Faster Algorithm for Bisimulation Equivalence of Normed Context-Free Processes

Sławomir Lasota Wojciech Rytter

Warsaw University

GAMES'06

(to be presented at MFCS'06)

Context-free processes

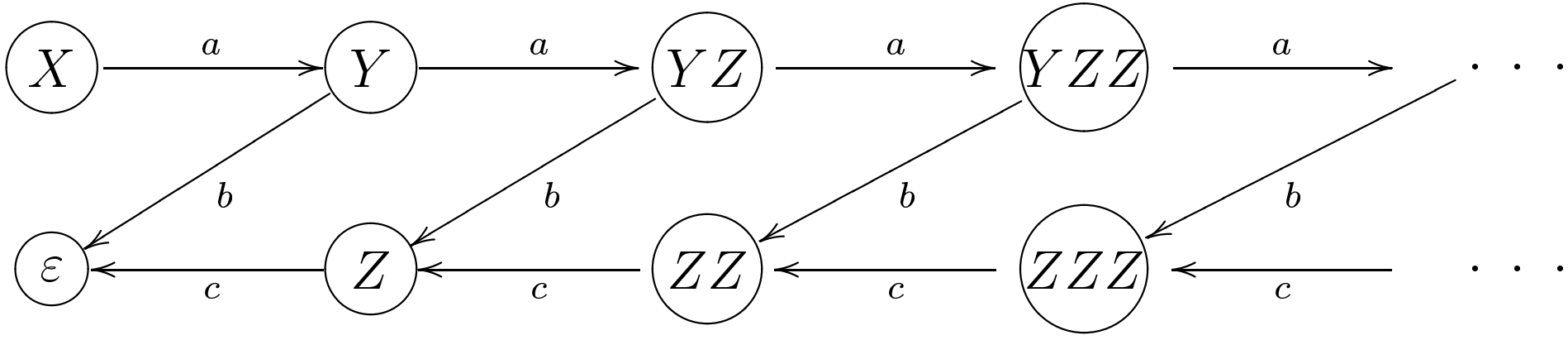
- CFG in Greibach Normal Form:

$$X \longrightarrow aY \qquad Y \longrightarrow aYZ \qquad Y \longrightarrow b \qquad Z \longrightarrow c.$$

- Rewrite rules:

$$X \stackrel{a}{\longrightarrow} Y \qquad Y \stackrel{a}{\longrightarrow} YZ \qquad Y \stackrel{b}{\longrightarrow} \varepsilon \qquad Z \stackrel{c}{\longrightarrow} \varepsilon.$$

- Process graph:



Equivalences on context-free processes

- The language equivalence $L(X) = L(Y)$ is undecidable
- The bisimulation equivalence $X \sim Y$ is decidable (2-EXP)

Normed context-free processes

In the sequel assume that all non-terminals X are normed:

Process view: X has a finite sequence

$$X \xrightarrow{a_1} \alpha_1 \ldots \xrightarrow{a_k} \alpha_k = \varepsilon$$

leading to the empty process.

Language view: $L(X)$ is non-empty.

Norm: $|X| =$ the smallest k.

Equivalences on normed c.-f. processes

- The language equivalence $L(X) = L(Y)$ is undecidable
- The bisimulation equivalence $X \sim Y$ is in polynomial-time

Simple grammars

- Simple (deterministic) grammars:

$$X \xrightarrow{a} \alpha, \quad X \xrightarrow{a} \beta \quad \Longrightarrow \quad \alpha = \beta$$

The language equivalence problem for simple grammars is subsumed by the bisimulation equivalence problem.

