Resource-Bounded Reachability on Pushdown Graphs

Martin Lang & Christof Löding

Lehrstuhl für Informatik 7
RWTH Aachen University

10. September 2011
1 Resource Pushdown Systems & Graphs

2 Reachability Analysis

3 Resource Reachability Games

4 Conclusion
Resource Pushdown Systems & Graphs
Reminder: Pushdown Systems

- A pushdown system is a pushdown automaton without input.
  - $Q$ a finite set of states
  - $\Gamma$ a finite alphabet
  - $\Delta$ a finite set of transitions
- A configuration is of the form $(p, w)$
Reminder: Pushdown Systems

- A pushdown system is a pushdown automaton without input.
  - Q a finite set of states
  - Γ a finite alphabet
  - Δ a finite set of transitions

- A configuration is of the form \((p, w)\)

- A transition is of the form \(pa \rightarrow qbaa\)
Reminder: Pushdown Systems

- A pushdown system is a pushdown automaton without input.
  - $Q$ a finite set of states
  - $\Gamma$ a finite alphabet
  - $\Delta$ a finite set of transitions
- A configuration is of the form $(p, w)$
- A transition is of the form $pa \rightarrow qbaa$
Reminder: Pushdown Systems

- A pushdown system is a pushdown automaton without input.
  - $Q$ a finite set of states
  - $\Gamma$ a finite alphabet
  - $\Delta$ a finite set of transitions
- A configuration is of the form $(p, w)$
- A transition is of the form $pa \rightarrow qbaa$
- Can be used to model recursive programs
Resource Extension of Pushdown Systems

- Idea: Model recursive programs using integer resources
e.g. paper, I/O capacity, energy, ...
Resource Extension of Pushdown Systems

- Idea: Model recursive programs using integer resources e.g. paper, I/O capacity, energy, ...
- Resources are tokens which are consumed step-by-step or completely refreshed at once (refill)
Resource Extension of Pushdown Systems

- Idea: Model recursive programs using integer resources e.g. paper, I/O capacity, energy, ...
- Resources are tokens which are consumed step-by-step or completely refreshed at once (refill)
- Resources are represented by integer counters and three operations per resource
  (i) $n$ - no resource used / leave counter unchanged
  (ii) $i$ - increment resource usage
  (iii) $r$ - reset resource / set the resource counter to zero
Resource Extension of Pushdown Systems

- Idea: Model recursive programs using integer resources e.g. paper, I/O capacity, energy, ...
- Resources are tokens which are consumed step-by-step or completely refreshed at once (refill)
- Resources are represented by integer counters and three operations per resource
  (i) $n$ - no resource used / leave counter unchanged
  (ii) $i$ - increment resource usage
  (iii) $r$ - reset resource / set the resource counter to zero
- We annotate pushdown rules with the resource operation:

  $$pa \xrightarrow{n} qbaa \quad \text{or} \quad pa \xrightarrow{i} qbaa \quad \text{or} \quad pa \xrightarrow{r} qbaa$$
Resource Pushdown Graphs

Consider the configuration graph of a resource pushdown system:

\[ Q = \{ p \}, \Gamma = \{ a, b \} \]

\[
(p, \varepsilon) \rightarrow (p, a) \rightarrow (p, aa) \rightarrow (p, aaa) \rightarrow \cdots
\]

\[
(p, \varepsilon) \rightarrow (p, b) \rightarrow (p, ab) \rightarrow (p, aab) \rightarrow \cdots
\]

\[
(p, \varepsilon) \rightarrow (p, \varepsilon) \rightarrow (p, \varepsilon) \rightarrow (p, \varepsilon) \rightarrow \cdots
\]
Resource Pushdown Graphs

Consider the configuration graph of a resource pushdown system
\[ Q = \{ p \}, \quad \Gamma = \{ a, b \} \]
Resource Pushdown Graphs

Consider the configuration graph of a resource pushdown system

\[ Q = \{ p \}, \Gamma = \{ a, b \} \]
Resource Pushdown Graphs

Consider the configuration graph of a resource pushdown system

\[ Q = \{ p \}, \Gamma = \{ a, b \} \]
Resource Pushdown Graphs

A run of a program is associated with a path in the graph
Resource Pushdown Graphs

How much resources are needed for reachability?
Resource Pushdown Graphs

Resourced needed for reachability are not unique!

\[
\begin{align*}
(p, a) & \rightarrow i (p, aa) \\
(p, ba) & \rightarrow n (p, baa) \\
(p, ab) & \rightarrow i (p, aba) \\
(p, bb) & \rightarrow n (p, bba) \\
(p, ε) & \rightarrow r (p, ε) \\
(p, ε) & \rightarrow i (p, ε) \\
(p, ε) & \rightarrow r (p, ε) \\
\end{align*}
\]
Resource Pushdown Graphs

