrow{a} \varepsilon \quad X_2 \xrightarrow{a} X_1 \quad X_2 \xrightarrow{b} X_1X_1 \quad X_3 \xrightarrow{a} X_2$

$X_4 \xrightarrow{a} X_3X_2 \quad X_4 \xrightarrow{b} X_4 \quad X_5 \xrightarrow{a} X_4 \quad X_5 \xrightarrow{a} X_3X_3$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

$\alpha_1 = \varepsilon \quad \alpha_2 = X_1 \quad \alpha_3 = X_2 \quad \alpha_4 = X_3X_2 \quad \alpha_5 = X_3X_3$

$$P = \{\, X_1 \,\}$$

$$\begin{aligned} E = \{\, X_2 &= X_1\, X_1 \\ X_3 &= X_1\, X_1\, X_1 \\ X_4 &= X_1\, X_1\, X_1\, X_1\, X_1\, X_1 \\ X_5 &= X_1\, X_1\, X_1\, X_1\, X_1\, X_1\, X_1 \} \end{aligned}$$

Example:

first iteration

$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$

$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$

$$\begin{aligned} P &= \{\, X_1 \,\} \\ E &= \{\, X_2 = X_1\,X_1 \\ &\quad\; X_3 = X_1\,X_1\,X_1 \\ &\quad\; X_4 = X_1\,X_1\,X_1\,X_1\,X_1\,X_1 \\ &\quad\; X_5 = X_1\,X_1\,X_1\,X_1\,X_1\,X_1\,X_1 \} \end{aligned}$$

$$\begin{aligned} P' &= \{\, X_1, X_2, X_5 \,\} \\ E' &= \{\, X_3 = X_1\,X_2 \\ &\quad\; X_4 = X_2\,X_2\,X_2 \,\} \end{aligned}$$

Example:

first iteration

$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$

$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$

$$P = \{ X_1 \}$$

$$\begin{aligned} E = \{ X_2 &= X_1\,X_1 \\ X_3 &= X_1\,X_1\,X_1 \\ X_4 &= X_1\,X_1\,X_1\,X_1\,X_1\,X_1 \\ X_5 &= X_1\,X_1\,X_1\,X_1\,X_1\,X_1\,X_1 \} \end{aligned}$$

$$X_2 \equiv_{B'} X_1\,\mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1\,X_1 \;?$$

$$P' = \{ X_1, X_2, X_5 \}$$

$$\begin{aligned} E' = \{ X_3 &= X_1\,X_2 \\ X_4 &= X_2\,X_2\,X_2 \} \end{aligned}$$

Example:

first iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1 \}$$

$$E = \{ X_2 = X_1 X_1, \; X_3 = X_1 X_1 X_1, \; X_4 = X_1 X_1 X_1 X_1 X_1 X_1, \; X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$$

$$X_2 \equiv_{B'} X_1 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1 X_1 \; ?$$

$X_i \equiv_{B'} \alpha$

$(a) \quad (X_i, \alpha) \in \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_5 \}$$

$$E' = \{ X_3 = X_1 X_2, \; X_4 = X_2 X_2 X_2 \}$$

Example:

first iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1 \}$$

$$E = \{ X_2 = X_1 X_1, \; X_3 = X_1 X_1 X_1, \; X_4 = X_1 X_1 X_1 X_1 X_1 X_1, \; X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$$

$$X_2 \equiv_{B'} X_1 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1 X_1 \ ?$$ **no!** (b)

$X_i \equiv_{B'} \alpha$

$(a) \quad (X_i, \alpha) \in \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr-exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_5 \}$$

$$E' = \{ X_3 = X_1 X_2, \; X_4 = X_2 X_2 X_2 \}$$

Example:

first iteration

$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$

$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$

$P = \{ X_1 \}$

$E = \{ X_2 = X_1\,X_1$
$X_3 = X_1\,X_1\,X_1$
$X_4 = X_1\,X_1\,X_1\,X_1\,X_1\,X_1$
$X_5 = X_1\,X_1\,X_1\,X_1\,X_1\,X_1\,X_1 \}$

$X_2 \equiv_{B'} X_1\,\mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1\,X_1$? **no!** (b)

$X_3 \equiv_{B'} X_1\,\mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1\,X_2$?

$X_i \equiv_{B'} \alpha$
$(a) \quad (X_i, \alpha) \in \equiv_B$
$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$
$(c) \quad (X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$P' = \{ X_1, X_2, X_5 \}$

$E' = \{ X_3 = X_1\,X_2$
$X_4 = X_2\,X_2\,X_2 \}$

Example: first iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1 \}$$
$$E = \{ X_2 = X_1 X_1, \; X_3 = X_1 X_1 X_1, \; X_4 = X_1 X_1 X_1 X_1 X_1 X_1, \; X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$$

$$X_2 \equiv_{B'} X_1 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1 \, X_1 \; ?$$ no! (b)

$$X_3 \equiv_{B'} X_1 \, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 \, X_2 \; ?$$ yes!

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr\!-\!exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_5 \}$$
$$E' = \{ X_3 = X_1 \, X_2, \; X_4 = X_2 \, X_2 \, X_2 \}$$

Example: first iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1 \}$$

$$E = \{ X_2 = X_1 X_1, \quad X_3 = X_1 X_1 X_1, \quad X_4 = X_1 X_1 X_1 X_1 X_1 X_1, \quad X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$$

$X_2 \equiv_{B'} X_1 \,\mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1 X_1$? no! (b)

$X_3 \equiv_{B'} X_1 \,\mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 X_2$? yes!

$X_3 \equiv_{B'} X_2 \,\mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$?

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr\!-\!exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_5 \}$$

$$E' = \{ X_3 = X_1 X_2, \quad X_4 = X_2 X_2 X_2 \}$$

Example:

first iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1 \}$$

$$E = \{ X_2 = X_1 X_1, \quad X_3 = X_1 X_1 X_1, \quad X_4 = X_1 X_1 X_1 X_1 X_1 X_1, \quad X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$$

$$X_2 \equiv_{B'} X_1 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1 \, X_1 \; ?$$ no! (b)

$$X_3 \equiv_{B'} X_1 \, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 \, X_2 \; ?$$ yes!

$$X_3 \equiv_{B'} X_2 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3)) \; ?$$ undefined!

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_5 \}$$

$$E' = \{ X_3 = X_1 X_2, \quad X_4 = X_2 X_2 X_2 \}$$

Example: first iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1 \}$$
$$E = \{ X_2 = X_1 X_1, \; X_3 = X_1 X_1 X_1, \; X_4 = X_1 X_1 X_1 X_1 X_1 X_1, \; X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$$

$X_2 \equiv_{B'} X_1 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1 \, X_1$? no! (b)

$X_3 \equiv_{B'} X_1 \, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 \, X_2$? yes!

$X_3 \equiv_{B'} X_2 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$? undefined!

$X_4 \equiv_{B'} X_1 \, \mathrm{suffix}_5(\mathrm{dec}_{B'}(\alpha_4)) = X_1 \, X_1 \, X_2 \, X_2$?