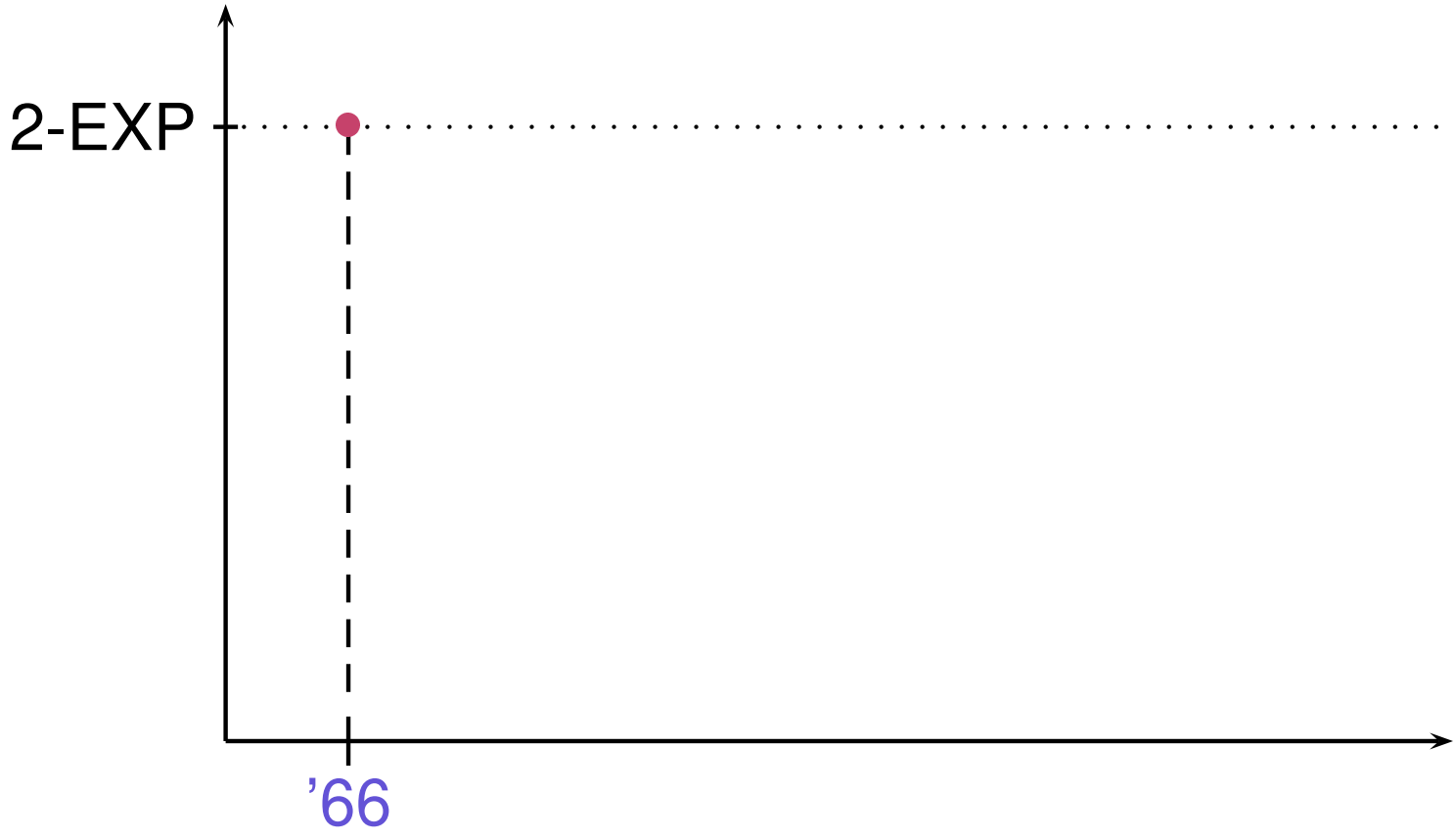
In the sequel we cosider two problems:

- equivalence of simple grammars
- bisimulation equivalence of normed processes

History of the problems

• - eq. of simple grammars

• - bisimulation eq.

