A pumping-like lemma for languages over infinite alphabets

Yoav Danieli

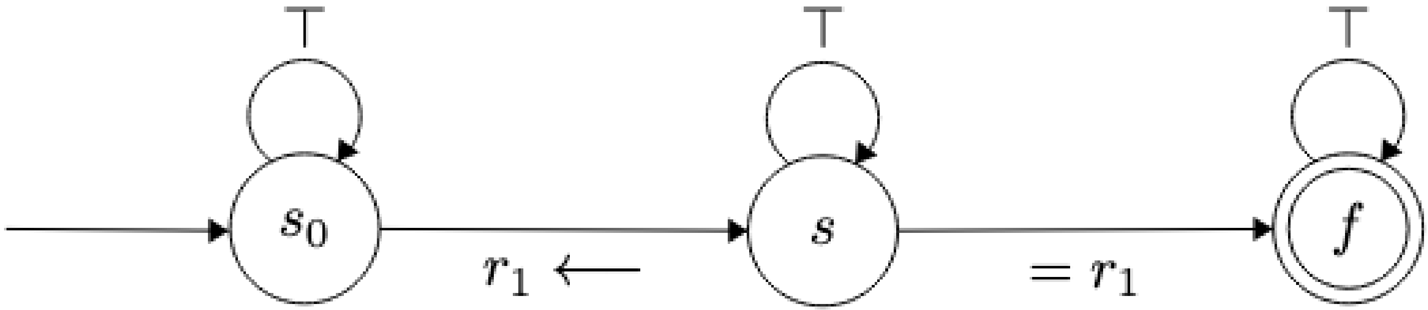
Technion – Israel Institute of Technology, Haifa, Israel

Finite-memory automata (FMA)

- Finite-state machines equipped with a finite set of registers.
- Transitions depends on the *current state* and the comparison of the *input value* to the *contents of the registers*, then the registers may be updated and the state changed.
- The class of languages they recognize is known as quasi-regular languages.

Examples

- $L_{eq} = \{\sigma_1\sigma_2 \cdots \sigma_n : n \geq 2, \exists i < j, \sigma_i = \sigma_j\}$.
- ‘there is some symbol that occurs at least twice’.

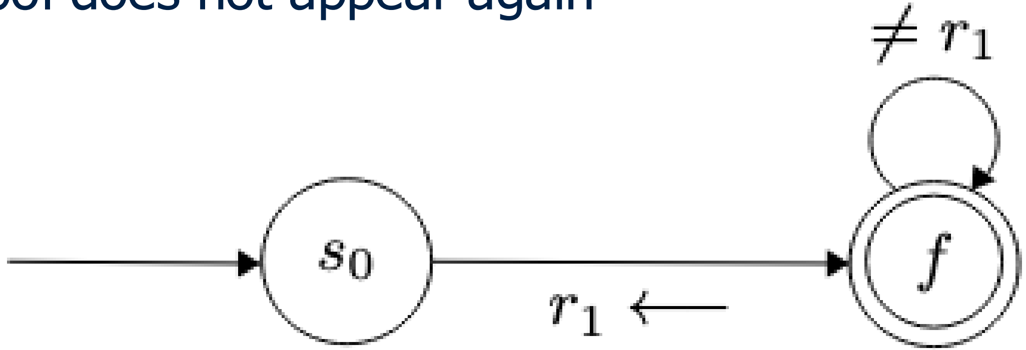


- Its complement is
- $L_{diff} = \{\sigma_1\sigma_2 \cdots \sigma_n : \forall i < j, \sigma_i \neq \sigma_j\}$.

Is not recognized by FMA

Examples

- $L_{first} = \{\sigma_1\sigma_2 \cdots \sigma_n : n \geq 2, \forall 1 < i \leq n, \sigma_i \neq \sigma_1\}$.
- 'the first symbol does not appear again'



- Its reversal is
- $L_{last} = \{\sigma_1\sigma_2 \cdots \sigma_n : n \geq 2, \forall 1 \leq i < n, \sigma_i \neq \sigma_n\}$.

Is not recognized by FMA

TECHNION
Israel Institute of Technology

Regular properties of quasi-regular languages

	The restriction to any finite alphabet is a regular language.
	Closure under union, intersection, concatenation, and Kleene star.

	Not closed under comlementation and reversal.
	Non-emptiness is NP-complete.
	Universality is undecidable.

TECHNION
Israel Institute of Technology

Toolbox for quasi-regular languages

- For deterministic quasi-regular languages there is a Myhill-Nerode characterization using orbit-finite set theory.
- Is there a pumping lemma for quasi-regular languages?
- Yes, but with renaming.