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_5 \}$$
$$E' = \{ X_3 = X_1 \, X_2, \; X_4 = X_2 \, X_2 \, X_2 \}$$

Example: **first iteration**

$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$

$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$

$P = \{ X_1 \}$

$E = \{ X_2 = X_1 X_1$
$X_3 = X_1 X_1 X_1$
$X_4 = X_1 X_1 X_1 X_1 X_1 X_1$
$X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$

$X_2 \equiv_{B'} X_1 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1 X_1$? **no!** (b)

$X_3 \equiv_{B'} X_1 \, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 X_2$? **yes!**

$X_3 \equiv_{B'} X_2 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$? **undefined!**

$X_4 \equiv_{B'} X_1 \, \mathrm{suffix}_5(\mathrm{dec}_{B'}(\alpha_4)) = X_1 X_1 X_2 X_2$? **no!** (b)

$X_i \equiv_{B'} \alpha$

$(a) \quad (X_i, \alpha) \in \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$P' = \{ X_1, X_2, X_5 \}$

$E' = \{ X_3 = X_1 X_2$
$X_4 = X_2 X_2 X_2 \}$

Example:

first iteration

$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$

$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$

$P = \{ X_1 \}$

$E = \{ X_2 = X_1 X_1$
$X_3 = X_1 X_1 X_1$
$X_4 = X_1 X_1 X_1 X_1 X_1 X_1$
$X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$

$X_2 \equiv_{B'} X_1 \,\mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1 X_1$? **no!** (b)

$X_3 \equiv_{B'} X_1 \,\mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 X_2$? **yes!**

$X_3 \equiv_{B'} X_2 \,\mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$? **undefined!**

$X_4 \equiv_{B'} X_1 \,\mathrm{suffix}_5(\mathrm{dec}_{B'}(\alpha_4)) = X_1 X_1 X_2 X_2$? **no!** (b)

$X_4 \equiv_{B'} X_2 \,\mathrm{suffix}_4(\mathrm{dec}_{B'}(\alpha_4)) = X_2 X_2 X_2$?

$X_i \equiv_{B'} \alpha$
$(a) \quad (X_i, \alpha) \in \equiv_B$
$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$
$(c) \quad (X_i, \alpha) \in \mathtt{nr\!-\!exp}(\equiv_{B'})$

$P' = \{ X_1, X_2, X_5 \}$

$E' = \{ X_3 = X_1 X_2$
$X_4 = X_2 X_2 X_2 \}$

Example: first iteration

$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$

$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$

$P = \{ X_1 \}$

$E = \{ X_2 = X_1 X_1$
$X_3 = X_1 X_1 X_1$
$X_4 = X_1 X_1 X_1 X_1 X_1 X_1$
$X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$

$X_2 \equiv_{B'} X_1\, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1 X_1$? no! (b)

$X_3 \equiv_{B'} X_1\, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 X_2$? yes!

$X_3 \equiv_{B'} X_2\, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$? undefined!

$X_4 \equiv_{B'} X_1\, \mathrm{suffix}_5(\mathrm{dec}_{B'}(\alpha_4)) = X_1 X_1 X_2 X_2$? no! (b)

$X_4 \equiv_{B'} X_2\, \mathrm{suffix}_4(\mathrm{dec}_{B'}(\alpha_4)) = X_2 X_2 X_2$? yes!

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$P' = \{ X_1, X_2, X_5 \}$

$E' = \{ X_3 = X_1 X_2$
$X_4 = X_2 X_2 X_2 \}$

Example:

first iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$P = \{ X_1 \}$

$E = \{ X_2 = X_1 X_1$
$X_3 = X_1 X_1 X_1$
$X_4 = X_1 X_1 X_1 X_1 X_1 X_1$
$X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$

$X_2 \equiv_{B'} X_1 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1 \, X_1$? no! (b)

$X_3 \equiv_{B'} X_1 \, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 \, X_2$? yes!

$X_3 \equiv_{B'} X_2 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$? undefined!

$X_4 \equiv_{B'} X_1 \, \mathrm{suffix}_5(\mathrm{dec}_{B'}(\alpha_4)) = X_1 \, X_1 \, X_2 \, X_2$? no! (b)

$X_4 \equiv_{B'} X_2 \, \mathrm{suffix}_4(\mathrm{dec}_{B'}(\alpha_4)) = X_2 \, X_2 \, X_2$? yes!

$X_5 \equiv_{B'} X_1 \, \mathrm{suffix}_6(\mathrm{dec}_{B'}(\alpha_5)) = X_1 \, X_1 \, X_2 \, X_1 \, X_2$?

$X_i \equiv_{B'} \alpha$
$(a) \quad (X_i, \alpha) \in \equiv_B$
$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$
$(c) \quad (X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$P' = \{ X_1, X_2, X_5 \}$

$E' = \{ X_3 = X_1 \, X_2$
$X_4 = X_2 \, X_2 \, X_2 \}$

Example: first iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$P = \{ X_1 \}$
$E = \{ X_2 = X_1 X_1$
$X_3 = X_1 X_1 X_1$
$X_4 = X_1 X_1 X_1 X_1 X_1 X_1$
$X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$

$X_2 \equiv_{B'} X_1 \,\mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1 X_1$? no! (b)

$X_3 \equiv_{B'} X_1 \,\mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 X_2$? yes!
$X_3 \equiv_{B'} X_2 \,\mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$? undefined!

$X_4 \equiv_{B'} X_1 \,\mathrm{suffix}_5(\mathrm{dec}_{B'}(\alpha_4)) = X_1 X_1 X_2 X_2$? no! (b)
$X_4 \equiv_{B'} X_2 \,\mathrm{suffix}_4(\mathrm{dec}_{B'}(\alpha_4)) = X_2 X_2 X_2$? yes!

$X_5 \equiv_{B'} X_1 \,\mathrm{suffix}_6(\mathrm{dec}_{B'}(\alpha_5)) = X_1 X_1 X_2 X_1 X_2$? no! (c)

$X_i \equiv_{B'} \alpha$
(a) $(X_i, \alpha) \in \equiv_B$
(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$
(c) $(X_i, \alpha) \in \mathtt{nr}-\mathtt{exp}(\equiv_{B'})$

$P' = \{ X_1, X_2, X_5 \}$
$E' = \{ X_3 = X_1 X_2$
$X_4 = X_2 X_2 X_2 \}$

Example: first iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1 \}$$

$$E = \{ X_2 = X_1 X_1, \; X_3 = X_1 X_1 X_1, \; X_4 = X_1 X_1 X_1 X_1 X_1 X_1, \; X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$$

$X_2 \equiv_{B'} X_1\,\mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1\,X_1$? **no!** (b)

$X_3 \equiv_{B'} X_1\,\mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1\,X_2$? **yes!**

$X_3 \equiv_{B'} X_2\,\mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$? **undefined!**

$X_4 \equiv_{B'} X_1\,\mathrm{suffix}_5(\mathrm{dec}_{B'}(\alpha_4)) = X_1\,X_1\,X_2\,X_2$? **no!** (b)

$X_4 \equiv_{B'} X_2\,\mathrm{suffix}_4(\mathrm{dec}_{B'}(\alpha_4)) = X_2\,X_2\,X_2$? **yes!**

$X_5 \xrightarrow{a} X_4$

$X_5 \equiv_{B'} X_1\,\mathrm{suffix}_6(\mathrm{dec}_{B'}(\alpha_5)) = X_1\,X_1\,X_2\,X_1\,X_2$? **no!** (c)

