Forwards- and Backwards-Reachability for Cooperating Multi-Pushdown Systems

24th International Symposium on Fundamentals of Computation Theory, Trier

Chris Köcher[1,(2)] Dietrich Kuske[2]

[1]Max Planck Institute for Software Systems, Kaiserslautern
[2]Technische Universität Ilmenau

September 21, 2023

Motivation

- Consider automata with one or more pushdowns.
  - Model distributed systems with recursive function calls.
- In general, 2-pushdown automata are Turing-complete!
  - $\Rightarrow$ Verification problems are undecidable.
- Here: consider a special restriction to the automata.
  - $\rightsquigarrow$ cooperating multi-pushdown systems

(Mazurkiewicz) Traces

- Distributed alphabet:

$$\mathcal{D} = \left(\quad A \quad , \quad P \quad , \quad \mathcal{L} \quad \right)$$

alphabet of tasks

finite set of processes

$\mathcal{L} \subseteq A \times P$

(Mazurkiewicz) Traces

- Distributed alphabet:

$$\mathcal{D} = \left(\{a, b, c\}, \{1, 2\}, \begin{tikzcd}[arrows=dash, column sep=small, row sep=small] a \arrow[d] & b \arrow[d] & c \arrow[dll] \arrow[dl] \\ 1 & 2 & \end{tikzcd} \right)$$

alphabet of tasks — finite set of processes — $\mathcal{L} \subseteq A \times P$

- $\mathcal{D}$ induces vectors: $\binom{a}{\varepsilon}$, $\binom{\varepsilon}{b}$, $\binom{c}{c}$
- Trace monoid (or *free partially commutative monoid*): $\mathbb{M}(\mathcal{D}) = \left\{ \binom{a}{\varepsilon}, \binom{\varepsilon}{b}, \binom{c}{c} \right\}^*$
  - $\binom{a}{\varepsilon} \cdot \binom{\varepsilon}{b} = \binom{a}{b} = \binom{\varepsilon}{b} \cdot \binom{a}{\varepsilon}$
  - $\binom{a}{\varepsilon} \cdot \binom{c}{c} = \binom{ac}{c} \neq \binom{ca}{c} = \binom{c}{c} \cdot \binom{a}{\varepsilon}$
- Processes in $\tau \in \mathbb{M}(\mathcal{D})$: $P_\tau = \{ i \in P \mid \tau[i] \neq \varepsilon \}$

Cooperating Multi-Pushdown Systems (CPDS)

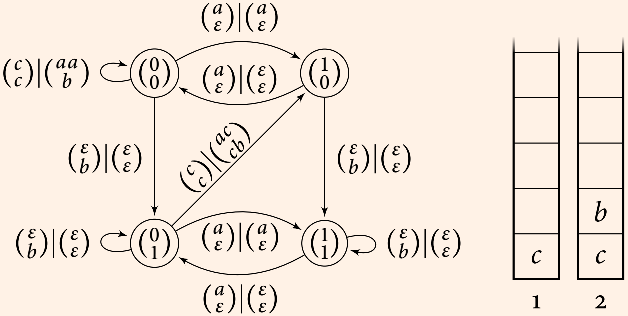
$$\mathfrak{P} = \Big(\prod_{i \in P} Q_i \, , \, \Delta \Big)$$

global states

finite sets of local states

finite set of transitions $\overline{p} \xrightarrow{\alpha \mid \tau} \overline{q}$ with $\overline{p}, \overline{q} \in \prod_{i \in P_\alpha} Q_i$ and $P_\tau \subseteq P_\alpha$

Example



Cooperating Multi-Pushdown Systems (CPDS)

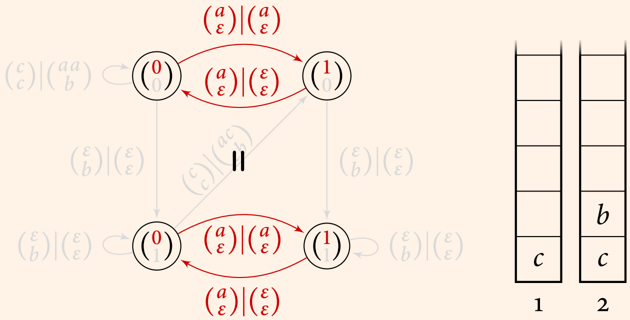
$$\mathfrak{P} = \left(\prod_{i \in P} Q_i \,,\, \Delta \right)$$

global states

finite sets of local states

finite set of transitions $\overline{p} \xrightarrow{\alpha \,|\, \tau} \overline{q}$ with $\overline{p}, \overline{q} \in \prod_{i \in P_\alpha} Q_i$ and $P_\tau \subseteq P_\alpha$

