FO Model Checking on Nested Pushdown Trees

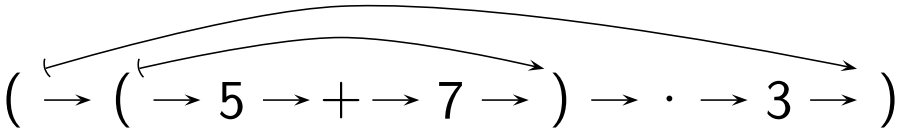
and more: on Collapsible Pushdown Graphs

TECHNISCHE UNIVERSITÄT DARMSTADT

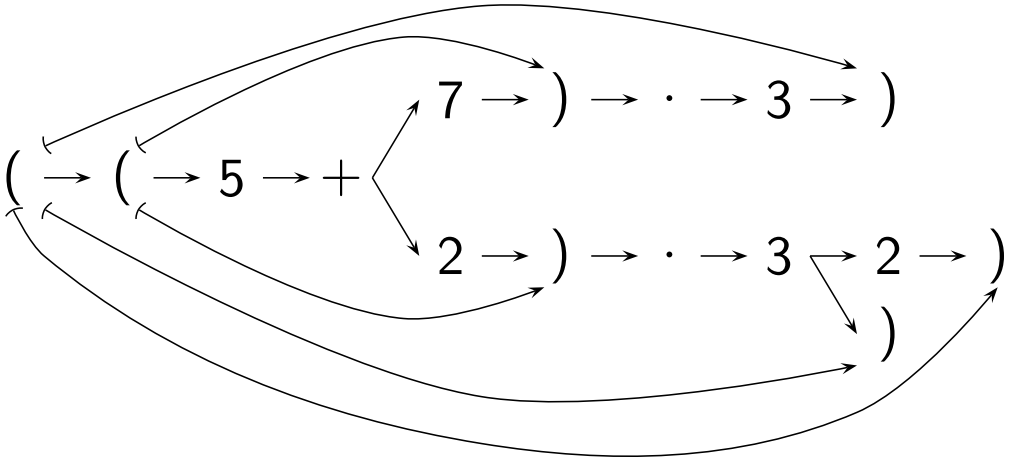
Nested Trees

TECHNISCHE UNIVERSITÄT DARMSTADT

- Nested words: word structures + jump-relation



- Nice theory of finite automata on nested words
- Nested trees: every path is a nested word



Nested Pushdown trees (NPT)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Nested pushdown tree $NPT(P)$: $(V, \rightarrow, (P_q)_{q \in Q}, \hookrightarrow)$

$(V, \rightarrow)$: tree of runs of P

P_q: last state of the run

$\hookrightarrow$: binary relation between corresponding push and pop

Theorem (Alur, Chaudhuri, Madhusudan)

modal μ-calculus model checking is decidable
monadic second-order model checking is undecidable

NPT example

TECHNISCHE UNIVERSITÄT DARMSTADT

Transitions:

$(q, \bot) \to (q, \mathrm{push}_a) \quad (q, a) \to (q, \mathrm{push}_a) \quad (q, a) \to (p, \mathrm{id}) \quad (p, a) \to (p, \mathrm{pop})$

finitely branching, inf. tree

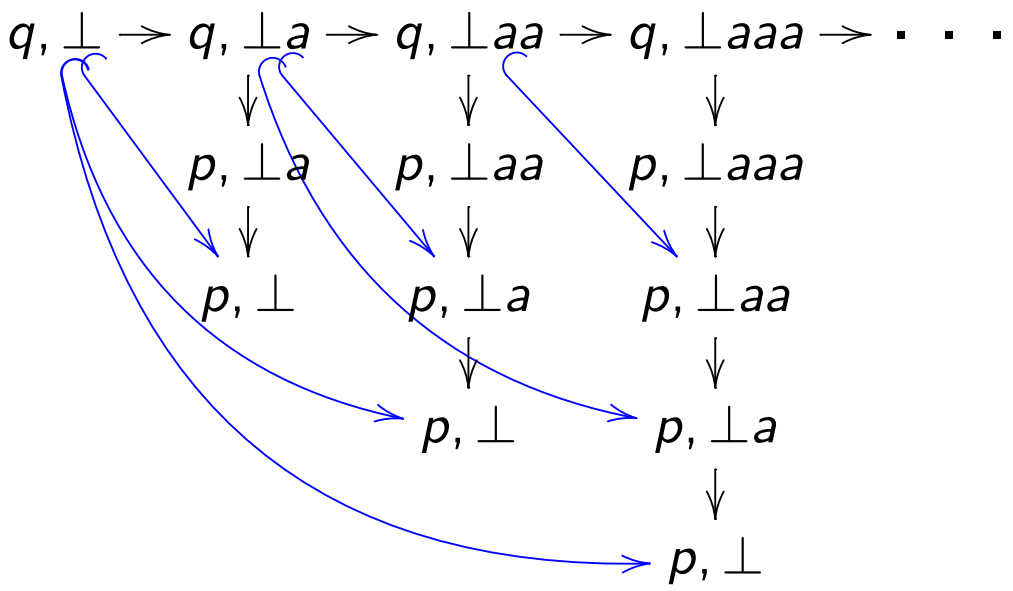
$$
\begin{tikzcd}
q, \bot \arrow[r] & q, \bot a \arrow[r] \arrow[d] & q, \bot aa \arrow[r] \arrow[d] & q, \bot aaa \arrow[r] \arrow[d] & \cdots \\
& p, \bot a \arrow[d] & p, \bot aa \arrow[d] & p, \bot aaa \arrow[d] & \\
& p, \bot & p, \bot a \arrow[d] & p, \bot aa \arrow[d] & \\
& & p, \bot & p, \bot a \arrow[d] & \\
& & & p, \bot &
\end{tikzcd}
$$

NPT example

TECHNISCHE UNIVERSITÄT DARMSTADT

Transitions:

$(q, \bot) \to (q, \mathrm{push}_a) \quad (q, a) \to (q, \mathrm{push}_a) \quad (q, a) \to (p, \mathrm{id}) \quad (p, a) \to (p, \mathrm{pop})$



finitely branching, inf. tree

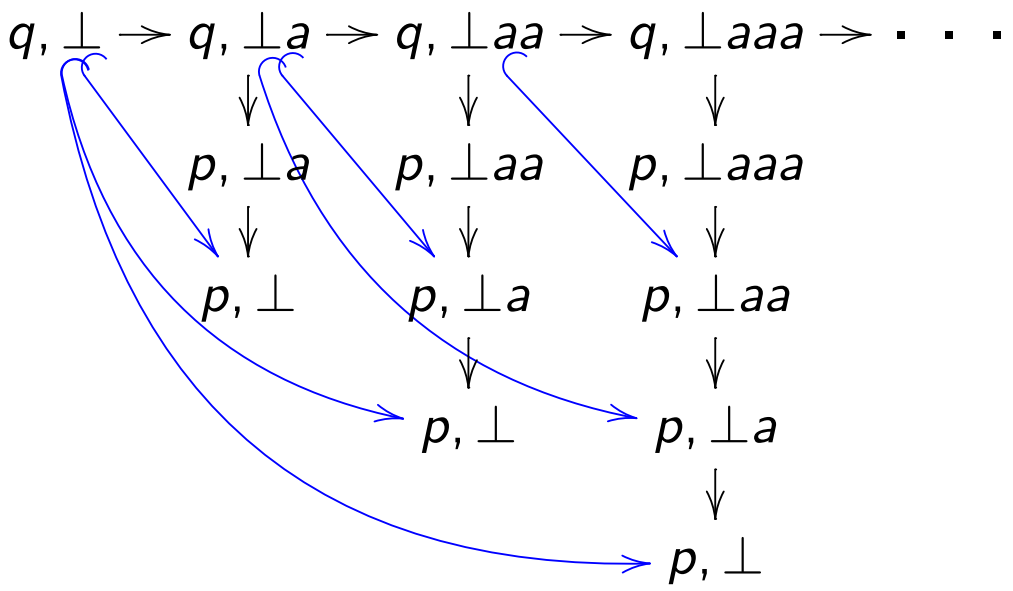
jump edges: inf. outdegree

NPT example

TECHNISCHE UNIVERSITÄT DARMSTADT

Transitions:

$(q, \bot) \to (q, \mathrm{push}_a) \quad (q, a) \to (q, \mathrm{push}_a) \quad (q, a) \to (p, \mathrm{id}) \quad (p, a) \to (p, \mathrm{pop})$



finitely branching, inf. tree
jump edges: inf. outdegree

Interpretation of grid $\Rightarrow$ undecidable for MSO

FO Model Checking on Nested Pushdown Trees

TECHNISCHE UNIVERSITÄT DARMSTADT

Theorem

FO-model checking on NPT is decidable

Proof idea:

- Simulation: NPT $\longrightarrow$ Collapsible Pushdown Graphs (CPG)
- FO-interpretation: $\varphi \in FO_m \longrightarrow \varphi' \in FO_{m+3}$

FO-model checking on CPG: via tree-automaticity

Collapsible Pushdown Systems (of order 2)

TECHNISCHE UNIVERSITÄT DARMSTADT

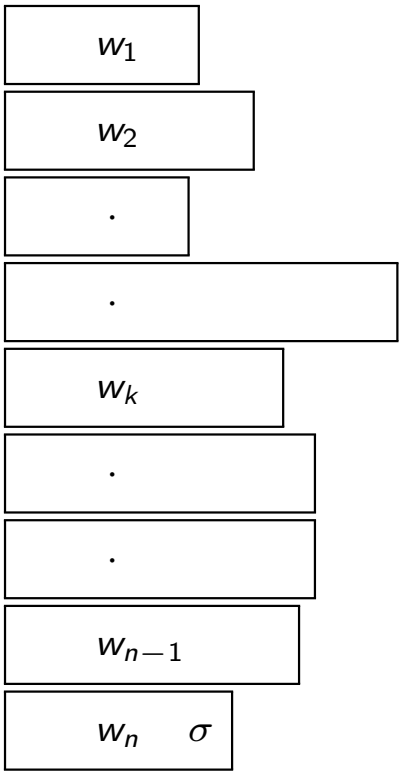
- Higher-order pushdown system: pushdown system with a stack of stacks
- Collapsible Pushdown System(CPS):
  higher-order pushdown systems + collapse-operation.

Theorem (Hague, Murawski, Ong, Serre)

modal μ-calculus model checking: decidable
MSO model checking: undecidable

stack of stacks and stack-operations

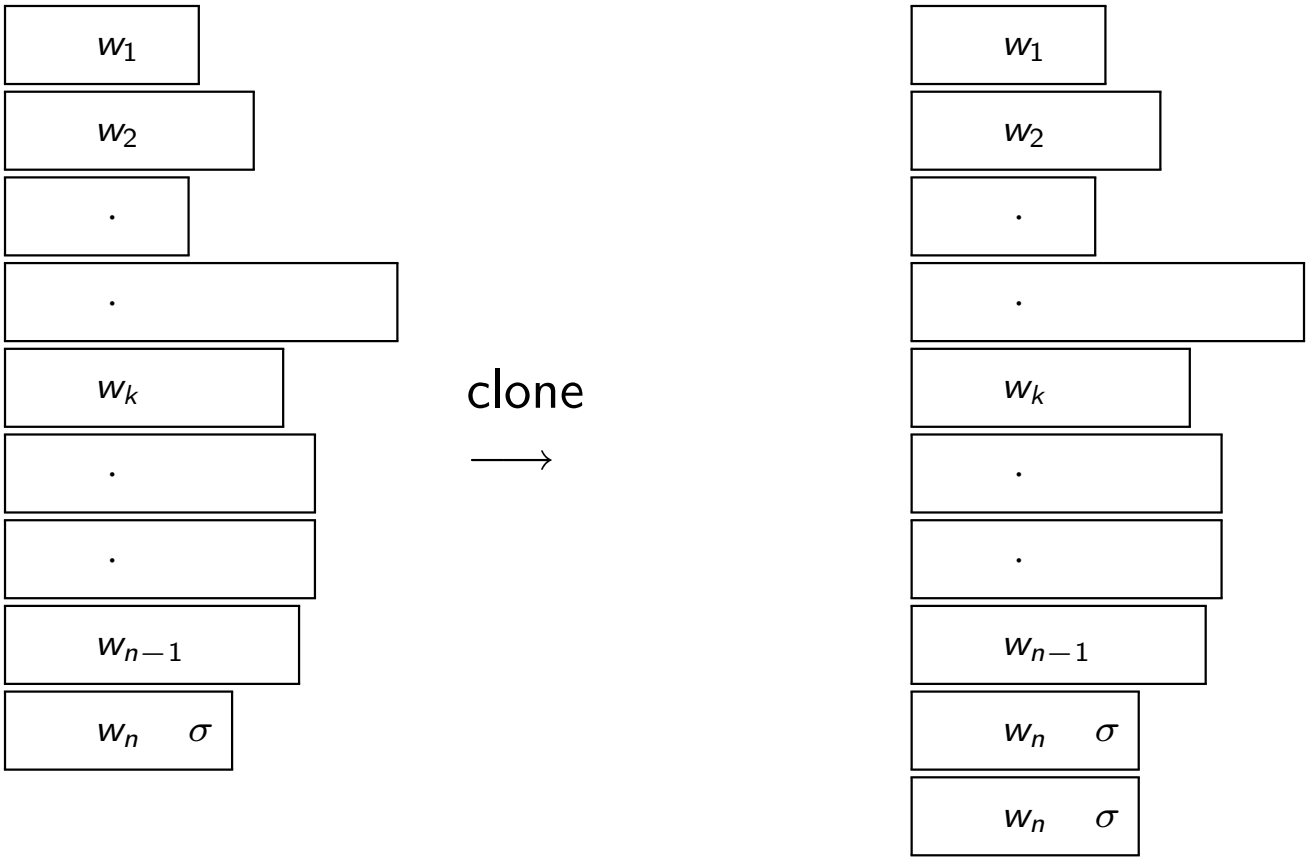
A level 2 stack is a list of stacks.