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr-exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_5 \}$$

$$E' = \{ X_3 = X_1\,X_2, \; X_4 = X_2\,X_2\,X_2 \}$$

Example: **first iteration**

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1 \}$$

$$E = \{ X_2 = X_1 X_1,\ X_3 = X_1 X_1 X_1,\ X_4 = X_1 X_1 X_1 X_1 X_1 X_1,\ X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$$

$X_2 \equiv_{B'} X_1\,\mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1\,X_1$? **no!** (b)

$X_3 \equiv_{B'} X_1\,\mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1\,X_2$? **yes!**

$X_3 \equiv_{B'} X_2\,\mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$? **undefined!**

$X_4 \equiv_{B'} X_1\,\mathrm{suffix}_5(\mathrm{dec}_{B'}(\alpha_4)) = X_1\,X_1\,X_2\,X_2$? **no!** (b)

$X_4 \equiv_{B'} X_2\,\mathrm{suffix}_4(\mathrm{dec}_{B'}(\alpha_4)) = X_2\,X_2\,X_2$? **yes!**

$$X_5 \xrightarrow{a} X_4$$

$$X_4 \not\equiv_{B'} X_1\,X_2\,X_1\,X_2$$

$X_5 \equiv_{B'} X_1\,\mathrm{suffix}_6(\mathrm{dec}_{B'}(\alpha_5)) = X_1\,X_1\,X_2\,X_1\,X_2$? **no!** (c)

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_5 \}$$

$$E' = \{ X_3 = X_1\,X_2,\ X_4 = X_2\,X_2\,X_2 \}$$

Example: **first iteration**

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1 \}$$
$$E = \{ X_2 = X_1 X_1, \; X_3 = X_1 X_1 X_1, \; X_4 = X_1 X_1 X_1 X_1 X_1 X_1, \; X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$$

$X_2 \equiv_{B'} X_1 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1 X_1$? **no!** (b)

$X_3 \equiv_{B'} X_1 \, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 X_2$? **yes!**

$X_3 \equiv_{B'} X_2 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$? **undefined!**

$X_4 \equiv_{B'} X_1 \, \mathrm{suffix}_5(\mathrm{dec}_{B'}(\alpha_4)) = X_1 X_1 X_2 X_2$? **no!** (b)

$X_4 \equiv_{B'} X_2 \, \mathrm{suffix}_4(\mathrm{dec}_{B'}(\alpha_4)) = X_2 X_2 X_2$? **yes!**

$$X_5 \xrightarrow{a} X_4$$
$$X_4 \not\equiv_{B'} X_1 X_2 X_1 X_2$$

$X_5 \equiv_{B'} X_1 \, \mathrm{suffix}_6(\mathrm{dec}_{B'}(\alpha_5)) = X_1 X_1 X_2 X_1 X_2$? **no!** (c)

$X_5 \equiv_{B'} X_2 \, \mathrm{suffix}_5(\mathrm{dec}_{B'}(\alpha_5)) = X_2 X_2 X_1 X_2$?

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_5 \}$$
$$E' = \{ X_3 = X_1 X_2, \; X_4 = X_2 X_2 X_2 \}$$

Example: first iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1 \}$$

$$E = \{ X_2 = X_1 X_1, \; X_3 = X_1 X_1 X_1, \; X_4 = X_1 X_1 X_1 X_1 X_1 X_1, \; X_5 = X_1 X_1 X_1 X_1 X_1 X_1 X_1 \}$$

$X_2 \equiv_{B'} X_1 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_2)) = X_1 X_1$? **no!** (b)

$X_3 \equiv_{B'} X_1 \, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 X_2$? **yes!**

$X_3 \equiv_{B'} X_2 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$? **undefined!**

$X_4 \equiv_{B'} X_1 \, \mathrm{suffix}_5(\mathrm{dec}_{B'}(\alpha_4)) = X_1 X_1 X_2 X_2$? **no!** (b)

$X_4 \equiv_{B'} X_2 \, \mathrm{suffix}_4(\mathrm{dec}_{B'}(\alpha_4)) = X_2 X_2 X_2$? **yes!**

$$X_5 \xrightarrow{a} X_4 \qquad X_4 \not\equiv_{B'} X_1 X_2 X_1 X_2$$

$X_5 \equiv_{B'} X_1 \, \mathrm{suffix}_6(\mathrm{dec}_{B'}(\alpha_5)) = X_1 X_1 X_2 X_1 X_2$? **no!** (c)

$X_5 \equiv_{B'} X_2 \, \mathrm{suffix}_5(\mathrm{dec}_{B'}(\alpha_5)) = X_2 X_2 X_1 X_2$? **no!** (b)

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_5 \}$$

$$E' = \{ X_3 = X_1 X_2, \; X_4 = X_2 X_2 X_2 \}$$

Example:

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

Example:

$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$

$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$

second iteration

Example: second iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1, X_2, X_5 \}$$

$$E = \{ X_3 = X_1 X_2$$

$$X_4 = X_2 X_2 X_2 \}$$

Example:

second iteration

$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$

$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$

$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$

$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$

$$P = \{ X_1, X_2, X_5 \}$$

$$E = \{ X_3 = X_1 X_2, \quad X_4 = X_2 X_2 X_2 \}$$

$$P' = \{ X_1, X_2, X_4, X_5 \}$$

$$E' = \{ X_3 = X_1 X_2 \}$$

Example: second iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1, X_2, X_5 \}$$
$$E = \{ X_3 = X_1 X_2 \quad X_4 = X_2 X_2 X_2 \}$$

$$X_3 \equiv_{B'} X_1 \, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 X_2 \ ?$$

$$P' = \{ X_1, X_2, X_4, X_5 \}$$
$$E' = \{ X_3 = X_1 X_2 \}$$

Example:

second iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{\, X_1, X_2, X_5 \,\}$$

$$E = \{\, X_3 = X_1\, X_2, \quad X_4 = X_2\, X_2\, X_2 \,\}$$

$$X_3 \equiv_{B'} X_1\, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1\, X_2 \ ?$$

$X_i \equiv_{B'} \alpha$

$(a) \quad (X_i, \alpha) \in \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$P' = \{\, X_1, X_2, X_4, X_5 \,\}$$

$$E' = \{\, X_3 = X_1\, X_2 \,\}$$

Example: second iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1, X_2, X_5 \}$$

$$E = \{ X_3 = X_1 X_2, \quad X_4 = X_2 X_2 X_2 \}$$

$$X_3 \equiv_{B'} X_1 \, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 X_2 \ ?$$ yes!

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_4, X_5 \}$$

$$E' = \{ X_3 = X_1 X_2 \}$$

Example: **second iteration**

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1, X_2, X_5 \}$$
$$E = \{ X_3 = X_1 X_2, \quad X_4 = X_2 X_2 X_2 \}$$

$$X_3 \equiv_{B'} X_1 \, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 X_2 \ ?$$ yes!

$$X_3 \equiv_{B'} X_2 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3)) \ ?$$

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_4, X_5 \}$$
$$E' = \{ X_3 = X_1 X_2 \}$$

Example: second iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1 X_1 \qquad X_3 \xrightarrow{a} X_2$$
$$X_4 \xrightarrow{a} X_3 X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3 X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3 X_2 \qquad \alpha_5 = X_3 X_3$$

$$P = \{ X_1, X_2, X_5 \}$$
$$E = \{ X_3 = X_1 X_2, \quad X_4 = X_2 X_2 X_2 \}$$

$X_3 \equiv_{B'} X_1 \, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 X_2$? yes!

