The Complexity of the Membership Problem for Linear and Regular Permutation Languages

Grzegorz Madejski
Institute of Informatics,
University of Gdańsk,
gmadejsk@inf.ug.edu.pl

18 August 2015

Motivation

- relation to concurrency theory (similarity to shuffle languages),
- subclass of permutation languages called partially-commutative context-free languages (PCCFL) was studied in terms of finding a good substitute for the Process Algebra [Czerwinski, Lasota],
- relation with natural languages with a relatively free word order

Are permutation grammars parsable in polynomial time?

Definitions

- Parikh mapping. For example for alphabet $\{a, b, c, d\}$: $\Psi(abad) = (2, 1, 0, 1)$.
- Shuffle operation and shuffle closure: $ab \odot cd = \{abcd, acbd, acdb, cdab, cadb, ...\}$

Definitions

Let $G = (N, A, P, S)$ be a grammar.

- A context-free rule is of the form $X \to \alpha$, where $X \in N$, $\alpha \in (A \cup N)^*$.
- A linear rule is of the form $X \to uYv$ or $X \to u$, where $X, Y \in N$, $u, v \in A^*$.
- A regular rule is of the form $X \to uY$ or $X \to u$, where $X, Y \in N$, $u \in A^*$.
- A permutation (or interchange) rule is of the form $\alpha \to \beta$, where $\alpha, \beta \in (N \cup A)^* \, N \, (N \cup A)^*$, $\alpha \neq \beta$, $\Psi(\alpha) = \Psi(\beta)$. Example $Abc \to bAc$.

Definitions

We consider families of languages generated by permutation grammars:

- $PermCF$ = Context-Free Rules + Permutation Rules
- $PermLin$ = Linear Rules + Permutation Rules
- $PermReg$ = Regular (Right-Hand) Rules + Permutation Rules

$$PermReg \subseteq PermLin \subseteq PermCF.$$

Definitions

- Uniform membership problem:
  Input: a language $L \subseteq A^*$, a word $w \in A^*$,
  Question: $w \in L$?
  where A is an alphabet.
- Non-uniform membership problem for fixed $L \subseteq A^*$
  Input: a word $w \in A^*$,
  Question: $w \in L$?
- Uniform is harder!

Results

For *PermCF* the NP-completeness result was already known. We show the following:

Membership Problem	*PermLin*	*PermReg*
Non-Uniform	**NP-complete**	?
Uniform	**NP-hard**	**NP-hard**

We also show W[1]-hardness for *PermReg*.

The Non-uniform Membership Problem for *PermLin*

- Let $A = \{a, b, c, d, e, f\}$ be an alphabet. The language $L = \{ab^k cde^k f : k \geqslant 0\}^{\otimes}$ over A is known to be NP-complete [Warmuth, Haussler]. Words of the form: $ab^3cde^3f \odot ab^{10}cde^{10}f \odot acdf$.
- We show that it reduces to the language $L_\$ = \bigcup_{w \in L, |w|=n} w \odot \$^{n+1}$ over alphabet $A \cup \{\$\}$ which is in *PermLin*.

The Non-uniform Membership Problem for *PermLin*

- Why not showing directly that $L \in \mathit{PermLin}$?
- We want to shuffle $\boldsymbol{abbcdeef}$ into $abcdef$. We could however become $abc\boldsymbol{bbacdeef}def$.
- The permutation rules shuffle too freely!

The Non-uniform Membership Problem for *PermLin*

- In order to prevent that from happening, we add a special, guarding symbol \$ to the alphabet.
- Each time a letter from A is generated, it will come with the symbol \$ as a neighbor and only together will it be possible for this pair of symbols to be shuffled into already generated string of symbols.
- Then, after the shuffling of the new subword is done, the guarding symbols can be pushed to the sides of the configuration to prevent them from hindering the next steps of derivation process.

Let us illustrate these steps on an example.

The Non-uniform Membership Problem for *PermLin*

We want to shuffle ***abbcdeef*** into *abcdef* to get *a**ab**b**bc**c**d**d**e**e**ef**f*.

Generate the first subword.	*a\$b\$c\$\$d\$e\$f*
Push the \$-signs to the left.	$\$^6$*abcdef*
Generate the first letter of the new subword.	$\$^6$*abcdef* ***a\$***
Move it left, but do not pass any \$-signs.	$\$^6$*a**a\$**bcdef*
Generate the last letter of the new subword.	$\$^6$*a**a\$\$f**bcdef*
Move it right, but do not pass any \$-signs.	$\$^6$*a**a\$**bcde**\$f**f*
Generate the second letter of the new subword.	$\$^6$*a**a\$**bcde**b\$\$f**f*
Move it left, but do not pass any \$-signs.	$\$^6$*a**a\$b\$**bcde**\$f**f*

We repeat these steps for all letters of the new subword:

$\$^6$*a**a\$b\$**b**b\$c\$**c**\$d**d**\$e**e**\$e\$f**f*

and after that push the \$-signs to the left: $\$^{14}$*a**ab**b**bc**c**d**d**e**e**ef**f*.

