Linear Markov Properties and Undecidability

東邦大学理学部 小林 ゆう治 (Yuji Kobayashi)
京都産業大学理学部 勝良 昌司 (Massahi Katsura)

1 Introduction and Preliminaries

A finitely presented monoid $M$ is given by a finite alphabet (a finite set of generators) and a finite
rewriting system (a finite set of defining relations). Even though $M$ is defined by a finite set of data,
many algebraic properties of $M$ are undecidable. In fact, for any Markov property $P$ of monoids,
there is no algorithm to decide whether a given finitely presented monoid satisfies $P$ (Markov [4]).
In the proof of this undecidability result, a monoid with unsolvable word problem was used. So the
undecidability of Markov properties was proved in a class of monoids that contains monoids with
unsolvable word problem.

Sattler-Klein [8] proved that some of Markov properties are still undecidable in a class of finitely
presented monoids with solvable word problem (see also [4]). She employed her monoids, which she
used to show the divergence phenomena of the completion procedure ([6], [7]). Actually, for any
recursively enumerable language $L$, she constructed finitely presented monoids $S_w$ parameterized by
words $w$ with the following properties. The monoid $S_w$ has word problem solvable in polynomial
time, and it is trivial if $w \in L$, on the other hand it is infinite, non-commutative, non-free etc. if
$w \notin L$. Thus, such properties as finiteness of monoids are undecidable in the class of monoids with
word problem solvable in polynomial time.

In this paper we improve her results in two directions. First we improve the results from polynomial
to linear time. Secondly we give a systematic way to carry over Markov's proof of undecidability to
our restricted class of monoids, so that we can prove that any Markov property related to linear
complexity in some sense is undecidable for finitely presented monoids with word problem solvable in
linear time. In fact, most of ordinary Markov properties are Markov properties in our sense.

Let $\Sigma$ be a (finite) alphabet and let $\Sigma^*$ be the free monoid generated by $\Sigma$. The empty word,
which is an identity element of the monoid, is denoted by 1. Set $\Sigma^+ = \Sigma^* \setminus \{1\}$. For a word $x \in \Sigma^*$,
$|x|$ denotes its length. A rewriting system $R$ is a set of ordered pairs $(u, v)$ with $u, v \in \Sigma^*$. An element
$(u, v)$ of $R$ is called a rule and written as $u \rightarrow v$. For $x, y \in \Sigma^*$, we write $x \rightarrow_R y$ if $x = x_1ux_2$ and
$y = x_1vx_2$ for some $x_1, x_2 \in \Sigma^*$ and $u \rightarrow v \in R$. As usual, $\rightarrow_R$ is the reflexive transitive closure of
$\rightarrow_R$. If $x \rightarrow_R^* y$, $x$ is an ancestor of $y$ and $y$ is a descendant of $x$.

The reflexive symmetric transitive closure $\leftrightarrow_R$ is the Thue congruence generated by $R$. The monoid
$M(\Sigma, R)$ presented by $\Sigma, R$ is the quotient monoid $\Sigma^*/\leftrightarrow_R$. The word problem for $M = M(\Sigma, R)$
is the following decision problem: Given two words $x, y \in \Sigma^*$, decide if $x = y$ in $M$. Two systems $R$
and $R'$ over $\Sigma$ are equivalent if $\leftrightarrow_R = \leftrightarrow_{R'}$, that is, $R$ and $R'$ define the same quotient monoid.
When the system $R$ is fixed and there is no confusion, we simply write $\rightarrow$, $\rightarrow^*$ and $\leftrightarrow^*$ for $\rightarrow_R$, $\rightarrow_R^*$ and $\leftrightarrow_R^*$ respectively.

A rewriting system $R$ is noetherian (terminating) if there is no infinite sequence $x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_n \rightarrow \cdots$. It is confluent if any words $x, y \in \Sigma^*$ with common ancestor have a common descendant. A system $R$ is called complete if it is both noetherian and confluent. A word is irreducible if no rule from $R$ can be applied to it. An irreducible descendant of $x$ is called a normal form of $x$. If $R$ is complete, for any $x \in \Sigma^*$, there is a unique normal form which is denoted by $\hat{x}$. Moreover, for two words $x, y \in \Sigma^*$, $x \leftrightarrow^* y$ if and only if $\hat{x} = \hat{y}$. Hence, the word problem for a finite complete system $R$ is solved by a normal form algorithm, namely, given words $x$ and $y$ we compute the normal forms $\hat{x}$ and $\hat{y}$ of $x$ and $y$ and check whether they are identical.