Resourced needed for reachability are not unique!

\[ (p, \varepsilon) \xrightarrow{i} (p, a) \]
\[ (p, a) \xrightarrow{i} (p, aa) \]
\[ (p, b) \xrightarrow{i} (p, ab) \]
\[ (p, \varepsilon) \xrightarrow{r} (p, b) \]
\[ (p, b) \xrightarrow{r} (p, bb) \]
\[ (p, aa) \xrightarrow{r} (p, baa) \]
\[ (p, aba) \xrightarrow{r} (p, bba) \]
\[ (p, aab) \xrightarrow{r} (p, bab) \]
\[ (p, aab) \xrightarrow{r} (p, abbb) \]
Bounded Reachability

$Q \times \Gamma^*$

$\begin{array}{c}
(p, w) & (q, w) \\
\vdots & A \\
(q, x) & (p, u) \\
(q, wx) & (q, v) \\
(q, uw) & (q, \epsilon) \\
\end{array}$
Bounded Reachability

\[ Q \times \Gamma^* \]

\[ (q, wx) \quad (q, v) \quad (q, w) \]

\[ (p, w) \quad (q, w) \]

\[ (p, u) \]

\[ (q, x) \]

\[ (p, xx) \quad (p, u) \]

\[ A \]

\[ \forall a \in A \exists b \in B : a \vdash^* b \]

\[ B \]

\[ (p, x) \]

\[ (p, x) \quad (p, y) \]

\[ (p, x) \quad (p, wy) \]

\[ (q, u) \]

\[ (q, y) \]

\[ (q, \varepsilon) \]

\[ (q, uw) \]

\[ (p, \varepsilon) \]

\[ (q, \varepsilon) \]
Bounded Reachability

\[ Q \times \Gamma^* \]

- \((q, wx)\)
- \((q, v)\)
- \((p, w)\)
- \((q, w)\)
- \((p, u)\)
- \((q, x)\)
- \((p, xx)\)
- \((q, uw)\) \(\leq k\)
- \((p, \varepsilon)\)
- \((q, y)\)
- \((q, u)\) \(\leq k\)
- \((p, x)\)
- \((q, \varepsilon)\)
- \((p, wy)\)

\[ \exists k \in \mathbb{N} \ \forall a \in A \ \exists b \in B : a \vdash^*_{\leq k} b \]

M. Lang (RWTH Aachen)
Bounded Reachability - Example

Reconsider the previous example:

\[ A = \{(p, a^n) \mid n > 0\} \]

\[ B = \{(p, \varepsilon)\} \]

\[ p.a \rightarrow_i p.\varepsilon \]

\[ p.b \rightarrow_r p.\varepsilon \]

\[ p.a \rightarrow_n p.\varepsilon \]
Bounded Reachability - Example

$B$ is boundedly reachable from $A$ (with resource-cost 0)

$A = \{(p, a^n) \mid n > 0\}$

$B = \{(p, \varepsilon)\}$
Bounded Reachability - Example

The resource-costs are unbounded.

\[ A = \{(p, a^n) \mid n > 0\} \]
\[ B = \{(p, \varepsilon)\} \]
Reachability Analysis
Saturation Procedure for pre\(^*(B)\)

**Theorem (Book, Otto 1993; implicitly Benois, Sakarovitch 1986)**

The set of pushdown-predecessor configurations of a regular set \(B\) can be computed by subsequently adding transitions to an automaton recognizing \(B\) (saturation).

---

The set of pushdown-predecessor configurations of a regular set \(B\) can be computed by subsequently adding transitions to an automaton recognizing \(B\) (saturation).
Saturation Procedure for pre*(B)

Theorem (Book, Otto 1993; implicitly Benois, Sakarovitch 1986)

The set of pushdown-predecessor configurations of a regular set B can be computed by subsequently adding transitions to an automaton recognizing B (saturation).

Idea:
Consider the pushdown system with one state p, Γ = \{a, b\} and one pushdown rule pb → paa.
Let B = \{(p, aa)\}.
Saturation Procedure for pre*(B)

Theorem (Book, Otto 1993; implicitly Benois, Sakarovitch 1986)

The set of pushdown-predecessor configurations of a regular set B can be computed by subsequently adding transitions to an automaton recognizing B (saturation).

Idea:
Consider the pushdown system with one state p, Γ = \{a, b\} and one pushdown rule \(pb \rightarrow paa\).
Let \(B = \{(p, aa)\}\).

![Diagram](image-url)
Saturation Procedure for $\text{pre}^*(B)$

**Theorem (Book, Otto 1993; implicitly Benois, Sakarovitch 1986)**

The set of pushdown-predecessor configurations of a regular set $B$ can be computed by subsequently adding transitions to an automaton recognizing $B$ (saturation).

