Deep Pushdown Automata

Lukáš Vrábel

Department of Information Systems
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, Brno 61266, Czech Republic

Formal Language Theory Workshop, 2010

Motivation

- hierarchy between context-free and context-sensitive languages
- automaton counterpart to state grammars
- generalization of the classical pushdown automata
- expansion deeper on stack - expand *n*-th non-input symbol

Deep Pushdown Automaton

Deep Pushdown Automaton

A Deep Pushdown Automaton is septuple

$$M = (Q, \Sigma, \Gamma, R, s, S, F)$$

Q finite set of states

Σ input alphabet

Γ pushdown alphabet, where $\Sigma \subset \Gamma$

R finite set of rules

s is the start state, $s \in Q$

S is the start pushdown symbol, $S \in \Gamma$

F set of final states, $F \subset Q$

where I, Q and Γ are pairwise disjoint

Rules

Rule

A Rule is quintuple

$$mqA \to pv$$

where

m is the depth of expansion, $m \in I$

$q, p \in Q$

A non-input symbol, $A \in \Gamma - \Sigma$

v string of pushdown symbols, $v \in \Gamma^+$

Depth

- Finite number of rules $\Rightarrow$ exists n such that depth of each rule $\leq n$
- M_n denotes automaton with depth n

Computational Step

Configuration

$$x \in Q \times \Sigma^* \times \Gamma^*$$

Move

Let x, y be configurations. Then

$$x \Rightarrow y$$

if and only if one of the following holds:

- **pop** $x = (q, au, az), y = (q, u, z)$
- **expand** $x = (q, w, uAz), y = (p, w, uvz)$ with $mqA \rightarrow pv \in R$ and $occur(u, \Gamma - \Sigma) = m - 1$

with $u, v, w, z \in \Gamma^*$

Accepted Language

Accepted Word

Deep pushdown automaton M accepts $w \in \Sigma^*$ if

$$(s, w, S) \Rightarrow^* (f, \varepsilon, \varepsilon)$$

with $f \in F$ and $\Rightarrow^*$ denoting the reflexive and transitive closure of $\Rightarrow$

Accepted Language

All words accepted by M is the language of M, denoted by $L(M)$:

$$L(M) = \{w \in \Sigma^* : (s, w, S) \Rightarrow^* (f, \varepsilon, \varepsilon), f \in F\}$$

Example

Example

$$M = (\{s, p, q, r, f\}, \{a, b, c\}, \{a, b, c, S, X\}, R, s, \{f\})$$

$$\begin{aligned} R = \{ \quad & 1\colon 1sS \to pXX, \quad 2\colon 1pX \to qaXb, \quad 3\colon 2qX \to pXc, \\ & 4\colon 1pX \to rab, \quad 5\colon 2rX \to fc \quad \} \end{aligned}$$

Example

$$M = (\{s, p, q, r, f\}, \{a, b, c\}, \{a, b, c, S, X\}, R, s, \{f\})$$

$$R = \{ \quad 1: 1sS \to pXX, \quad 2: 1pX \to qaXb, \quad 3: 2qX \to pXc,$$
$$4: 1pX \to rab, \quad 5: 2rX \to fc \quad \}$$

$$(s, aabbcc, S) \quad \Rightarrow \quad (p, aabbcc, XX)\ [1]$$

Example

Example

$$M = (\{s, p, q, r, f\}, \{a, b, c\}, \{a, b, c, S, X\}, R, s, \{f\})$$

$$R = \{ \quad 1\colon 1sS \to pXX, \quad 2\colon 1pX \to qaXb, \quad 3\colon 2qX \to pXc, \quad 4\colon 1pX \to rab, \quad 5\colon 2rX \to fc \quad \}$$

$$(s, aabbcc, S) \Rightarrow (p, aabbcc, XX)\ [1]$$
$$(p, aabbcc, XX) \Rightarrow (q, aabbcc, aXbX)\ [2]$$

Example

Example

$$M = (\{s, p, q, r, f\}, \{a, b, c\}, \{a, b, c, S, X\}, R, s, \{f\})$$

$$R = \{ \quad 1\colon 1sS \to pXX, \quad 2\colon 1pX \to qaXb, \quad 3\colon 2qX \to pXc, \quad 4\colon 1pX \to rab, \quad 5\colon 2rX \to fc \quad \}$$

$$\begin{array}{lcl}
(s, aabbcc, S) & \Rightarrow & (p, aabbcc, XX)\ [1] \\
(p, aabbcc, XX) & \Rightarrow & (q, aabbcc, aXbX)\ [2] \\
(q, aabbcc, aXbX) & \Rightarrow & (q, abbcc, XbX)\ [pop]
\end{array}$$

