TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik

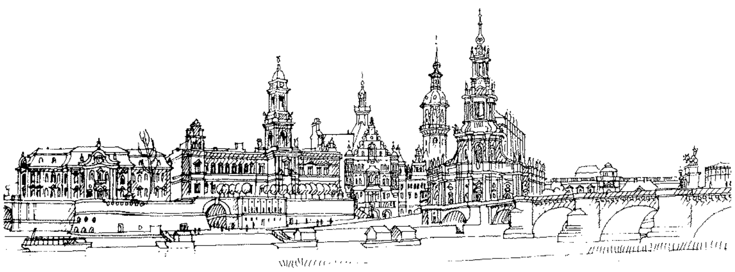
TUD / FI 98 / 09 - October 1998

Daniel Kirsten

Grundlagen der Programmierung
Institut für Softwaretechnik I

A Connection between the Star Problem and the Finite Power Property in Trace Monoids

Technische Berichte
Technical Reports
ISSN 1430-211X



Technische Universität Dresden
Fakultät Informatik
D-01062 Dresden
Germany

URL: http://www.inf.tu-dresden.de/

A Connection between the Star Problem and the Finite Power Property in Trace Monoids*

Daniel Kirsten
Department of Computer Science
Dresden University of Technology
D-01062 Dresden, Germany

Daniel.Kirsten@inf.tu-dresden.de
http://www.inf.tu-dresden.de/~dk11

October 12, 1998

Abstract

This paper deals with a connection between two decision problems for recognizable trace languages: the star problem and the finite power property problem. Due to a theorem by Richomme from 1994 [26, 28], we know that both problems are decidable in trace monoids which do not contain a C4 submonoid. It is not known, whether the star problem or the finite power property are decidable in the C4 or in trace monoids containing a C4.

In this paper, we show a new connection between these problems. Assume a trace monoid $\mathbb{M}(\Sigma, I)$ which is isomorphic to the Cartesian Product of two disjoint trace monoids $\mathbb{M}(\Sigma_1, I_1)$ and $\mathbb{M}(\Sigma_2, I_2)$. Assume further a recognizable language L in $\mathbb{M}(\Sigma, I)$ such that every trace in L contains at least one letter in Σ_1 and at least one letter in Σ_2. Then, the main theorem of this paper asserts that L^* is recognizable iff L has the finite power property.

*This work has been supported by the postgraduate program "Specification of discrete processes and systems of processes by operational models and logics" of the German Research Community (Deutsche Forschungsgemeinschaft).

Contents

1 Introduction

Free partially commutative monoids, also called trace monoids, were introduced by Cartier and Foata in 1969 [3]. In 1977, Mazurkiewicz proposed trace monoids as a potential model for concurrent processes [18], which marks the beginning of a systematic study of trace monoids by mathematicians and theoretical computer scientists, see e.g., the recent surveys [7, 8, 9]. A part of the research in trace theory deals with examinations of well-known classic results for free monoids in the framework of traces.

One main stream in trace theory is the study of recognizable trace languages, which can be considered as an extension of the well studied concept of regular languages in free monoids. A major step in this research is Ochmański's PhD thesis from 1984 [24]. Some of the results concerning regular languages in free monoids can be generalized to recognizable languages in trace monoids. However, there is one major difference: The iteration of a recognizable trace language does not necessarily yield a recognizable language. This fact raises the so called star problem: Given a recognizable language L, is L^* recognizable? In general, it is not known whether the star problem is decidable.

The finite power property problem, for short FPPP, is related to the star problem: Given a recognizable language L, has L the finite power property, for short FPP, i.e., is there a natural number n such that $L^* = L^0 \cup L^1 \cup \ldots \cup L^n$?

The strategy to achieve partial solutions of these problems by examining connections between them turned out to be fruitful. The main result after a stream of publications is a theorem stated by Richomme in 1994, saying that both the star problem and the FPP are decidable in trace monoids which do not contain a particular submonoid called C4 [26, 28]. It is not known whether the star problem and the FPP are decidable in trace monoids with a C4 submonoid. Although the FPP for finite trace languages is obviously decidable, decidability of the star problem for finite trace languages is only known for just a few special cases.

In this paper, we establish a new connection between the star problem and the FPP. We deal with trace monoids which are isomorphic to Cartesian Products of two trace monoids. We show that for a certain class of recognizable languages in Cartesian Products, the iteration of a language is recognizable iff the language has the FPP.

The paper is organized as follows. In Section 2, we show some concepts from algebra and formal language theory. In the first subsection, we get familiar with some basic notions from algebra, formal language theory, and trace theory. In the next subsection, we deal with recognizable sets and rational sets. Then, we discuss the two main problems for recognizable trace languages, their connections, and solutions as far as known. After this discussion, I explain the main results of this paper.

In Section 3 and 4, we deal with concepts which we require to prove the main results. In Section 3, we get familiar with ideal theory which will be a crucial tool. In Section 4, we deal with product automata.

After these preliminary tools, we prove the main theorems in Section 5. In several subsections, we deal with special cases. In the last subsection, we summarize these special cases.

2 Formal Overview

2.1 Monoids, Languages and Traces

I briefly introduce the basic notions from algebra and trace theory. Unless I do not state precise sources, I consider the concepts and notions as well-known.

By $\mathbb{N}$, we denote the set of natural numbers including zero, i.e., $\mathbb{N} = \{0, 1, 2, \ldots\}$.

We say a set K is a *subset* of a set L, and we denote this by $K \subseteq L$ iff every element of K

belongs to L. We say K is a *proper subset* of L, and we denote this by $K \subset L$ iff every element of K belongs to L, and additionally, there is an element in L which does not belong to K.

A *semigroup* $(S, \cdot_S)$ is an algebraic structure consisting of a set S and a binary operation $\cdot_S$ such that $\cdot_S$ is associative, i.e., for every $p, q, r \in S$, we have $(p \cdot_S q) \cdot_S r = p \cdot_S (q \cdot_S r)$. We call S the *underlying set* and we call $\cdot_S$ the *operation* or *product* of S. Usually, we drop the symbol $\cdot_S$ and denote the product by juxtaposition. We call a semigroup $(S, \cdot_S)$ *finite* iff S is a finite set. We use the letter S to denote both the semigroup and its underlying set.

We call a semigroup $(\mathbb{M}, \cdot_{\mathbb{M}})$ a *monoid* iff $\mathbb{M}$ contains an element $\lambda_{\mathbb{M}}$ such that for every $m \in \mathbb{M}$, we have $\lambda_{\mathbb{M}} \cdot_{\mathbb{M}} m = m \cdot_{\mathbb{M}} \lambda_{\mathbb{M}} = m$. We call $\lambda_{\mathbb{M}}$ the *neutral element*. We drop the index at $\lambda_{\mathbb{M}}$ as long as no confusion arises.

In the rest of this subsection, we assume some semigroup S and some monoid $\mathbb{M}$.

For every natural number $n \geq 1$, we define the n-fold product as follows: For every p in S, p^1 yields p itself, and further, for $n \geq 1$, p^{n+1} denotes $p^n p$. For every monoid $\mathbb{M}$, we extend this definition by claiming that for every $p \in \mathbb{M}$, we have $p^0 = \lambda_{\mathbb{M}}$.

We extend the product to subsets of S. If K and L are two subsets of S, the set KL contains all elements kl for some k in K and l in L. We extend the n-fold product to sets. We define $K^1 = K$, and for $n \geq 1$, we set $K^{n+1} = K^n K$. For every subset K of $\mathbb{M}$, we define $K^0 = \{\lambda_{\mathbb{M}}\}$.

For every subset K of S, we define the *non-empty iteration* K^+ as the union of the sets K^1, K^2, K^3, ... For every subset K of $\mathbb{M}$, we additionally define the *iteration* of K by K^* and define it by $K^* := K^+ \cup \{\lambda_{\mathbb{M}}\}$.

For every subset K of S and for every $n \geq 1$, we abbreviate the union $K^1 \cup \ldots \cup K^n$ by $K^{1,\ldots,n}$. For every subset K of $\mathbb{M}$ and for every n, we abbreviate $K^0 \cup \ldots \cup K^n$ by $K^{\leq n}$.

Assume two semigroups S and S'. Their Cartesian Product is the semigroup denoted by $\binom{S}{S'}$ and defined in the following way: The underlying set is the Cartesian Product of the underlying sets of S and S'. We define the operation componentwise, i.e., for every pair of elements $\binom{p_1}{p'_1}$ and $\binom{p_2}{p'_2}$, their product yields $\binom{p_1 p_2}{p'_1 p'_2}$. The products $p_1 p_2$ and $p'_1 p'_2$ are the products in S and S', respectively. The Cartesian Product of two monoids yields a monoid.

Again, assume two semigroups S and S'. We call a function $h : S \to S'$ a *homomorphism* iff h preserves the product, i.e., for every $k, l \in S$, we have $h(k) \cdot_{S'} h(l) = h(k \cdot_S l)$. We call a homomorphism h between two monoids $\mathbb{M}$ and $\mathbb{M}'$ a *monoid homomorphism* iff h preserves the neutral element, i.e., we have $h(\lambda_{\mathbb{M}}) = \lambda_{\mathbb{M}'}$. There are homomorphisms between two monoids which do not preserve the neutral element.

We extend the notion of homomorphisms to sets. If K is a subset of S, we define $h(K)$ as the set of all $q \in S'$ such that for some $p \in K$, we have $h(p) = q$. We denote the *inverse* of h by h^{-1}. We define it on subsets of S': If L is a subset of S', $h^{-1}(L)$ yields the set of all $m \in S$ such that $h(m) \in L$. We call h a *surjective* homomorphism from S to S' iff $h(S) = S'$. We call h an *isomorphism* iff for every $p \in S'$, the set $h^{-1}(\{p\})$ is a singleton. Then, we can regard h^{-1} as a homomorphism from S' to S. We call the semigroups *isomorphic* iff an isomorphism between S and S' exists.

A special kind of homomorphisms are the projections. Assume two semigroups S and S'. We define the *projections* Π_1 and Π_2 from the Cartesian Product $\binom{S}{S'}$ to S and S', respectively: For every $\binom{p}{q} \in \binom{S}{S'}$, we define $\Pi_1\binom{p}{q} = p$ and $\Pi_2\binom{p}{q} = q$.

Assume three semigroups S_1, S_2, and S_3. Assume two homomorphisms $g : S_1 \to S_2$ and $h : S_2 \to S_3$. We denote the *composition* of g and h by $h \circ g$. It yields a homomorphism from S_1 to S_3. For every $p \in S_1$, $h \circ g(p)$ yields $h(g(p))$, i.e., we apply g on p, and we apply h on the result of $g(p)$.

Assume four semigroups S_1, S_2, S'_1, and S'_2. Assume two homomorphisms $g : S_1 \to S'_1$ and $h : S_2 \to S'_2$. We can define a homomorphism $\binom{g}{h}$ from $\binom{S_1}{S_2}$ to $\binom{S'_1}{S'_2}$ as follows: For every

$\binom{p}{q} \in \binom{S_1}{S_2}$, $\binom{g}{h}\binom{p}{q}$ yields $\binom{g(p)}{h(q)}$. If both g and h are surjective homomorphisms from S_1 to S_1' and S_2 to S_2', resp., then $\binom{g}{h}$ is a surjective homomorphism from $\binom{S_1}{S_2}$ to $\binom{S_1'}{S_2'}$. As an exercise, you can verify that the homomorphisms $h \circ \Pi_2$ and $\Pi_2 \circ \binom{g}{h}$ from $\binom{S_1}{S_2}$ to S_2' are identical.

