Bounded Multipushdown Systems
Complexity analysis of temporal model-checking problems

Kshitij Bansal\textsuperscript{1}, Stéphane Demri\textsuperscript{1,2}

\textsuperscript{1}New York University \textsuperscript{2}CNRS

June 29, 2013
CSR, Yekaterinburg, Russia
Pushdown systems

\[ P = (Q, N, \Gamma, \Delta), \text{ such that:} \]

- \( Q \) is a non-empty finite set of states,
- \( \Gamma \) is the finite stack alphabet
- \( \Delta \) is the transition relation.

E.g. \( q_0 (a, \text{push}(b)) \rightarrow q_1 \in \Delta \)
Multipushdown systems

$P = (Q, N, \Gamma, \Delta_1, \ldots, \Delta_N)$, for some $N \geq 1$ such that:

- $Q$ is a non-empty finite set of states,
- $\Gamma$ is the finite stack alphabet
- for every $s \in [N]$, $\Delta_s$ is the transition relation acting on the $s$-th stack.

E.g. $q_0 \xrightarrow{(a,push(b))} q_1 \in \Delta_1$
**Configuration:** $c = (q, w_1, w_2, \ldots w_N)$.

**Run:** Sequence of configurations: $c^0 \rightarrow c^1 \rightarrow \ldots$ such that consecutive configurations respect transition relation.

**Example of run.**

\[
\begin{array}{cccccccccccc}
q_1 & q_2 & q_1 & q_1 & q_1 & q_1 & q_1 & q_1 & q_1 \\
\epsilon & \rightarrow_1 & a & \rightarrow_2 & a & \rightarrow_1 & ab & \rightarrow_1 & aa & \rightarrow_2 & aa & \rightarrow_1 & a & \rightarrow_1 & \epsilon \\
\epsilon & \epsilon & a & a & a & b & b & b & b
\end{array}
\]

Natural for modeling stack behavior of concurrent programs.
What do we know?

Problem (REACH)

input: a multipushdown system, an initial state $q_0$ and final configuration $c$

question: Is there a run from $(q_0, (\epsilon)^N) \rightarrow^* c$?

Theorem [Minsky'67]
With just two stacks, the reachability problem is undecidable.

Theorem e.g. [Schwoon '02]
With one stack, polynomial time procedure for reachability.

This talk
Restrictions of multipushdown systems in other directions to regain decidability and understand complexity.
Restrictions on runs

Divide runs into “phases”. Ask if there is a run with finite number of “phases” that satisfy.

- Context bounded (B-REACH): phase $\equiv$ all operations only on one stack

\[
\begin{bmatrix}
q_1 & q_2 \\
\epsilon & a & \epsilon & \epsilon
\end{bmatrix}
\begin{bmatrix}
q_1 & q_1 \\
a & ab & a & a
\end{bmatrix}
\begin{bmatrix}
q_1 & q_1 \\
aa & a & a & a
\end{bmatrix}
\begin{bmatrix}
q_1 & q_1 \\
a & b & b & b
\end{bmatrix}
\]

- Phase bounded (PB-REACH): all pop operations on one stack

- Ordered pushdown systems (OB-REACH): operations only on “first” non-empty stack

Note: runs can be infinite, more general than considering bound on length of runs
Restrictions on runs

Divide runs into “phases”. Ask if there is a run with finite number of “phases” that satisfy.

- **Context bounded (B-REACH):** phase $\equiv$ all operations only on one stack

- **Phase bounded (PB-REACH):** phase $\equiv$ all pop operations on one stack

- **Ordered pushdown systems (OB-REACH):** operations only on “first” non-empty stack

Note: runs can be infinite, more general than considering bound on length of runs
More expressive properties of runs

Problem (Repeated reachability)

input: \((P, q_0, F)\) : \(P\) multi-pushdown system, \(q_0 \in Q\) initial state, and \(F\) is a Büchi acceptance set.

question: \(\exists\) infinite run \(\rho\) from \((q_0, (\epsilon)^N)\) such that a \(q_f \in F\) is repeated infinitely often?
More expressive properties of runs

Problem (Repeated reachability)

input: \((P, q_0, F)\) : \(P\) multi-pushdown system, \(q_0 \in Q\) initial state, and \(F\) is a Büchi acceptance set.

question: \(\exists\) infinite run \(\rho\) from \((q_0, (\epsilon)^N)\) such that a \(q_f \in F\) is repeated infinitely often?

Problem (Model checking)

input: \((P, q_0, \phi)\) where \(P\) is a multi-pushdown system, \(q_0\) initial state, \(\phi\) a formula specifying property on runs,

question: \(\exists\) infinite run \(\rho\) from \((q_0, (\epsilon)^N)\) such that \(\rho \models \phi\)?
What kind of properties?

Properties

- Current state?
- Current thread?
- Current stack operation – push, pop etc.
What kind of properties?

