

FO Model Checking on Nested Pushdown Trees



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Nested Pushdown Trees (NPT)

- ▶ Nested words: $(\curvearrowleft (\rightarrow 5 \rightarrow + \rightarrow 7 \rightarrow) \rightarrow \cdot \rightarrow 3 \rightarrow)$
- ▶ Nested trees: every path is a nested word
- ▶ 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)

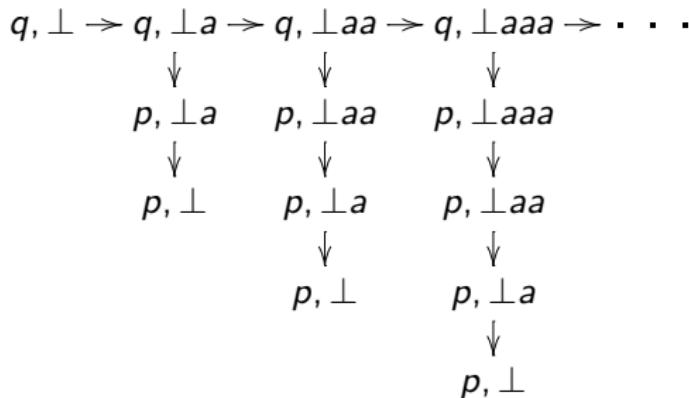
modal μ -calculus model checking is decidable

monadic second-order model checking is undecidable

NPT example

Transitions:

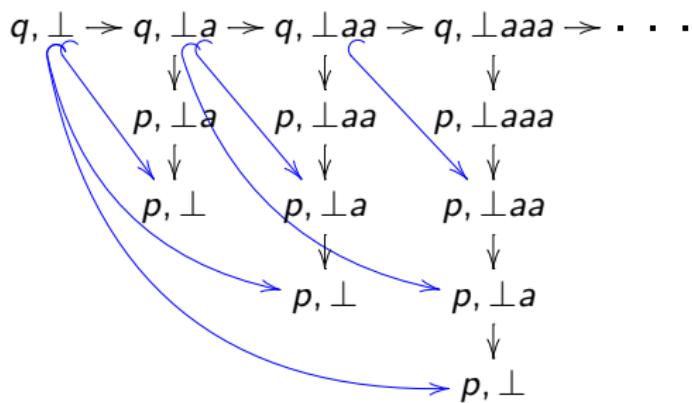
$$(q, \perp) \rightarrow (q, \text{push}_a) \quad (q, a) \rightarrow (q, \text{push}_a) \quad (q, a) \rightarrow (p, \text{id}) \quad (p, a) \rightarrow (p, \text{pop})$$



NPT example

Transitions:

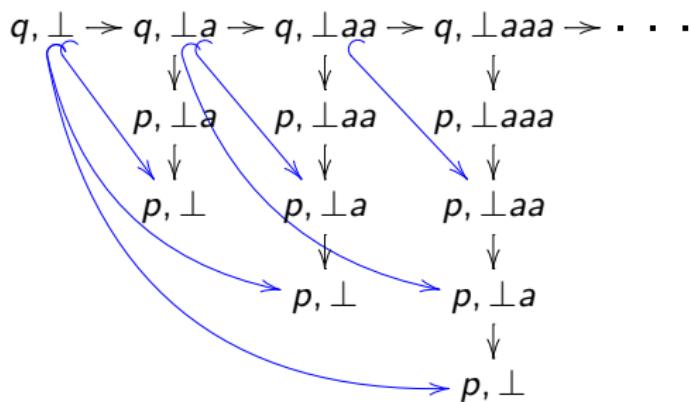
$$(q, \perp) \rightarrow (q, \text{push}_a) \quad (q, a) \rightarrow (q, \text{push}_a) \quad (q, a) \rightarrow (p, \text{id}) \quad (p, a) \rightarrow (p, \text{pop})$$



NPT example

Transitions:

$$(q, \perp) \rightarrow (q, \text{push}_a) \quad (q, a) \rightarrow (q, \text{push}_a) \quad (q, a) \rightarrow (p, \text{id}) \quad (p, a) \rightarrow (p, \text{pop})$$



Interpretation of grid \Rightarrow undecidable for MSO

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 nice CPG:

- ▶ Pumping lemmas
- ▶ Gaifman-like argument
- ▶ Brute force inspection of small witnesses

Collapsible Pushdown Systems (of order 2)

- ▶ 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)

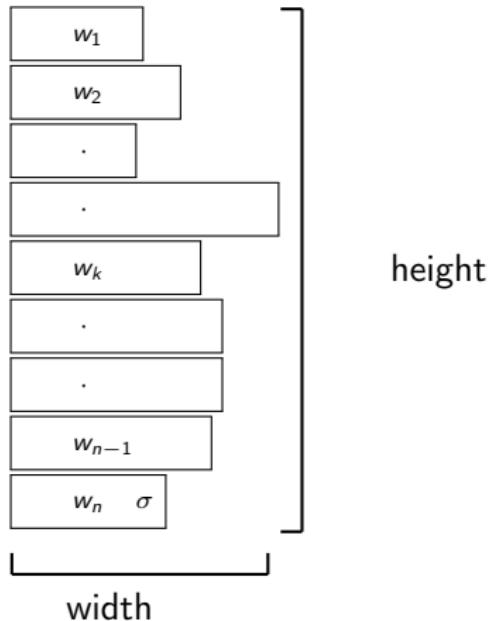
modal μ -calculus model checking: decidable

MSO model checking: undecidable

Status of FO model checking: open

stack of stacks and stack-operations

A level 2 stack is a list of stacks $s = w_1 : w_2 : \dots : w_n$.

