View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Dagstuhl Research Online Publication Server

Separating Without Any Ambiguity

Thomas Place
LaBRI, University of Bordeaux and IUF, France

Marc Zeitoun
LaBRI, University of Bordeaux, France

Abstract

We investigate a standard operator on classes of languages: unambiguous polynomial closure. We show that if $\mathcal{C}$ is a class of regular languages having some mild properties, the membership problem for its unambiguous polynomial closure $UPol(\mathcal{C})$ reduces to the same problem for $\mathcal{C}$. We give a new, self-contained and elementary proof of this result. We also show that unambiguous polynomial closure coincides with alternating left and right deterministic closure. Finally, if additionally $\mathcal{C}$ is finite, we show that the separation and covering problems are decidable for $UPol(\mathcal{C})$.

2012 ACM Subject Classification Theory of computation → Regular languages

Keywords and phrases Regular languages, separation problem, decidable characterizations

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.137

Related Version A full version of the paper is available at `https://hal.archives-ouvertes.fr/hal-01798847`.

Funding Both authors acknowledge support from the DeLTA project (ANR-16-CE40-0007).

1 Introduction

Most of the interesting classes of regular languages are built using a restricted set of operators. From a class $\mathcal{C}$, one may consider its Boolean closure $Bool(\mathcal{C})$, its polynomial closure $Pol(\mathcal{C})$, and deterministic variants thereof, which yield usually a more elaborate class than $\mathcal{C}$. It is therefore desirable to investigate the operators themselves rather than individual classes.

The *polynomial closure* $Pol(\mathcal{C})$ of a class $\mathcal{C}$ is its closure under union and marked concatenation (a *marked concatenation* of K and L is a language of the form KaL for a letter a). Together with the Boolean closure, it is used to define concatenation hierarchies: starting from a given class (*level 0* in the hierarchy), level $n + \frac{1}{2}$ is the polynomial closure of level n, and level $n+1$ is the Boolean closure of level $n+\frac{1}{2}$. The importance of these hierarchies stems from the fact that they are the combinatorial counterpart of quantifier alternation hierarchies in logic, which count the number of $\forall/\exists$ alternations needed to define a language [29, 23].

The main question when investigating a class of languages is the *membership* problem: can we decide whether an input language belongs to the class? Despite decades of research on concatenation hierarchies, one knows little about it. The state of the art is that when level 0 is finite and has some mild properties, membership is decidable for levels $\frac{1}{2}$, 1, $\frac{3}{2}$, and $\frac{5}{2}$ [21, 19, 16, 17, 23]. These results encompass those that were obtained previously [3, 2, 26, 14, 15] and even go beyond by investigating the *separation problem*, a generalization of membership. This problem for a class $\mathcal{C}$ takes *two arbitrary regular* languages as input (unlike membership, which takes a single one). It asks whether there exists a third language from $\mathcal{C}$, containing the first and disjoint from the second. Membership is the special case of separation when the input consists of a language and its complement. Although more difficult than membership, separation is also more rewarding. This is witnessed by a transfer


© Thomas Place and Marc Zeitoun;
licensed under Creative Commons License CC-BY
45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 137; pp. 137:1–137:14


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

theorem [19, 23]: membership for $Pol(\mathcal{C})$ reduces to separation for $\mathcal{C}$. The above results on membership come from this theorem and the fact that separation is decidable for $Pol(\mathcal{C})$, $BPol(\mathcal{C}) \stackrel{\text{def}}{=} Bool(Pol(\mathcal{C}))$ and $Pol(BPol(\mathcal{C}))$ when $\mathcal{C}$ is finite with some mild properties. See [12, 23] for detailed surveys on concatenation hierarchies.

Unambiguous closure. Deterministic variants of the polynomial closure are also important. The most classical example is the unambiguous closure, where marked concatenations are required to be unambiguous. A marked concatenation KaL is *unambiguous* if every word w of KaL has a *unique* factorization $w = w'aw''$ with $w' \in K$ and $w'' \in L$. The *unambiguous closure* $UPol(\mathcal{C})$ of $\mathcal{C}$ is the closure of $\mathcal{C}$ under *disjoint* union and *unambiguous* concatenation. Note that it is not clear on the definition whether $UPol(\mathcal{C})$ is a Boolean algebra, even when $\mathcal{C}$ is.

A prominent example of a class built using unambiguous concatenation is that of *unambiguous languages* [25]. It is the unambiguous polynomial closure of the Boolean algebra generated by languages of the form A^*aA^*, where A is the working alphabet. Its robustness makes it one of the most investigated classes: it enjoys a number of equivalent definitions [27, 5, 7, 6, 28].

State of the art. The class $UPol(\mathcal{C})$ was described in algebraic terms in [13], following earlier work on deterministic products by Pin [9]. Note however that [9] starts from an alternate definition that assumes closure under Boolean operations already. Both papers use elaborate mathematical tools (categories, bilateral kernel, relational morphisms) as well as black boxes (results by Schützenberger [25] in [9] and a result of Rhodes [24] in [13]). Unambiguous polynomial closure also appears in concatenation hierarchies as *intermediate levels*: Pin and Weil [14, 15] have proved that $UPol(\mathcal{C}) = Pol(\mathcal{C}) \cap co\text{-}Pol(\mathcal{C})$, where $co\text{-}Pol(\mathcal{C})$ is the class consisting of all complements of languages in $Pol(\mathcal{C})$. Finally, a reduction from $UPol(\mathcal{C})$-membership to $\mathcal{C}$-membership was obtained in [1]. This proof is indirect: it is based on the nontrivial equality $UPol(\mathcal{C}) = Pol(\mathcal{C}) \cap co\text{-}Pol(\mathcal{C})$, which itself depends on the algebraic characterizations of $UPol(\mathcal{C})$ and $Pol(\mathcal{C})$ obtained in [13, 14, 15, 4].

Contributions. Unambiguous polynomial closure was not yet investigated with respect to separation, aside from the particular case of unambiguous languages [18]. This is the starting point of this paper: we look for a generic separation result applying to $UPol(\mathcal{C})$, similar to the ones obtained for $Pol(\mathcal{C})$ and $BPol(\mathcal{C})$ in [21]. This paper presents such a result: our main theorem states that when $\mathcal{C}$ is finite and satisfies some mild hypotheses, separation for $UPol(\mathcal{C})$ is decidable. However, as it is usual with separation, we also obtain several extra results as a byproduct of our work, improving our understanding of the $UPol$ operator:

- We had to rethink the way membership is classically handled for $UPol(\mathcal{C})$ in order to lift the techniques to separation. This yields a completely new, self-contained and elementary proof that under some natural hypothesis on $\mathcal{C}$, membership for $UPol(\mathcal{C})$ reduces to membership for $\mathcal{C}$. This proof also precisely pinpoints why this result holds for $UPol(\mathcal{C})$ but not $Pol(\mathcal{C})$. More precisely, we show that the languages from $\mathcal{C}$ needed to construct an $UPol(\mathcal{C})$ expression for a language L are all recognized by any recognizer of L.
- We obtain a new proof that $UPol(\mathcal{C})$ is a quotienting Boolean algebra when $\mathcal{C}$ is one.
- We obtain a new proof that $UPol(\mathcal{C}) = Pol(\mathcal{C}) \cap co\text{-}Pol(\mathcal{C})$ using our results on $Pol(\mathcal{C})$ [23].
- We obtain a previously unknown characterization of $UPol(\mathcal{C})$ in terms of alternating left and right deterministic concatenations, which are restricted forms of unambiguous concatenation. A marked concatenation KaL is *left (resp. right) deterministic* when $KaA^* \cap K = \emptyset$ (resp. $A^*aL \cap L = \emptyset$). We prove that $UPol(\mathcal{C})$ coincides with $ADet(\mathcal{C})$, the closure of $\mathcal{C}$ under disjoint union and left and right deterministic concatenation.