Example

$$M = (\{s, p, q, r, f\}, \{a, b, c\}, \{a, b, c, S, X\}, R, s, \{f\})$$

$$R = \{ \quad 1\colon 1sS \rightarrow pXX, \quad 2\colon 1pX \rightarrow qaXb, \quad 3\colon 2qX \rightarrow pXc, \quad 4\colon 1pX \rightarrow rab, \quad 5\colon 2rX \rightarrow fc \quad \}$$

$$\begin{array}{lcl}
(s, aabbcc, S) & \Rightarrow & (p, aabbcc, XX)\ [1] \\
(p, aabbcc, XX) & \Rightarrow & (q, aabbcc, aXbX)\ [2] \\
(q, aabbcc, aXbX) & \Rightarrow & (q, abbcc, XbX)\ [pop] \\
(q, abbcc, XbX) & \Rightarrow & (p, abbcc, XbXc)\ [3]
\end{array}$$

Example

Example

$$M = (\{s, p, q, r, f\}, \{a, b, c\}, \{a, b, c, S, X\}, R, s, \{f\})$$

$$R = \{ \quad 1: 1sS \rightarrow pXX, \quad 2: 1pX \rightarrow qaXb, \quad 3: 2qX \rightarrow pXc, \quad 4: 1pX \rightarrow rab, \quad 5: 2rX \rightarrow fc \quad \}$$

$$\begin{array}{lcl}
(s, aabbcc, S) & \Rightarrow & (p, aabbcc, XX)\ [1] \\
(p, aabbcc, XX) & \Rightarrow & (q, aabbcc, aXbX)\ [2] \\
(q, aabbcc, aXbX) & \Rightarrow & (q, abbcc, XbX)\ [pop] \\
(q, abbcc, XbX) & \Rightarrow & (p, abbcc, XbXc)\ [3] \\
(p, abbcc, XbXc) & \Rightarrow & (r, abbcc, abbXc)\ [4]
\end{array}$$

Example

Example

$$M = (\{s, p, q, r, f\}, \{a, b, c\}, \{a, b, c, S, X\}, R, s, \{f\})$$

$$R = \{ \quad 1: 1sS \rightarrow pXX, \quad 2: 1pX \rightarrow qaXb, \quad 3: 2qX \rightarrow pXc, \quad 4: 1pX \rightarrow rab, \quad 5: 2rX \rightarrow fc \quad \}$$

$$\begin{array}{lcl}
(s, aabbcc, S) & \Rightarrow & (p, aabbcc, XX)\ [1] \\
(p, aabbcc, XX) & \Rightarrow & (q, aabbcc, aXbX)\ [2] \\
(q, aabbcc, aXbX) & \Rightarrow & (q, abbcc, XbX)\ [pop] \\
(q, abbcc, XbX) & \Rightarrow & (p, abbcc, XbXc)\ [3] \\
(p, abbcc, XbXc) & \Rightarrow & (r, abbcc, abbXc)\ [4] \\
(r, abbcc, abbXc) & \Rightarrow^{3} & (r, cc, Xc)\ [pop, pop, pop]
\end{array}$$

Example

Example

$$M = (\{s, p, q, r, f\}, \{a, b, c\}, \{a, b, c, S, X\}, R, s, \{f\})$$

$$R = \{ \quad 1: 1sS \to pXX, \quad 2: 1pX \to qaXb, \quad 3: 2qX \to pXc, \quad 4: 1pX \to rab, \quad 5: 2rX \to fc \quad \}$$

$$\begin{array}{lcl}
(s, aabbcc, S) & \Rightarrow & (p, aabbcc, XX)\ [1] \\
(p, aabbcc, XX) & \Rightarrow & (q, aabbcc, aXbX)\ [2] \\
(q, aabbcc, aXbX) & \Rightarrow & (q, abbcc, XbX)\ [pop] \\
(q, abbcc, XbX) & \Rightarrow & (p, abbcc, XbXc)\ [3] \\
(p, abbcc, XbXc) & \Rightarrow & (r, abbcc, abbXc)\ [4] \\
(r, abbcc, abbXc) & \Rightarrow^3 & (r, cc, Xc)\ [pop, pop, pop] \\
(r, cc, Xc) & \Rightarrow & (f, cc, cc)\ [5]
\end{array}$$