Example



Cooperating Multi-Pushdown Systems (CPDS)

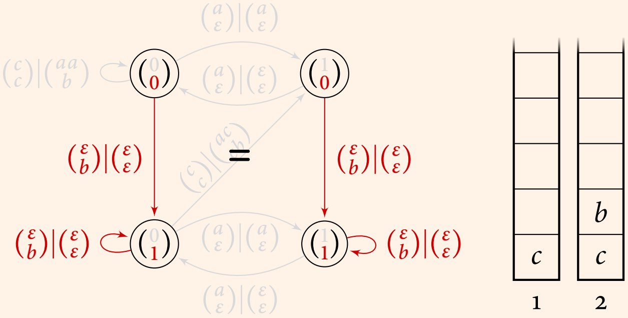
$$\mathfrak{P} = \left(\prod_{i \in P} Q_i \, , \, \Delta \right)$$

global states

finite sets of local states

finite set of transitions $\overline{p} \xrightarrow{\alpha \mid \tau} \overline{q}$ with $\overline{p}, \overline{q} \in \prod_{i \in P_\alpha} Q_i$ and $P_\tau \subseteq P_\alpha$

Example



Cooperating Multi-Pushdown Systems (CPDS)

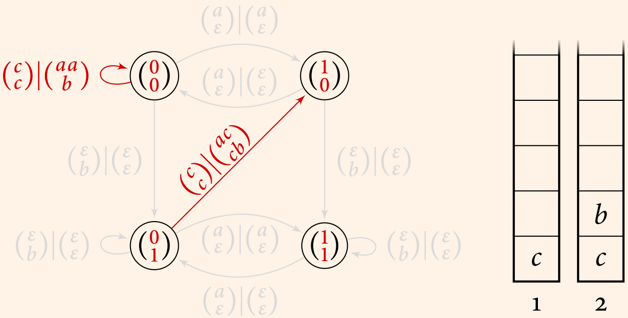
$$\mathfrak{P} = \Big(\prod_{i \in P} Q_i \, , \, \Delta \Big)$$

global states

finite sets of local states

finite set of transitions $\overline{p} \xrightarrow{\alpha \mid \tau} \overline{q}$ with $\overline{p}, \overline{q} \in \prod_{i \in P_\alpha} Q_i$ and $P_\tau \subseteq P_\alpha$

Example



Cooperating Multi-Pushdown Systems (CPDS)

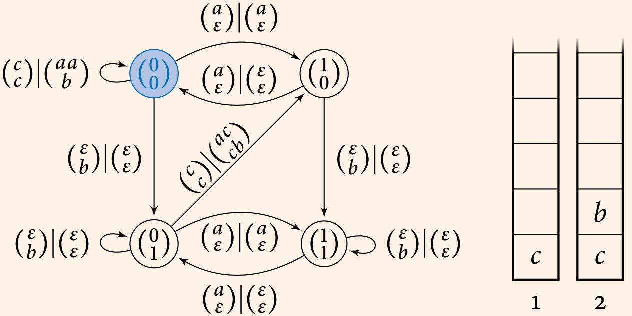
$$\mathfrak{P} = \Big(\prod_{i \in P} Q_i \, , \, \Delta \Big)$$

global states

finite sets of local states

finite set of transitions $\overline{p} \xrightarrow{\alpha \,|\, \tau} \overline{q}$
with $\overline{p}, \overline{q} \in \prod_{i \in P_\alpha} Q_i$ and $P_\tau \subseteq P_\alpha$

Example



Cooperating Multi-Pushdown Systems (CPDS)

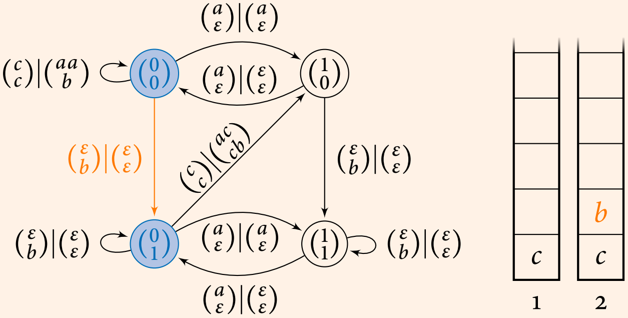
$$\mathfrak{P} = \Big(\prod_{i \in P} Q_i \,,\, \Delta \Big)$$

global states

finite sets of local states

finite set of transitions $\overline{p} \xrightarrow{\alpha \,|\, \tau} \overline{q}$
with $\overline{p}, \overline{q} \in \prod_{i \in P_\alpha} Q_i$ and $P_\tau \subseteq P_\alpha$

