Reachability for Stateless Multi-Stack Automata

Wojciech Czerwinski, Piotr Hofman and Sławomir Lasota
Institute of Informatics, University of Warsaw

Plan

1. Motivation and Model
2. Results
3. A proof

SLMSA

Idea

Why do we investigate automata with many stacks?

Stateless automaton with many stacks

1. Multiple stacks.
2. Stacks alphabets are disjoint.
3. Transitions like rules in CFG in GNF. $X \longrightarrow aYZ$
4. Acceptance condition: all stacks empty.

Additional assumption

For each stack symbol there is a sequence of transitions which annihilates it.

Digression (another reason why it is fun)

1. Let's extend a CFG in GNF by adding new rules of the form $XY \longrightarrow YX$.
2. Let's take D a complement of an independence relation. Assume that D is an **equivalence relation**.
3. Then the model of automaton is exactly **SLMSA**.

Example (grammar)

- Nonterminals: A, B, C, X Terminals: a, b, c
- Initial symbol X
- Rules

$$
\begin{array}{ccccccccc}
A & \longrightarrow & a & B & \longrightarrow & b & C & \longrightarrow & c \\
X & \longrightarrow & & X & \longrightarrow & XABC & & &
\end{array}
$$

- Swap rules

$$
\begin{array}{ccccccccc}
AB & \longleftrightarrow & BA & CA & \longleftrightarrow & AC & BC & \longleftrightarrow & CB \\
XB & \longleftrightarrow & BX & XC & \longleftrightarrow & CX & & &
\end{array}
$$

- Generated language is $\sharp a = \sharp b = \sharp c$

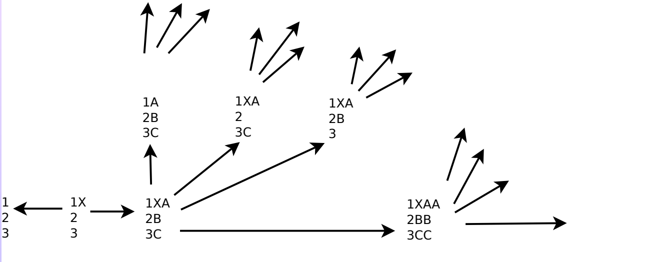
Example (automaton)

- Stack symbols: A, B, C, X alphabet: a, b, c
- 3 stacks $(A, X)(B)(C)$
- Initial configuration $(X)()()$
- Rules

$$\begin{array}{lllllllll} A & \longrightarrow & a & B & \longrightarrow & b & C & \longrightarrow & c \\ X & \longrightarrow & & X & \longrightarrow & XABC & & & \end{array}$$

- Generated language is $\sharp a = \sharp b = \sharp c$

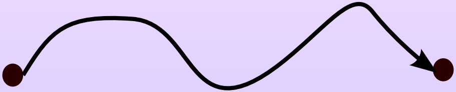
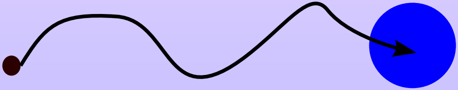
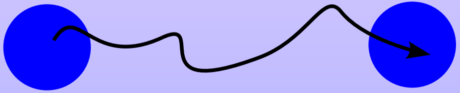
We want to investigate the configuration graph of SLMSA.



Reachability

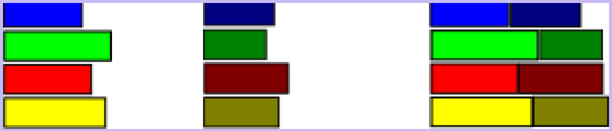
For a given two sets of configurations S and F we ask if there exists a path from an element of S to an element of F.

Problems

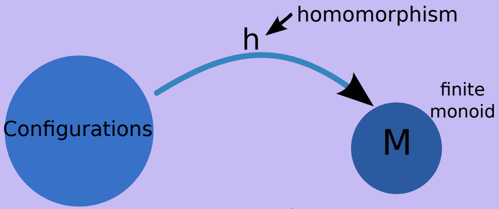
1. 
2. Which sets of configurations are regular?
3. What is $Pre^*(regular\ set)$?
4. 
5. 

Regular set of configurations

Concatenation of two configurations.



Regular set of configurations.



$L \subset M$, regular set $= h^{-1}(L)$

Equivalent definitions

1. Concatenation of stacks is a regular language.
2. Shuffle of stacks is a regular language.
3. There are some others characterizations...

Example

$(A^*)(B)(C^*)$ is a regular set but $(A^n)(B^n)$ is not.