Example

Example

$$M = (\{s, p, q, r, f\}, \{a, b, c\}, \{a, b, c, S, X\}, R, s, \{f\})$$

$$\begin{aligned} R = \{ \quad & 1\colon 1sS \to pXX, \quad 2\colon 1pX \to qaXb, \quad 3\colon 2qX \to pXc, \\ & 4\colon 1pX \to rab, \quad 5\colon 2rX \to fc \quad \} \end{aligned}$$

$$\begin{array}{lcl}
(s, aabbcc, S) & \Rightarrow & (p, aabbcc, XX)\ [1] \\
(p, aabbcc, XX) & \Rightarrow & (q, aabbcc, aXbX)\ [2] \\
(q, aabbcc, aXbX) & \Rightarrow & (q, abbcc, XbX)\ [pop] \\
(q, abbcc, XbX) & \Rightarrow & (p, abbcc, XbXc)\ [3] \\
(p, abbcc, XbXc) & \Rightarrow & (r, abbcc, abbXc)\ [4] \\
(r, abbcc, abbXc) & \Rightarrow^{3} & (r, cc, Xc)\ [pop, pop, pop] \\
(r, cc, Xc) & \Rightarrow & (f, cc, cc)\ [5] \\
(f, cc, cc) & \Rightarrow^{2} & (f, \varepsilon, \varepsilon)\ [pop, pop]
\end{array}$$

Example

Example

$$M = (\{s, p, q, r, f\}, \{a, b, c\}, \{a, b, c, S, X\}, R, s, \{f\})$$

$$\begin{aligned} R = \{ \quad & 1\colon 1sS \to pXX, \quad 2\colon 1pX \to qaXb, \quad 3\colon 2qX \to pXc, \\ & 4\colon 1pX \to rab, \quad 5\colon 2rX \to fc \quad \} \end{aligned}$$

$$\begin{array}{lcl}
(s, aabbcc, S) & \Rightarrow & (p, aabbcc, XX)\ [1] \\
(p, aabbcc, XX) & \Rightarrow & (q, aabbcc, aXbX)\ [2] \\
(q, aabbcc, aXbX) & \Rightarrow & (q, abbcc, XbX)\ [pop] \\
(q, abbcc, XbX) & \Rightarrow & (p, abbcc, XbXc)\ [3] \\
(p, abbcc, XbXc) & \Rightarrow & (r, abbcc, abbXc)\ [4] \\
(r, abbcc, abbXc) & \Rightarrow^3 & (r, cc, Xc)\ [pop, pop, pop] \\
(r, cc, Xc) & \Rightarrow & (f, cc, cc)\ [5] \\
(f, cc, cc) & \Rightarrow^2 & (f, \varepsilon, \varepsilon)\ [pop, pop]
\end{array}$$

Accepted language: $L = \{a^n b^n c^n : n \geq 1\}$

Infinite Hierarchy

Family of Languages

$deepPD_n$ denotes family of languages accepted by deep pushdown automata of depth k, where $1 \leq k \leq n$

Theorem

$deepPD_1 = CF$

Theorem

For every $n \geq 1$, $deepPD_n \subset deepPD_{n+1} \subset CS$

Generalization

ε-Rules

- Natural generalization - adding ε rules in the form: $mqA \to p\varepsilon$
- $deepPD_n^{\varepsilon}$ denotes family of languages accepted by deep pushdown automata fo depth n with ε rules

Generalization

ε-Rules

- Natural generalization - adding ε rules in the form: $mqA \to p\varepsilon$
- $deepPD_n^{\varepsilon}$ denotes family of languages accepted by deep pushdown automata fo depth n with ε rules

Theorem

$deepPD_n \subset deepPD_n^{\varepsilon}$

Proof.

M cannot accept empty string without ε rules. $\square$

Generalization

ε-Rules

- Natural generalization - adding ε rules in the form: $mqA \rightarrow p\varepsilon$
- $deepPD_n^{\varepsilon}$ denotes family of languages accepted by deep pushdown automata fo depth n with ε rules

Theorem

$deepPD_n \subset deepPD_n^{\varepsilon}$

Proof.

M cannot accept empty string without ε rules. □

Open Promblem

What about languages without empty string?

