On All Things Star-Free

Thomas Place
Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400, Talence and IUF, France

Marc Zeitoun
Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400, Talence, France

Abstract

We investigate the star-free closure, which associates to a class of languages its closure under Boolean operations and marked concatenation. We prove that the star-free closure of any finite class and of any class of groups languages with decidable separation (plus mild additional properties) has decidable separation. We actually show decidability of a stronger property, called covering. This generalizes many results on the subject in a unified framework. A key ingredient is that star-free closure coincides with another closure operator where Kleene stars are also allowed in restricted contexts.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory

Keywords and phrases Regular languages, separation problem, star-free closure, group languages

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.126

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version `https://arxiv.org/abs/1904.11863`

Funding DeLTA project (ANR-16-CE40-0007)

1 Introduction

This paper investigates a remarkable operation on classes of languages: the *star-free closure*. It builds a new class $SF(\mathcal{C})$ from an input class $\mathcal{C}$ by closing it under union, complement and concatenation. This generalizes an important specific class: the one of *star-free languages*, *i.e.*, the star-free closure of the class consisting of all finite languages. Star-free languages are those that can be defined in first order logic [12]. The correspondence was lifted to the quantifier alternation hierarchy of first order logic by Thomas [30], which corresponds to a classification of star-free languages: the dot-depth hierarchy [4]. These results extend to the star-free closure [22]. For each input class $\mathcal{C}$, $SF(\mathcal{C})$ corresponds to a variant of first-order logic (specified by the set of predicates that are allowed). Moreover, its quantifier alternation hierarchy corresponds to a classification of $SF(\mathcal{C})$: the concatenation hierarchy of basis $\mathcal{C}$.

Schützenberger proved that one may decide whether a regular language is star-free [27]. This result established a framework for investigating and understanding classes of languages, based on the *membership problem*: is it decidable to test whether an input regular language belongs to the class under investigation? Similar results were obtained for other prominent classes. Yet, this fruitful line of research also includes some of the most famous open problems in automata theory. For example, only the first levels of the dot-depth hierarchy are known to have decidable membership (see [14] for a survey).

Recently, these results were unified and generalized. First, the problem itself was strengthened: membership was replaced by separation as a means to investigate classes. The separation problem asks whether two input languages can be separated by one from the class under study. While more general and difficult than membership, separation is also more flexible. This was exploited to show that separation is decidable for several levels in the dot-depth hierarchy [19, 17]. In fact, this is a particular instance of a *generic* result applying to every hierarchy whose basis $\mathcal{C}$ is *finite* and satisfies some mild properties [18, 21]. Moreover,

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 126; pp. 126:1–126:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the same result was obtained when the basis $\mathcal{C}$ is a class of group languages (*i.e.*, recognized by a finite group) with decidable separation [26]. Altogether, these results generalize most of the known results regarding the decidability of levels in concatenation hierarchies.

Contributions. This paper is a continuation of these research efforts. Instead of looking at levels within hierarchies, we investigate the star-free closure as a whole. First, we show that the star-free closure of a finite class has decidable separation. We then use this result to establish our main theorem: the star-free closure of a class of group languages with *decidable separation* has also decidable separation. In both cases, we actually prove the decidability of a stronger property called covering. Let us mention some important features of this work.

A first point is that the case of a finite class is important by itself. Foremost, it is a crucial step for the main result on the star-free closure of classes of group languages. Second, it yields a new proof that covering is decidable for the star-free languages (this is shown in [20] or can be derived from [9, 1]). This new proof is simpler and generic. While the original underlying technique goes back to Wilke [31], the proof has been simplified at several levels. The main simplification is obtained thanks to an abstract framework, introduced in [24]. It is based on the central notion of rating map, which is meant to measure the quality of a separator. For the framework to be relevant, we actually need to generalize separation to multiple input languages, which leads to the covering problem. Another key difference is that previously existing proofs (specific to the star-free languages) involve abstracting words by new letters at some point, which requires a relabeling procedure and a change of alphabet. Here, we cannot use this approach as the classes we build with star-free closure are less robust in general. We work with a fixed alphabet, which also makes the proof simpler.

A crucial ingredient in the proof is the notion of prefix code with bounded synchronization delay. Generalizing a definition of Schützenberger [28] which was also considered by Diekert and Walter [6, 7], we define a new closure operator that permits Kleene stars on such languages (this is a semantic property). This yields an operator that happens to coincide with the star-free closure when applied to the classes that we investigate. It serves as a key intermediary: in our proofs, we heavily rely on Kleene stars to construct languages. We therefore present this important step in the body of the paper (Theorem 7). Moreover, its proof provides yet another characterization of $SF(\mathcal{C})$, which is effective when the class $\mathcal{C}$ is finite (thus generalizing Schützenberger's membership result). At last regarding membership, it is worth pointing out that not only do we cover more cases, but also that it is straightforward to reprove the known algebraic characterizations from our results (see *e.g.*, [3]).

Finally, let us present important applications of our main result applying to input classes made of group languages. First, one may look at the input class containing all group languages. Straubing [29] described an algebraic counterpart of the star-free closure of this class, which was then shown to be recursive by Rhodes and Karnofsky [10]. Altogether, this implies that membership is decidable for the star-free closure of group languages, as noted by Margolis and Pin [11]. Here, we are able to generalize this result to separation and covering as separation is known to be decidable for the group languages [2].

Another important application is the class of languages definable by first-order logic with modular predicates $\mathrm{FO}(<, \mathrm{MOD})$. This class is known to have decidable membership [3]. Moreover, it is the star-free closure of the class consisting of the languages counting the length of words modulo some number. Since this input class is easily shown to have decidable separation (see [26] for example), our main theorem applies.

The third application applies to first-order logic endowed with predicates counting the number of occurrences of a letter before a position modulo some integer. The languages definable in this logic form the star-free closure of the languages recognized by Abelian groups:

this follows from a generic correspondence between star-free closure and variants of first-order logic [22, 13], together with the description of languages recognized by Abelian groups [8]. Our main theorem applies, since the class of Abelian groups has decidable separation [5, 1].

Organization. In Section 2, we recall some useful background. Section 3 presents a generic characterization of star-free closure. Then, Sections 4 and 5 are devoted to our two main theorems applying respectively to finite input classes and those made of group languages. Due to space limitations, several proofs are postponed to the full version of the paper [25].

2 Preliminaries

We fix a finite alphabet A for the whole paper. As usual, A^* denotes the set of all words over A, including the empty word ε. For $u, v \in A^*$, we denote by uv the word obtained by concatenating u and v. A *language* is a subset of A^*. We lift concatenation to languages: for $K, L \subseteq A^*$, we let $KL = \{uv \mid u \in K \text{ and } v \in L\}$. Finally, we use Kleene star: if $K \subseteq A^*$, K^+ denotes the union of all languages K^n for $n \geq 1$ and $K^* = K^+ \cup \{\varepsilon\}$.

A *class of languages* is a set of languages. A class $\mathcal{C}$ is a *Boolean algebra* when it is closed under union, intersection and complement. Moreover, $\mathcal{C}$ is *quotient-closed* if for every $L \in \mathcal{C}$ and $w \in A^*$, the languages $w^{-1}L \stackrel{\text{def}}{=} \{u \in A^* \mid wu \in L\}$ and $Lw^{-1} \stackrel{\text{def}}{=} \{u \in A^* \mid uw \in L\}$ belong to $\mathcal{C}$. All classes considered in the paper are quotient-closed Boolean algebras containing only *regular languages* (this will be implicit in our statements). These are the languages that can be equivalently defined by monadic second-order logic, finite automata or finite monoids. We briefly recall the monoid-based definition below.