Example



Cooperating Multi-Pushdown Systems (CPDS)

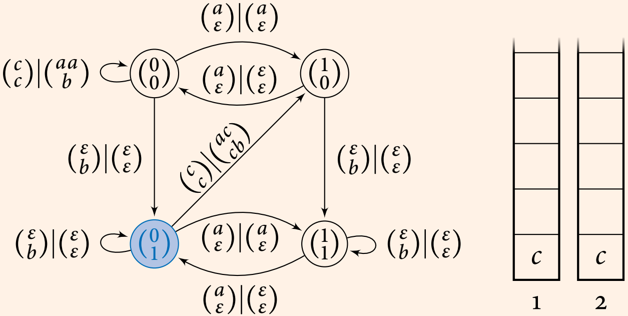
$$\mathfrak{P} = \Big(\prod_{i \in P} Q_i \, , \, \Delta \Big)$$

global states

finite sets of local states

finite set of transitions $\overline{p} \xrightarrow{\alpha \mid \tau} \overline{q}$
with $\overline{p}, \overline{q} \in \prod_{i \in P_\alpha} Q_i$ and $P_\tau \subseteq P_\alpha$

Example



Cooperating Multi-Pushdown Systems (CPDS)

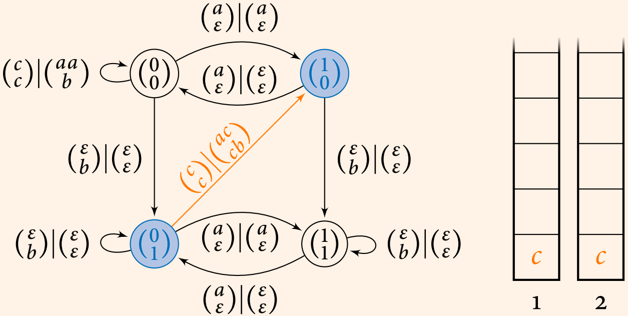
$$\mathfrak{P} = \Big(\prod_{i \in P} Q_i \,,\, \Delta \Big)$$

global states

finite sets of local states

finite set of transitions $\overline{p} \xrightarrow{\alpha \,|\, \tau} \overline{q}$ with $\overline{p}, \overline{q} \in \prod_{i \in P_\alpha} Q_i$ and $P_\tau \subseteq P_\alpha$

Example



Cooperating Multi-Pushdown Systems (CPDS)

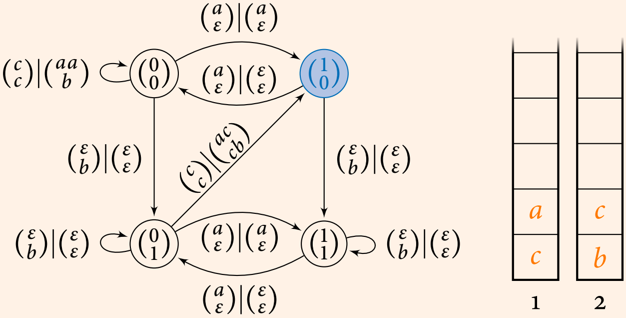
$$\mathfrak{P} = \Big(\prod_{i \in P} Q_i \, , \, \Delta \Big)$$

global states

finite sets of local states

finite set of transitions $\overline{p} \xrightarrow{\alpha \mid \tau} \overline{q}$
with $\overline{p}, \overline{q} \in \prod_{i \in P_\alpha} Q_i$ and $P_\tau \subseteq P_\alpha$

Example



Cooperating Multi-Pushdown Systems (CPDS)

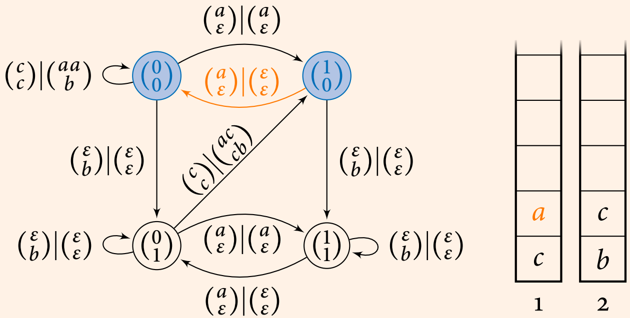
$$\mathfrak{P} = \Big(\prod_{i \in P} Q_i \,,\, \Delta \Big)$$

global states

finite sets of local states

finite set of transitions $\overline{p} \xrightarrow{\alpha \mid \tau} \overline{q}$ with $\overline{p}, \overline{q} \in \prod_{i \in P_\alpha} Q_i$ and $P_\tau \subseteq P_\alpha$

