Computability of the emptiness problem for alternating finite-memory automata

Seminar by Yevgeni Longo and Nathan Rodrigue

Based on the article “A note on the emptiness problem for alternating finite-memory automata”.
Section 1: intro
What is an alternating finite-memory automata

Finite-memory automata is an automata with an infinite set of alphabetical symbols and an option for registers that store values during the computation.

The languages that are accepted by these automata are called quasi-regular.

Automata is called “alternating” if its transition function can either choose all the matching transitions non-deterministically, or choose one of the matching transitions deterministically (even if multiple transitions are possible).

This paper is limited to automata with a single register.
We shall prove the theorem:
The emptiness problem for alternating finite-memory automata with single register is decidable
The Idea

we show that, if the language of a one-register alternating finite-memory automata is nonempty, then it contains a “short” word whose length is computable.
Section 2: Alternating finite-memory automata

In this section we will call the definition of alternating finite-memory automata and prove its basic properties needed for the proof of Theorem 1.

Throughout this paper we use the following conventions:

- \( \Sigma \) is a fixed infinite alphabet.
- Symbols in \( \Sigma \) are denoted by \( \sigma \) (sometimes indexed or primed), \( \tau \), or \( \delta \).
- \( w, w', \text{ and } w'' \) denote words over \( \Sigma \).
- \( \sigma \) denotes a letter.
3. Automata definition

$S$ - finite set of states

$s_0 \in S$ - initial state

$F \subseteq S$ - set of accepting states

$\Delta \subset \Sigma$ - a finite set of distinguished symbols

$\delta \in \Sigma \setminus \Delta$ - the initial window assignment
Transition function $\mu_c$:
Given 3 transition functions $\mu_\Delta$, $\mu_\tau$, and $\mu_\neq$ define $\mu_c$ to be
if $\sigma \in \Delta$: $\mu_c((s,\tau),\sigma) = \{Q \times \{\tau\}: Q \in \mu_\Delta(s,\sigma)\}$
if $\sigma = \tau$: $\mu_c((s,\tau),\sigma) = \{Q \times \{\tau\}: Q \in \mu_\tau(s)\}$
if $\sigma \notin \Delta \cup \{\tau\}$: $\mu_c((s,\tau),\sigma) = \{Q' \times \{\tau\} \cup Q'' \times \{\sigma\}: (Q',Q'') \in \mu_\neq(s)\}$
4. Remark: What is a run?

$c = (s, \tau)$ is a configuration

$C$ is a set of configurations

A run of $C$ is:

$C = C_0$

$C_{i+1} \in \mu^c(C_i, \sigma_{i+1}), \ i=0,1,\ldots,n-1$

This run is accepting if $C_n \subseteq F \times (\Sigma \setminus \Delta)$

Automata $A$ $C$ accepts a word $w \in \Sigma^*$ if there is an accepting $C$-run of $A$ on $w$.

$L(A_C) = \bigcap_{c \in C} L(A_c)$

The set of all words $C$-accepted by $A$ is denoted by $L(A_C)$. 
5. Proposition:

Let $C$ and $C'$ be finite sets of configurations such that $C \subseteq C'$. Then $L(A_C) \subseteq L(A_{C'})$. 
Let $\alpha$ be an automorphism of $\Sigma$. We extend $\alpha$ to configurations by $\alpha(s,\sigma) = (s,\alpha(\sigma))$ and then to finite sets of configurations by $\alpha(C) = \{\alpha(c) : c \in C\}$.
7. Definition

Automorphisms of $\Sigma$ which are invariant on $\Delta$ are called $\Delta$-automorphisms.

Reminder: invariant Automorphisms means that for every automorphism $\alpha$ of $\Sigma$ when $\sigma \in \Delta$, $\alpha(\sigma) = \sigma$. 
Let $A$ be a one-window alternating finite-memory automaton, $w = \sigma_1 \sigma_2 \ldots \sigma_n \in \Sigma^*$, $C$ be a finite set of configurations of $A$, $C_0, C_1, \ldots, C_n$ be a $C$-run of $A$ on $w$, and let $\alpha$ be a $\Delta$-automorphism of $\Sigma$. Then $\alpha(C_0), \alpha(C_1), \ldots, \alpha(C_n)$ is an $\alpha(C)$-run of $A$ on $\alpha(w)$. 
Let $C$ be a finite set of configurations of $A$ and let $\alpha$ be a $\Delta$-automorphism of $\Sigma$. Then $\alpha(L(A_C)) = L(A_{\alpha(C)})$. 
Section 3 : Proof of Theorem 1
let $A = \langle S, s_0, F, \Delta, \delta, \mu_\Delta, \mu_\leq, \mu \neq \rangle$ be a one-window alternating finite-memory automaton. To decide whether $L(A) = \emptyset$, we shall compute a positive integer $N$ such that $L(A)$ is nonempty if and only if it contains a word of length less than $N$.