We shall often consider *finite* quotient-closed Boolean algebras. If $\mathcal{C}$ is such a class, one may associate a canonical equivalence $\sim_{\mathcal{C}}$ over A^*. For $w, w' \in A^*$, $w \sim_{\mathcal{C}} w'$ if and only if $w \in L \Leftrightarrow w' \in L$ for every $L \in \mathcal{C}$. Moreover, we write $[w]_{\mathcal{C}} \in A^*/{\sim_{\mathcal{C}}}$ for the $\sim_{\mathcal{C}}$-class of w. One may then verify that the languages in $\mathcal{C}$ are exactly the unions of $\sim_{\mathcal{C}}$-classes. Moreover, since $\mathcal{C}$ is quotient-closed, $\sim_{\mathcal{C}}$ is a congruence for word concatenation (see [22] for proofs).

Regular languages. A *monoid* is a set M endowed with an associative multiplication $(s, t) \mapsto s \cdot t$ (also denoted by st) having a neutral element 1_M. An *idempotent* of a monoid M is an element $e \in M$ such that $ee = e$. It is folklore that for any *finite* monoid M, there exists a natural number $\omega(M)$ (denoted by ω when M is understood) such that s^ω is an idempotent for every $s \in M$. Observe that A^* is a monoid whose multiplication is concatenation (the neutral element is ε). Thus, we may consider monoid morphisms $\alpha : A^* \to M$ where M is an arbitrary monoid. Given such a morphism and $L \subseteq A^*$, we say that L is *recognized* by α when there exists a set $F \subseteq M$ such that $L = \alpha^{-1}(F)$. A language L is *regular* if and only if it is recognized by a morphism into a *finite* monoid. Moreover, it is known that there exists a canonical recognizer of L, which can be computed from any representation of L (such as a finite automaton): the syntactic morphism of L. We refer the reader to [15] for details.

Group languages. A group is a monoid G in which every element $g \in G$ has an inverse $g^{-1} \in G$, *i.e.*, $gg^{-1} = g^{-1}g = 1_G$. A "*group language*" is a language L recognized by a morphism into a *finite group*. All classes of group languages investigated here are quotient-closed Boolean algebras. Typically, publications on the topic consider *varieties* of group languages which is more restrictive: they involve an additional closure property called "inverse morphic image" (see [13]). For example, the class MOD described below is *not* a variety.

▶ **Example 1.** A simple example of quotient-closed Boolean algebra of group languages is the class of *all* group languages: GR. Another one is MOD, which contains the Boolean combinations of languages $\{w \in A^* \mid |w| = k \mod m\}$ with $k, m \in \mathbb{N}$ such that $k < m$.

Decision problems. We rely on three decision problems to investigate classes of languages. Each one depends on a parameter class $\mathcal{C}$, which we fix for the definition. The first problem, *$\mathcal{C}$-membership*, takes a single regular language L as input and asks whether $L \in \mathcal{C}$.

The second one, *$\mathcal{C}$-separation*, takes two regular languages L_1 and L_2 as input and asks whether L_1 is $\mathcal{C}$-separable from L_2 (is there a third language $K \in \mathcal{C}$ such that $L_1 \subseteq K$ and $L_2 \cap K = \emptyset$). This generalizes membership: $L \in \mathcal{C}$ if and only if L is $\mathcal{C}$-separable from $A^* \setminus L$.

The third problem, *$\mathcal{C}$-covering* was introduced in [24]. Given a language L, a *cover of L* is a finite set of languages $\mathbf{K}$ such that $L \subseteq \bigcup_{K \in \mathbf{K}} K$. Moreover, a $\mathcal{C}$-cover of L is a cover $\mathbf{K}$ of L such that all $K \in \mathbf{K}$ belong to $\mathcal{C}$. Consider a pair $(L_1, \mathbf{L}_2)$ where L_1 is a language and $\mathbf{L}_2$ is a *finite set of languages*. We say that $(L_1, \mathbf{L}_2)$ is *$\mathcal{C}$-coverable* when there exists a $\mathcal{C}$-cover $\mathbf{K}$ of L_1 such that for every $K \in \mathbf{K}$, there exists $L \in \mathbf{L}_2$ satisfying $K \cap L = \emptyset$. The *$\mathcal{C}$-covering problem* takes as input a single regular language L_1 and a finite set of regular languages $\mathbf{L}_2$. It asks whether $(L_1, \mathbf{L}_2)$ $\mathcal{C}$-coverable. Covering generalizes separation if $\mathcal{C}$ is closed under union: L_1 is $\mathcal{C}$-separable from L_2, if and only if $(L_1, \{L_2\})$ is $\mathcal{C}$-coverable (see [24]).

Star-free closure and main results. We investigate an operation defined on classes: *star-free closure*. Consider a class $\mathcal{C}$. The *star-free closure of $\mathcal{C}$*, denoted by $SF(\mathcal{C})$, is the least class containing $\mathcal{C}$ and the singletons $\{a\}$ for every $a \in A$, and closed under Boolean operations and concatenation. It is standard and simple to verify that when $\mathcal{C}$ is a quotient-closed Boolean algebra (which will always be the case here), this is also the case for $SF(\mathcal{C})$.

Our main theorems state conditions on the input class $\mathcal{C}$ guaranteeing decidability of our decision problems for $SF(\mathcal{C})$. First, we may handle *finite* classes.

▶ **Theorem 2.** *Let $\mathcal{C}$ be a finite quotient-closed Boolean algebra. Then, membership, separation and covering are decidable for $SF(\mathcal{C})$.*

The second theorem applies to input classes made of group languages.

▶ **Theorem 3.** *Let $\mathcal{C}$ be a quotient-closed Boolean algebra of group languages with decidable separation. Then, membership, separation and covering are decidable for $SF(\mathcal{C})$.*

The remainder of the paper is devoted to proving these theorems. We first focus on *$SF(\mathcal{C})$-membership* in Section 3. Naturally, this is weaker than directly handling $SF(\mathcal{C})$-covering. Yet, detailing membership independently allows to introduce many proof ideas and techniques that are needed to prove the "full" theorems. We detail these theorems in Sections 4 and 5. We only present the algorithms: proofs are deferred to the full paper [25].

3 Bounded synchronization delay and algebraic characterization

This section is devoted to $SF(\mathcal{C})$-membership. We handle it with a generic algebraic characterization of $SF(\mathcal{C})$ (effective under the hypotheses of Theorems 2 and 3), generalizing earlier work by Pin, Straubing and Thérien [29, 16]. We rely on an alternate definition of star-free closure involving a semantic restriction of the Kleene star, which we first present.

3.1 Bounded synchronization delay

We define a second operation on classes of languages $\mathcal{C} \mapsto SD(\mathcal{C})$. We shall later prove that it coincides with star-free closure (provided that $\mathcal{C}$ satisfies mild hypotheses). It is based on the work of Schützenberger [28] who defined a single class SD corresponding to the star-free languages (*i.e.*, $SF(\{\emptyset, A^*\})$). Here, we generalize it as an operation. The definition involves a semantic restriction of the Kleene star operation on languages: it may only be applied to "*prefix codes with bounded synchronization delay*". Introducing this notion requires basic definitions from coding theory that we first recall.

A language $K \subseteq A^*$ is a *prefix code* when $\varepsilon \notin K$ and $K \cap KA^+ = \emptyset$ (no word in K has a strict prefix in K). Note that this implies the following weaker property that we shall use implicitly: every $w \in K^*$ admits a *unique* decomposition $w = w_1 \cdots w_n$ with $w_1, \dots, w_n \in K$ (this property actually defines *codes* which are more general).

Given $d \geq 1$, a prefix code $K \subseteq A^+$ has *synchronization delay* d if for every $u, v, w \in A^*$ such that $uvw \in K^+$ and $v \in K^d$, we have $uv \in K^+$. Finally, a prefix code $K \subseteq A^+$ has *bounded synchronization delay* when it has synchronization delay d for some $d \geq 1$.

