First-order Model Checking on Generalisations of Pushdown Graphs

Alexander Kartzow

Universität Leipzig

June, 2012

Generalising Pushdown Systems

1. Higher-order pushdowns: (collapsible) higher-order pushdown systems (CHOPS/ HOPS)
2. Nested pushdown trees (NPT)

1+2. Higher-order nested pushdown trees (HONPT)

Generalising Pushdown Systems

1. Higher-order pushdowns: (collapsible) higher-order pushdown systems (CHOPS/ HOPS)
2. Nested pushdown trees (NPT)

1+2. Higher-order nested pushdown trees (HONPT)

Question

Decidability of first-order logic (FO) on structures generated by generalised pushdown systems?

Pushdown Graphs and Trees

Definition

CHOPS: Pushdown system with nested stack;
Operations push_i, pop_i, collapse

Fix CHOPS $\mathcal{S}$

Definition (configuration graph (CPG))

Graph of $\mathcal{S}$ = reachable configurations + edges induced by transitions.

Definition (generated tree)

Tree of $\mathcal{S}$ = unfolding of its graph

- Node: run
- Edge: extension of run by one transition

The FO model checking problem on $\mathcal{C}$

Problem

Input: $\mathfrak{G} \in \mathcal{C}, \varphi \in \mathrm{FO}$
Output: $\mathfrak{G} \models \varphi$

Remark

Graphs of HOPS have decidable MSO
Graphs of CHOPS have undecidable MSO
Graphs of CHOPS have decidable μ-calculus

FO on CPG

1-CPG = PG
FO: ATIME(exp(n), cn)
FO+Reach: nonelementary

2-CPG
FO: nonelementary
FO+Reach: nonelementary

3-CPG
FO: undecidable

4-CPG
FO: undecidable

⋮

Tree-Automaticity

Definition

Graph $\mathfrak{G} = (V, E)$ is *tree-automatic* if

- $\exists f : V \to R$, R language of a finite tree-automaton, f bijective
- $f(E)$ language of synchonous 2-tape finite tree-automaton.

Theorem (Blumensath '99)

$\mathcal{C}$ a class of tree-automatic structures
FO *model checking on $\mathcal{C}$: decidable.*

2-CPG are Tree-Automatic

Principle of Encoding

$\bot\sigma_n w_l$

$\vdots$

$\bot\sigma_n w_k$

$\vdots$

$\bot\sigma_2 w_j$

$\vdots$

$\bot\sigma_2 w_{i+2}$

$\bot\sigma_2 w_{i+1}$

$\bot\sigma_1 w_i$

$\vdots$

$\bot\sigma_1 w_2$

$\bot\sigma_1 w_1$

2-CPG are Tree-Automatic

Principle of Encoding

$\bot\sigma_n w_l$

$\vdots$

$\bot\sigma_n w_k$

$\vdots$

$\bot\sigma_2 w_j$

$\vdots$

$\bot\sigma_2 w_{i+2}$

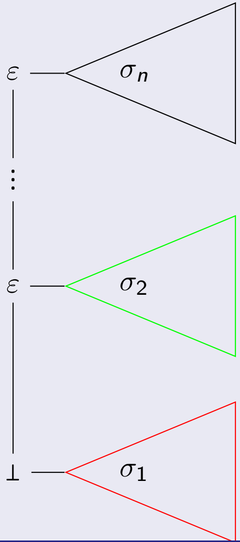
$\bot\sigma_2 w_{i+1}$

$\bot\sigma_1 w_i$

$\vdots$

$\bot\sigma_1 w_2$

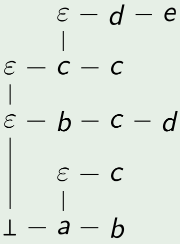
$\bot\sigma_1 w_1$



Concrete Example

Example

$\bot cde$
$\bot cc$
$\bot bcd$
$\bot ac$
$\bot ab$



Lemma

Tree encoding turns transitions into
a language of a 2-tape synchonous automaton

Reachability is Tree-Automatic

Lemma

- *Image of reachable configurations is regular*
- *Reachability is 2-tape regular*

Theorem (Kartzow '10)

The FO *model checking problem on 2-CPG*+Reach*: decidable*
Running time of algorithm: nonelementary.