stack of stacks and stack-operations

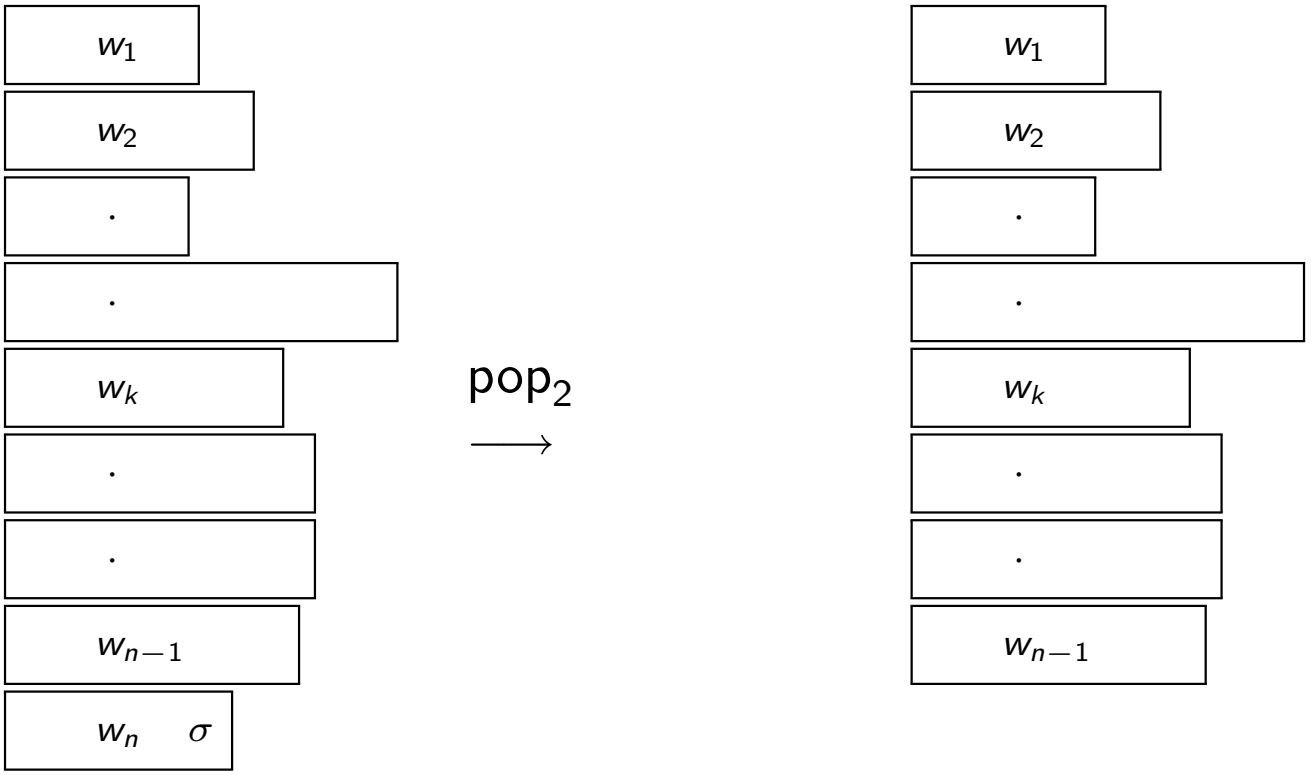
TECHNISCHE UNIVERSITÄT DARMSTADT

A level 2 stack is a list of stacks.



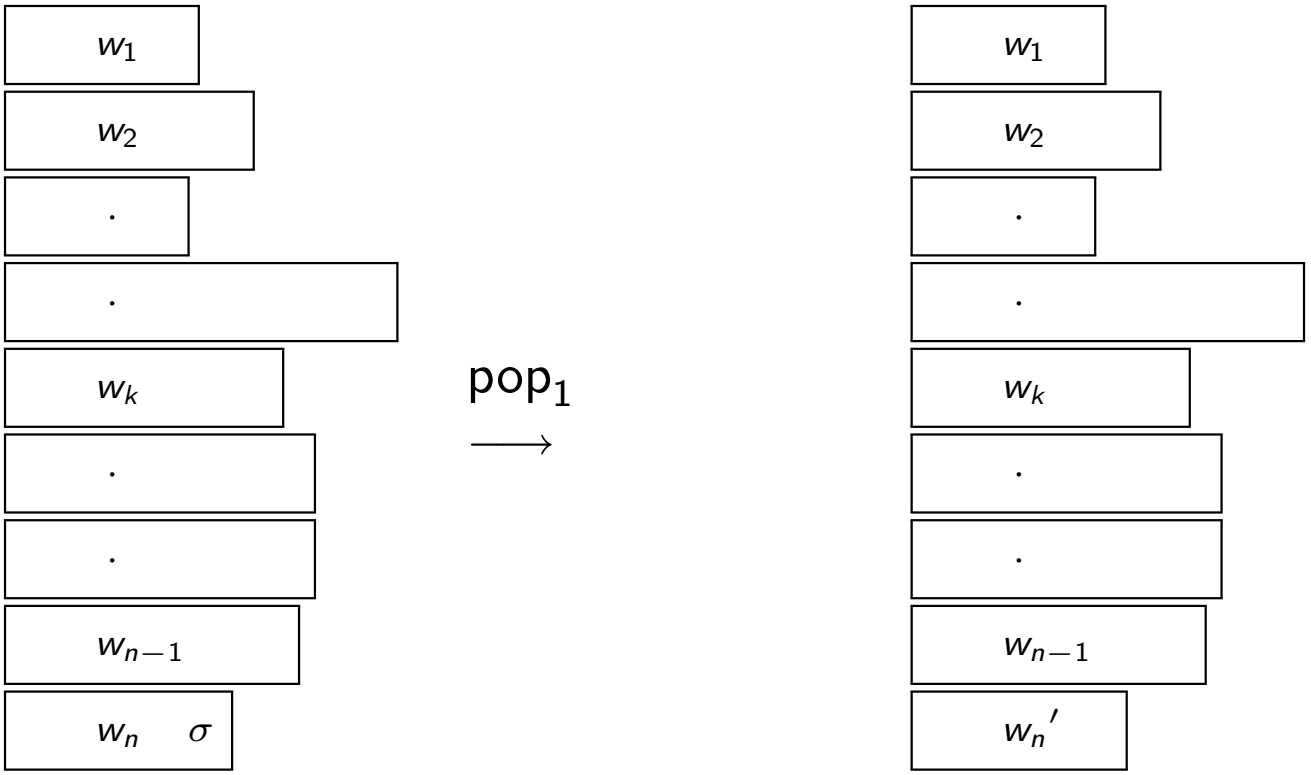
stack of stacks and stack-operations

A level 2 stack is a list of stacks.