▶ **Example 4.** Let $A = \{a, b\}$. Clearly, $\{ab\}$ is a prefix code with synchronization delay 1: if $uvw \in (ab)^+$ and $v = ab$, we have $uv \in (ab)^+$. Similarly, one may verify that $(aab)^*ab$ is a prefix code with synchronization delay 2 (but not 1). On the other hand, $\{aa\}$ does not have bounded synchronization delay. If $d \geq 1$, $a(aa)^d a \in (aa)^*$ but $a(aa)^d \notin (aa)^*$.

We present the operation $\mathcal{C} \mapsto SD(\mathcal{C})$. The definition involves *unambiguous concatenation*. Given $K, L \subseteq A^*$, their concatenation KL is *unambiguous* when every word $w \in KL$ admits a *unique* decomposition $w = uv$ with $u \in K$ and $v \in L$. Given a class $\mathcal{C}$, $SD(\mathcal{C})$ is the least class containing $\emptyset$ and $\{a\}$ for every $a \in A$, and closed under the following properties:

- **Intersection with $\mathcal{C}$:** if $K \in SD(\mathcal{C})$ and $L \in \mathcal{C}$, then $K \cap L \in SD(\mathcal{C})$.
- **Disjoint union:** if $K, L \in SD(\mathcal{C})$ are disjoint, then $K \uplus L \in SD(\mathcal{C})$.
- **Unambiguous product:** if $K, L \in SD(\mathcal{C})$ and KL is unambiguous, then $KL \in SD(\mathcal{C})$.
- **Kleene star for prefix codes with bounded synchronization delay:** if $K \in SD(\mathcal{C})$ is a prefix code with bounded synchronization delay, then $K^* \in SD(\mathcal{C})$.

▶ Remark 5. Schützenberger proved in [28] that $SD(\{\emptyset, A^*\}) = SF(\{\emptyset, A^*\})$. His definition of $SD(\{\emptyset, A^*\})$ was slightly less restrictive than ours: it does not require that the unions are disjoint and the concatenations unambiguous. It will be immediate from the correspondence with star-free closure that the two definitions are equivalent.

▶ Remark 6. This closure operation is different from standard ones. Instead of requiring that $\mathcal{C} \subseteq SD(\mathcal{C})$, we impose a stronger requirement: intersection with languages in $\mathcal{C}$ is allowed. If we only asked that $\mathcal{C} \subseteq SD(\mathcal{C})$, we would get a weaker operation which does not correspond to star-free closure in general. For example, let $A = \{a, b\}$ and consider the class MOD of Example 1. Observe that $(aa)^* \in SD(\text{MOD})$. Indeed, $\{a\} \in SD(\text{MOD})$ has bounded synchronization delay, $(AA)^* \in \text{MOD}$ and $(aa)^* = a^* \cap (AA)^*$. Yet, one may verify that $(aa)^*$ cannot be built from the languages of MOD with union, concatenation and Kleene star applied to prefix codes with bounded synchronization delay.

3.2 Algebraic characterization of star-free closure

We now reduce deciding membership for $SF(\mathcal{C})$ to computing $\mathcal{C}$-*stutters*. Let us first define this new notion. Let $\mathcal{C}$ be a quotient-closed Boolean algebra and $\alpha : A^* \to M$ be a morphism. A $\mathcal{C}$-*stutter* for α is an element $s \in M$ such that for every $\mathcal{C}$-cover $\mathbf{K}$ of $\alpha^{-1}(s)$, there exists

$K \in \mathbf{K}$ satisfying $K \cap KK \neq \emptyset$. When α is understood, we simply speak of a $\mathcal{C}$-stutter. Finally, we say that α is *$\mathcal{C}$-aperiodic* when for every $\mathcal{C}$-stutter $s \in M$, we have $s^\omega = s^{\omega+1}$. The reduction is stated in the following theorem.

▶ **Theorem 7.** *Let $\mathcal{C}$ be a quotient-closed Boolean algebra and consider a regular language $L \subseteq A^*$. The following properties are equivalent:*

1. *$L \in SF(\mathcal{C})$.*
2. *$L \in SD(\mathcal{C})$.*
3. *The syntactic morphism of L is $\mathcal{C}$-aperiodic.*

Naturally, the characterization need not be effective: this depends on $\mathcal{C}$. Deciding whether a morphism is $\mathcal{C}$-aperiodic boils down to computing $\mathcal{C}$-stutters. Yet, this is possible under the hypotheses of Theorems 2 and 3. First, if $\mathcal{C}$ is a finite quotient-closed Boolean algebra, deciding whether an element is a $\mathcal{C}$-stutter is simple: there are finitely many $\mathcal{C}$-covers and we may check them all. If $\mathcal{C}$ is a quotient-closed Boolean algebra of group languages, the question boils down to $\mathcal{C}$-separation as stated in the next lemma (proved in [25]).

▶ **Lemma 8.** *Let $\mathcal{C}$ be a quotient-closed Boolean algebra of group languages and $\alpha : A^* \to M$ be a morphism. For all $s \in M$, s is a $\mathcal{C}$-stutter if and only if $\{\varepsilon\}$ is **not** $\mathcal{C}$-separable from $\alpha^{-1}(s)$.*

Altogether, we obtain the membership part in Theorems 20 and 25. We conclude the section with an extended proof sketch for the most interesting direction in Theorem 7: 3) ⇒ 2) (a detailed proof for the two other directions is provided in the full version of this paper [25]).

Proof of 3) ⇒ 2) in Theorem 7. Let $\mathcal{C}$ be a quotient-closed Boolean algebra and $\alpha : A^* \to M$ be a $\mathcal{C}$-aperiodic morphism. We show that all languages recognized by α belong to $SD(\mathcal{C})$.

Given $K \subseteq A^*$ and $s \in M$, we say that K is *s-safe* when $s\alpha(u) = s\alpha(v)$ for every $u, v \in K$. We extend this notion to sets of languages: such a set $\mathbf{K}$ is s-safe when every $K \in \mathbf{K}$ is s-safe. We shall use s as an induction parameter. Finally, given a language $P \subseteq A^*$, an $SD(\mathcal{C})$-partition of P is a finite partition of P into languages of $SD(\mathcal{C})$.

▶ **Proposition 9.** *Let $P \subseteq A^+$ be a prefix code with bounded synchronization delay. Assume that there exists a 1_M-safe $SD(\mathcal{C})$-partition of P. Then, for every $s \in M$, there exists an s-safe $SD(\mathcal{C})$-partition of P^*.*

We first apply Proposition 9 to conclude the main argument. We show that every language recognized by α belongs to $SD(\mathcal{C})$. By definition, $SD(\mathcal{C})$ is closed under disjoint union. Hence, it suffices to show that $\alpha^{-1}(t) \in SD(\mathcal{C})$ for every $t \in M$. We fix $t \in M$ for the proof.

Clearly, $A \subseteq A^+$ is a prefix code with bounded synchronization delay and $\{\{a\} \mid a \in A\}$ is a 1_M-safe $SD(\mathcal{C})$-partition of A. Hence, Proposition 9 (applied in the case $s = 1_M$) yields a 1_M-safe $SD(\mathcal{C})$-partition $\mathbf{K}$ of A^*. One may verify that $\alpha^{-1}(t)$ is the disjoint union of all $K \in \mathbf{K}$ intersecting $\alpha^{-1}(t)$. Hence, $\alpha^{-1}(t) \in SD(\mathcal{C})$ which concludes the main argument.

It remains to prove Proposition 9. We let $P \subseteq A^*$ be a prefix code with bounded synchronization delay, $\mathbf{H}$ a 1_M-safe $SD(\mathcal{C})$-partition of P and $s \in M$. We need to build an $SD(\mathcal{C})$-partition $\mathbf{K}$ of P^* such that every $K \in \mathbf{K}$ is s-safe. We proceed by induction on the three following parameters listed by order of importance: (1) the size of $\alpha(P^+) \subseteq M$, (2) the size of $\mathbf{H}$ and (3) the size of $s \cdot \alpha(P^*) \subseteq M$. We distinguish two cases depending on the following property of s and $\mathbf{H}$. We say that s *is $\mathbf{H}$-stable* when the following holds:

$$\text{for every } H \in \mathbf{H}, \quad s \cdot \alpha(P^*) = s \cdot \alpha(P^* H). \tag{1}$$

The base case happens when s is $\mathbf{H}$-stable. Otherwise, we use induction on our parameters.