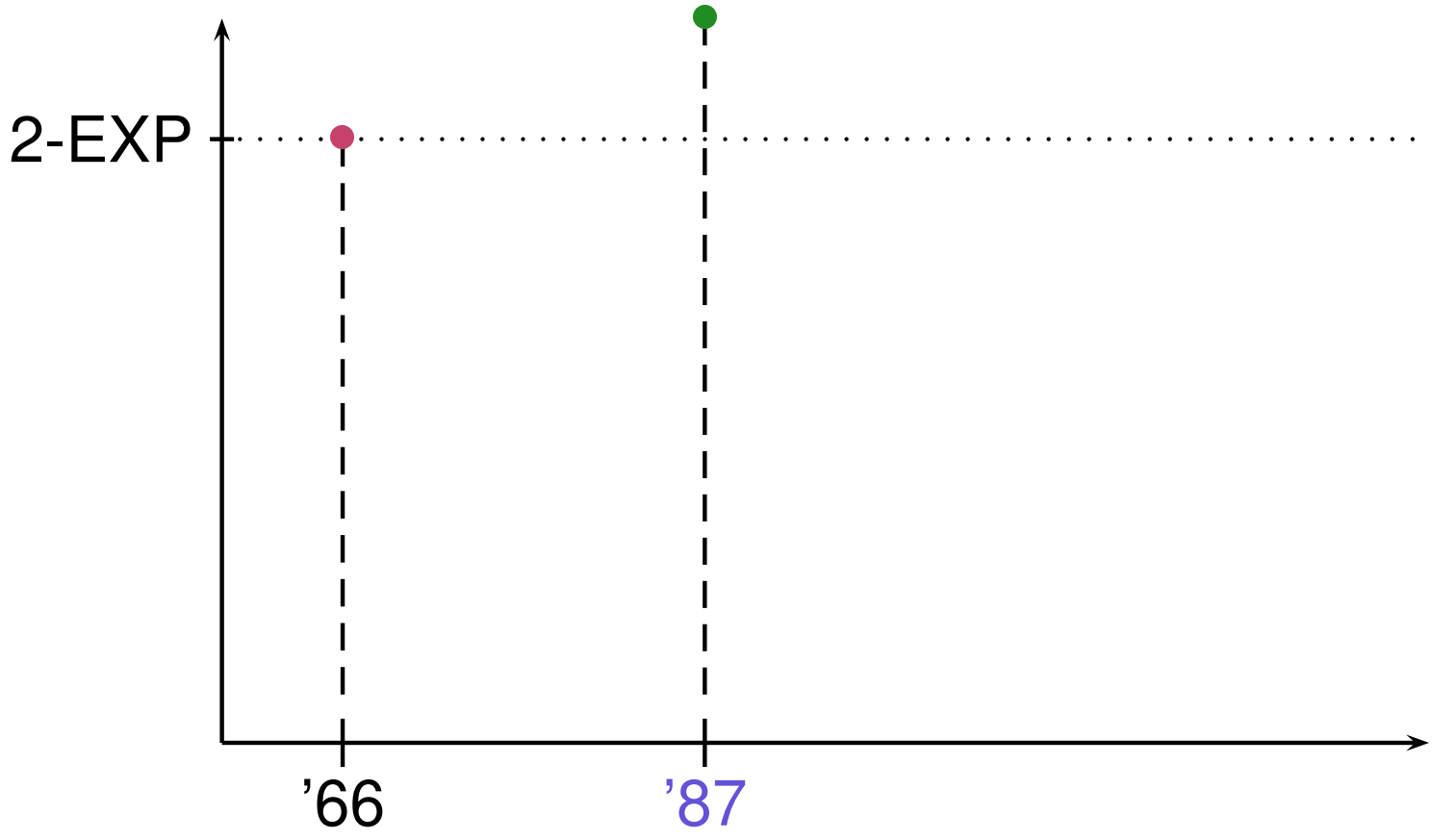


[Korenjak, Hopcroft]

History of the problems

• - eq. of simple grammars

• - bisimulation eq.