stack of stacks and stack-operations

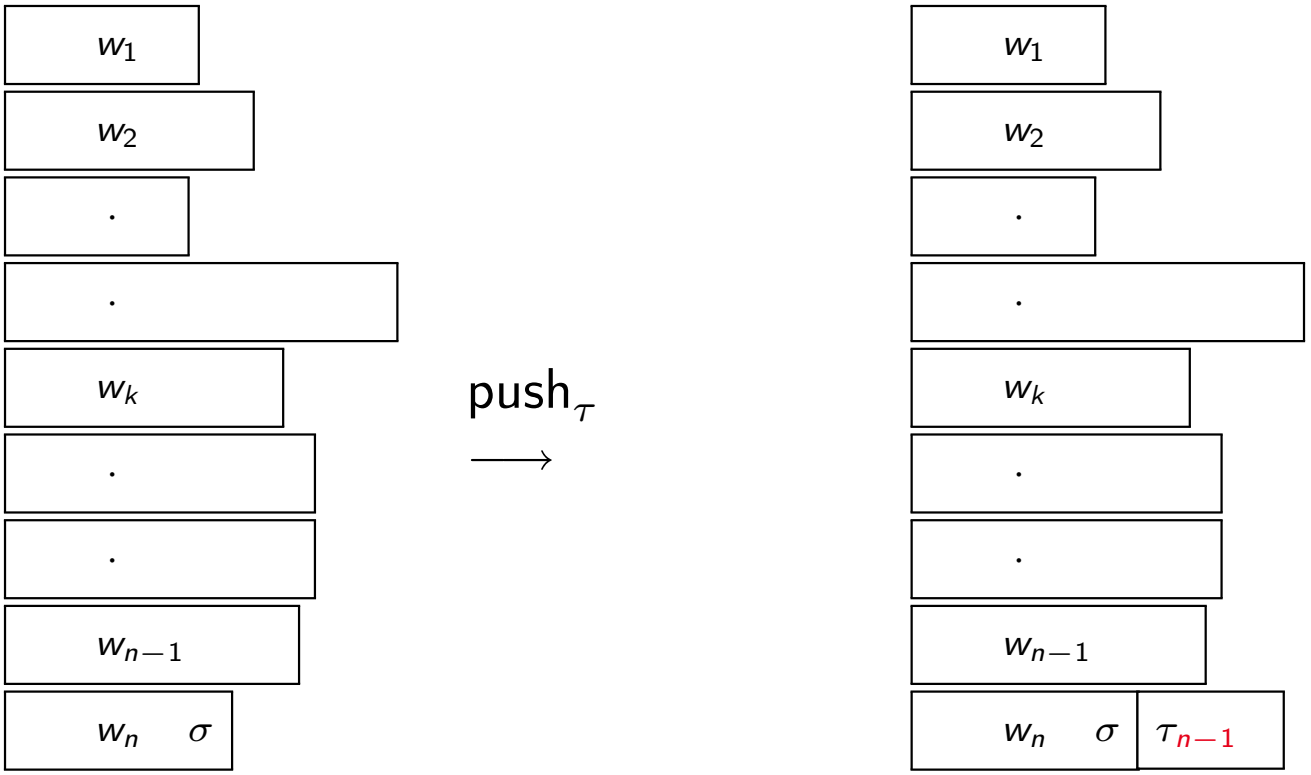
A level 2 stack is a list of stacks.



stack of stacks and stack-operations

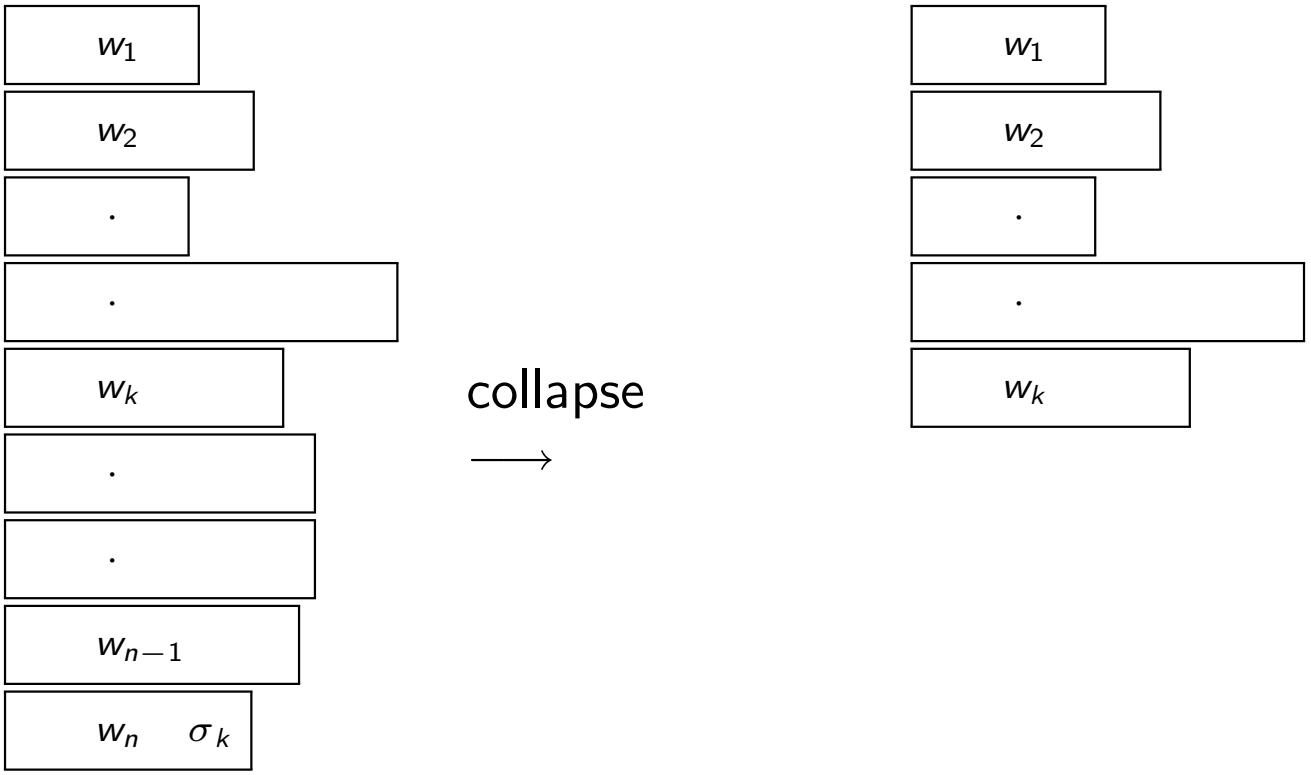
TECHNISCHE UNIVERSITÄT DARMSTADT

A level 2 stack is a list of stacks.



stack of stacks and stack-operations

A level 2 stack is a list of stacks.



Collapsible Pushdown Graphs (CPG)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Collapsible pushdown system with stack of stacks
- Operations: $\text{push}_\sigma, \text{clone}, \text{pop}_1, \text{pop}_2, \text{collapse}$
- Configurations: (q, s) – q a state, s a stack
  Edges: $(q, s) \xrightarrow{op} (q', s')$
  for transitions $(q, \text{top}_1(s)) \rightarrow (q', op)$ with $op(s) = s'$
- CPG: Graph of *reachable* configurations with labeled transitions