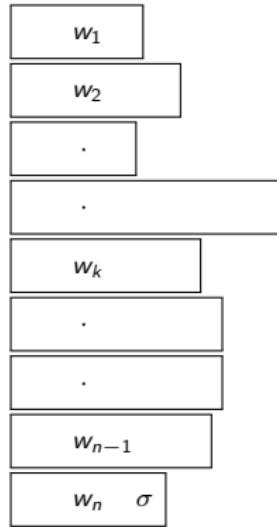


stack of stacks and stack-operations

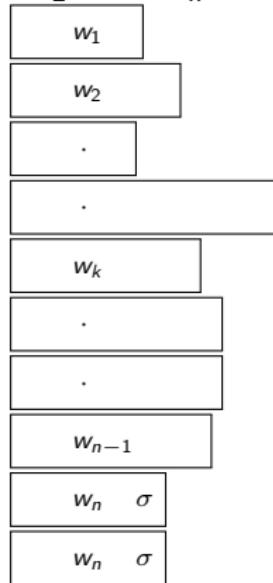


TECHNISCHE
UNIVERSITÄT
DARMSTADT

A level 2 stack is a list of stacks $s = w_1 : w_2 : \dots : w_n$.

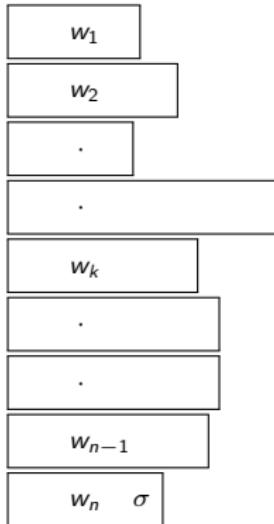


clone
→

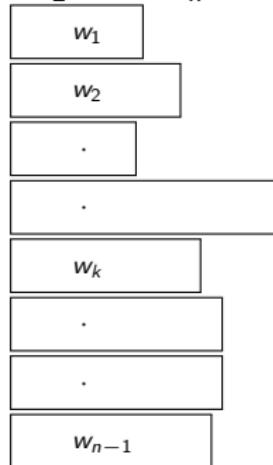


stack of stacks and stack-operations

A level 2 stack is a list of stacks $s = w_1 : w_2 : \dots : w_n$.

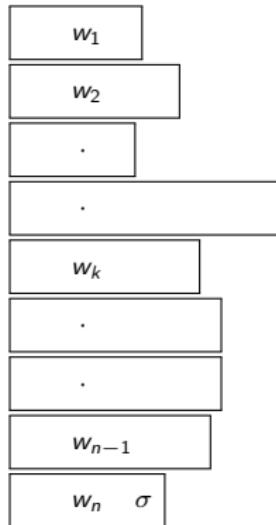


pop_2
→

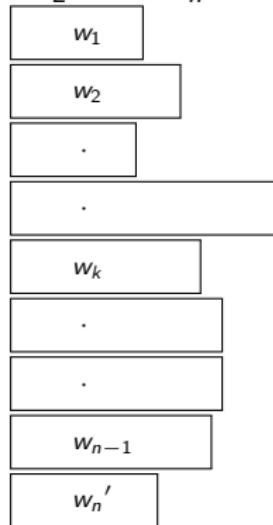


stack of stacks and stack-operations

A level 2 stack is a list of stacks $s = w_1 : w_2 : \dots : w_n$.

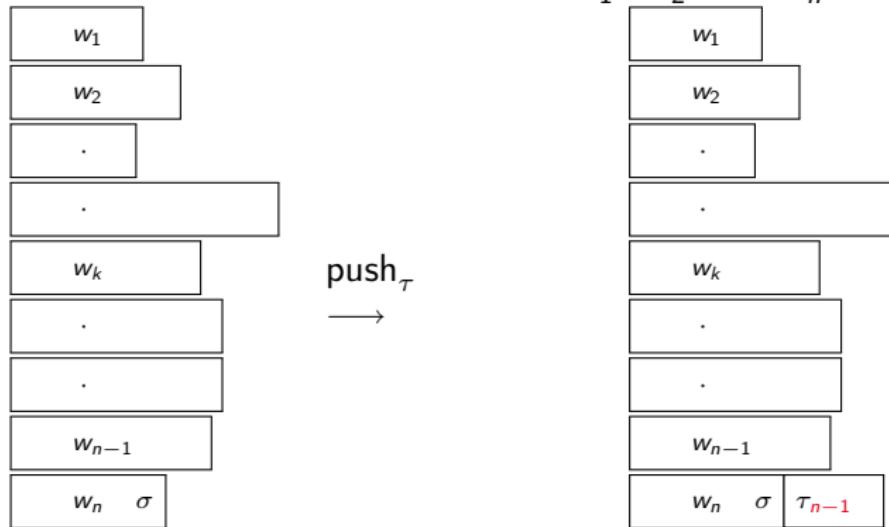


pop_1
→



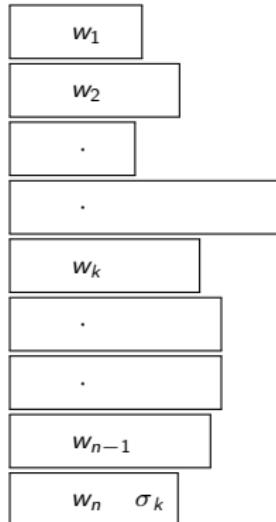
stack of stacks and stack-operations

A level 2 stack is a list of stacks $s = w_1 : w_2 : \dots : w_n$.