Properties

- Current state?
- Current thread?
- Current stack operation – push, pop etc.
- Current stack contents? In particular, we consider regularity constraints. [Esparza, Kučera, S. Schwoon '01]
What kind of properties?

Properties

- Current state?
- Current thread?
- Current stack operation – push, pop etc.
- Current stack contents? In particular, we consider regularity constraints. [Esparza, Kučera, S. Schwoon '01] (given automata $A$ over $\Sigma$ and stack $s$: is the content of stack $s$ accepted by $A$).
What kind of properties?

Properties

- Current state?
- Current thread?
- Current stack operation – push, pop etc.
- Current stack contents? In particular, we consider regularity constraints. [Esparza, Kučera, S. Schwoon ’01] (given automata $A$ over $\Sigma$ and stack $s$: is the content of stack $s$ accepted by $A$).

When in the run?

- LTL operators: next, until, finally etc.
- Call and returns: “when the function returns”, “the function that called”. [Alur, Etessami, Madhusudan ’04]
  - parameterized by stack.
Multi-CaRet$^{reg}$: A logic of calls and returns for multiple stacks with regularity constraints

\[ \phi := q | s | \text{in}(s, \mathcal{A}) | \text{call} | \text{return} | \text{internal} \\
| X\phi | \phi U \phi | X^a \phi | \phi U^a \phi | X^c \phi | \phi U^c \phi \\
| \phi \lor \phi | \neg \phi \]

where \( s \in [N] \), \( q \in Q \).

\[ \rho, t \models q \] iff state at position \( t \) is \( q \)

\[ \rho, t \models s \] iff active thread at \( t \) is \( s \)

\[ \rho, t \models \text{in}(s, \mathcal{A}) \] iff \( w^t_s \in L(\mathcal{A}) \)

\[ \rho, t \models X^\text{abstract}_s \phi \] iff \( \rho, \text{Next}^\text{abstract}_s(t) \models \phi \)

\( \text{Next}^\text{abstract}_s(t) \) is defined as next position when the same stack active within the same call.
Example

Can be used to express other related operators, like: 

\[ \neg s \cup (s \wedge \phi) \] 
\[ \wedge s (s \Rightarrow X \alpha \phi) \] 
\[ \phi_1 \cup \alpha \phi_2 \equiv \wedge s (s \Rightarrow \phi_1 \cup \alpha s \phi_2) \]
Example

\( X_1^{\text{abstract}}: \text{undefined} \)

\[ X_1^{\text{abstract}} \text{ in}(1, A) \implies \text{in}(1, A) \]
Example

Can be used to express other related operators, like:

- $X_s \phi \equiv (\neg sU(s \land \phi))$,
- $X^a \phi \equiv (\land_s (s \Rightarrow X^a_s \phi))$,
- $\phi_1 U^a \phi_2 \equiv (\land_s (s \Rightarrow \phi_1 U^a_s \phi_2))$. 
### What was known?

<table>
<thead>
<tr>
<th></th>
<th>Reach</th>
<th>Rep</th>
<th>LTL</th>
<th>(Multi)CaRet</th>
</tr>
</thead>
<tbody>
<tr>
<td>One stack</td>
<td>P</td>
<td>P</td>
<td>EXP_TIME</td>
<td>EXP_TIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-complete</td>
<td>-complete</td>
</tr>
</tbody>
</table>
## What is now known?

<table>
<thead>
<tr>
<th></th>
<th>Reach</th>
<th>Rep</th>
<th>LTL</th>
<th>(Multi)CaRet</th>
</tr>
</thead>
<tbody>
<tr>
<td>One stack</td>
<td>P</td>
<td>P</td>
<td>EXPTime</td>
<td>EXPTime -complete</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXPTime -complete</td>
</tr>
<tr>
<td>k-context* bounded</td>
<td>NP</td>
<td>EXPTime</td>
<td>EXPTime</td>
<td>EXPTime (this paper)</td>
</tr>
<tr>
<td></td>
<td>-comp</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* *k* part of input, encoded in unary
## What is now known?

<table>
<thead>
<tr>
<th></th>
<th>Reach</th>
<th>Rep</th>
<th>LTL</th>
<th>(Multi)CaRet</th>
</tr>
</thead>
<tbody>
<tr>
<td>One stack</td>
<td>P</td>
<td>P</td>
<td>ExpTime</td>
<td>ExpTime</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-complete</td>
<td>-complete</td>
</tr>
<tr>
<td>k-context*</td>
<td>NP</td>
<td>ExpTime</td>
<td>ExpTime</td>
<td>ExpTime</td>
</tr>
<tr>
<td>bounded</td>
<td>-comp</td>
<td></td>
<td></td>
<td>(this paper)</td>
</tr>
</tbody>
</table>