Informal Proof

Idea

- simulate moves of M_n^ε by M_n
- do not generate to-be-erased symbols on stack
- but we must be able to simulate the moves involving this symbols
- solution: "save" them into the state logic
- restriction: as set of states is finite, the number of "remembered" symbols is limited by some k

Example

$$(q, u, aaAbbBcc) \rightarrow (\langle q, AB\rangle, u, aaAbbBcc)$$

Example

Example

With ε-Rules

$$\begin{array}{lcll} (q, u, aaAbbBcc) & \Rightarrow & (p, u, aaAbbBccc) & [2qB \rightarrow pBc] \\ (p, u, aaAbbBccc) & \Rightarrow & (p, u, aaAbbccc) & [2pB \rightarrow p\varepsilon] \end{array}$$

Example

Example

With ε-Rules

$$\begin{array}{lcll} (q, u, aaAbbBcc) & \Rightarrow & (p, u, aaAbbBccc) & [2qB \rightarrow pBc] \\ (p, u, aaAbbBccc) & \Rightarrow & (p, u, aaAbbccc) & [2pB \rightarrow p\varepsilon] \end{array}$$

Example

Without ε-Rules

$$\begin{array}{ll} 1 : 2\langle q, AB\rangle B \rightarrow \langle p, A\bar{B}\rangle c & 3 : 1\langle p, A\bar{B}\rangle A \rightarrow \langle p, A\rangle A \\ 2 : 2\langle q, AB\rangle B \rightarrow \langle p, AB\rangle Bc & \end{array}$$

Example

Example

With ε-Rules

$$\begin{array}{lcll} (q, u, aaAbbBcc) & \Rightarrow & (p, u, aaAbbBccc) & [2qB \rightarrow pBc] \\ (p, u, aaAbbBccc) & \Rightarrow & (p, u, aaAbbccc) & [2pB \rightarrow p\varepsilon] \end{array}$$

Example

Without ε-Rules

$$\begin{array}{ll} 1 : 2\langle q, AB\rangle B \rightarrow \langle p, A\bar{B}\rangle c & 3 : 1\langle p, A\bar{B}\rangle A \rightarrow \langle p, A\rangle A \\ 2 : 2\langle q, AB\rangle B \rightarrow \langle p, AB\rangle Bc & \end{array}$$

$$\begin{array}{lcll} (\langle q, AB\rangle, u, aaAbbBcc) & \Rightarrow & (\langle p, A\bar{B}\rangle, u, aaAbbccc) & [1] \\ (\langle p, A\bar{B}\rangle, u, aaAbbccc) & \Rightarrow & (\langle p, A\rangle, u, aaAbbccc) & [3] \end{array}$$

k-Limited Erasing

As there is finite number of states, we can track the erased symbols only to some depth k - we can eliminate the ε-rules only from automata with k-limited erasing.

k-Limited Erasing

Let $M_n^\varepsilon = (Q, \Sigma, \Gamma, R, s, S, F)$ be a deep pushdown automaton with erasing rules. M_n^ε erases its non-input symbols in k-limited way, if for every $w \in L(M_n^\varepsilon)$ there exists a sequence of configurations $(s, w, S) \Rightarrow^* (f, \varepsilon, \varepsilon)$, $f \in F$, that satisfies following properties:

- Let N_ε be the set of non-input symbols erased at some point of derivation.
- There exists such $k \in I$, that the depth of each $A \in N_\varepsilon$ in each configuration is $\leq k$

Sketch of Proof

Lemma

For each deep PDA M_n^{ε}, which erases its non-input symbols in k-limited way, there exists a deep PDA M_n such that $L(M_n^{\varepsilon}) = L(M_n)$. $\varepsilon \notin L(M_n^{\varepsilon})$.