Base case: s is H-stable. Since α is $\mathcal{C}$-aperiodic, we have the following simple fact.

▶ **Fact 10.** *There is a* finite *quotient-closed Boolean algebra $\mathcal{D} \subseteq \mathcal{C}$ such that α is $\mathcal{D}$-aperiodic.*

Since $\mathcal{D}$ is finite, we may consider the associated canonical equivalence $\sim_\mathcal{D}$ over A^*. We let $\mathbf{K} = \{P^* \cap D \mid D \in A^*/{\sim_\mathcal{D}}\}$. Clearly, $\mathbf{K}$ is a partition of P^*. Let us verify that it only contains languages in $SD(\mathcal{C})$. We have $P \in SD(\mathcal{C})$: it is the disjoint union of all languages in the $SD(\mathcal{C})$-partition $\mathbf{H}$ of P. Moreover, $P^* \in SD(\mathcal{C})$ since P is a prefix code with bounded synchronization delay. Hence, $P^* \cap D \in SD(\mathcal{C})$ for every $D \in A^*/{\sim_\mathcal{D}}$ since $D \in \mathcal{D} \subseteq \mathcal{C}$. Therefore, it remains to show that every language $K \in \mathbf{K}$ is s-safe. This is a consequence of the following lemma which is proved using the hypothesis (1) that s is $\mathbf{H}$-stable.

▶ **Lemma 11.** *For every $u, v \in P^*$ such that $u \sim_\mathcal{D} v$, we have $s\alpha(u) = s\alpha(v)$.*

Inductive step: s is not H-stable. By hypothesis, we know that (1) does not hold. Therefore, we get some $H \in \mathbf{H}$ such that the following **strict** inclusion holds,

$$s \cdot \alpha(P^*H) \subsetneq s \cdot \alpha(P^*). \tag{2}$$

We fix this language $H \in \mathbf{H}$ for the remainder of the proof. The following lemma is proved by induction on our second parameter (the size of $\mathbf{H}$).

▶ **Lemma 12.** *There exists a 1_M-safe $SD(\mathcal{C})$-partition $\mathbf{U}$ of $(P \setminus H)^*$.*

We fix the partition $\mathbf{U}$ of $(P\setminus H)^*$ given by Lemma 12 and distinguish two independent sub-cases. Since $H \subseteq P$ (as H is an element of the partition $\mathbf{H}$ of P), we have $\alpha(P^*H) \subseteq \alpha(P^+)$. We use a different argument depending on whether this inclusion is strict or not.

Sub-case 1: $\alpha(P^*H) = \alpha(P^+)$. Since H is 1_M-safe by hypothesis, there exists $t \in M$ such that $\alpha(H) = \{t\}$. Similarly, since every $U \in \mathbf{U}$ is 1_M-safe, there exists $r_U \in M$ such that $\alpha(U) = \{r_U\}$. The construction of $\mathbf{K}$ is based on the next lemma which is proved using (2), the hypothesis of Sub-case 1 and induction on our third parameter (the size of $s \cdot \alpha(P^*) \subseteq M$).

▶ **Lemma 13.** *For every $U \in \mathbf{U}$, there exists an sr_Ut-safe $SD(\mathcal{C})$-partition $\mathbf{W}_U$ of P^*.*

We are ready to define the partition $\mathbf{K}$ of P^*. Using Lemma 13, we define,

$$\mathbf{K} = \mathbf{U} \cup \bigcup_{U \in \mathbf{U}} \{UHW \mid W \in \mathbf{W}_U\}$$

It remains to show that $\mathbf{K}$ is an s-safe $SD(\mathcal{C})$-partition of P^*. First, $\mathbf{K}$ is a partition of P^* since P is a prefix code and $H \subseteq P$. Indeed, every word $w \in P^*$ admits a *unique* decomposition $w = w_1 \cdots w_n$ with $w_1, \dots, w_n \in P$. If no factor w_i belongs to H, then $w \in (P \setminus H)^*$ and w belongs to some unique $U \in \mathbf{U}$. Otherwise, let w_i be the leftmost factor such that $w_i \in H$. Thus, $w_1 \cdots w_{i-1} \in (P \setminus H)^*$, which also yields a unique $U \in \mathbf{U}$ such that $w_1 \cdots w_{i-1} \in U$ and $w_{i+1} \cdots w_n \in P^*$ which yields a unique $W \in \mathbf{W}_U$ such that $w_{i+1} \cdots w_n \in W$. Thus, $w \in UHW$ which is an element of $\mathbf{K}$ (the only one containing w).

Moreover, every $K \in \mathbf{K}$ belongs to $SD(\mathcal{C})$. If $K \in \mathbf{U}$, this is immediate by definition of $\mathbf{U}$ in Lemma 12. Otherwise, $K = UHW$ with $U \in \mathbf{U}$ and $W \in \mathbf{W}_U$. We know that $U, H, W \in SD(\mathcal{C})$ by definition. Moreover, one may verify that the concatenation UHW is *unambiguous* since P is a prefix code, $U \subseteq (P \setminus H)^*$ and $W \subseteq H^*$. Hence, $K \in SD(\mathcal{C})$.

Finally, we verify that $\mathbf{K}$ is s-safe. Consider $K \in \mathbf{K}$ and $w, w' \in K$, we show that $s\alpha(w) = s\alpha(w')$. If $K \in \mathbf{U}$, this is immediate: $\mathbf{U}$ is 1_M-safe by definition. Otherwise, $K = UHW$ with $U \in \mathbf{U}$ and $W \in \mathbf{W}_U$. By definition, $\alpha(H) = \{t\}$ and $\alpha(U) = \{r_U\}$ which implies that $s\alpha(w) = sr_U t\alpha(x)$ and $s\alpha(w') = sr_U t\alpha(x')$ for $x, x' \in W$. Moreover, $W \in \mathbf{W}_U$ is $sr_U t$-safe by definition. Hence, $s\alpha(w) = s\alpha(w')$, which concludes the proof of this sub-case.

Sub-case 2: $\boldsymbol{\alpha(P^*H) \subsetneq \alpha(P^+)}$. Consider $w \in P^*$. Since P is a prefix code, w admits a unique decomposition $w = w_1 \cdots w_n$ with $w_1, \dots, w_n \in P$. We may look at the rightmost factor $w_i \in H \subseteq P$ to uniquely decompose w in two parts (each of them possibly empty): the prefix $w_1 \cdots w_i \in ((P \setminus H)^*H)^*$ and the suffix in $w_{i+1} \cdots w_n \in (P \setminus H)^*$. Using induction, we construct $SD(\mathcal{C})$-partitions of the possible languages of prefixes and suffixes. Then, we combine them to construct a partition of the whole set P^*. We already handled the suffixes: $\mathbf{U}$ is an $SD(\mathcal{C})$-partition of $(P \setminus H)^*$. The prefixes are handled using the hypothesis of Sub-case 2 and induction on our first parameter (the size of $\alpha(P^+)$).