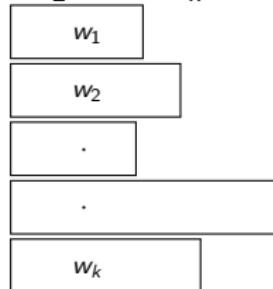


stack of stacks and stack-operations

A level 2 stack is a list of stacks $s = w_1 : w_2 : \dots : w_n$.



collapse
→



Collapsible Pushdown Graphs (CPG)



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Collapsible pushdown system with stack of stacks
- ▶ Operations: push_σ , clone, pop_1 , pop_2 , collapse
- ▶ Configurations: (q, s) – q a state, s a stack
Edges: $(q, s) \xrightarrow{\text{op}} (q', s')$
for transitions $(q, \text{top}_1(s)) \rightarrow (q', \text{op})$ with $\text{op}(s) = s'$
- ▶ CPG: Graph of *reachable* configurations with labeled transitions

Vertices in NPT: runs of pushdown system

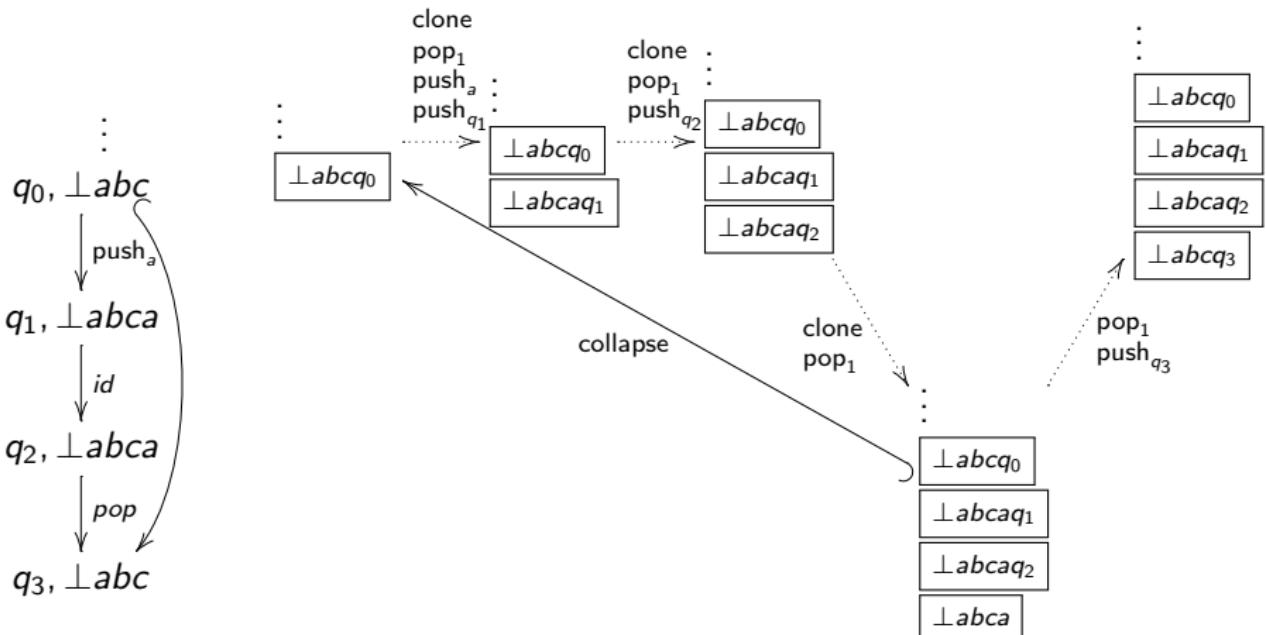
Vertices in CPG: configurations of higher-order pushdown system