Proof

- Let $M_n^{\varepsilon} = (Q, \Sigma, \Gamma, R, s, S, F)$ that erases its non-input symbols in k-limited way.
- We will construct $M_n = (Q', \Sigma, \Gamma, R', \langle s, S\rangle, S, F')$, that simulates M_n^{ε}'s derivations.
- We must describe the construction of Q', R' and F'

Construction of Q'

Notation

N	the set of non-input symbols, $N = \Gamma - \Sigma$
N_ε	the set of symbols that can be erased in arbitrary number of moves, $N_\varepsilon \subseteq N$
$\overline{N}$	$= \{\overline{A} \mid A \in N\}$
$prefix(u, i)$	is u's prefix of length i if $\lvert u \rvert \geq i$, otherwise it is u. $i \geq 0$.
$suffix(u, i)$	$u = prefix(u, i) suffix(u, i)$.
$occur(u, W)$	number of occurrences of symbols from W in the word u

Construction of Q'

$$Q' = \{\langle q, u \rangle \mid q \in Q, u \in prefix((N \cup \overline{N})^*, k)\}$$

Construction of F' and R'

Construction of F'

Add each $\langle f, \varepsilon \rangle \in Q'$, such that $f \in F$, to F'

Construction of R'

Construction of R' consists of three steps:

1. transfer rules transfering non-input symbols from stack to the state
2. simulation of erasing rules
3. simulation of expansion rules

Transfer Rules

Transfer Rules

For each $\langle q, u \rangle \in Q', |u| < k$ and for each $A \in N$:

- let $d = occur(u, N) + 1$
- add $d\langle q, u \rangle A \rightarrow \langle q, uA \rangle A$ to R'

Transfer Rules

For each $\langle q, u\rangle \in Q', |u| < k$ and for each $A \in N$:

- let $d = occur(u, N) + 1$
- add $d\langle q, u\rangle A \rightarrow \langle q, uA\rangle A$ to R'

Notation

$f_N(u)$ homomorphism over $(\Gamma \cup \overline{N})^*$ defined as $f_N(A) = A$ for $A \in (N \cup \overline{N})$ and $f_N(a) = \varepsilon$ otherwise.

Example

- $f_N(aAb\overline{B}c) = A\overline{B}$

Erasing Rules

Erasing Rules

For each $mqA \to p\varepsilon \in R$ and each $\langle q, u\overline{A}v\rangle \in Q'$:

- u or v must contain at least one non-input symbol
- let $X \in N$ denote the first non-input symbol occuring in the word $u\overline{A}v$
- add $1\langle q, u\overline{A}v\rangle X \to \langle p, uv\rangle X$ to R' if $occur(u, N \cup \overline{N}) = m - 1$

Erasing Rules

Erasing Rules

For each $mqA \to p\varepsilon \in R$ and each $\langle q, u\overline{A}v\rangle \in Q'$:

- u or v must contain at least one non-input symbol
- let $X \in N$ denote the first non-input symbol occuring in the word $u\overline{A}v$
- add $1\langle q, u\overline{A}v\rangle X \to \langle p, uv\rangle X$ to R' if $occur(u, N \cup \overline{N}) = m - 1$

What about states without non-input symbols? We can't use rules.

Erasing Rules

Erasing Rules

For each $mqA \rightarrow p\varepsilon \in R$ and each $\langle q, u\overline{A}v\rangle \in Q'$:

- u or v must contain at least one non-input symbol
- let $X \in N$ denote the first non-input symbol occuring in the word $u\overline{A}v$
- add $1\langle q, u\overline{A}v\rangle X \rightarrow \langle p, uv\rangle X$ to R' if $occur(u, N \cup \overline{N}) = m - 1$

What about states without non-input symbols? We can't use rules.

"Empty" States

For each $\langle q, \overline{u}\rangle \in Q'$ where $\overline{u} \in \overline{N}^*$:

- $u \in N^*$ denotes the equivalent word to $\overline{u}$
- if there is sequence of configurations $(q, v, w) \Rightarrow^* (p, \varepsilon, \varepsilon), p \in F$ for M_n^{ε}, such that $f_N(w) = u$, then add $\langle q, \overline{u}\rangle$ to F'

Progress

Progress

- construction of Q': done
- construction of F': done
- construction of R':
  - Erasing rules: done
  - Transfer rules: done
  - Expansion rules

Expansion Rules

Notation

$non_\varepsilon(u)$ homomorphism over $(\Gamma \cup \overline{N})^*$ defined as $non_\varepsilon(\overline{A}) = \varepsilon$ for $\overline{A} \in \overline{N}$ and $non_\varepsilon(A) = A$ otherwise.

$\sigma(u)$ substitution over $(\Gamma \cup \overline{N})^*$ defined as $\sigma(A) = \{A, \overline{A}\}$ for $A \in N_\varepsilon$ and $\sigma(A) = \{A\}$ otherwise.