▶ **Lemma 14.** *There exists a 1_M-safe $SD(\mathcal{C})$-partition $\mathbf{V}$ of $((P \setminus H)^*H)^*$.*

Using Lemma 14, we define $\mathbf{K} = \{VU \mid V \in \mathbf{V} \text{ and } U \in \mathbf{U}\}$. It follows from the above discussion that $\mathbf{K}$ is a partition of P^* since $\mathbf{V}$ and $\mathbf{U}$ are partitions of $((P \setminus H)^*H)^*$ and $(P \setminus H)^*$, respectively. Moreover, every $K \in \mathbf{K}$ belongs to $SD(\mathcal{C})$: $K = VU$ with $V \in \mathbf{V}$ and $U \in \mathbf{U}$, and one may verify that this is an *unambiguous* concatenation. It remains to show that $\mathbf{K}$ is s-safe. Let $K \in \mathbf{K}$ and $w, w' \in K$. We show that $s\alpha(w) = s\alpha(w')$. By definition, we have $K = VU$ with $V \in \mathbf{V}$ and $U \in \mathbf{U}$. Therefore, $w = vu$ and $w' = v'u'$ with $u, u' \in U$ and $v, v' \in V$. Since U and V are both 1_M-safe by definition, we have $\alpha(u) = \alpha(u')$ and $\alpha(v) = \alpha(v')$. It follows that $s\alpha(w) = s\alpha(w')$, which concludes the proof of Proposition 9. ◀

4 Covering when the input class is finite

This section is devoted to Theorem 2. We show that when $\mathcal{C}$ is a finite quotient-closed Boolean algebra, $SF(\mathcal{C})$-covering is decidable by presenting a generic algorithm. It is formulated within a framework designed to handle covering questions, which was originally introduced in [24]. We start by briefly recalling it (we refer the reader to [24] for details).

4.1 Rating maps and optimal imprints

The framework is based on an algebraic object called "rating map". These are morphisms of commutative and idempotent monoids. We write such monoids $(R, +)$: the binary operation "+" is called *addition* and the neutral element is denoted by 0_R. Being idempotent means that $r + r = r$ for every $r \in R$. For every commutative and idempotent monoid $(R, +)$, one may define a canonical ordering $\leq$ over R: for $r, s \in R$, we have $r \leq s$ when $r + s = s$. One may verify that $\leq$ is a partial order which is compatible with addition.

▶ **Example 15.** For every set E, $(2^E, \cup)$ is an idempotent and commutative monoid. The neutral element is $\emptyset$ and the canonical ordering is inclusion.

A rating map is a morphism $\rho : (2^{A^*}, \cup) \to (R, +)$ where $(R, +)$ is a *finite* idempotent and commutative monoid, called the *rating set of* ρ. That is, ρ is a map from 2^{A^*} to R such that $\rho(\emptyset) = 0_R$ and $\rho(K_1 \cup K_2) = \rho(K_1) + \rho(K_2)$ for every $K_1, K_2 \subseteq A^*$.

For the sake of improved readability, when applying a rating map ρ to a singleton set $\{w\}$, we write $\rho(w)$ for $\rho(\{w\})$. Moreover, we write $\rho_* : A^* \to R$ for the restriction of ρ to A^*: for every $w \in A^*$, we have $\rho_*(w) = \rho(w)$ (this notation is useful when referring to the language $\rho_*^{-1}(r) \subseteq A^*$, which consists of all words $w \in A^*$ such that $\rho(w) = r$).

Most of the theory makes sense for arbitrary rating maps. However, we shall often have to work with special rating maps satisfying additional properties. We define two kinds.

Nice rating maps. A rating map $\rho : 2^{A^*} \to R$ is *nice* when, for every nonempty language $K \subseteq A^*$, there exist finitely many words $w_1, \dots, w_n \in K$ such that $\rho(K) = \rho(w_1)+\cdots+\rho(w_k)$.

When a rating map $\rho : 2^{A^*} \to R$ is nice, it is characterized by the canonical map $\rho_* : A^* \to R$. Indeed, for $K \subseteq A^*$, we may consider the sum of all elements $\rho(w)$ for $w \in K$: while it may be infinite, this sum boils down to a finite one since R is commutative and idempotent. The hypothesis that ρ is nice implies that $\rho(K)$ is equal to this sum.

Multiplicative rating maps. A rating map $\rho : 2^{A^*} \to R$ is multiplicative when its rating set R has more structure: it needs to be an *idempotent semiring*. A *semiring* is a tuple $(R, +, \cdot)$ where R is a set and "$+$" and "$\cdot$" are two binary operations called addition and multiplication. Moreover, $(R, +)$ is a commutative monoid, $(R, \cdot)$ is a monoid (the neutral element is denoted by 1_R), the multiplication distributes over addition and the neutral element "0_R" of $(R, +)$ is a zero for $(R, \cdot)$ ($0_R \cdot r = r \cdot 0_R = 0_R$ for every $r \in R$). A semiring R is *idempotent* when $r + r = r$ for every $r \in R$, *i.e.*, when the additive monoid $(R, +)$ is idempotent (there is no additional constraint on the multiplicative monoid $(R, \cdot)$).

▶ **Example 16.** A key example of an infinite idempotent semiring is the set 2^{A^*}. Union is the addition and language concatenation is the multiplication (with $\{\varepsilon\}$ as neutral element).

Let $\rho : 2^{A^*} \to R$ be a rating map: $(R, +)$ is an idempotent commutative monoid and ρ is a morphism from $(2^{A^*}, \cup)$ to $(R, +)$. We say that ρ is *multiplicative* when the rating set R is equipped with a multiplication "$\cdot$" such that $(R, +, \cdot)$ is an idempotent semiring and ρ is also a monoid morphism from $(2^{A^*}, \cdot)$ to $(R, \cdot)$. That is, the two following additional axioms have to be satisfied: $\rho(\varepsilon) = 1_R$ and $\rho(K_1K_2) = \rho(K_1) \cdot \rho(K_2)$ for every $K_1, K_2 \subseteq A^*$.

▶ Remark 17. Rating maps which are both nice and multiplicative are finitely representable. As we explained, if $\rho : 2^{A^*} \to R$ is nice, it is characterized by the canonical map $\rho_* : A^* \to R$. When ρ is also multiplicative, ρ_* is finitely representable: it is a morphism into a finite monoid. Hence, we may speak of algorithms whose input is a nice multiplicative rating map.

Rating maps which are not nice and multiplicative cannot be finitely represented in general. Yet, they are crucial: while our main statements consider nice multiplicative rating maps, many proofs involve auxiliary rating maps which are neither nice nor multiplicative.

Optimal imprints. Now that we have rating maps, we turn to imprints. Consider a rating map $\rho : 2^{A^*} \to R$. Given any finite set of languages $\mathbf{K}$, we define the ρ-imprint of $\mathbf{K}$. Intuitively, when $\mathbf{K}$ is a cover of some language L, this object measures the "quality" of $\mathbf{K}$. The ρ-imprint *of* $\mathbf{K}$ is the following subset of R:

$$\mathcal{I}[\rho](\mathbf{K}) = \{r \mid r \leq \rho(K) \text{ for some } K \in \mathbf{K}\}.$$

We may now define optimality. Consider an arbitrary rating map $\rho : 2^{A^*} \to R$ and a Boolean algebra $\mathcal{C}$. Given a language L, an optimal $\mathcal{C}$-cover of L for ρ is a $\mathcal{C}$-cover $\mathbf{K}$ of L which satisfies the following property:

$$\mathcal{I}[\rho](\mathbf{K}) \subseteq \mathcal{I}[\rho](\mathbf{K}') \quad \text{for every } \mathcal{C}\text{-cover } \mathbf{K}' \text{ of } L.$$

In general, there can be infinitely many optimal $\mathcal{C}$-covers for a given rating map ρ. It is shown in [24] that there always exists at least one (using closure under intersection for $\mathcal{C}$).

Clearly, for a Boolean algebra $\mathcal{C}$, a language L and a rating map ρ, all optimal $\mathcal{C}$-covers of L for ρ have the same ρ-imprint. Hence, this unique ρ-imprint is a *canonical* object for $\mathcal{C}$, L and ρ. We call it the *$\mathcal{C}$-optimal ρ-imprint on L* and we write it $\mathcal{I}_{\mathcal{C}}[L,\rho]$:

$$\mathcal{I}_{\mathcal{C}}[L,\rho] = \mathcal{I}[\rho](\mathbf{K}) \quad \text{for any optimal } \mathcal{C}\text{-cover } \mathbf{K} \text{ of } L \text{ for } \rho.$$

We complete the definition with a simple useful fact (a proof is available in [23]).