Simulation of NPT with CPS

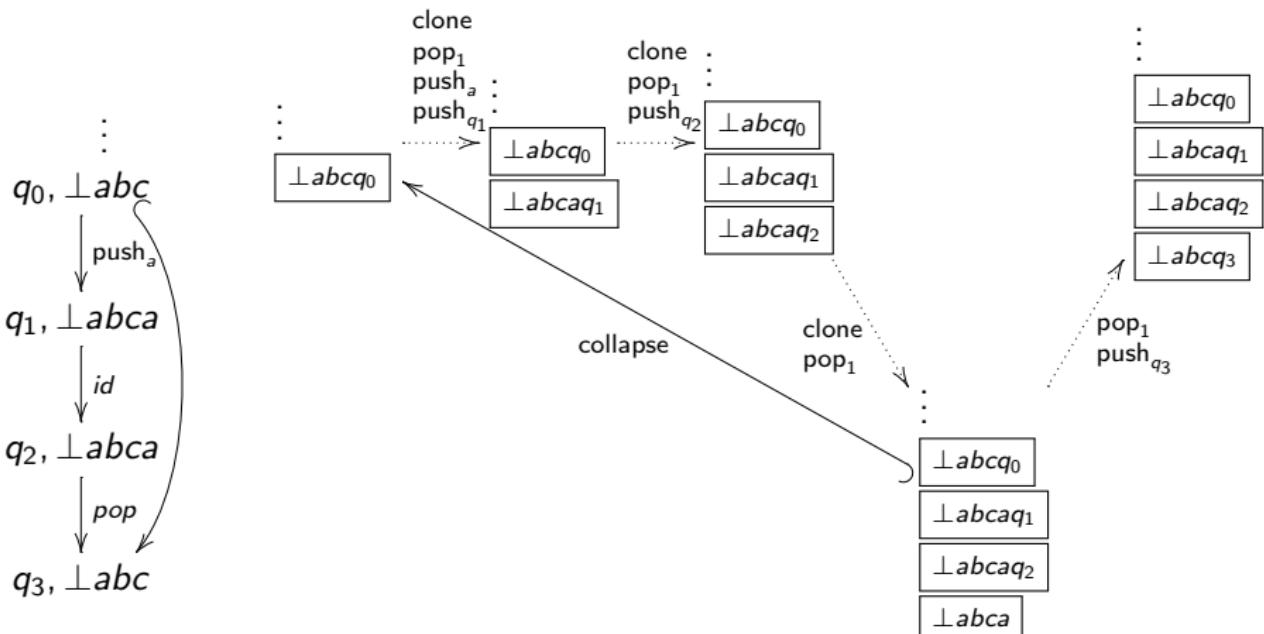


NPT	CPS
run of a pushdown system	list of stacks + state
nesting relation \hookrightarrow	collapse
1 stack operation (level 1)	up to 4 stack operations (level 2)

Simulation of NPT in CPG

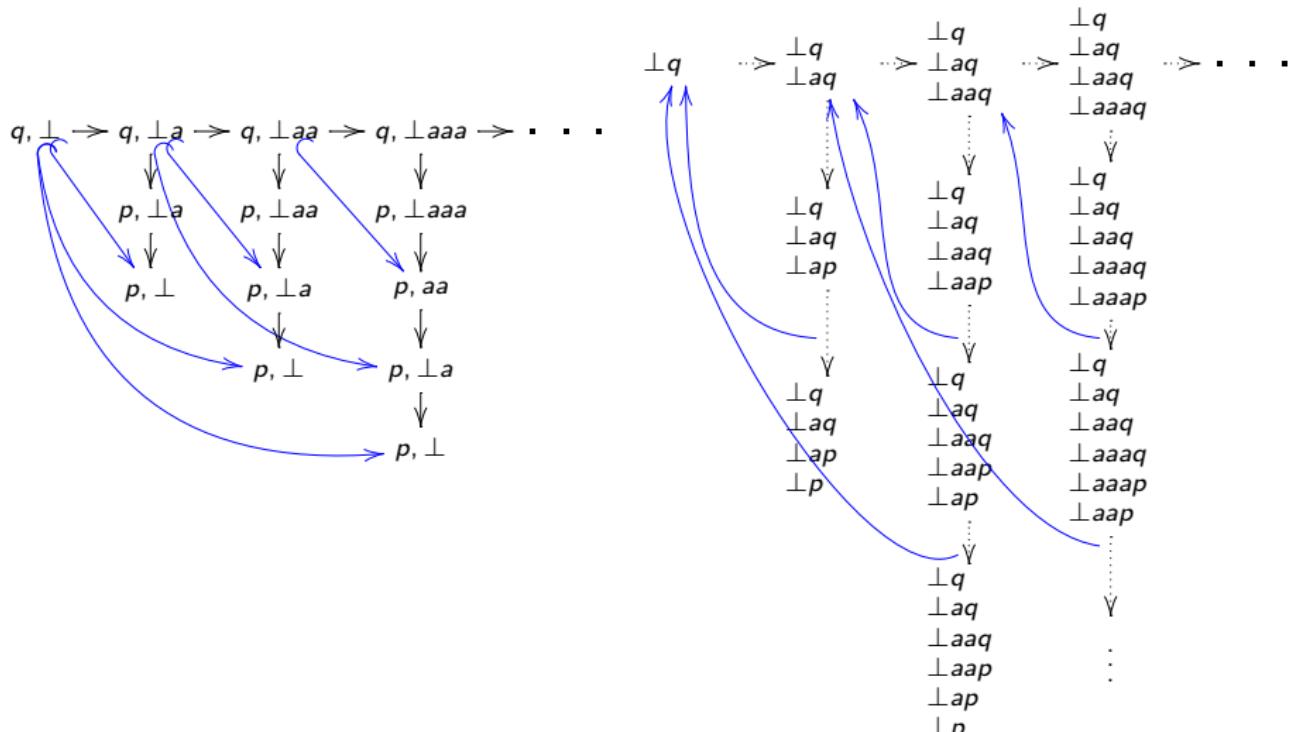


Simulation of NPT in CPG



Nice collapse: Preservation of last word!

