Algorithms to decide fragment enriched by regular predicates\textsuperscript{1}

Luc Dartois and Charles Paperman

LIAFA
Université Paris Diderot

Highlights of logic, Game and Automata
September, 2014

\textsuperscript{1}The authors are supported by the ANR project FREC, the second author is supported by Fondation CFM.
Context: Monadic Second Order logic ($\text{MSO}[<]$) on finite words over a finite alphabet.

Fragment ($\mathcal{F}$): set of closed formulae satisfying:
- Atomic replacement rule
- Stability by $\land$ and $\lor$ connectives
- Containing 0-ary predicates

Aim: to obtain decidability of $\mathcal{F}[\mathcal{P}]$ from decidability of $\mathcal{F}$. 
Regular predicates

We will consider the following regular predicates:

- **descriptive local predicates** $L_{OC_k}$
  \[
  \begin{cases}
  a(x - t) & \text{for } t \leq k \\
  a(t) & \\
  a(\text{max} - t) 
  \end{cases}
  \]

- **modular predicates** $\text{Mod}_d$
  \[
  \begin{cases}
  x \equiv r \mod d \\
  \text{max} \equiv r \mod d 
  \end{cases}
  \text{ for } r < d
  \]
Regular predicates

We will consider the following regular predicates:

- **descriptive local predicates** \( \mathcal{L}_{\text{oc}}_k \) \[
\begin{cases} 
  a(x - t) & \text{for } t \leq k \\
  a(t) & \\
  a(\max - t) & 
\end{cases}
\]

- **modular predicates** \( \mathcal{M}_{\text{od}}_d \) \[
\begin{cases} 
  x \equiv r \mod d & \text{for } r < d \\
  \max \equiv r \mod d & 
\end{cases}
\]

**Example:** Can we obtain an algorithm to decide if language belongs to \( \text{FO}[<,\mathcal{M}_{\text{od}}_3] \) from the algorithm for \( \text{FO}[<] \)?
The $\mathcal{L}$-separation problem

$\mathcal{L}$ is a class of languages.

**input:**

Two languages $L_1$ and $L_2$

**problem:**

Is there $S \in \mathcal{L}$ such that \[
\begin{align*}
L_1 &\subseteq S \\
L_2 \cap S &= \emptyset
\end{align*}
\]?
A first result

Theorem
Let $\mathcal{F}$ be a fragment with a decidable separation problem. Then the fragments $\mathcal{F}[\text{Loc}_k], \mathcal{F}[\text{Mod}_d], \mathcal{F}[\text{Loc}_k, \text{Mod}_d]$ are decidable for any $k, d \in \mathbb{N}$.

Corollary
Let $\mathcal{F}$ be a decidable fragment equivalent to a variety and containing the language $(ab)^*$. Then the fragments $\mathcal{F}[\text{Loc}_k], \mathcal{F}[\text{Mod}_d], \mathcal{F}[\text{Loc}_k, \text{Mod}_d]$ are decidable for any $k, d \in \mathbb{N}$. 
The algorithm for $\mathcal{F}[\mathcal{L}_{0c_2}]$

notations:
- $B = A \times A^{\leq 2}$
- $\pi : B^* \to A^*$ the projection
- $K = \left\{ (a_1, 1)(a_2, a_1)(a_3, a_1a_2)(a_4, a_2a_3) \cdots (a_n, a_{n-2}a_{n-1}) \right\}$

input:
A regular language $L$ on $A^*$

algorithm:
For each $u \in A^{\leq 2}$, construct

$$L_u^1 = \pi^{-1}(L \cap A^*u) \cap K \quad \text{and} \quad L_u^2 = \pi^{-1}(L^c \cap A^*u) \cap K$$

Accept iff for each $u \in A^{\leq 2}$, $L_u^1$ is $\mathcal{F}$-separable from $L_u^2$
The algorithm for $\mathcal{F}[\mathcal{L}oc_2]$

notations:

