Regulated Nondeterminism in Pushdown Automata

Martin Kutrib Andreas Malcher Larissa Werlein

Institut für Informatik, Universität Giessen, Germany

Institut für Informatik, Johann Wolfgang Goethe-Universität Frankfurt
Frankfurt am Main, Germany

CIAA 2007, Prague, Czech Republic

Nondeterministic models versus deterministic models

- Models with unbounded nondeterminism
- Models with no nondeterminism, i.e., deterministic models
- Models with limited nondeterminism

Limited nondeterminism

- Turing machines
- pushdown automata
- finite automata

Nondeterministic PDAs versus deterministic PDAs

Context-free languages (context-free grammars, PDAs)

- parsing is possible in more than quadratic time, but less than cubic time
- generative capacity
- many questions are undecidable

Deterministic context-free languages ($LR(k)$ grammars, DPDAs)

- parsing is possible in linear time
- lower generative capacity
- better decidability results (e.g., equivalence, regularity)

Nondeterminism regulated by contexts

In case of limited nondeterminism only the total number of nondeterministic steps is bounded, but not the point of time or a certain situation in which a nondeterministic step may be applied.

Investigate DPDAs with context-dependent nondeterminism, i.e., DPDAs which are allowed to perform nondeterministic steps only within certain situations or contexts.

- ➜ Consider the situations "initial state," "empty stack," and combination.
- ➜ Consider PDAs with a finite and infinite amount of nondeterminism.

Nondeterminism regulated by contexts 2

- Results (Kutrib, Malcher, DLT 2006)

Restriction	Characterization
$\mathit{fin}, (\mathit{fin}, q_0), (\mathit{fin}, Z_0), (\mathit{fin}, q_0, Z_0)$	$\Gamma_{\cup}(\mathrm{DCFL})$
$\infty, (\infty, q_0)$	CFL
(∞, Z_0)	$\Gamma_{\mathsf{REG}}(\mathrm{DCFL})$
(∞, q_0, Z_0)	$\mathscr{L}_*$

- $\Gamma_{\mathsf{REG}}(\mathrm{DCFL})$ contains inherently ambiguous languages such as $\{a^m b^m c^n \mid n \geq 0\} \cup \{a^m b^n c^n \mid n \geq 0\}$.
- The time complexity is of order $O(n)$ (Bertsch, Nederhof 99).

Regulated rewriting

- Impose restrictions to some (context-free) grammar on how to use the productions.
- The restrictions are usually realized by some control device.
- Extensive investigations of this concept in many areas of formal language theory have been done.
- Cf. textbook of Dassow and Păun (1989) and Handbook of Formal Languages (1997).

For automata, this concept has been adapted by Meduna and Kolář (2000,2002).

- Idea: limit the computations in such a way that the sequence of transition steps has to form some words of a given control language.
- Result: recursively enumerable languages are already characterized by using very simple context-free control languages for one-turn regulated pushdown automata.

Nondeterminism regulated by stack contents

Idea: Nondeterministic steps are only allowed when the current content of the stack forms a word belonging to some control language R.

Formally, $\mathcal{M}$ is called an R-PDA if

- $\mathcal{M} = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$ is a PDA,
- $R \subseteq (\Gamma \setminus Z_0)^*$ is a control language,
- δ can be decomposed as $\delta(q, a, Z) = \delta_d(q, a, Z) \cup \delta_n(q, a, Z)$, where $\langle Q, \Sigma, \Gamma, \delta_d, q_0, Z_0, F \rangle$ is a DPDA and $\langle Q, \Sigma, \Gamma, \delta_n, q_0, Z_0, F \rangle$ is a PDA ($q \in Q$, $a \in \Sigma_\lambda$, and $Z \in \Gamma$).
- for all $q, q' \in Q$, $a \in \Sigma_\lambda$, $w \in \Sigma^*$, $Z \in \Gamma$, and $\gamma \in \Gamma^*$,
  - $(q, aw, Z\gamma) \vdash (q', w, \gamma'\gamma)$, if $(q', \gamma') \in \delta_n(q, a, Z)$ and $Z\gamma = \gamma'' Z_0$ with $\gamma'' \in R$,
  - $(q, aw, Z\gamma) \vdash (q', w, \gamma'\gamma)$, if $\delta_d(q, a, Z) = (q', \gamma')$ and $Z\gamma = \gamma'' Z_0$ with $\gamma'' \notin R$.