Example of Simulation



- ▶ Only use “nice” CPG: adjacent configurations have similar last stack

Lemma

FO-model checking on “nice” CPG is decidable

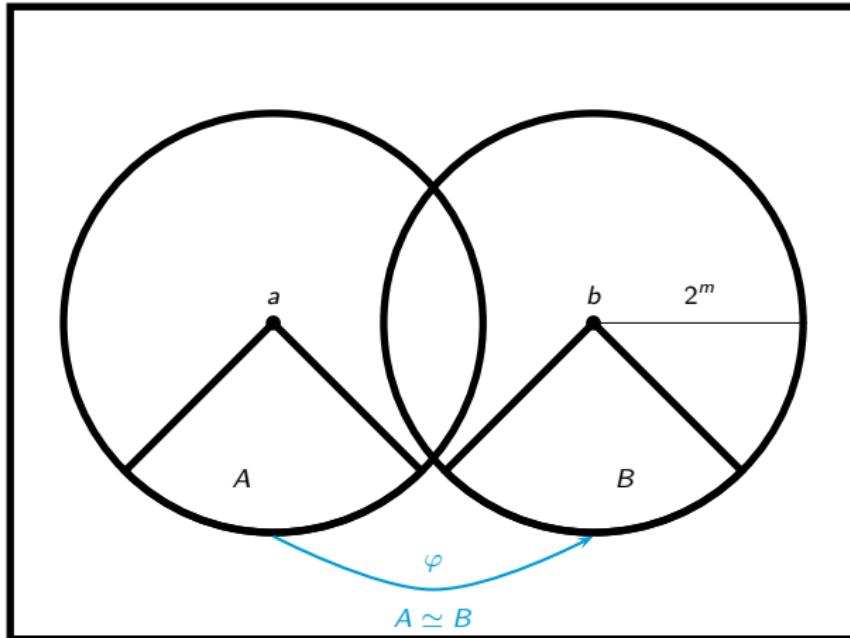
Proof idea:

1. Pumping lemmas: large stack $s \rightarrow$ small stack s'
2. Gaifman-like argument: $s \simeq_m s'$



Allows brute force inspection of small configurations

Gaifman-like Lemma



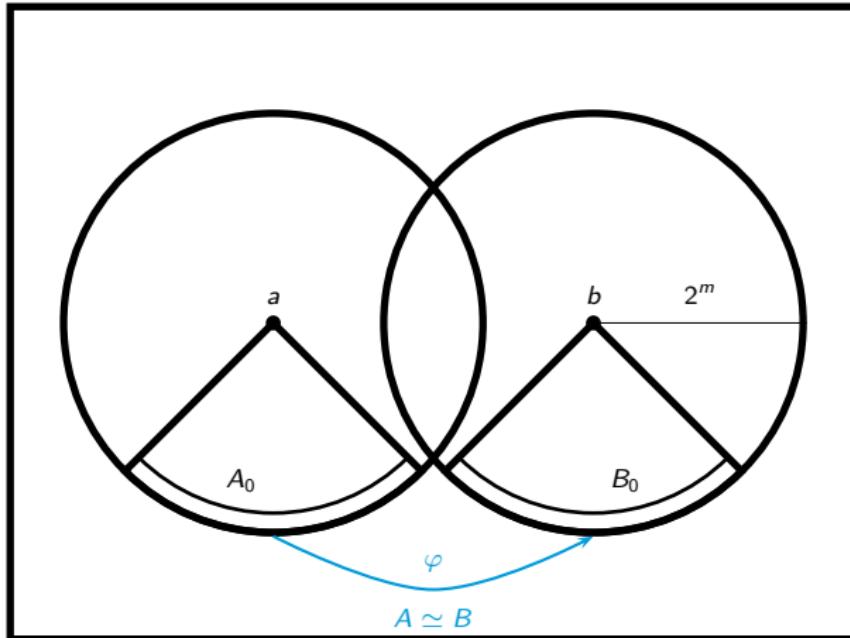
$$A \cap B = \emptyset$$

$$A \times B \cap E = \emptyset$$

$$C = G \setminus (A \cup B)$$

$$\varphi : A, a \simeq B, b$$

Gaifman-like Lemma



$$A \cap B = \emptyset$$

$$A \times B \cap E = \emptyset$$

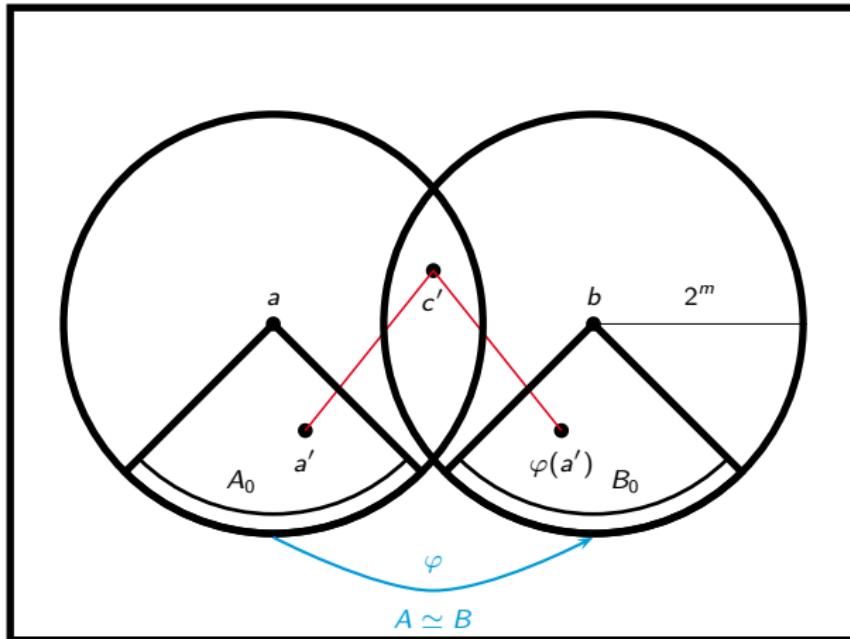
$$C = G \setminus (A \cup B)$$

$$\varphi : A, a \simeq B, b$$

$$A_0 = A \cap N_{2^m-1}(a)$$

$$B_0 = B \cap N_{2^m-1}(b)$$

Gaifman-like Lemma



$$A \cap B = \emptyset$$

$$A \times B \cap E = \emptyset$$

$$C = G \setminus (A \cup B)$$

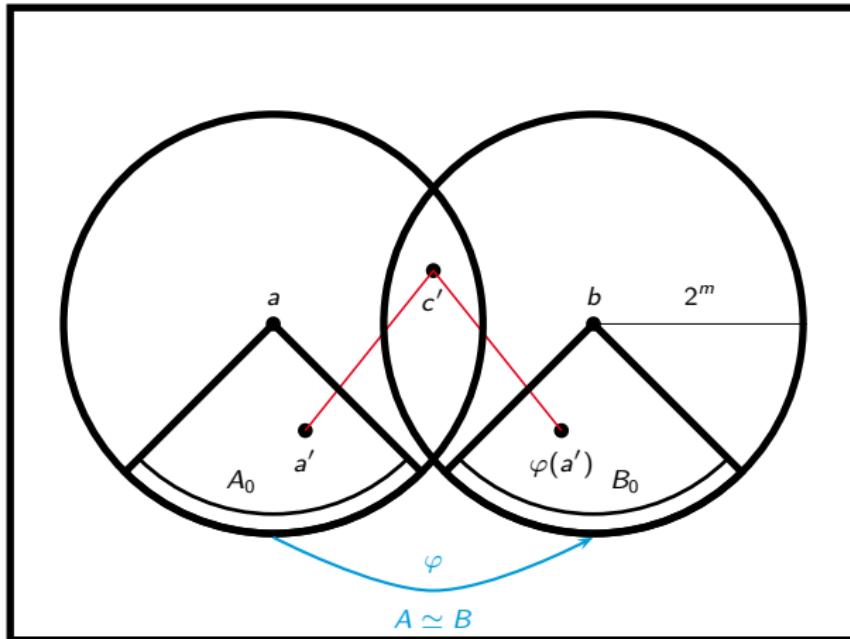
$$\varphi : A, a \simeq B, b$$

$$A_0 = A \cap N_{2^m-1}(a)$$

$$B_0 = B \cap N_{2^m-1}(b)$$

$$\varphi : A_0 \xrightarrow[C]{} B_0$$

Gaifman-like Lemma



$$A \cap B = \emptyset$$

$$A \times B \cap E = \emptyset$$

$$C = G \setminus (A \cup B)$$

$$\varphi : A, a \simeq B, b$$

$$A_0 = A \cap N_{2^m-1}(a)$$

$$B_0 = B \cap N_{2^m-1}(b)$$