The "trick" we use is to suppose that we already have the $N$. But in the end we will show that it is really computable.
Decision procedure

The decision procedure is as follows:

Assume that we have computed the above number $N$.

Let $\tau_1, \tau_2, \ldots, \tau_{N-1}$ be $\tau_i \notin \Delta \cup \{\delta\}$, and let $\Sigma' = \{\tau_i\}_{i=1}^{N-1} \cup \Delta \cup \{\delta\}$.

Then $L(A)$ is nonempty if and only if it contains a word over $\Sigma'$ of length less than $N$.

The "if" direction is trivial. For the "only if" direction, let $w = \sigma_1, \sigma_2, \ldots, \sigma_n \in L(A)$, $n < N$. Since $n < N$, there is a $\Delta$-automorphism $\alpha$ of $\Sigma$ such that $\alpha(\delta) = \delta$ and $\alpha(w)$ is a word over $\Sigma'$.

Then by Corollary 9 with $C = \{(s_0, \delta)\}$ and $\alpha(\delta) = \delta$, $\alpha(w) \in L(A)$ as well.

Alternatively, the set of all words over $\Sigma'$ of length less than $N$ is finite and for each such word we can check whether it is accepted by $A$. 
therefore, \( L(A) \neq \emptyset \) if and only if \( L(A) \cap \Sigma^* \neq \emptyset \). 

\( \Sigma' \) is finite and it is possible to construct a finite alternating automaton that accepts \( L(A) \cap \Sigma^* \).

by theorem*, this language is regular and our problem is reduced to decidability of the emptiness problem for regular languages.

So, it remains to compute the above upper bound $N$ on the minimum length words in $L(A)$.

For a finite set $C$ of configurations of $A$, we denote by $\Sigma_C$ the following subset of $\Sigma \setminus \Delta$
\[\Sigma_C = \{ \sigma \in \Sigma \setminus \Delta : \text{for some } s \in S, \ (s, \sigma) \in C \}.\]
$\Sigma_C$ consists of all elements of $\Sigma \setminus \Delta$ which occur in $C$ configurations.
$S_C^\sigma = \{ s : (s, \sigma) \in C \}$

let $P$ be a nonempty subset of $S$ and let the subset $\Sigma_C^P$ of $\Sigma_C$ be defined by

$\Sigma_C^P = \{ \sigma : S_C^\sigma = P \}$

then $\Sigma_C = \bigcup_{P \subseteq 2^S \setminus \{\emptyset\}} \Sigma_C^P$

for example if $P = \{s_1, s_2\}$ and $C = \{(s_1, a), (s_2, a), (s_1, b), (s_2, b)\}$
then $\Sigma_C^P = \{a, b\}$
in what follows we would deal with the function \( N^{2^s \setminus \emptyset} \) with the following partial order \( \leq \).

For \( f, h \in N^{2^s \setminus \emptyset} \), \( f \leq h \) if and only if for all \( P \in 2^s \), \( f(P) \leq h(P) \).

We define \( f_c \) as follows:

- For \( P \subseteq 2^s \setminus \{\emptyset\} \), \( f_c(P) = |\sum_c| \)
Proposition 11

Let $C$ be a finite set of configurations and let $\alpha$ be a $\Delta$-automorphism of $\Sigma$. Then $f_C = f_{\alpha(C)}$

**Proof:** We have to show that $f_C = f_{\alpha(C)}$. $f_C = f_{\alpha(C)}$ is immediate because for $P \subseteq 2^S \setminus \{\emptyset\}$, $\alpha(\sum^P_C) = \sum^P_{\alpha(C)}$, implying $|\sum^P_C| = |\alpha(\sum^P_C)| = |\sum^P_{\alpha(C)}|$. □
Proposition 12

Let $C$ and $C'$ be finite sets of configurations. Then $f_C \leq f_{C'}$ if and only if there is a $\Delta$-automorphism $\alpha$ of $\Sigma$ such that for all $P \in 2^S \setminus \{\emptyset\}$, $\alpha(\Sigma^P_C) \subseteq \Sigma^P_{C'}$.