Example

Let $R = \{b^n a^n \mid n \geq 1\}$ and consider the following R-PDA $\mathcal{M}$ on input $a^*b^*c^*$:

- Push all as read on the stack.
- First b is read: an a is popped and the following bs are pushed on the stack.
- First c is read: check the stack content γ.
- $\gamma \in R$: match cs against bs
- Accept, if all bs are popped and the last c is matched against the topmost a
- Reject otherwise.

$\mathcal{M}$ accepts the non-context-free language $\{a^n b^n c^n \mid n \geq 2\}$.

More examples

- $R = \emptyset$ means no nondeterminism. Thus, $\mathscr{L}(\emptyset\text{-PDA}) = \mathrm{DCFL}$.
- If $R = (\Gamma \setminus Z_0)^*$, then $\mathscr{L}(R\text{-PDA}) = \mathrm{CFL}$. Especially, $\mathscr{L}(\{a, b\}^*\text{-PDA}) = \mathrm{CFL}$.
- Recall: (∞, Z_0)-PDAs characterize $\Gamma_{\mathsf{REG}}(\mathrm{DCFL})$. In other words, $\mathscr{L}(\{\lambda\}\text{-PDA}) = \Gamma_{\mathsf{REG}}(\mathrm{DCFL})$.
- The family of one-counter languages is a proper subset of $\mathscr{L}(\{a\}^*\text{-PDA})$.
  Consider $L = \{a^n bwcw^R ba^n \mid n \geq 1, w \in \{a, b\}^*\}$.

Theorem

Let R be a regular set and $\mathcal{M}$ be an R-PDA.
Then an equivalent PDA $\mathcal{M}'$ can effectively be constructed.

Constructions

Theorem

Let R be a regular set and $\mathcal{M}$ be an R-PDA.
Then an equivalent PDA $\mathcal{M}'$ can effectively be constructed.

Proof idea:

- Consider the state control/stack of $\mathcal{M}'$ to have two components.
- The first component simulates the state control/stack of $\mathcal{M}$.
- The second component of the stack stores the history of a computation of a DFA $\mathcal{A}$. The second component of the state stores the current state of $\mathcal{A}$.
- Thus, it can be checked whether or not the current content of the stack belongs to R.

Theorem

Let $R \neq \{\lambda\}$ be not empty. Then the families $\mathscr{L}((R \cup \{\lambda\})$-$PDA)$ and $\mathscr{L}((R \setminus \{\lambda\})$-$PDA)$ are equal.

Finite control sets

Theorem

Let R be finite and not empty. Then the families $\mathscr{L}(R\text{-}PDA)$ and $\mathscr{L}(\{\lambda\}\text{-}PDA)$ are equal.

Proof idea:

- W.l.o.g. $\lambda \in R$. Thus, $\mathscr{L}(\{\lambda\}\text{-PDA}) \subseteq \mathscr{L}(R\text{-PDA})$.
- W.l.o.g. we may assume that the second component of the state indicates whether the current content of the stack is a word of R.
- Simulate an equivalent $\{\lambda\}$-PDA $\mathcal{M}'$:
  - $\mathcal{M}'$'s stack contains no word from R: $\mathcal{M}'$ works as $\mathcal{M}$.
  - $\mathcal{M}'$'s stack contains $w \in R$: store w in the state and empty the stack via λ-transitions. If the stack is empty, $\mathcal{M}'$ may guess the nondeterministic step of $\mathcal{M}$ and the successor stack content is pushed again on the stack.
- $\mathscr{L}(R\text{-PDA}) \subseteq \mathscr{L}(\{\lambda\}\text{-PDA})$

Hierarchy

Consider the following four control sets: $\emptyset$, $\{\lambda\}$, $\{a\}^*$, $\{a,b\}^*$.
Since

$$\emptyset \subset \{\lambda\} \subset \{a\}^* \subset \{a,b\}^*$$

we obtain

$$\mathscr{L}(\emptyset\text{-PDA}) \subseteq \mathscr{L}(\{\lambda\}\text{-PDA}) \subseteq \mathscr{L}(\{a\}^*\text{-PDA}) \subseteq \mathscr{L}(\{a,b\}^*\text{-PDA})$$

Goal: Show the properness of the inclusions.