$X_3 \equiv_{B'} X_2 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$? undefined!

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_4, X_5 \}$$
$$E' = \{ X_3 = X_1 X_2 \}$$

Example:

second iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1, X_2, X_5 \}$$
$$E = \{ X_3 = X_1 X_2 \quad X_4 = X_2 X_2 X_2 \}$$

$X_3 \equiv_{B'} X_1 \, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 X_2$? yes!

$X_3 \equiv_{B'} X_2 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$? undefined!

$X_4 \equiv_{B'} X_2 \, \mathrm{suffix}_4(\mathrm{dec}_{B'}(\alpha_4)) = X_2 X_2 X_2$?

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_4, X_5 \}$$
$$E' = \{ X_3 = X_1 X_2 \}$$

Example:

second iteration

$$X_1 \xrightarrow{a} \varepsilon \qquad X_2 \xrightarrow{a} X_1 \qquad X_2 \xrightarrow{b} X_1X_1 \qquad X_3 \xrightarrow{a} X_2$$

$$X_4 \xrightarrow{a} X_3X_2 \qquad X_4 \xrightarrow{b} X_4 \qquad X_5 \xrightarrow{a} X_4 \qquad X_5 \xrightarrow{a} X_3X_3$$

$$|X_1| = 1 \quad |X_2| = 2 \quad |X_3| = 3 \quad |X_4| = 6 \quad |X_5| = 7$$

$$\alpha_1 = \varepsilon \qquad \alpha_2 = X_1 \qquad \alpha_3 = X_2 \qquad \alpha_4 = X_3X_2 \qquad \alpha_5 = X_3X_3$$

$$P = \{ X_1, X_2, X_5 \}$$

$$E = \{ X_3 = X_1 X_2 \quad X_4 = X_2 X_2 X_2 \}$$

$X_3 \equiv_{B'} X_1 \, \mathrm{suffix}_2(\mathrm{dec}_{B'}(\alpha_3)) = X_1 X_2$? yes!

$X_3 \equiv_{B'} X_2 \, \mathrm{suffix}_1(\mathrm{dec}_{B'}(\alpha_3))$? undefined!

$X_4 \equiv_{B'} X_2 \, \mathrm{suffix}_4(\mathrm{dec}_{B'}(\alpha_4)) = X_2 X_2 X_2$? no! (b) and (c)

$$X_4 \xrightarrow{b} X_4 \equiv_B X_2 X_2 X_2$$

$$X_2 X_2 X_2 \xrightarrow{b} X_1 X_1 X_2 X_2$$

$X_i \equiv_{B'} \alpha$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$P' = \{ X_1, X_2, X_4, X_5 \}$$

$$E' = \{ X_3 = X_1 X_2 \}$$

Outline

- Background
  - Context-free graphs and commutative context-free graphs
  - Bisimulation equivalence problem
  - Norm
  - History of the problem
  - Unique decomposition
  - Naive algorithm
- Efficient algorithm for BPA and BPP
  - Outline of the algorithm
  - Refinement
  - Efficient computation of refinement for BPA
  - **Time-cost analysis**
  - Partially-commutative context-free graphs

Algorithm 2 Efficient algorithm for CFG

initialise $B = (P, E)$ as the base of $\equiv_0$;
$P' := P$;
repeat
$E' := \emptyset$;
for all $X_i \in (\{X_2, \ldots, X_n\} \setminus P)$ **do**
if $\neg \texttt{lpftest}_{(P',E')}(X_i, \texttt{lpf}_{(P,E)}(X_i))$ **then**
for all $X_j \in \{X_{\texttt{lpfindex}_{(P,E)}+1}, \ldots, X_{i-1}\} \cap (P' \setminus P)$ **do**
if $\texttt{lpftest}_{(P',E')}(X_i, X_j)$ **then**
$s := \texttt{dec}_{(P',E')}(\alpha_i)$;
$E' := E' \cup \{(X_i, X_j\ \texttt{suffix}_{|X_i|-|X_j|}(s))\}$;
break the inner for loop;
end if
end for
if the inner for loop not broken **then**
$P' := P' \cup \{X_i\}$;
end if
else
$X := \texttt{lpf}_{(P,E)}(X_i)$;
$s := \texttt{dec}_{(P',E')}(\alpha_i)$;
$E' := E' \cup \{(X_i, X\ \texttt{suffix}_{|X_i|-|X|}(s))\}$;
end if
end for
$P := P'$; $E := E'$;
until P does not change

$\mathcal{O}(n^2)$ invocations

Algorithm 2 Efficient algorithm for CFG

initialise $B = (P, E)$ as the base of $\equiv_0$;
$P' := P$;
repeat
  $E' := \emptyset$;
  for all $X_i \in (\{X_2, \dots, X_n\} \setminus P)$ **do**
    if $\neg \texttt{lpftest}_{(P',E')}(X_i, \texttt{lpf}_{(P,E)}(X_i))$ **then**
      for all $X_j \in \{X_{\texttt{lpfindex}_{(P,E)}+1}, \dots, X_{i-1}\} \cap (P' \setminus P)$ **do**
        if $\texttt{lpftest}_{(P',E')}(X_i, X_j)$ **then**
          $s := \texttt{dec}_{(P',E')}(\alpha_i)$;
          $E' := E' \cup \{(X_i, X_j\ \texttt{suffix}_{|X_i|-|X_j|}(s))\}$;
          break the inner for loop;
        end if
      end for
      if the inner for loop not broken **then**
        $P' := P' \cup \{X_i\}$;
      end if
    else
      $X := \texttt{lpf}_{(P,E)}(X_i)$;
      $s := \texttt{dec}_{(P',E')}(\alpha_i)$;
      $E' := E' \cup \{(X_i, X\ \texttt{suffix}_{|X_i|-|X|}(s))\}$;
    end if
  end for
  $P := P'$; $E := E'$;
until P does not change

$\mathcal{O}(n^2)$ invocations

lpftest

$$X_j \; = \; \mathrm{lpf}_{B'}(X_i)$$

$$X_i \;\; \equiv_{B'} \;\; \alpha$$

$$\alpha \;\; = \;\; X_j \, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$(a) \quad (X_i, \alpha) \in \, \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

- $\textsc{String}(a)$ for $a \in \Sigma$: creates a string "a"
- $\textsc{Concatenate}(s_1, s_2)$:
- $\textsc{Split}(s, i)$: splits into a prefix and suffix
- $\textsc{Equal}(s_1, s_2)$: equality test

$$X_i \equiv_{B'} \alpha$$
$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr-exp}(\equiv_{B'})$

lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

- $\textsc{String}(a)$ for $a \in \Sigma$: creates a string "a"
- $\textsc{Concatenate}(s_1, s_2)$:
- $\textsc{Split}(s, i)$: splits into a prefix and suffix
- $\textsc{Equal}(s_1, s_2)$: equality test

$$X_i \equiv_{B'} \alpha$$
$$\alpha = X_j \,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr-exp}(\equiv_{B'})$