Organization of the paper. Section 2 sets up the notation and the terminology. Section 3 presents a solution of the membership problem for $UPol(\mathcal{C})$ when $\mathcal{C}$ has mild closure properties. This result also yields the above byproducts. Finally, in Section 4, we present an algorithm for solving separation for $UPol(\mathcal{C})$ when $\mathcal{C}$ is additionally finite.

2 Preliminaries

Words and languages. For the whole paper, we fix an arbitrary finite alphabet A. We denote by A^* the set of all finite words over A, and by $\varepsilon \in A^*$ the empty word. Given two words $u, v \in A^*$, we write uv for their concatenation. A *language (over A)* is a subset of A^*. Abusing terminology, we denote by u the singleton language $\{u\}$. It is standard to extend concatenation to languages: given $K, L \subseteq A^*$, we write $KL = \{uv \mid u \in K \text{ and } v \in L\}$. Moreover, we also consider marked concatenation, which is less standard. Given $K, L \subseteq A^*$, *a marked concatenation* of K with L is a language of the form KaL, for some $a \in A$.

A class of languages $\mathcal{C}$ is a set of languages. We say that $\mathcal{C}$ is a *lattice* when $\emptyset \in \mathcal{C}$, $A^* \in \mathcal{C}$ and $\mathcal{C}$ is closed under finite union and finite intersection: for any $K, L \in \mathcal{C}$, we have $K \cup L \in \mathcal{C}$ and $K \cap L \in \mathcal{C}$. Moreover, a *Boolean algebra* is a lattice $\mathcal{C}$ which is additionally closed under complement: for any $L \in \mathcal{C}$, we have $A^* \setminus L \in \mathcal{C}$. Finally, a class $\mathcal{C}$ is *quotienting* if it is closed under quotients. That is, for any $L \in \mathcal{C}$ and any word $u \in A^*$, the following properties hold:

$$u^{-1}L \stackrel{\text{def}}{=} \{w \in A^* \mid uw \in L\} \quad \text{and} \quad Lu^{-1} \stackrel{\text{def}}{=} \{w \in A^* \mid wu \in L\} \quad \text{both belong to } \mathcal{C}.$$

All classes that we consider are quotienting Boolean algebras of regular languages.

Regular languages. These are the languages that can be equivalently defined by nondeterministic finite automata, finite monoids or monadic second-order logic. In the paper, we work with the definition by monoids, which we recall now.

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, *i.e.*, such that $1_M \cdot s = s \cdot 1_M = s$ for every $s \in M$. 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 for any $s \in M$, the element s^ω is an idempotent.

Our proofs make use of the Green relations [8], which are defined on monoids (we use them as induction parameters). We briefly recall them. Given a monoid M and $s, t \in M$,

$$\begin{array}{ll} s \leqslant_{\mathcal{J}} t & \text{when there exist } x, y \in M \text{ such that } s = xty, \\ s \leqslant_{\mathcal{L}} t & \text{when there exists } x \in M \text{ such that } s = xt, \\ s \leqslant_{\mathcal{R}} t & \text{when there exists } y \in M \text{ such that } s = ty. \end{array}$$

Clearly, $\leqslant_{\mathcal{J}}$, $\leqslant_{\mathcal{L}}$ and $\leqslant_{\mathcal{R}}$ are preorders (*i.e.*, they are reflexive and transitive). We write $<_{\mathcal{J}}$, $<_{\mathcal{L}}$ and $<_{\mathcal{R}}$ for their strict variants (for example, $s <_{\mathcal{J}} t$ when $s \leqslant_{\mathcal{J}} t$ but $t \not\leqslant_{\mathcal{J}} s$). Finally, we write $\mathcal{J}$, $\mathcal{L}$ and $\mathcal{R}$ for the corresponding equivalence relations (for example, $s \mathcal{J} t$ when $s \leqslant_{\mathcal{J}} t$ and $t \leqslant_{\mathcal{J}} s$). There are many technical results about Green relations. We shall only need the following simple lemma which applies to *finite* monoids (see [11]).

▶ **Lemma 1.** *Consider a finite monoid M and $s, t \in M$ such that $s \mathcal{J} t$. Then, $s \leqslant_{\mathcal{R}} t$ implies $s \mathcal{R} t$. Symmetrically, $s \leqslant_{\mathcal{L}} t$ implies $s \mathcal{L} t$.*

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 some language $L \subseteq A^*$, we say that L is *recognized* by α when there exists a set $F \subseteq M$ such that $L = \alpha^{-1}(F)$.

Given any language L, there exists a canonical morphism which recognizes it. Let us briefly recall its definition. One may associate to L an equivalence $\equiv_L$ over A^*: the *syntactic congruence of* L. Given $u, v \in A^*$, $u \equiv_L v$ if and only if $xuy \in L \Leftrightarrow xvy \in L$ for any $x, y \in A^*$. It is known and simple to verify that "$\equiv_L$" is a congruence on A^*. Thus, the set of equivalence classes $M_L = A^*/\equiv_L$ is a monoid and the map $\alpha_L : A^* \to M_L$ sending any word to its equivalence class is a morphism recognizing L, called the *syntactic morphism of* L. Finally, it is known that L is regular if and only if M_L is finite (*i.e.*, $\equiv_L$ has finite index): this is Myhill-Nerode theorem. In that case, one may compute the syntactic morphism $\alpha_L : A^* \to M_L$ from any representation of L (such as a finite automaton).

Decision problems. The two problems that we consider in the paper are both parametrized by an arbitrary class of languages $\mathcal{C}$: they serve as mathematical tools for analyzing $\mathcal{C}$. The *$\mathcal{C}$-membership problem* is the simplest one. It takes as input a single regular language L and asks whether $L \in \mathcal{C}$. The second one, *$\mathcal{C}$-separation*, is more general: it takes **two** regular languages L_1, L_2 as input and asks whether L_1 is $\mathcal{C}$-separable from L_2, that is, whether there exists $K \in \mathcal{C}$ such that $L_1 \subseteq K$ and $L_2 \cap K = \emptyset$. The language K is called a *separator* of L_1 and L_2. Note that $\mathcal{C}$-membership is easily reduced to $\mathcal{C}$-separation: given any regular language L, we have $L \in \mathcal{C}$ if and only if L is $\mathcal{C}$-separable from $A^* \setminus L$ (which is also regular).

3 Unambiguous polynomial closure

In this section, we define the unambiguous polynomial closure operation, which is the main focus of the paper. Furthermore, we investigate the associated membership problem.

3.1 Definition

Given two languages $H, L \subseteq A^*$, we say that their concatenation HL is *unambiguous* when any word $w \in HL$ admits a *unique* decomposition witnessing this membership: for any $u, u' \in H$ and $v, v' \in L$, if $w = uv = u'v'$, then $u = u'$ and $v = v'$. More generally, we say that a product of n languages $L_1 \cdots L_n$ is *unambiguous* when any word $w \in L_1 \cdots L_n$ admits a *unique* decomposition witnessing this membership. Note that unambiguous *marked* concatenations are well-defined: HaL is a product of three languages, namely H, $\{a\}$ and L.

▶ Remark. Clearly, not all products are unambiguous. For example, A^*aA^* is ambiguous: $aa \in A^*aA^*$ admits two decompositions witnessing this membership (εaa and $aa\varepsilon$).