By an *alphabet*, we mean a finite set of symbols. We call its elements *letters*. Assume an alphabet Σ. We denote the *free monoid over* Σ by Σ^*. Its underlying set is the set of all words (strings) consisting of letters of Σ, the monoid product is the concatenation, and the neutral element is the empty string. We call the subsets of the free monoid *languages*. For every word w in Σ^*, we call the number of letters of w the *length* of w, and denote it by $|w|$.

Cartier and Foata introduced the concept of the free partially commutative monoids in 1969 [3]. In 1977, Mazurkiewicz considered this concept as a potential model for concurrent systems [18]. Since then, free partially commutative monoids are examined by both mathematicians and theoretical computer scientists. For a general overview, I recommend the surveys [7, 8, 9].

Assume an alphabet Σ. We call a binary relation I over Σ an *independence relation* iff I is irreflexive and symmetric. For every pair of letters a and b with aIb, we say that a and b are *independent*, otherwise a and b are *dependent*. We call the pair (Σ, I) an *independence alphabet*. We call two words w_1, w_2 in Σ^* equivalent iff we can transform w_1 into w_2 by finitely many exchanges of independent adjacent letters which we denote by $w_1 \sim_I w_2$. For instance, if a and c are independent letters, *baacbac*, *bacabac* and *bcaabca* are mutually equivalent words.

The relation $\sim_I$ is an equivalence relation. For every word w in Σ^*, we denote by $[w]_I$ the equivalence class of w. Moreover, $\sim_I$ is a congruence relation. This means, for every words w_1, w_1', w_2, w_2' in Σ^* with $w_1 \sim_I w_2$ and $w_1' \sim_I w_2'$, we have $w_1 w_1' \sim_I w_2 w_2'$. Therefore, we can define a monoid with the sets $[w]_I$ as elements. For any words w_1 and w_2, we define the product of $[w_1]_I$ and $[w_2]_I$ by $[w_1 w_2]_I$. We denote this monoid by $\mathbb{M}(\Sigma, I)$ and call it the *trace monoid* over Σ and I. Its elements, i.e., the equivalence classes $[w]_I$, are called *traces* and its subsets are called *trace languages* or shortly *languages*. The function $[\,]_I$ is a homomorphism from the free monoid Σ^* to $\mathbb{M}(\Sigma, I)$. As long as no confusion arises, we omit the index I at $[\,]_I$.

If I is the empty relation over Σ, the trace monoid $\mathbb{M}(\Sigma, I)$ is the free monoid. If I is the largest irreflexive relation over Σ, i.e., two letters a and b are independent iff a and b are different, then the trace monoid $\mathbb{M}(\Sigma, I)$ is the free commutative monoid over Σ.

Assume two disjoint independence alphabets (Σ_1, I_1) and (Σ_2, I_2). We can define another independence alphabet (Σ, I) by merging the alphabets (Σ_1, I_1) and (Σ_2, I_2) in the following way: $(\Sigma, I) = \Big(\Sigma_1 \cup \Sigma_2\,,\, I_1 \cup I_2 \cup (\Sigma_1 \times \Sigma_2) \cup (\Sigma_2 \times \Sigma_1)\Big)$. Two letters a and b in Σ are independent iff either they do not belong to the same alphabet or both a and b belong to Σ_1 (resp. Σ_2) and we have aI_1b (resp. aI_2b). The trace monoid $\mathbb{M}(\Sigma, I)$ is naturally isomorphic to the Cartesian Product $\binom{\mathbb{M}(\Sigma_1, I_1)}{\mathbb{M}(\Sigma_2, I_2)}$. Consequently, we can regard the Cartesian Product of two trace monoids as a trace monoid.

We define the trace monoid P3 by the independence alphabet consisting of $\Sigma_{\mathrm{P3}} = \{a, b, c\}$ and $I_{\mathrm{P3}} = \Big\{(a, b), (b, a), (c, b), (b, c)\Big\}$. The P3 is isomorphic to the Cartesian Product $\binom{\{a,c\}^*}{\{b\}^*}$. We define the trace monoid C4 by the independence alphabet consisting of $\Sigma_{\mathrm{C4}} = \{a, b, c, d\}$ and $I_{\mathrm{C4}} = I_{\mathrm{P3}} \cup \Big\{(a, d), (d, a), (c, d), (d, c)\Big\}$. The C4 is isomorphic to $\binom{\{a,c\}^*}{\{b,d\}^*}$.

Assume an independence alphabet (Σ, I). A trace t in $\mathbb{M}(\Sigma, I)$ is called *connected* iff for every non-empty traces t_1 and t_2 with $t = t_1 t_2$, there is a letter a occurring in t_1 and there is a letter b occurring in t_2, such that a and b are dependent. A trace language L is called *connected* iff every trace in L is connected. A trace $\binom{u}{v}$ in P3 or C4 is connected iff u or v is the empty word λ.

Assume some trace monoid $\mathbb{M}(\Sigma, I)$. Assume a trace language T in $\mathbb{M}(\Sigma, I)$ such that for every traces $t_1, t_2 \in T$, their concatenation $t_1 t_2$ belongs to T. Then, we say that T is *concatenation closed*. Moreover, T is a semigroup. If additionally $\lambda \in T$, then T is a monoid. Consequently, it is quite natural to ask for generators of T.

Definition 2.1 Assume a trace monoid $\mathbb{M}(\Sigma, I)$ and some concatenation closed language T in $\mathbb{M}(\Sigma, I)$ such that $\lambda \notin T$. The set of *generators* of T is defined by $\text{Gen}(T) = T \setminus T^2$. □

Consequently, the generators of T are the traces in T which cannot be factorized into some traces in T.

Lemma 2.2 Assume a trace monoid $\mathbb{M}(\Sigma, I)$ and a concatenation closed language T in $\mathbb{M}(\Sigma, I)$ such that $\lambda \notin T$. We have

1. $\text{Gen}(T)^+ = T$, and
2. for every language $L \subseteq \mathbb{M}(\Sigma, I)$ with $L^+ = T$, we have $\text{Gen}(T) \subseteq L$. □

Proof: At first, we show assertion 1. We have $\text{Gen}(T) \subseteq T$. Hence, we have $\text{Gen}(T)^+ \subseteq T^+$. Because T is concatenation closed, we have $T^+ = T$. Consequently, we have $\text{Gen}(T)^+ \subseteq T$.

We show that T is a subset of $\text{Gen}(T)^+$ by a contradiction. Assume that there are traces in $T \setminus \text{Gen}(T)^+$. Then, let t be a smallest trace in $T \setminus \text{Gen}(T)^+$, i.e., for every trace t' in T with $|t'| < |t|$, we have $t' \in \text{Gen}(T)^+$. The trace t does not belong to $\text{Gen}(T)^+$. Thus, t does not belong to $\text{Gen}(T)$, i.e., t does not belong to $T \setminus T^2$. Consequently, t belongs to T^2. We can factorize t into t_1 and t_2 in T. We have $|t_1| < |t|$ and $|t_2| < |t|$. Hence, the traces t_1 and t_2 belong to $\text{Gen}(T)^+$, but, t does not belong to $\text{Gen}(T)^+$. This contradicts that $\text{Gen}(T)^+$ is concatenation closed.

Now, we prove assertion 2 by a contradiction. Assume a language L such that L^+ yields T, but, $\text{Gen}(T)$ is not a subset of L. Then, let t be a trace in $\text{Gen}(T) \setminus L$. The trace t belongs to $\text{Gen}(T)$. Thus, t belongs to T, i.e., t belongs to L^+. Because t does not belong to L, we can factorize t into two non-empty traces t_1 and t_2 in L^+. Thus, t belongs to $(L^+)^2$. This means, t belongs to T^2. Consequently, t does not belong to $T \setminus T^2$, i.e., we contradicted that t belongs to $\text{Gen}(T)$. □

2.2 Recognizable Sets

The concept of recognizability describes a formal method how to use finite machines to deal with infinite objects. It originates from Mezei and Wright from 1967 [22]. There are numerous equivalent definitions. I introduce it as far as we use it in this paper, for a more general overview I recommend [1, 10]. I took most of the contents of this section from there.

Definition 2.3 Assume a monoid $\mathbb{M}$. An $\mathbb{M}$-*automaton* is a triple $\mathcal{A} = [Q, h, F]$, where Q is a finite monoid, h is a homomorphism $h : \mathbb{M} \to Q$ and F is a subset of Q. The *language* of an $\mathbb{M}$-automaton $\mathcal{A}$ is defined by $L(\mathcal{A}) = h^{-1}(F)$. □

We call Q the underlying monoid of $\mathcal{A}$, and the elements of Q the states of $\mathcal{A}$. We further call F the set of accepting states of $\mathcal{A}$, and h the homomorphism of $\mathcal{A}$. Without loss of generality, we can assume that h is a surjective homomorphism from $\mathbb{M}$ to F. If $L(\mathcal{A}) = L$, then we say that $\mathcal{A}$ defines L or $\mathcal{A}$ is an $\mathbb{M}$-automaton for L. We call a subset L of $\mathbb{M}$ a *recognizable* language over $\mathbb{M}$ iff there is an $\mathbb{M}$-automaton $\mathcal{A}$, such that $L = L(\mathcal{A})$. We denote the class of all recognizable languages over $\mathbb{M}$ by $\text{REC}(\mathbb{M})$.

Definition 2.3 shows a common way to define recognizability in arbitrary monoids. Following Courcelle [6], we call the triple $[Q, h, F]$ an $\mathbb{M}$-automaton.

Assume a product automation $\mathcal{A}$ such that $L(\mathcal{A})$ is closed under monoid product. Then, F is the image of the semigroup $L(\mathcal{A})$ under the surjective homomorphism h. Hence, F is a subsemigroup of Q.

The following theorem is a classic one, you find the proof, e.g., in [1, 10].

Theorem 2.4 Assume a monoid $\mathbb{M}$. The class $\mathrm{REC}(\mathbb{M})$ contains the empty set $\emptyset$, $\mathbb{M}$ itself and it is closed under union, intersection, complement and inverse homomorphisms. $\square$

There are monoids containing finite subsets which are not recognizable and monoids in which the product of two recognizable subsets is not always a recognizable set. And further, the iteration of a recognizable set is not always recognizable. However, we have the following theorem for trace monoids:

Theorem 2.5 Assume a trace monoid $\mathbb{M}(\Sigma, I)$. The class $\mathrm{REC}(\mathbb{M}(\Sigma, I))$ contains all finite subsets of $\mathbb{M}(\Sigma, I)$, and it is closed under monoid product and under iteration of *connected* trace languages. $\square$

Recognizability of finite trace languages is obvious. The proof of the closure under monoid product originates from Fliess [11]. Closure under iteration of connected trace languages is due to Ochmański [24], Clerbout and Latteux [4], and Métivier [19]. In [23], you find a recent survey on recognizable trace languages, it contains neat little proofs of the assertions in Theorem 2.5.