Example:

$$\begin{aligned}
s_0 &:= \textsc{String}(a)\\
tmp &:= \textsc{String}(b)\\
s_1 &:= \textsc{Concatenate}(s_0, tmp)\\
s_2 &:= \textsc{Concatenate}(s_1, s_0)\\
s_3 &:= \textsc{Concatenate}(s_2, s_1)\\
&\ldots
\end{aligned}$$

lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

- $\textsc{String}(a)$ for $a \in \Sigma$: creates a string "a"
- $\textsc{Concatenate}(s_1, s_2)$:
- $\textsc{Split}(s, i)$: splits into a prefix and suffix
- $\textsc{Equal}(s_1, s_2)$: equality test

$$X_i \equiv_{B'} \alpha$$
$$\alpha = X_j \,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$(a) \quad (X_i, \alpha) \in \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

Example:

$$\begin{aligned} s_0 &:= \textsc{String}(a) \\ tmp &:= \textsc{String}(b) \\ s_1 &:= \textsc{Concatenate}(s_0, tmp) \\ s_2 &:= \textsc{Concatenate}(s_1, s_0) \\ s_3 &:= \textsc{Concatenate}(s_2, s_1) \\ &\dots \end{aligned}$$

ab

lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$
$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

- $\textsc{String}(a)$ for $a \in \Sigma$: creates a string "a"
- $\textsc{Concatenate}(s_1, s_2)$:
- $\textsc{Split}(s, i)$: splits into a prefix and suffix
- $\textsc{Equal}(s_1, s_2)$: equality test

Example:

$$\begin{aligned}
s_0 &:= \textsc{String}(a)\\
tmp &:= \textsc{String}(b)\\
s_1 &:= \textsc{Concatenate}(s_0, tmp) && \text{ab}\\
s_2 &:= \textsc{Concatenate}(s_1, s_0) && \text{aba}\\
s_3 &:= \textsc{Concatenate}(s_2, s_1)\\
&\dots
\end{aligned}$$

lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$
$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

- $\textsc{String}(a)$ for $a \in \Sigma$: creates a string "a"
- $\textsc{Concatenate}(s_1, s_2)$:
- $\textsc{Split}(s, i)$: splits into a prefix and suffix
- $\textsc{Equal}(s_1, s_2)$: equality test

Example:

$$\begin{aligned}
s_0 &:= \textsc{String}(a) \\
tmp &:= \textsc{String}(b) \\
s_1 &:= \textsc{Concatenate}(s_0, tmp) && \text{ab} \\
s_2 &:= \textsc{Concatenate}(s_1, s_0) && \text{aba} \\
s_3 &:= \textsc{Concatenate}(s_2, s_1) && \text{abaab} \\
&\dots
\end{aligned}$$

lpftest

$$X_j \;=\; \mathrm{lpf}_{B'}(X_i)$$

- $\textsc{String}(a)$ for $a \in \Sigma$: creates a string $"a"$
- $\textsc{Concatenate}(s_1, s_2)$:
- $\textsc{Split}(s, i)$: splits into a prefix and suffix
- $\textsc{Equal}(s_1, s_2)$: equality test

$$X_i \;\equiv_{B'}\; \alpha$$
$$\alpha \;=\; X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

(a) $\quad (X_i, \alpha) \in \equiv_B$

(b) $\quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $\quad (X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

Example:

$$\begin{array}{rcll} s_0 & := & \textsc{String}(a) & \\ tmp & := & \textsc{String}(b) & \\ s_1 & := & \textsc{Concatenate}(s_0, tmp) & \text{ab} \\ s_2 & := & \textsc{Concatenate}(s_1, s_0) & \text{aba} \\ s_3 & := & \textsc{Concatenate}(s_2, s_1) & \text{abaab} \\ \ldots & & & \text{abaababa} \end{array}$$

lpftest

$$X_j \;=\; \mathrm{lpf}_{B'}(X_i)$$

- $\textsc{String}(a)$ for $a \in \Sigma$: creates a string $''a''$
- $\textsc{Concatenate}(s_1, s_2)$:
- $\textsc{Split}(s, i)$: splits into a prefix and suffix
- $\textsc{Equal}(s_1, s_2)$: equality test

$$X_i \;\equiv_{B'}\; \alpha$$
$$\alpha \;=\; X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$(a) \quad (X_i, \alpha) \in\; \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr-exp}(\equiv_{B'})$

Example:

$$\begin{array}{rcl} s_0 & := & \textsc{String}(a) \\ tmp & := & \textsc{String}(b) \\ s_1 & := & \textsc{Concatenate}(s_0, tmp) \\ s_2 & := & \textsc{Concatenate}(s_1, s_0) \\ s_3 & := & \textsc{Concatenate}(s_2, s_1) \\ \ldots & & \end{array}$$

ab
aba
abaab
abaababa
abaababaabaab

lpftest

$$X_j \;=\; \mathrm{lpf}_{B'}(X_i)$$

$X_i \;\equiv_{B'}\; \alpha$

$\alpha \;=\; X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$

$(a) \quad (X_i, \alpha) \in\, \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr-exp}(\equiv_{B'})$

- $\textsc{String}(a)$ for $a \in \Sigma$: creates a string "a"
- $\textsc{Concatenate}(s_1, s_2)$:
- $\textsc{Split}(s, i)$: splits into a prefix and suffix
- $\textsc{Equal}(s_1, s_2)$: equality test

Example:

$$\begin{array}{rcll}
s_0 & := & \textsc{String}(a) & \\
tmp & := & \textsc{String}(b) & \\
s_1 & := & \textsc{Concatenate}(s_0, tmp) & \text{ab} \\
s_2 & := & \textsc{Concatenate}(s_1, s_0) & \text{aba} \\
s_3 & := & \textsc{Concatenate}(s_2, s_1) & \text{abaab} \\
\ldots & & & \text{abaababa} \\
 & & & \text{abaababaabaab} \\
 & & & \ldots
\end{array}$$

lpftest

$$X_j \; = \; \mathrm{lpf}_{B'}(X_i)$$

$$X_i \;\; \equiv_{B'} \;\; \alpha$$

$$\alpha \;\; = \;\; X_j\, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$(a) \quad (X_i, \alpha) \in \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr}{-}\mathtt{exp}(\equiv_{B'})$

lpftest

$$X_j \;=\; \mathrm{lpf}_{B'}(X_i)$$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N \text{ polylog}(N))$;
- SPLIT: $\mathcal{O}(N \text{ polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$

[Alstrup, Brodal, Rauhe 2000]

$$X_i \;\equiv_{B'}\; \alpha$$
$$\alpha \;=\; X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr-exp}(\equiv_{B'})$

lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N \text{ polylog}(N))$;
- SPLIT: $\mathcal{O}(N \text{ polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$ [Alstrup, Brodal, Rauhe 2000]

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j \, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

- exponential compression

lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N \text{ polylog}(N))$;
- SPLIT: $\mathcal{O}(N \text{ polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$

$X_i \equiv_{B'} \alpha$

$\alpha = X_j \, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$

$(a) \quad (X_i, \alpha) \in \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

[Alstrup, Brodal, Rauhe 2000]

- exponential compression
- efficient manipulation without decompression

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j \, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$(a) \quad (X_i, \alpha) \in \equiv_B$$

$$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$$

$$(c) \quad (X_i, \alpha) \in \mathtt{nr}-\mathtt{exp}(\equiv_{B'})$$

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

- String: $\mathcal{O}(N \log N)$;
- Concatenate: $\mathcal{O}(N \text{ polylog}(N))$;
- Split: $\mathcal{O}(N \text{ polylog}(N))$;
- Equal: $\mathcal{O}(\log N)$.