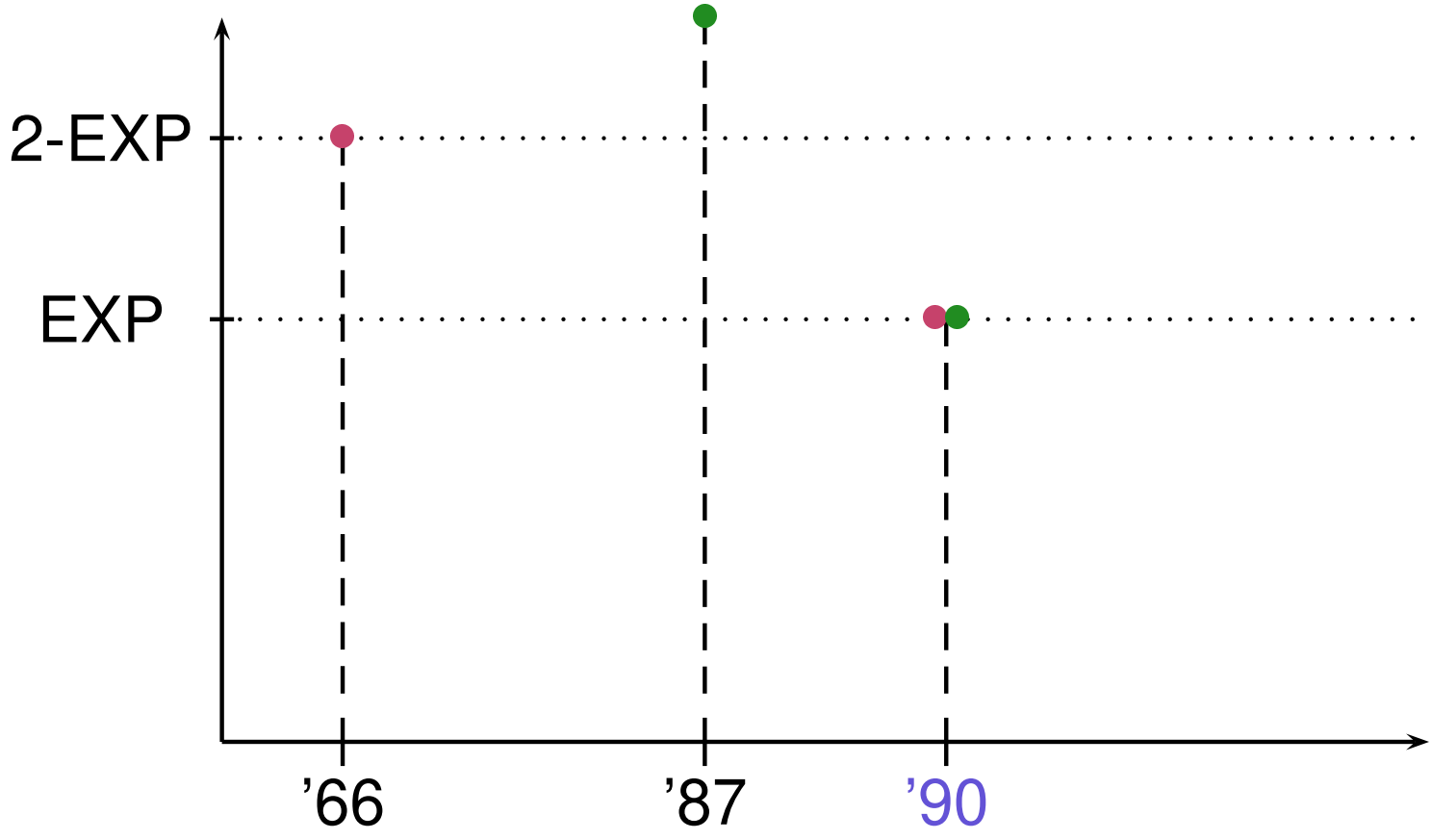


[Beaten, Bergstra, Klop]

History of the problems

• - eq. of simple grammars

• - bisimulation eq.