▶ Remark. Being unambiguous is a *semantic* property: whether HL is unambiguous may not be apparent on the definitions of H and L. Moreover, this depends on the *product* HL and not only on the resulting language $K = HL$. It may happen that two products represent the same language but one is unambiguous while the other is not. For example, A^*aA^* is ambiguous while $(A \setminus \{a\})^*aA^*$ (which represents the same language) is unambiguous.

In the paper, we shall only need two special kinds of unambiguous products, which we now present. Let $K, L \subseteq A^*$ and $a \in A$. We say that the marked concatenation KaL,

- is *left deterministic* when $K \cap KaA^* = \emptyset$,
- is *right deterministic* when $L \cap A^*aL = \emptyset$.

▶ **Fact 2.** *Any left or right deterministic marked concatenation is unambiguous.*

We use these definitions to introduce three standard operations on classes of languages. Consider an arbitrary class $\mathcal{C}$.

- The *polynomial closure* of $\mathcal{C}$, denoted by $Pol(\mathcal{C})$, is the smallest class containing $\mathcal{C}$ and closed under marked concatenation and union: for any $H, L \in Pol(\mathcal{C})$ and $a \in A$, we have $HaL \in Pol(\mathcal{C})$ and $H \cup L \in Pol(\mathcal{C})$. Furthermore, we denote by *co-Pol*$(\mathcal{C})$ the class containing all complements of languages in $Pol(\mathcal{C})$: $L \in$ *co-Pol*$(\mathcal{C})$ when $A^* \setminus L \in Pol(\mathcal{C})$.
- The *unambiguous polynomial closure* of $\mathcal{C}$, denoted by $UPol(\mathcal{C})$, is the smallest class containing $\mathcal{C}$ and closed under *unambiguous* marked concatenation and *disjoint* union. That is, for any $H, L \in UPol(\mathcal{C})$ and $a \in A$, if HaL is unambiguous, then $HaL \in UPol(\mathcal{C})$ and if $H \cap L = \emptyset$, then $H \uplus L \in UPol(\mathcal{C})$. Here, we denote union by "$\uplus$" to underline the fact that H and L are disjoint (we use this convention in the whole paper).
- The *alternating deterministic closure* of $\mathcal{C}$, denoted by $ADet(\mathcal{C})$, is the smallest class containing $\mathcal{C}$ and closed under *deterministic* marked concatenation and *disjoint* union. That is, for any $H, L \in ADet(\mathcal{C})$ and $a \in A$, if HaL is either left or right deterministic, then $HaL \in ADet(\mathcal{C})$ and if $H \cap L = \emptyset$, then $H \uplus L \in ADet(\mathcal{C})$.

It is immediate by definition and Fact 2 that we have $\mathcal{C} \subseteq ADet(\mathcal{C}) \subseteq UPol(\mathcal{C}) \subseteq Pol(\mathcal{C})$. In general the inclusion $UPol(\mathcal{C}) \subseteq Pol(\mathcal{C})$ is strict. On the other hand, we shall prove that when $\mathcal{C}$ is a quotienting Boolean algebra, $ADet(\mathcal{C}) = UPol(\mathcal{C})$.

It is not immediate that $Pol(\mathcal{C})$, $UPol(\mathcal{C})$ and $ADet(\mathcal{C})$ have robust closure properties beyond those explicitly stated in the definitions. However, it turns out that when $\mathcal{C}$ satisfies robust properties itself, this is the case for these three classes as well. It was shown by Arfi [3] that when $\mathcal{C}$ is a quotienting Boolean algebra of regular languages, then $Pol(\mathcal{C})$ is a quotienting lattice. Pin [10] extended the result for the case when $\mathcal{C}$ is a quotienting lattice. Here, we are mostly interested in $UPol(\mathcal{C})$. We prove the following theorem which combines and extends several results by Pin, Straubing, Thérien and Weil [13, 15].

▶ **Theorem 3.** *Let $\mathcal{C}$ be a quotienting Boolean algebra of regular languages. Then, $UPol(\mathcal{C})$ is a quotienting Boolean algebra as well. Moreover, $UPol(\mathcal{C}) = ADet(\mathcal{C}) = Pol(\mathcal{C}) \cap$ co-$Pol(\mathcal{C})$.*

That $UPol(\mathcal{C})$ is a quotienting Boolean algebra of regular languages is due to Pin, Straubing and Thérien [13]. The correspondence between $UPol(\mathcal{C})$ and $Pol(\mathcal{C}) \cap$ *co-Pol*$(\mathcal{C})$ is due to Pin and Weil [15]. The correspondence between $UPol(\mathcal{C})$ and $ADet(\mathcal{C})$ is a new result, to the best of our knowledge. Let us point out that the original proofs of these results require a stronger hypothesis on $\mathcal{C}$, which needs additionally to be closed under inverse morphic image. Moreover, these proofs require to introduce and manipulate a lot of algebraic machinery. This is because they are based on a generic algebraic characterization of $UPol(\mathcal{C})$.

While we use a similar approach (*i.e.*, we prove a generic algebraic characterization of $UPol(\mathcal{C})$), our argument is much more elementary. The only algebraic notion that we need is the syntactic morphism of a regular language.

3.2 Algebraic characterization

We now present a generic algebraic characterization of $UPol(\mathcal{C})$. It holds provided that $\mathcal{C}$ is a quotienting Boolean algebra of regular languages. It implies Theorem 3, but also that $UPol(\mathcal{C})$-membership reduces to $\mathcal{C}$-membership.

The characterization is parameterized by two relations that we define now. Let $\mathcal{C}$ be some class of languages. Consider a finite monoid M and a *surjective* morphism $\alpha : A^* \to M$ (such as the syntactic morphism of some language). Given a pair $(s, t) \in M \times M$,

- (s, t) is a *$\mathcal{C}$-pair* (for α) when **no** language of $\mathcal{C}$ can separate $\alpha^{-1}(s)$ from $\alpha^{-1}(t)$.
- (s, t) is a *weak $\mathcal{C}$-pair* (for α) when **no** language of $\mathcal{C}$ **recognized by α** can separate $\alpha^{-1}(s)$ from $\alpha^{-1}(t)$.

Note that any $\mathcal{C}$-pair is also a weak $\mathcal{C}$-pair (the converse is not true in general). By definition, we are able to compute all $\mathcal{C}$-pairs as soon as we have an algorithm for $\mathcal{C}$-separation. On the other hand, computing all weak $\mathcal{C}$-pairs boils down to deciding $\mathcal{C}$-membership, as it suffices to check which languages recognized by α (the potential separators) belong to $\mathcal{C}$.

▶ Remark. An equivalent definition of the weak $\mathcal{C}$-pairs is to introduce them as the transitive closure of the $\mathcal{C}$-pairs. We prove this in the full version. In fact, when $\mathcal{C}$ is a quotienting Boolean algebra, the weak $\mathcal{C}$-pair relation is a congruence whose equivalence classes correspond exactly to the languages recognized by α and belonging to $\mathcal{C}$.

We may now state the following characterization of $UPol(\mathcal{C})$.

