Regular Separability and Intersection Emptiness are Independent Problems

Ramanathan S. Thinniyam, Georg Zetzsche

Max Planck Institute for Software Systems

Highlights 2019
September 18, 2019
Intersection Emptiness and Regular Separability

- Intersection emptiness for language classes $C, D$ (denoted $\text{IE}(C_1, C_2)$):
  Input: $L_1 \in C, L_2 \in D$.
  Question: $L_1 \cap L_2 = \emptyset$?

- Regular Separability for language classes $C, D$ (denoted $\text{RS}(C_1, C_2)$):
  Input: $L_1 \in C, L_2 \in D$.
  Question: Is there regular $S$ such that $L_1 \subseteq S$, $L_2 \cap S = \emptyset$?

Regular separator $S$ provides an easy to verify certificate of disjointness of $L_1$ and $L_2$. 
Intersection Emptiness and Regular Separability

- Intersection emptiness for language classes \( C, D \) (denoted \( \text{IE}(C_1, C_2) \)):
  Input: \( L_1 \in C, L_2 \in D \).
  Question: \( L_1 \cap L_2 = \emptyset \)?

- Verifying safety properties in concurrent programs with shared memory reduces to intersection of PDA languages.

\[
\begin{array}{c}
L_1 \in C \\
\circ \\
? \\
\circ \\
L_2 \in D \\
\Sigma^*
\end{array}
\]
Intersection Emptiness and Regular Separability

Intersection emptiness for language classes $C, D$ (denoted $\text{IE}(C_1, C_2)$):
Input: $L_1 \in C$, $L_2 \in D$.
Question: $L_1 \cap L_2 = \emptyset$?

Verifying safety properties in concurrent programs with shared memory reduces to intersection of PDA languages.

Regular Separability for language classes $C, D$ (denoted $\text{RS}(C_1, C_2)$):
Input: $L_1 \in C$, $L_2 \in D$.
Question: Is there regular $S$ such that $L_1 \subseteq S$, $L_2 \cap S = \emptyset$?

Regular separator $S$ provides an easy to verify certificate of disjointness of $L_1$ and $L_2$. 
Intersection Emptiness and Regular Separability

Intersection emptiness for language classes $C, D$ (denoted $IE(C_1, C_2)$):
Input: $L_1 \in C$, $L_2 \in D$.
Question: $L_1 \cap L_2 = \emptyset$?

Verifying safety properties in concurrent programs with shared memory reduces to intersection of PDA languages.

Regular Separability for language classes $C, D$ (denoted $RS(C_1, C_2)$):
Input: $L_1 \in C$, $L_2 \in D$.
Question: Is there regular $S$ such that $L_1 \subseteq S, L_2 \cap S = \emptyset$?

Regular separator $S$ provides an easy to verify certificate of disjointness of $L_1$ and $L_2$. 
Decidability of Intersection Emptiness vs Regular Separability

- There exist language classes for which decidability does not coincide: example Visibly Pushdown Languages (undecidable regular separability Kopczyński ’18, decidable intersection emptiness Alur and Madhusudhan ’04).
Decidability of Intersection Emptiness vs Regular Separability

- There exist language classes for which decidability does not coincide: example Visibly Pushdown Languages (undecidable regular separability Kopczyński ’18, decidable intersection emptiness Alur and Madhusudhan ’04).

- However, where language classes are closed under rational transductions (aka full trios), for all known results decidability coincides.
Intersection Emptiness vs Regular Separability for Full Trios

Intersection Emptiness: green=decidable, red=undecidable, blue=open. Regular Separability: ✓=decidable, X=undecidable.

<table>
<thead>
<tr>
<th></th>
<th>1-VASS</th>
<th>OCA</th>
<th>PDA</th>
<th>ℤ-VASS</th>
<th>VASS</th>
<th>WSTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-VASS</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>red</td>
</tr>
<tr>
<td>OCA</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>red</td>
</tr>
<tr>
<td>PDA</td>
<td></td>
<td></td>
<td>X</td>
<td>✓</td>
<td></td>
<td>red</td>
</tr>
<tr>
<td>ℤ-VASS</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>red</td>
</tr>
<tr>
<td>VASS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>red</td>
</tr>
<tr>
<td>WSTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Intersection Emptiness: green=decidable, red=undecidable, blue=open. Regular Separability: ✓=decidable, ✗=undecidable.

