A Pumping Lemma for Collapsible Pushdown Graphs of Level 2

Alexander Kartzow

Universität Leipzig

September 15, 2011
Collapsible Pushdown Systems (CPS)

- Higher-order pushdown systems (HOPS) [Maslov’76]
  - Pushdown systems with nested stack of ... of stacks
  - Operation: push / pop for each stack level

Motivation:

Theorem (Knapik, Niwinski, Urzyczyn ’02)

trees of HOPS = trees of safe higher-order recursion schemes
Collapsible Pushdown Systems (CPS)

- Higher-order pushdown systems (HOPS) [Maslov’76]
  - Pushdown systems with nested stack of . . . of stacks
  - Operation: push / pop for each stack level
- Collapsible pushdown system (CPS)
  Extension by “Collapse” operation
- defined by Hague, Murawski, Ong and Serre in ’08
- Motivation:

Theorem (Knapik, Niwinski, Urzyczyn ’02)

trees of HOPS = trees of safe higher-order recursion schemes

Theorem (Hague et al. ’08)

trees of CPS = trees of higher-order recursion schemes
**Theorem (Carayol, Wöhrle ’03)**

\[ \text{HOPG}/\varepsilon = \text{Cauca-l-hierarchy} \]

**Corollary**

*MSO decidable on HOPG*/\varepsilon

**Theorem (Model checking on CPG*/\varepsilon*)**

<table>
<thead>
<tr>
<th>Formalism</th>
<th>decidability</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSO</td>
<td>undecidable</td>
<td>(Hague et al. ’08)</td>
</tr>
<tr>
<td>(L_\mu)</td>
<td>decidable</td>
<td>(Hague et al. ’08)</td>
</tr>
<tr>
<td>(FO + \text{Reach})</td>
<td>decidable on level 2</td>
<td>(Kartzow ’10)</td>
</tr>
<tr>
<td>(FO)</td>
<td>undecidable on higher levels</td>
<td>(Broadbent)</td>
</tr>
</tbody>
</table>
We do not understand the structure of CPG
No tools for “upper bounds”:
  - $\emptyset$ is not a CPG?
  - $\emptyset$ is not a level $i$–CPG?
Possible tool: pumping lemma for level $i$ CPG
Today: pumping lemma for level 2 CPG
Stack Operations

\[
\sigma_1 \cdots \sigma_{m-1} \sigma_m \\
W_{m-1}
\]
\[
W_n \\
W_{n-1}
\]
\[
W_3 \\
W_2 \\
W_1
\]
Stack Operations

\[ \sigma_1 \cdots \sigma_{m-1} \sigma_m \]

\[ W_{m-1} \]

\[ \vdots \]

\[ W_n \]
\[ W_{n-1} \]

\[ \vdots \]

\[ W_3 \]
\[ W_2 \]
\[ W_1 \]

\[ \text{push}_{\tau} \]

\[ \sigma_1 \cdots \sigma_{m-1} \sigma_m \tau \]

\[ W_{m-1} \]

\[ \vdots \]

\[ W_n \]
\[ W_{n-1} \]

\[ \vdots \]

\[ W_3 \]
\[ W_2 \]
\[ W_1 \]
Stack Operations

\[ \sigma_1 \cdots \sigma_{m-1} \sigma_m \]

\[ W_{m-1} \]

\[ W_n \]
\[ W_{n-1} \]

\[ W_3 \]
\[ W_2 \]
\[ W_1 \]

\[ \sigma_1 \cdots \sigma_{m-1} \]

\[ W_{m-1} \]

\[ W_n \]
\[ W_{n-1} \]

\[ W_3 \]
\[ W_2 \]
\[ W_1 \]

\[ \text{pop}_1 \]
Stack Operations

\[ \sigma_1 \cdots \sigma_{m-1} \sigma_m \]

\[ W_{m-1} \]

\[ \vdots \]

\[ W_n \]

\[ W_{n-1} \]

\[ \vdots \]

\[ W_3 \]

\[ W_2 \]

\[ W_1 \]

\[ \Rightarrow \]

\[ \text{pop}_2 \]

\[ \vdots \]

\[ W_n \]

\[ W_{n-1} \]

\[ \vdots \]

\[ W_3 \]

\[ W_2 \]

\[ W_1 \]
Stack Operations

\[ \sigma_1 \cdots \sigma_{m-1} \sigma_m \]