$Pre^*(L)$

Set of configurations from which L is reachable.

Results

	PDA	$SLMSA$
$Conf\ \ to\ \ Conf$	P	**NP**
$Conf\ \ to\ \ Reg$	P	**NP**

theorem

$Pre^{*}(Reg)$ is a regular set.

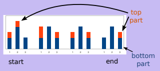
theorem

$Post^{*}(Reg)$ is not a regular set.

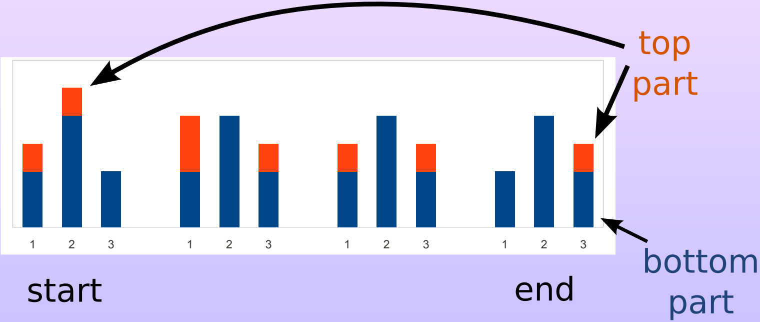
Proof

- $R = h^{-1}(\{m1, m2 \cdots m_k\})$ - is regular.
- $Pre^*(R) = Pre^*(h^{-1}(m1)) \cup \cdots \cup Pre^*(h^{-1}(m_k))$.
- So we only need to prove that $Pre^*(h^{-1}(m_i))$ is regular.

- Bottom and top parts.



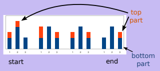
Proof



- $(Pre^*(x), x)$ define Pre^* relation.
  From every such pair we cancel bottom part and obtain $P\bar{r}e^*$ relation.

Proof

- $R = h^{-1}(\{m1, m2 \cdots m_k)$ - is regular.
- $Pre^*(R) = Pre^*(h^{-1}(m1)) \cup \cdots \cup Pre^*(h^{-1}(m_k))$.
- So we only need to prove that $Pre^*(h^{-1}(m_i))$ is regular.



- Bottom and top parts.
- Let $m_i = b_i a_i$
  Using monoids properties we can prove that
  Pre^* is regular if
  $P\bar{r}e^*(h^{-1}(b_i))$ is regular.

WQO

Well Quasi Order

For every infinite sequence a_i there exist $k < j$ such that $a_k \leq a_j$.

- A subword order abadaba.

Higman's lemma (more or less)

A subword order over words is WQO.

Product of WQO

A product of two WQOs orders is WQO.

A configuration is a tuple of words.

WQO

Upward-closed sets

If S is upward-closed then it has finite many minimal elements.

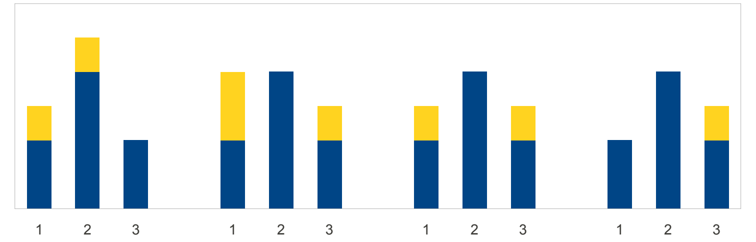
Observation

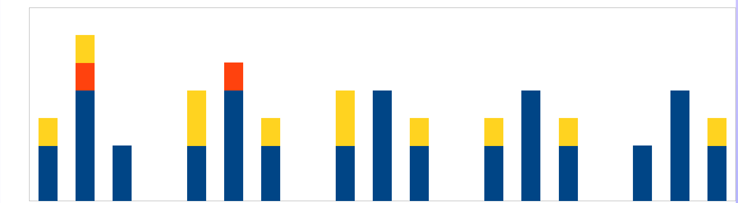
Upward closure of a word **aba** is $\Sigma^* a \Sigma^* b \Sigma^* a \Sigma^*$. So upward-closed sets are regular.

Proof

It suffice to show that $\overline{Pre}^*(h^{-1}(b_i))$ is upward closed!

$\bar{Pre}^*(h^{-1}(b_i))$





Future work

- *LTL* in *SLMSA*
- Sets which are not regular, or some specific class of sets like upward-closed.
- Case when there are stack symbols which can not be annihilated.
- Many states but with some order on them.
- Computing Pre^*.
- Many others questions...