$$\varphi : A_0 \xrightarrow[C]{} B_0$$



$$a, \bar{c} \simeq_m b, \bar{c}$$

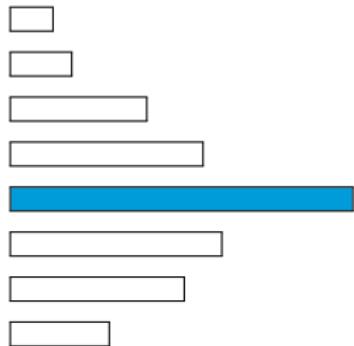
for all $\bar{c} \in C$

Pumping Lemmas

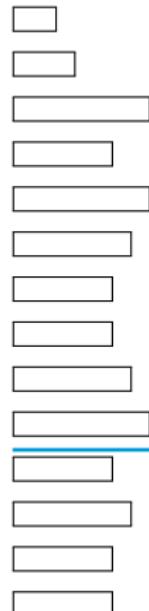


TECHNISCHE
UNIVERSITÄT
DARMSTADT

large width



small width, large height

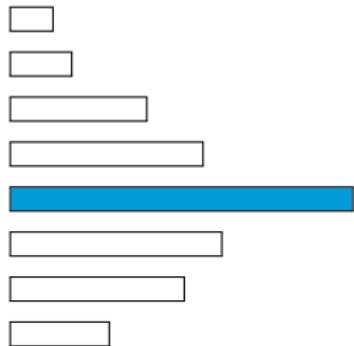


Pumping Lemmas

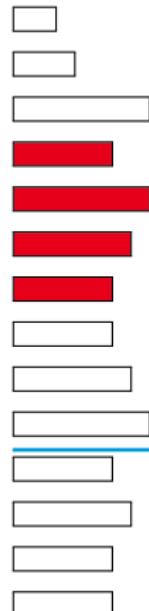


TECHNISCHE
UNIVERSITÄT
DARMSTADT

large width

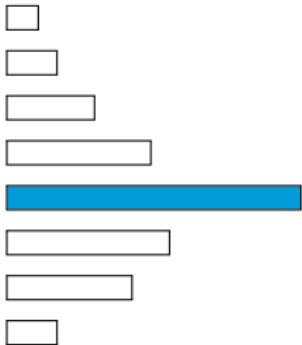


small width, large height



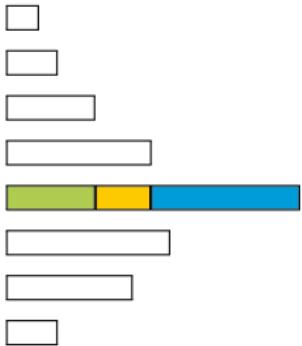
Pumping and \simeq_m on nice CPG

a



Pumping and \simeq_m on nice CPG

a

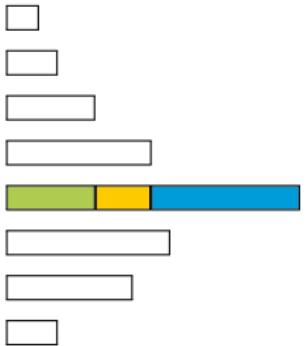


Pumping and \simeq_m on nice CPG



TECHNISCHE
UNIVERSITÄT
DARMSTADT

a

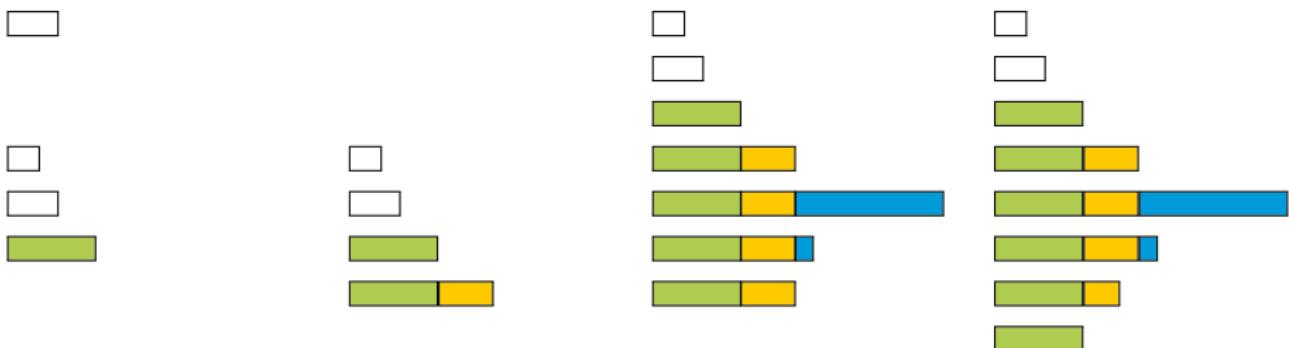


Pumping and \simeq_m on nice CPG



TECHNISCHE
UNIVERSITÄT
DARMSTADT

a



Pumping and \simeq_m on nice CPG

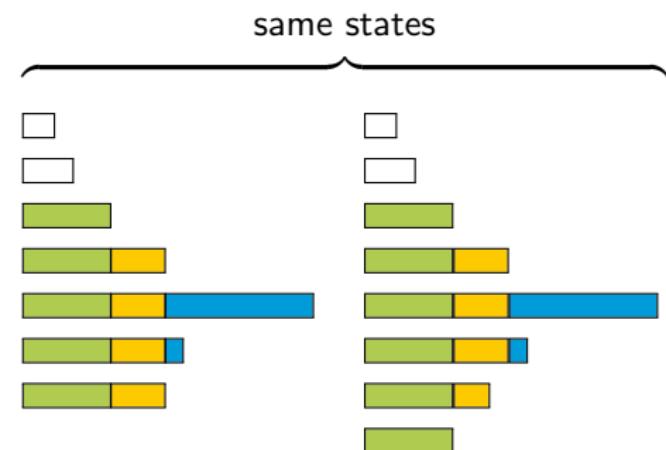


TECHNISCHE
UNIVERSITÄT
DARMSTADT

a



same states



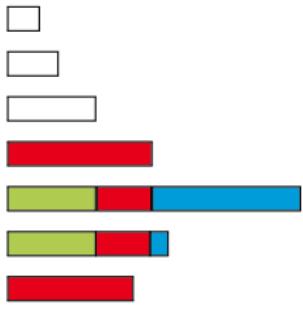
same states

Pumping and \simeq_m on nice CPG