▶ **Fact 18.** *Let $\mathcal{C}$ be a Boolean algebra, $\rho : 2^{A^*} \to R$ a rating map and $L_1, L_2 \subseteq A^*$. Then, $\mathcal{I}_{\mathcal{C}}[L_1,\rho] \cup \mathcal{I}_{\mathcal{C}}[L_2,\rho] = \mathcal{I}_{\mathcal{C}}[L_1 \cup L_2,\rho]$.*

Connection with covering. Consider the special case when the language L that needs to be covered is A^*. In that case, we write $\mathcal{I}_{\mathcal{C}}[\rho]$ for $\mathcal{I}_{\mathcal{C}}[A^*,\rho]$. It is shown in [24] that for every *Boolean algebra* $\mathcal{C}$, deciding $\mathcal{C}$-covering formally reduces to computing $\mathcal{C}$-optimal imprints from input nice multiplicative rating maps.

▶ **Proposition 19.** *Let $\mathcal{C}$ be a Boolean algebra. Assume that there exists an algorithm which computes $\mathcal{I}_{\mathcal{C}}[\rho]$ from an input nice multiplicative rating map ρ. Then, $\mathcal{C}$-covering is decidable.*

4.2 Algorithm

We may now present our algorithm for $SF(\mathcal{C})$-covering when $\mathcal{C}$ is a finite quotient-closed Boolean algebra. We fix $\mathcal{C}$ for the presentation. In view of Proposition 19, we need to prove that one may compute $\mathcal{I}_{SF(\mathcal{C})}[\rho]$ from an input nice multiplicative rating map ρ.

Our algorithm actually computes slightly more information. Since $\mathcal{C}$ is a finite quotient-closed Boolean algebra, we may consider the equivalence $\sim_{\mathcal{C}}$ over A^*. In particular, the set $A^*/{\sim_{\mathcal{C}}}$ of $\sim_{\mathcal{C}}$-classes is a finite monoid (we write "$\bullet$" for its multiplication) and the map $w \mapsto [w]_{\mathcal{C}}$ is a morphism. Given a rating map $\rho : 2^{A^*} \to R$ we define:

$$\mathcal{P}^{\mathcal{C}}_{SF(\mathcal{C})}[\rho] = \{(C,r) \in (A^*/{\sim_{\mathcal{C}}}) \times R \mid r \in \mathcal{I}_{SF(\mathcal{C})}[C,\rho]\}$$

Observe that $\mathcal{P}^{\mathcal{C}}_{SF(\mathcal{C})}[\rho]$ captures more information than $\mathcal{I}_{SF(\mathcal{C})}[\rho]$. Indeed, it encodes all sets $\mathcal{I}_{SF(\mathcal{C})}[C,\rho]$ for $C \in A^*/{\sim_{\mathcal{C}}}$ and by Fact 18, $\mathcal{I}_{SF(\mathcal{C})}[\rho]$ is the union of all these sets.

Our main result is a least fixpoint procedure for computing $\mathcal{P}^{\mathcal{C}}_{SF(\mathcal{C})}[\rho]$ from a nice multiplicative rating map ρ. It is based on a generic characterization theorem which we first present. Given an arbitrary nice multiplicative rating map $\rho : 2^{A^*} \to R$ and a set $S \subseteq (A^*/{\sim_{\mathcal{C}}}) \times R$, we say that S is *$SF(\mathcal{C})$-saturated for ρ* when the following properties are satisfied:

1. **Trivial elements.** For every $w \in A^*$, we have $([w]_{\mathcal{C}}, \rho(w)) \in S$.
2. **Downset.** For every $(C,r) \in S$ and $q \in R$, if $q \leq r$, then $(C,q) \in S$.
3. **Multiplication.** For every $(C,q),(D,r) \in S$, we have $(C \bullet D, qr) \in S$.
4. ***SF*($\mathcal{C}$)-closure.** For all $(E,r) \in S$, if $E \in A^*/{\sim_{\mathcal{C}}}$ is idempotent, then $(E, r^\omega + r^{\omega+1}) \in S$.

▶ **Theorem 20** ($SF(\mathcal{C})$-optimal imprints ($\mathcal{C}$ finite))**.** *Let $\rho : 2^{A^*} \to R$ be a* nice *multiplicative rating map. Then, $\mathcal{P}^{\mathcal{C}}_{SF(\mathcal{C})}[\rho]$ is the least $SF(\mathcal{C})$-saturated subset of $(A^*/{\sim_{\mathcal{C}}}) \times R$ for ρ.*

Given a nice multiplicative rating map $\rho : 2^{A^*} \to R$ as input, it is clear that one may compute the least $SF(\mathcal{C})$-saturated subset of $(A^*/{\sim_{\mathcal{C}}}) \times R$ with a least fixpoint procedure. Hence, Theorem 20 provides an algorithm for computing $\mathcal{P}^{\mathcal{C}}_{SF(\mathcal{C})}[\rho]$. As we explained above, we may then compute $\mathcal{I}_{SF(\mathcal{C})}[\rho]$ from this set. Together with Proposition 19, this yields Theorem 2 as a corollary: $SF(\mathcal{C})$-covering is decidable when $\mathcal{C}$ is a finite quotient-closed Boolean algebra. Theorem 20 is proved in the full version of this paper [25].

5 Covering when the input class is made of group languages

This section is devoted to Theorem 3. We show that when $\mathcal{C}$ is a quotient-closed Boolean algebra of group languages with decidable separation, $SF(\mathcal{C})$-covering is decidable.

As in Section 4, we rely on Proposition 19: we present an algorithm computing $\mathcal{I}_{SF(\mathcal{C})}[\rho]$ from an input nice multiplicative rating map ρ. We do not work with $\mathcal{I}_{SF(\mathcal{C})}[\rho]$ itself but with another set carrying more information. Its definition requires introducing a few additional concepts. We first present them and then turn to the algorithm. For more details, see [26].

5.1 Preliminary definitions

Optimal ε-approximations. In this case, handling $SF(\mathcal{C})$ involves considering $\mathcal{C}$-optimal covers of $\{\varepsilon\}$. Since $\{\varepsilon\}$ is a singleton, there always exists such a cover consisting of a single language, which leads to the following definition.

Let $\mathcal{C}$ be a Boolean algebra (we shall use the case when $\mathcal{C}$ contains only group languages but this is not required for the definitions) and $\tau : 2^{A^*} \to Q$ a rating map. A *$\mathcal{C}$-optimal ε-approximation for τ* is a language $L \in \mathcal{C}$ such that $\varepsilon \in L$ and $\tau(L) \leq \tau(L')$ for every $L' \in \mathcal{C}$ satisfying $\varepsilon \in L'$. As expected, there always exists a $\mathcal{C}$-optimal ε-approximation for any rating map τ (see the full version of this paper [25] for a proof).

By definition, all $\mathcal{C}$-optimal ε-approximations for τ have the same image under τ. We write it $\mathbb{i}_{\mathcal{C}}[\tau] \in Q$: $\mathbb{i}_{\mathcal{C}}[\tau] = \tau(L)$ for every $\mathcal{C}$-optimal ε-approximation L for τ. It turns out that when τ is nice and multiplicative, computing $\mathbb{i}_{\mathcal{C}}[\tau]$ from τ boils down to $\mathcal{C}$-separation. This is important: this is exactly how our algorithm for $SF(\mathcal{C})$-covering depends on $\mathcal{C}$-separation.