Example 2.6 Assume the free monoids $\mathbb{M}_1 = \{a\}^*$ and $\mathbb{M}_2 = \{b\}^*$. Let L_1 be the singleton language $\left\{\binom{a}{b}\right\}$ in the trace monoid $\binom{\mathbb{M}_1}{\mathbb{M}_2}$. The language L_1 is recognizable, because it is finite. Further, L_1 is not a connected language. We have $L_1^* = \left\{ \binom{a^n}{b^n} \,\middle|\, n \in \mathbb{N} \right\}$. Below, we show that L_1^* is not recognizable. $\square$

We need a theorem by Mezei concerning recognizable sets in Cartesian Products. It is not published by the author himself, but, it is widely known as Mezei's Theorem, you find it in, e.g., [1, 10].

Theorem 2.7 Assume two monoids $\mathbb{M}$ and $\mathbb{M}'$. A set L is recognizable in $\binom{\mathbb{M}}{\mathbb{M}'}$ iff there are a number n, recognizable sets $L_1, \ldots, L_n \subseteq \mathbb{M}$ and recognizable sets $L'_1, \ldots, L'_n \subseteq \mathbb{M}'$, such that:

$$L = \binom{L_1}{L'_1} \cup \ldots \cup \binom{L_n}{L'_n} \qquad \square$$

Example 2.6 (continued) We show by a contradiction that L_1^* is not recognizable. Assume that L_1^* is recognizable. Then, by Theorem 2.7, L_1^* is the union of finitely many Cartesian Products. Consequently, there are two numbers $i \neq j$, such that $\binom{a^i}{b^i}$ and $\binom{a^j}{b^j}$ belong to the same Cartesian Product. Hence, the traces $\binom{a^i}{b^j}$ and $\binom{a^j}{b^i}$ belong to this Cartesian Product, i.e., they belong to L_1^*. This is a contradiction. $\square$

Let us shortly mention the notion of rational sets. Assume some monoid $\mathbb{M}$. The class of rational subsets of $\mathbb{M}$ is the smallest class which contains the empty set and every singleton subset of $\mathbb{M}$, and is closed under union, monoid product, and iteration. We have Kleene's classic result which asserts that in free monoids the recognizable sets and the rational sets coincide [17, 31]. In trace monoids, we have just one direction: Due to a more general result by McKnight, every recognizable trace language is a rational trace language. However, there are rational trace languages which are not recognizable unless the underlying trace monoid is a free monoid. For instance, the language L_1^* in Example 2.6 is a rational language which is not recognizable. See [2, 8] for more information on rational trace languages.

2.3 Some Decision Problems for Trace Languages

Two decision problems concerning the iteration of recognizable trace languages arise:

- **Star Problem:** Can we decide whether the iteration of a recognizable trace language yields a recognizable language?

- **Finite Power Property Problem:** Can we decide whether a recognizable language has the finite power property, i.e., for a recognizable language L, can we decide whether there is a natural number n such that we have $L^* = L^{\leq n}$.

By FPP, we abbreviate the finite power property, and by FPPP, we abbreviate the finite power property problem. If a recognizable language L has the FPP, then we have $L^* = L^0 \cup L^1 \cup \ldots \cup L^n$ for some $n \in \mathbb{N}$. Hence, if L has the FPP, then L^* is recognizable by Theorem 2.4 and 2.5. Below, we will see languages L such that L^* is recognizable but L does not have the FPP.

Although during the recent 14 years many papers have dealt with the star problem and the FPPP, only partial results have been achieved. In general, both problems have remained unsolved. I give a survey about their history.

The star problem in the free monoid is trivial due to Kleene's Theorem from 1956, and it is decidable in free commutative monoids due to Ginsburg and Spanier [13, 14]. Brzozowski raised the FPPP in the free monoid in 1966, and it took more than ten years till Simon and Hashiguchi independently showed its decidability [30, 16]. In 1984, Ochmański examined recognizable trace languages in his PhD thesis [24] and stated the star problem. During the eighties, Ochmański [24], Clerbout and Latteux [4], and Métivier [19] independently proved that the iteration of a connected recognizable trace language yields a recognizable trace language. In 1990, Ochmański showed the decidability of the star problem in C4 for finite trace languages containing at most one non-connected trace [25]. He used the decidability of the FPP in free monoids. This marks the beginning of the examination of connections between the FPP and the star problem. In 1992, Sakarovitch solved the recognizability problem [29]:

Theorem 2.8 Assume a trace monoid $\mathbb{M}(\Sigma, I)$. We can decide whether a rational language in $\mathbb{M}(\Sigma, I)$ is recognizable iff $\mathbb{M}(\Sigma, I)$ does not contain a P3 submonoid. □

This theorem includes that the star problem is decidable in trace monoids which do not contain a P3 submonoid. For a short time, one had hope to solve the star problem. One conjectured that the above theorem can be generalized to the star problem. However, just in the same year, Gastin, Ochmański, Petit and Rozoy showed the decidability of the star problem in P3 [12]. Decidability of the FPP in free monoids plays a crucial role in their proof.

During the subsequent years, Métivier and Richomme developed these ideas. They showed decidability of the FPP for connected trace languages and decidability of the star problem for trace languages containing at most four traces as well as for finite languages containing at most two connected traces [20, 21]. They showed the following connections between the star problem and the FPP.

Theorem 2.9 Assume an independence alphabet (Σ, I) and a letter b with $b \notin \Sigma$.

1. If the star problem is decidable in $\binom{\mathbb{M}(\Sigma,I)}{\{b\}^*}$, then the FPP is decidable in $\mathbb{M}(\Sigma, I)$ [20, 21].

2. If both the star problem and the FPP are decidable in $\mathbb{M}(\Sigma, I)$ then both problems are decidable in $\binom{\mathbb{M}(\Sigma,I)}{\{b\}^*}$ [26, 28]. □

Assertion 1 is a conclusion from the following connection [20, 21]:

Theorem 2.10 Assume an independence alphabet (Σ, I) and a letter b with $b \notin \Sigma$. Assume a recognizable language T in $\mathbb{M}(\Sigma, I)$ and let $K = \binom{T}{\{b\}^+}$. Then, the language T has the FPP iff K^* is recognizable. □

An obvious conclusion from assertion 1 in Theorem 2.9 is that if the star problem is decidable in any trace monoid, then so is the FPP [20, 21]. Richomme improved the induction in assertion 2 and showed the following theorem:

Theorem 2.11 Assume a trace monoid $\mathbb{M}(\Sigma, I)$. If the monoid $\mathbb{M}(\Sigma, I)$ does not contain a C4-submonoid, then the star problem and the FPP are decidable. □

The main ideas of the proof are in [28], the complete proof is in [26]. Please note that the *if* in the theorem has just one *f*.

In 1994, Richomme tried to prove that the trace monoids with a decidable star problem are exactly the trace monoids with a decidable FPP [27]. However, one of the proofs in this report contains an error such that the result was not proved.

The subsequent years were designated by stagnation. One did not achieve new results and one ceased the research on the star problem and the FPPP. Today, especially two questions are interesting:

1. Are the star problem and the FPP decidable in trace monoids which contain a C4 submonoid?

2. Are the trace monoids with a decidable FPP exactly the trace monoids with a decidable star problem?

2.4 Main Results

In this paper, we work on connections between the star problem and the FPP. As already mentioned, for every recognizable language L, the iteration of L is recognizable if L has the FPP. The other direction is not true. Let us consider some examples in the trace monoid $\binom{\{a\}^*}{\{b\}^*}$.

Example 2.12 Assume the language $L_2 = \left\{\binom{a}{\lambda}, \binom{\lambda}{b}\right\}$. Both L_2 and L_2^* are recognizable, because L_2 is finite and L_2^* is the complete monoid. However, L_2 does not have the FPP. For every $n \in \mathbb{N}$, the language $L_2^{\leq n}$ contains only traces consisting of at most n letters, i.e., $L_2^{\leq n}$ is finite, and thus, it is different from L_2^*. □

Example 2.13 We examine the language $L_3 = \binom{\{a\}}{\{b\}^*}$. We have $L_3^* = \left\{\binom{\lambda}{\lambda}\right\} \cup \binom{\{a\}^+}{\{b\}^*}$. Both L_3 and L_3^* are recognizable. Recognizability of L_3^* follows, e.g., from Theorem 2.7. As well as L_2, the language L_3 does not posses the FPP. For every $n \in \mathbb{N}$, $L_3^{\leq n}$ contains only traces in which the letter a occurs at most n times, i.e., $L_3^{\leq n}$ is different from L_3^*. □

Example 2.14 Assume the recognizable language $L_4 = \left\{\binom{\lambda}{\lambda}\right\} \cup \binom{\{a\}^+}{\{b\}^+}$. We have $L_4 = L_4^*$. Hence, L_4 has the FPP. We ask whether there is a language L such that $L^* = L_4$ and further L does not have the FPP. We show that such a language L does not exist. Just assume a language L such that $L^* = L_4$.

Assume some number $n \geq 1$. The trace $\binom{a}{b^n}$ belongs to L_4, i.e, it belongs to L^*. We cannot factorize $\binom{a}{b^n}$ into some non-empty traces in L_4. Otherwise, we could conclude that $\binom{a}{b^n}$ contains the letter a more than once. Hence, $\binom{a}{b^n}$ belongs to L. Accordingly, the trace $\binom{a^n}{b}$ belongs to L. To sum up, for every $n \geq 1$, the traces $\binom{a}{b^n}$ and $\binom{a^n}{b}$ belong to L.

Now, assume two numbers $n > 1$, $m > 1$. The trace $\binom{a^n}{b^m}$ belongs to L_4. We do not know whether it belongs to L. However, we can factorize it into the traces $\binom{a}{b^{m-1}}$ and $\binom{a^{n-1}}{b}$ in L such that $\binom{a^n}{b^m}$ belongs to L^2. Hence, for every $n > 1$, $m > 1$, the trace $\binom{a^n}{b^m}$ belongs to L^2.

Thus, we have $L_4 = L^{\leq 2}$. Consequently, for every language L such that $L^* = L_4$, we know that L has the FPP. Note that this statement is not restricted to recognizable languages L. □

By examining many similar examples, one makes an observation. Assume two disjoint trace monoids $\mathbb{M}_1$ and $\mathbb{M}_2$. Whenever one considers recognizable languages L in $\binom{\mathbb{M}_1}{\mathbb{M}_2}$, such that L^* is recognizable but L does not have the FPP, then certain traces play a crucial role. Namely, traces which have an empty and a non-empty compound, i.e., non-empty traces in $\binom{\{\lambda\}}{\mathbb{M}_2}$ and $\binom{\mathbb{M}_1}{\{\lambda\}}$ play a crucial role. This leads to Theorem 2.15, which is the main result of this paper. Richomme stated it in [27], but, the proof there contains an error. Up to now, it remained open to correct the error or to disprove the theorem.

Theorem 2.15 Assume two disjoint independence alphabets (Σ_1, I_1) and (Σ_2, I_2). Assume some recognizable language L in $\binom{\mathbb{M}(\Sigma_1,I_1)}{\mathbb{M}(\Sigma_2,I_2)}$ such that every trace in L contains at least one letter of Σ_1 and at least one letter of Σ_2.
Then, the language L^* is recognizable iff L has the FPP. □

We will prove Theorem 2.15 as a corollary of the following theorem. Its proof is the main part of the present paper.

Theorem 2.16 Assume two disjoint independence alphabets (Σ_1, I_1) and (Σ_2, I_2). Assume some recognizable, concatenation closed language T in $\binom{\mathbb{M}(\Sigma_1,I_1)}{\mathbb{M}(\Sigma_2,I_2)}$ such that every trace t in T contains at least one letter of Σ_1 and at least one letter of Σ_2.
Then, the set of generators of T has the FPP. □

We close this section by deriving Theorem 2.15 from Theorem 2.16.