Example



Cooperating Multi-Pushdown Systems (CPDS)

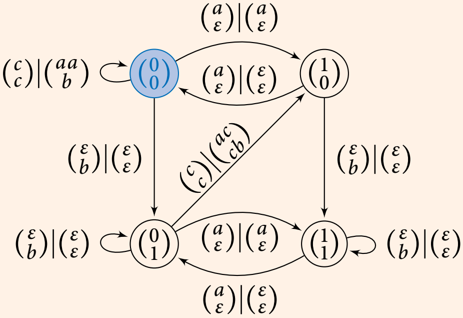
$$\mathfrak{P} = \Big(\prod_{i \in P} Q_i \,,\, \Delta \Big)$$

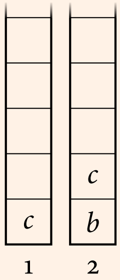
global states

finite sets of local states

finite set of transitions $\overline{p} \xrightarrow{\alpha \mid \tau} \overline{q}$
with $\overline{p}, \overline{q} \in \prod_{i \in P_\alpha} Q_i$ and $P_\tau \subseteq P_\alpha$

Example





Cooperating Multi-Pushdown Systems (CPDS)

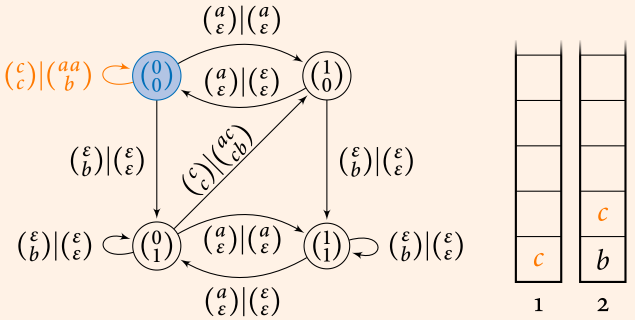
$$\mathfrak{P} = \left(\prod_{i \in P} Q_i \,,\, \Delta \right)$$

global states

finite sets of local states

finite set of transitions $\overline{p} \xrightarrow{\alpha \mid \tau} \overline{q}$
with $\overline{p}, \overline{q} \in \prod_{i \in P_\alpha} Q_i$ and $P_\tau \subseteq P_\alpha$

Example



Cooperating Multi-Pushdown Systems (CPDS)

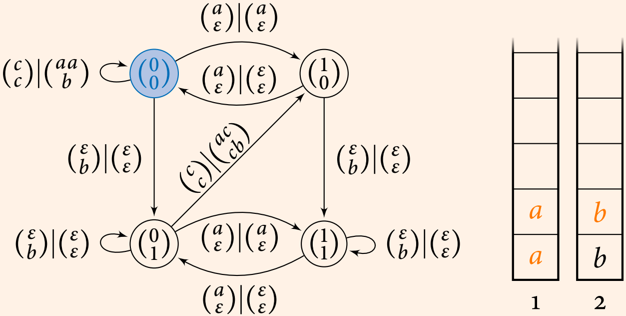
$$\mathfrak{P} = \Big(\prod_{i \in P} Q_i \,,\, \Delta \Big)$$

global states

finite sets of local states

finite set of transitions $\overline{p} \xrightarrow{\alpha \mid \tau} \overline{q}$ with $\overline{p}, \overline{q} \in \prod_{i \in P_\alpha} Q_i$ and $P_\tau \subseteq P_\alpha$

Example



Configurations

- A configuration of $\mathfrak{P}$ is a tuple from $\prod_{i \in P} Q_i \times \mathbb{M}(\mathcal{D})$.
- Let C be a set of configurations of $\mathfrak{P}$.

- $\mathrm{pre}^*_{\mathfrak{P}}(C) := \{d \mid \exists c \in C\colon d \to^*_{\mathfrak{P}} c\}$
- $\mathrm{post}^*_{\mathfrak{P}}(C) := \{d \mid \exists c \in C\colon c \to^*_{\mathfrak{P}} d\}$

- C is recognizable iff for each $\overline{q} \in \prod_{i \in P} Q_i$ the language $C_{\overline{q}} = \{\tau \in \mathbb{M}(\mathcal{D}) \mid (\overline{q}, \tau) \in C\}$ is recognizable.
  - ⇝ accepted by an asynchronous automaton.
- C is rational iff for each $\overline{q} \in \prod_{i \in P} Q_i$ the language $C_{\overline{q}}$ is rational.
  - ⇝ constructed from finite sets using $\cup$, $\cdot$, and *.