**Push Operation**

\[ \begin{array}{c}
\vdots \\
W_n \\
W_{n-1} \\
W_3 \\
W_2 \\
W_1 \\
\vdots \\
\end{array} \]

\[ \begin{array}{c}
\vdots \\
W_n \\
W_{n-1} \\
W_3 \\
W_2 \\
W_1 \\
\vdots \\
\end{array} \]

\[ \text{push}_2 \]
Stack Operations

\[
\begin{align*}
\sigma_1 \cdots \sigma_{m-1} \sigma_m &= w_m \\
W_{m-1} &= w_m v_{m-1} \\
W_n &= w_m v_n \\
W_{n-1} &\neq w_m v_{n-1} \quad \text{collapse} \\
W_3 &= w_3 \\
W_2 &= w_2 \\
W_1 &= w_1 \\
\end{align*}
\]
Definition CPG

- Transition relation $\Delta$:
  state + topmost letter $\mapsto$ new state + stack-operation
  
e.g. $\delta = (q, \sigma) \mapsto (q', \text{pop}_2)$

- Configuration $(q, s)$ – $q$ state, $s$ stack (of level 2)
  
- $(q, s) \xrightarrow{\delta} (q', \text{pop}_2(s))$

- CPG: configurations of CPS + labelled transition relation

- CPG/$\varepsilon$: $\varepsilon$-contraction of CPG
Example of CPG

Grid MSO-interpretable ⇒ MSO undecidable

Alexander Kartzow (Universität Leipzig)
Example of CPG

Grid MSO-interpretable $\Rightarrow$ MSO undecidable
Example

$\mathcal{L} := (T, \text{succ})$ with $T := \{0\}^* \cup \{0^{n-1}1^j : 0 \leq j \leq 2^n\}$ is a 2-CPG/$\varepsilon$
The Pumping Lemma for 2-CPG

Definition

\[ G = (V, (\gamma \rightarrow)_{\gamma \in \Gamma}) \]

\[ L \subseteq \Gamma^* \]

\[ \overset{L}{\rightarrow} := \{(v_1, v_2) : v_1 \overset{\gamma_1}{\rightarrow} \ldots \overset{\gamma_n}{\rightarrow} v_2, \gamma_1 \ldots \gamma_n \in L\} \]

Theorem

\[ G \text{ 2-CPG}/\epsilon, g_0 \in G; \]

L, K regular languages, \[ \overset{L}{\rightarrow} \text{ finitely branching} \]

\[ \exists \ c, d \text{ s.t. } g_0 \overset{L}{\rightarrow} g_1 \overset{L}{\rightarrow} \ldots \overset{L}{\rightarrow} g_n \text{ and } |\{g : g_n \overset{K}{\rightarrow} g\}| > 2^{2c+dn} \]

\[ \Rightarrow |\{g : g_n \overset{K}{\rightarrow} g\}| = \infty. \]

Suffices: \[ \overset{L}{\rightarrow} \text{ finitely branching at } g_0, g_1, \ldots, g_{n-1}. \]
Application to trees

Example

\[ T := (T, \text{succ}) \text{ with } \]
\[ T := \{0\}^* \cup \{0^{n-1}1j : 0 \leq j \leq 2^{\log(n)n}\} \]

is not a 2-CPG/ε

Proof.

For \( L = K = \Gamma \), we get \( c, d \)

Choose \( n_0 > 2^{c+d} \) then \( \log(n_0)n_0 > c + dn_0 \)

\( \text{P.L.} \Rightarrow 0^{n_0-1}1 \) has infinitely many successors.
Application to Graphs

\[ \mathcal{G} : 2 \text{-CPG}/\varepsilon \]
Application to Graphs

\[ G: 2\text{-}\text{CPG}/\varepsilon \]