Proof of Theorem 2.15: Assume some recognizable language L in $\binom{\mathbb{M}_1}{\mathbb{M}_2}$ such that every trace in L contains at least one letter from Σ_1 and at least one letter from Σ_2. If L has the FPP, then L^* is recognizable because of the closure properties of recognizable trace languages in Theorem 2.4 and 2.5.

To show the other direction, assume that L^* is recognizable. We examine L^+. It is also recognizable. Further, every trace in L^+ contains at least one letter of Σ_1 and at least one letter of Σ_2. We can use Theorem 2.16 on L^+. Hence, there is a natural number $l > 1$ such that we have $\mathrm{Gen}(L^+)^{1,\ldots,l} = L^+$.

By Lemma 2.2, we have $\mathrm{Gen}(L^+) \subseteq L$. Hence, we have $\mathrm{Gen}(L^+)^{1,\ldots,l} \subseteq L^{1,\ldots,l}$. Thus, we have $L^+ \subseteq L^{1,\ldots,l}$, i.e., $L^+ = L^{1,\ldots,l}$. Consequently, we have $L^* = L^{\leq l}$, i.e., L has the FPP. □

3 Semigroups and Ideals

We deal with some notions on semigroups. In the first subsection, we get familiar with ideals. Then, we examine non-empty finite semigroups without and with proper ideals. In the last subsection, we use ideal theory to work out a useful classification of non-empty finite semigroups. To understand the rest of the paper, you have to become familiar with the notions of left ideals and ideals and you have to understand Lemma 3.2 and Proposition 3.7.

3.1 Basic Definitions and Notions

Ideal theory originates mainly from Green. This subsection contains a suitable adaptation of a tiny selection of notions and results from ideal theory. For more detailed information, I recommend teaching books on semigroups, e.g. [5, 15], rather than books concerning automata theory. I developed this subsection in a way that the reader does not require previous knowledge in ideal theory.

As already mentioned, a *semigroup* is a set together with a binary associative operation. Assume some semigroup S. We call a semigroup H a subsemigroup of S iff the underlying set of H is a subset of S and the operation of H is the operation of S restricted to the elements in H. Hence, we can regard H as a subset of S which is closed under the operation of S.

We call a subset U of S a *left ideal* of S iff we have $SU \subseteq U$. Hence, a left ideal is a special subsemigroup. Every semigroup has itself and the empty set as left ideals. We call a left ideal U of S *proper* iff we have $\emptyset \subset U \subset Q$. The intersection and the union of two left ideals of S yield left ideals of S.

We call a subset J of S an *ideal* of S iff we have $JS \subseteq J$ and $SJ \subseteq J$. Hence, an ideal is a special left ideal. We call an ideal J of S *proper* iff we have $\emptyset \subset J \subset Q$.

For simplicity, we develop the following notions just for finite semigroups. Assume some finite semigroup Q. Assume some ideal J of Q which is different from Q. We call a left ideal U of Q *J-minimal* iff we have $J \subset U$ and there is not any left ideal U' with $J \subset U' \subset U$. Assume two different J-minimal left ideals U and V of Q. Their intersection contains J. Assume J is properly contained in the left ideal $U \cap V$. Then, one of the left ideals U or V is not J-minimal, because we have $J \subset (U \cap V) \subset U$ or $J \subset (U \cap V) \subset V$. Consequently, we have $U \cap V = J$. If $J = \emptyset$, we shortly say *minimal* instead of $\emptyset$-minimal.

We close this subsection with a technical lemma:

Lemma 3.1 Assume a non-empty finite semigroup Q and an ideal $J \neq Q$. Then, the union of all J-minimal left ideals yields an ideal of Q. □

Proof: There is at least one left ideal properly containing J, namely Q itself. Hence, there is also some smallest left ideal which contains J properly.

Let J' be the union of all J-minimal left ideals. Then, J' is a left ideal with $J \subset J'$. We have to show $J'Q \subseteq J'$. We prove this by showing that for every J-minimal left ideal L and for every element $q \in Q$, the set $J \cup Lq$ yields J or some J-minimal left ideal. Just assume $J \subset (J \cup Lq)$.

Because L is a left ideal, we have $QL \subseteq L$. Thus, we have $QLq \subseteq Lq$. Therefore, Lq and $J \cup Lq$ are left ideals of Q.

Now, we show by a contradiction that $J \cup Lq$ is J-minimal. Just assume a left ideal K such that we have $J \subset K \subset (J \cup Lq)$. We define a set K' by $K' := \{x \in L \mid xq \in K\}$. We show the proper inclusions $J \subset K' \subset L$.

We have $J \subseteq L$ and $Jq \subseteq J \subset K$. Hence, we have $J \subseteq K'$.

We show that the inclusion $J \subseteq K'$ is proper: There is some $p \in K \setminus J$. Then, $p \in Lq$. Hence, there is some $p' \in L$ with $p = p'q$. We have $p' \notin J$, because J is an ideal and $p = p'q$ does not belong to J. However, $p' \in K'$.

The inclusion $K' \subseteq L$ is obvious. There is some $r \in (J \cup Lq) \setminus K$. Then, we have $r \in Lq \setminus J$. Thus, there is some $r' \in L$ with $r'q = r$. Then, $r' \notin K'$, i.e., we have $K' \subset L$.

We show that K' is a left ideal. Just assume some x in K' and some y in Q. We have $yx \in L$, because x belongs to L which is a left ideal. Further, we have $yxq \in K$, because xq belongs to the left ideal K. Thus, we have $yx \in K'$.

Hence, the set K' is a left ideal with $J \subset K' \subset L$, i.e., L is not J-minimal. This is a contradiction, such that the assumed left ideal K does not exist. Therefore, the set $J \cup Lq$ is a J-minimal left ideal. □

3.2 Finite Semigroups without Proper Ideals

In this subsection, we examine non-empty finite semigroups without proper ideals. Assume some non-empty finite semigroup Q without proper ideals. We distinguish two cases: Q has or Q does not have proper left ideals. At first, we deal with the case that Q does not have proper left ideals.

Lemma 3.2 Assume a non-empty finite semigroup Q which has not any proper left ideal. Then, for every elements p, p' and q in Q, the equality $pq = p'q$ implies $p = p'$. $\Box$

Proof: We show that a counter example implies the existence of a proper left ideal. Just assume three elements $p, p', q \in Q$ such that $pq = p'q$, but nevertheless, $p \neq p'$. We have $QQ \subseteq Q$, and thus, $Q\,Qq \subseteq Qq$ such that Qq is a left ideal. Further, the product Qq yields a proper subset of Q, because the result of the product pq "occurs twice", such that at least one element of Q cannot occur in Qq. Consequently, the set Qq is a proper left ideal of Q. This contradicts the presumption of the lemma. $\Box$

Now, we examine non-empty finite semigroups with proper left ideals but without proper ideals.

Example 3.3 Assume the semigroup Q_1 with the underlying set $\{a', \hat{a}, \bar{a}, c', \hat{c}, \bar{c}\}$. We define the operation in Q_1 as follows: For every two elements p and q in Q_1, the result of the product pq has the letter (i.e. a or c) of p with the index (i.e. $'$, $\hat{}$ or $\bar{}$) of q. For instance, we have $a'c' = \hat{a}c' = \bar{a}a' = a'$ and $\hat{c}\bar{c} = c'\bar{a} = \bar{c}\bar{c} = \bar{c}$. This operation is associative. Obviously, for two elements p and q of Q_1, we have $qpq = q$.

The semigroup Q_1 does not have proper ideals. Assume J is a non-empty ideal of Q_1. Assume an element $p \in J$. For every $q \in Q_1$, the product qp belongs to J. Then, we also have $qpq = q \in J$. Thus, $J = Q$.

It is immediate that the subsets $\{a', c'\}$, $\{\hat{a}, \hat{c}\}$ and $\{\bar{a}, \bar{c}\}$ are the minimal left ideals of Q_1. By merging two of these minimal left ideals, we obtain three proper left ideals of Q_1 which are not minimal. $\Box$

Lemma 3.4 Assume a non-empty finite semigroup Q with at least one proper left ideal but without proper ideals. Then, Q has two disjoint proper left ideals U and V with $Q = U \cup V$. $\Box$

Proof: We apply Lemma 3.1 with $J = \emptyset$. The union of all minimal left ideals of Q yields an ideal of Q. Because Q does not have proper ideals, the union of all minimal left ideals of Q yields Q itself.

Now, assume that Q has exactly one minimal left ideal. Then, this minimal left ideal is Q itself. Thus, the semigroup Q does not have proper left ideals, which is a contradiction. Hence, Q has at least two minimal left ideals.

Let U be an minimal left ideal and let V be the union of all other minimal left ideals of Q. Then, the subsets U and V are two disjoint left ideals and their union yields Q. $\Box$

Example 3.3 (continued) The semigroup Q_1 fulfills Lemma 3.4, e.g., we choose $U = \{a', c'\}$ and $V = \{\hat{a}, \bar{a}, \hat{c}, \bar{c}\}$. $\Box$

By Lemma 3.2 and 3.4, we have strong assertions for finite semigroups without proper ideals. Depending on whether a finite semigroup without proper ideals has a proper left ideal or not, we can apply either Lemma 3.4 or Lemma 3.2.

3.3 Finite Semigroups with Proper Ideals

In this subsection, we deal with finite semigroups with proper ideals.

Lemma 3.5 Assume a non-empty finite semigroup Q with a proper ideal. Then, there is a proper ideal J of Q such that Q and J fulfill one of the following assertions:

1. The set $Q \setminus J$ yields a singleton $\{r\}$ and r^2 belongs to J.
2. The set $Q \setminus J$ yields a subsemigroup of Q.
3. There are two proper left ideals U and V of Q, such that $U \cup V = Q$ and $U \cap V = J$. □

Note that in assertion three both the left ideals U and V are different from J. Just assume that $U = J$. Then, we have $J \cup V = Q$. Because J is contained in V, we have $V = Q$, which contradicts that V is a proper left ideal.

Proof: Let J be a proper ideal of Q such that there is not any ideal J' with $J \subset J' \subset Q$. Such an ideal exists because Q is finite and Q has at least one proper ideal. We show that Q and J fulfill assertion 3, provided that they contradict assertion 1 and 2.

Since J is proper, there is some $r \in Q \setminus J$. Then, $Q \setminus J = \{r\}$ implies assertion 1 or 2, depending on whether $r^2 \in J$ or $r^2 = r$. Hence, $Q \setminus J$ contains at least two elements.

Since, $Q \setminus J$ is not a subsemigroup of Q, there are $p, q \in Q \setminus J$ with $pq \in J$. We have $J \cup Qq = J \cup \Big(J \cup \{p\} \cup Q \setminus J \setminus \{p\}\Big)q = J \cup Jq \cup \{pq\} \cup \Big(Q \setminus J \setminus \{p\}\Big)q$. The sets Jq and $\{pq\}$ are contained in J such that we have $J \cup Qq = J \cup \Big(Q \setminus J \setminus \{p\}\Big)q$.

