Revisiting Membership Problems in Subclasses of Rational Relations

Pascal Bergsträßer 1  Moses Ganardi 2

1Department of Computer Science, University of Kaiserslautern-Landau
2Max Planck Institute for Software Systems (MPI-SWS)

LICS 2023, Boston, USA
\[ R = \{(x, y) \mid x \text{ is an infix of } y\} \]
Rational Relations

\[ R = \{(x, y) \mid x \text{ is an infix of } y\} \]
Rational Relations

\[ R = \{(x, y) \mid x \text{ is an infix of } y\} \]

\[ q_0 \xrightarrow{(\varepsilon, \ast)} q_1 \xrightarrow{(a, a), (b, b)} q_2 \]

a b b a

b a a b b a a b a
Rational Relations

$$R = \{(x, y) \mid x \text{ is an infix of } y\}$$

\[ (\varepsilon) \quad (a, b) \quad (\varepsilon) \]

\[ q_0 \quad q_1 \quad q_2 \]

\[ a \ b \ b \ a \]

\[ b \ a \ a \ b \ b \ a \ b \ a \]
Rational Relations

\[ R = \{(x, y) \mid x \text{ is an infix of } y\} \]

\[
\begin{array}{l}
(q_0, (\varepsilon)) 
\rightarrow q_0 
\rightarrow (a, a), (b, b) 
\rightarrow q_1 
\rightarrow (\varepsilon, \varepsilon) 
\rightarrow q_2
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{cccc}
a & b & b & a
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{cccc}
b & a & a & b & b & a & b & a
\end{array}
\end{array}
\]
Rational Relations

\[ R = \{(x, y) \mid x \text{ is an infix of } y\} \]
Rational Relations

\[ R = \{(x, y) \mid x \text{ is an infix of } y\} \]
Rational Relations

\[ R = \{(x, y) \mid x \text{ is an infix of } y\} \]
Rational Relations

\[ R = \{(x, y) \mid x \text{ is an infix of } y\}\]
Deterministic Rational Relations

\[ R = \{ (x, y) \mid x \text{ is a subword of } y \} \]
\[ R = \{(x, y) \mid x \text{ is a subword of } y\} \]
$R = \{(x, y) \mid x \text{ is a subword of } y\}$

\[
\begin{array}{c}
q_0 \rightarrow (\varepsilon) \\
q_1 \rightarrow (b) \\
q_2 \rightarrow (\varepsilon a) \\
q_3 \rightarrow (\varepsilon a), (\varepsilon b) \\
q_4 \rightarrow (\varepsilon) \\
\end{array}
\]
$R = \{(x, y) \mid x \text{ is a subword of } y\}$
Deterministic Rational Relations

\[ R = \{(x, y) \mid x \text{ is a subword of } y\} \]
$R = \{(x, y) \mid x \text{ is a subword of } y\}$
Deterministic Rational Relations

\[ R = \{ (x, y) \mid x \text{ is a subword of } y \} \]
Deterministic Rational Relations

\[ R = \{(x, y) \mid x \text{ is a subword of } y\} \]
$R = \{(x, y) \mid x \text{ is a subword of } y\}$
Synchronous Relations

\[ R = \{ (x, y) \mid x \text{ is a prefix of } y \} \]

\[ (a, b), \left( \frac{1}{a} \right), \left( \frac{1}{b} \right) \]

\[ q_0 \xrightarrow{(a)} (a, b) \xrightarrow{(\frac{1}{a})} q_0 \]

\[ q_0 \xrightarrow{(\frac{1}{b})} q_1 \]

\[ \begin{array}{cccc}
a & b & a & \bot & \bot \\
\end{array} \]

\[ \begin{array}{cccc}
a & b & a & b & b \\
\end{array} \]
Synchronous Relations

\[ R = \{(x, y) \mid x \text{ is a prefix of } y\} \]
Synchronous Relations

\[ R = \{(x, y) \mid x \text{ is a prefix of } y\} \]

\[ (a), (b) \]

\[ (\perp a), (\perp b) \]

\[ q_0 \rightarrow q_1 \]