▶ **Theorem 4.** *Let $\mathcal{C}$ be a quotienting Boolean algebra of regular languages. Consider a regular language L and let $\alpha : A^* \to M$ be its syntactic morphism. The following are equivalent:*

1. $L \in UPol(\mathcal{C})$.
2. $L \in ADet(\mathcal{C})$.
3. $L \in Pol(\mathcal{C}) \cap co\text{-}Pol(\mathcal{C})$.
4. *For all $\mathcal{C}$-pairs $(s,t) \in M^2$, we have $s^{\omega+1} = s^\omega t s^\omega$.*
5. *For all weak $\mathcal{C}$-pairs $(s,t) \in M^2$, we have $s^{\omega+1} = s^\omega t s^\omega$.*

Theorem 3 is a simple corollary of Theorem 4 (it is standard that any class satisfying a property such as Item (4) in the theorem is a quotienting Boolean algebra). Another consequence is that if $\mathcal{C}$ is a quotienting Boolean algebra of regular languages, $UPol(\mathcal{C})$-membership reduces to the same problem for $\mathcal{C}$. Indeed, given as input a regular language L, one may compute its syntactic morphism α. By Theorem 4, deciding whether $L \in UPol(\mathcal{C})$ amounts to checking whether α satisfies Item (5). This is possible provided that we have all weak $\mathcal{C}$-pairs for α in hand. In turn, an algorithm for $\mathcal{C}$-membership immediately yields an algorithm for computing them all. Altogether, we obtain the following corollary.

▶ **Corollary 5.** *Let $\mathcal{C}$ be a quotienting Boolean algebra of regular languages and assume that $\mathcal{C}$-membership is decidable. Then $UPol(\mathcal{C})$-membership is decidable as well.*

We now focus on proving Theorem 4. A first point is that we do not show the equivalence (3) $\Leftrightarrow$ (4): it is a simple corollary of the generic characterization of $Pol(\mathcal{C})$ which is not our main focus in the paper (a full proof is available in [23]). Here, we concentrate on proving the implications (1) $\Rightarrow$ (4) $\Rightarrow$ (5) $\Rightarrow$ (2) $\Rightarrow$ (1). The implication (2) $\Rightarrow$ (1) ($ADet(\mathcal{C}) \subseteq UPol(\mathcal{C})$) is immediate. Even though the presentation is different, the equivalence (4) $\Leftrightarrow$ (5) is a result of [1] (which investigates $Pol(\mathcal{C}) \cap co\text{-}Pol(\mathcal{C})$) and is based on algebraic manipulations. We prove this equivalence as well as the implication (1) $\Rightarrow$ (4) in the full version, to focus on (5) $\Rightarrow$ (2), which is the most interesting implication: when a language satisfies (5), we show that it belongs to $ADet(\mathcal{C})$.

We fix a quotienting Boolean algebra of regular languages $\mathcal{C}$ for the proof. Consider an arbitrary surjective morphism $\alpha : A^* \to M$ satisfying Item (5) in Theorem 4. We show that any language recognized by α belongs to $ADet(\mathcal{C})$. We start with a preliminary lemma.

▶ **Lemma 6.** *There exists a finite monoid N and a surjective morphism $\beta : M \to N$ which satisfies the following properties:*

- *For any $s,t \in M$, (s,t) is a weak $\mathcal{C}$-pair if and only if $\beta(s) = \beta(t)$.*
- *Any language recognized by the composition $\gamma = \beta \circ \alpha : A^* \to N$ belongs to $\mathcal{C}$.*

Lemma 6 is obtained by proving that the weak $\mathcal{C}$-pair relation is a congruence on M and that for any equivalence class $F \subseteq M$, we have $\alpha^{-1}(F) \in \mathcal{C}$. It then suffices to define N as the quotient of M by this congruence. The proof is presented in the full version of the paper.

Let us come back to the main proof. Let $\beta : M \to N$ and the composition $\gamma = \beta \circ \alpha$ be defined as in Lemma 6. Given any $r_1, r_2, s \in M$ and any $x \in N$, we define:

$$L_s^x[r_1, r_2] = \{w \in \gamma^{-1}(x) \mid r_1\alpha(w)r_2 = s\}.$$

The purpose of introducing $L_s^x[r_1, r_2]$ is that it provides induction parameters s, r_1, r_2 and it coincides with $\alpha^{-1}(s)$ when $x = \beta(s)$, $r_1 = r_2 = 1_M$. Our goal is to show that it is in $ADet(\mathcal{C})$.

▶ **Proposition 7.** *Let $r_1, r_2, s \in M$ and $x \in N$. Then, $L_s^x[r_1, r_2] \in ADet(\mathcal{C})$.*

Before proving this proposition, let us use it to finish the main proof. By definition, a language recognized by α is a disjoint union of sets $\alpha^{-1}(s)$ for $s \in M$. Therefore, it suffices to prove that $\alpha^{-1}(s) \in ADet(\mathcal{C})$ for any $s \in M$. Let $x = \beta(s)$. Clearly, $L_s^{\beta(s)}[1_M, 1_M] = \alpha^{-1}(s)$. Thus, Proposition 7 yields that $\alpha^{-1}(s) \in ADet(\mathcal{C})$, finishing the proof.

It remains to prove Proposition 7. We let $r_1, r_2, s \in M$ and $x \in N$. Our objective is to show that $L_s^x[r_1, r_2] \in ADet(\mathcal{C})$. Observe that we may assume without loss of generality that $\beta(s) = \beta(r_1)x\beta(r_2)$. Otherwise, $L_s^x[r_1, r_2] = \emptyset \in ADet(\mathcal{C})$ by definition and the result is immediate. The proof is an induction on the three following parameters listed by order of importance (the three of them depend on Green's relations in both M and N):

1. The *rank of* $\beta(s)$ which is the number of elements $y \in N$ such that $\beta(s) \leqslant_{\mathcal{J}} y$.
2. The *right index of* r_1 which is the number of elements $t \in M$ such that $t \leqslant_{\mathcal{R}} r_1$.
3. The *left index of* r_2 which is the number of elements $t \in M$ such that $t \leqslant_{\mathcal{L}} r_2$.

We consider three cases depending on the following properties of s, r_1, r_2 and x.

- We say that x is *smooth* when $x \mathrel{\mathcal{J}} \beta(s)$.
- We say that r_1 is *right stable* when there exists $t \in M$ such that $\beta(t) \mathrel{\mathcal{R}} x$ and $r_1 t \mathrel{\mathcal{R}} r_1$.
- We say that r_2 is *left stable* when there exists $t \in M$ such that $\beta(t) \mathrel{\mathcal{L}} x$ and $tr_2 \mathrel{\mathcal{L}} r_2$.

In the base case, we assume that all three properties hold. Otherwise, we consider two inductive cases. First, we assume that x is not smooth. Then, we assume that either r_1 is not right stable or r_2 is not left stable.

Base case. Assume that x is smooth and that r_1, r_2 are respectively right and left stable. We use this hypothesis to prove the following lemma.

▶ **Lemma 8.** *For any $u, v \in \gamma^{-1}(x)$, we have $r_1\alpha(u)r_2 = r_1\alpha(v)r_2$.*

Observe that Lemma 8 concludes the proof. Indeed, by definition of $L_s^x[r_1, r_2]$, it implies that either $L_s^x[r_1, r_2] = \gamma^{-1}(x)$ (when $r_1\alpha(w)r_2 = s$ for all $w \in \gamma^{-1}(x)$) or $L_s^x[r_1, r_2] = \emptyset$ (when $r_1\alpha(w)r_2 \neq s$ for all $w \in \gamma^{-1}(x)$). Since both of these languages belong to $\mathcal{C} \subseteq ADet(\mathcal{C})$ by Lemma 6, Proposition 7 follows. It remains to prove Lemma 8 to conclude the base case. The argument relies on the following fact (this is where we use our hypothesis on r_1 and r_2).