Pumping lemma for infinite alphabets

- Let $\pi: \Sigma \to \Sigma$ be a permutation, the order of π is the minimal integer d such that $\pi^d = id$.
- Infinite-alphabets pumping lemma: Let L be a quasi-regular language. There is a constant N such that, for every word $z \in L$ with $|z| > N$ there is a decomposition $z = \tau \upsilon \varphi$ and a finite-order permutation π such that:

1. $|\upsilon| > 0$.
2. $|\tau\upsilon| \leq N$.
3. $\tau\pi^{-1}(\varphi) \in L$.
4. For all $k = 1,2, \dots,$

$$\tau\upsilon\pi(\upsilon)\pi^2(\upsilon)\cdots\pi^k(\upsilon)\pi^k(\varphi) \in L.$$

A strong extension of finite-memory automata: alternating automata with one register ($AFMA_1$)

- Adding universal and existential branching.
- Is closed under complementation.
- Has decidable non-emptiness!.
- Can recognize
  - $L_{diff} = \{\sigma_1\sigma_2 \cdots \sigma_n : \forall i < j, \sigma_i \neq \sigma_j\}$.
  - $L_{last} = \{\sigma_1\sigma_2 \cdots \sigma_n : n \geq 2, \forall 1 \leq i < n, \sigma_i \neq \sigma_n\}$.
  - $L_{\subseteq} = \{\psi_1\#\psi_2 : \psi_1, \psi_2 \in L_{diff}, [\psi_1] \subseteq [\psi_2] \subseteq \Sigma \setminus \{\#\}\}$.

A well-studied model

- Non-emptiness is decidable [Demri, Lazić; LICS’06]
- The determinization problem is decidable [Clemente, Lasota, Piórkowski; LMCS’22]
- Subsumes decidable pebble automata [Tan; MFCS’09]
- Tight connection to timed automata with one clock [Figueira, Hofman, Lasota; MSCS’16]

No toolbox for $AFMA_1$

- Given a language $L \subseteq \Sigma^*$, can we determine if it is not accepted by any alternating automata with one register?
- For example:
  - $L_{\supseteq} = \{\psi_1 \# \psi_2 \colon \psi_1, \psi_2 \in L_{diff}, \Sigma \setminus \{\#\} \supseteq [\psi_1] \supseteq [\psi_2]\}$.
  - $L_{one} = \{\psi_1 a \psi_2 \colon a \notin [\psi_1 \psi_2]\}$.
- Is there a pumping lemma for languages recognized by $AFMA_1$?

Pumping lemma for $AFMA_1$?

- Can we extend a word repeatedly by the same permutation?
- For $L_{diff} = \{\sigma_1\sigma_2 \cdots \sigma_n : \forall i < j, \sigma_i \neq \sigma_j\}$, any finite-order permutation would not work, only infinite-order permutations.
- For $L_{\subseteq} = \{\psi_1 \# \psi_2 : \psi_1, \psi_2 \in L_{diff}, [\psi_1] \subseteq [\psi_2]\}$, the pumping cannot occur in the prefix, only in the suffix.

Pumping lemma for $AFMA_1$!

- Theorem: Let L be the language of an $AFMA_1$ A. There is a computable constant N_A such that, for every word $z \in L$ with $|z| > N_A$ there is a decomposition $z = \tau\upsilon\varphi$ and a permutation π such that:

1. $|\upsilon| > 0$,
2. $|\upsilon\varphi| \leq N_A$,
3. For all $k = 1,2,\ldots,$

$$\tau\upsilon\pi(\upsilon)\pi^2(\upsilon)\cdots\pi^k(\upsilon)\pi^k(\varphi) \in L.$$

Review of decidability of non-emptiness

- A run of A on a word $w_1 w_2 \cdots w_n$

$$C_0 \xrightarrow{w_1} C_1 \xrightarrow{w_2} C_2 \rightarrow \cdots C_{n-1} \xrightarrow{w_n} C_n$$

Review of decidability of non-emptiness

- A partial order on finite sets of configurations $C \leq C'$:
  - If for all nonempty subsets of states $P \subseteq Q$, the number of symbols that are in exactly the states P in C is less than those in C':
  $$|\Sigma^P(C)| \leq |\Sigma^P(C')|.$$
  - Equivalently, there is a finite-order permutation α such that:
  $$\alpha(C) \subseteq C'.$$
- This is a well-quasi-order, i.e., no infinite anti-chains.

Review of decidability of non-emptiness

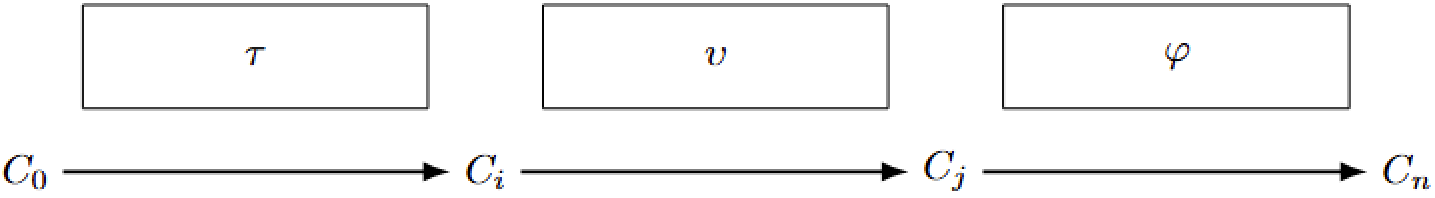
- A run of A on a word $w_1 w_2 \cdots w_n$

$$C_0 \xrightarrow{w_1} C_1 \xrightarrow{w_2} C_2 \rightarrow \cdots C_{n-1} \xrightarrow{w_n} C_n$$