<table>
<thead>
<tr>
<th></th>
<th>1-VASS</th>
<th>OCA</th>
<th>PDA</th>
<th>(\mathbb{Z})-VASS</th>
<th>VASS</th>
<th>WSTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-VASS</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>OCA</td>
<td></td>
<td>✗</td>
<td>✗</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDA</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\mathbb{Z})-VASS</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>VASS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WSTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Is this always the case?
Why this Seemed Plausible

Piecewise Testable Separability for full trios shown to be equivalent to Simultaneous Unboundedness Problem. Is Regular Separability equivalent to Intersection Emptiness or another related combinatorial problem?
Why this Seemed Plausible

- Piecewise Testable Separability for full trios shown to be equivalent to Simultaneous Unboundedness Problem.
Why this Seemed Plausible

- Piecewise Testable Separability for full trios shown to be equivalent to Simultaneous Unboundedness Problem.

- Is Regular Separability equivalent to Intersection Emptiness or another related combinatorial problem?
Intersection Emptiness vs Regular Separability for Full Trios II

Intersection Emptiness: green=decidable, red=undecidable, blue=open. Regular Separability: ✓=decidable, ✗=undecidable.

<table>
<thead>
<tr>
<th></th>
<th>1-VASS</th>
<th>OCA</th>
<th>PDA</th>
<th>(\mathbb{Z})-VASS</th>
<th>VASS</th>
<th>WSTS</th>
<th>(C_1)</th>
<th>(C_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-VASS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OCA</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDA</td>
<td>✗</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\mathbb{Z})-VASS</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VASS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WSTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\mathcal{D}_1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✗</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\mathcal{D}_2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Consider $R$, $R \cap f(\Sigma^*) = \emptyset$ if and only if $(R \cup f(L_1)) \cap f(L_2) = \emptyset$.

Regular separability is affected by such distortions.

**Lemma (Unary Regular Separability)**

Let $S_0, S_1 \subseteq N$ and $N \setminus 2 N \subseteq S_1$. Then $a \in S_0$ and $a \in S_1$ are regularly separable if and only if $S_0$ is finite and disjoint from $S_1$. 
Consider $R$, $R \cap f(\Sigma^*) = \emptyset$

$L_1 \cap L_2 = \emptyset \iff (R \cup f(L_1)) \cap f(L_2) = \emptyset$
Key Idea

Consider $R$, $R \cap f(\Sigma^*) = \emptyset$

$$L_1 \cap L_2 = \emptyset \iff (R \cup f(L_1)) \cap f(L_2) = \emptyset$$

Regular Separability is affected by such distortions.
Key Idea

Consider $R$, $R \cap f(\Sigma^*) = \emptyset$

$L_1 \cap L_2 = \emptyset \iff (R \cup f(L_1)) \cap f(L_2) = \emptyset$

Regular Separability is affected by such distortions.

**Lemma (Unary Regular Separability)**

Let $S_0, S_1 \subseteq \mathbb{N}$ and $\mathbb{N} \setminus 2^\mathbb{N} \subseteq S_1$. Then $
abla S_0$ and $\nabla S_1$ are regularly separable if and only if $S_0$ is finite and disjoint from $S_1$. 
Construction of Counterexamples

- The Lemma allows construction of language classes where regular separability reflects finiteness of input language (aka infinity problem).
Construction of Counterexamples

- The Lemma allows construction of language classes where regular separability reflects finiteness of input language (aka infinity problem).
- Starting with a language class that has decidable intersection and undecidable infinity, distort to get language class with decidable intersection and undecidable regular separability; vice versa.
Construction of Counterexamples

- The Lemma allows construction of language classes where regular separability reflects finiteness of input language (aka infinity problem).
- Starting with a language class that has decidable intersection and undecidable infinity, distort to get language class with decidable intersection and undecidable regular separability; vice versa.
- Conclusion: any combinatorial property that characterizes regular separability must be incomparable with intersection emptiness.
THANK YOU