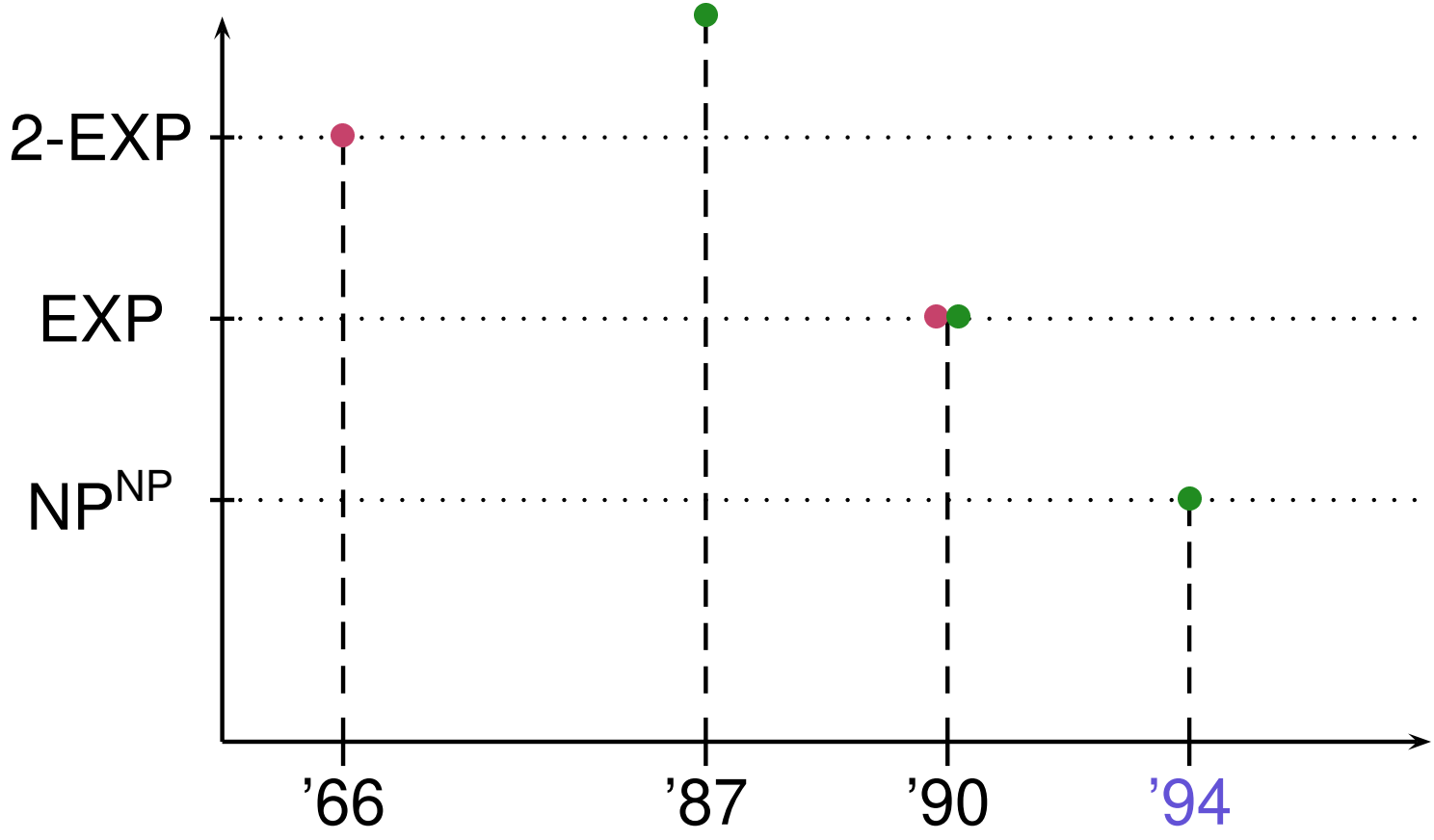


[Caucal, Huttel, Stirling, Groote, . . .]