Lemma

$$C \textit{ is recognizable } \underset{\not\Longleftarrow}{\Longrightarrow} C \textit{ is rational.}$$

Backwards Reachability

Theorem

Let $\mathfrak{P}$ be a CPDS and C be a recognizable set of configurations of $\mathfrak{P}$. Then $\mathrm{pre}^*_{\mathfrak{P}}(C)$ is effectively recognizable (in polynomial time).

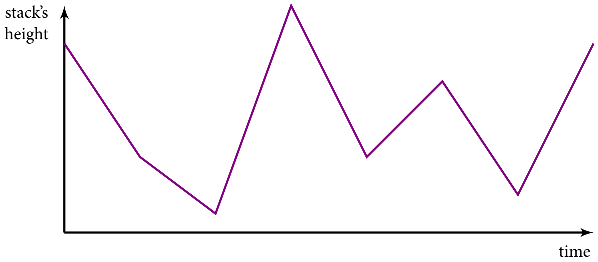
Proof idea: The construction adapts ideas by Bouajjani, Maler, and Esparza (CONCUR 1997) from NFAs to asynchronous automata. □

Forwards Reachability

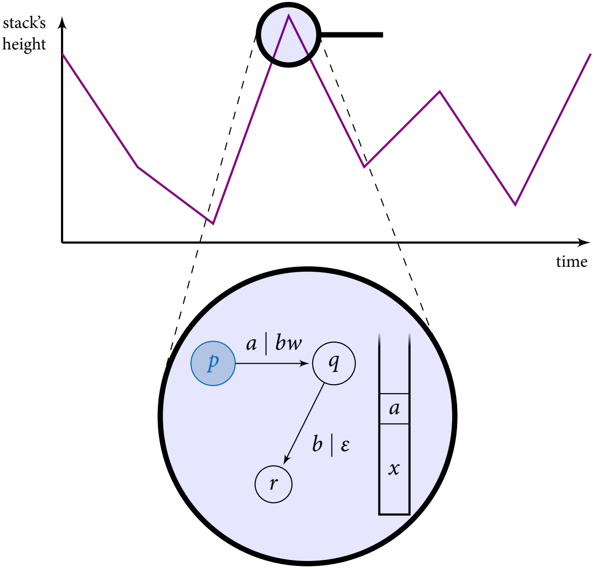
Theorem

Let $\mathfrak{P}$ be a CPDS and C be a rational set of configurations of $\mathfrak{P}$. Then $\mathrm{post}^*_{\mathfrak{P}}(C)$ is effectively rational. If the underlying distributed alphabet $\mathcal{D}$ is fixed, our construction is possible in polynomial time.

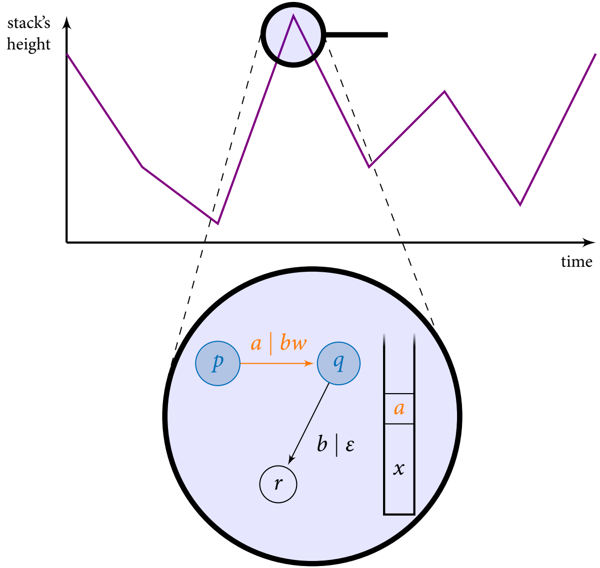
Proof Idea: The One Stack Case [Finkel et al. 1997]



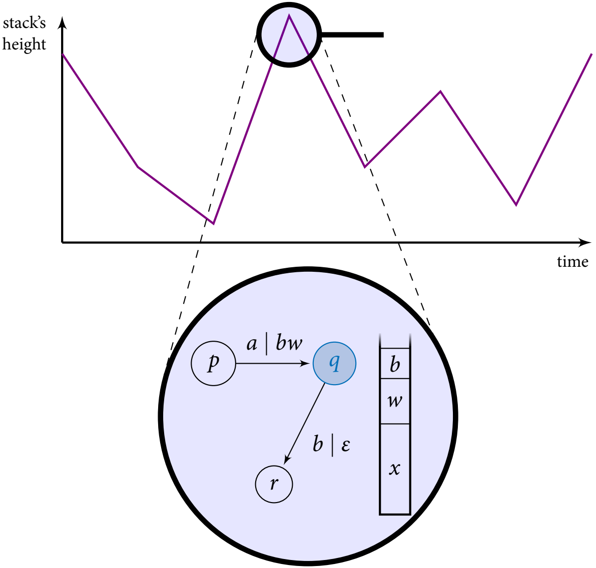
Proof Idea: The One Stack Case [Finkel et al. 1997]