- $B = A \times A^{\leq 2}$
- $\pi : B^* \to A^*$ the projection
- $K = \left\{ (a_1, 1)(a_2, a_1)(a_3, a_1a_2)(a_4, a_2a_3) \cdots (a_n, a_{n-2}a_{n-1}) \right\}$

input:

A regular language $L$ on $A^*$

algorithm if $\mathcal{F}$ is a variety that contains $(ab)^*$:

For each $u \in A^{\leq 2}$, construct

$$L_1^u = \pi^{-1}(L \cap A^* u) \cap K$$

Accept iff for each $u \in A^{\leq 2}$, $L_1^u \in \mathcal{F}$
The algorithm for $\mathcal{F}[\text{Mod}_3]$

notations:

- $B = A \times \mathbb{Z}/3\mathbb{Z}$
- $\pi : B^* \to A^*$ the projection
- $K = \{(a_1, 0)(a_2, 1)(a_3, 2)(a_4, 0)\cdots(a_n, n \text{ mod } 3)\}$

input:

A regular language $L$ on $A^*$

algorithm:

For each $r \in \mathbb{Z}/3\mathbb{Z}$, construct

$L_1^r = \pi^{-1}(L \cap (A^3)^*A^r) \cap K$ and $L_2^r = \pi^{-1}(L^c \cap (A^3)^*A^r) \cap K$

Accept iff for each $r \in \mathbb{Z}/3\mathbb{Z}$, $L_1^r$ is $\mathcal{F}$-separable from $L_2^r$
The algorithm for $\mathcal{F}[\text{Mod}_3]$

notations:
- $B = A \times \mathbb{Z}/3\mathbb{Z}$
- $\pi : B^* \rightarrow A^*$ the projection
- $K = \left\{(a_1, 0)(a_2, 1)(a_3, 2)(a_4, 0)\cdots(a_n, n \mod 3)\right\}$

input:
A regular language $L$ on $A^*$

algorithm if $\mathcal{F}$ is a variety that contains $(ab)^*$:
For each $r \in \mathbb{Z}/3\mathbb{Z}$, construct

$$L_1' = \pi^{-1}(L \cap (A^3)^* A^r) \cap K$$

Accept iff for each $r \in \mathbb{Z}/3\mathbb{Z}$, $L_1' \in \mathcal{F}$
The delay question

notations:
\[ \mathcal{L}_{oc} = \bigcup_k \mathcal{L}_{oc_k} \]
\[ \mathcal{M}_{od} = \bigcup_d \mathcal{M}_{od_d} \]

input:
A language \( L \)

problem(s):
Can we decide if \( L \in \mathcal{F}[\mathcal{L}_{oc}], \mathcal{F}[\mathcal{M}_{od}] \) from an algorithm that decide the separation problem for \( \mathcal{F} \)?
The delay question

notations:

\[ \text{Loc} = \bigcup_k \text{Loc}_k \]
\[ \text{Mod} = \bigcup_d \text{Mod}_d \]

input:

A language \( L \)

problem(s):

Can we decide if \( L \in \mathcal{F}[\text{Loc}], \mathcal{F}[\text{Mod}] \) from an algorithm that decide the separation problem for \( \mathcal{F} \)?

the delay:

Compute \( k \) such that \( L \in \mathcal{F}[\text{Loc}] \rightarrow L \in \mathcal{F}[\text{Loc}_k] \)

Compute \( d \) such that \( L \in \mathcal{F}[\text{Mod}] \rightarrow L \in \mathcal{F}[\text{Mod}_d] \)
Conclusion

**known result:**
If $\mathcal{F}$ is equivalent to a variety, the delay for $\mathcal{Loc}$ is at most the size of the syntactical monoid (Straubing 1985).

**our result:**
If $\mathcal{F}$ is equivalent to a variety of finite rank, the delay for $\mathcal{Mod}$ is computable.

**open question:**
Can one compute the delay for $\mathcal{Mod}$ for the fragment $B\Sigma_m$?