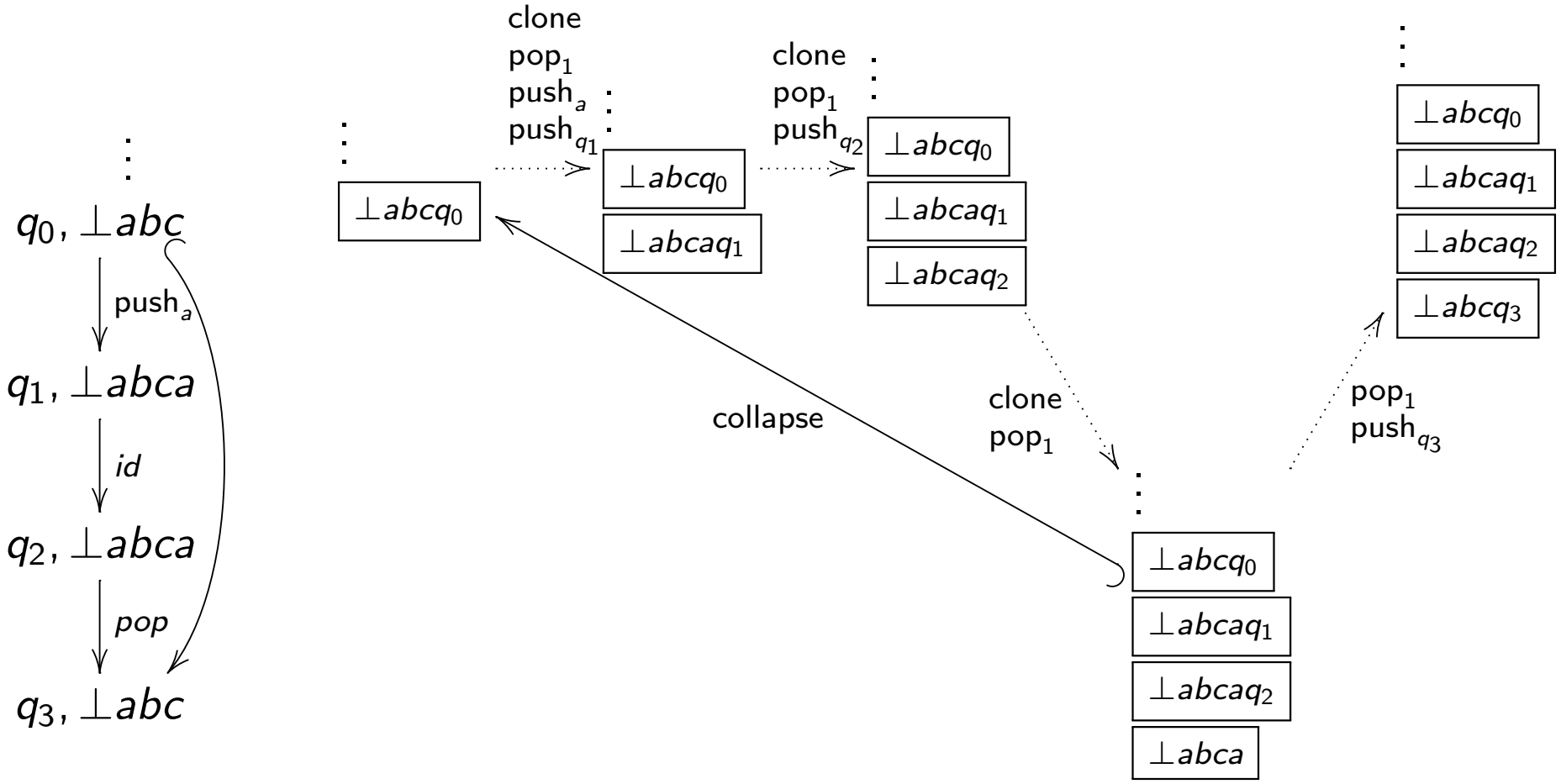
Vertices in NPT: runs of pushdown system
Vertices in CPG: configurations of higher-order pushdown system

Simulation of NPT with CPS

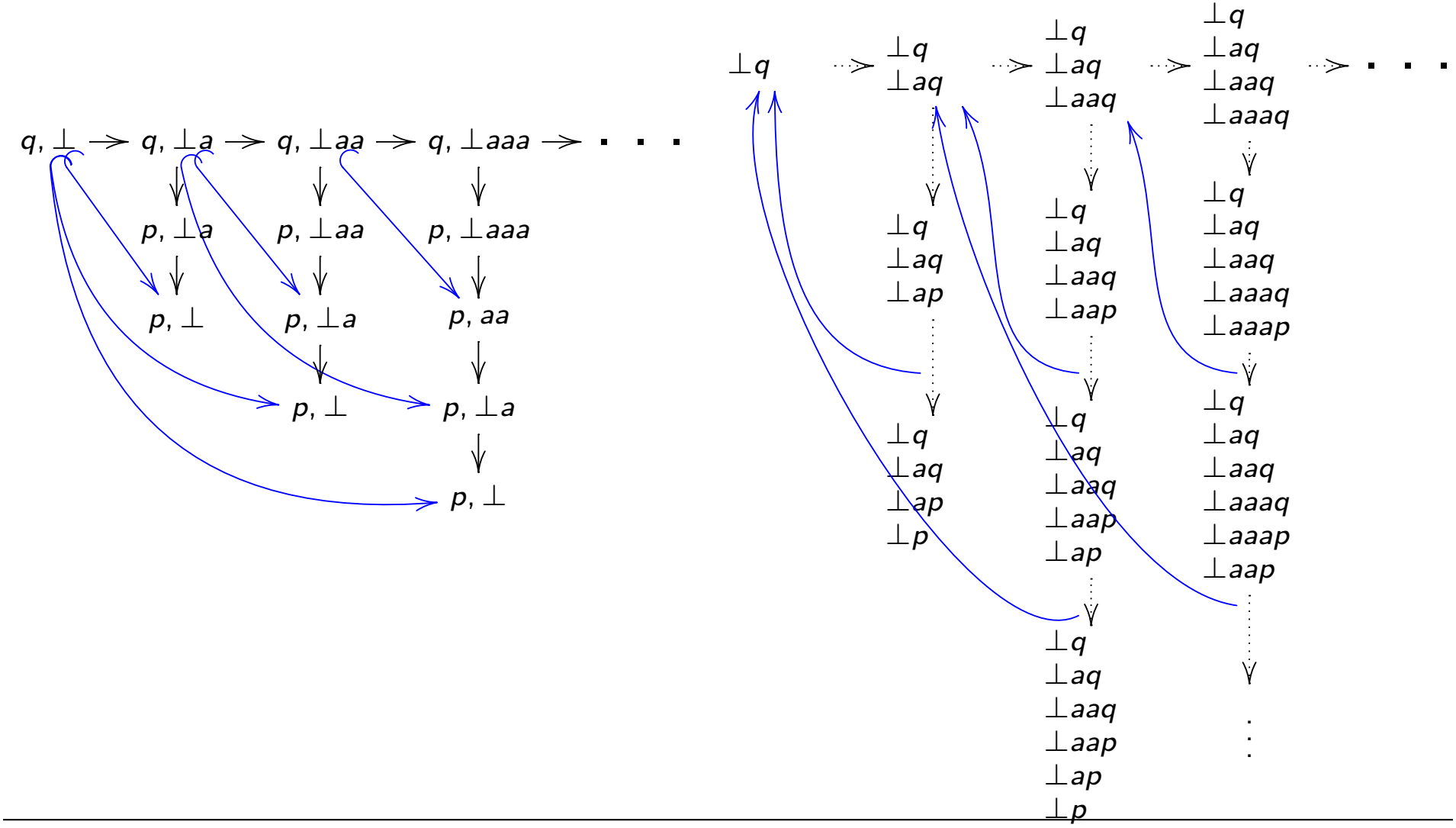
TECHNISCHE UNIVERSITÄT DARMSTADT

NPT	CPS
run of a pushdown system	list of stacks + state
jump relation $\hookrightarrow$	collapse
1 stack operation (level 1)	up to 4 stack operations (level 2)

Simulation of NPT in CPG



Example of Simulation



Tree-automaticity

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition

A tree-automaton is a tuple $A := (\Sigma, Q, \Delta, q_I, q_0)$ with
$|\Sigma|, |Q| < \infty$, $q_I \subseteq Q$, $q_0 \in Q$ and $\Delta \subseteq Q \times \Sigma \times Q \times Q$.
A run of A on T: function $r : \text{domain}(T) \to Q$ respecting Δ
r accepting: all leaves labelled by q_I and the root by q_0

Definition

A structure $\mathbb{S} := (S, E_1, E_2, \dots, E_n)$ is tree-automatic iff there are tree-automata $A_S, A_{E_1}, \dots, A_{E_n}$ and a bijection $f : S \to L(A_S)$ such that
$(s_1, s_2) \in E_i$ iff A_{E_i} accepts $f(s_1) \otimes f(s_2)$

Theorem

FO model checking on every tree-automatic structure is decidable.

