On the Complexity and Decidability of Some Problems Involving Shuffle

Joey Eremondi,
Department of Information and Computing Sciences,
Utrecht University, The Netherlands

Oscar Ibarra,
Department of Computer Science,
University of California, Santa Barbara, USA,

Ian McQuillan,
Department of Computer Science,
University of Saskatchewan, Canada.

Shuffle

- The shuffle operation has received a lot of recent attention.
- It interleaves subwords of two (or k) strings, in a completely nondeterministic fashion.

Shuffle

Let $u, v \in \Sigma^*$. The *shuffle* of u and v, $u \sqcup\!\sqcup v$ is

$$\{u_1 v_1 \cdots u_n v_n \mid u = u_1 \cdots u_n, v = v_1 \cdots v_n, \\ u_i, v_i \in \Sigma^*, 1 \leq i \leq n\}.$$

Shuffle

Problem:

Given a regular language L, does there exist non-trivial R, S such that $L = R \sqcup\!\sqcup S$?

- This problem is open.
- Also open if R, S are restricted to be regular languages.

Decidability

- It is undecidable for context-free languages.
- It is decidable for commutative regular languages, and locally testable languages.

Known Results: Decidability

Proposition (Sosík, L. Kari, TCS 2005)

It is decidable, given regular languages L_1, L_2, R with regular trajectory set T, whether $R = L_1 \sqcup\!\sqcup_T L_2$.

- A trajectory set $T \subseteq \{0,1\}^*$ restricts possible ways of interleaving.

Known Results: Decidability

Proposition (Sosík, L. Kari, TCS 2005)

It is undecidable, given context-free L_1, regular R, L_2, and any T where words have an unbounded number of 0's, whether $R = L_1 \sqcup\!\sqcup_T L_2$.

Simplify to Shuffle of Words

What are some properties of the shuffle of 2 or more words?

Finite Languages

- Shuffle on non-unary words is unique (up to commutation) (Berstel, Boasson, TCS 2002).
- This is not the case for finite languages.
- $(u \sqcup\!\sqcup v) \sqcup\!\sqcup w = u \sqcup\!\sqcup (v \sqcup\!\sqcup w)$.

Shuffle of Words

Interesting questions, complexity of:

1. Given DFA M, is $L(M)$ is decomposable into the shuffle of words, $L(M) = u \sqcup\!\sqcup v$?
2. Given DFA M, $u, v \in \Sigma^+$, is $L(M) = u \sqcup\!\sqcup v$?
3. Given DFA M, $u, v \in \Sigma^+$, is $L(M) \subseteq u \sqcup\!\sqcup v$?
4. Given DFA M, $u, v \in \Sigma^+$, is $u \sqcup\!\sqcup v \subseteq L(M)$?

Solved Questions

Of these four questions, two has been previously solved:

Theorem (Biegler, Daley, McQuillan, TCS 2012)

Let M be a DFA and let $u, v \subseteq \Sigma^*$. Then there is a polynomial time algorithm to decide if $u \sqcup\!\sqcup\, v \subseteq L(M)$.

Solved Questions

Theorem

Let M be a DFA and let $u, v \in \Sigma^+$, where Σ has at least two letters. The problem of determining whether $L(M) \subseteq u \sqcup\!\sqcup v$ is coNP-Complete.

Background

It is also known that the first two questions are related:

1. Given DFA M, is $L(M)$ is decomposable into the shuffle of words, $L(M) = u \sqcup\!\sqcup v$?
2. Given DFA M, $u, v \in \Sigma^+$, is $L(M) = u \sqcup\!\sqcup v$?

Background

Theorem (Biegler, Daley, McQuillan, TCS 2012)

Let M be an acyclic, trim, non-unary DFA. We can find $u, v \in \Sigma^+$ such that, $L(M)$ has a shuffle decomposition into words implies $L(M) = u \sqcup\!\sqcup v$ is the unique decomposition. This can be calculated in $O(|u| + |v|)$ time.

Background

- There is a polynomial time reduction to transform each problem into the other.
- The are equivalent in terms of computational complexity.

Interesting Questions

complexity of:

1. Given DFA M, is $L(M)$ is decomposable into the shuffle of words, $L(M) = u \sqcup\!\sqcup v$?
2. Given DFA M, $u, v \in \Sigma^+$, is $L(M) = u \sqcup\!\sqcup v$?