▶ **Lemma 21.** *Let $\tau : 2^{A^*} \to Q$ be a nice rating map and $\mathcal{C}$ a Boolean algebra. Then, $\mathbb{i}_{\mathcal{C}}[\tau]$ is the sum of all $q \in Q$ such that $\{\varepsilon\}$ is not $\mathcal{C}$-separable from $\tau_*^{-1}(q)$.*

Nested rating maps. We want an algorithm which computes $\mathcal{I}_{SF(\mathcal{C})}[\rho]$ from an input nice multiplicative rating map ρ for a fixed quotient-closed Boolean algebra of group languages $\mathcal{C}$. Yet, we shall not use optimal ε-approximations with this input rating map ρ. Instead, we consider an auxiliary rating map built from ρ (the definition is taken from [23]).

Consider a Boolean algebra $\mathcal{D}$ (we shall use the case $\mathcal{D} = SF(\mathcal{C})$) and a rating map $\rho : 2^{A^*} \to R$. We build a new map $\xi_{\mathcal{D}}[\rho] : 2^{A^*} \to 2^R$ whose rating set is $(2^R, \cup)$. For every $K \subseteq A^*$, we define $\xi_{\mathcal{D}}[\rho](K) = \mathcal{I}_{\mathcal{D}}[K, \rho] \in 2^R$. It follows from Fact 18 that this is indeed a rating map (on the other hand $\xi_{\mathcal{D}}[\rho]$ need not be nice nor multiplicative, see [23] for details).

We may now explain which set is computed by our algorithm instead of $\mathcal{I}_{SF(\mathcal{C})}[\rho]$. Consider a nice multiplicative rating map $\rho : 2^{A^*} \to R$. Since $\xi_{SF(\mathcal{C})}[\rho] : 2^{A^*} \to 2^R$ is a rating map, we may consider the element $\mathbb{i}_{\mathcal{C}}[\xi_{SF(\mathcal{C})}[\rho]] \in 2^R$. By definition, $\mathbb{i}_{\mathcal{C}}[\xi_{SF(\mathcal{C})}[\rho]] = \xi_{SF(\mathcal{C})}[\rho](L)$ where L is a $\mathcal{C}$-optimal ε-approximation for $\xi_{SF(\mathcal{C})}[\rho]$. Therefore, $\mathbb{i}_{\mathcal{C}}[\xi_{SF(\mathcal{C})}[\rho]]$ is a subset of $\xi_{SF(\mathcal{C})}[\rho](A^*) = \mathcal{I}_{SF(\mathcal{C})}[A^*, \rho] = \mathcal{I}_{SF(\mathcal{C})}[\rho]$. When $\mathcal{C}$ is a quotient-closed Boolean algebra of group languages, one may compute the whole set $\mathcal{I}_{SF(\mathcal{C})}[\rho]$ from this subset.

▶ **Proposition 22.** *Let $\mathcal{C}$ be a quotient-closed Boolean algebra of group languages and $\rho : 2^{A^*} \to R$ a nice multiplicative rating map. Then, $\mathcal{I}_{SF(\mathcal{C})}[\rho]$ is the least subset of R containing $\mathbb{i}_{\mathcal{C}}[\xi_{SF(\mathcal{C})}[\rho]]$ and satisfying the three following properties:*

- ***Trivial elements.*** *For every $w \in A$, $\rho(w) \in \mathcal{I}_{SF(\mathcal{C})}[\rho]$.*
- ***Downset.*** *For every $r \in \mathcal{I}_{SF(\mathcal{C})}[\rho]$ and $q \leq r$, we have $q \in \mathcal{I}_{SF(\mathcal{C})}[\rho]$.*
- ***Multiplication.*** *For every $q, r \in \mathcal{I}_{SF(\mathcal{C})}[\rho]$, we have $qr \in \mathcal{I}_{SF(\mathcal{C})}[\rho]$.*

▶ Remark 23. Intuitively, we use $\mathbb{i}_{\mathcal{C}}[\xi_{SF(\mathcal{C})}[\rho]]$ to "nest" two optimizations: one for $\mathcal{C}$ and the other for $SF(\mathcal{C})$. Indeed, $\mathbb{i}_{\mathcal{C}}[\xi_{SF(\mathcal{C})}[\rho]] = \xi_{SF(\mathcal{C})}[\rho](L) = \mathcal{I}_{SF(\mathcal{C})}[L, \rho]$ where L is a $\mathcal{C}$-optimal ε-approximation for $\xi_{SF(\mathcal{C})}[\rho]$. Hence, $\mathbb{i}_{\mathcal{C}}[\xi_{SF(\mathcal{C})}[\rho]]$ is least set $\mathcal{I}[\rho](\mathbf{K}) \subseteq R$ (with respect to inclusion), over all $SF(\mathcal{C})$-covers $\mathbf{K}$ of some language $L \in \mathcal{C}$ containing ε.

5.2 Algorithm

We may now present our algorithm for computing $\mathcal{I}_{SF(\mathcal{C})}[\rho]$. We fix a quotient-closed Boolean algebra of group languages $\mathcal{C}$ for the presentation. As expected, the main procedure computes $\mathbb{i}_{\mathcal{C}}[\xi_{SF(\mathcal{C})}[\rho]]$ (see Proposition 22). In this case as well, this procedure is obtained from a characterization theorem.

Consider a nice multiplicative rating map $\rho : 2^{A^*} \to R$. We define the $SF(\mathcal{C})$-complete subsets of R for ρ. The definition depends on auxiliary nice multiplicative rating maps. We first present them. Clearly, 2^R is an idempotent semiring (addition is union and the multiplication is lifted from the one of R). For every $S \subseteq R$, we use it as the rating set of a nice multiplicative rating map $\eta_{\rho,S} : 2^{A^*} \to 2^R$. Since we are defining a *nice* multiplicative rating map, it suffices to specify the evaluation of letters. For $a \in A$, we let $\eta_{\rho,S}(a) = S \cdot \{\rho(a)\} \cdot S \in 2^R$. Observe that by definition, we have $\mathbb{i}_{\mathcal{C}}[\eta_{\rho,S}] \subseteq R$.

We are ready to define the $SF(\mathcal{C})$-complete subsets of R. Consider $S \subseteq R$. We say that S is *$SF(\mathcal{C})$-complete for ρ* when the following conditions are satisfied:

1. **Downset.** For every $r \in S$ and $q \leq r$, we have $q \in S$.
2. **Multiplication.** For every $q, r \in S$, we have $qr \in S$.
3. **$\mathcal{C}$-operation.** We have $\mathbb{i}_{\mathcal{C}}[\eta_{\rho,S}] \subseteq S$.
4. **$SF(\mathcal{C})$-closure.** For every $r \in S$, we have $r^\omega + r^{\omega+1} \in S$.

▶ Remark 24. The definition of $SF(\mathcal{C})$-complete subsets does not explicitly require that they contain some trivial elements. Yet, this is implied by $\mathcal{C}$-operation. Indeed, if $S \subseteq R$ is $SF(\mathcal{C})$-complete, then $\eta_{\rho,S}(\varepsilon) = \{1_R\}$ (this is the multiplicative neutral element of 2^R). This implies that $1_R \in \mathbb{i}_{\mathcal{C}}[\eta_{\rho,S}]$ and we obtain from $\mathcal{C}$-operation that $1_R \in S$.

▶ **Theorem 25** ($SF(\mathcal{C})$-optimal imprints ($\mathcal{C}$ made of group languages))**.** *Let $\rho : 2^{A^*} \to R$ be a nice multiplicative rating map. Then, $\mathbb{i}_{\mathcal{C}}[\xi_{SF(\mathcal{C})}[\rho]]$ is the least $SF(\mathcal{C})$-complete subset of R.*

When $\mathcal{C}$-separation is decidable, Theorem 25 yields a least fixpoint procedure for computing $\mathbb{i}_{\mathcal{C}}[\xi_{SF(\mathcal{C})}[\rho]]$ from a nice multiplicative rating map $\rho : 2^{A^*} \to R$. The computation starts from the empty set and saturates it with the four operations in the definition of $SF(\mathcal{C})$-complete subsets. It is clear that we may implement downset, multiplication and $SF(\mathcal{C})$-closure. Moreover, we may implement $\mathcal{C}$-operation as this boils down to $\mathcal{C}$-separation by Lemma 21. Eventually, the computation reaches a fixpoint and it is straightforward to verify that this set is the least $SF(\mathcal{C})$-complete subset of R, *i.e.*, $\mathbb{i}_{\mathcal{C}}[\xi_{SF(\mathcal{C})}[\rho]]$ by Theorem 25.

