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: Programme flow
  - Property of Programme: Formalised in MSO
  - Check whether program flow satisfies property via MSO model checking on the pushdown tree

Pushdown system $S$

Definition (Pushdown Tree)

Pushdown tree:
- domain: all runs of $S$ (from initial configuration).
- $\delta$-labelled edges: extension of run by transition $\delta$
From Pushdown to Nested Pushdown

\[ \mathcal{C} \text{ class of structures, } \mathcal{L} \text{ logic} \]

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

Input: \( \mathcal{G} \in \mathcal{C}, \varphi \in \mathcal{L} \)

Output: \( \mathcal{G} \models \varphi \)

**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
From Pushdown to Nested Pushdown

\( \mathcal{C} \) class of structures, \( \mathcal{L} \) logic

\( \mathcal{L} \)-Model Checking on \( \mathcal{C} \)

Input: \( \mathcal{G} \in \mathcal{C}, \varphi \in \mathcal{L} \)
Output: \( \mathcal{G} \models \varphi \)

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

- $r, a$
- $r, aa$
- $r, a$
- $r, aa$
- $r, aaa$
- $q, a$
- $q, aa$
- $q, aaa$
- $q, a^4$
- ...
Example of NPT

Grid is MSO-definable
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: $\text{ATIME}(\exp_2(cn), cn)$

Proof Idea.

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

$\square$
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 \( \text{ATIME}(\exp_2(cn), cn) \) lower bound.

Corollary
\( (\mathcal{L}_n)_{n \in \mathbb{N}} \stackrel{\text{reset-loglin MSO-to-FO}}{\longrightarrow} \{ \mathcal{A} \} \)
\Rightarrow \text{FO-theory of } \mathcal{A} \text{ is } \text{ATIME}(\exp_2(cn), cn)\text{-hard.}
Definition (linear recursive definitions)

\((\varphi_n)_{n \in \mathbb{N}}\) is defined by linear recursion:

\[ \varphi_{n+1} = \exists x_1 \forall x_2, \ldots \forall x_c.n(\psi \rightarrow \varphi_n) \]

Properties of linear recursive definitions

Unfolding of \(\varphi_n\): formula of size \(c \cdot n\)

Lemma (Compton and Henson)

\((\varphi_n)_{n \in \mathbb{N}} \text{ defined using linear recursion} \Rightarrow n \mapsto \varphi_n \text{ is reset-loglin computable}\)
Large Linear Orders in Nested Pushdown Trees

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

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

\[
\begin{align*}
  a \xrightarrow{=1} b & := a \xrightarrow{} b \lor a \xrightarrow{} b \\
  a \xrightarrow{=\exp(n)} b & := \exists c (a \xrightarrow{=\exp(n-1)} c) \land (c \xrightarrow{=\exp(n-1)} b)
\end{align*}
\]
1: Paths of length $\exp(n)$

Paths along jump and pop edges

$$a \overset{=1}{\leftrightarrow} b := a \leftrightarrow b \lor a \rightarrow b$$

$$a \overset{=\exp(n)}{\leftrightarrow} b := \exists c \forall x, y \ ((x, y) = (a, c) \lor (x, y) = (c, b))$$

$$\overset{=\exp(n-1)}{\rightarrow} x \overset{\rightarrow}{\leftrightarrow} y$$
1: Paths of length $\exp(n)$

Paths along jump and pop edges

\[ a \overset{1}{\leftrightarrow} b := a \leftrightarrow b \lor a \rightarrow b \]
\[ a \overset{\exp(n)}{\leftrightarrow} b := \exists c \forall x, y \left((x, y) = (a, c) \lor (x, y) = (c, b)\right) \]
\[ \overset{\exp(n-1)}{\rightarrow} x \overset{\leftrightarrow}{\leftrightarrow} y \]