▶ **Fact 9.** *When Item (5) in Theorem 4 holds, the two following properties hold as well:*

- *For all $t \in M$ such that $\beta(t) \mathrel{\mathcal{R}} x$, we have $r_1 t \mathrel{\mathcal{R}} r_1$.*
- *For all $t \in M$ such that $\beta(t) \mathrel{\mathcal{L}} x$, we have $tr_2 \mathrel{\mathcal{L}} r_2$.*

Let us first use the fact to prove Lemma 8 and finish the base case. Consider $u, v \in \gamma^{-1}(x)$, *i.e.*, $\beta(\alpha(u)) = \beta(\alpha(v)) = x$. We show that $r_1\alpha(u)r_2 = r_1\alpha(v)r_2$.

By hypothesis, we have $\beta(s) = \beta(r_1)x\beta(r_2)$. Moreover, $\beta(s) \mathrel{\mathcal{J}} x$ since x is smooth by hypothesis. Thus, $x\beta(r_2) \mathrel{\mathcal{J}} x$ and $\beta(r_1)x \mathrel{\mathcal{J}} x$. Hence, since $x\beta(r_2) \leqslant_{\mathcal{R}} x$ and $\beta(r_1)x \leqslant_{\mathcal{L}} x$, Lemma 1 implies $x\beta(r_2) \mathrel{\mathcal{R}} x$ and $\beta(r_1)x \mathrel{\mathcal{L}} x$. Since $\beta(\alpha(u)) = x$, this yields $\beta(\alpha(u)r_2) \mathrel{\mathcal{R}} x$ and $\beta(r_1\alpha(u)) \mathrel{\mathcal{L}} x$. Applying Fact 9 with $t = \alpha(u)r_2$, one gets $r_1\alpha(u)r_2 \mathrel{\mathcal{R}} r_1$ and $r_1\alpha(u)r_2 \mathrel{\mathcal{L}} r_2$.

We get $p, q \in M$ such that $r_1 = r_1\alpha(u)r_2p$ and $r_2 = qr_1\alpha(u)r_2$. Let $t = qr_1\alpha(u)r_2p = r_2p = qr_1$. We combine our two equalities for r_1 and r_2 to obtain,

$$r_1 = r_1\alpha(u)t = r_1(\alpha(u)t)^\omega \quad \text{and} \quad r_2 = t\alpha(u)r_2 = (t\alpha(u))^{\omega+1}r_2. \tag{1}$$

Since $\beta(\alpha(u)) = \beta(\alpha(v))$, we know that $\beta(\alpha(u)t) = \beta(\alpha(v)t)$. Therefore, $(\alpha(u)t, \alpha(v)t)$ is a weak $\mathcal{C}$-pair by Lemma 6, and Item (5) yields $(\alpha(u)t)^{\omega+1} = (\alpha(u)t)^\omega\alpha(v)t(\alpha(u)t)^\omega$. We may now multiply by r_1 on the left and by $\alpha(u)r_2$ on the right to get,

$$r_1(\alpha(u)t)^\omega\alpha(u)(t\alpha(u))^{\omega+1}r_2 = r_1(\alpha(u)t)^\omega\alpha(v)(t\alpha(u))^{\omega+1}r_2.$$

Since we already know from (1) that $r_1 = r_1(\alpha(u)t)^\omega$ and $r_2 = (t\alpha(u))^{\omega+1}r_2$, we get as desired that $r_1\alpha(u)r_2 = r_1\alpha(v)r_2$, finishing the proof of Lemma 8. It remains to prove Fact 9.

Proof of Fact 9. By symmetry, we focus on the first property and leave the second to the reader. Let $t \in M$ such that $\beta(t) \mathrel{\mathcal{R}} x$. We show that $r_1t \mathrel{\mathcal{R}} r_1$. By hypothesis, r_1 is right stable which yields $t' \in M$ such that $\beta(t') \mathrel{\mathcal{R}} x \mathrel{\mathcal{R}} \beta(t)$ and $r_1t' \mathrel{\mathcal{R}} r_1$. Since $\beta(t') \mathrel{\mathcal{R}} \beta(t)$, we have $y \in N$ such that $\beta(t') = \beta(t)y$. Let $p \in M$ such that $\beta(p) = y$: we have $\beta(t') = \beta(tp)$. Since $r_1t' \mathrel{\mathcal{R}} r_1$, we have $q \in M$ such that $r_1 = r_1t'q$ which yields $r_1 = r_1(t'q)^\omega = r_1(t'q)^{\omega+1}$. We have $\beta(t'q) = \beta(tpq)$ which means that $(t'q, tpq)$ is a weak $\mathcal{C}$-pair by Lemma 6. Therefore, Equation (5) yields that $(t'q)^{\omega+1} = (t'q)^\omega tpq(t'q)^\omega$. Finally, we obtain,

$$r_1 = r_1(t'q)^{\omega+1} = r_1(t'q)^\omega tpq(t'q)^\omega = r_1tpq(t'q)^\omega.$$

This implies that $r_1 \leqslant_{\mathcal{R}} r_1t$. Since it is immediate that $r_1t \leqslant_{\mathcal{R}} r_1$, we get $r_1t \mathrel{\mathcal{R}} r_1$. ◀

First inductive case. We now assume that x is not smooth: x and $\beta(s)$ are not $\mathcal{J}$-equivalent. We use induction on our first parameter (the rank of $\beta(s)$). Recall that we assumed $\beta(s) = \beta(r_1)x\beta(r_2)$, which yields $\beta(s) \leqslant_{\mathcal{J}} x$. Thus, we have $\beta(s) <_{\mathcal{J}} x$ by hypothesis.

By definition, $L_s^x[r_1, r_2]$ is the disjoint union of all languages $\alpha^{-1}(t)$ where $t \in M$ satisfies $\beta(t) = x$ and $r_1tr_2 = s$. Therefore, it suffices to show that for any $t \in M$ such that $\beta(t) = x$, we have $\alpha^{-1}(t) \in ADet(\mathcal{C})$. This is immediate by induction. Indeed, since $\beta(t) = x$, we have $\alpha^{-1}(t) = L_t^x[1_M, 1_M]$. Moreover, since $\beta(s) <_{\mathcal{J}} x$, we have $\beta(s) <_{\mathcal{J}} \beta(t)$. It follows that the rank of $\beta(t)$ is strictly smaller than the one of $\beta(s)$. Hence, we may apply induction on our first and most important parameter to get $L_t^x[1_M, 1_M] \in ADet(\mathcal{C})$.

Second inductive case. We assume that either r_1 is not *right stable* or r_2 is not *left stable*. By symmetry, we treat the case when r_1 is not right stable and leave the other to the reader.

▶ Remark. We only apply induction on our two first parameters. Moreover, we show that $L_s^x[r_1, r_2]$ is built from languages in $ADet(\mathcal{C})$ (obtained from induction) using only disjoint union and left deterministic marked concatenations. Induction on our third parameter and right deterministic marked concatenations are used in the case when r_2 is not left stable.

Observe that we have $x <_{\mathcal{J}} 1_N$ (x is not maximal for $\leqslant_{\mathcal{J}}$). Indeed, otherwise, we would have $x \mathrel{\mathcal{R}} 1_N$ by Lemma 1 and r_1 would be left stable: $1_M \in M$ would satisfy $\beta(1_M) = 1_N \mathrel{\mathcal{R}} x$ and $r_11_M = r_1 \mathrel{\mathcal{R}} r_1$. Therefore, there are elements $y \in N$ such that $x <_{\mathcal{J}} y$.

We use this observation to define T as the set of all triples $(y, a, z) \in N \times A \times N$ such that $x = y\gamma(a)z$, $x <_{\mathcal{J}} y$ and $x \mathrel{\mathcal{J}} y\gamma(a)$. Using the definition of T and the fact that $x <_{\mathcal{J}} 1_N$, one may decompose $L_s^x[r_1, r_2]$ as follows (this lemma is proved in the full version).