It is well known that a noetherian system $R$ is complete if and only if all the critical pairs are resolvable. Here, a pair $(z_1, z_2)$ of words is a critical pair, if there are rules $u_1 \rightarrow v_1$, $u_2 \rightarrow v_2 \in R$ such that one of the following holds:

(i) $u_1 = xu_2y$, $z_1 = v_1$, $z_2 = xv_2y$ for some $x, y \in \Sigma^*$ ($u_1 \rightarrow v_1$, $u_2 \rightarrow v_2$ are different), or

(ii) $u_1 = xz$, $u_2 = zy$, $z_1 = v_1y$, $z_2 = xv_2$ for some $x, y, z \in \Sigma^+$.

A critical pair $(z_1, z_2)$ is resolvable if $z_1$ and $z_2$ have a common descendant.

We fix a compatible well-order $<$ on $\Sigma^*$. A rewriting system $R$ is $<\rightarrow$-reducing if $u > v$ for all $u \rightarrow v \in R$. If $R$ is $<\rightarrow$-reducing, it is noetherian. If a finite system $R$ is not complete, we can apply the completion procedure (the Knuth-Bendix completion procedure [3], see also [1]) to get a complete system equivalent to $R$. First, orient $R$ so that $R$ becomes $<\rightarrow$-reducing. If there is a critical pair $(x, y)$, compute normal forms $\hat{x}$ and $\hat{y}$ of $x$ and $y$ respectively (we can compute them because $R$ is finite and noetherian). If $\hat{x} = \hat{y}$, then the critical pair is resolved. If $\hat{x} > \hat{y}$ (resp. $\hat{y} > \hat{x}$), add the rule $\hat{x} \rightarrow \hat{y}$ (resp. $\hat{y} \rightarrow \hat{x}$) to the system. Repeat this until all the critical pairs are resolvable. This procedure may not terminate even if the original system $R$ is finite. It terminates if and only if there is a $<\rightarrow$-reducing finite complete system equivalent to $R$, and if it terminates, it gives such a system. Even if the procedure does not terminate, it gives, in the limit, a $<\rightarrow$-reducing infinite complete system equivalent to $R$.

2 Linear Markov properties

Let $C_1$ be the class of finitely presented monoids with word problem solvable in linear time. By a property $P$ of monoids, we mean an invariant property of monoids, that is, if a monoid $M$ satisfies $P$, every monoid isomorphic to $M$ satisfies $P$.

A property $P$ of monoids is called a Markov property relative to linear complexity (a linear Markov property for short), if

(i) there is a monoid $M_1$ in $C_1$ with property $P$, and

(ii) there is a monoid $M_2$ in $C_1$ that is not embeddable in any monoid in $C_1$ with property $P$, in other words, any monoid in $C_1$ containing a submonoid isomorphic to $M_2$ does not satisfy $P$.

Example. (1) Left-cancellativity is linear Markov. So the following stronger properties are also linear
- cancellativity  
- being a group  
- freeness  
- triviality  etc.

(2) Satisfying some fixed nontrivial (quasi-)identities is linear Markov, for example:
- commutativity  
- idempotency  
- nilpotency  
- finiteness  etc.

(3) Negation of having an element (subset) with some local properties is linear Markov. For example, the negations of the following:
- having a nontrivial idempotent  
- containing a nontrivial subgroup  etc.

Here we state our main theorem. A sketch of the proof is given below but the details will appear in [2].

**Theorem 2.1** Any linear Markov property is undecidable for finitely presented monoids with word problem solvable in linear time.