History of the problems

• - eq. of simple grammars

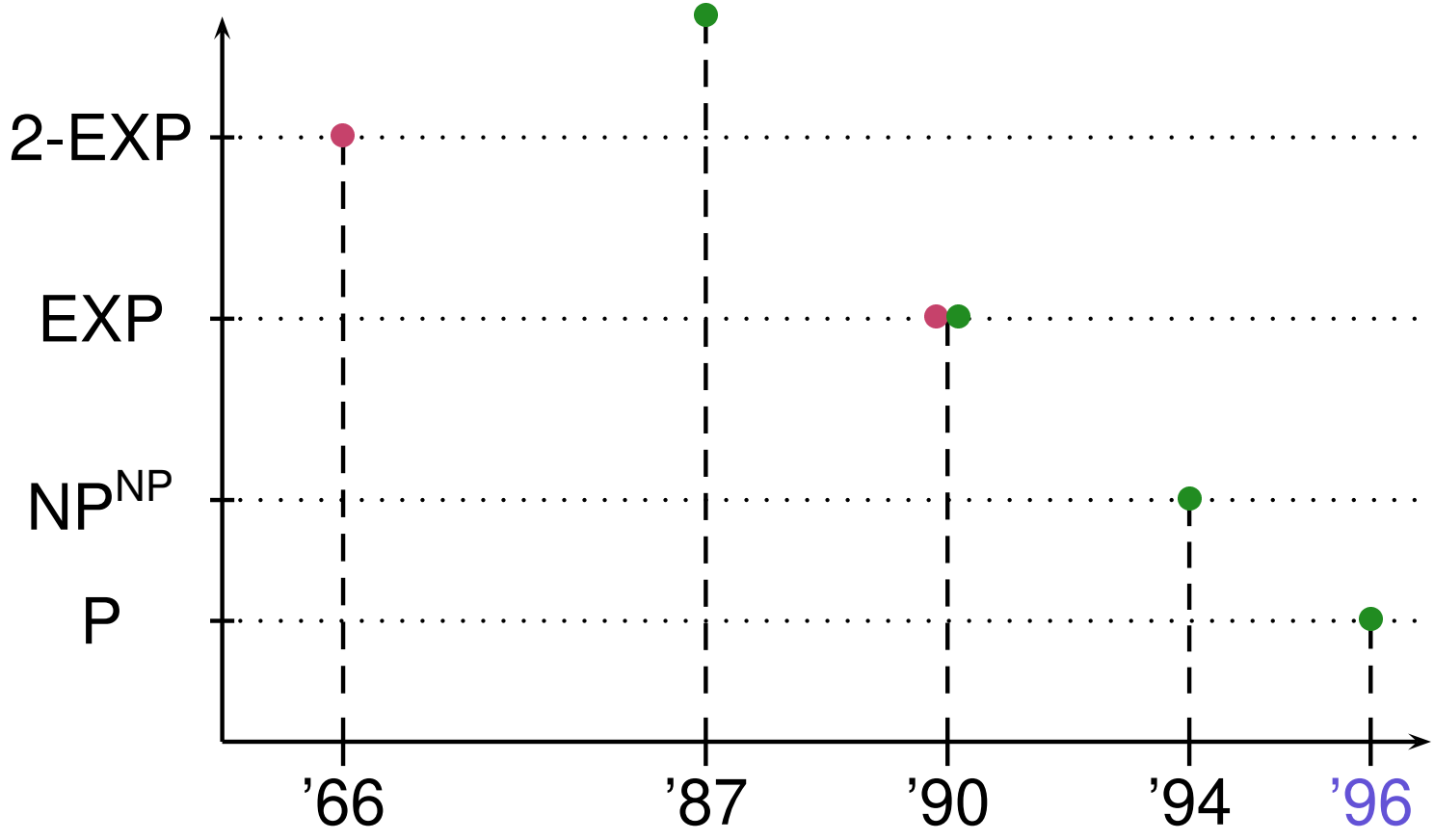
• - bisimulation eq.



[Huynh, Tian]

History of the problems

• - eq. of simple grammars • - bisimulation eq.



[Hirshfeld, Jerrum, Moller]

State of the art

Thm [Hirshfeld, Jerrum, Moller '96]:
Bisimulation equivalence can be decided in time $\mathcal{O}(n^{13})$.

Thm [Bastien,Czyzowicz,Fraczak,Rytter '05]:
Equivalence of simple grammars can be decided in time $\widetilde{\mathcal{O}}(n^{7})$.

State of the art

Thm [Hirshfeld, Jerrum, Moller '96]:
Bisimulation equivalence can be decided in time $\mathcal{O}(n^{13})$.

Thm [Bastien,Czyzowicz,Fraczak,Rytter '05]:
Equivalence of simple grammars can be decided in time $\widetilde{\mathcal{O}}(n^7)$.