Idea:
Consider the pushdown system with one state $p$, $\Gamma = \{a, b\}$ and one pushdown rule $pb \rightarrow paa$.
Let $B = \{(p, aa)\}$.

\[ \text{pre}^*(B) = \{(p, aa), (p, b)\} \]
Resource Automata

Idea: Finite automata with resource counters similar to resource pushdown systems

\[ A : \]

\[ q_0 \quad b : n \quad q_1 \quad a : i \quad b : n \quad q_2 \]

A word which starts and ends with \( b \) is mapped to the maximal number of \( a \)'s in a sequence. Other words are mapped to \( \infty \).
Resource Automata

Idea: Finite automata with resource counters similar to resource pushdown systems

\[ \mathcal{A} : \]

\[ [\mathcal{A}] : \Sigma^* \rightarrow \mathbb{N} \cup \{\infty\}. \]

A word which starts and ends with \( b \) is mapped to the maximal number of \( a \)'s in a sequence. Other words are mapped to \( \infty \).
Resource Automata - Brief Overview

- Concept was introduced by Hashiguchi in 1982 in the context of the star height problem.
  Called distance automata
- Further development by Kirsten in 2004.
  Model of nested distance desert automata
- Recent, very flexible framework by Colcombet in 2009.
  Models of B-automata and S-automata
Bounded Reachability - Solution Idea

Reconsider the previous saturation example:
The resource pushdown system consists of one state $p$, $\Gamma = \{a, b\}$ and one replacement rule $pb_i \rightarrow paa$

![Diagram of states and transitions](image)
Bounded Reachability - Solution Idea

Reconsider the previous saturation example:
The resource pushdown system consists of one state $p$, $\Gamma = \{a, b\}$ and one replacement rule $pb_1 \rightarrow paa$.
Bounded Reachability - Solution Idea

Reconsider the previous saturation example:
The resource pushdown system consists of one state \( p \), \( \Gamma = \{a, b\} \) and one replacement rule \( pb \xrightarrow{i} paa \)

\[
\begin{align*}
\text{Theorem} & \quad B^* \text{ be the result of the saturation procedure for a regular set } B. \\
\text{A state } (p, w) & \in \text{pre}^* (B) \text{ reaches a state in } B \text{ with resource-cost at most } k \\
& \iff J_{B^*} K((p, w)) \leq k.
\end{align*}
\]
Bounded Reachability - Solution Idea

Reconsider the previous saturation example:
The resource pushdown system consists of one state $p$, $\Gamma = \{a, b\}$ and one replacement rule $pb_1 \rightarrow pa$.

Theorem

Let $B^*$ be the result of the saturation procedure for a regular set $B$.
A state $(p, w) \in \text{pre}^*(B)$ reaches a state in $B$ with resource-cost at most $k$ if and only if $\lceil B^* \rceil((p, w)) \leq k$. 

Diagram:

\[\begin{array}{ccc}
  p & \xrightarrow{a:n} & q_1 & \xrightarrow{a:n} & q_2 \\
  & & b:inn & & \\
\end{array}\]
Theorem (Hashiguchi, Kirsten, Colcombet)

It is decidable whether there is a \( k \in \mathbb{N} \) such that

\[
\{ w \in \Sigma^* \mid [\mathcal{A}](w) < \infty \} = \{ w \in \Sigma^* \mid [\mathcal{A}](w) \leq k \}
\]
Boundedness Problem & Further Results

Theorem (Hashiguchi, Kirsten, Colcombet)

*It is decidable whether there is a* $k \in \mathbb{N}$ *such that*

$$\{ w \in \Sigma^* \mid \mathbb{K}(w) < \infty \} = \{ w \in \Sigma^* \mid \mathbb{K}(w) \leq k \}$$

**Theorem**

*It is decidable whether there is a* $k \in \mathbb{N}$ *such that all configurations* $(p, w) \in \text{pre}^*(B)$ *can reach a configuration in* $B$ *with a resource-cost bound of* $k$. 
Boundedness Problem & Further Results

Theorem (Hashiguchi, Kirsten, Colcombet)

*It is decidable whether there is a* \( k \in \mathbb{N} \) *such that*

\[
\left\{ w \in \Sigma^* \mid \ulcorner A \urcorner(w) < \infty \right\} = \left\{ w \in \Sigma^* \mid \ulcorner A \urcorner(w) \leq k \right\}
\]

Theorem

*It is decidable whether there is a* \( k \in \mathbb{N} \) *such that all configurations* \((p, w) \in \text{pre}^*(B)\) *can reach a configuration in* \( B \) *with a resource-cost bound of* \( k \).