3  **Rewriting systems simulating Turing machines**

For the proof of the main theorem, we need to consider a Turing machine accepting a non-recursive language and a rewriting system simulating the machine. Let $L$ be a recursively enumerable language over a finite alphabet $\Gamma$. Let $\text{TM} = (\Gamma, Q, q_0, q_h, \delta)$ be a single-tape deterministic Turing machine accepting $L$ given as follows. $\Gamma$ is the set of tape symbols, $Q = \{q_0, q_1, \ldots, q_n\}$ is the set of states, $q_0$ is the initial state, and $q_h$ is the halting state. We suppose $k \geq 1$ and set $Q' = Q \setminus \{q_h\}$.

Let $\Gamma_b = \Gamma \cup \{b\}$, where $b$ is the blank symbol outside $\Gamma$. The transition function is a mapping $\delta : Q' \times \Gamma_b \rightarrow Q \times \Gamma_b \times \{R, L\}$, where $L$ and $R$ are the symbols for the right and left moves of the head respectively.

A word $xqy$ with $x, y \in \Gamma_b^*$ and $q \in Q$ is a configuration of $\text{TM}$. Let $\vdash$ denote the one-step computation relation on the set of configurations of $\text{TM}$, that is,

(a) $xqay \vdash xa'q'y$ if $\delta(q, a) = (q', a', R)$ for $a, a' \in \Gamma_b$ and $q, q' \in Q'$,

(b) $xqay \vdash xqca'y$ if $\delta(q, a) = (q', c, L)$ for $a, c \in \Gamma_b$ and $q, q' \in Q'$,

(c) $xq \vdash x'q'y$ if $xq \vdash x'q'y'$ by (a) or (b) for $x, x', y' \in \Gamma_b^*$ and $q, q' \in Q'$.

If a configuration $x'q'y'$ is obtained from a configuration $xqy$ through $n$ computation steps, we write $xqy \vdash^n x'q'y'$. Given a word $w$ in $L$ as input $\text{TM}$ will stop in state $q_h$ after a finite number of computation steps, and on the other hand given a word $w$ not in $L$, $\text{TM}$ will not stop and run forever;

$$L = \{w \in \Gamma^* | q_0w \vdash^n xqhy \text{ for some } x, y \in \Gamma_b^*\},$$

where $\vdash^n$ denotes the reflexive transitive closure of $\vdash$, that is, $\vdash^n = \bigcup_{n=0}^{\infty} \vdash^n$. Moreover, without loss of generality we may assume that the head of $\text{TM}$ never moves to the left of the initial position.

Now, we give a rewriting system $T$ simulating the machine $\text{TM}$ in some way. Let $\Sigma = \Gamma_b \cup Q \cup \{H, E, A, \bar{A}, B, \bar{B}, O\}$, where $H, E, A, \bar{A}, B, \bar{B}, O$ are new letters. Below, $a, a'$ and $c$ are arbitrary letters in $\Gamma_b$, $q$ and $q'$ are arbitrary states in $Q$, and for a set $X$ of words, $X \rightarrow O$ denotes the collection of rules $x \rightarrow O$ for $x \in X$. The system $T$ consists of the following rules:

1a : $a\bar{A} \rightarrow \bar{A}a$,

1b : $H\bar{A} \rightarrow HA$, 

3 : $\bar{A} \rightarrow A$, 

4 : $\bar{B} \rightarrow B$
1c : \( Aa \rightarrow aA \),
1a' : \( \bar{B}a \rightarrow a\bar{B} \),
1b' : \( \bar{B}E \rightarrow B\bar{B}E \),
1c' : \( aB \rightarrow Ba \),
2a : \( AqBa \rightarrow \bar{A}aq\bar{B} \) for \( (q, a, q', a', R) \in \delta \),
2b : \( cAqBa \rightarrow \bar{A}q'ca'\bar{B} \) for \( (q, a, q', a', L) \in \delta \),
3a : \( aAq_kB \rightarrow Aq_kB \), \( Aq_kB \rightarrow Aq_kB \),
3b : \( HAq_kB \rightarrow HE \),
3c : \( HEE \rightarrow HE \),
4a : \( O\sigma \rightarrow O, \sigma O \rightarrow O \) for \( \sigma \in \Xi \),
4b : \( \{A, \bar{A}, B, \bar{B}\}^2 \setminus \{\bar{A}, B\}\{A, \bar{B}\} \rightarrow O \),
4c : \( \{qB, q', \bar{B}a, q\bar{A}, qAq', qBq'\} \rightarrow O \),
4d : \( \{AE, HB\} \rightarrow O \),
4e : \( \sigma H \rightarrow O \) for \( \sigma \in \Xi \),
4e' : \( E\sigma \rightarrow O \) for \( \sigma \in \Xi \setminus \{E\} \).

