A Lower Bound for FO Model Checking on Nested Pushdown Trees

Alexander Kartzow | Uni Leipzig

March 20, 2012

kartzow@informatik.uni-leipzig.de

Motivation from Verification

- Verification of Recursive Programmes
  - Pushdown Tree: Programm flow
  - Property of Programm: Formalised in MSO
  - Check whether program flow satisfies property via MSO model checking on the pushdown tree

Pushdown system $\mathcal{S}$

Definition (Pushdown Tree)

Pushdown tree:

- domain: all runs of $\mathcal{S}$ (from intial configuration).
- δ-labelled edges: extension of run by transition δ

From Pushdown to Nested Pushdown

$\mathcal{C}$ class of structures, $\mathcal{L}$ logic

$\mathcal{L}$-Model Checking on $\mathcal{C}$

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

Theorem (Muller, Schupp)

MSO model checking on Pushdown Trees is decidable

From Pushdown to Nested Pushdown

$\mathcal{C}$ class of structures, $\mathcal{L}$ logic

$\mathcal{L}$-Model Checking on $\mathcal{C}$

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

Theorem (Muller, Schupp)

MSO model checking on Pushdown Trees is decidable

Problem for Verification

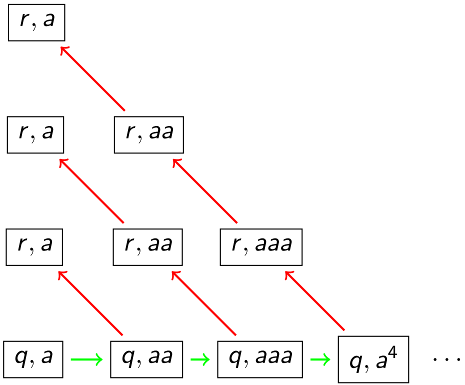
Pre- and postconditions on function calls not expressible
" A holds before call of function f $\Rightarrow$ B holds after f"

Possible solution

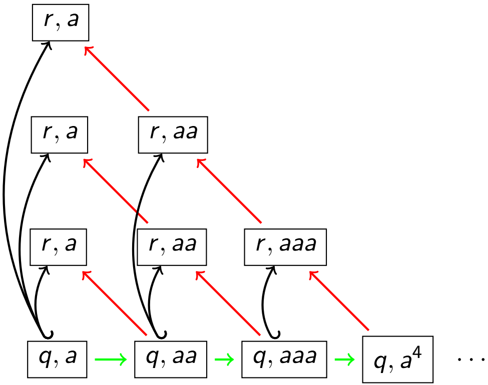
Nested pushdown trees (Alur et al. :

- Make corresponding function call and return visible
- Pushdown tree + *jump relation*

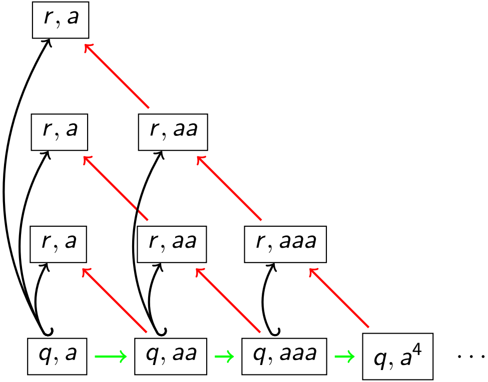
Example of NPT



Example of NPT



Example of NPT



Grid is MSO-definable

Properties of Nested Pushdown Trees

Theorem (Alur et al.)

MSO model checking on NPT: undecidable
$L\mu$ model checking on NPT: EXPTIME

Theorem (Kartzow)

FO model checking on NPT: $\mathrm{ATIME}(\exp_2(cn), cn)$

Proof Idea.

Analyse Ehrenfeucht-Fraísse games:
satisfiable formula $\exists x \varphi(x) \Rightarrow \varphi(\rho)$ holds with $|\rho| \leq \exp_2(|\varphi|)$ □

Main Result

Theorem (Kartzow)

FO model checking on NPT: ATIME$(\exp_2(cn), cn)$*-complete (with respect to reset-loglin-reductions)*

- Reset-loglin-reduction: fixed finite number of resets, logarithmic space, linear time
- Proof via interpretation method (Compton and Henson)

  Reset-loglin computable sequences of
  MSO-to-FO interpretations turn
  difficult MSO-theories into difficult FO-theories.

Proof Technique: Interpretation Method

Definition

$\mathcal{L}_n$: linear orders of size $\exp_2(13n)$ with unary predicate P

Straightforward adaptation of Comton's and Henson's work

$(\mathcal{L}_n)_{n\in\mathbb{N}}$ has hereditary $\mathrm{ATIME}(\exp_2(cn), cn)$ lower bound.

