State Complexity of Permutation and Related Decision Problems on Alphabetical Pattern Constraints

Stefan Hoffmann

University of Trier

International Conference on Implementation and Application of Automata (CIAA) 2021,
July 19 – July 22, 2021 (online)
Organizers: Sebastian Maneth, Peter Leupold, Kathryn Lorenz, Martin Vu

Basic Notions

1. $[n] = \{1, \ldots, n\}$.
2. Non-deterministic automata (NFAs).
3. Deterministc partial automata (PDFA).
4. Partially ordered (pto) NFAs: The reachability relation induced by the words is a partial order.
   Equivalently, the only loops and cycles are self-loops.
   (removing self-loops yields a directed acyclic graph)

Basic Notions

1. $[n] = \{1, \ldots, n\}$.
2. Non-deterministic automata (NFAs).
3. Deterministc partial automata (PDFA).
4. Partially ordered (pto) NFAs: The reachability relation induced by the words is a partial order.
   Equivalently, the only loops and cycles are self-loops.
   (removing self-loops yields a directed acyclic graph)

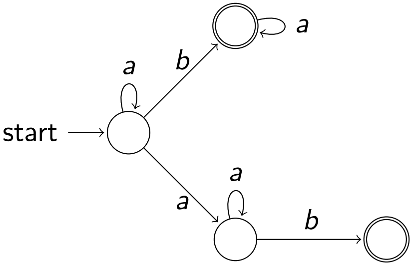
Remark

pto NFAs are strictly more expressive than pto PDFAs. (example, famous NFA/DFA tradeoff language: $\{a, b\}^* a\{a, b\}^n$, or $\{a, b\}^* aa^*$)

Basic Notions

Let $\mathcal{A}$ be an NFA.

1. A labeled path in $\mathcal{A}$ is called simple, if the states read after each prefix are distinct, i.e., no state is repeated along the path.
2. $L^{\text{simple}}(\mathcal{A}) = \{w \text{ labels a simple accepting path in } \mathcal{A}\}$.



A pto NFA $\mathcal{A}$ with $L^{\text{simple}}(\mathcal{A}) = \{b, ab\}$.

Shuffle Operation

Definition (Shuffle operation)

The shuffle operation, denoted by $\sqcup\!\sqcup$, is defined by

$$u \sqcup\!\sqcup v := \left\{ x_1 y_1 x_2 y_2 \cdots x_n y_n \;\middle|\; \begin{array}{l} u = x_1 x_2 \cdots x_n,\, v = y_1 y_2 \cdots y_n, \\ x_i, y_i \in \Sigma^*, 1 \leq i \leq n, n \geq 1 \end{array} \right\},$$

for $u, v \in \Sigma^*$ and $L_1 \sqcup\!\sqcup L_2 := \bigcup_{x \in L_1, y \in L_2} (x \sqcup\!\sqcup y)$ for $L_1, L_2 \subseteq \Sigma^*$.

Shuffle Operation

Definition (Shuffle operation)

The shuffle operation, denoted by $\sqcup\!\sqcup$, is defined by

$$u \sqcup\!\sqcup v := \left\{ x_1 y_1 x_2 y_2 \cdots x_n y_n \;\middle|\; \begin{array}{l} u = x_1 x_2 \cdots x_n,\, v = y_1 y_2 \cdots y_n, \\ x_i, y_i \in \Sigma^*, 1 \le i \le n, n \ge 1 \end{array} \right\},$$

for $u, v \in \Sigma^*$ and $L_1 \sqcup\!\sqcup L_2 := \bigcup_{x \in L_1, y \in L_2} (x \sqcup\!\sqcup y)$ for $L_1, L_2 \subseteq \Sigma^*$.

$\{ab\} \sqcup\!\sqcup \{cd\} = \{abcd, acbd, acdb, cadb, cdab, cabd\}$

State Complexity of Operations

1. State complexity of a regular language: size of a minimal deterministic automaton for the language.
2. State complexty of a regularity-preserving operation: greatest state complexity of the result of this operation (usually measured in terms of the state complexities of the input languages).

State Complexity of Operations

1. State complexity of a regular language: size of a minimal deterministic automaton for the language.
2. State complexty of a regularity-preserving operation: greatest state complexity of the result of this operation (usually measured in terms of the state complexities of the input languages).
3. Classically investigated for deterministic (partial) automata. But has been also investigated for non-deterministic automata, 2-way automata models, unambiguous automata, alternating automata.
4. Example: bounds nm are tight for union/intersection, bound $2^{n-1} + 2^{n-2}$ for Kleene star etc .

Commutative Closure and APCs

Straubing-Thérien Hierarchy (Straubing 1981, Thérien 1981)

Start with $\{\emptyset, \Sigma^*\}$ and build alternately finite unions of marked products $L_0 a_1 L_1 \cdots a_n L_n$ with $L_1, \cdots, L_n$ from the previous level (half-levels) or the boolean closure of the previous level (full levels).

Example

Level 1/2: $\Sigma^* a \Sigma^* b \Sigma^* \cup \Sigma b \Sigma^*$.