- For sufficiently large n, there are $i < j$ such that

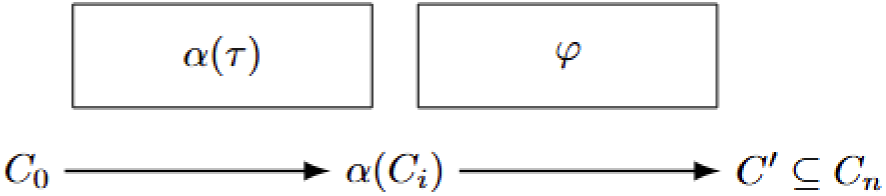
$$C_i \leq C_j$$

Review of decidability of non-emptiness



$$C_i \leq C_j, \quad (\alpha(C_i) \subseteq C_j)$$

$$\Downarrow$$



Towards pumping lemma

- We wish to extend the word, not shorten it.
- To obtain the pumping lemma we need the computation to satisfy the inverse relation, namely $i < j$ with $C_i \geq C_j$.
- However, there is no guarantee for that!
- 0,1,2,3,4,…

Key insight

- If a word is accepted from C after n transitions, then the acceptance depends only on the symbols in the suffix of length n.
- Therefore, it suffice to prune each configuration to only the symbols of the suffix that is yet to be read.

New partial order

- We define a partial order on sets of configurations with respect to different domains

$$C_1, \Sigma_1 \leq C_2, \Sigma_2.$$

- It is also a well-quasi-order.

Backward computation sequence

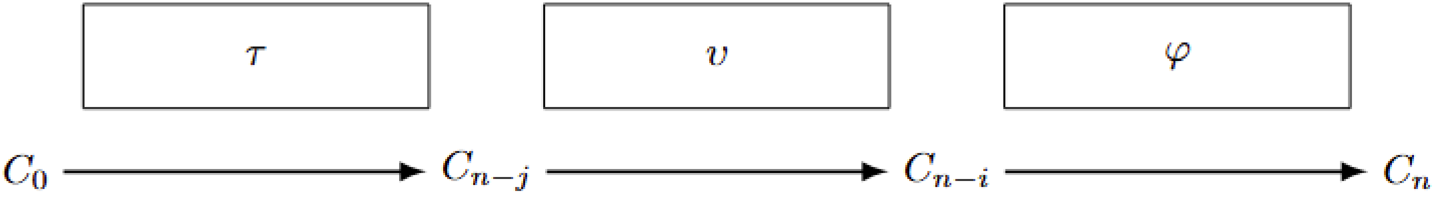
- A run of A on a word $w_1 w_2 \cdots w_n$

$$C_0 \xrightarrow{w_1} C_1 \xrightarrow{w_2} C_2 \rightarrow \cdots C_{n-1} \xrightarrow{w_n} C_n$$

- The backwards computation with the respected suffix-domains is:

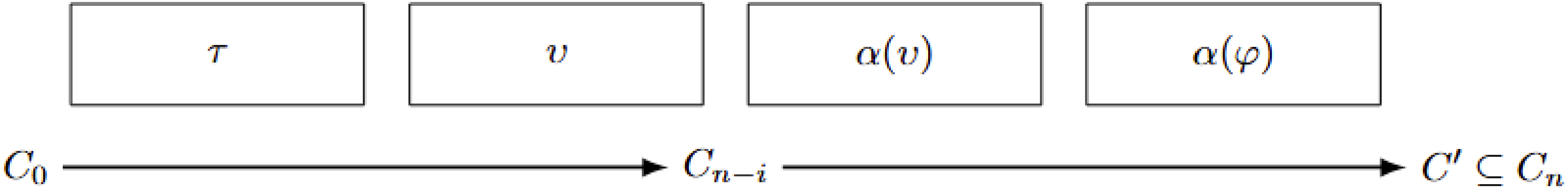
$$(C_n, \emptyset), (C_{n-1}, [w_n]), (C_{n-2}, [w_{n-1} w_n]), \ldots$$

Pumping in one step



$$C_{n-j}, [\upsilon\varphi] \geq C_{n-i}, [\varphi]$$

$$\Downarrow$$



Some applications

- The pumping lemma can be directly used to show that the following languages are not accepted by $AFMA_1$:
  - $L_{\supseteq} = \{\psi_1 \# \psi_2 : \psi_1, \psi_2 \in L_{diff}, \Sigma \setminus \{\#\} \supseteq [\psi_1] \supseteq [\psi_2]\}$.
  - $L_{one} = \{\psi_1 a \psi_2 : a \notin [\psi_1 \psi_2]\}$.
- Some more direct corollaries from our pumping lemma:
- Corollary: The class of languages recognizable by $AFMA_1$ is **not** closed under reversal, concatenation, nor Kleene star.
- Corollary: The set of lengths of a language of an $AFMA_1$ is semi-linear.

Thank you!