\( \xrightarrow{L} \): \( L \) regular
Application to Graphs

\[ \mathcal{G} : \text{2-CPG}/\varepsilon \]
\[ \rightarrow : L \text{ regular} \]
\[ (M_{g_0}, \rightarrow) \not\cong \mathcal{T} \text{ for} \]
\[ M_{g_0} := \{ g \in \mathcal{G} : g_0 \xrightarrow{L^*} g \} \]
1. Stacs’10:
   - Encoding of vertices of 2-CPG/ε in trees
   - $L \rightarrow$ is represented by finite tree-automaton $A_L$

2. Apply regular pumping lemma to $A_L$. 
Automaticity

- $\mathcal{T}$: set of all trees (finite binary $\Sigma$-labelled)
- automaton = (nondeterministic) finite tree-automaton

**Definition**

$R \subseteq \mathcal{T} \times \mathcal{T}$ binary relation of trees

$A$ automaton

$R$ automatic via $A$: $A$ accepts $t_1 \otimes t_2 \iff (t_1, t_2) \in R$ ($L(A) = R$)
**Definition**

\( R \subseteq M \times M \) relation over arbitrary set

\( f : M \rightarrow T \) injective function

\( A \) automaton

\((f, A)\) **automatic presentation of** \( R \): \( f(R) \) automatic via \( A \) where

\[ f(R) := \{(f(m_1), f(m_2)) : (m_1, m_2) \in R\} \]

**Theorem (Kartzow’10)**

\( \mathcal{G} \) 2-CPG/\( \varepsilon \) with edge labels from \( \Gamma \)

\( L \subseteq \Gamma^* \) regular,

then \( L \rightarrow \) has a tree-automatic presentation \((f, A)\).
Lemma (regular pumping lemma)

\[ A \text{ with } d \text{ states} \]
\[ \exists t \in L(A) \text{ with } |t| > d \Rightarrow |L(A)| = \infty \]

- \( |u| \leq d \)
- \( t \)
- \( u \)
- \( t' \)
- \( u' \)
Lemma

Automaton $A$ with $d$ states
$R$ automatic via $A$
Trees $t_1, t_2$ s.t. $|t_2| > |t_1| + d$
$(t_1, t_2) \in R \Rightarrow \{ t : (t_1, t) \in R \}$ is infinite

Proof.
Lemma

Automaton $A$ with $d$ states

$R$ automatic via $A$

Trees $t_1, t_2$ s.t. $|t_2| > |t_1| + d$

$(t_1, t_2) \in R \Rightarrow \{ t : (t_1, t) \in R \}$ is infinite

Corollary

$|\{ t : (t_1, t) \in R \}| > (|\Sigma| + 1)^2^{|t_1|+d} \Rightarrow \{ t : (t_1, t) \in R \}$ is infinite

Proof.

There are $(|\Sigma| + 1)^{2n}$ different $\Sigma$-labelled trees of depth $n$. □
Lemma

Automaton \( A \) with \( d \) states

\[ R \text{ automatic via } A \]

Trees \( t_1, t_2 \) s.t. \( |t_2| > |t_1| + d \)

\((t_1, t_2) \in R \Rightarrow \{ t : (t_1, t) \in R \} \text{ is infinite set.}\)

Lemma

\( \emptyset \text{ 2-CPG}/\varepsilon; \)

\( L \) regular such that \( L \) is finitely branching.

There is a constant \( d \) such that \( g_0 \xrightarrow{L} g_1 \xrightarrow{L} \ldots \xrightarrow{L} g_n \Rightarrow |g_n| \leq |g_0| + dn. \)
Theorem

Let $\mathcal{CPG}/\varepsilon; L, K$ regular languages, $\to^L$ finitely branching

$\exists c, d \text{ s.t. } g_0 \xrightarrow{L} g_1 \xrightarrow{L} \ldots \xrightarrow{L} g_n \text{ and } |\{g : g_n \xrightarrow{K} g\}| > 2^{c+dn} \Rightarrow |\{g : g_n \xrightarrow{K} g\}| = \infty.$
Conclusion

- pumping lemma for 2-CPG/ε: tool for disproving membership
- regular reachability $\xrightarrow{L}$ on 2-CPG/ε is tree-automatic
- pumping lemma for finite automata applied to regular reachability yields pumping lemma for 2-CPG/ε.

Open questions

- pumping lemmas for higher level CPG
- pumping lemma for the whole hierarchy
- other techniques for disproving membership