Level 1: $\Sigma^* \setminus (\Sigma^* a \Sigma^*)$.

Level 3/2: $\{a\}^* a \{b, c\}^* a \{c\}^* \cup \{a, b\}^* c$.

Commutative Closure and APCs

Straubing-Thérien Hierarchy (Straubing 1981, Thérien 1981)

Start with $\{\varnothing, \Sigma^*\}$ and build alternately finite unions of marked products $L_0 a_1 L_1 \cdots a_n L_n$ with $L_1, \cdots, L_n$ from the previous level (half-levels) or the boolean closure of the previous level (full levels).

Example

Level 1/2: $\Sigma^* a \Sigma^* b \Sigma^* \cup \Sigma b \Sigma^*$.

Level 1: $\Sigma^* \setminus (\Sigma^* a \Sigma^*)$.

Level 3/2: $\{a\}^* a \{b, c\}^* a \{c\}^* \cup \{a, b\}^* c$.

Definition (Bouajjani, Muscholl & Touili 2007)

The languages from level $3/2$ are called Alphabetical Pattern Constraints (APCs).

Remark (Schwentick, Thérien & Vollmer 2001)

Partially ordered NFAs characterize APCs.

Commutative Closure and APCs

The commutative closure is regularity-preserving on APCs. In fact, as

$$\mathrm{perm}(\Sigma_0^* a_1 \Sigma_1^* \cdots a_n \Sigma_n^*) = \mathrm{perm}(a_1 \cdots a_n) \sqcup\!\sqcup (\Sigma_0 \cup \ldots \cup \Sigma_n)^*.$$

and

$$u \sqcup\!\sqcup \Gamma^* = (u \sqcup\!\sqcup \Sigma^*) \cap \overline{\bigcup_{a \in \Sigma \setminus \Gamma} \mathrm{perm}(va) \sqcup\!\sqcup \Sigma^*}$$

for $\Gamma \subseteq \Sigma$, the commutative closure is a level one language.

Remark

We have $(ab)^* = (a\Sigma^* \cap \Sigma^* b) \setminus (\Sigma^* aa \Sigma^* \cup \Sigma^* bb \Sigma^*)$ *and*

$$\mathrm{perm}((ab)^*) = \{ \text{ words with equal number of } a\text{'s and } b\text{'s } \}.$$

So, for the next (full) level of the Straubing-Thérien hierarchy, the commutative closure is not regularity-preserving.

Commutative Closure and APCs

Theorem: Let L be an APC recognized by a pto NFA $\mathcal{A}$. Then, $\mathrm{perm}(L)$ is recognizable by a PDFA of size at most

$$\prod_{a \in \Sigma} (\max\{|u|_a : u \in L^{\mathrm{simple}}(\mathcal{A})\} + 1)$$

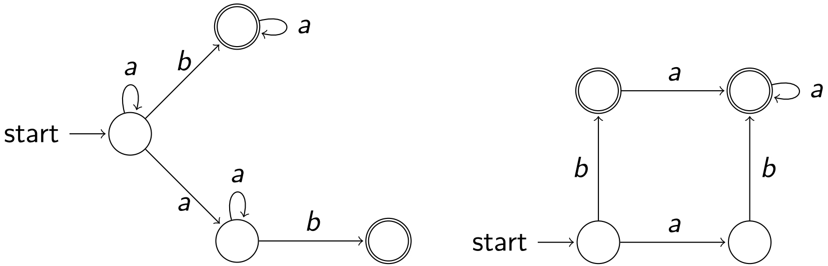
$\Sigma = \{a_1, \ldots, a_k\}$.

1. Set $n_j = \max\{|u|_{a_j} \mid u \in L^{\mathrm{simple}}(\mathcal{A})\} + 1$, $j \in \{1, \ldots, k\}$.
2. PDFA for $\mathrm{perm}(L)$: $\mathcal{B} = (\Sigma, Q, \delta, q_0, F)$ with $Q = [n_1 + 1] \times \ldots \times [n_k + 1]$ and $\delta((s_1, \ldots, s_k), a_j)$ equals

$$\begin{cases} (s_1, \ldots, s_{j-1}, s_j + 1, s_{j+1}, \ldots, s_k) & \text{if } s_j < n_j \\ (s_1, \ldots, s_k) & \text{if } s_j = n_j, a_1^{s_1} \cdots a_k^{s_k} a_j \in \mathrm{perm}(\mathrm{Pref}(L)). \end{cases}$$

3. $q_0 = (0, \ldots, 0)$,
4. $F = \{\delta(q_0, w) \mid w \in L \text{ and } \forall j \in \{1, \ldots, k\} : |w|_{a_j} \le n_j\}$. $\square$

Proof Example



A pto NFA $\mathcal{A}$ with $L^{\text{simple}}(\mathcal{A}) = \{b, ab\}$.

$$\text{perm}(L(\mathcal{A})) = \text{perm}(a^* ba^* \cup a^* ab) = \{a, b\}^* b\{a, b\}^*.$$

Commutative Closure and APCs

Corollary

Let L be an APC recognized by a partially ordered NFA with n states. Then, $\mathrm{perm}(L)$ is recognizable by a PDFA with at most $n^{|\Sigma|}$ many states.