**Lemma 3.1** The system \( T \) is complete.

The following lemma shows how the system \( T \) simulates \( \text{TM} \).

**Lemma 3.2** Let \( x, y, x', y' \in \Gamma^* \), \( q, q' \in Q \) and \( n \geq 0 \). If \( xqyb^n \vdash^{n} x'q'y' \), then

\[ HxAq\bar{B}E^t \rightarrow^{*}_T Hx'Aq'y'\bar{B}E^{t-n} \]

for \( t > n \). If, moreover, \( q' = q_k \) and \( t > n + 1 \), we have

\[ HxAq\bar{B}E^t \rightarrow^{*}_T HE \]

For each word \( w \in \Gamma^* \) we consider the rule

\( 0_w : HAq_0w\bar{B}E \rightarrow O \).

**Definition 3.3** Let \( w \in \Gamma^* \). Define the system \( T_w \) by adding rule \( 0_w \) to \( T \);

\[ T_w = T \cup \{0_w\}, \]

and let \( N_w = M(\Xi, T_w) \) be the monoid presented by \( (\Xi, T_w) \).

The system \( T_w \) is noetherian but not complete any more. In fact, applying rule \( 0_w \) to the word \( HAq_0w\bar{B}E^{t+1} \) for \( t > 0 \), we obtain \( OE^t \), which is reduced to \( O \). On the other hand, if \( q_0wb^t \vdash^* xqy \)

for some \( x, y \in \Gamma^* \), then by Lemma 3.2 we have \( HAq_0w\bar{B}E^{t+1} \rightarrow^{*}_T HxAq\bar{B}E \). Thus,

\[ HxAq\bar{B}E \leftrightarrow^{*}_T O. \quad (3.1) \]

Here, if \( q = q_k \), then \( HxAq\bar{B}E \rightarrow^{*}_T HE \) by Lemma 3.2. Hence, we see

\[ HE \leftrightarrow^{*}_T O. \quad (3.2) \]
Now, if \( w \) is not in \( L \), then for any \( t > 0 \), there uniquely exist \( x_t, y_t \in \Sigma^* \) and \( q(t) \in Q' \) such that \( q_0 w^t \vdash^t x_t q(t) y_t \), because \( TM \) is deterministic. The words on both sides of (3.1) are \( T_w \)-irreducible. 

So, to make the system complete, we add the rule

\[
0^t_w : H x_t A q(t) y_t \overrightarrow{E} O
\]

for every \( t > 0 \).

On the other hand, if \( w \) is in \( L \), then \( q_0 w^n \vdash^n x q_k y \) for some \( n > 0 \) and some \( x, y \in \Gamma^n \), and (3.2) holds. To make the system complete we add the rule

\[
4f : HE \rightarrow O.
\]

In this case we remove rule 3c, because it is a consequence of 4f.

In this way we have the complete system \( \hat{T}_w \) equivalent to \( T_w \) in the following lemma.

**Lemma 3.4** (1) If \( w \) is not in \( L \),

\[
\hat{T}_w = T_w \cup \{0^t_w | t = 1, 2, \ldots\}
\]

is an infinite complete system equivalent to \( T_w \).

(2) If \( w \) is in \( L \) and \( n \) is the positive integer such that \( q_0 w^n \vdash^n x_{q_k} y \) for \( x, y \in \Gamma^n \), then

\[
\hat{T}_w = (T_w \setminus \{3c\}) \cup \{0^t_w | t = 1, 2, \ldots, n\} \cup \{4f\}
\]

is a finite complete system equivalent to \( T \).

**Corollary 3.5** A word \( w \in \Gamma^* \) is in \( L \), if and only if \( HE = O \) holds in the monoid \( N_w \).

An important feature of our construction is stated in the following lemma.

**Lemma 3.6** The monoid \( N_w \) has word problem solvable in linear time.

Summarizing we have

**Theorem 3.7** The monoid \( N_w \) has word problem solvable in linear time, and we have the following.

(1) If \( w \) is in \( L \), then \( HE = O \) in \( N_w \).