Now, we have $\Big|J \cup Qq\Big| = \Big|J \cup (Q \setminus J \setminus \{p\})q\Big| \leq \Big|J\Big| + \Big|(Q \setminus J \setminus \{p\})q\Big| \leq \Big|J\Big| + \Big|Q \setminus J \setminus \{p\}\Big|$. We have $p \in Q \setminus J$, and thus, $\Big|J\Big| + \Big|Q \setminus J \setminus \{p\}\Big| < \Big|J\Big| + \Big|Q \setminus J\Big| = \Big|Q\Big|$. Therefore, we have $\Big|J \cup Qq\Big| < \Big|Q\Big|$. Hence, we have the proper inclusion $J \cup Qq \subset Q$.

We show the existence of some left ideal U' of Q with $J \subset U' \subset Q$. Assume that Qq is not a subset of J. Then, the union $J \cup Qq$ yields the desired left ideal U'. Assume that Qq is a subset of J. Then, the set $J \cup \{q\}$ is the desired left ideal U'. The inclusion $(J \cup \{q\}) \subset Q$ is proper since $Q \setminus J$ contains at least two different elements.

Now, we can apply Lemma 3.1. The union of all J-minimal left ideals of Q yields an ideal. This ideal properly contains J. The only ideal properly containing J is Q itself. Hence, the union of all J-minimal left ideals yields Q itself.

Assume there is exactly one J-minimal left ideal. Then, this J-minimal left ideal is Q itself. However, Q cannot be a J-minimal left ideal, because we have shown that there is some left ideal U' with $J \subset U' \subset Q$. Therefore, there are at least two different J-minimal left ideals.

Now, let U be a J-minimal left ideal and let V be the union of all other J-minimal left ideals. Then, U and V are the desired left ideals in assertion 3. □

The assertions in Lemma 3.5 are not exclusive. Assume a non-empty finite semigroup Q with a proper ideal J such that Q and J satisfy assertion 1 in Lemma 3.5. They obviously contradict assertion two by $r^2 \in J$. They also contradict assertion three. Depending on whether r belongs to U, we either have $J = U$ or $Q = U$, which is a contradiction. However, assertions 2 and 3 are not exclusive.

Example 3.6 Assume the semigroup Q with the underlying set $\{0, a, b\}$. We define the operation in Q as follows: For every $p \in Q$, we set $0p = p0 = 0$. For $p, q \in Q \setminus \{0\}$, the product pq yields q. Every ideal of Q has to contain 0. Assume that an ideal J contains one of the elements a or b. Then, the product JQ yields Q. Consequently, the set $\{0\}$ is the only proper ideal of Q.

We see that Q and $\{0\}$ fulfill assertion two. However, Q and $\{0\}$ also fulfill assertion three by $U = \{0, a\}$ and $V = \{0, b\}$. □

3.4 A Classification

We work out a suitable classification of all non-empty finite semigroups. We use ideal theory to distinguish several classes of non-empty finite semigroups.

Proposition 3.7 Every non-empty finite semigroup Q fulfills one of the following assertions:

(A) Q has not any proper left ideal.

(B) Q has two proper left ideals U, V such that $U \cup V = Q$ and $U \cap V$ is an ideal of Q.

(C) Q has an ideal J such that $Q \setminus J$ yields a singleton $\{r\}$ with $r^2 \in J$.

(D) Q has a proper ideal J and a subsemigroup H such that $J \cap H = \emptyset$ and $J \cup H = Q$. □

Proof: Assume Q does not have a proper ideal. If Q does not have a proper left ideal, it fulfills assertion (A). If Q has a proper left ideal, then Q fulfills assertion (B) by Lemma 3.4, because the empty set is an ideal.

Now, assume Q has a proper ideal. Then, by Lemma 3.5 there is some proper ideal J of Q such that J fulfills one of the assertions (C), (D) or (B). □

Assume a non-empty finite semigroup Q. Every proper ideal is also a proper left ideal. Thus, if Q fulfills one of the assertions (B), (C), or (D), then Q cannot satisfy assertion (A). However, the assertions (B), (C), and (D) are not exclusive.

Example 3.8 Consider the semigroup Q_2 with the underlying set $\{0', \hat{0}, b', \hat{b}, c'\}$ and the operation given by the following table:

	$0'$	$\hat{0}$	b'	$\hat{b}$	c'
$0'$	$0'$	$\hat{0}$	$0'$	$\hat{0}$	$0'$
$\hat{0}$	$0'$	$\hat{0}$	$0'$	$\hat{0}$	$0'$
b'	$0'$	$\hat{0}$	b'	$\hat{b}$	$0'$
$\hat{b}$	$0'$	$\hat{0}$	b'	$\hat{b}$	$0'$
c'	$0'$	$\hat{0}$	$0'$	$\hat{0}$	$0'$

The operation can be understood intuitively: For $n > 1$ and $p_1, \ldots, p_n \in Q_2$, we can calculate the product $p_1 \ldots p_n$ as follows: If a c', $0'$ or $\hat{0}$ occurs among $p_1, \ldots, p_n$, the result has a 0. Otherwise, the result has a b. The index of $p_1 \ldots p_n$ (i.e. $'$ or $\hat{}$) is the index of p_n. The semigroup Q_2 fulfills the assertions (B), (C), and (D).

(B) The semigroup Q_2 has proper ideals such that we cannot apply Lemma 3.4. Nevertheless, we can split Q_2 into two disjoint left ideals by $U = \{0', b', c'\}$ and $V = \{\hat{0}, \hat{b}\}$. We could also choose proper left ideals which are not disjoint, e.g., by $U = \{0', \hat{0}, b', c'\}$ and $V = \{0', \hat{0}, \hat{b}\}$.

(C) We simply set $J = \{0', \hat{0}, b', \hat{b}\}$.

(D) Q_2 satisfies this assertion by $J = \{0', \hat{0}, c'\}$ and $H = \{b', \hat{b}\}$. □

4 Product Automata

In this section, we deal with a special kind of automata. We adapt the notion of automata from Definition 2.3. We use ideas from the proof of Theorem 2.7 (cf. [1, 10]). In the first subsection, we get familiar with product automata. In the second subsection, we examine connections to ideal theory.

Throughout this section, we assume two disjoint independence alphabets (Σ_1, I_1) and (Σ_2, I_2). We abbreviate the trace monoids $\mathbb{M}(\Sigma_1, I_1)$ and $\mathbb{M}(\Sigma_2, I_2)$ by $\mathbb{M}_1$ and $\mathbb{M}_2$, respectively.

4.1 Definitions

We define product automata.

Definition 4.1 Assume two disjoint trace monoids $\mathbb{M}_1$ and $\mathbb{M}_2$. A product automaton $\mathcal{A}$ over $\binom{\mathbb{M}_1}{\mathbb{M}_2}$ is a quintuple $[P, R, g, h, F]$, where

- P and R are non-empty finite semigroups,
- g and h are surjective homomorphisms $g : \mathbb{M}_1 \to P,\ h : \mathbb{M}_2 \to R$,
- F is a subset of $\binom{P}{R}$. □

We can regard every product automaton $[P, R, g, h, F]$ as an $\binom{\mathbb{M}_1}{\mathbb{M}_2}$-automaton $\left[\binom{P}{R}, \binom{g}{h}, F\right]$ by Definition 2.3. A product automaton $\mathcal{A}$ defines a recognizable language by $L(\mathcal{A}) = \binom{g}{h}^{-1}(F)$. This means that a trace $t \in \binom{\mathbb{M}_1}{\mathbb{M}_2}$ belongs to $L(\mathcal{A})$ iff we obtain a pair in F when we apply g and h on the first and second compound of t, respectively.

Let us assume that $L(\mathcal{A})$ is closed under concatenation. Then, F is a subsemigroup of $\binom{P}{R}$. Similarly, $\Pi_1(F)$ and $\Pi_2(F)$ are subsemigroups of P and R, respectively.

We are going to use product automata to prove assertions on recognizable languages in $\binom{\mathbb{M}_1}{\mathbb{M}_2}$. Therefore, we have to show that every recognizable language T in $\binom{\mathbb{M}_1}{\mathbb{M}_2}$ is the language of some product automaton.

Lemma 4.2 Assume two disjoint trace monoids $\mathbb{M}_1$ and $\mathbb{M}_2$, and a recognizable language T in $\binom{\mathbb{M}_1}{\mathbb{M}_2}$. There is a product automaton for T. □

Proof: By Theorem 2.7, there is some $n \in \mathbb{N}$ and recognizable languages $T_1, \ldots, T_n \subseteq \mathbb{M}_1$ and $T'_1, \ldots, T'_n \subseteq \mathbb{M}_2$ such that

$$T = \binom{T_1}{T'_1} \cup \ldots \cup \binom{T_n}{T'_n}$$

For $i = 1, \ldots, n$, let $[P_i, g_i, F_i]$ (resp. $[R_i, h_i, F'_i]$) be an automaton for T_i (resp. T'_i). We can freely assume $P_i = P_j$, $g_i = g_j$, $R_i = R_j$, and $h_i = h_j$ for any $1 \leq i \leq j \leq n$. Then, T is the language of the product automaton $[P_1, R_1, g_1, h_1, F]$ with $F = \binom{F_1}{F'_1} \cup \ldots \cup \binom{F_n}{F'_n}$. □

4.2 Connections to Subsemigroups and Ideals

We examine connections between product automata and ideal theory. Assume a recognizable language T in $\binom{\mathbb{M}_1}{\mathbb{M}_2}$ which is closed under concatenation. Assume further a product automaton $\mathcal{A} = [P, R, g, h, F]$ for T. Let us denote $\Pi_2(F)$ by Q. Then, Q is a subsemigroup of R. We can easily verify that $Q = h \circ \Pi_2(T) = \Pi_2 \circ \binom{g}{h}(T)$. Assume some subset W of Q. We define a language T_W by

$$T_W = \{\ t \in T \mid h \circ \Pi_2(t) \in W\ \}$$

The language T_W is obviously a subset of T. Some trace t in $\binom{\mathbb{M}_1}{\mathbb{M}_2}$ belongs to T_W iff we have $\binom{g}{h}(t) \in F \cap \binom{P}{W}$.

Proposition 4.3 Assume some non-empty, concatenation closed language T in $\binom{\mathbb{M}_1}{\mathbb{M}_2}$. Assume a product automaton $\mathcal{A} = [P, R, g, h, F]$ for T. Let Q denote the semigroup $\Pi_2(F)$.

1. For $W \subseteq Q$, the product automaton $\mathcal{A}_W = [P, R, g, h, F \cap \binom{P}{W}]$ defines T_W. Therefore, T_W is recognizable.
2. For every non-empty $W \subseteq Q$, the language T_W is non-empty.
3. For every subsemigroup $H \subseteq Q$, the language T_H is concatenation closed.
4. For every left ideal $U \subseteq Q$, the language T_U is a left ideal of T.
5. For every ideal $J \subseteq Q$, the language T_J is an ideal of T. $\square$

Proof:

1. We can straightforwardly verify that $\mathcal{A}_W$ is a product automaton, because $\mathcal{A}$ is a product automaton. For $t \in T_W$, we have $\binom{g}{h}(t) \in F$ and $\binom{g}{h}(t) \in \binom{P}{W}$. Thus, we have $\binom{g}{h}(t) \in F \cap \binom{P}{W}$. Hence, $T_W \subseteq L(\mathcal{A}_W)$.