\*k part of input, encoded in unary

**Step 1:**

**Step 2:**
Step 1: From Model-checking to Repeated reachability

\[ P( Q, \Gamma, \ldots ), \phi \]

\[ \hat{P}( \hat{Q}, \hat{\Gamma}, \ldots ), \mathcal{F} \]

\( P \) has a run satisfying \( \phi \) iff \( \hat{P} \) has a run with states in \( \mathcal{F} \) infinitely often. (Büchi acceptance)
Step 1: From Model-checking to Repeated reachability

\[ P( Q, \Gamma, \ldots ), \phi \]

\[ \hat{P}( Q \times \text{“extra”}, \Gamma \times \text{“extra”}, \ldots ), \mathcal{F} \]

\( P \) has a run satisfying \( \phi \) iff \( \hat{P} \) has a run with states in \( \mathcal{F} \) infinitely often. (Büchi acceptance)

How?

- Simulate original system
- Keep extra information to verify satisfaction of formula.
Step 1: From Model-checking to Repeated reachability

\[ P( Q, \Gamma, \ldots ), \phi \]
\[ \hat{P}( Q \times \text{"extra"}, \Gamma \times \text{"extra"}, \ldots ), F \]

\( P \) has a run satisfying \( \phi \) iff \( \hat{P} \) has a run with states in \( F \) infinitely often. (Büchi acceptance)

How?
- Simulate original system
- Keep extra information to verify satisfaction of formula.

What all information?
Step 1: From MC to REP: high-level details

In particular we track,

1. In state, about truth value of each subformula (Fischer-Ladner closure for LTL): but for each stack separately.
2. In stack, information about call and returns (CaRet).
3. In state, whether stack will ever be active again.
4. In state and stack, automata for satisfaction of regularity constraint.
Step 1: From MC to REP: **Upshot**

\[ MC(P, \phi) \quad \text{size:} \quad |P| + |\phi| \]
\[ \exists \rho \models \phi \iff \exists \text{accepting } \hat{\rho} \]

Operation on stacks, stack order as well as actions are preserved:

\[ k\text{-bounded} \iff k\text{-bounded} \]
\[ k\text{-phase bounded} \iff k\text{-phase-bounded} \]
\[ \preceq\text{-bounded} \iff \preceq\text{-bounded} \]
Step 2: Bounded repeated reachability: from $N$ to 1 stacks

**Lemma**

*BREP* can be solved in time $\mathcal{O}(|P|^{k+1} \times p(k, |P|))$ for some polynomial $p(\cdot, \cdot)$.

At heart of complexity analysis of this step, two main ideas:

- For a pushdown system $P$, $\text{post}^*$ (regular set of configurations) is regular. Also: can be done polynomial time, size of new automata only additive in $|P|$. [Bouajjani, Esparza, Maler '97]

- Repeated reachability for 1-stack is in $P$. [Boujanni, Esparza, Maler '97]

Connect above two ideas by guessing context-switches and intermediate states.
## Other restrictions

<table>
<thead>
<tr>
<th></th>
<th>Rep Reach</th>
<th>Multi-CaRetMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>One stack</td>
<td>P</td>
<td>\textsc{ExpTime} (CaRet)</td>
</tr>
<tr>
<td>k-Context bounded*</td>
<td>\textsc{ExpTime} -complete</td>
<td>\textsc{ExpTime} -complete</td>
</tr>
<tr>
<td>k-Phase bounded*</td>
<td>2-\textsc{ExpTime}</td>
<td>2-\textsc{ExpTime}</td>
</tr>
<tr>
<td>Order-bounded</td>
<td>2-\textsc{ExpTime} (2\textsc{ETIME}-hard)</td>
<td>2-\textsc{ExpTime} (2\textsc{ETIME}-hard)</td>
</tr>
</tbody>
</table>

*k part of input, encoded in unary. For binary encoding, 2\textsc{ExpTime} (resp. 3\textsc{ExpTime}) upper bound for context-bounded (resp. phase-bounded) MC.

**Step 1:** same analysis

**Step 2:** [Atig, Bollig, Habermehl ’08] for PBMC, [Atig ’10] for OBMP.
M. Atig, A. Bouajjani, K. N. Kumar, and P. Saivasan. Linear-time model-checking for multithreaded programs under scope-bounding. In ATVA’12, volume 7561 of LNCS, Springer.

S. La Torre and M. Napoli. A temporal logic for multi-threaded programs. In TCS’12, volume 7604 of LNCS, Springer.

P. Madhusudan and G. Parlato. The tree width of auxiliary storage. In POPL’11, pages 283–294. ACM.

Conclusion

- \textbf{ExpTime} bound for $k$-bounded runs, Multi-CaRet operators, regularity constraints on stacks. Matches hardness result for single-stack LTL model checking.
- Approaches to unification
  - Classes of notions of boundedness (e.g., tree-width bounded).
  - Classes of logical operators (e.g., specifiable using classes of MSO formulas).