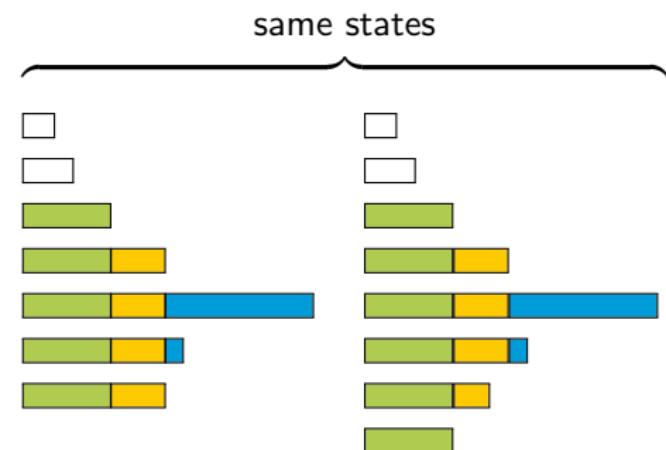


TECHNISCHE
UNIVERSITÄT
DARMSTADT

a



same states

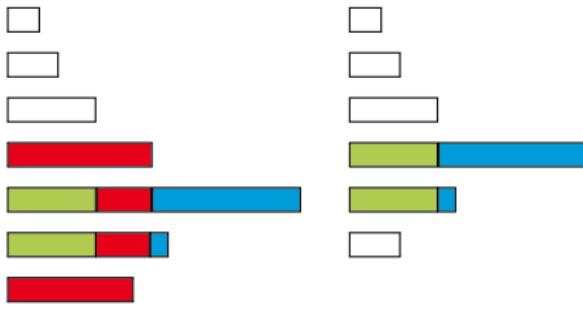


Pumping and \simeq_m on nice CPG

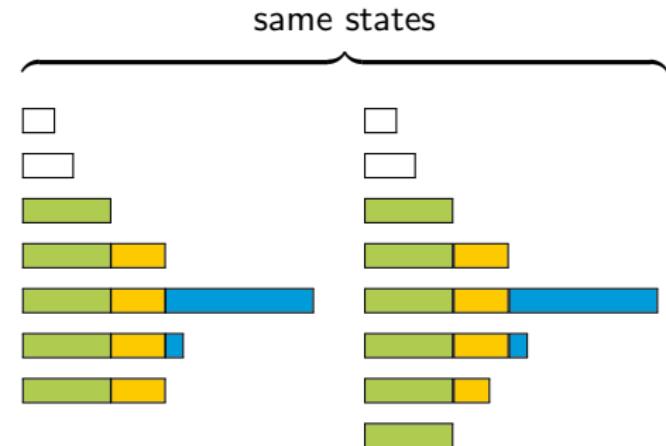


TECHNISCHE
UNIVERSITÄT
DARMSTADT

a



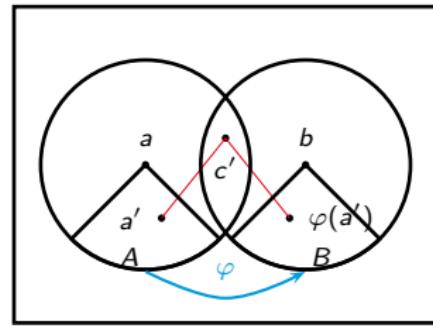
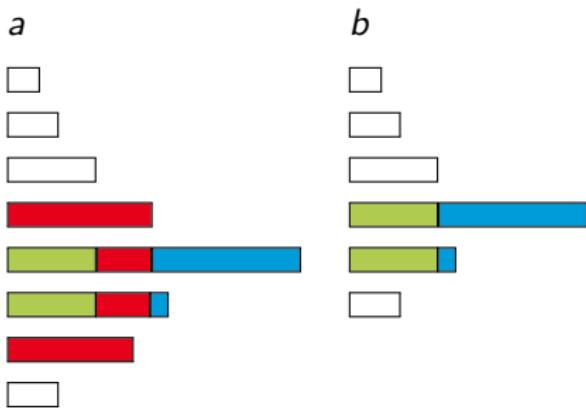
same states



Pumping and \simeq_m on nice CPG



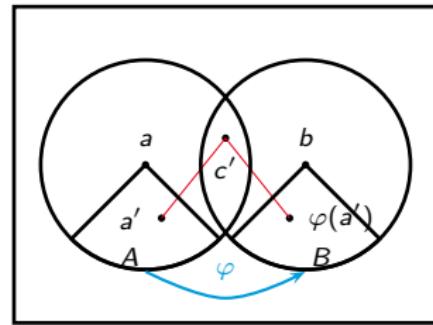
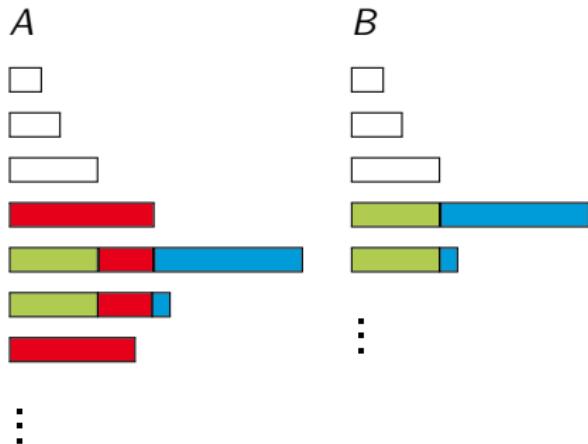
TECHNISCHE
UNIVERSITÄT
DARMSTADT



Pumping and \simeq_m on nice CPG



TECHNISCHE
UNIVERSITÄT
DARMSTADT

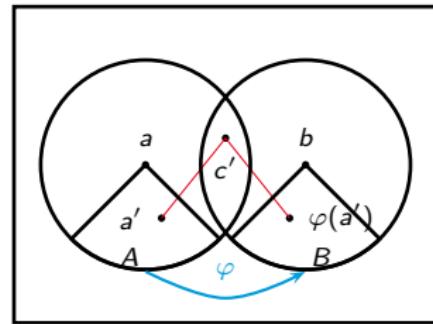
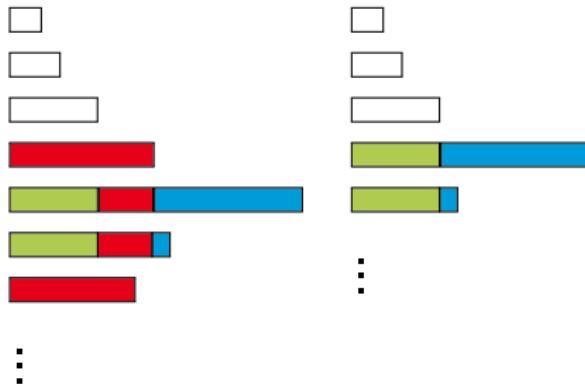


Pumping and \simeq_m on nice CPG



TECHNISCHE
UNIVERSITÄT
DARMSTADT

A B



⋮

↓ collapse



↓ collapse



Summary

Known results (on NPT and CPG):

- ▶ Decidable for modal μ -calculus
- ▶ Undecidable for MSO

New result:

- ▶ Decidable FO model checking on NPT

Proof:

- ▶ Simulating NPT on “nice” CPG
- ▶ Pumping Lemma + Gaifman-like argument on graphs of small diameter

Still open:

- ▶ FO model checking on arbitrary CPG