\[ a \quad b \quad a \quad \perp \quad \perp \]

\[ a \quad b \quad a \quad b \quad b \]
Synchronous Relations

\[ R = \{(x, y) \mid x \text{ is a prefix of } y\} \]

\[
\begin{array}{c}
(a, a), (b, b) \\
\downarrow \\
q_0 \\
\begin{array}{cccc}
a & b & a & \bot & \bot \\
\end{array}
\end{array}
\begin{array}{ll}
(\bot, a), (\bot, b) \\
\downarrow \\
q_1 \\
\begin{array}{cccc}
a & b & a & b & b \\
\end{array}
\end{array}
\]
Synchronous Relations

\( R = \{(x, y) \mid x \text{ is a prefix of } y\} \)

\[
\begin{align*}
(a, a), (b, b) & \quad (\frac{1}{a}, \frac{1}{b}) \\
\rightarrow q_0 & \quad (\frac{1}{a}, \frac{1}{b}) \rightarrow q_1
\end{align*}
\]
Synchronous Relations

\[ R = \{(x, y) \mid x \text{ is a prefix of } y\} \]

\[
\begin{align*}
(a, a), (b, b) & \quad (\perp, a), (\perp, b) \\
q_0 & \quad (\perp, a), (\perp, b) \\
q_1 & \quad (\perp, a), (\perp, b)
\end{align*}
\]
Recognizable Relations

Definition

$R$ is recognizable if it can be written as

$$R = \bigcup_{i=1}^{n} L_{i,1} \times \cdots \times L_{i,k}$$

where $L_{i,j}$ are regular languages.

Example

$R = \{(x, y) \mid |x| + |y| \geq 2\}$ can be written as

$$\Sigma^{\geq 2} \times \Sigma^* \cup \Sigma^{\geq 1} \times \Sigma^{\geq 1} \cup \Sigma^* \times \Sigma^{\geq 2}$$
Membership Problems

\[ \text{Rec} \subseteq \text{Sync} \subseteq \text{DRat} \subseteq \text{Rat} \]

\[ \omega-\text{Rec} \subseteq \omega-\text{Sync} \subseteq \omega-\text{DRat} \subseteq \omega-\text{Rat} \]

Pascal Bergsträßer (RPTU)
Recognizability via Finite-Index Equivalences

For $k$-ary relation $R$ define equivalence relation $x \approx^R_j y$ on words such that for all $(z_1, \ldots, z_{k-1})$:

\[(z_1, \ldots, z_{j-1}, x, z_j, \ldots, z_{k-1}) \in R \iff (z_1, \ldots, z_{j-1}, y, z_j, \ldots, z_{k-1}) \in R\]

**Proposition**

Let $R \in \text{Rat} \cup \omega\text{-Sync}$. The following are equivalent:

- $R$ is ($\omega$-)recognizable.
- $\approx^R_j$ has finite index for all $j \in [1, k - 1]$. 
Deciding $\omega$-Recognizability in $\omega$-Sync

- $\text{Rec} \subseteq \omega\text{-Rec} \subseteq \omega\text{-Sync} \subseteq \omega\text{-DRat} \subseteq \omega\text{-Rat}$

- NL-c. for DFAs
- PSPACE-c. for NFAs
  - [Barceló et al. ’19]

- $\text{Rec} \subseteq \text{Sync} \subseteq \text{DRat} \subseteq \text{Rat}$

- 2-EXP for DPAs
- 3-EXP for NBAs
  - [Löding, Spinrath ‘19]
Deciding $\omega$-Recognizability in $\omega$-Sync

\[
\text{Rat} \uparrow \text{DRat} \uparrow \text{Sync} \uparrow \text{Rec} \uparrow \omega\text{-Rat} \uparrow \omega\text{-DRat} \uparrow \omega\text{-Sync} \uparrow \omega\text{-Rec}
\]

NL-c. for DFAs
PSPACE-c. for NFAs
[Barcelo et al. '19]