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j \,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$(a) \quad (X_i, \alpha) \in \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N \text{ polylog}(N))$;
- SPLIT: $\mathcal{O}(N \text{ polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$.

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j \,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr-exp}(\equiv_{B'})$

- String: $\mathcal{O}(N \log N)$;
- Concatenate: $\mathcal{O}(N \text{ polylog}(N))$;
- Split: $\mathcal{O}(N \text{ polylog}(N))$;
- Equal: $\mathcal{O}(\log N)$.

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j \,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

EQUAL

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N \text{ polylog}(N))$;
- SPLIT: $\mathcal{O}(N \text{ polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$.

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

EQUAL $\mathcal{O}(\log N)$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N \text{ polylog}(N))$;
- SPLIT: $\mathcal{O}(N \text{ polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$.

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$X_i \equiv_{B'} \alpha$

$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$

$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$

Split, Concatenate

Equal $\mathcal{O}(\log N)$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

- String: $\mathcal{O}(N \log N)$;
- Concatenate: $\mathcal{O}(N \text{ polylog}(N))$;
- Split: $\mathcal{O}(N \text{ polylog}(N))$;
- Equal: $\mathcal{O}(\log N)$.

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

SPLIT, CONCATENATE	$\mathcal{O}(N\,\mathrm{polylog}(N))$
EQUAL	$\mathcal{O}(\log N)$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- SPLIT: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$.

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

SPLIT, CONCATENATE $\mathcal{O}(N\,\mathrm{polylog}(N))$

EQUAL $\mathcal{O}(\log N)$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- SPLIT: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$.

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

SPLIT, CONCATENATE $\mathcal{O}(N\,\mathrm{polylog}(N))$

EQUAL $\mathcal{O}(\log N)$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- SPLIT: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$.

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

SPLIT, CONCATENATE $\mathcal{O}(N\,\mathrm{polylog}(N))$

EQUAL $\mathcal{O}(\log N)$

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta\, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- SPLIT: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$.

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

Split, Concatenate $\mathcal{O}(N\,\mathrm{polylog}(N))$

Equal $\mathcal{O}(\log N)$

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(\gamma) = \mathrm{dec}_B(\delta)\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

- String: $\mathcal{O}(N \log N)$;
- Concatenate: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- Split: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- Equal: $\mathcal{O}(\log N)$.

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

SPLIT, CONCATENATE $\mathcal{O}(N\,\mathrm{polylog}(N))$

EQUAL $\mathcal{O}(\log N)$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(\gamma) = \mathrm{dec}_B(\delta)\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- SPLIT: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$.

$\mathcal{O}(N^2)$ invocations

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

Split, Concatenate $\mathcal{O}(N\,\mathrm{polylog}(N))$

Equal $\mathcal{O}(\log N)$

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(\gamma) = \mathrm{dec}_B(\delta)\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

- String: $\mathcal{O}(N \log N)$;
- Concatenate: $\mathcal{O}(N\ \mathrm{polylog}(N))$; $\mathcal{O}(N)$ invocations
- Split: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- Equal: $\mathcal{O}(\log N)$. $\mathcal{O}(N^2)$ invocations

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

SPLIT, CONCATENATE $\mathcal{O}(N\,\mathrm{polylog}(N))$ — (a) $(X_i, \alpha) \in \equiv_B$

EQUAL $\mathcal{O}(\log N)$

$\mathcal{O}(N^2\,\mathrm{polylog}(N))$ — (b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(\gamma) = \mathrm{dec}_B(\delta)\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N\ \mathrm{polylog}(N))$; $\mathcal{O}(N)$ invocations
- SPLIT: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$. $\mathcal{O}(N^2)$ invocations

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

SPLIT, CONCATENATE	$\mathcal{O}(N\,\mathrm{polylog}(N))$		
EQUAL	$\mathcal{O}(\log N)$	(a)	$(X_i, \alpha) \in \equiv_B$
	$\mathcal{O}(N^2\,\mathrm{polylog}(N))$	(b)	$(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$
		(c)	$(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(\gamma) = \mathrm{dec}_B(\delta)\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N\ \mathrm{polylog}(N))$; $\mathcal{O}(N)$ invocations
- SPLIT: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$. $\mathcal{O}(N^2)$ invocations

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

SPLIT, CONCATENATE $\mathcal{O}(N\,\mathrm{polylog}(N))$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

EQUAL $\mathcal{O}(\log N)$ (a) $(X_i, \alpha) \in \equiv_B$

$\mathcal{O}(N^2\,\mathrm{polylog}(N))$ (b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

similarly (c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(\gamma) = \mathrm{dec}_B(\delta)\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N\ \mathrm{polylog}(N))$; $\mathcal{O}(N)$ invocations
- SPLIT: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$. $\mathcal{O}(N^2)$ invocations

Time cost of lpftest

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(X_i) = \mathrm{dec}_B(\alpha)$$

SPLIT, CONCATENATE	$\mathcal{O}(N\,\mathrm{polylog}(N))$	(a) $(X_i, \alpha) \in \equiv_B$
EQUAL	$\mathcal{O}(\log N)$	(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$
dominating	$\mathcal{O}(N^2\,\mathrm{polylog}(N))$	
	similarly	(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(\gamma) = \mathrm{dec}_B(\delta)\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

- STRING: $\mathcal{O}(N \log N)$;
- CONCATENATE: $\mathcal{O}(N\ \mathrm{polylog}(N))$; $\mathcal{O}(N)$ invocations
- SPLIT: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- EQUAL: $\mathcal{O}(\log N)$. $\mathcal{O}(N^2)$ invocations

Total time cost

Total time cost

BPA: $\mathcal{O}(N^4 \, \mathrm{polylog}(N))$

Total time cost

BPA: $\mathcal{O}(N^4 \operatorname{polylog}(N))$

Simple grammars: $\mathcal{O}(N^3 \operatorname{polylog}(N))$

Simple grammars

$$X_j = \text{lpf}_{B'}(X_i)$$

$\mathcal{O}(N \,\text{polylog}(N))$ $\quad X_i \equiv_{B'} \alpha$

$\alpha = X_j \,\text{suffix}_{|X_i|-|X_j|}(\text{dec}_{B'}(\alpha_i))$

$\mathcal{O}(\log N)$ $\quad (a) \quad (X_i, \alpha) \in \equiv_B$

$\mathcal{O}(N^2 \,\text{polylog}(N))$ $\quad (b) \quad (X_i, \alpha) \in \texttt{exp}(\equiv_B)$

similarly $\quad (c) \quad (X_i, \alpha) \in \texttt{nr-exp}(\equiv_{B'})$

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta \,\text{suffix}_{|X_i|-|X_j|}(\text{dec}_{B'}(\alpha_i))$$

$$\text{dec}_B(\gamma) = \text{dec}_B(\delta)\,\text{suffix}_{|X_i|-|X_j|}(\text{dec}_{B'}(\alpha_i))$$

- String: $\mathcal{O}(N \log N)$;
- Concatenate: $\mathcal{O}(N \,\text{polylog}(N))$; $\quad \mathcal{O}(N)$ invocations
- Split: $\mathcal{O}(N \,\text{polylog}(N))$;
- Equal: $\mathcal{O}(\log N)$. $\quad \mathcal{O}(N^2)$ invocations