   Conversely, let $t \in L(\mathcal{A}_W)$. Then, we have $\binom{g}{h}(t) \in F$ and $\binom{g}{h}(t) \in \binom{P}{W}$. Hence, $t \in T$ and $h \circ \Pi_2(t) \in W$, i.e., t belongs to T_W. Consequently, $L(\mathcal{A}_W) \subseteq T_W$.
2. Because W is non-empty, we can assume some r in W. This r belongs to Q, i.e., there is some p in P such that the pair $\binom{p}{r} \in F$. Because the homomorphism $\binom{g}{h}$ is a surjection, the set $\binom{g}{h}^{-1}\binom{p}{r}$ is non-empty. Further, we have $\binom{g}{h}^{-1}\binom{p}{r} \subseteq T_W$.
3. Note that $T_W = \Pi_2^{-1} \circ h^{-1}(W) \cap T$. Since W is a subsemigroup of Q, its preimage $\Pi_2^{-1} \circ h^{-1}(W)$ is a subsemigroup of $\binom{\mathbb{M}_1}{\mathbb{M}_2}$. Then, T_W is a subsemigroup because it is the intersection of two subsemigroups.
4. Let f be the restriction of $h \circ \Pi_2$ to T. Then, f is a surjective homomorphism from T to Q and $T_U = f^{-1}(U)$. Since U is a left ideal of Q, so is its preimage T_U under f.
5. As the previous point. $\square$

5 Proof of Theorem 2.16

In this section, we prove Theorem 2.16. We take over the notions (Σ_1, I_1), (Σ_2, I_2), $\mathbb{M}_1$ and $\mathbb{M}_2$ from the previous section. We abbreviate $\mathbb{M}_1 \setminus \{\lambda\}$ and $\mathbb{M}_2 \setminus \{\lambda\}$ by $\mathbb{M}_1^+$ and $\mathbb{M}_2^+$, respectively. The traces in $\binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$ are exactly the traces in $\binom{\mathbb{M}_1}{\mathbb{M}_2}$ which contain at least one letter in Σ_1 and at least one letter in Σ_2.

Theorem 2.16 Assume two disjoint independence alphabets (Σ_1, I_1) and (Σ_2, I_2). Assume some concatenation closed, recognizable language T in $\binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$.
Then, the set of generators of T has the FPP. $\square$

Theorem 2.16 is obviously true if the language T is empty. Thus, we just need to prove it for non-empty languages T. The general structure of the proof is the following: Just assume a non-empty, concatenation closed, recognizable language T in $\binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$. By Lemma 4.2, there is a product automaton $\mathcal{A} = [P, R, g, h, F]$ for T. We denote $\Pi_2(F)$ by Q. Because T is non-empty, Q is non-empty. We use Proposition 3.7 on Q. Consequently, the proof of Theorem 2.16 consists of four parts. In the first subsection, we deal with the case that Q does not contain proper left ideals. After that, we deal with the cases that Q fulfills one of the assertions (B), (C), or (D) in Proposition 3.7. We will do this by an induction on the number of elements of Q.

In the last subsection, we summarize the results to prove Theorem 2.16.

5.1 Q does not have proper left ideals

In this subsection, we prove the following special case of Theorem 2.16.

Proposition 5.1 Assume a non-empty, concatenation closed language T in $\binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$ which is recognized by a product automaton $[P, R, g, h, F]$, such that the semigroup $\Pi_2(F)$ does not have proper left ideals.
Then, $\mathrm{Gen}(T)$ has the FPP. Moreover, we have $T = \mathrm{Gen}(T)^{1,\ldots,|\Pi_2(F)|+1}$. □

I introduce some notions especially for the proof of this proposition. We assume a language T as in Proposition 5.1.

Definition 5.2 Assume some traces t, t_1, s_1 in T. We call the pair (t_1, s_1) a *most oblique cut* of t iff $t = t_1 s_1$ and for every traces $t_1', s_1' \in T$ with $t = t_1' s_1'$ we have either

- $|\Pi_1(t_1')| > |\Pi_1(t_1)|$ or
- $|\Pi_1(t_1')| = |\Pi_1(t_1)|$ and $|\Pi_2(t_1')| \leq |\Pi_2(t_1)|$. □

Intuitively, we can understand the definition as follows. We try to factorize t in T into two traces t_1 and s_1 of T. We try to do this in a way that the first compound of t_1 is small, but, the second compound of t_1 is big.

If the trace t is not a generator of T, then there is at least one most oblique cut of t. A most oblique cut cannot exist if the trace t is a generator of T.

Lemma 5.3 Assume some traces t, t_1, s_1 in T such that (t_1, s_1) is a most oblique cut of t. Then, the trace t_1 is a generator of T. □

Proof: We prove the lemma by contradiction. Just assume that t_1 is not a generator of T. Then, there are two traces t_{1a} and t_{1b} in T such that $t = t_{1a} t_{1b}$. We can factorize t into t_{1a} and $t_{1b} s_1$. The traces t_{1a} and $t_{1b} s_1$ belong to T. Further, $\Pi_1(t_{1a})$ contains properly less letters than $\Pi_1(t_1)$, since $\Pi_i(t_1 b) \neq \lambda$. This contradicts that (t_1, s_1) is a most oblique cut. □

We can factorize some trace t in T into generators by successive most oblique cuts. We factorize t into a generator t_1 and a trace s_1 in T. Then, we factorize s_1 by a most oblique cut and so on, until a most oblique cut yields two generators. This iterative factorization terminates, because "the remaining part of t" becomes properly shorter in every most oblique cut.

Proof of Proposition 5.1: Assume some trace t in T. We denote by Q the semigroup $\Pi_2(F)$. We show that a factorization of t by successive most oblique cuts yields a factorization of t into at most $|Q| + 1$ generators of T.

We factorize t into generators of T by successive most oblique cuts. We obtain a natural number n and generators $t_1, \ldots, t_n$ of T such that $t_1 \ldots t_n$ yields t. For every i in $1, \ldots, n-1$, the pair $(t_i, t_{i+1} \ldots t_n)$ is a most oblique cut of the trace $t_i \ldots t_n$.

We introduce two notions for lucidity. For every i in $1, \ldots, n$, we define $u_i = \Pi_1(t_i)$ and $v_i = \Pi_2(t_i)$, i.e., we have $t_i = \binom{u_i}{v_i}$. For every i in $1, \ldots, n$, we have $h(v_i) \in Q$, because the traces $t_1, \ldots, t_n$ belong to T.

We show by a contradiction that $n \leq |Q| + 1$. Assume $n > |Q| + 1$.

By $h(v_{i+1} \ldots v_n) = h(v_{i+1}) \ldots h(v_n) \in Q$ for $1 \leq i < n$ and $n-1 > |Q|$, we get the existence of $1 \leq i < j < n$ such that $h(v_{i+1} \ldots v_n) = h(v_{j+1} \ldots v_n)$.

Then, $h(v_i) \cdot_Q h(v_{i+1} \ldots v_n) = h(v_i \ldots v_n) = h(v_i \ldots v_j) \cdot_Q h(v_{j+1} \ldots v_n)$. Since Q does not have proper left ideals, we can apply Lemma 3.2 and get $h(v_i) = h(v_i \ldots v_j)$.

By $t_i \in T$, we have $\binom{g}{h}\binom{u_i}{v_i} \in F$. Because of $h(v_i) = h(v_i \dots v_j)$, we get $\binom{g}{h}\binom{u_i}{v_i \dots v_j} \in F$, and thus, $\binom{u_i}{v_i \dots v_j} \in T$. Similarly, $t_{i+1} \dots t_n \in T$ implies $\binom{g}{h}\binom{u_{i+1} \dots u_n}{v_{i+1} \dots v_n} \in F$. By $h(v_{i+1} \dots v_n) = h(v_{j+1} \dots v_n)$, we have $\binom{g}{h}\binom{u_{i+1} \dots u_n}{v_{j+1} \dots v_n} \in F$, and therefore, $\binom{u_{i+1} \dots u_n}{v_{j+1} \dots v_n} \in T$.

Thus, $\binom{u_i}{v_i \dots v_j}$ and $\binom{u_{i+1} \dots u_n}{v_{j+1} \dots v_n}$ are a factorization of $t_i \dots t_n$ into two traces from T. Since $(t_i, t_{i+1} \dots t_n)$ is a most oblique cut of t and $\Pi_1\binom{u_i}{v_i \dots v_j} = \Pi_1(t_i)$, we obtain $|\Pi_2\binom{u_i}{v_i \dots v_j}| \leq |\Pi_2(t_i)|$. Hence, $|v_i \dots v_j| \leq |v_i|$. Because v_i is a prefix of $v_i \dots v_j$, we have $|v_i \dots v_j| = |v_i|$. Consequently, $v_{i+1} \dots v_j = \lambda$. This is a contradiction, because every trace in T contains at least one letter from Σ_2.

Finally, our assumption $n > |Q| + 1$ lead to a contradiction. Hence, we have $n \leq |Q| + 1$. □

The method of most oblique cuts is a very suitable method to prove Theorem 2.16 in the case that the semigroup Q does not have proper left ideals. Let us consider an example.

Example 5.4 Assume the free monoids over singletons $\mathbb{M}_1 = \{a\}^*$ and $\mathbb{M}_2 = \{b\}^*$. Consider the language $T = \left\{\binom{a}{b}\right\} \cup \left\{\binom{a^n}{b^m} \,\middle|\, n \geq 2, m \geq 2\right\}$ in $\binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$. The language T fulfills every presumption of Theorem 2.16. However, we cannot prove that Gen(T) has the FPP by factorizations with most oblique cuts. For every $n \geq 1$, the application of successive most oblique cuts factorizes the trace $\binom{a^n}{b^n}$ into $\binom{a}{b} \dots \binom{a}{b}$, i.e., we obtain n generators. Hence, the number of generators which we obtain by successive most oblique cuts is unlimited. □

5.2 Q fulfills assertion (C)

We prove the remaining cases of Theorem 2.16 by an induction on the number of elements in Q. In the case that Q is a singleton, we already know by Proposition 5.1 that Theorem 2.16 is true for T, because the singleton semigroup does not have proper left ideals.

Assume some natural number $n > 1$. By induction, we presume that Theorem 2.16 is true for languages T' if there is a product automaton $[P', R', g', h', F']$ for T' such that the number of elements of $\Pi_2(F')$ (i.e. Q') is properly smaller than n. Then, we show that Theorem 2.16 is true if Q has n elements and Q fulfills one of the assertions (B), (C), or (D) in Proposition 3.7. We do this by a decomposition of Q and T into subsemigroups, left ideals and ideals.

We perform the first induction step. We start with the case that Q fulfills assertion (C) in Proposition 3.7 because this is the most simple one.

Proposition 5.5 Let $n > 1$. Assume that Theorem 2.16 holds for every non-empty, concatenation closed language T' in $\binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$ which is recognized by a product automaton $[P', R', g', h', F']$ with $|\Pi_2(F')| < n$. Let $[P, R, g, h, F]$ be a product automaton for a language T such that

- T is a non-empty, concatenation closed language in $\binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$,
- $|\Pi_2(F)| = n$, and,
- there is an $r \in \Pi_2(F)$ such that $r^2 \neq r$ and $\Pi_2(F) \setminus \{r\}$ is an ideal.

Then, Gen(T) has the FPP. □

Proof: For simplicity, let $Q = \Pi_2(F)$ and $J = \Pi_2(F) \setminus \{r\}$. We use Proposition 4.3 to examine the language $T_J = \{ t \in T \mid h \circ \Pi_2(t) \in J \}$. The traces in $T \setminus T_J$ are exactly these traces t in T with $h \circ \Pi_2(t) = r$.