The Non-uniform Membership Problem for *PermLin*

There are a few important observations to be made:

- The letters are generated in an alternating manner: the first one, the last one, the second one, the one before last and so on.
- The letters from the first half of the word (a, b, c) are always pushed left and they have their guarding symbol \$ on the right.
- The other letters (d, e, f) are pushed right and have their \$ on the left side.
- Such positions of the \$-signs are important: a letter *b* is moved left, but it cannot pass the letter *a* from the same subword. Therefore, *a* is protected with the \$ from the right.
- Imagine a worker sitting in a middle of the word, pushing letters left and right without being able to move past the letters he already placed in their position. This worker is actually a variable from a grammar that we now define.

The Non-uniform Membership Problem for *PermLin*

Now we construct a linear permutation grammar
$G_\$ = (\{S, X_1, X_2, X_3, X_4, X_5\}, A \cup \{\$\}, P, S)$ generating $L_\$$.

- The set P contains the following linear rules:

$$S \stackrel{0}{\rightarrow} \$, \quad S \stackrel{1}{\rightarrow} a\$X_1, \quad X_1 \stackrel{2}{\rightarrow} X_2\$f, \quad X_2 \stackrel{3}{\rightarrow} b\$X_3,$$

$$X_3 \stackrel{4}{\rightarrow} X_2\$e, \quad X_2 \stackrel{5}{\rightarrow} c\$X_4, \quad X_4 \stackrel{6}{\rightarrow} X_5\$d, \quad X_5 \stackrel{7}{\rightarrow} S,$$

- and for arbitrary $t \in A$ the following permutation rules:

$$St \stackrel{8}{\leftrightarrow} tS, \quad S\$ \stackrel{9}{\leftrightarrow} \$S, \quad tS\$ \stackrel{10}{\leftrightarrow} \$St,$$

$$ta\$X_1 \stackrel{11}{\rightarrow} a\$X_1t, \quad X_2\$ft \stackrel{12}{\rightarrow} tX_2\$f, \quad tb\$X_3 \stackrel{13}{\rightarrow} b\$X_3t,$$
$$X_2\$et \stackrel{14}{\rightarrow} tX_2\$e, \quad tc\$X_4 \stackrel{15}{\rightarrow} c\$X_4t, \quad X_5\$dt \stackrel{16}{\rightarrow} tX_5\$d.$$

The Non-uniform Membership Problem for *PermLin*

Now we construct a linear permutation grammar $G_\$ = (\{S, X_1, X_2, X_3, X_4, X_5\}, A \cup \{\$\}, P, S)$ generating $L_\$$.

- The set P contains the following linear rules:

$$S \xrightarrow{0} \$, \quad S \xrightarrow{1} a\$X_1, \quad X_1 \xrightarrow{2} X_2\$f, \quad X_2 \xrightarrow{3} b\$X_3,$$

$$X_3 \xrightarrow{4} X_2\$e, \quad X_2 \xrightarrow{5} c\$X_4, \quad X_4 \xrightarrow{6} X_5\$d, \quad X_5 \xrightarrow{7} S,$$

- and for arbitrary $t \in A$ the following permutation rules:

$$St \overset{8}{\leftrightarrow} tS, \quad S\$ \overset{9}{\leftrightarrow} \$S, \quad tS\$ \overset{10}{\leftrightarrow} \$St,$$

$$ta\$X_1 \xrightarrow{11} a\$X_1t, \quad X_2\$ft \xrightarrow{12} tX_2\$f, \quad tb\$X_3 \xrightarrow{13} b\$X_3t,$$

$$X_2\$et \xrightarrow{14} tX_2\$e, \quad tc\$X_4 \xrightarrow{15} c\$X_4t, \quad X_5\$dt \xrightarrow{16} tX_5\$d.$$

The Non-uniform Membership Problem for *PermLin*

Let us look briefly at how the word $w = \$^{14}\boldsymbol{a}b\boldsymbol{b}b\boldsymbol{c}c\boldsymbol{d}d\boldsymbol{e}e\boldsymbol{e}\boldsymbol{f}f$ from previous example is derived:

- $S \overset{1-6}{\Longrightarrow} \ldots \overset{1-6}{\Longrightarrow} a\$b\$c\$X_5\$d\$e\$f \overset{7}{\Longrightarrow} a\$b\$c\$S\$d\$e\$f \overset{8-10}{\Longrightarrow} \ldots$
- $\overset{8-10}{\Longrightarrow} \$^6 abcdefS \overset{1}{\Longrightarrow} \$^6 abcdef\,\boldsymbol{a\$X_1} \overset{11}{\Longrightarrow} \ldots \overset{11}{\Longrightarrow}$ $\$^6 a\boldsymbol{a\$X_1}bcdef \overset{2}{\Longrightarrow}$
- $\overset{2}{\Longrightarrow} \$^6 a\boldsymbol{a\$X_2\$f}bcdef \overset{12}{\Longrightarrow} \ldots \overset{12}{\Longrightarrow} \$^6 a\boldsymbol{a\$}bcde\boldsymbol{X_2\$f}f \overset{3}{\Longrightarrow}$
- $\overset{3}{\Longrightarrow} \$^6 a\boldsymbol{a\$}bcde\boldsymbol{b\$X_3\$f}f \overset{13}{\Longrightarrow} \ldots \overset{13}{\Longrightarrow} \$^6 a\boldsymbol{a\$b\$X_3}bcde\boldsymbol{\$f}f \Longrightarrow$
- $\ldots \Longrightarrow \$^6 a\boldsymbol{a\$b\$}b\boldsymbol{b\$c\$}c\boldsymbol{S\$d}d\boldsymbol{\$e}e\boldsymbol{\$e\$f}f \overset{8-10}{\Longrightarrow} \ldots \overset{0}{\Longrightarrow} w.$

The Non-uniform Membership Problem for *PermLin*

We need to show that the grammar $G_\$$ indeed generates the language $L_\$$.

Lemma

$L_\$ = \mathcal{L}(G_\$)$

THe proof $\subseteq$ is easy to follow, when we consider the above examples. The proof $\supseteq$ is inductive over the number of "loops" in the derviation. It is shown that after arbitrary number of loops, the derived word is always in $L_\$$.

Theorem

$L_\$ \in$ PermLin is an NP-Complete language.

The Uniform Membership Problem for *PermReg*

We show that the uniform membership problem for *PermReg* is NP-hard.
Input: a grammar G, a word w,
Question: $w \in \mathcal{L}(G)$?
We do this by a reduction from the k-Clique problem, which is known to be NP-complete.
Input: a graph H, a number k
Question: Does graph H have a k-clique as subgraph?

The Uniform Membership Problem for *PermReg*

- The graph $H = (V_H, E_H)$ (where $|V_H| = n$, $|E_H| = m$) is undirected and we will not encode it as a standard neighbors list, but as a string $V'_H \# E'_H$, where $V'_H = v_1 v_2 \cdots v_n$ is a string containing all vertices in lexicographical order and $E'_H = \bigotimes_{v_i v_j \in E_H, i<j} v_i v_j \$$ is a string of edges sorted in lexicographical order and separated with the \$-signs.
- We assume that the alphabet is proportional to the number of vertices. One could encode each vertex as a binary number enabling the alphabet to be of constant size. This has a slight impact on the complexity. However, for the sake of clarity, we consider each vertex symbol to be a single sign.

The Uniform Membership Problem for *PermReg*

We construct a grammar $G_H = (N, A, P, S)$, where $N = \{S, X_1, X_2, ..., X_k, Y,$ $T_1, T_2, ..., T_{m-\frac{k(k-1)}{2}}\}$, $A = \{v_1, v_2, ..., v_n, \$\}$ and the set P contains the following rules:

- the starting rule,

$$S \rightarrow \$^{\frac{k(k-1)}{2}} X_1$$

- the X-rules, which guess the vertices of the k-clique and generate $k-1$ of copies for each vertex:

$$X_1 \rightarrow v_1^{k-1} X_2 \mid v_2^{k-1} X_2 \mid ... \mid v_n^{k-1} X_2,$$

$$X_2 \rightarrow v_1^{k-1} X_3 \mid v_2^{k-1} X_3 \mid ... \mid v_n^{k-1} X_3,$$

$$...$$

$$X_k \rightarrow v_1^{k-1} Y \mid v_2^{k-1} Y \mid ... \mid v_n^{k-1} Y,$$

The Uniform Membership Problem for *PermReg*

- the Y-permutations, which reorder all symbols in an arbitrary manner:

$$a_1 a_2 Y \to a_2 a_1 Y, \quad a_1 Y \leftrightarrow Y a_1, \text{ where } a_1, a_2 \in A, a_1 \neq a_2,$$

- the T-rules, which generate all edges outside of k-clique:

$$Y \to T_1,$$

$$T_1 \to v_i v_j \$ T_2,\ T_2 \to v_i v_j \$ T_3,\ \dots,\ T_{m-\frac{k(k-1)}{2}} \to v_i v_j \$$$

where $1 \leqslant i < j \leqslant n$ and $v_i v_j \in E_H$.