CPG are tree-automatic

TECHNISCHE UNIVERSITÄT DARMSTADT

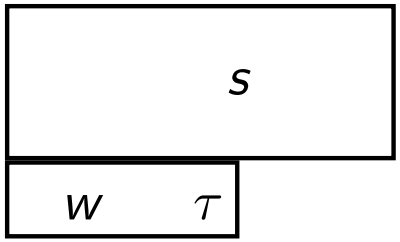
Theorem

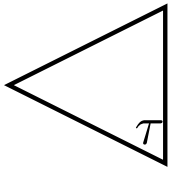
CPG are tree-automatic.

Proof by

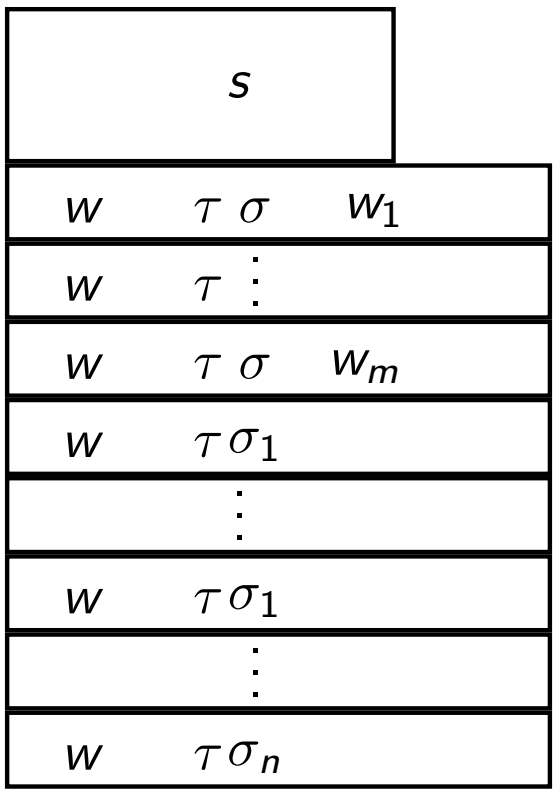
- Encoding stacks $\longrightarrow$ trees
- stack-operations $\longrightarrow$ easy tree-operations.
- valid configurations $\longrightarrow$ accepted trees.

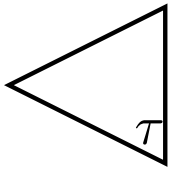
Coding Stacks in Trees



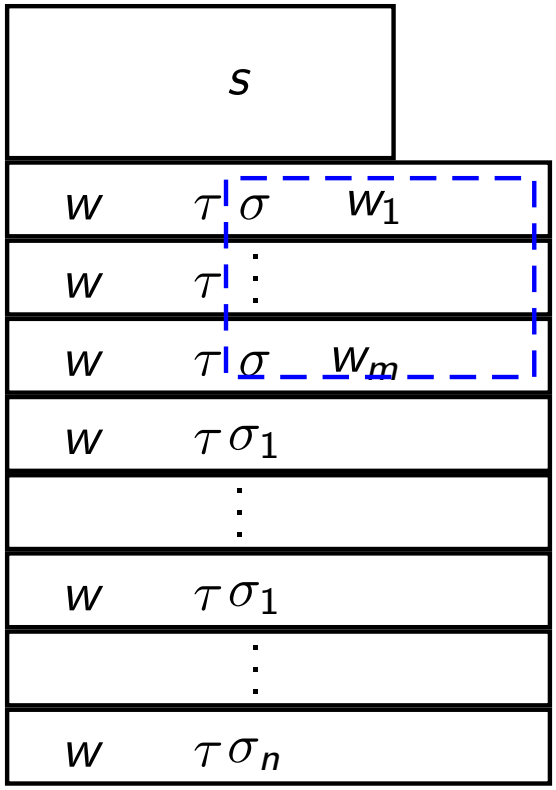


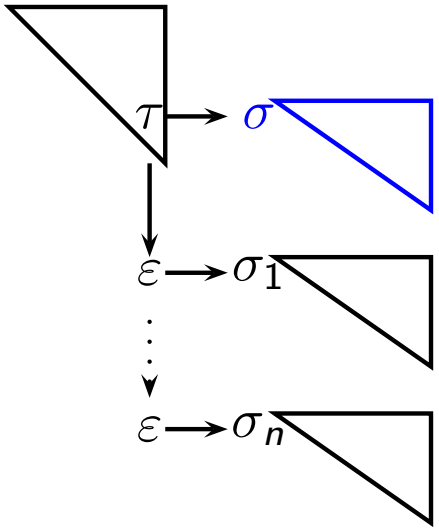
Coding Stacks in Trees





Coding Stacks in Trees





Encoding Example

TECHNISCHE UNIVERSITÄT DARMSTADT

$$
\begin{array}{lllll}
\bot & b & c & & \\
\bot & b & d & & \\
\bot & b & d & e & f \\
\bot & b & d & & \\
\bot & b & & &
\end{array}
$$

$$
\begin{tikzcd}
\bot \arrow[r] & b \arrow[r] \arrow[d] & c & & \\
& \varepsilon \arrow[r] \arrow[dd] & d \arrow[d] & & \\
& & \varepsilon \arrow[r] \arrow[d] & e \arrow[r] & f \\
& & \varepsilon & & \\
& \varepsilon & & &
\end{tikzcd}
$$

Encoding Example

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\begin{array}{lllll}
\bot & b & c & & \\
\bot & b & d & & \\
\bot & b & d & e & f \\
\bot & b & d & & \\
\bot & b & & &
\end{array}$$

$$\begin{tikzcd}
\bot \arrow[r] & b \arrow[r] \arrow[d] & c & & \\
& \varepsilon \arrow[r] \arrow[dd] & d \arrow[d] & & \\
& & \varepsilon \arrow[r] \arrow[d] & e \arrow[r] & f \\
& & \varepsilon & & \\
& \varepsilon & & &
\end{tikzcd}$$

push_g

$$\begin{array}{lllll}
\bot & b & c & & \\
\bot & b & d & & \\
\bot & b & d & e & f \\
\bot & b & d & & \\
\bot & b & g & &
\end{array}$$

$$\begin{tikzcd}
\bot \arrow[r] & b \arrow[r] \arrow[d] & c & & \\
& \varepsilon \arrow[r] \arrow[dd] & d \arrow[d] & & \\
& & \varepsilon \arrow[r] \arrow[d] & e \arrow[r] & f \\
& & \varepsilon & & \\
& \varepsilon \arrow[r] & g & &
\end{tikzcd}$$