Remark

Even for FO only: we cannot do better
(FO-interpret infinite binary order tree in a 2-CPG).

FO model checking on 3-CPG

Theorem (Broadbent'12)

Post's correspondence problem reduces to FO *model checking on 3-CPG*

Corollary

For $n \geq 3$, FO *model checking on n-CPG is undecidable.*

FO on CPG

1-CPG = PG
FO: ATIME(exp(n), cn)
FO+Reach: nonelementary

2-CPG
FO: nonelementary
FO+Reach: nonelementary

3_2-CPG
FO: nonelementary
FO+Reach: undecidable

3_3-CPG
FO: undecidable

3-CPG
FO: undecidable

4-CPG
FO: undecidable

⋮

Nested Pushdown Trees

Motivation

Make function calls in (first-order) recursive programs visible.

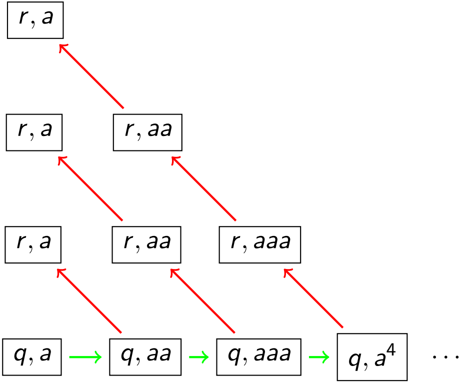
Fix a PS $\mathcal{S}$.

Definition (Alur, Chauduri, Madhusudan)

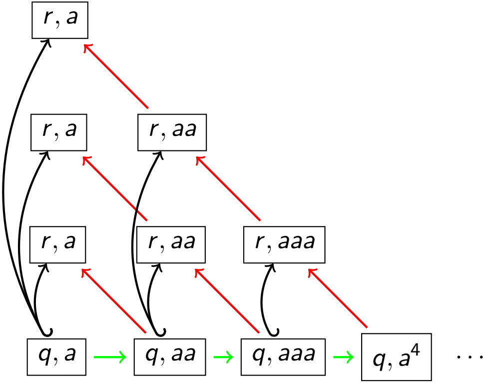
Nested pushdown tree (NPT) of $\mathcal{S}$:

- Pushdown tree, expanded by
- Jump edges: connect corresponding push and pop.

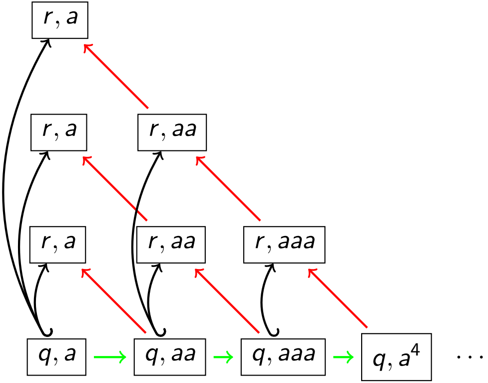
Example of NPT



Example of NPT



Example of NPT



Grid is MSO-definable ⇒ MSO undecidable

But μ-calculus is decidable

Higher-Order Nested Puhsdown Trees

Concept of jump edges generalises to order k-PS:

Definition (higher-order nested pushdown tree)

HONPT: HOPT + edges between corresponding push_k and pop_k.

Note: we start with a non-collapsing system.

Connection of the Hierarchies

Lemma

$\forall \mathcal{S}$ level k PS $\exists \mathcal{T}$ level $k+1$ CPS such that
$\mathfrak{N} := \mathrm{HONPT}(\mathcal{S})$ is FO-interpretable in the graph $\mathfrak{G}$ of $\mathcal{T}$.

Proof.