Corollary

$(\mathcal{L}_n)_{n\in\mathbb{N}} \overset{\textit{reset-loglin MSO-to-FO}}{\longrightarrow} \{\mathfrak{A}\}$

$\Rightarrow$ *FO-theory of* $\mathfrak{A}$ *is* $\mathrm{ATIME}(\exp_2(cn), cn)$*-hard.*

Reset-Loglin Computable Formulas

Definition (linear recursive definitions)

$(\varphi_n)_{n\in\mathbb{N}}$ is defined by linear recursion:
$\varphi_{n+1} = \exists x_1 \forall x_2, \ldots \forall x_{c\cdot n}(\psi \rightarrow \varphi_n)$

Properties of linear recursive definitions

Unfolding of φ_n: formula of size $c \cdot n$

Lemma (Compton and Henson)

$(\varphi_n)_{n\in\mathbb{N}}$ *defined using linear recursion*
$\Rightarrow n \mapsto \varphi_n$ *is reset-loglin computable*

Large Linear Orders in Nested Pushdown Trees

Goal: $(\mathcal{L}_n)_{n\in\mathbb{N}} \stackrel{\text{MSO-to-FO}}{\longrightarrow} NPT(\mathcal{S})$

Simplification of Presentation: linear orders of size $\exp_2(n)$

Idea

1. Paths of length $\exp(n)$ defined by $O(n)$-size FO-formula
2. Find nodes with $\exp_2(n)$ many ancestors at distance $\exp(n)$
3. Interpret order using the $\exp(n)$ paths
4. Interpret predicate P: use a 2-state pushdown system
5. Interpret set quantification using first-order quantification

1: Paths of length $\exp(n)$

Paths along jump and pop edges

$$a \overset{=1}{\underset{\hookrightarrow}{\rightarrow}} b \quad := \quad a \hookrightarrow b \vee a \rightarrow b$$

$$a \overset{=\exp(n)}{\underset{\hookrightarrow}{\rightarrow}} b \quad := \quad \exists c \; (a \overset{=\exp(n-1)}{\underset{\hookrightarrow}{\rightarrow}} c) \wedge (c \overset{=\exp(n-1)}{\underset{\hookrightarrow}{\rightarrow}} b)$$

1: Paths of length $\exp(n)$

Paths along jump and pop edges

$$
\begin{aligned}
a \overset{=1}{\underset{\hookrightarrow}{\color{red}\rightarrow}} b \quad &:= \quad a \hookrightarrow b \vee a \,{\color{red}\rightarrow}\, b \\
a \overset{=\exp(n)}{\underset{\hookrightarrow}{\color{red}\rightarrow}} b \quad &:= \quad \exists c \forall x, y \ \big((x, y) = (a, c) \vee (x, y) = (c, b) \big) \\
& \qquad \rightarrow x \overset{=\exp(n-1)}{\underset{\hookrightarrow}{\color{red}\rightarrow}} y \big)
\end{aligned}
$$

1: Paths of length exp(n)

Paths along jump and pop edges

$$
\begin{aligned}
a \overset{=1}{\underset{\hookrightarrow}{\color{red}\rightarrow}} b \quad &:= \quad a \hookrightarrow b \vee a \color{red}{\rightarrow} \color{black} b \\
a \overset{=\exp(n)}{\underset{\hookrightarrow}{\color{red}\rightarrow}} b \quad &:= \quad \exists c \forall x, y \ \big((x, y) = (a, c) \vee (x, y) = (c, b) \big) \\
& \qquad \rightarrow x \overset{=\exp(n-1)}{\underset{\hookrightarrow}{\color{red}\rightarrow}} y \big)
\end{aligned}
$$

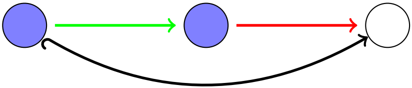
Analogously:

$$
\begin{aligned}
a \overset{\leq 1}{\underset{\hookrightarrow}{\color{red}\rightarrow}} b \quad &:= \quad a \hookrightarrow b \vee a \color{red}{\rightarrow} \color{black} b \vee a = b \\
a \overset{\leq\exp(n)}{\underset{\hookrightarrow}{\color{red}\rightarrow}} b \quad &:= \quad \exists c \forall x, y \ \big((x, y) = (a, c) \vee (x, y) = (c, b) \big) \\
& \qquad \rightarrow x \overset{\leq\exp(n-1)}{\underset{\hookrightarrow}{\color{red}\rightarrow}} y \big)
\end{aligned}
$$