Encoding Example

TECHNISCHE UNIVERSITÄT DARMSTADT

$$
\begin{array}{lllll}
\bot & b & c & & \\
\bot & b & d & & \\
\bot & b & d & e & f \\
\bot & b & d & & \\
\bot & b & g & &
\end{array}
$$

$$
\begin{tikzcd}
\bot \arrow[r] & b \arrow[r] \arrow[d] & c & & \\
& \varepsilon \arrow[r] \arrow[dd] & d \arrow[d] & & \\
& & \varepsilon \arrow[r] \arrow[d] & e \arrow[r] & f \\
& & \varepsilon & & \\
& \varepsilon \arrow[r] & g & &
\end{tikzcd}
$$

pop_1

$$
\begin{array}{lllll}
\bot & b & c & & \\
\bot & b & d & & \\
\bot & b & d & e & f \\
\bot & b & d & & \\
\bot & b & & &
\end{array}
$$

$$
\begin{tikzcd}
\bot \arrow[r] & b \arrow[r] \arrow[d] & c & & \\
& \varepsilon \arrow[r] \arrow[dd] & d \arrow[d] & & \\
& & \varepsilon \arrow[r] \arrow[d] & e \arrow[r] & f \\
& & \varepsilon & & \\
& \varepsilon & & &
\end{tikzcd}
$$

Encoding Example

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\begin{array}{lllll}
\bot & b & c & & \\
\bot & b & d & & \\
\bot & b & d & e & f \\
\bot & b & d & & \\
\bot & b & & &
\end{array}$$

$$\begin{tikzcd}
\bot \arrow[r] & b \arrow[r] \arrow[d] & c & & \\
 & \varepsilon \arrow[r] \arrow[dd] & d \arrow[d] & & \\
 & & \varepsilon \arrow[r] \arrow[d] & e \arrow[r] & f \\
 & & \varepsilon & & \\
 & \varepsilon & & &
\end{tikzcd}$$

pop_1

$$\begin{array}{lllll}
\bot & b & c & & \\
\bot & b & d & & \\
\bot & b & d & e & f \\
\bot & b & d & & \\
\bot & & & &
\end{array}$$

$$\begin{tikzcd}
\bot \arrow[r] \arrow[dd] & b \arrow[r] \arrow[d] & c & & \\
 & \varepsilon \arrow[r] & d \arrow[d] & & \\
 & & \varepsilon \arrow[r] \arrow[d] & e \arrow[r] & f \\
 & & \varepsilon & & \\
\varepsilon & & & &
\end{tikzcd}$$

Encoding Example

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\begin{array}{ccccc} \bot & b & c & & \\ \bot & b & d & & \\ \bot & b & d & e & f \\ \bot & b & d & & \\ \bot & b & & & \end{array}$$

$$\begin{tikzcd} \bot \arrow[r] & b \arrow[r] \arrow[d] & c & & \\ & \varepsilon \arrow[r] \arrow[dd] & d \arrow[d] & & \\ & & \varepsilon \arrow[r] \arrow[d] & e \arrow[r] & f \\ & & \varepsilon & & \\ & \varepsilon & & & \end{tikzcd}$$

pop_2

$$\begin{array}{ccccc} \bot & b & c & & \\ \bot & b & d & & \\ \bot & b & d & e & f \\ \bot & b & d & & \end{array}$$

$$\begin{tikzcd} \bot \arrow[r] & b \arrow[r] \arrow[d] & c & & \\ & \varepsilon \arrow[r] & d \arrow[d] & & \\ & & \varepsilon \arrow[r] \arrow[d] & e \arrow[r] & f \\ & & \varepsilon & & \end{tikzcd}$$

Encoding Example

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\begin{array}{lllll} \bot & b & c & & \\ \bot & b & d & & \\ \bot & b & d & e & f \\ \bot & b & d & & \end{array}$$

$$\begin{tikzcd} \bot \arrow[r] & b \arrow[r] \arrow[d] & c & & \\ & \varepsilon \arrow[r] & d \arrow[d] & & \\ & & \varepsilon \arrow[r] \arrow[d] & e \arrow[r] & f \\ & & \varepsilon & & \end{tikzcd}$$

collapse

$$\begin{array}{lll} \bot & b & c \end{array}$$

$$\bot \to b \to c$$

Detecting Valid Configurations

TECHNISCHE UNIVERSITÄT DARMSTADT

- Nodes in a CPG: *reachable* configurations
- Need: A accepts T if Decode(T) reachable in CPG
- Idea: Label every $t \in T$ with the state at the induced substack in the run to Decode(T).
- Problem: Loops

The Loop Problem

TECHNISCHE UNIVERSITÄT DARMSTADT

Δ: $(\bot, q_0, q_1, \text{clone}), (\bot, q_1, q_2, \text{push}_a), (q_2, a, q_0, \text{push}_a), (q_0, a, q_1, \text{pop}_2)$

Is $(q_1, \bot)$ reachable?

The Loop Problem

TECHNISCHE UNIVERSITÄT DARMSTADT

Δ: $(\bot, q_0, q_1, \text{clone}), (\bot, q_1, q_2, \text{push}_a), (q_2, a, q_0, \text{push}_a), (q_0, a, q_1, \text{pop}_2)$

Is $(q_1, \bot)$ reachable? $q_0, \bot \rightarrow q_1, \bot \longrightarrow q_2, \bot \longrightarrow q_0, \bot \longrightarrow q_1, \bot$

$\bot \qquad \bot a \qquad \bot aa$

The Loop Problem

TECHNISCHE UNIVERSITÄT DARMSTADT

Δ: $(\bot, q_0, q_1, \text{clone}), (\bot, q_1, q_2, \text{push}_a), (q_2, a, q_0, \text{push}_a), (q_0, a, q_1, \text{pop}_2)$

Is $(q_1, \bot)$ reachable? $\quad q_0, \bot \rightarrow q_1, \bot \longrightarrow q_2, \bot \longrightarrow q_0, \bot \longrightarrow q_1, \bot$

$\bot \qquad \bot a \qquad \bot aa$

Problem: A only reads the encoding of $q_1, \bot$!

Task: Determine Loops$(s) := \{(q_1, q_2)$ s. t. $\exists$ loop q_1, s to $q_2, s\}$?

The Loop Problem

TECHNISCHE UNIVERSITÄT DARMSTADT

Δ: $(\bot, q_0, q_1, \text{clone}), (\bot, q_1, q_2, \text{push}_a), (q_2, a, q_0, \text{push}_a), (q_0, a, q_1, \text{pop}_2)$

Is $(q_1, \bot)$ reachable? $\quad q_0, \bot \rightarrow q_1, \bot \longrightarrow q_2, \bot \longrightarrow q_0, \bot \longrightarrow q_1, \bot$

$\bot \qquad \bot a \qquad \bot aa$

Problem: A only reads the encoding of $q_1, \bot$!
Task: Determine $\text{Loops}(s) := \{(q_1, q_2) \text{ s. t. } \exists \text{ loop } q_1, s \text{ to } q_2, s\}$?
Solution:

Lemma (Loop-Lemma)

For a stack s with last word w: $Loops(s)$ are determined by $Loops(pop_1(w))$ and the top-symbol of s.

Build loops of $\bot$ and the Loop-Lemma into the automaton.