Node in $\mathfrak{N}$:	Run $(q_1, s_1), \dots, (q_n, s_n)$
Seen as $k+1$-pushdown:	$\mathrm{push}_{q_1}(s_1) : \dots : \mathrm{push}_{q_n}(s_n)$
Edges of $\mathfrak{N}$:	4 level $k+1$ pushdown operations
Jump edges:	reversed collapse edges □

FO on CPG and HONPT

1-CPG = PG
FO: ATIME($\exp(n)$, cn)
FO+Reach: nonelementary

2-CPG
FO: nonelementary
FO+Reach: nonelementary

NPT
FO:
FO+Reach:

3_2-CPG
FO: nonelementary
FO+Reach: undecidable

3_3-CPG
FO: undecidable

2-NPT
FO:
FO+Reach:

3-CPG
FO: undecidable

4-CPG
FO: undecidable

⋮

3-NPT
FO:
FO+Reach:

More tractable "subclass" of CPG for FO model checking

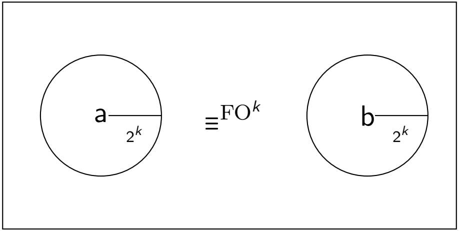
Theorem

FO *model checking on PT: ATIME*$(\exp(n), cn)$*-complete*
FO *model checking on NPT: ATIME*$(\exp_2(n), cn)$*-complete*
FO *model checking on 2-NPT: decidable*

Proof(Sketch).

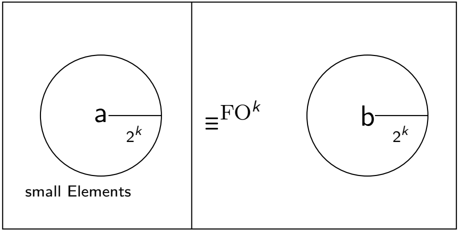
Locality analysis and pumping: short witnesses for $\exists x \dots$ □

Locality and Bounded Search for Witnesses



- $\mathfrak{G} \models \varphi(a)$ iff $\mathfrak{G} \models \varphi(b)$

Locality and Bounded Search for Witnesses



- $\mathfrak{G} \models \varphi(a)$ iff $\mathfrak{G} \models \varphi(b)$
- Small elements: n-tuples representable in space $f(n)$
- Small witness property:

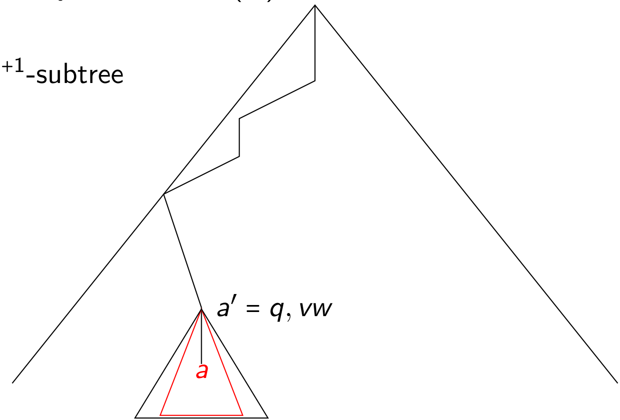
  $\forall \varphi(x) \in \mathrm{FO}$ $\quad \mathfrak{G} \models \exists x \varphi \Leftrightarrow$ there is a small $a \in \mathfrak{G}$ $\quad \mathfrak{G} \models \varphi(a)$

  $\quad \mathfrak{G} \models \forall x \varphi \Leftrightarrow$ for all small $a \in \mathfrak{G}$ $\quad \mathfrak{G} \models \varphi(a)$
- $\mathcal{C}$ a class of graphs with small witness property

  $\Rightarrow$ FO model checking on $\mathcal{C}$ reduces to
  FO model checking on finite structures of size $\exp(f(n))$

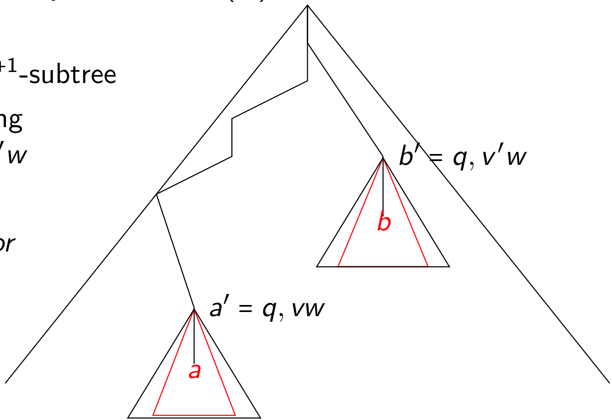
Locality in Pushdown Trees

- 2^k-ball(a) determined by 2^{k+1}-subtree(a')
- $|w| = 2^{k+1}$
- q, w determines 2^{k+1}-subtree