Theorem

$$\mathscr{L}(\emptyset\text{-PDA}) \subset \mathscr{L}(\{\lambda\}\text{-PDA}) \subset \mathscr{L}(\{a\}^*\text{-PDA}) \subset \mathscr{L}(\{a,b\}^*\text{-PDA})$$

- $\mathscr{L}(\emptyset\text{-PDA}) \subset \mathscr{L}(\{\lambda\}\text{-PDA})$, since $\mathscr{L}(\emptyset\text{-PDA}) = \mathrm{DCFL}$ and $\mathscr{L}(\{\lambda\}\text{-PDA}) = \Gamma_{\mathsf{REG}}(\mathrm{DCFL})$.

- ➜ $\mathscr{L}(\{\lambda\}\text{-PDA}) \subset \mathscr{L}(\{a\}^*\text{-PDA})$ by the following lemma.

Lemma

The language $L = \{a^n bwba^n b \mid n \geq 1, w \in \{a,b\}^*\}$ does not belong to the family $\mathscr{L}(\{\lambda\}\text{-}PDA)$.

- ➜ $\mathscr{L}(\{a\}^*\text{-PDA}) \subset \mathscr{L}(\{a,b\}^*\text{-PDA})$ by the following lemma.

Lemma

The language $L = \{a^m b^n cww^R cb^n a^m \mid m,n \geq 1, w \in \{a,b\}^*\}$ does not belong to the family $\mathscr{L}(\{a\}^*\text{-}PDA)$.

Closure properties

Let R be a non-empty regular set.

- $\mathscr{L}(R\text{-PDA})$ is closed under union.
- $\mathscr{L}(R\text{-PDA})$ is closed under intersection with regular sets and inverse homomorphism.
- $\mathscr{L}(R\text{-PDA})$ is not closed under complementation.
- If $\mathscr{L}(R\text{-PDA}) \neq \mathrm{CFL}$, then it is not closed under homomorphism.

Let $R = \{\lambda\}$.

- $\mathscr{L}(R\text{-PDA})$ is closed under concatenation and Kleene star.

Summary

Hierarchy

$$\mathscr{L}(\emptyset\text{-PDA}) \subset \mathscr{L}(\{\lambda\}\text{-PDA}) \subset \mathscr{L}(\{a\}^*\text{-PDA}) \subset \mathscr{L}(\{a,b\}^*\text{-PDA})$$

Closure properties

Language Class	$\cup$	$\bullet$	$*$	h	h^{-1}	$\cap_{reg}$	$\sim$
$\mathscr{L}$(1-counter)	$+$	$+$	$+$	$+$	$+$	$+$	$-$
$\mathscr{L}(\emptyset\text{-PDA})$	$-$	$-$	$-$	$-$	$+$	$+$	$+$
$\mathscr{L}(\{\lambda\}\text{-PDA})$	$+$	$+$	$+$	$-$	$+$	$+$	$-$
$\mathscr{L}(R\text{-PDA})$	$+$	?	?	$-$	$+$	$+$	$-$
CFL	$+$	$+$	$+$	$+$	$+$	$+$	$-$

Table: Closure properties of pushdown automata languages with regulated nondeterminism, where R is a non-empty regular set such that $\mathscr{L}(R\text{-PDA}) \neq \mathrm{CFL}$.

Open questions

- Prove or disprove closure under concatenation and Kleene star.
- Investigate the equivalence of acceptance modes.
- Try to find conditions on the structure of regular sets R, S such that $R \subset S$ implies $\mathscr{L}(R\text{-PDA}) \subset \mathscr{L}(S\text{-PDA})$.
- Investigate parsing algorithms.
- Investigate context-free control sets R.