2: Many Ancestors in generic Nested Pushdown Tree

Nested Pushdown Tree with arbitrary Push / Pop Sequences

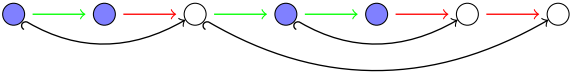
$$|\{x : x \overset{=1}{\underset{\hookrightarrow}{\rightarrow}} p\}| = \exp(1)$$



2: Many Ancestors in generic Nested Pushdown Tree

Nested Pushdown Tree with arbitrary Push / Pop Sequences

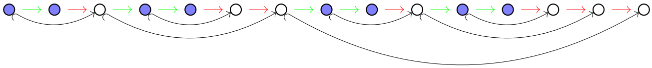
$|\{x : x \overset{=2}{\hookrightarrow} p\}| = \exp(2)$



2: Many Ancestors in generic Nested Pushdown Tree

Nested Pushdown Tree with arbitrary Push / Pop Sequences

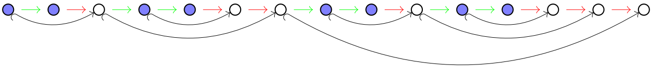
$|\{x : x \overset{=3}{\hookrightarrow} p\}| = \exp(3)$



2: Many Ancestors in generic Nested Pushdown Tree

Nested Pushdown Tree with arbitrary Push / Pop Sequences

$|\{x : x \overset{=3}{\underset{\hookrightarrow}{\rightarrow}} p\}| = \exp(3)$



General rule: $|\{x : x \overset{=exp(n)}{\underset{\hookrightarrow}{\rightarrow}} p\}| = \exp_2(n)$

Definition

$$\delta_n(x,p) \quad := \quad x \overset{=\exp(n)}{\underset{\hookrightarrow}{\rightarrow}} p$$

- defines set of $\exp_2(n)$ many nodes
- is of size $O(n)$

3: Order using linear sized formula

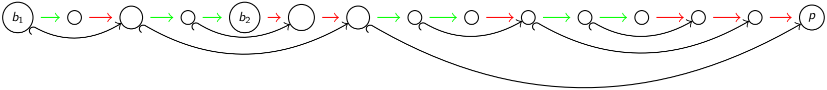
Lemma

Let $b_1 \overset{\leq \exp(n)}{\underset{\hookrightarrow}{\rightarrow}} p$ *and* $b_2 \overset{\leq \exp(n)}{\underset{\hookrightarrow}{\rightarrow}} p$

$$b_1 \text{ proper ancestor of } b_2 \quad \Leftrightarrow \varphi_n^{\leq}(b_1, b_2, p) \text{ holds}$$

$$:=$$

$$\exists c, d, e \quad c \overset{\leq \exp(n)}{\underset{\hookrightarrow}{\rightarrow}} p \wedge d \rightarrow c \wedge e \hookrightarrow c \wedge b_2 \overset{\leq \exp(n)}{\underset{\hookrightarrow}{\rightarrow}} e \wedge b_1 \overset{\leq \exp(n)}{\underset{\hookrightarrow}{\rightarrow}} d$$



3: Order using linear sized formula

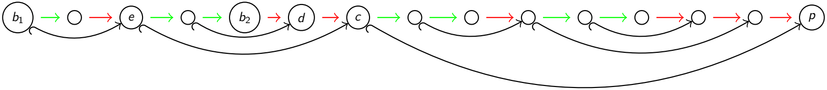
Lemma

Let $b_1 \overset{\leq \exp(n)}{\substack{\rightarrow \\ \hookrightarrow}} p$ *and* $b_2 \overset{\leq \exp(n)}{\substack{\rightarrow \\ \hookrightarrow}} p$

$$b_1 \text{ proper ancestor of } b_2 \quad \Leftrightarrow \varphi_n^{\leq}(b_1, b_2, p) \text{ holds}$$

$$:=$$

$$\exists c, d, e \quad c \overset{\leq \exp(n)}{\substack{\rightarrow \\ \hookrightarrow}} p \wedge d \rightarrow c \wedge e \hookrightarrow c \wedge b_2 \overset{\leq \exp(n)}{\substack{\rightarrow \\ \hookrightarrow}} e \wedge b_1 \overset{\leq \exp(n)}{\substack{\rightarrow \\ \hookrightarrow}} d$$