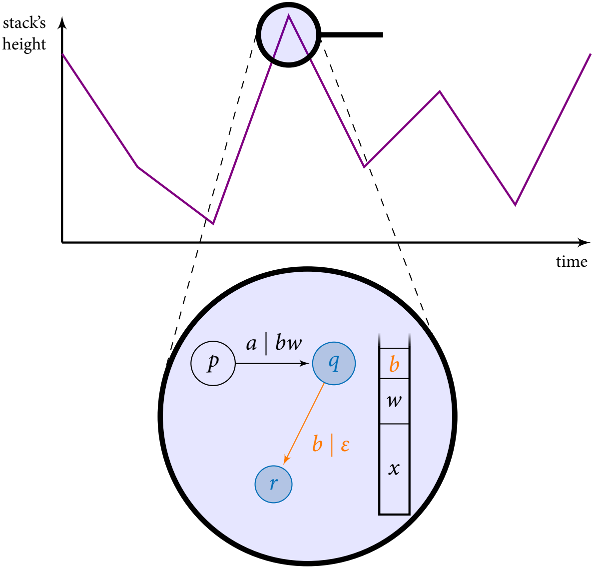
Proof Idea: The One Stack Case [Finkel et al. 1997]



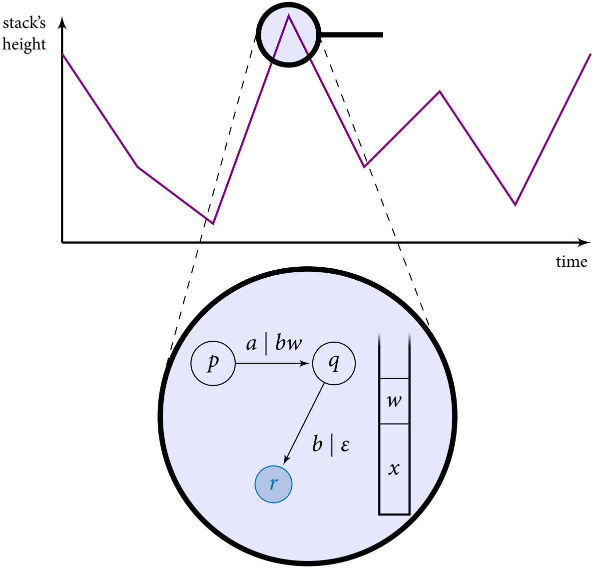
Proof Idea: The One Stack Case [Finkel et al. 1997]



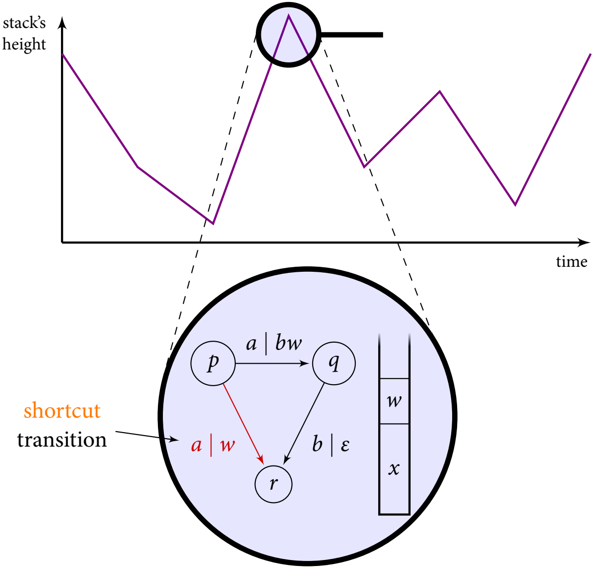
Proof Idea: The One Stack Case [Finkel et al. 1997]



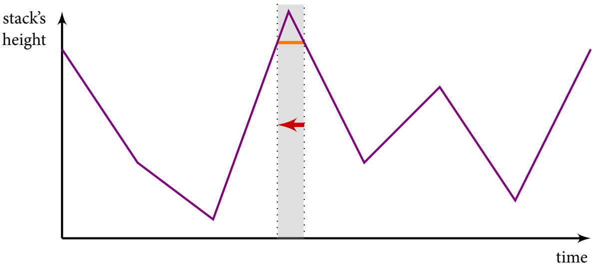
Proof Idea: The One Stack Case [Finkel et al. 1997]



Proof Idea: The One Stack Case [Finkel et al. 1997]