Analogously:

\[ a \overset{\leq 1}{\leftrightarrow} b := a \leftrightarrow b \lor a \rightarrow b \lor a = b \]
\[ a \overset{\leq \exp(n)}{\leftrightarrow} b := \exists c \forall x, y \left((x, y) = (a, c) \lor (x, y) = (c, b)\right) \]
\[ \overset{\leq \exp(n-1)}{\rightarrow} x \overset{\leftrightarrow}{\leftrightarrow} y \]
2: Many Ancestors in generic Nested Pushdown Tree

Nested Pushdown Tree with arbitrary Push / Pop Sequences

\[ \{|x : x \xrightarrow{\text{=1}} p}\} = \exp(1) \]
2: Many Ancestors in generic Nested Pushdown Tree

Nested Pushdown Tree with arbitrary Push / Pop Sequences

\[ |\{ x : x \xrightarrow{2} p \} | = \exp(2) \]
2: Many Ancestors in 
generic Nested Pushdown Tree

Nested Pushdown Tree with arbitrary Push / Pop Sequences

$$|\{x : x \overset{=3}{\Rightarrow} p\}| = \exp(3)$$
2: Many Ancestors in generic Nested Pushdown Tree

Nested Pushdown Tree with arbitrary Push / Pop Sequences

\[ |\{x : x \overset{3}{\rightarrow} p\}| = \exp(3) \]

\[ \mathcal{G} = \exp(3) \]

General rule: \[ |\{x : x \overset{\exp(n)}{\rightarrow} p\}| = \exp_2(n) \]

Definition

\[ \delta_n(x, p) := x \overset{\exp(n)}{\rightarrow} p \]

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

Let \( b_1 \leq \exp(n) \xrightarrow{\preceq} p \) and \( b_2 \leq \exp(n) \xrightarrow{\preceq} p \)

\( b_1 \) proper ancestor of \( b_2 \) \iff \( \varphi_n^\leq(b_1, b_2, p) \) holds

\[
\exists c, d, e \quad c \xrightarrow{\leq \exp(n)} p \land d \rightarrow c \land e \rightarrow c \land b_2 \xrightarrow{\leq \exp(n)} e \land b_1 \xrightarrow{\leq \exp(n)} d
\]
3: Order using linear sized formula

Lemma

Let $b_1 \xrightarrow{\leq \exp(n)} p$ and $b_2 \xrightarrow{\leq \exp(n)} p$

$b_1$ proper ancestor of $b_2 \iff \varphi_n(b_1, b_2, p)$ holds

$$\exists c, d, e \quad c \xrightarrow{\leq \exp(n)} p \land d \rightarrow c \land e \rightarrow c \land b_2 \xrightarrow{\leq \exp(n)} e \land b_1 \xrightarrow{\leq \exp(n)} d$$
Lemma

Let \( b_1 \xrightarrow{\leq \exp(n)} p \) and \( b_2 \xrightarrow{\leq \exp(n)} p \)

\( b_1 \) proper ancestor of \( b_2 \) \iff \( \varphi_n^\leq(b_1, b_2, p) \) holds

\[ \exists c, d, e \quad c \xrightarrow{\leq \exp(n)} p \wedge d \rightarrow c \wedge e \rightarrow c \wedge b_2 \xrightarrow{\leq \exp(n)} e \wedge b_1 \xrightarrow{\leq \exp(n)} d \]

Proof.

\( b_2 \xrightarrow{*} d \rightarrow c \): stacks between \( b_1 \) and \( d \) greater than stack of \( c \)

\( b_1 \xrightarrow{*} e \rightarrow c \): stack of \( e \) equals stack of \( c \)

\( e \) proper ancestor of \( c \) \Rightarrow \( e \) ancestor of \( d \) \Rightarrow \( e \) ancestor of \( b_2 \) \qed
3: Order using linear sized formula

Lemma

Let \( b_1 \overset{\leq \exp(n)}{\leftrightarrow} p \) and \( b_2 \overset{\leq \exp(n)}{\leftrightarrow} p \)