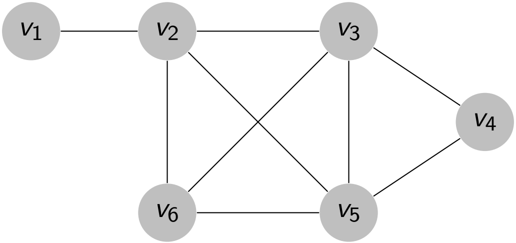
- the T-permutations, which allow to put the missing edges in correct places

$$T_l v_i v_j \$ \to v_i v_j \$ T_l, \text{ where } l \in \{1, 2, \dots, m - \frac{k(k-1)}{2}\},$$

$$1 \leqslant i < j \leqslant n, v_i v_j \in E_H$$

The Uniform Membership Problem for *PermReg*

Suppose we have the following graph and we ask whether it contains a 4-clique.



The Uniform Membership Problem for *PermReg*

The derivation consists of three major steps:

- We can nondeterministically guess that the 4-clique consists of vertices v_2, v_3, v_5, v_6. Using the X-rules we generate a string $\$^6 v_2^3 v_3^3 v_5^3 v_6^3 Y$.
- Using the Y-permutations, we reorder the string to contain 6 edges that form the 4-clique:
  $Y v_2 v_3 \$ v_2 v_5 \$ v_2 v_6 \$ v_3 v_5 \$ v_3 v_6 \$ v_5 v_6 \$$.
- We can generate all the 3 missing edges (bold font) which are not part of the k-clique. This is done with T-rules and T-permutations:
  $\boldsymbol{T_1} v_2 v_3 \$ v_2 v_5 \$ v_2 v_6 \$ v_3 v_5 \$ v_3 v_6 \$ v_5 v_6 \$ \Rightarrow$
  $\boldsymbol{v_1 v_2 \$ T_2} v_2 v_3 \$ v_2 v_5 \$ v_2 v_6 \$ v_3 v_5 \$ v_3 v_6 \$ v_5 v_6 \$ \Rightarrow \ldots \Rightarrow$
  $\boldsymbol{v_1 v_2 \$} v_2 v_3 \$ v_2 v_5 \$ v_2 v_6 \$ \boldsymbol{T_2} v_3 v_5 \$ v_3 v_6 \$ v_5 v_6 \$ \Rightarrow$
  $\boldsymbol{v_1 v_2 \$} v_2 v_3 \$ v_2 v_5 \$ v_2 v_6 \$ \boldsymbol{v_3 v_4 \$ T_3} v_3 v_5 \$ v_3 v_6 \$ v_5 v_6 \$ \Rightarrow \ldots \Rightarrow$
  $\boldsymbol{v_1 v_2 \$} v_2 v_3 \$ v_2 v_5 \$ v_2 v_6 \$ \boldsymbol{v_3 v_4 \$} v_3 v_5 \$ v_3 v_6 \$ \boldsymbol{T_3} v_5 v_6 \$ \Rightarrow$
  $\boldsymbol{v_1 v_2 \$} v_2 v_3 \$ v_2 v_5 \$ v_2 v_6 \$ \boldsymbol{v_3 v_4 \$} v_3 v_5 \$ v_3 v_6 \$ \boldsymbol{v_4 v_5 \$} v_5 v_6 \$$

The Uniform Membership Problem for *PermReg*

The main results:

Lemma

Let $H = (V_H, E_H)$ be an undirected graph and $k \geqslant 1$. Then: H contains a k-clique $\Leftrightarrow E'_H \in \mathcal{L}(G_H)$.

Theorem

The uniform membership problem for PermReg is NP-hard.

The Parameterized Complexity of *PermReg*

- the runtime of the above problems is measured in terms of the input size only
- is the runtime is polynomial in the input size and exponential or worse in a parameter k
- we set k: the number of occurrences of permutation rules in derivation
- with time bounded by a function $f(k)p(|input|)$, for some polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$ and computable function $f : \mathbb{N} \rightarrow \mathbb{N}$ this problem would be fixed-parameter tractable

The Parameterized Complexity of *PermReg*

Theorem

The parameterized uniform membership problem for PermReg with the number of applications of permutation rules set as parameter is $W[1]$*-hard.*

Open problems

- *PermReg* - non-uniform membership problem
- any other restrictions on permutation rules to make the problem polynomial?
- parameterized complexity - other parameteres?
- NP-hardness was proved for uniform problem, what about completeness?
- other problems - decidability and complexity

The End

Thank you!