The ideal J of Q is a subsemigroup of Q, because every ideal is a subsemigroup. By assertion three of Proposition 4.3, the language T_J is concatenation closed.

By assertion one of Proposition 4.3, we know that T_J is recognizable. Moreover, the product automaton $\mathcal{A}_J = [P, R, g, h, F \cap \binom{P}{J}]$ defines T_J. We see that $\Pi_2(F \cap \binom{P}{J})$ yields J. We have $|\Pi_2(F \cap \binom{P}{J})| = |Q| - 1$, i.e., we have $|\Pi_2(F \cap \binom{P}{J})| < n$. By the inductive hypothesis, we have an $l_J \in \mathbb{N}$ such that $T_J = \mathrm{Gen}(T_J)^{1,\ldots,l_J}$.

By assertion five of Proposition 4.3, we know that T_J is an ideal of T. Further, we can show that for every $t_1, t_2 \in T$, the concatenation t_1t_2 belongs to T_J. If one of the traces t_1 and t_2 belongs to T_J, then the trace t_1t_2 belongs to T_J because T_J is an ideal. If both t_1 and t_2 do not belong to T_J, then we have $h \circ \Pi_2(t_1) = h \circ \Pi_2(t_2) = r$. We have $h \circ \Pi_2(t_1t_2) = h \circ \Pi_2(t_1) \cdot_Q h \circ \Pi_2(t_2) = r^2 \in J$. Hence, the trace t_1t_2 belongs to T_J, i.e., $TT \subseteq T_J$.

Let $l = 3l_J$. We show $T = \mathrm{Gen}(T)^{1,\ldots,l}$. For this, let $t \in T$.

Case 1: The trace t does not belong to T_J.

Because $TT \subseteq T_J$, we have $t \notin TT$. Hence, t is a generator of T. Thus, $t \in \mathrm{Gen}(T)^{1,\ldots,l}$.

Case 2: The trace t is a generator of T_J.

Assume we can factorize t into more than three generators of T. Then, we can also factorize t into exactly four traces of T. Thus, there are traces $t_1, \ldots, t_4 \in T$ such that $t_1 \ldots t_4 = t$. We have $t_1t_2 \in T_J$ and $t_3t_4 \in T_J$. Consequently, t is not a generator of T_J. This contradicts the case presumption. Hence, $t \in \mathrm{Gen}(T)^{1,\ldots,3}$.

Case 3: The trace t belongs to T_J.

Because of $T_J = \mathrm{Gen}(T_J)^{1,\ldots,l_J}$ we have a $k \leq l_J$ and $t_1, \ldots, t_k \in T_J$ such that $t_1 \ldots t_k = t$. By case 2, we can factorize every trace among $t_1, \ldots, t_k$ into three or less generators of T. Hence, we can factorize t into $3l_J = l$ or less generators of T, i.e., $t \in \mathrm{Gen}(T)^{1,\ldots,l}$.

We have $T = \mathrm{Gen}(T)^{1,\ldots,l}$, i.e., the set $\mathrm{Gen}(T)$ has the FPP. □

5.3 Q fulfills assertion (D)

We deal with the case that we can split Q into a proper ideal and a subsemigroup.

Proposition 5.6 Let $n > 1$. Assume that Theorem 2.16 holds for every non-empty, concatenation closed language T' in $\binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$ which is recognized by a product automaton $[P', R', g', h', F']$ with $|\Pi_2(F')| < n$. Let $[P, R, g, h, F]$ be a product automaton for a language T such that

- T is a non-empty, concatenation closed language in $\binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$,
- $|\Pi_2(F)| = n$, and,
- there are a proper ideal J and a subsemigroup H in $\Pi_2(F)$ with $J \cup H = \Pi_2(F)$ and $J \cap H = \emptyset$

Then, $\mathrm{Gen}(T)$ has the FPP. □

Proof: We denote by Q the semigroup $\Pi_2(F)$. We examine the languages T_J and T_H. They are non-empty subsets of T. They are disjoint and their union yields T. Both T_J and T_H are recognizable and concatenation closed. Further, the language T_J is an ideal of T. There are two natural numbers l_J and l_H such that $T_J = \mathrm{Gen}(T_J)^{1,\ldots,l_J}$ and $T_H = \mathrm{Gen}(T_H)^{1,\ldots,l_H}$.

Let $l = 2l_Hl_J + l_J$. We show $T = \mathrm{Gen}(T)^{1,\ldots,l}$. Assume some trace t in T.

Case 1: t is a generator of T_H.

Assume that there are $t_1, t_2 \in T$ such that $t_1t_2 = t$. If one of the traces t_1 or t_2 belongs to the ideal T_J, then t belongs to T_J. Hence, both t_1 and t_2 belong to T_H. This contradicts that t is a generator of T_H. Thus, t is a generator of T.

Case 2: t is a trace of T_H.

We can factorize t into l_H or less generators of T_H. By case 1, such a factorization is a factorization of t into generators of T. Thus, $t \in \mathrm{Gen}(T)^{1,\ldots,l_H}$.

Case 3: t is a generator of T_J.

The trace t is not necessarily a generator of T. We examine factorizations of t as follows. Assume three traces $t_1, t_2, t_3 \in T \cup \{\lambda\}$ such that

$$t_1t_2t_3 = t, \qquad t_1 \in T_H \cup \{\lambda\}, \qquad t_2 \in T_J \cup \{\lambda\}, \qquad \text{and} \qquad t_3 \in T_H \cup \{\lambda\}.$$

There are traces t_1, t_2, t_3 which fulfill these conditions: namely $t_1 = \lambda$, $t_2 = t$, $t_3 = \lambda$. We choose a triple t_1, t_2, t_3 such that the number of letters of t_2 is minimal.

We can apply case 2 to the traces t_1 and t_3. The traces t_1 and t_3 are either λ, or they belong to T_H such that we can factorize each of t_1 and t_3 into l_H or less generators of T.

We deal with t_2. Assume t_2 is not the empty trace. By a contradiction, we can show that t_2 is a generator of T. Just assume $t_2', t_2'' \in T$ with $t_2't_2'' = t_2$. We distinguish four cases:

- $t_2' \in T_J$ and $t_2'' \in T_J$
  The language T_J is an ideal of T such that $t_1t_2', t_2''t_3 \in T_J$. Thus, t_1t_2' and $t_2''t_3$ form a factorization of t into two traces of T_J. This contradicts that t is a generator of T_J.
- $t_2' \in T_H$ and $t_2'' \in T_J$
  Then $t_1t_2' \in T_H$, because T_H is a subsemigroup. The trace t_2'' contains properly less letters than t_2. Consequently, the traces t_1t_2', t_2'', and t_3 contradict the choice of t_1, t_2, and t_3, above.
- $t_2' \in T_J$ and $t_2'' \in T_H$
  Similar to the previous case, the traces t_1, t_2', and $t_2''t_3$ contradict the choice of the traces t_1, t_2, and t_3.
- $t_2' \in T_H$ and $t_2'' \in T_H$
  The traces t_1t_2', λ, and $t_2''t_3$ contradict the choice of t_1, t_2, and t_3.

Thus, the trace t_2 is either λ or a generator of T. We can factorize t_1 and t_3 into l_H or less generators of T. Therefore, $t \in \mathrm{Gen}(T)^{1,\ldots,2l_H+1}$.

Case 4: t is a trace in T_J.

We can factorize t into l_J or less generators of T_J. By case 3, we can factorize every generator of T_J into $2l_H + 1$ generators of T. Consequently, $t \in \mathrm{Gen}(T)^{1,\ldots,2l_Hl_J+l_J}$.

We have $T = \mathrm{Gen}(T)^{1,\ldots,2l_Hl_J+l_J}$, i.e., the set $\mathrm{Gen}(T)$ has the FPP. □

5.4 Q fulfills assertion (B)

Just one case remains. The semigroup Q has two left ideals U and V such that their union yields Q and their intersection yields some ideal of Q. This case is the most involved one.

Proposition 5.7 Let $n > 1$. Assume that Theorem 2.16 holds for every non-empty, concatenation closed language T' in $\binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$ which is recognized by a product automaton $[P', R', g', h', F']$ with $|\Pi_2(F')| < n$. Let $[P, R, g, h, F]$ be a product automaton for a language T such that

- T is a non-empty, concatenation closed language in $\binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$,
- $|\Pi_2(F)| = n$, and,

- there are two proper left ideals U and V such that $U \cup V = \Pi_2(F)$ and $U \cap V$ yields an ideal of $\Pi_2(F)$.

Then, $\mathrm{Gen}(T)$ has the FPP. $\square$

Proof: As in the previous proofs, we set $Q = \Pi_2(F)$. For lucidity, we set $J = U \cap V$.

As in the previous proofs, the languages T_U and T_V are non-empty, recognizable and concatenation closed subsets of T. Further, T_U and T_V are left ideals of T, and T_J is an ideal of T. We have two natural numbers l_U and l_V such that $T_U = \mathrm{Gen}(T_U)^{1,\ldots,l_U}$ and $T_V = \mathrm{Gen}(T_V)^{1,\ldots,l_V}$. Provided that J is non-empty, there is some natural number l_J such that $T_J = \mathrm{Gen}(T_J)^{1,\ldots,l_J}$. If $J = \emptyset$, then T_J is also empty.

We have $U \cup V = Q$ and $U \cap V = J$. For every $t \in T$, we have $h \circ \Pi_2(t) \in U$ or $h \circ \Pi_2(t) \in V$. Thus, $T_U \cup T_V = T$. Further, for every $t \in T$, we have $h \circ \Pi_2(t) \in J$ iff $h \circ \Pi_2(t) \in U$ and $h \circ \Pi_2(t) \in V$. Hence, we have $T_U \cap T_V = T_J$.

Let $l = 3\, l_J(l_U l_V + \max(l_U, l_V)) + l_J$. We show $T = \mathrm{Gen}(T)^{1,\ldots,l}$. Assume some trace t in T.

Case 1: t is a generator of T_U. Further, t does not belong to T_J.

Then, t cannot belong to T_V. Furthermore, if we factorize t into some traces in T, not any factor does belong to the ideal T_J. Otherwise, t would belong to T_J. Consequently, if we factorize t into some traces in T, the factors either belong to T_U or T_V, but they do not belong to $T_U \cap T_V$.

The trace t is not necessarily a generator of T. In the case that t is a generator of T, we are done. So assume that t is not a generator of T. There is a trace $x \in T$ and a generator y of T with $xy = t$.

Assume y belongs to the left ideal T_V. Then, $xy = t$ also belongs to T_V. This is a contradiction. Hence, $y \in T_U$. Assume x also belongs to T_U. Then, $xy = t$ is not a generator of T_U. Hence, we have $x \in T_V$ and $y \in T_U$.

We factorize the trace x into l_V or less generators of T_V. There are a $k \leq l_V$ and generators $x_1, \ldots, x_k$ of T_V such that $x_1 \ldots x_k = x$.

We show by a contradiction that the traces $x_1, \ldots, x_k$ are generators of T. Just assume some natural number i with $1 \leq i \leq k$ such that x_i can be factorized into two traces x_i' and x_i'' in T. Assume that x_i'' belongs to T_U. Then, x_i belongs to T_U, which is a contradiction. Hence, $x_i'' \in T_V$. Now, assume that x_i' belongs to T_V. Then, x_i is not a generator of T_V. Thus, we have $x_i' \in T_U$ and $x_i'' \in T_V$.