By Proposition 22, we may compute $\mathcal{I}_{SF(\mathcal{C})}[\rho]$ from $\mathbb{i}_{\mathcal{C}}[\xi_{SF(\mathcal{C})}[\rho]]$. Altogether, this yields the decidability of $SF(\mathcal{C})$-covering by Proposition 19. Hence, Theorem 3 is proved.

6 Conclusion

We proved that for any quotient-closed Boolean algebra $\mathcal{C}$, $SF(\mathcal{C})$-covering is decidable whenever $\mathcal{C}$ is either finite or made of group languages and with decidable separation. Moreover, we presented an algebraic characterization of $SF(\mathcal{C})$ which holds for every quotient-closed Boolean algebra $\mathcal{C}$, generalizing earlier results [29, 16]. A key proof ingredient is an alternative definition of star-free closure: the operation $\mathcal{C} \mapsto SD(\mathcal{C})$ which we prove to be equivalent. This correspondence generalizes the work of Schützenberger [28] who introduced a single class SD (i.e. $SD(\{\emptyset, A^*\})$) corresponding to the star-free languages (i.e. $SF(\{\emptyset, A^*\})$).

Our results can be instantiated for several input classes $\mathcal{C}$. Theorem 2 applies when $\mathcal{C}$ is finite. In this case, the only prominent application is the class of star-free languages itself. It was already known that covering is decidable for this class [9, 20]. However, Theorem 2 is important for two reasons. First, its proof is actually simpler than the earlier ones specific to the star-free languages (this is achieved by relying on the operation $\mathcal{C} \mapsto SD(\mathcal{C})$). More importantly, Theorem 2 is used as a key ingredient for proving our second generic statement: Theorem 3, which applies to classes made of group languages with decidable separation. It is known that separation is decidable for the class GR of all group languages [2]. Hence, we obtain that $SF(\text{GR})$-covering is decidable. Another application is the class MOD consisting of languages counting the length of words modulo some number (deciding MOD-separation is a simple exercise). We get the decidability of $SF(\text{MOD})$-covering. This is important, as the languages in $SF(\text{MOD})$ are those definable in first-order logic with modular predicates ($\text{FO}(<, \text{MOD})$). A last example is given by the input class consisting of all languages counting the number of occurrences of letters modulo some number. These are exactly the languages recognized by finite commutative groups, for which separation is decidable [5].

References

1 Jorge Almeida. Some Algorithmic Problems for Pseudovarieties. *Publicationes Mathematicae Debrecen*, 54:531–552, 1999.

2 Christopher J. Ash. Inevitable Graphs: a Proof of the Type II Conjecture and some Related Decision Procedures. *IJAC*, 1(1):127–146, 1991.

3 David A. Mix Barrington, Kevin Compton, Howard Straubing, and Denis Thérien. Regular languages in NC1. *Journal of Computer and System Sciences*, 44(3):478–499, 1992.

4 Janusz A. Brzozowski and Rina S. Cohen. Dot-Depth of Star-Free Events. *J. Comp. Sys. Sci.*, 5(1):1–16, 1971.

5 Manuel Delgado. Abelian pointlikes of a monoid. *Semigroup Forum*, 56:339–361, 1998.

6 Volker Diekert and Tobias Walter. Characterizing classes of regular languages using prefix codes of bounded synchronization delay. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming*, ICALP'16, pages 129:1–129:14, 2016.

7 Volker Diekert and Tobias Walter. Characterizing classes of regular languages using prefix codes of bounded synchronization delay. *IJAC*, 27(6):561–590, 2017.

8 Samuel Eilenberg. *Automata, languages, and machines*, volume B. Academic Press, 1976.

9 Karsten Henckell. Pointlike sets: the finest aperiodic cover of a finite semigroup. *Journal of Pure Applied Algebra*, 55(1-2):85–126, 1988.

10 Joel Karnofsky and John Rhodes. Decidability of complexity one-half for finite semigroups. *Semigroup Forum*, 24(1):55–66, 1982. doi:10.1007/BF02572755.

11 Stuart W. Margolis and Jean-Éric Pin. Products of group languages. In *FCT'85*. Springer, 1985.

12 Robert McNaughton and Seymour A. Papert. *Counter-Free Automata.* MIT Press, 1971.

13 Jean-Éric Pin. Bridges for Concatenation Hierarchies. In *ICALP'98*, pages 431–442. Springer, 1998.

14 Jean-Éric Pin. The dot-depth hierarchy, 45 years later. In *The Role of Theory in Computer Science. Essays Dedicated to Janusz Brzozowski.* World Scientific Publishing, 2017.

15 Jean-Éric Pin. Mathematical Foundations of Automata Theory. Lecture notes, 2019. URL: https://www.irif.fr/~jep/PDF/MPRI/MPRI.pdf.

16 Jean-Éric Pin, Howard Straubing, and Denis Thérien. Locally trivial categories and unambiguous concatenation. *Journal of Pure and Applied Algebra*, 52(3):297–311, 1988.

17 Thomas Place. Separating Regular Languages with Two Quantifiers Alternations. In *LICS'15*, pages 202–213. IEEE Computer Society, 2015.

18 Thomas Place. Separating regular languages with two quantifier alternations. *Logical Methods in Computer Science*, 14(4), 2018.

19 Thomas Place and Marc Zeitoun. Going Higher in the First-Order Quantifier Alternation Hierarchy on Words. In *ICALP'14*, pages 342–353, 2014.

20 Thomas Place and Marc Zeitoun. Separating Regular Languages with First-Order Logic. *Logical Methods in Computer Science*, 12(1), 2016.

21 Thomas Place and Marc Zeitoun. Separation for Dot-Depth Two. In *32th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS'17, 2017. `arXiv:1901.03361`.

22 Thomas Place and Marc Zeitoun. Generic results for concatenation hierarchies. *Theory of Computing Systems (ToCS)*, 2018. Selected papers from CSR'17.

23 Thomas Place and Marc Zeitoun. Separation for dot-depth two. In preparation, preprint at `http://www.labri.fr/~zeitoun/research/pdf/boolpol-full.pdf`, 2018.

24 Thomas Place and Marc Zeitoun. The Covering Problem. *Logical Methods in Computer Science*, 14(3), 2018.

25 Thomas Place and Marc Zeitoun. On all things star-free. Full version of this paper, `arXiv:1904.11863`, 2019.

26 Thomas Place and Marc Zeitoun. Separation and covering for group based concatenation hierarchies. In *LICS'19*, 2019.

27 Marcel Paul Schützenberger. On Finite Monoids Having Only Trivial Subgroups. *Information and Control*, 8(2):190–194, 1965.

28 Marcel Paul Schützenberger. Sur certaines opérations de fermeture dans les langages rationnels. *Symposia Mathematica*, XV:245–253, 1975. Convegno di Informatica Teorica, INDAM, Roma, 1973.

29 Howard Straubing. Aperiodic homomorphisms and the concatenation product of recognizable sets. *Journal of Pure and Applied Algebra*, 15(3):319–327, 1979.

30 Wolfgang Thomas. Classifying Regular Events in Symbolic Logic. *J. Comp. Sys. Sci.*, 25(3):360–376, 1982.

31 Thomas Wilke. Classifying Discrete Temporal Properties. In *Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science*, STACS'99, pages 32–46, 1999.