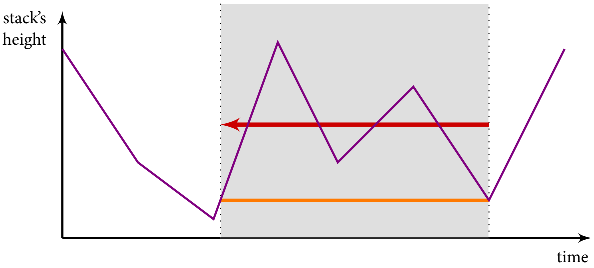


Proof Idea: The One Stack Case [Finkel et al. 1997]



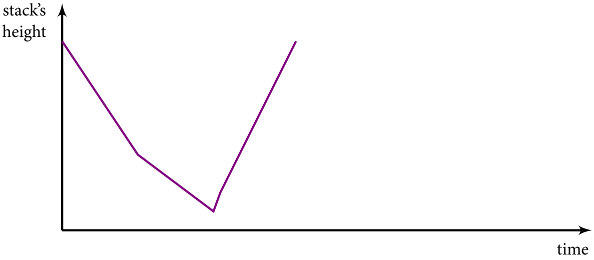
Proof Idea: The One Stack Case [Finkel et al. 1997]

Proof Idea: The One Stack Case [Finkel et al. 1997]



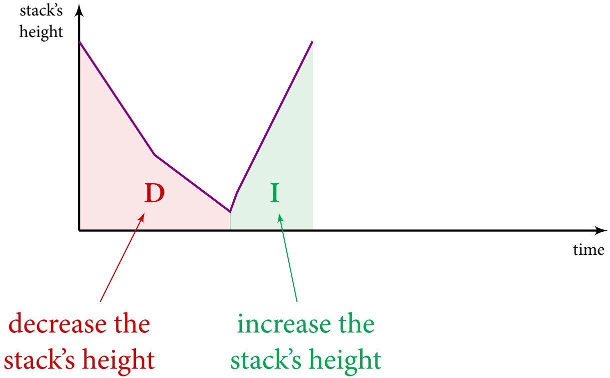
- PDS is saturated if we cannot add more shortcut transitions.

Proof Idea: The One Stack Case [Finkel et al. 1997]



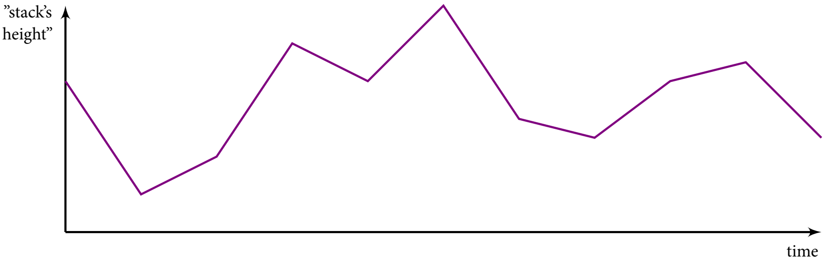
- PDS is saturated if we cannot add more shortcut transitions.

Proof Idea: The One Stack Case [Finkel et al. 1997]



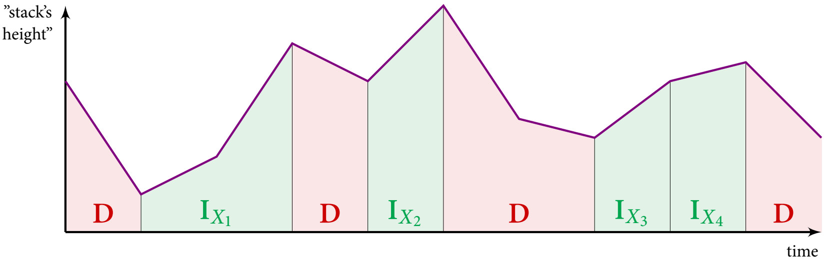
- PDS is saturated if we cannot add more shortcut transitions.
- Computing the effect of a decrease / increase phase in saturated PDS is easy!