However, this yields a contradiction: We factorize t into the traces $x_1 \ldots x_{i-1} x_i'$ and $x_i'' x_{i+1} \ldots x_k y$. Both factors belong to T_U, because x_i' and y belong to the left ideal T_U. Hence, we can factorize t into two traces from T_U which is a contradiction.

The assumption that some trace among $x_1, \ldots, x_k$ is not a generator of T yields a contradiction. Thus, we have by $x_1, \ldots, x_k, y$ a factorization of t into generators of T. Hence, $t \in \mathrm{Gen}(T)^{1,\ldots,l_V+1}$.

Case 2: t is a trace in $T_U \setminus T_J$.

There are a $k \leq l_U$ and generators $t_1, \ldots, t_k$ of T_U such that $t_1 \ldots t_k = t$. The generators $t_1, \ldots, t_k$ cannot belong to T_J. By case 1, we have $t_1, \ldots, t_k \in \mathrm{Gen}(T)^{1,\ldots,l_V+1}$. Because $k \leq l_U$, we have $t \in \mathrm{Gen}(T)^{1,\ldots,l_U l_V + l_U}$.

Case 3: t is a trace in $T \setminus T_J$.

We can deal with the traces in $T_V \setminus T_J$ as we dealt with the traces in $T_U \setminus T_J$. We obtain $(T_V \setminus T_J) \subseteq \mathrm{Gen}(T)^{1,\ldots,l_U l_V + l_V}$.

Hence, we have $(T \setminus T_J) \subseteq \mathrm{Gen}(T)^{1,\ldots,l_U l_V + \max(l_U, l_V)}$.

Case 4: t is a generator of T_J.

We use the triple factorization method from case 3 in the proof of Proposition 5.6. Assume three traces t_1, t_2, t_3 in $T \cup \{\lambda\}$ such that we have

$$t_1 t_2 t_3 = t, \qquad t_1, t_3 \in \Big(T \setminus T_J\Big) \cup \{\lambda\}, \qquad \text{and} \qquad t_2 \in T_J \cup \{\lambda\}.$$

There are traces t_1, t_2, t_3 which fulfill these conditions, e.g., λ, t, λ, respectively. We choose a triple t_1, t_2, t_3 which fulfills the above conditions such that the number of letters of t_2 is minimal. We can apply case 3 on the traces t_1 and t_3 if they are not empty.

We deal with t_2. If t_2 is the empty trace, we have $t = t_1 t_3$. Then, we can factorize t into $2\Big(l_U l_V + \max(l_U, l_V)\Big)$ or less generators of T. If t_2 is a generator of T, then we need one more generator to factorize t into generators of T.

We deal with the case that t_2 is not a generator of T. Then, we can factorize t_2 into a generator t_2' of T and some trace t_2'' in T. Assume that $t_2'' \in T_J$. We distinguish two cases depending on whether we have $t_1 t_2' \in T_J$ or not. If we have $t_1 t_2' \in T_J$, then can factorize t into the traces $t_1 t_2'$ and $t_2'' t_3$ in T_J which contradicts that t is a generator of T_J. If we have $t_1 t_2' \in T \setminus T_J$, then we have a factorization of t into $t_1 t_2'$, t_2'' and t_3. This contradicts the choice of t_1, t_2, t_3, above. Consequently, $t_2'' \notin T_J$.

To sum up, we have a factorization of t into t_1, t_2', t_2'' and t_3. The trace t_2' is a generator of T. We can apply case 3 on t_1, t_2'' and t_3.

Thus, $t \in \mathrm{Gen}(T)^{1,\ldots,3\left(l_U l_V + \max(l_U, l_V)\right)+1}$.

Case 5: t is a trace in T_J.

We can factorize t into l_J or less generators of T_J and apply case 4 to every generator.

Consequently, $t \in \mathrm{Gen}(T)^{1,\ldots,3 l_J\left(l_U l_V + \max(l_U, l_V)\right)+l_J}$.

We have $T = \mathrm{Gen}(T)^{1,\ldots,3 l_J\left(l_U l_V + \max(l_U, l_V)\right)+l_J}$, i.e., the set $\mathrm{Gen}(T)$ has the FPP. $\square$

5.5 Completion of the Proof

Proof of Theorem 2.16: The theorem is obviously true if T is the empty set. As a conclusion of Proposition 5.1, Theorem 2.16 holds for every non-empty, concatenation closed language $T \subseteq \binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$, if there is a product automaton $[P, R, g, h, F]$ for T such that $\Pi_2(F)$ is a singleton.

Assume some natural number $n > 1$. Assume that Theorem 2.16 is true for every concatenation closed language $T' \subseteq \binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$, if there is a product automaton $[P', R', g', h', F']$ for T' with $|\Pi_2(F')| < n$.

Now, let T be a concatenation closed language in $\binom{\mathbb{M}_1^+}{\mathbb{M}_2^+}$ defined by a product automaton $[P, R, g, h, F]$ with $\Pi_2(F) = n$. Then, by Proposition 3.7, the semigroup $\Pi_2(F)$ satisfies one of the assertions (A), (B), (C) or (D) such that we can apply one of the Propositions 5.1, 5.7, 5.5, or 5.6, respectively. $\square$

6 Conclusions and Future Steps

By proving Theorem 2.15, we corrected an error in [27]. Provided that this report does not contain further errors, its main result is true, i.e., the trace monoids with a decidable star problem are exactly the trace monoids with a decidable FPP.

Opposed to the previously known connections between the star problem and the FPP in Theorem 2.9 and 2.10, we have by Theorem 2.15 a connection between the star problem and the FPP in one and the same trace monoid.

It raises the following decision problem which is a special case of the star problem: Assume two disjoint trace monoids $\mathbb{M}(\Sigma_1, I_1)$ and $\mathbb{M}(\Sigma_2, I_2)$ and a language L as in Theorem 2.15. Can we decide whether L^* is recognizable, i.e., whether L has the FPP, provided that both the star problem and the FPP are decidable in both $\mathbb{M}(\Sigma_1, I_1)$ and $\mathbb{M}(\Sigma_2, I_2)$?

7 Acknowledgments

I acknowledge the helpful discussions with Manfred Droste and Dietrich Kuske at the Institute of Algebra. I especially acknowledge Manfred Droste's lecture "Algebraic Theory of Automata" in which I came in tough with ideal theory and the helpful teaching book [15]. I thank to Dietrich Kuske for reading a preliminary version of this paper and giving many useful hints. I thank to Gwénaël Richomme for bringing my attention to this particular problem.

References

[1] J. Berstel. *Transductions and Context-Free Languages*. B. G. Teubner, Stuttgart, 1979.

[2] A. Bertoni, M. Goldwurm, G. Mauri, and N. Sabadini. Counting techniques for inclusion, equivalence and membership problems. Chapter 5 in [9], pages 131-163.

[3] P. Cartier and D. Foata. *Problèmes combinatoires de commutation et réarrangements*, volume 85 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1969.

[4] M. Clerbout and M. Latteux. Semi-commutations. *Information and Computation*, 73:59–74, 1987.

[5] A. H. Clifford and G. B. Preston. *The Algebraic Theory of Semigroups*, volume I. Amer. Math. Soc., Providence RI, 1961.

[6] B. Courcelle. Basic notions of universal algebra for language theory and graph grammars. *Theoretical Computer Science*, 163:1–54, 1996.

[7] V. Diekert. *Combinatorics on Traces*, volume 454 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin Heidelberg New York, 1990.

[8] V. Diekert and Y. Métivier. Partial commutation and traces. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, Vol. 3, Beyond Words*, pages 457–534. Springer-Verlag, Berlin Heidelberg, 1997.

[9] V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, Singapore, 1995.

[10] S. Eilenberg. *Automata, Languages, and Machines,* Volume A. Academic Press, New York, 1974.

[11] M. Fliess. Matrices de hankel. *J. Math. Pures et Appl.*, 53:197–224, 1974.

[12] P. Gastin, E. Ochmański, A. Petit, and B. Rozoy. Decidability of the star problem in $A^* \times \{b\}^*$. *Information Processing Letters*, 44:65–71, 1992.

[13] S. Ginsburg and E. Spanier. Bounded regular sets. In *Proceedings of the AMS*, volume 17:5, pages 1043–1049, 1966.

[14] S. Ginsburg and E. Spanier. Semigroups, presburger formulas and languages. *Pacific Journal of Mathematics*, 16:285–296, 1966.

[15] P. A. Grillet. *Semigroups: An Introduction to the Structure Theory*, volume 193 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel Drekker, Inc., New York, 1995.

[16] K. Hashiguchi. A decision procedure for the order of regular events. *Theoretical Computer Science*, 8:69–72, 1979.

[17] S. C. Kleene. Representation of events in nerve nets and finite automata. In C. Shannon and J. McCarthy, editors, *Automata Studies, Annals of Math. Studies 34*, pages 3–40. Princeton, New Jersey, 1956.

[18] A. Mazurkiewicz. Concurrent program schemes and their interpretations. DAIMI Rep. PB 78, Aarhus University, 1977.

[19] Y. Métivier. Une condition suffisante de reconnaissabilité dans un monoïde partiellement commutatif. *R.A.I.R.O. - Informatique Théorique et Applications*, 20:121–127, 1986.

[20] Y. Métivier and G. Richomme. On the star operation and the finite power property in free partially commutative monoids. In Patrice Enjalbert et al., editors, *STACS'94 Proceedings*, volume 775 of *Lecture Notes in Computer Science*, pages 341–352. Springer-Verlag, Berlin, 1994.

[21] Y. Métivier and G. Richomme. New results on the star problem in trace monoids. *Information and Computation*, 119(2):240–251, 1995.

[22] J. Mezei and J. B. Wright. Algebraic automata and context-free sets. *Information and Control*, 11:3–29, 1967.

[23] E. Ochmański. Recognizable trace languages. Chapter 6 in [9], pages 167-204.

[24] E. Ochmański. *Regular Trace Languages (in Polish)*. PhD thesis, Warszawa, 1984.

[25] E. Ochmański. Notes on a star mystery. *Bulletin of the EATCS*, 40:252–257, 1990.

[26] G. Richomme. Some trace monoids where both the star problem and the finite power property problem are decidable. Internal report 755-93, LaBRI - Université Bordeaux I, 1993.

[27] G. Richomme. Decidability equivalence between the star problem and the finite power property problem in trace monoids. Internal report 835-94, LaBRI - Université Bordeaux I, 1994.

[28] G. Richomme. Some trace monoids where both the star problem and the finite power property problem are decidable. In I. Privara et al., editors, *MFCS'94 Proceedings*, volume 841 of *Lecture Notes in Computer Science*, pages 577–586. Springer-Verlag, Berlin, 1994.

[29] J. Sakarovitch. The "last" decision problem for rational trace languages. In I. Simon, editor, *Proceedings of the 1st Latin American Symposium on Theoretical Computer Science*, volume 583 of *Lecture Notes in Computer Science*, pages 460–473. Springer-Verlag, Berlin Heidelberg New York, 1992.

[30] I. Simon. Limited subsets of a free monoid. In *Proceedings of the 19th IEEE Annual Symposium on Foundations of Computer Science*, pages 143–150. North Carolina Press, 1978.

[31] S. Yu. Regular languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, Vol. 1, Word, Language, Grammar*, pages 41–110. Springer-Verlag, Berlin Heidelberg, 1997.