(2) If \( w \) is not in \( L \), then \( HE \neq O \) in \( N_w \).

4 Embedding lemma and a proof of the main theorem

Let \( (\Sigma, R) \) be an arbitrary monoid presentation and let \( M = M(\Sigma, R) \). Let \( \alpha, \beta, \gamma \) be new letters outside \( \Sigma \) and let \( x, y \in \Sigma^* \). Consider the system \( S \) over \( \Sigma' = \Sigma \cup \{\alpha, \beta, \gamma\} \) given by

\[
S = \{\alpha x \beta \rightarrow 1, \alpha y \beta \rightarrow \gamma\} \cup \{\sigma \gamma \rightarrow \gamma, \gamma \sigma \rightarrow \gamma | \sigma \in \Sigma'\}.
\]

We define a monoid \( \Phi_{x,y}(M) \), which is determined by \( x, y \) and \( (\Sigma, R) \), by \( \Phi_{x,y}(M) = M(\Sigma', R \cup S) \). Let \( \phi : M \rightarrow \Phi_{x,y}(M) \) be the morphism of monoids induced by the inclusion \( \Sigma \hookrightarrow \Sigma' \).

**Lemma 4.1** If \( x = y \) in \( M \), \( \Phi_{x,y}(M) \) is the trivial monoid. If \( x \neq y \) in \( M \), \( \phi \) is injective. Moreover, if \( M \) has word problem solvable in linear time, so does \( \Phi_{x,y}(M) \).
Definition 4.2 For a monoid $M = M(\Sigma, R)$ and a word $w \in \Gamma^*$ we define

$$\Psi_w(M) = \Phi_HE,O(M \ast N_w)$$

with the free product $M \ast N_w$ of $M$ and $N_w$, that is, the monoid $\Psi_w(M)$ is defined over the alphabet $\Sigma' \cup \Xi$ with $\Sigma' \cap \Xi = \emptyset$ by the relation $R \cup T_w \cup S$, where

$$S = \{aHE\beta \rightarrow 1, aO\beta \rightarrow \gamma, \sigma\gamma \rightarrow \gamma, \gamma\sigma \rightarrow \gamma | \sigma \in \Sigma' \cup \Xi\}.$$ 

Theorem 4.3 (1) If $w$ is in $L$, $\Psi_w(M)$ is the trivial monoid.

(2) If $w$ is not in $L$, $\Psi_w(M)$ contains $M$ as submonoid.

(3) If $M$ has word problem solvable in linear time, so does $\Psi_w(M)$. 

Proof of the main theorem

Once we have Theorem 4.3, the proof of Theorem 2.1 is now standard. Let $P$ be a linear Markov property. Let $M_1$ be a monoid in $C_1$ with the property $P$ and $M$ be a monoid in $C_1$ that is not embeddable in a monoid in $C_1$ with $P$. We choose the recursively enumerable language $L$ to be nonrecursive. For $w \in \Gamma^*$ let $M_w = M_1 \times \Psi_w(M)$ be the direct product of $M_1$ and $\Psi_w(M)$. If $M_1$ and $M$ are presented by $(\Sigma_1, R_1)$ and $(\Sigma, R)$ with $\Sigma_1 \cap \Sigma = \emptyset$, respectively, $M_w$ is presented by $(\Sigma_1 \cup \Sigma' \cup \Xi, R_1 \cup R \cup T_w \cup S \cup S')$, where

$$S' = \{\tau\sigma \rightarrow \sigma\tau | \sigma \in \Sigma_1, \tau \in \Sigma' \cup \Xi\}.$$ 

It is easy to see that $M_w$ has word problem solvable in linear time because both $M_1$ and $\Psi_w(M)$ have linear word problem by Theorem 4.3, that is, $M_w \in C_1$. Moreover, if $w \in L$, $M_w$ is isomorphic to $M_1$ because $\Psi_w(M)$ is trivial, and otherwise, $M_w$ contains $M$ as a submonoid because so does $\Psi_w(M)$. So, $M_w$ satisfies $P$ if and only if $w$ is in $L$. This completes the proof of the main theorem. □

References