▶ **Lemma 10.** *The language $L_s^x[r_1, r_2]$ is equal to the following disjoint union,*

$$L_s^x[r_1, r_2] = \biguplus_{(y,a,z)\in T} \left(\biguplus_{t\in\beta^{-1}(y)} \alpha^{-1}(t) \cdot a \cdot L_s^z[r_1 t\alpha(a), r_2] \right).$$

We now use Lemma 10 to show as desired that $L_s^x[r_1, r_2] \in ADet(\mathcal{C})$. Since $ADet(\mathcal{C})$ is closed under disjoint union by definition, it suffices to show that for any $(y, a, z) \in T$ and any $t \in \beta^{-1}(y)$, we have,

$$\alpha^{-1}(t) \cdot a \cdot L_s^z[r_1 t\alpha(a), r_2] \in ADet(\mathcal{C}).$$

We prove that this is a left deterministic marked concatenation of two languages in $ADet(\mathcal{C})$ which concludes the proof.

We start with $\alpha^{-1}(t) \in ADet(\mathcal{C})$. Since $y = \beta(t)$, we have $\alpha^{-1}(t) = L_t^y[1_M, 1_M]$. Moreover, since $\beta(s) = \beta(r_1)x\beta(r_2)$, we have $\beta(s) \leqslant_{\mathcal{J}} x$. Finally, by definition of T we have $x <_{\mathcal{J}} y = \beta(t)$. Altogether, we get $\beta(s) <_{\mathcal{J}} \beta(t)$: the rank of $\beta(t)$ is strictly smaller than the one of $\beta(s)$ and induction on our first parameter yields $\alpha^{-1}(t) = L_t^y[1_M, 1_M] \in ADet(\mathcal{C})$.

We turn to $L_s^z[r_1 t\alpha(a), r_2] \in ADet(\mathcal{C})$. By definition of T, we have $x \mathcal{J} y\gamma(a)$ and $x = y\gamma(a)z$ which yields that $x \mathcal{R} y\gamma(a)$ by Lemma 1. Moreover, since $y = \beta(t)$, it follows that $x \mathcal{R} \beta(t\alpha(a))$. Therefore, since we know that r_1 is **not** right stable (this is our hypothesis), it follows that r_1 and $r_1 t\alpha(a)$ are not $\mathcal{R}$-equivalent. Since it is clear that $r_1 t\alpha(a) \leqslant_{\mathcal{R}} r_1$, it follows that $r_1 t\alpha(a) <_{\mathcal{R}} r_1$: the right index of $r_1 t\alpha(a)$ is strictly smaller than the one of r_1. By induction on our second parameter, we then get that $L_s^z[r_1 t\alpha(a), r_2] \in ADet(\mathcal{C})$.

It remains to show that $\alpha^{-1}(t) \cdot a \cdot L_s^z[r_1 t\alpha(a), r_2]$ is a left deterministic marked concatenation, *i.e.*, that $\alpha^{-1}(t) \cap \alpha^{-1}(t)aA^* = \emptyset$. Since $\beta(t) = y$, we have $\alpha^{-1}(t) \subseteq \gamma^{-1}(y)$ and it suffices to show that $\gamma^{-1}(y) \cap \gamma^{-1}(y)aA^* = \emptyset$. Let $w \in \gamma^{-1}(y)$ and $w' \in \gamma^{-1}(y)aA^*$, we show that $w \neq w'$. Since $(x, a, z) \in T$, we have $x <_{\mathcal{J}} y$ and $x \mathcal{J} y\gamma(a)$. It follows that $y\gamma(a) <_{\mathcal{J}} y$. Finally, we have $\gamma(w) = y$ and $\gamma(w') = y\gamma(a)y'$ for some $y' \in N$. This implies that $\gamma(w') \leqslant_{\mathcal{J}} y\gamma(a) <_{\mathcal{J}} y = \gamma(w)$. Therefore $\gamma(w) \neq \gamma(w')$ which implies that $w \neq w'$.

4 Separation

We now turn to separation for $UPol(\mathcal{C})$ and show that the problem is decidable for any finite quotienting Boolean algebra $\mathcal{C}$. For the sake of avoiding clutter, we fix $\mathcal{C}$ for the section.

▶ Remark. This result may seem weak: our solution for $UPol(\mathcal{C})$-separation requires $\mathcal{C}$ to be finite while $UPol(\mathcal{C})$-membership reduces to $\mathcal{C}$-membership. This intuition is wrong: the result on separation is the strongest. The proof of Theorem 4 shows that when $L \in UPol(\mathcal{C})$, the basic languages in $\mathcal{C}$ needed to build L are all recognized by the syntactic morphism of L. Hence, $L \in UPol(\mathcal{C})$ if and only if $L \in UPol(\mathcal{D})$ where $\mathcal{D} \subseteq \mathcal{C}$ is a *finite class* obtained from the syntactic morphism of L. We lose this when moving to separation: the languages in $\mathcal{C}$ needed to build a potential separator in $UPol(\mathcal{C})$ may not be encoded in our two inputs.

Our algorithm is based on a general framework designed to handle separation problems and to present solutions in an elegant way. It was introduced in [20, 22]. We first summarize what we need in this framework to present our solution for $UPol(\mathcal{C})$-separation.

▶ Remark. The framework of [20, 22] is actually designed to handle a more general decision problem: covering, which generalizes separation to arbitrarily many input languages. Thus, our solution actually yields an algorithm for $UPol(\mathcal{C})$-covering as well. While we do not detail this point due to lack of space, this follows from the definitions of [20, 22].

4.1 Methodology

We briefly recall the framework of [20, 22]. We refer the reader to [22] for details. The approach is based on "rating maps", a notion designed to measure how well a language separates others.

The definition of rating maps relies on semirings. A *semiring* is a set R equipped with two binary operations $+$ and $\cdot$, called addition and multiplication, satisfying the following axioms:

- $(R, +)$ is a commutative monoid whose neutral element is denoted by 0_R.
- $(R, \cdot)$ is a monoid whose neutral element is denoted by 1_R.
- The multiplication distributes over addition: $r \cdot (s + t) = rs + rt$ and $(s + t) \cdot r = sr + tr$.
- The element 0_R is a zero for multiplication: for any $r \in R$, $0_R \cdot r = r \cdot 0_R = 0_R$.

Moreover, we say that a semiring R is *idempotent* when any element $r \in R$ is idempotent for addition: $r + r = r$. Any idempotent semiring R can be equipped with a canonical order "$\leq$": given $s, r \in R$, we have $s \leq r$ when $s + r = r$. It can be verified that this is indeed an order which is compatible with addition and multiplication (R being idempotent is required).

▶ **Example 11.** The set 2^{A^*} of all languages over A is an idempotent semiring: the addition is union and the multiplication is language concatenation. In this case, the canonical order is inclusion ($H \subseteq L$ if and only if $H \cup L = L$). Another important example is the powerset 2^M of any monoid M. Again the addition is union (therefore, the order is inclusion). The multiplication is obtained from the one of M: given $S, T \in 2^M$, $S \cdot T = \{st \mid s \in S \text{ and } t \in T\}$.

Rating maps. A rating map[1] is a *semiring morphism*, $\rho : 2^{A^*} \to R$ where R is a *finite idempotent semiring*. It can be verified that any rating map is compatible with the canonical order ($K \subseteq L \Rightarrow \rho(K) \leq \rho(L)$). For the sake of improved readability, when applying a rating map ρ to a singleton language $\{w\}$, we shall simply write $\rho(w)$ for $\rho(\{w\})$. The connection with separation only requires to consider special rating maps called "*nice*". A rating map $\rho : 2^{A^*} \to R$ is *nice* if for any language $K \subseteq A^*$, we have $\rho(K) = \sum_{w \in K} \rho(w)$ (note that this sum boils down to a finite one, as R is a finite idempotent commutative monoid for addition).