Example Valid Configurations

$$q_0, \bot \xrightarrow[\text{push}_a]{\text{Loop+}} \bot a \xrightarrow[\text{clone}]{\text{Loop+}} \begin{matrix} \bot a \\ \bot a \end{matrix} \xrightarrow[\text{push}_b]{\text{Loop+}} \begin{matrix} \bot a \\ \bot ab \end{matrix} \xrightarrow[\text{push}_c]{\text{Loop+}} q_4, \begin{matrix} \bot a \\ \bot abc \end{matrix}$$

$$\bot \qquad \bot \to a \qquad \begin{matrix} \bot \to a \\ \downarrow \\ \varepsilon \end{matrix} \qquad \begin{matrix} \bot \to a \\ \downarrow \\ \varepsilon \to b \end{matrix} \qquad \begin{matrix} \bot \to a \\ \downarrow \\ \varepsilon \to b \to c \end{matrix}$$

Example Valid Configurations

$$q_0, \bot \xrightarrow[\text{push}_a]{\text{Loop+}} \bot a \xrightarrow[\text{clone}]{\text{Loop+}} \begin{array}{c}\bot a\\ \bot a\end{array} \xrightarrow[\text{push}_b]{\text{Loop+}} \begin{array}{c}\bot a\\ \bot ab\end{array} \xrightarrow[\text{push}_c]{\text{Loop+}} q_4, \begin{array}{c}\bot a\\ \bot abc\end{array}$$

$$\begin{array}{ccccc}
 & & & & q_0 \quad q_1 \\
\bot & \bot \to a & \bot \to a & \bot \to a & \bot \to a \\
 & & \downarrow & \downarrow & \downarrow \\
 & & \varepsilon & \varepsilon \to b & \varepsilon \to b \to c \\
 & & & & q_2 \quad q_3 \quad q_4
\end{array}$$

Labelling valid if

Example Valid Configurations

TECHNISCHE UNIVERSITÄT DARMSTADT

$$q_0, \bot \xrightarrow[\text{push}_a]{\text{Loop+}} q_1, \bot a \xrightarrow[\text{clone}]{\text{Loop+}} \begin{array}{c} \bot a \\ \bot a \end{array} \xrightarrow[\text{push}_b]{\text{Loop+}} \begin{array}{c} \bot a \\ \bot ab \end{array} \xrightarrow[\text{push}_c]{\text{Loop+}} q_4, \begin{array}{c} \bot a \\ \bot abc \end{array}$$

$$\begin{array}{ccccc} & & & & q_0 \quad q_1 \\ \bot & \bot \to a & \bot \to a & \bot \to a & \bot \to a \\ & & \downarrow & \downarrow & \downarrow \\ & & \varepsilon & \varepsilon \to b & \varepsilon \to b \to c \\ & & & & q_2 \quad q_3 \quad q_4 \end{array}$$

$$\exists q_0' \ \exists \text{loop}: \quad q_0, \bot \to q_0', \bot \text{ and } (q_0', \bot, q_1, \text{push}_a) \in \Delta$$

Labelling valid if

Example Valid Configurations

TECHNISCHE UNIVERSITÄT DARMSTADT

$$q_0, \bot \xrightarrow[\text{push}_a]{\text{Loop+}} q_1, \bot a \xrightarrow[\text{clone}]{\text{Loop+}} q_2, \begin{matrix}\bot a \\ \bot a\end{matrix} \xrightarrow[\text{push}_b]{\text{Loop+}} \begin{matrix}\bot a \\ \bot ab\end{matrix} \xrightarrow[\text{push}_c]{\text{Loop+}} q_4, \begin{matrix}\bot a \\ \bot abc\end{matrix}$$

q_0 q_1

$$\bot \qquad \bot \to a \qquad \begin{matrix}\bot \to a \\ \downarrow \\ \varepsilon\end{matrix} \qquad \begin{matrix}\bot \to a \\ \downarrow \\ \varepsilon \to b\end{matrix} \qquad \begin{matrix}\bot \to a \\ \downarrow \\ \varepsilon \to b \to c\end{matrix}$$

q_2 q_3 q_4

Labelling valid if

$$\exists q_0' \ \exists \text{loop}: \quad q_0, \bot \to q_0', \bot \text{ and } (q_0', \bot, q_1, \text{push}_a) \in \Delta$$
$$\exists q_1' \ \exists \text{loop}: \quad q_1, \bot a \to q_1', \bot a \text{ and } (q_1', a, q_2, \text{clone}) \in \Delta$$

Example Valid Configurations

TECHNISCHE UNIVERSITÄT DARMSTADT

$$q_0, \bot \xrightarrow[\text{push}_a]{\text{Loop+}} q_1, \bot a \xrightarrow[\text{clone}]{\text{Loop+}} q_2, \bot a \xrightarrow[\text{push}_b]{\text{Loop+}} q_3, \bot a \xrightarrow[\text{push}_c]{\text{Loop+}} q_4, \bot a$$

$$\bot a \qquad \bot ab \qquad \bot abc$$

$$q_0 \qquad q_1$$

$$\bot \qquad \bot \to a \qquad \bot \to a \qquad \bot \to a \qquad \bot \to a$$

$$\downarrow \qquad \downarrow \qquad \downarrow$$

$$\varepsilon \qquad \varepsilon \to b \qquad \varepsilon \to b \to c$$

$$q_2 \qquad q_3 \qquad q_4$$

Labelling valid if

$$\begin{aligned}
&\exists q_0' \ \exists \text{loop}: && q_0, \bot \to q_0', \bot \text{ and } (q_0', \bot, q_1, \text{push}_a) \in \Delta \\
&\exists q_1' \ \exists \text{loop}: && q_1, \bot a \to q_1', \bot a \text{ and } (q_1', a, q_2, \text{clone}) \in \Delta \\
&\exists q_2' \ \exists \text{loop}: && q_2, \bot a \to q_2', \bot a \text{ and } (q_2', a, q_3, \text{push}_b) \in \Delta
\end{aligned}$$

Example Valid Configurations

TECHNISCHE UNIVERSITÄT DARMSTADT

$$q_0, \bot \xrightarrow[\text{push}_a]{\text{Loop+}} q_1, \bot a \xrightarrow[\text{clone}]{\text{Loop+}} q_2, \bot a \xrightarrow[\text{push}_b]{\text{Loop+}} q_3, \bot a \xrightarrow[\text{push}_c]{\text{Loop+}} q_4, \bot a$$

$$\bot a \qquad \bot ab \qquad \bot abc$$

$$q_0 \qquad q_1$$

$$\bot \qquad \bot \to a \qquad \bot \to a \qquad \bot \to a \qquad \bot \to a$$

$$\downarrow \qquad \downarrow \qquad \downarrow$$

$$\varepsilon \qquad \varepsilon \to b \qquad \varepsilon \to b \to c$$

$$q_2 \qquad q_3 \qquad q_4$$

Labelling valid if

$$\begin{aligned}
&\exists q_0' \ \exists \text{loop}: && q_0, \bot \to q_0', \bot \text{ and } (q_0', \bot, q_1, \text{push}_a) \in \Delta \\
&\exists q_1' \ \exists \text{loop}: && q_1, \bot a \to q_1', \bot a \text{ and } (q_1', a, q_2, \text{clone}) \in \Delta \\
&\exists q_2' \ \exists \text{loop}: && q_2, \bot a \to q_2', \bot a \text{ and } (q_2', a, q_3, \text{push}_b) \in \Delta \\
&\exists q_3' \ \exists \text{loop}: && q_3, \bot ab \to q_3', \bot ab \text{ and } (q_3', \bot, q_4, \text{push}_c) \in \Delta
\end{aligned}$$

Summary

TECHNISCHE UNIVERSITÄT DARMSTADT

Known results (on NPT and CPG):

- Decidable for modal μ-calculus
- Undecidable for MSO

New result:

- Decidable FO model checking on NPT and (2)-CPG

Proof:

- Simulating NPT on CPG
- Tree-automaticity of 2-CPG

Still open:

- FO model checking on arbitrary CPG
- Are 3-CPG tree-automatic?