Simple grammars

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j \,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$\mathcal{O}(N^2 \,\mathrm{polylog}(N))$

similarly

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta \,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(\gamma) = \mathrm{dec}_B(\delta)\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

- String: $\mathcal{O}(N \log N)$;
- Concatenate: $\mathcal{O}(N \,\mathrm{polylog}(N))$; $\mathcal{O}(N)$ invocations
- Split: $\mathcal{O}(N \,\mathrm{polylog}(N))$;
- Equal: $\mathcal{O}(\log N)$. $\mathcal{O}(N^2)$ invocations

Simple grammars

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

assume $\{a : X_i \xrightarrow{a} \gamma\} = \{a : X_j \xrightarrow{a} \delta\}$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j \,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$(a) \quad (X_i, \alpha) \in \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr-exp}(\equiv_{B'})$

$\mathcal{O}(N^2 \,\mathrm{polylog}(N))$

similarly

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta \,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(\gamma) = \mathrm{dec}_B(\delta) \,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

- String: $\mathcal{O}(N \log N)$;
- Concatenate: $\mathcal{O}(N \,\mathrm{polylog}(N))$; $\mathcal{O}(N)$ invocations
- Split: $\mathcal{O}(N \,\mathrm{polylog}(N))$;
- Equal: $\mathcal{O}(\log N)$. $\mathcal{O}(N^2)$ invocations

Simple grammars

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

assume $\{a : X_i \xrightarrow{a} \gamma\} = \{a : X_j \xrightarrow{a} \delta\}$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j \,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

(a) $(X_i, \alpha) \in \equiv_B$

(b) $(X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

(c) $(X_i, \alpha) \in \mathtt{nr-exp}(\equiv_{B'})$

$\mathcal{O}(N^2 \,\mathrm{polylog}(N))$

similarly

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta \,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(\gamma) = \mathrm{dec}_B(\delta)\,\mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

- String: $\mathcal{O}(N \log N)$;
- Concatenate: $\mathcal{O}(N \,\mathrm{polylog}(N))$; $\mathcal{O}(N)$ invocations
- Split: $\mathcal{O}(N \,\mathrm{polylog}(N))$;
- Equal: $\mathcal{O}(\log N)$. $\mathcal{O}(N)$ invocations

Simple grammars

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

assume $\{a : X_i \xrightarrow{a} \gamma\} = \{a : X_j \xrightarrow{a} \delta\}$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j \, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$(a) \quad (X_i, \alpha) \in \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr-exp}(\equiv_{B'})$

$\mathcal{O}(N^2 \, \mathrm{polylog}(N))$

similarly

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta \, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(\gamma) = \mathrm{dec}_B(\delta) \, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

- String: $\mathcal{O}(N \log N)$;
- Concatenate: $\mathcal{O}(N \ \mathrm{polylog}(N))$; $\mathcal{O}(N)$ invocations
- Split: $\mathcal{O}(N \ \mathrm{polylog}(N))$;
- Equal: $\mathcal{O}(\log N)$.

Simple grammars

$$X_j = \mathrm{lpf}_{B'}(X_i)$$

assume $\{a : X_i \xrightarrow{a} \gamma\} = \{a : X_j \xrightarrow{a} \delta\}$

$$X_i \equiv_{B'} \alpha$$

$$\alpha = X_j\, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$(a) \quad (X_i, \alpha) \in \equiv_B$

$(b) \quad (X_i, \alpha) \in \mathtt{exp}(\equiv_B)$

$(c) \quad (X_i, \alpha) \in \mathtt{nr{-}exp}(\equiv_{B'})$

$\mathcal{O}(N^2\, \mathrm{polylog}(N))$

similarly

$$X_i \xrightarrow{a} \gamma \qquad X_j \xrightarrow{a} \delta$$

$$\gamma \equiv_B \delta\, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

$$\mathrm{dec}_B(\gamma) = \mathrm{dec}_B(\delta)\, \mathrm{suffix}_{|X_i|-|X_j|}(\mathrm{dec}_{B'}(\alpha_i))$$

- String: $\mathcal{O}(N \log N)$;
- Concatenate: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- Split: $\mathcal{O}(N\ \mathrm{polylog}(N))$;
- Equal: $\mathcal{O}(\log N)$.

$\mathcal{O}(N^2)$ invocations in all lpftest's

Total time cost

BPA: $\mathcal{O}(N^4 \,\mathrm{polylog}(N))$

Simple grammars: $\mathcal{O}(N^3 \,\mathrm{polylog}(N))$

BPP: $\mathcal{O}(N^3)$ [Jancar 2003]

Outline

- Background
  - Context-free graphs and commutative context-free graphs
  - Bisimulation equivalence problem
  - Norm
  - History of the problem
  - Unique decomposition
  - Naive algorithm
- Efficient algorithm for BPA and BPP
  - Outline of the algorithm
  - Refinement
  - Efficient computation of refinement for BPA
  - Time-cost analysis
  - **Partially-commutative context-free graphs**

One algorithm for both BPA and BPP?

One algorithm for both BPA and BPP?

Yes, in the framework of partially-commutative context-free graphs

Partial commutation

partially-commutative CFGs :

instead of words or multisets, consider partial orders

Partial commutation

partially-commutative CFGs :
instead of words or multisets, consider partial orders

Example:

Partial commutation

partially-commutative CFGs :
instead of words or multisets, consider partial orders

Example:

variables: $\{\ A, B, C, D, E\ \}$

Partial commutation

partially-commutative CFGs :
instead of words or multisets, consider partial orders

Example:

variables: $\{ A, B, C, D, E \}$

dependence: (A, B), (B, C), (C, E), (D, E)

Partial commutation

partially-commutative CFGs :
instead of words or multisets, consider partial orders

Example:

variables: $\{ A, B, C, D, E \}$

dependence: (A, B), (B, C), (C, E), (D, E)

independence: (A, C), (A, D), (A, E), (B, D), (B, E), (C, D)

Partial commutation

partially-commutative CFGs :
instead of words or multisets, consider partial orders

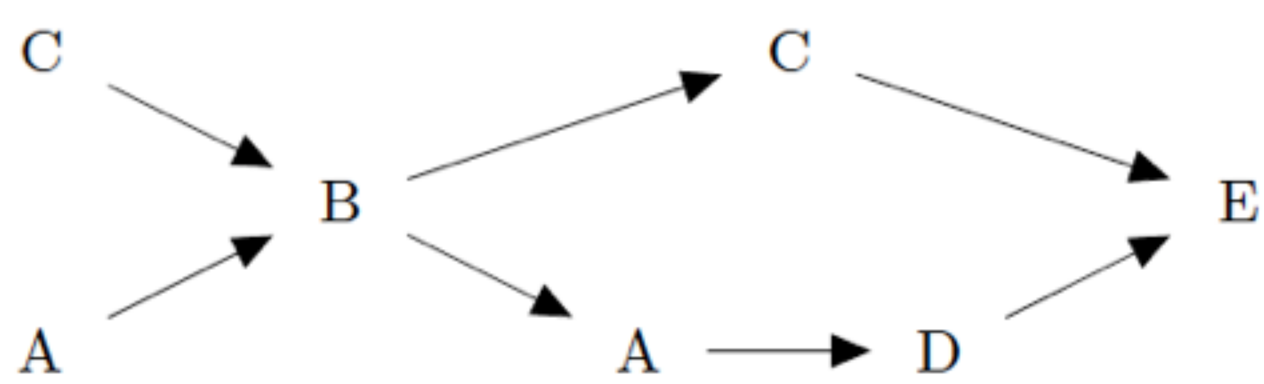
Example:

variables: $\{\ A, B, C, D, E\ \}$

dependence: (A, B), (B, C), (C, E), (D, E)

independence: (A, C), (A, D), (A, E), (B, D), (B, E), (C, D)

configuration:



Partial commutation

partially-commutative CFGs :
instead of words or multisets, consider partial orders

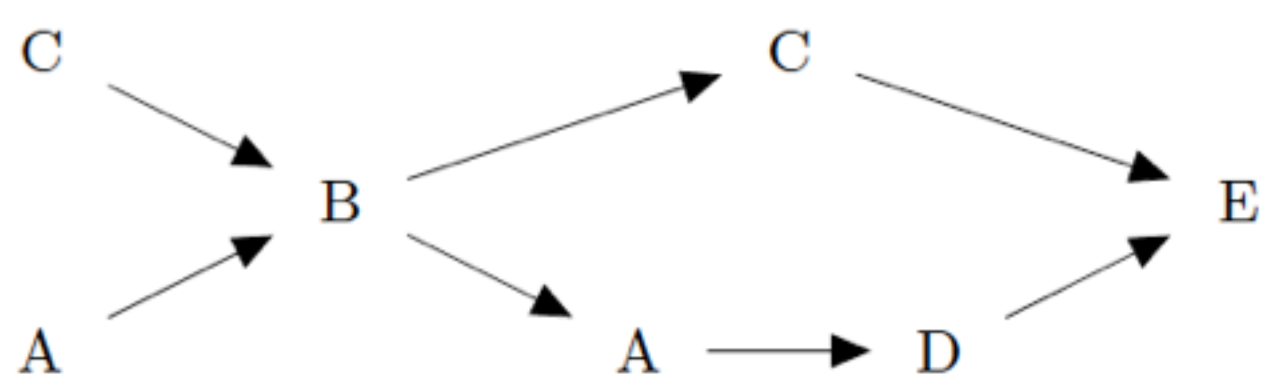
Example:

variables: $\{\ A, B, C, D, E\ \}$

dependence: (A, B), (B, C), (C, E), (D, E)

independence: (A, C), (A, D), (A, E), (B, D), (B, E), (C, D)

configuration:



BPA and BPP are special cases

Partial commutation

partially-commutative CFGs :
instead of words or multisets, consider partial orders

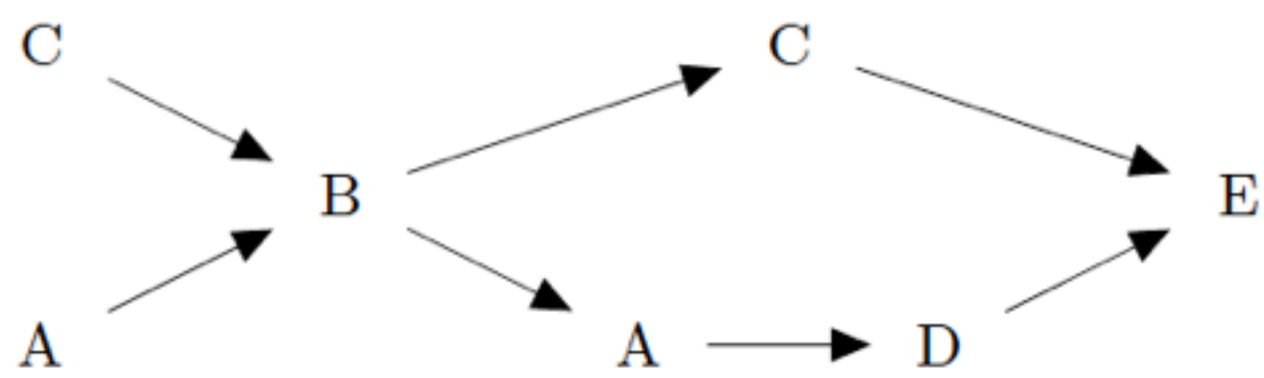
Example:

variables: $\{ A, B, C, D, E \}$

dependence: (A, B), (B, C), (C, E), (D, E)

independence: (A, C), (A, D), (A, E), (B, D), (B, E), (C, D)

configuration:



no independence: $C\ A\ B\ A\ C\ D\ E$

BPA and BPP are special cases

Partial commutation

partially-commutative CFGs :
instead of words or multisets, consider partial orders

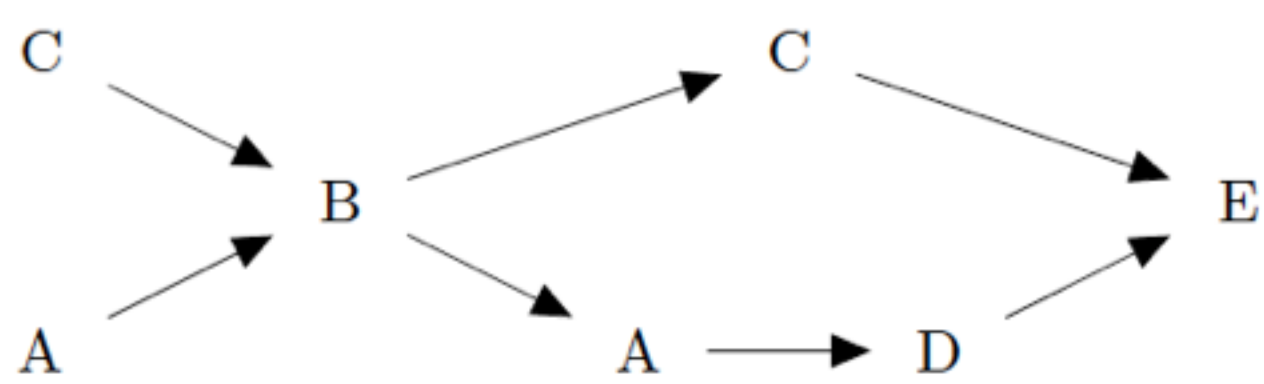
Example:

variables: $\{\ A, B, C, D, E\ \}$

dependence: (A, B), (B, C), (C, E), (D, E)

independence: (A, C), (A, D), (A, E), (B, D), (B, E), (C, D)

configuration:



BPA and BPP are special cases

no independence: $C\,A\,B\,A\,C\,D\,E$

no dependence: $A^2\,B\,C^2\,DE$

Open question

	normed	unnormed
BPA	P-complete	EXPTIME-hard in 2-EXPTIME
BPP	P-complete	PSPACE-complete

Open question

	normed	unnormed
BPA	P-complete	EXPTIME-hard in 2-EXPTIME
BPP	P-complete	PSPACE-complete

Open question

	normed	unnormed
BPA	P-complete	EXPTIME-hard in 2-EXPTIME
BPP	P-complete	PSPACE-complete

- the Prover-Refuter Game may be restricted to configurations of exponential length

Open question

	normed	unnormed
BPA	P-complete	EXPTIME-hard in 2-EXPTIME
BPP	P-complete	PSPACE-complete

- the Prover-Refuter Game may be restricted to configurations of exponential length
- can these configurations be compressed exponentially?

attractive PhD positions

at the University of Warsaw

thank you!