▶ Remark. Any nice rating map $\rho : 2^{A^*} \to R$ is finitely representable: it is determined by the images $\rho(a)$ of letters $a \in A$. We may speak of algorithms whose inputs are nice rating maps.

Solving $UPol(\mathcal{C})$-separation requires to consider a special class of rating maps: the *$\mathcal{C}$-compatible ones* (our algorithm is restricted to them). The definition is based on a canonical equivalence $\sim_{\mathcal{C}}$ on A^* associated to $\mathcal{C}$. Given $u, v \in A^*$, we write $u \sim_{\mathcal{C}} v$ if and only if $u \in L \Leftrightarrow v \in L$ for all $L \in \mathcal{C}$. Clearly, $\sim_{\mathcal{C}}$ is an equivalence relation. For any word $w \in A^*$, we write $[w]_{\mathcal{C}} \subseteq A^*$ for the $\sim_{\mathcal{C}}$-class of w. Moreover, since $\mathcal{C}$ is a finite quotienting Boolean algebra, we have the following classical properties.

▶ **Lemma 12.** *The equivalence $\sim_{\mathcal{C}}$ is a congruence of finite index for word concatenation. Moreover, for any language $L \subseteq A^*$, we have $L \in \mathcal{C}$ if and only if L is a union of $\sim_{\mathcal{C}}$-classes.*

Lemma 12 implies that the set $A^*/\sim_{\mathcal{C}}$ of $\sim_{\mathcal{C}}$-classes is a finite monoid and the map $w \mapsto [w]_{\mathcal{C}}$ is a morphism. For the sake of avoiding confusion with language concatenation, we shall write "$\bullet$" for the monoid multiplication of $A^*/\sim_{\mathcal{C}}$. In general, if $C, D \subseteq A^*$ are $\sim_{\mathcal{C}}$-classes, then $C \bullet D \neq CD$ (indeed, CD is not even a $\sim_{\mathcal{C}}$-class in general).

[1] What we call rating map here is called **multiplicative** rating map in [22] (the "true" rating maps are weaker and do not require a multiplication). We abuse terminology for the sake of improved readability.

We may now define $\mathcal{C}$-compatibility. We say that a rating map $\rho : 2^{A^*} \to R$ is *$\mathcal{C}$-compatible* when there exists a map $r \mapsto [\![r]\!]_{\mathcal{C}}$ from R to $2^{A^*/\sim_{\mathcal{C}}}$ such that (1) for every $K \subseteq A^*$, we have $[\![\rho(K)]\!]_{\mathcal{C}} = \{[w]_{\mathcal{C}} \mid w \in K\}$ and (2) for all $r, r' \in R$, such that $r \leq r'$, we have $[\![r]\!]_{\mathcal{C}} \subseteq [\![r']\!]_{\mathcal{C}}$.

▶ Remark. Intuitively, Condition (1) in the definition of $\mathcal{C}$-compatibility states that the image $\rho(K)$ of any language K records the $\sim_{\mathcal{C}}$-classes of words of K. The purpose of Condition (2) is to constrain the definition of the map $[\![\ ]\!]_{\mathcal{C}}$ on elements that have no preimage under ρ.

Optimal covers. We use rating maps to define objects called "optimal universal $\mathcal{D}$-covers", which encode separation-related information. We fix an arbitrary Boolean algebra $\mathcal{D}$ for which one wants a $\mathcal{D}$-separation algorithm (we are interested in the case $\mathcal{D} = \mathit{UPol}(\mathcal{C})$).

A *cover of some language* L is a *finite* set of languages $\mathbf{K}$ such that $L \subseteq \bigcup_{K \in \mathbf{K}} K$. When $L = A^*$, we speak of *universal cover*. Moreover, we say that $\mathbf{K}$ is a $\mathcal{D}$-cover when all $K \in \mathbf{K}$ belong to $\mathcal{D}$. A fixed rating map $\rho : 2^{A^*} \to R$ is used to define a "quality measure" for $\mathcal{D}$-covers which yields a notion of "best" universal $\mathcal{D}$-cover. Given a finite set of languages $\mathbf{K}$ (such as a universal $\mathcal{D}$-cover), the *ρ-imprint* $\mathcal{I}[\rho](\mathbf{K})$ of $\mathbf{K}$ is the following subset of R:

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

We now define the optimal universal $\mathcal{D}$-covers as those with the smallest possible ρ-imprint (with respect to inclusion). A *universal $\mathcal{D}$-cover $\mathbf{K}$ is optimal for ρ* when $\mathcal{I}[\rho](\mathbf{K}) \subseteq \mathcal{I}[\rho](\mathbf{K}')$ for any universal $\mathcal{D}$-cover $\mathbf{K}'$. In general, there can be infinitely many optimal universal $\mathcal{D}$-covers for a given rating map ρ. The crucial point is that there always exists a least one. This is simple and proved in [22]. The key idea is that there are finitely many possible ρ-imprints (since R is finite) and given two universal $\mathcal{D}$-covers, one may always build a third one which has a smaller ρ-imprint than the first two, by simple use of language intersections.

Finally, a key observation is that by definition, all optimal universal $\mathcal{D}$-covers for ρ share the same ρ-imprint. This unique ρ-imprint is a *canonical* object for $\mathcal{D}$ and ρ called the *$\mathcal{D}$-optimal universal ρ-imprint* and we denote it by $\mathcal{I}_{\mathcal{D}}[\rho]$. That is, $\mathcal{I}_{\mathcal{D}}[\rho] = \mathcal{I}[\rho](\mathbf{K})$ for any optimal universal $\mathcal{D}$-cover $\mathbf{K}$ for ρ.

The connection with separation. We may now explain how these notions are used to handle separation. This is summarized by the following lemma.

▶ **Lemma 13.** *Let $\mathcal{D}$ be a Boolean algebra. If there exists an algorithm that takes as input a nice $\mathcal{C}$-compatible rating map $\rho : 2^{A^*} \to R$ and outputs $\mathcal{I}_{\mathcal{D}}[\rho]$, then $\mathcal{D}$-separation is decidable.*

Let us sketch how to go from computing $\mathcal{D}$-optimal ρ-imprints to $\mathcal{D}$-separation (see [20, 22] for a full proof of Lemma 13). Consider two regular languages L_1 and L_2: we wish to know whether L_1 is $\mathcal{D}$-separable from L_2. Since $\mathcal{C}$ is finite, one can build a monoid morphism $\alpha : A^* \to M$, with M finite, recognizing both L_1 and L_2 as well as all languages in $\mathcal{C}$. Furthermore, one may lift α as a map $\rho : 2^{A^*} \to 2^M$ by defining $\rho(K) = \{\alpha(w) \mid w \in K\}$ for any language $K \subseteq A^*$. It is simple to verify that this map ρ is a nice $\mathcal{C}$-compatible rating map. Moreover, the two following properties (which we prove in the full version) hold:

- L_1 is $\mathcal{D}$-separable from L_2 iff for any $s_1 \in \alpha(L_1)$ and $s_2 \in \alpha(L_2)$, we have $\{s_1, s_2\} \notin \mathcal{I}_{\mathcal{D}}[\rho]$.
- When the first item holds, one may build a separator in $\mathcal{D}$ from any optimal universal $\mathcal{D}$-cover $\mathbf{K}$ for ρ: this separator is the union of all languages intersecting L_1 in $\mathbf{K}$.