3: Order using linear sized formula

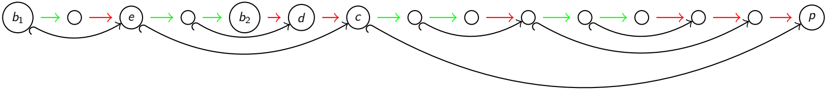
Lemma

Let $b_1 \overset{\leq \exp(n)}{\underset{\hookrightarrow}{\rightarrow}} p$ *and* $b_2 \overset{\leq \exp(n)}{\underset{\hookrightarrow}{\rightarrow}} p$

$$b_1 \textit{ proper ancestor of } b_2 \quad \Leftrightarrow \varphi_n^{\leq}(b_1, b_2, p) \textit{ holds}$$

$$:=$$

$$\exists c, d, e \quad c \overset{\leq \exp(n)}{\underset{\hookrightarrow}{\rightarrow}} p \wedge d \rightarrow c \wedge e \hookrightarrow c \wedge b_2 \overset{\leq \exp(n)}{\underset{\hookrightarrow}{\rightarrow}} e \wedge b_1 \overset{\leq \exp(n)}{\underset{\hookrightarrow}{\rightarrow}} d$$



Proof.

$b_2 \overset{*}{\underset{\hookrightarrow}{\rightarrow}} d \rightarrow c$: stacks between b_1 and d *greater than* stack of c

$b_1 \overset{*}{\underset{\hookrightarrow}{\rightarrow}} e \hookrightarrow c$: stack of e *equals* stack of c

e proper ancestor of $c \Rightarrow e$ ancestor of $d \Rightarrow e$ ancestor of b_2 □

3: Order using linear sized formula

Lemma

Let $b_1 \overset{\leq \exp(n)}{\overset{\rightarrow}{\hookrightarrow}} p$ *and* $b_2 \overset{\leq \exp(n)}{\overset{\rightarrow}{\hookrightarrow}} p$

$$b_1 \text{ proper ancestor of } b_2 \quad \Leftrightarrow \varphi_n^{\leq}(b_1, b_2, p) \text{ holds}$$

$$:=$$

$$\exists c, d, e \quad c \overset{\leq \exp(n)}{\overset{\rightarrow}{\hookrightarrow}} p \wedge d \rightarrow c \wedge e \hookrightarrow c \wedge b_2 \overset{\leq \exp(n)}{\overset{\rightarrow}{\hookrightarrow}} e \wedge b_1 \overset{\leq \exp(n)}{\overset{\rightarrow}{\hookrightarrow}} d$$

Corollary

$\overset{\rightarrow}{\hookrightarrow}^{*}$*-paths are unique*

Corollary

Ancestor ordering on $\{x : \delta_n(x, p)\}$ *is defined by* $O(n)$ *sized formula* $\varphi_n^{\leq}(x, y, p)$

4. States as Unary Predicate

- So far: only used nondeterministic choice of push or pop
- Now: nondeterministic choice of state q or r

Definition

$\varphi^P(x) \quad := \quad \mathrm{state}(x) = r$

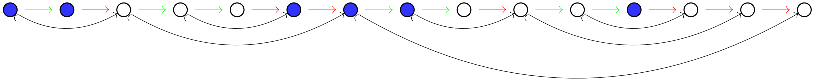
Theorem

For appropriate parameter p
$(\delta_n, \varphi_n^{\leq}, \varphi^P)$ interprets FO theory of linear orders of size $\exp_2(n)$ in FO theory of a generic nested pushdown tree

Definition (Ord(p))

$\mathrm{Ord}(p) :=$ linear order with predicate P obtained using $(\delta_n, \varphi_n^{\leq}, \varphi^P)$ and parameter p

5. Set Quantification



Definition ("$b \overset{=\exp(n)}{\rightrightarrows} p$ equals $b' \overset{=\exp(n)}{\rightrightarrows} p'$")

$$\varphi_0(b,p,b',p') := (b \to p \wedge b' \to p') \vee (b \hookrightarrow p \wedge b' \hookrightarrow p')$$

$$\varphi_{n+1}(b,p,b',p') := \exists c, c' \ \ (\varphi_n(c,p,c',p') \wedge \varphi_n(b,c,b',c'))$$

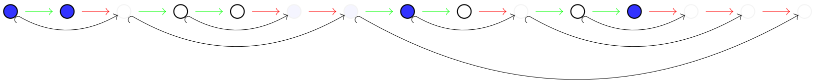
Lemma

$\exists X \quad \longrightarrow \quad \exists p'$

$x \in X \quad \longrightarrow \quad \exists x' \varphi_n((x,p,x',p') \wedge \varphi^P(x')$

interprets set quantification on linear orders in the FO theory of the nested pushdown tree