Unknown if this bound is tight.

Strict Shuffle Languages

Let $L \subseteq \Sigma^*$. If $L = \bigsqcup_{a \in \Sigma} \{a^{|u|_a} \mid u \in L\}$, then we call it a strict shuffle language.
(equivalently, $L = \pi_{a_1}(L) \sqcup \ldots \sqcup \pi_{a_k}(L)$ for one-letter projection languages and $\Sigma = \{a_1, \ldots, a_k\}$)

Example

1. If $u \in \Sigma^*$, then $\mathrm{perm}(u)$ is a strict shuffle language.
2. The language
$\{u \in \{a,b\}^* \mid |u|_a = 1 \text{ and } 2 \le |u| \le n\} = \{a\} \sqcup \{b, b^2, \ldots, b^{n-1}\}$
is a strict shuffle language.
3. $\mathrm{perm}(\{aabb, ab\})$ is not a strict shuffle language.
4. $\mathrm{perm}(\{aaabbb, abbb, aaab, ab\})$ is a strict shuffle language.

Strict Shuffle Languages

Theorem: Let L be an APC language recognized by a pto NFA $\mathcal{A}$ with n states s.t. $\mathrm{perm}(L^{\mathrm{simple}}(\mathcal{A}))$ is a strict shuffle language. Then, $\mathrm{perm}(L)$ is recognizable by a PDFA with

$$\left\lceil \frac{n-1}{|\Sigma|} + 1 \right\rceil^{|\Sigma|}$$

many states and this bound is sharp even for finite languages.

1. Aut. for $L^{\mathrm{simple}}(\mathcal{A})$ at least $(\sum_{a \in \Sigma} \max\{|u|_a : u \in \pi_a(L)\}) + 1$ states.

$\Rightarrow$ So $0 \leq (\sum_{a \in \Sigma} \max\{|u|_a : u \in \pi_a(L)\}) + 1 \leq n$.

2. The value $\prod_{a \in \Sigma}(\max\{|u|_a : u \in L^{\mathrm{simple}}(\mathcal{A})\} + 1)$ with $0 \leq (\sum_{a \in \Sigma} \max\{|u|_a : u \in \pi_a(L)\}) + 1 \leq n$ is maximized if $\max\{|u|_a : u \in L^{\mathrm{simple}}(\mathcal{A})\}$ equals $(n-1)/|\Sigma|$ for every $a \in \Sigma$.

Strict Shuffle Languages

Corollary

Let $L = \Sigma_0^* a_1 \Sigma_1^* a_2 \cdots a_m \Sigma_m^*$. *Then,* $\mathrm{perm}(L)$ *is recognizable by a PDFA with at most* $\lceil m/|\Sigma| + 1 \rceil^{|\Sigma|}$ *many states. In particular, the commutative closure of a single word* u *could be recognized by a PDFA with at most* $\lceil |u|/|\Sigma| + 1 \rceil^{|\Sigma|}$ *many states and this bound is sharp.*

Remark (Sharpness)

$L = \{a_1^m \cdots a_k^m\}$. *Recognizable by pto NFA with* $km + 1$ *states, PDFA for commutative closure needs* $(m+1)^k$ *states.*

Lemma

For a given partially ordered NFA $\mathcal{A}$ an APC expression of $L(\mathcal{A})$ could be computed in P and for every APC expression a partially ordered NFA is computable in P. This result also holds for variable input alphabets.

Proposition

Given a partially ordered NFA $\mathcal{A}$ with n states, the recognizing PDFA for $\mathrm{perm}(L(\mathcal{A}))$ from above could be constructed in polynomial time for a fixed alphabet. More precisely in time $O(n^{|\Sigma|+2})$.

Computational Complexity

Theorem

Fix an alphabet Σ. Then, the following problem is in P:
Input: Two APC expressions L_1, L_2 over Σ^*.
Question: Is $\mathrm{perm}(L_1) \subseteq \mathrm{perm}(L_2)$?

Proof.

1. Construct two NFAs $\mathcal{A}_1$ and $\mathcal{A}_2$ for L_1 and L_2, in P by previous Lemma.
2. Compute PDFAs $\mathcal{B}_1$ and $\mathcal{B}_2$ for their respective commutative closures, which could be done in P.
3. For PDFAs,

$$L(\mathcal{B}_1) \subseteq L(\mathcal{B}_2) \Leftrightarrow L(\mathcal{B}_1) \cap \overline{L(\mathcal{B}_2)} = \emptyset.$$

could be done in P. □

Computational Complexity

Corollary

Fix an alphabet Σ. Then, the following problem is in P:
Input: An APC expression L over Σ^*.
Question: Is $\mathrm{perm}(L) = \Sigma^*$?

Corollary

Fix an alphabet Σ. Given an APC describing a commutative language, the universality problem is in P. Also, given two APCs describing commutative languages, the inclusion problem is solvable in polynomial time.

Thank you for your attention!

Thanks to the organizers Sebastian Maneth, Peter Leupold, Kathryn Lorenz and Martin Vu!

Hopefully next time in person! ☺