- exponential-time algorithm of [Caucal'90]
- compressed string-matching: First-Mismatch Problem
  [Karpiński, Rytter, Shinohara '95]
  [Shinohara, Miyazaki, Takeda '00]

Our results

Thm [Hirshfeld, Jerrum, Moller '96]:

Bisimulation equivalence can be decided in time $\mathcal{O}(n^{2.3})$.

Thm [Bastien,Czyzowicz,Fraczak,Rytter '05]:

Equivalence of simple grammars can be decided in time $\widetilde{\mathcal{O}}(n^7)$.

Our results:

Thm: Bisimulation equivalence $\mapsto$ $\widetilde{\mathcal{O}}(n^8)$.

Equivalence of simple grammars $\mapsto$ $\widetilde{\mathcal{O}}(n^6)$.

The insights

Insight 1:
Encode the bisimulation problem in a system of boolean equations, instead of iterative computation of the greatest fixed point.

Insight 2:
An improved algorithm for the FIRST-MISMATCH PROBLEM.

Algorithm

Algorithm:

- construct a system of boolean equations
- find the greatest solution

Algorithm

Algorithm:

- construct a system of boolean equations — hard
- find the greatest solution — easy

Insight 1: Boolean encoding

Prop. (left-cancellation): $\alpha\,\gamma \;\sim\; \alpha\,\gamma' \implies \gamma \sim \gamma'$.

Corollary: $B \sim A\,\gamma \implies \gamma$ unique up to $\sim$.

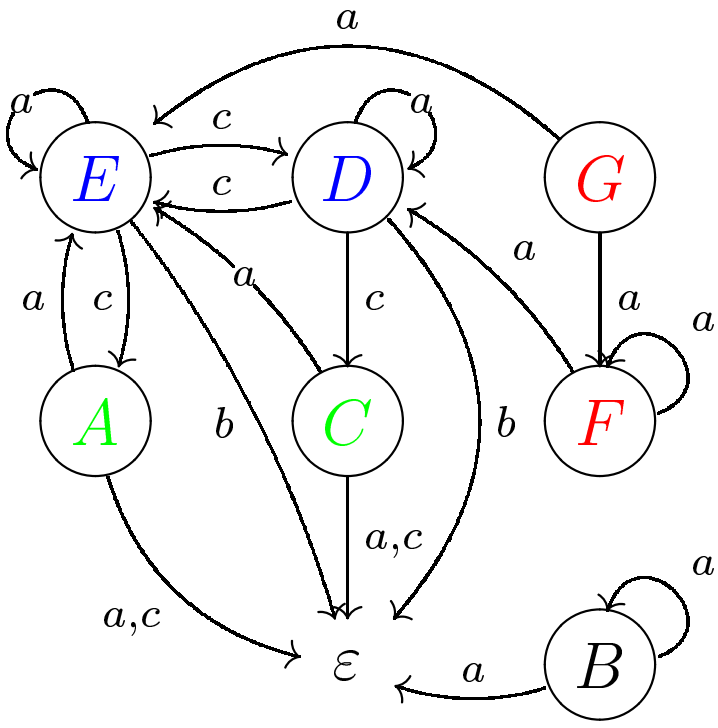
Encoding: $z_{BA} \equiv B \sim A\,\gamma$ for some γ.

Boolean equation system:

$$z_{BA} = \bigwedge_a \Big(\bigwedge_\beta \bigvee_\alpha \phi_{\beta,\alpha\gamma} \wedge \bigwedge_\alpha \bigvee_\beta \phi_{\beta,\alpha\gamma} \Big) \qquad (A \xrightarrow{a} \alpha \text{ and } B \xrightarrow{a} \beta)$$

Intuition: $\phi_{\beta,\alpha\gamma}$ approximates $\beta \sim \alpha\gamma$ (cf Insight 2).