Further result (even possible on prefix replacement systems):

Theorem

*It is possible to construct a synchronous transducer* \( A \) *with resource counters such that*

\[
\ulcorner A \urcorner((u, v)) \leq k \iff u \vdash^*_k v
\]
Resource Reachability Games
Resource Pushdown Reachability Game

- Idea: Resource reachability game (with one counter)

Eve has a resource limit $k$ to win the game.
Resource Pushdown Reachability Game

- Idea: Resource reachability game (with one counter)

Eve has a resource limit $k$ to win the game.
(Positional) Determinacy

Determinacy?

\[\text{No Resets: The resources needed to win the game can be computed by an adapted attractor computation. Both players have positional strategies on their winning regions.}\]
(Positional) Determinacy

How much memory is needed to win the game?

No Resets: The resources needed to win the game can be computed by an adapted attractor computation. Both players have positional strategies on their winning regions.
(Positional) Determinacy

How much memory is needed to win the game?

With Resets: Memory is needed to win the game.
Bounded Winning Strategies

Winning region of Eve when ignoring resources

\[ W_0^i = \{ v \in V \mid \text{Eve wins with resource-cost at most } i \} \]

Problem

Is there a \( k \in \mathbb{N} \) such that \( W_0^k = W_0 \)?
Resource Pushdown Reachability Game

- Idea: Resource reachability game (with one counter)
- Interest: Is it possible to win with bounded resources?

- Play this game on resource pushdown graphs without resets
  Partition the state space $P = P_E \cup P_A$
  Eve moves at configuration with state in $P_E$, and Adam accordingly at configurations with $P_A$ state.
Computing Winning Strategies

Theorem (Cachat 2002)

The winning region of reachability games on pushdown graphs with regular goal set is computable with a saturation procedure using alternating automata.

Theorem (Colcombet, Löding 2008)

For alternating tree automata with counters, it is decidable whether there is a $k \in \mathbb{N}$ such that

$\{ t \in T \mid J_{A}K(t) < \infty \} = \{ t \in T \mid J_{A}K(t) \leq k \}$
Computing Winning Strategies

**Theorem (Cachat 2002)**

The winning region of reachability games on pushdown graphs with regular goal set is computable with a saturation procedure using alternating automata.

**Theorem**

For every regular goal set $F$, there is a saturation procedure, which uses alternating resource automata, with result $\mathcal{A}^*$ such that

$$\left[\mathcal{A}^*\right]\left((p, w)\right) \leq k \iff (p, w) \in W_0^k(F)$$
Computing Winning Strategies

**Theorem (Cachat 2002)**

The winning region of reachability games on pushdown graphs with regular goal set is computable with a saturation procedure using alternating automata.

**Theorem**

For every regular goal set $F$, there is a saturation procedure, which uses alternating resource automata, with result $A^*$ such that

$$[A^*][(p, w)) \leq k \Leftrightarrow (p, w) \in W^k_0(F)$$

**Theorem (Colcombet, Löding 2008)**

For alternating tree automata with counters, it is decidable whether there is a $k \in \mathbb{N}$ such that

$$\{t \in T \mid [A](t) < \infty\} = \{t \in T \mid [A](t) \leq k\}$$
Conclusion
Conclusion

- Resource extensions of pushdown systems:
  Pushdown system + finite set of counters with operations $i, r, n$
- Motivation: Model recursive programs with resource consumption
Conclusion

- Resource extensions of pushdown systems:
  Pushdown system + finite set of counters with operations $i, r, n$
- Motivation: Model recursive programs with resource consumption
- Main question: Is there a resource-bound for reachability?
- Solutions for regular sets of interest on:
  - Prefix replacement systems with one counter
  - Reachability games without reset

Future work:
- Extend results to several counters
- Regular reachability
- Handle resets in games
- Temporal logic for specifications with resource bounds
Conclusion

- Resource extensions of pushdown systems:
  Pushdown system + finite set of counters with operations $i, r, n$
- Motivation: Model recursive programs with resource consumption
- Main question: Is there a resource-bound for reachability?
- Solutions for regular sets of interest on:
  - Prefix replacement systems with one counter
  - Reachability games without reset
- Specialized logic $\text{FO} + \text{RR}$:
  \[
  \forall a \in A \exists b \in B : a \xrightarrow{*} b |_k
  \]
Conclusion

- Resource extensions of pushdown systems:
  - Pushdown system + finite set of counters with operations $i, r, n$
- Motivation: Model recursive programs with resource consumption
- Main question: Is there a resource-bound for reachability?
- Solutions for regular sets of interest on:
  - Prefix replacement systems with one counter
  - Reachability games without reset
- Specialized logic $\text{FO}+\text{RR}$: $\forall a \in A \exists b \in B : a \rightarrow^* b\big|_k$
- Future work:
  - Extend results to several counters
  - Regular reachability
  - Handle resets in games
  - Temporal logic for specifications with resource bounds