**Proof:** The "if" part is immediate, because for all $P \in 2^S \setminus \{\emptyset\}$, $|\alpha(\Sigma^P_C)| \leq |\Sigma^P_C|$. The "only if" part is easy, because $f_C \leq f_{C'}$, by definition, for all $P \in 2^S \setminus \{\emptyset\}$, $|\Sigma^P_C| < |\Sigma^P_{C'}|$. Therefore there is an embedding $e: \Sigma_C \rightarrow \Sigma_{C'}$, such that for all $P \in 2^S \setminus \{\emptyset\}$, $e(\Sigma^P_C) \subseteq \Sigma^P_{C'}$. $e$ extends to a $\Delta$-automorphism of $\Sigma$, because it is one to one and $\Sigma$ is infinite. \(\square\)
Corollary 13

Let $C$ and $C'$ be finite sets of configurations such that $f_c \leq f_{c'}$. Then there is a $\Delta$-automorphism $\alpha$ of $\Sigma$ such that $\alpha(C) \subseteq C'$.

proof: by proposition 12, $f_c \leq f_{c'}$ if and only if there is a $\Delta$-automorphism $\alpha$ of $\Sigma$ such that for all $P \in 2^S \setminus \{\emptyset\}$, $\alpha(\Sigma_C^P) \subseteq \Sigma_{C'}^P$. That would mean $\Sigma_C \subseteq \Sigma_{C'}$, by definition, which means that there is a $\Delta$-automorphism $\alpha$ of $\Sigma$ such that $\alpha(C) \subseteq C'$. □
Lemma 14

Let $C$ and $C'$ be finite sets of configurations such that $f_c \leq f_{c'}$. If $L(A_{C'})$ contains a word of length $n$, then so does $L(A_C)$.

**proof:** $w \in L(A_{C'})$. Since $f_c \leq f_{c'}$, by corollary 13, there is an automorphism $\alpha$ of $\Sigma$ such that $\alpha(C) \subseteq C'$. Therefore, by proposition 5, $w \in L(A_{\alpha(C)})$, and by corollary 9, $\alpha^{-1}(w) \in L(A_C)$. □
**Definition 15, and lemma 16**

A sequence $f_0,f_1,\ldots,f_N$ of elements of $(2^S)^A$ is called reducible if there are $i$ and $j$, $i<j$, such that $f_i \leq f_j$.

Let a positive integer $N$ be such that for every word $w \in \Sigma^N$ and for every run $C_0,C_1,\ldots,C_N$ of $A$ on $w$, the associated sequence $f_{C_0},f_{C_1},\ldots,f_{C_N}$ is reducible. Then, $L(A_{C_0})$ is nonempty if and only if it contains a word shorter than $N$.

**proof:** the "if" part is immediate. For the "only if" part, assume that $L(A) \neq \emptyset$ and let $w = \sigma_0\sigma_1\cdots\sigma_n$ be a word of the minimum length belonging to $L(A)$ ($n<N$).
To prove our contention we assume to the contrary that $n \geq N$ and let $C_0, C_1, \ldots, C_N$ be an accepting $C_0$-run of $A$ on $w$. Then $C_0, C_1, \ldots, C_N$ is a $C_0$-run on $\sigma_0 \cdots \sigma_N$.

Let $i$ and $j$, $i < j$, be such that $f_{C_i} \leq f_{C_j}$.

Since by definition, $\sigma_{j+1} \cdots \sigma_n \in L(A_{C_j})$, by lemma 14, there exists a word $w'$ of length $n-j$ such that $w' \in L(A_{C_i})$.

Since $C_0, C_1, \ldots, C_i$ is a $C_0$-run of $A$ on $\sigma_j \cdots \sigma_i$, the word $w'' = \sigma_j \cdots \sigma_i w'$ is in $L(A_{C_0})$.

The length of $w''$ is $n-j+i < n$, in contradiction with the minimality assumption on the length $n$ of $w$. 
Lemma 17

we need the following notation in order to compute N from Lemma 16.
For positive integer \( n \), \( T_n \) denotes the set of all sequences of pairs of functions
\( f_{C_0}, f_{C_1}, \ldots, f_{C_n} \), \( n \geq 0 \), where \( C_0, C_1, \ldots, C_n \) is an \( \{ (s_0, \delta) \} \)-run of \( A \) on a word of length \( n \),
and \( T = \bigcup_{n=0}^{\infty} T_n \).

For any \( n \), the set \( T_n \) is computable.

**proof:** Let \( \tau_1, \tau_2, \ldots, \tau_n \) be pairwise distinct symbols different of those described
in \( A \) and let \( \Sigma' = \{ \tau_i \}_{i=1}^{n}, \Delta \cup \{ \delta \} \). Let \( f_{C_0}, f_{C_1}, \ldots, f_{C_n} \) be an element of \( T_n \) that results
from the \( \{ (s_0, \delta) \} \)-run \( C_0, C_1, \ldots, C_n \) of \( A \) on \( w=\sigma_0, \sigma_1, \ldots, \sigma_n \in \Sigma^n \). There exists a
\( \Delta \)-automorphism \( \alpha \) of \( \Sigma \) such that \( \alpha(\delta) = \delta \) and \( \alpha(w) \) is a word over the (finite) \( \Sigma' \).