5. Set Quantification



Definition (“$b \overset{=\exp(n)}{\substack{\rightarrow \\ \hookrightarrow}} p$ equals $b' \overset{=\exp(n)}{\substack{\rightarrow \\ \hookrightarrow}} p'$”)

$$\varphi_0(b,p,b',p') := (b \rightarrow p \wedge b' \rightarrow p') \vee (b \hookrightarrow p \wedge b' \hookrightarrow p')$$

$$\varphi_{n+1}(b,p,b',p') := \exists c, c' \ \ (\varphi_n(c,p,c',p') \wedge \varphi_n(b,c,b',c'))$$

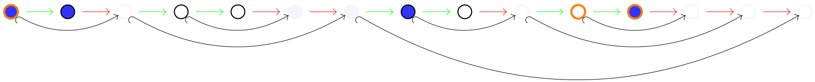
Lemma

$\exists X \quad \longrightarrow \quad \exists p'$

$x \in X \quad \longrightarrow \quad \exists x' \varphi_n((x,p,x',p') \wedge \varphi^P(x')$

interprets set quantification on linear orders in the FO theory of the nested pushdown tree

5. Set Quantification



Definition ("$b \overset{=\exp(n)}{\rightrightarrows} p$ equals $b' \overset{=\exp(n)}{\rightrightarrows} p'$")

$$\varphi_0(b, p, b', p') := (b \rightarrow p \wedge b' \rightarrow p') \vee (b \hookrightarrow p \wedge b' \hookrightarrow p')$$

$$\varphi_{n+1}(b, p, b', p') := \exists c, c' \ \ (\varphi_n(c, p, c', p') \wedge \varphi_n(b, c, b', c'))$$

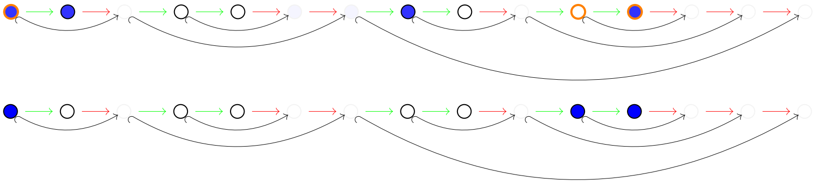
Lemma

$\exists X \quad \longrightarrow \quad \exists p'$

$x \in X \quad \longrightarrow \quad \exists x' \varphi_n((x, p, x', p') \wedge \varphi^P(x')$

interprets set quantification on linear orders in the FO theory of the nested pushdown tree

5. Set Quantification



Definition (“$b \overset{=\exp(n)}{\underset{\hookrightarrow}{\rightarrow}} p$ equals $b' \overset{=\exp(n)}{\underset{\hookrightarrow}{\rightarrow}} p'$”)

$$\varphi_0(b,p,b',p') := (b \rightarrow p \wedge b' \rightarrow p') \vee (b \hookrightarrow p \wedge b' \hookrightarrow p')$$

$$\varphi_{n+1}(b,p,b',p') := \exists c, c' \ \ (\varphi_n(c,p,c',p') \wedge \varphi_n(b,c,b',c'))$$

Lemma

$\exists X \quad \longrightarrow \quad \exists p'$

$x \in X \quad \longrightarrow \quad \exists x' \varphi_n((x,p,x',p') \wedge \varphi^P(x')$

interprets set quantification on linear orders in the FO theory of the nested pushdown tree

Theorem (Kartzow)

FO model checking on NPT: ATIME$(\exp_2(cn), cn)$*-complete (with respect to reset-loglin-reductions)*

Hardness Proof.

Take pushdown system that

- nondeterministically pushs and pops, and
- nondeterministically chooses state r and q.

$\exists$ reset-loglin computable MSO-to-FO-interpretation

- $\delta_n(x, p) \quad := \quad x \overset{=\exp(n)}{\hookrightarrow} p$ defines $\exp_2(n)$ ancestors of p
- $\varphi_n^{\leq}(b_1, b_2, p)$ defines b_1 proper ancestor of b_2
- $\varphi^P(x) \quad := \quad \mathrm{state}(x) = r$ defines predicate P
- $\varphi_n((x, p, x', p') \wedge \varphi^P(x')$ reduces $\exists X$ to $\exists p'$

$\square$

Summary

- Nested pushdown trees: models for verification of pre-/post-conditions of function calls
- MSO model checking: undecidable :(
- $L\mu$ and FO model checking: decidable :)
- FO model checking: $\mathrm{ATIME}(\exp_2(cn), cn)$-complete
- Hardness: interpret long linear orders in nested pushdown tree
    - $\exp_2(n)$ many ancestors definable with linear FO formula

Possible Future Work

Decidability of $L\mu$ / FO on *higher-order* nested pushdown trees