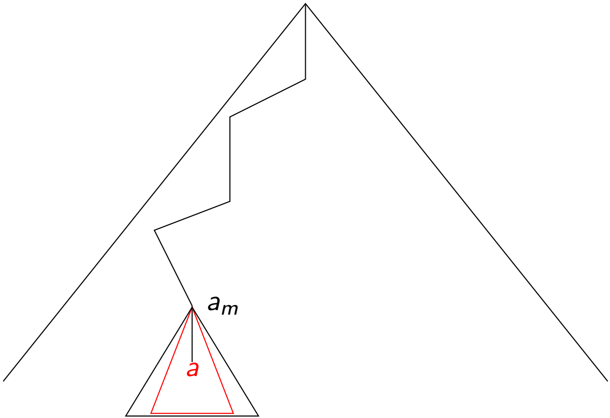
Locality in Pushdown Trees

- 2^k-ball(a) determined by 2^{k+1}-subtree(a')
- $|w| = 2^{k+1}$
- q, w determines 2^{k+1}-subtree
- context free pumping
  $\Rightarrow \exists$ small $b' = q, v'w$
- small: $O(\exp(k))$
- a': *relevant ancestor*



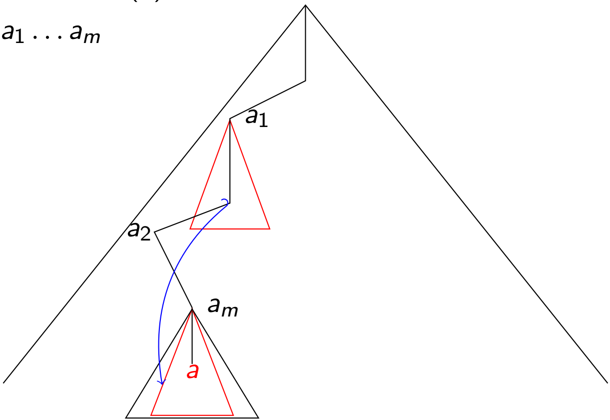
Corollary: FO model checking on PT is in ATIME$(\exp(n), cn)$

Locality in the 1-NPT case



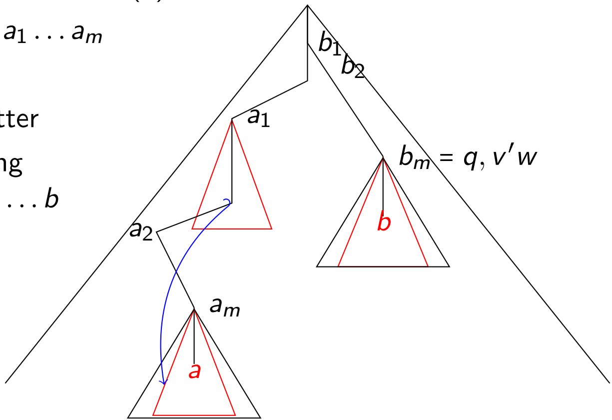
Locality in the 1-NPT case

- jumps cause 'nonlocal' 2^k-ball(a)
- *relevant ancestors*: $a_1 \ldots a_m$



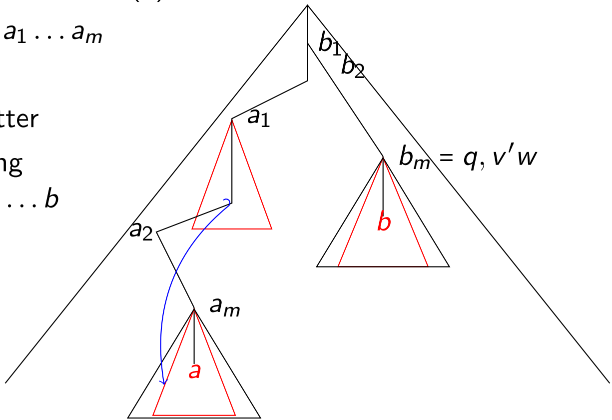
Locality in the 1-NPT case

- jumps cause 'nonlocal' 2^k-ball(a)
- *relevant ancestors*: $a_1 \dots a_m$
- $m \leq \exp_2(k)$.
- $a_i \dots a_{i+1}$: add 1 letter
- context free pumping
  $\Rightarrow \exists$ small $b_1 \dots b_m \dots b$
- small: $O(\exp_2(k))$