\( b_1 \) proper ancestor of \( b_2 \) \( \iff \varphi_n^{\leq}(b_1, b_2, p) \) holds

\[
\exists c, d, e \quad c \overset{\leq \exp(n)}{\leftrightarrow} p \land d \overset{\leftrightarrow}{\to} c \land e \overset{\leftrightarrow}{\to} c \land b_2 \overset{\leq \exp(n)}{\leftrightarrow} e \land b_1 \overset{\leq \exp(n)}{\leftrightarrow} d
\]

Corollary

\( \overset{\leftrightarrow}{*} \)-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
\[ \varphi^P(x) \ := \ \text{state}(x) = r \]

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

Definition (Ord(\( p \))
\[ \text{Ord}(p) \ := \ \text{linear order with predicate } P \text{ obtained using} \]
\( (\delta_n, \varphi \leq n, \varphi^P) \) and parameter \( p \)
5. Set Quantification

Definition ("$b \xrightarrow{\exp(n)} p$ equals $b' \xrightarrow{\exp(n)} p'$")

$$
\varphi_0(b, p, b', p') := (b \rightarrow p \land b' \rightarrow p') \lor (b \leftarrow p \land b' \leftarrow p')
$$

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

Lemma

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

interprets set quantification on linear orders in the FO theory of the nested pushdown tree
5. Set Quantification

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

\[
\varphi_0(b, p, b', p') := (b \rightarrow p \land b' \rightarrow p') \lor (b \leftarrow p \land b' \leftarrow p')
\]

\[
\varphi_{n+1}(b, p, b', p') := \exists c, c' \left( \varphi_n(c, p, c', p') \land \varphi_n(b, c, b', c') \right)
\]

Lemma

\[
\exists X \rightarrow \exists p' \\quad x \in X \rightarrow \exists x' \varphi_n((x, p, x', p') \land \varphi_P(x'))
\]

interprets set quantification on linear orders in the FO theory of the nested pushdown tree
5. Set Quantification

Definition ("$b \xrightarrow{\text{exp}(n)} \rightarrow p$ equals $b' \xrightarrow{\text{exp}(n)} \rightarrow p'$")

$\varphi_0(b, p, b', p') := (b \rightarrow p \land b' \rightarrow p') \lor (b \leftarrow p \land b' \leftarrow p')$

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

Lemma

$\exists X \rightarrow \exists p'$

$x \in X \rightarrow \exists x' \varphi_n((x, p, x', p') \land \varphi^P(x'))$

interprets set quantification on linear orders in the FO theory of the nested pushdown tree
5. Set Quantification

Definition ("\( b \overset{\text{exp}(n)}{\leftrightarrow} p \) equals \( b' \overset{\text{exp}(n)}{\leftrightarrow} p' \")

\[
\varphi_0(b, p, b', p') := (b \rightarrow p \land b' \rightarrow p') \lor (b \leftarrow p \land b' \leftarrow p')
\]

\[
\varphi_{n+1}(b, p, b', p') := \exists c, c' \ (\varphi_n(c, p, c', p') \land \varphi_n(b, c, b', c'))
\]

Lemma

\[
\exists X \rightarrow \exists p' \quad x \in X \rightarrow \exists x' \varphi_n((x, p, x', p') \land \varphi^p(x'))
\]

interprets set quantification on linear orders in the FO theory of the nested pushdown tree
Theorem (Kartzow)

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

**Hardness Proof.**

Take pushdown system that

- nondeterministically pushes and pops, and
- nondeterministically chooses state $r$ and $q$.

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

- $\delta_n(x, p) := x \overset{=\exp(n)}{\rightarrow} p$ defines $\exp_2(n)$ ancestors of $p$
- $\varphi_n^\leq(b_1, b_2, p)$ defines $b_1$ proper ancestor of $b_2$
- $\varphi_P^P(x) := \text{state}(x) = r$ defines predicate $P$
- $\varphi_n((x, p, x', p') \land \varphi_P^P(x'))$ reduces $\exists X$ to $\exists p'$
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: $\text{ATIME}(\exp_2(cn), cn)$-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