✓ Given DFA M, $u, v \in \Sigma^+$, is $L(M) \subseteq u \sqcup\!\sqcup v$?
coNP-Complete

✓ Given DFA M, $u, v \in \Sigma^+$, is $u \sqcup\!\sqcup v \subseteq L(M)$?
polynomial

1 and 2 are "equally difficult". Complexity is open.

NFAs Instead of DFAs

Since it is NP-complete to determine, given a DFA M and words u, v over an alphabet of at least two letters, if $L(M) \not\subseteq u \sqcup\!\sqcup\, v$, it follows that:

Corollary

It is NP-complete to determine, given an NFA M and words u, v over an alphabet of at least two letters, if $L(M) \not\subseteq u \sqcup\!\sqcup\, v$.

NFA comparison

Proposition

It is NP-complete to determine, given an NFA M and u, v over an alphabet of at least two letters, whether $u \sqcup\!\sqcup v \not\subseteq L(M)$.

- NP-hardness relies on a reduction with 3SAT.

Non-Equality

Proposition

It is NP-complete to test, given $a^p, b^q \in \Sigma^*, p, q \in \mathbb{N}_0$, and M an NFA over $\Sigma = \{a, b\}$, whether $L(M) \neq a^p \sqcup b^q$.

- This uses a reduction with the problem, given a DFA M and u, v over an alphabet of at least two letters, if $L(M) \not\subseteq u \sqcup v$.

Summary

Complexity of:

- ✓ Given NFA M, $u, v \in \Sigma^+$, is $L(M) \neq u \sqcup\!\sqcup v$?
  NP-complete
- ✓ Given NFA M, $u, v \in \Sigma^+$, is $L(M) \not\subseteq u \sqcup\!\sqcup v$?
  NP-complete
- ✓ Given DFA M, $u, v \in \Sigma^+$, is $u \sqcup\!\sqcup v \not\subseteq L(M)$?
  NP-complete

Recall:

Theorem

Let M be an acyclic, trim, non-unary DFA. We can find $u, v \in \Sigma^+$ such that, $L(M)$ has a shuffle decomposition into words implies $L(M) = u \sqcup\!\sqcup v$ is the unique decomposition. This can be calculated in $O(|u| + |v|)$ time.

NFAs

The following is shown:

Proposition

- *There is an algorithm that, given an acyclic, non-unary NFA M with states Q, can find $u, v \in \Sigma^+$, such that, $L(M)$ has a decomposition into two words implies $L(M) = u \sqcup\!\sqcup\, v$ is the unique decomposition.*
- *This algorithm runs in time $O((|u| + |v|)|Q|^2)$.*

More General Models

- More general models beyond words and finite automata are also of interest.
- Some of the results on words generalize to larger families.
- Some results on words imply partial results on larger families.

Recall:

Known previously:

Given DFA M, $u, v \in \Sigma^+$, is $u \sqcup\!\sqcup v \subseteq L(M)$?

- This can be generalized significantly.

Containment

Proposition

It is decidable, given $M_1, M_2 \in \text{NCM}$ *and* $M_3 \in \text{DCM}$*, whether* $L(M_1) \sqcup L(M_2) \subseteq L(M_3)$*. Moreover, the decision procedure is polynomial in* $n_1 + n_2 + n_3$*, where* n_i *is the size of* M_i*.*

Proposition

It is decidable, given NFAs M_1, M_2 *and* $M_3 \in \text{DPDA}$*, whether* $L(M_1) \sqcup L(M_2) \subseteq L(M_3)$*. Moreover, the decision procedure is polynomial in* $n_1 + n_2 + n_3$*, where* n_i *is the size of* M_i*.*

Containment

- However, it is important that the second family be deterministic.

Proposition

It is undecidable, given one-state DFAs M_1 *accepting* a^* *and* M_2 *accepting* b^*, *and an* $\mathrm{NCM}(1,1)$ *machine* M_3 *over* $\{a, b\}$, *whether* $L(M_1) \sqcup\!\sqcup L(M_2) \subseteq L(M_3)$.

Open Questions

- complexity of testing if the language accepted by a DFA is equal to shuffle of two words.

Open Questions

- When is "$L \subseteq L_1 \sqcup L_2$?" decidable, depending on family of L, and of L_1, L_2?
- It is clearly decidable if all languages regular.
- It's implied from existing results that it is PSPACE-hard if $L = \Sigma^*$ and L_1, L_2 are accepted by NFAs.

Open Questions

- decidability of shuffle decomposition on regular languages.

Thanks!