Proof Idea: The General Case



1. Saturate the CPDS $\mathfrak{P}$.
2. Decompose $\mathfrak{P}$ into homogeneous CPDS.

Proof Idea: The General Case

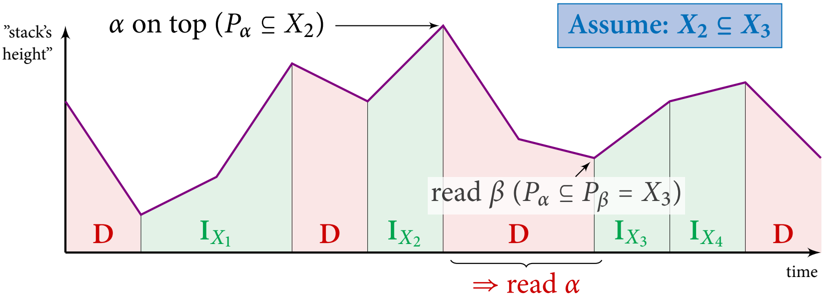


1. Saturate the CPDS $\mathfrak{P}$.
2. Decompose $\mathfrak{P}$ into homogeneous CPDS.
   - each transition in such system
     - only reads letters
     - reads from the processes $X \subseteq P$, writes at least one letter

$\rightsquigarrow$ Computation of post* is “easy” in such CPDS

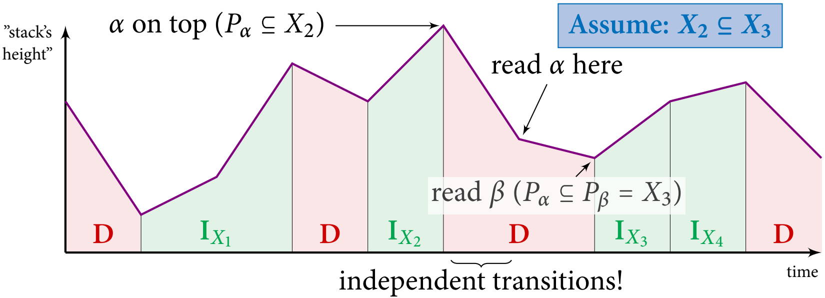
- Note: There are shortest runs with more than two phases!

Proof Idea: The General Case



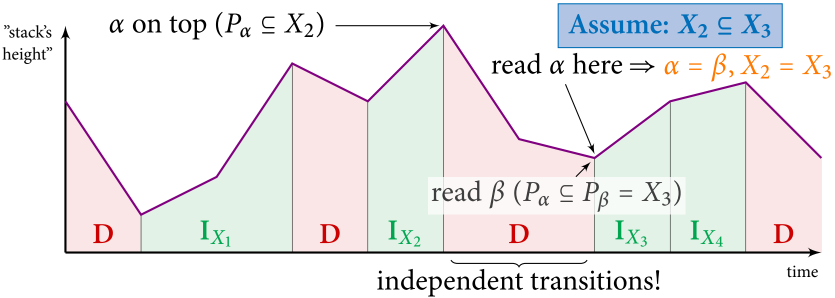
1. Saturate the CPDS $\mathfrak{P}$.
2. Decompose $\mathfrak{P}$ into homogeneous CPDS.
3. Reduce the number of phases of our run.
   - Use shortcuts and/or transpose transitions

Proof Idea: The General Case



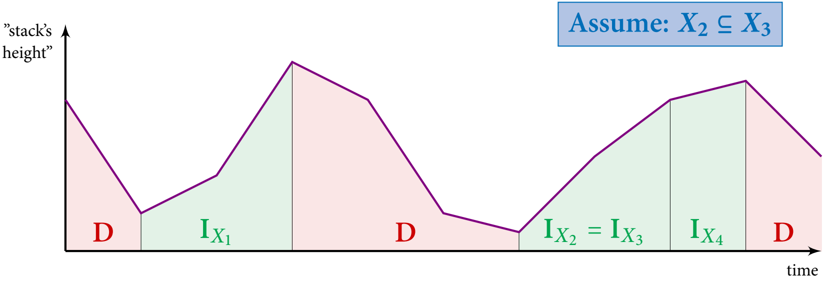
1. Saturate the CPDS $\mathfrak{P}$.
2. Decompose $\mathfrak{P}$ into homogeneous CPDS.
3. Reduce the number of phases of our run.
   - Use shortcuts and/or transpose transitions

Proof Idea: The General Case



1. Saturate the CPDS $\mathfrak{P}$.
2. Decompose $\mathfrak{P}$ into homogeneous CPDS.
3. Reduce the number of phases of our run.
   - Use shortcuts and/or transpose transitions

Proof Idea: The General Case



1. Saturate the CPDS $\mathfrak{P}$.
2. Decompose $\mathfrak{P}$ into homogeneous CPDS.
3. Reduce the number of phases of our run.
   - Use shortcuts and/or transpose transitions

   $\Rightarrow$ Number of phases can be bounded by $O(|A|)$ resp. $O(2^{|P|})$ $\square$

Conclusion

C is …	recognizable	rational
$\mathrm{pre}^*_{\mathfrak{P}}(C)$ is …	recognizable	recursively enumerable
$\mathrm{post}^*_{\mathfrak{P}}(C)$ is …	rational	rational

Thank you!