NL-c. for DPAs
PSPACE-c. for NBAs
Proof Sketch

- Given a (non)det. automaton for $R$, we can compute a nondet. automaton for $\not\approx_R^j$ in logspace (polynomial space).
- $R$ is $\omega$-recognizable iff $\approx_R^j$ has finite index for all $j < k$.
- $\approx_R^j$ has infinite index iff $\not\approx_R^j$ has an infinite clique.
- How to check for an infinite clique in an arbitrary $\omega$-synchronous relation is a longstanding open problem, but:

**Theorem**

*It is NL-complete to decide, given a nondet. Büchi automaton for an $\omega$-synchronous co-equivalence relation $\bar{E}$, whether $\bar{E}$ has an infinite clique.*
Proof Sketch of Theorem

$\overline{E}$ contains an infinite clique iff the automaton for $\overline{E}$ contains one of the following patterns:

\[
\begin{align*}
& & & & & & \\
& & \downarrow 3CP & \rightarrow \rightarrow 3CP & \rightarrow \rightarrow 3CP & \rightarrow \rightarrow 3CP & \\
& & & & & & \\
\end{align*}
\]

The existence of these patterns can be checked in NL.

- $\overline{E}$ contains an infinite clique iff the automaton for $\overline{E}$ contains one of the following patterns:

- The existence of these patterns can be checked in NL.
Deciding Recognizability in \( \text{DRat} \)

\[
\text{Rat} \\
\Downarrow \\
\text{DRat} \\
\Downarrow \\
\text{Sync} \\
\Downarrow \\
\text{Rec}
\]

2-EXP for \( k = 2 \) [Valiant ‘75]
decidable for \( k > 2 \) [Carton et al. ‘06]
Deciding Recognizability in \( \text{DRat} \)

\[
\text{Rat} \\
\cup^* \\
\text{DRat} \\
\cup^* \\
\text{Sync} \\
\cup^* \\
\text{Rec}
\]

- \( P \) for \( k = 2 \)
- \( \text{coREXP} \) for \( k > 2 \)
Proof Sketch for $k = 2$

To check if $\approx_1^R$ as infinite index, it suffices to check if the automaton for $R$ contains the pattern:

Detection of this pattern can be reduced to the inequivalence problem for binary det. rational relations which is in P.
Deciding Synchronicity in DRat

\[ \text{Rec} \subset \text{Sync} \subset \text{DRat} \subset \text{Rat} \]

Diagram:

- Rat
- DRat
- Sync
- Rec

Relationships:
- \( \subset \)
- open

Pascal Bergsträßer (RPTU)
Membership Problems in Subclasses of Rational Relations
LICS 2023
Deciding Synchronicity in $\text{DRat}$

\[ \text{Rec} \subseteq \text{Sync} \subseteq \text{DRat} \subseteq \text{Rat} \]

- $\text{P}$ for $k = 2$
- $(2k - 4)$-EXP for $k > 2$

Pascal Bergsträßer (RPTU)  
Membership Problems in Subclasses of Rational Relations  
LICS 2023 13 / 16
Proof Sketch for $k = 2$
Proof Sketch for $k = 2$

- **Bounded delay** in synchronous part guarantees synchronicity since symbols that are read ahead can be stored in a queue.
- In the asynchronous part there would be no bound on the queue size.
- We reduce to recognizability of binary relations in $\text{D RAT}$:

**Lemma**

$R$ is synchronous if and only if for each reachable state $q$ in the asynchronous part, the relation accepted from $q$ is recognizable.
Summary

\[ \text{Rec} \subseteq \text{Sync} \subseteq \text{DRat} \subseteq \text{Rat} \]

- NL-c. for DFAs
- PSPACE-c. for NFAs

[Barceló et al. ‘19]

\[ \text{P for } k = 2 \]
\[ (2^k - 4) \text{-EXP for } k > 2 \]

- \( \omega \)-Rec
- \( \omega \)-Sync
- \( \omega \)-DRat
- \( \omega \)-Rat

- P for \( k = 2 \)
- \( \text{coREXP for } k > 2 \)

- NL-c. for DPAs
- PSPACE-c. for NBAs

- open
- open