Locality in the 1-NPT case

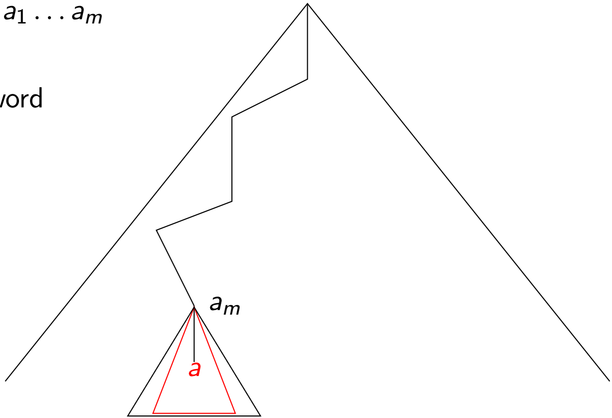
- jumps cause 'nonlocal' 2^k-ball(a)
- *relevant ancestors*: $a_1 \dots a_m$
- $m \leq \exp_2(k)$.
- $a_i \dots a_{i+1}$: add 1 letter
- context free pumping
  $\Rightarrow \exists$ small $b_1 \dots b_m \dots b$
- small: $O(\exp_2(k))$



Corollary: FO model checking on 1-NPT is in ATIME($\exp_2(n), cn$)

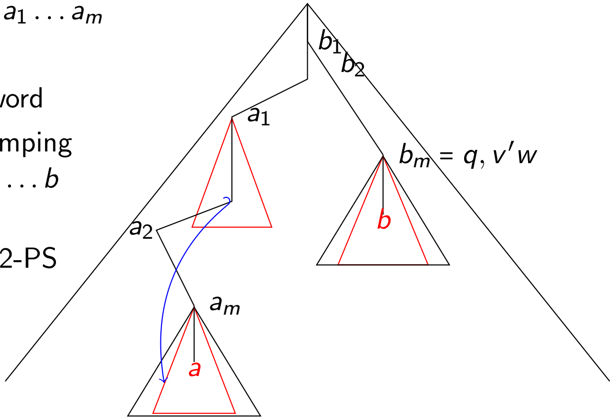
Locality in the 2-NPT case

- jumps cause 'nonlocal' 2^k-ball(a)
- *relevant ancestors*: $a_1 \ldots a_m$
- $m \leq \exp_2(k)$.
- $a_i \ldots a_{i+1}$: adds 1 word



Locality in the 2-NPT case

- jumps cause 'nonlocal' 2^k-ball(a)
- *relevant ancestors*: $a_1 \dots a_m$
- $m \leq \exp_2(k)$.
- $a_i \dots a_{i+1}$: adds 1 word
- iterated indexed pumping
  $\Rightarrow \exists$ small $b_1 \dots b_m \dots b$
- small: $O(f(n))$,
  f computable from 2-PS



Corollary: FO model checking on 2-NPT: decidable

FO on CPG and HONPT

1-CPG = PG
FO: ATIME$(\exp(n), cn)$
FO+Reach: nonelementary

2-CPG
FO: nonelementary
FO+Reach: nonelementary

NPT
FO: ATIME$(\exp_2(n), cn)$
FO+Reach: nonelementary

3_2-CPG
FO: nonelementary
FO+Reach: undecidable

3_3-CPG
FO: undecidable

2-NPT
FO: decidable (compl.?)
FO+Reach: ?
μ-calculus: ?

3-CPG
FO: undecidable

4-CPG
FO: undecidable

⋮

3-NPT
FO: ?
FO+Reach: ?
μ-calculus: ?