Insight 1: Example



$$|A| = \ldots = |E| = 1$$
$$|F| = |G| = 2$$

$$z_{GB} \equiv G \sim BE$$

The greatest solution:

$$\{z_{CA}, z_{DE}, z_{GF}, z_{FB}, z_{GB}\}$$

$$z_{GF} = \left(\phi_{E,D} \vee \phi_{E,F}\right) \wedge \left(\phi_{F,D} \vee \phi_{F,F}\right) \wedge \ \ldots \qquad \mapsto \qquad z_{GF} = z_{ED}$$

$$z_{ED} = z_{ED} \wedge \left(z_{EA} \vee z_{CA}\right) \wedge \left(z_{ED} \vee z_{DC}\right) \wedge \left(z_{CA} \vee z_{DC}\right) \wedge \left(z_{EA} \vee z_{ED}\right)$$

Intuition: $\phi_{\alpha,\beta}$ approximates $\alpha \sim \beta$.

Whenever

$$A \sim B\ \gamma$$

one can freely substitute

$$A \mapsto B\ \gamma.$$

By this we

- resolve a potential mismatch $A \neq B$
- "eliminate" non-terminal A.

Insight 2: First-Mismatch Problem

Input: Two words α, β and a CFG

$$A \longrightarrow \gamma$$

precisely one production for each non-terminal A.

grammar $=$ compressed representation of a single word

Question: Is $\alpha = \beta$ provable by substitutions $A \mapsto \gamma$?
If no, compute the left-most mismatching pair.

Insight 2: Example

$$A \longrightarrow BBD \qquad C \longrightarrow DE$$

- $AEBBBD = BBCCDA$ is not provable:

$$AEBBBD \xrightarrow{A \mapsto BBD} BBDEBBBD$$

$$BBCCDA \xrightarrow{C \mapsto DE} BBDEDEDA$$

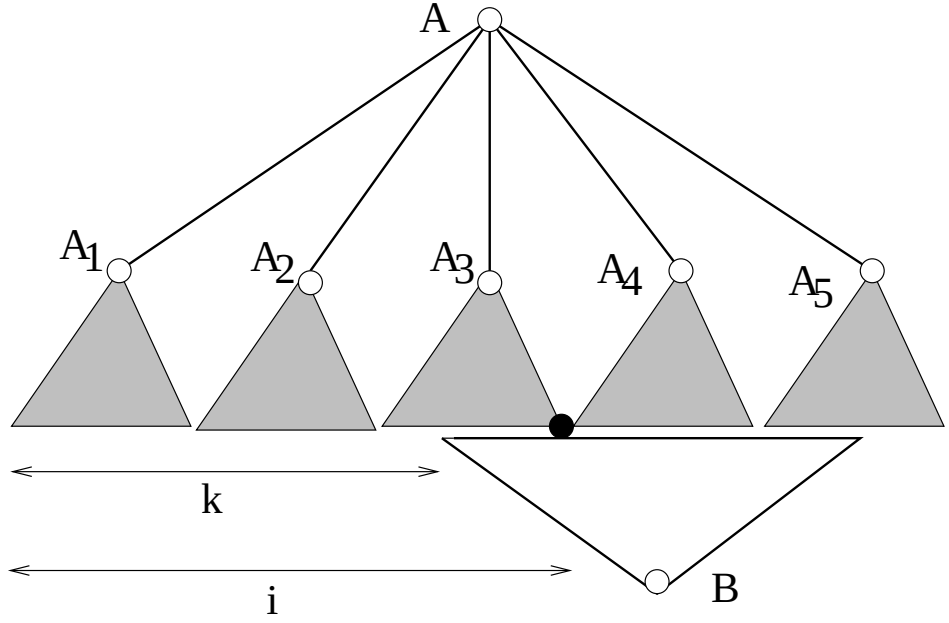
- Add $B \longrightarrow CD$; $AEBBBD = BBCCDA$ is now provable:

$$AEBBBD \xrightarrow{A \mapsto BBD} BBDEBBBD \xrightarrow{B \mapsto CD} BBDECDBBD$$

$$BBCCDA \xrightarrow{C \mapsto DE} BBDECDA \xrightarrow{A \mapsto BBD} BBDECDBBD$$

Insight 2: Result

Thm: The First-Mismatch Problem can be solved in time $\tilde{\mathcal{O}}(n^5)$.



Future plans

- Further progress in compressed string maching
- Apply the ideas to unnormed processes