By proposition 11, \( f_{C_0} = f_{\alpha(C_0)}, i=0, 1, \ldots, n \).

Therefore, by proposition 8 the sequence \( f_{C_0}, f_{C_1}, \ldots, f_{C_n} \) can also be obtained from the
run \( \alpha(C_0), \alpha(C_1), \ldots, \alpha(C_n) \) of \( A \) on \( \alpha(w) \).

Thus we may run \( A \) on words from \( \Sigma^n \) when computing \( T_n \).
\( \Sigma' \) is finite and the set of all sequences associated with runs of \( A \) on a given \( w \) is computable.
König’s Infinity Lemma

we will show that a tree $T$ with infinite nodes (each node has a finite number of children) has an infinite sequence.

Let $t_0$ be the root. By definition, it has a finite number of children. Suppose that all of these had a finite number of children. Then that would mean that $t_0$ had a finite number of descendants, and that would mean T was finite. So $t_0$ has at least one child with infinitely many descendants. Call the child with infinitely many descendants $t_1$.

Now, suppose node $t_k$ has infinitely many descendants. just like $t_0$ it has a child with an infinite sequence. call the child with infinitely many descendants $t_{k+1}$. $T$ has an infinite sequence. □
Parallel program schemata Lemma

Let $P_0, P_1, \ldots, P_n, \ldots$ be an infinite sequence of elements $(\mathbb{N} \cup \{w\})^*$. Then there is an infinite subsequence $P_{i_1}, P_{i_2}, \ldots, P_{i_n}, \ldots$ such that $P_{i_1} \leq P_{i_2} \leq \cdots \leq P_{i_n} \leq \cdots$

**proof**: Extract an infinite subsequence nondecreasing in the first coordinate, extract from this an infinite subsequence nondecreasing in the second coordinate, and so forth. □
Lemma 18

A positive integer $N$ satisfying the prerequisites of Lemma 16 is computable.

**proof:** For each $N$, applying Lemma 17, we can check whether for each $w \in \Sigma^N$, all sequences associated with runs of $A$ on $w$ are reducible. Therefore, if $N$ is satisfying the prerequisites of Lemma 16 exists, it can be found by checking all sequences of length 1, then 2, etc (the process must eventually terminate at the right $N$).

To complete the proof we have to show that there indeed always exists a $N$ satisfying the prerequisites of Lemma 16. For this, we impose on the set of sequences $T$ the following (unranked) tree structure.

sequence $f_{C_0} f_{C_1} \cdots f_{C_n}$ is a successor of sequence $f_{C_0} f_{C_1} \cdots f_{C_n}$ if and only if $n' = n + 1$ and $f_{C'_m} = f_{C_n}$, for all $m = 0, 1, \ldots, n$. 

It follows from the proof of Lemma 17 that the tree $T$ is of a finite branching degree and that for any two nodes of $T$ it is decidable whether one is a successor of the other.

**proof:** assume to the contrary that there is no such $N$, that is, for each $N=1,2,\ldots$, there exists an irreducible node $f_{C_{N,0}}, f_{C_{N,1}}, \ldots, f_{C_{N,N}}$ in $T$.

Since the number of such nodes is infinite and the tree $T$ is of a finite branching degree, by a straightforward modification of König's Infinity Lemma, there exists an infinite path in $T$ such that for each node $f$ on it there is an irreducible sequence $f_{C_{N,0}}, f_{C_{N,1}}, \ldots, f_{C_{N,N}}$ for which $f \leq f_{C_{N,0}}, f_{C_{N,1}}, \ldots, f_{C_{N,N}}$. 
The sequence of the nodes on the path is an infinite sequence of all prefixes of some infinite sequence of pairs of functions $f_0, f_1, \ldots, f_m, \ldots$. In addition, each $f_m$, $m=0, 1, \ldots$, being a prefix of some irreducible sequence $f_{c_{n0}}, f_{c_{n1}}, \ldots, f_{c_{kn}}$, is also irreducible. That is, the infinite sequence $f_0, f_1, \ldots, f_m, \ldots$ contains no two pairs of functions $f_i$ and $f_j$, $i<j$, such that $f_i \leq f_j$.

However, by Remark 10, this contradicts Parallel program schemata Lemma stating that exactly the opposite holds.

Now the decision procedure presented in the beginning of this section can be implemented.