Expansion Rules

Notation

$non_\varepsilon(u)$ homomorphism over $(\Gamma \cup \overline{N})^*$ defined as $non_\varepsilon(\overline{A}) = \varepsilon$ for $\overline{A} \in \overline{N}$ and $non_\varepsilon(A) = A$ otherwise.

$\sigma(u)$ substitution over $(\Gamma \cup \overline{N})^*$ defined as $\sigma(A) = \{A, \overline{A}\}$ for $A \in N_\varepsilon$ and $\sigma(A) = \{A\}$ otherwise.

Example

- $non_\varepsilon(AB\overline{C}D) = ABD$

Expansion Rules

Notation

$non_\varepsilon(u)$ homomorphism over $(\Gamma \cup \overline{N})^*$ defined as $non_\varepsilon(\overline{A}) = \varepsilon$ for $\overline{A} \in \overline{N}$ and $non_\varepsilon(A) = A$ otherwise.

$\sigma(u)$ substitution over $(\Gamma \cup \overline{N})^*$ defined as $\sigma(A) = \{A, \overline{A}\}$ for $A \in N_\varepsilon$ and $\sigma(A) = \{A\}$ otherwise.

Example

- $non_\varepsilon(AB\overline{C}D) = ABD$
- let $N_\varepsilon = \{A, B\}$. Then $\sigma(aAbBc) = \{aAbBc, a\overline{A}bBc, aAb\overline{B}c, a\overline{A}b\overline{B}c\}$

Expansion Rules

For each $mqA \to px$ in R, for each $w \in \sigma(x)$ and for each $\langle q, uAv \rangle \in Q'$:

Expansion Rules

For each $mqA \rightarrow px$ in R, for each $w \in \sigma(x)$ and for each $\langle q, uAv \rangle \in Q'$:

- let $d = occur(u, N) + 1$

Expansion Rules

For each $mqA \to px$ in R, for each $w \in \sigma(x)$ and for each $\langle q, uAv \rangle \in Q'$:

- let $d = occur(u, N) + 1$
- let $s = prefix(uf_N(w)v, k)$

Expansion Rules

For each $mqA \rightarrow px$ in R, for each $w \in \sigma(x)$ and for each $\langle q, uAv \rangle \in Q'$:

- let $d = occur(u, N) + 1$
- let $s = prefix(uf_N(w)v, k)$
- add $d\langle q, uAv \rangle A \rightarrow \langle p, s \rangle, non_\varepsilon(w)$ to R' iff it satisfies following rules:

Expansion Rules

For each $mqA \rightarrow px$ in R, for each $w \in \sigma(x)$ and for each $\langle q, uAv \rangle \in Q'$:

- let $d = occur(u, N) + 1$
- let $s = prefix(uf_N(w)v, k)$
- add $d\langle q, uAv \rangle A \rightarrow \langle p, s \rangle, non_\varepsilon(w)$ to R' iff it satisfies following rules:
  - $non_\varepsilon(w) \neq \varepsilon$

Expansion Rules

For each $mqA \to px$ in R, for each $w \in \sigma(x)$ and for each $\langle q, uAv \rangle \in Q'$:

- let $d = occur(u, N) + 1$
- let $s = prefix(uf_N(w)v, k)$
- add $d\langle q, uAv \rangle A \to \langle p, s \rangle, non_\varepsilon(w)$ to R' iff it satisfies following rules:
  - $non_\varepsilon(w) \neq \varepsilon$
  - $occur(u, N \cup \overline{N}) = m - 1$ (i.e. we must satisfy the original depth of expeansion)

Expansion Rules

For each $mqA \rightarrow px$ in R, for each $w \in \sigma(x)$ and for each $\langle q, uAv \rangle \in Q'$:

- let $d = occur(u, N) + 1$
- let $s = prefix(uf_N(w)v, k)$
- add $d\langle q, uAv \rangle A \rightarrow \langle p, s \rangle, non_\varepsilon(w)$ to R' iff it satisfies following rules:
  - $non_\varepsilon(w) \neq \varepsilon$
  - $occur(u, N \cup \overline{N}) = m - 1$ (i.e. we must satisfy the original depth of expeansion)
  - $suffix(uf_N(w)v, k)$ does not contain a symbol from $\overline{N}$ (i.e. we dont want to lose the to-be-erased symbols)

Note: $u, v \in (N \cup \overline{N})^*$

Bibliography

Alexander Meduna.
Deep pushdown automata.
Acta Inf., 42(8-9):541–552, 2006.