By the first item, having an algorithm that computes $\mathcal{I}_{\mathcal{D}}[\rho] \subseteq 2^M$ suffices to decide whether L_1 is $\mathcal{D}$-separable from L_2. Moreover, by the second item, having an algorithm that computes an optimal universal $\mathcal{D}$-cover $\mathbf{K}$ for ρ is enough to build a separator (when it exists).

▶ Remark. Here, we only use the sets of size two in $\mathcal{I}_{\mathcal{D}}[\rho] \subseteq 2^M$. However, $\mathcal{I}_{\mathcal{D}}[\rho]$ contains more information corresponding to the more general $\mathcal{D}$-covering problem considered in [20, 22].

4.2 Computing $UPol(\mathcal{C})$-optimal universal imprints

We use the framework defined above to present an algorithm for $UPol(\mathcal{C})$-separation. We give a characterization $UPol(\mathcal{C})$-optimal imprints. It yields a procedure for computing them.

Consider a rating map $\rho : 2^{A^*} \to R$. For any subset $S \subseteq R$, we say that S is $UPol(\mathcal{C})$-*saturated* (for ρ) if it contains the set $\mathcal{I}_{triv}[\rho] = \{r \in R \mid r \leq \rho(w) \text{ for some } w \in A^*\}$ and is closed under the following operations:

1. *Downset*: for any $s \in S$, if $r \in R$ satisfies $r \leq s$, then we have $r \in S$.
2. *Multiplication*: For any $s, t \in S$, we have $st \in S$.
3. $UPol(\mathcal{C})$-*closure*: Given two $\sim_{\mathcal{C}}$-classes C, D and $s, t \in S$ such that $[\![s]\!]_{\mathcal{C}} = \{C \bullet D\}$ and $[\![t]\!]_{\mathcal{C}} = \{D \bullet C\}$, we have $s^\omega \cdot \rho(C) \cdot t^\omega \in S$.

We are ready to state the main theorem of this section: when ρ is $\mathcal{C}$-compatible, $UPol(\mathcal{C})$-saturation characterizes the $UPol(\mathcal{C})$-optimal universal ρ-imprint.

▶ **Theorem 14.** *Let $\rho : 2^{A^*} \to R$ be a $\mathcal{C}$-compatible rating map. Then, $\mathcal{I}_{UPol(\mathcal{C})}[\rho]$ is the smallest $UPol(\mathcal{C})$-saturated subset of R (with respect to inclusion).*

Clearly, given a nice $\mathcal{C}$-compatible rating map $\rho : 2^{A^*} \to R$ as input, one may compute the smallest $UPol(\mathcal{C})$-saturated subset of R with a least fixpoint algorithm. One starts from $\mathcal{I}_{triv}[\rho]$ (which is clearly computable) and saturates this set with the three above operations. Thus, we get a procedure for computing $\mathcal{I}_{UPol(\mathcal{C})}[\rho]$ from any input nice $\mathcal{C}$-compatible rating map. By Lemma 13, this yields the desired corollary: $UPol(\mathcal{C})$-separation is decidable.

▶ **Corollary 15.** *For any finite quotienting Boolean algebra $\mathcal{C}$, $UPol(\mathcal{C})$-separation is decidable.*

The proof of Theorem 14 is a difficult generalization of the argument we used to show the algebraic characterization of $UPol(\mathcal{C})$ (*i.e.*, Theorem 4). We postpone it to the full version. An interesting byproduct of this proof is an algorithm which computes optimal universal $UPol(\mathcal{C})$-covers (and therefore $UPol(\mathcal{C})$-separators when they exist, as we explained above).

5 Conclusion

We presented a new, self-contained proof that for any quotienting Boolean algebra regular languages $\mathcal{C}$, membership for $UPol(\mathcal{C})$ reduces to membership for $\mathcal{C}$. An interesting byproduct of this proof is that $UPol(\mathcal{C})$ corresponds exactly to the class $ADet(\mathcal{C})$, which is obtained by restricting the unambiguous marked concatenations to left or right deterministic ones. Moreover, we showed that when $\mathcal{C}$ is a finite quotienting Boolean algebra, $UPol(\mathcal{C})$-separation is decidable. This completes similar results of [21] for $Pol(\mathcal{C})$ and $Bool(Pol(\mathcal{C}))$ and of [16, 17] for $Pol(Bool(Pol(\mathcal{C})))$. These results raise several natural questions.

Historically, $UPol(\mathcal{C})$ was investigated together with two weaker operations: left and right deterministic closures. The left (resp. right) deterministic closure of $\mathcal{C}$, is the smallest class containing $\mathcal{C}$ closed under disjoint union and left (resp. right) deterministic marked concatenation. Our results can be adapted to these two weaker operations. In both cases, membership reduces to $\mathcal{C}$-membership when $\mathcal{C}$ is a quotienting Boolean algebra of regular languages and separation is decidable when $\mathcal{C}$ is a finite quotienting Boolean algebra. In fact, these operations are simpler to handle than $UPol(\mathcal{C})$. We leave this for further work.

Another question is whether our results can be pushed to classes built by combining unambiguous polynomial closure with other operations. A natural example is as follows. It is known [17] that $Pol(Bool(Pol(\mathcal{C})))$-separation is decidable when $\mathcal{C}$ is a finite quotienting Boolean algebra. Is this true as well for $UPol(Bool(Pol(\mathcal{C})))$? This seems difficult: the proof of [17] crucially exploits the fact that $Pol(Bool(Pol(\mathcal{C})))$ is closed under concatenation (which is not the case for $UPol(Bool(Pol(\mathcal{C}))))$ to handle the first polynomial closure.

References

1. Jorge Almeida, Jana Bartonová, Ondrej Klíma, and Michal Kunc. On decidability of intermediate levels of concatenation hierarchies. In *Proceedings of the 19th International Conference on Developments in Language Theory, DLT'15*, volume 9168 of *Lecture Notes in Computer Science*, pages 58–70. Springer, 2015.
2. Mustapha Arfi. Polynomial operations on rational languages. In *Proceedings of the 4th Annual Symposium on Theoretical Aspects of Computer Science, STACS'87*, volume 247 of *Lecture Notes in Computer Science*, pages 198–206. Springer, 1987.
3. Mustapha Arfi. Opérations polynomiales et hiérarchies de concaténation. *Theoretical Computer Science*, 91(1):71–84, 1991.
4. Mário Branco and Jean-Éric Pin. Equations defining the polynomial closure of a lattice of regular languages. In *Proceedings of the 36th International Colloquium on Automata, Languages, and Programming, ICALP'09*, volume 5556 of *Lecture Notes in Computer Science*, pages 115–126. Springer, 2009.
5. Volker Diekert, Paul Gastin, and Manfred Kufleitner. A survey on small fragments of first-order logic over finite words. *International Journal of Foundations of Computer Science*, 19(3):513–548, 2008.
6. Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. First-order logic with two variables and unary temporal logic. In *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science, LICS'97*, pages 228–235. IEEE Computer Society, 1997.
7. Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. First-order logic with two variables and unary temporal logic. *Information and Computation*, 179(2):279–295, 2002.
8. James Alexander Green. On the structure of semigroups. *Annals of Mathematics*, 54(1):163–172, 1951.
9. Jean-Éric Pin. Propriétés syntactiques du produit non ambigu. In *Proceedings of the 7th International Colloquium on Automata, Languages and Programming, ICALP'80*, volume 85 of *Lecture Notes in Computer Science*, pages 483–499. Springer, 1980.
10. Jean-Éric Pin. An explicit formula for the intersection of two polynomials of regular languages. In *Proceedings of the 17th International Conference on Developments in Language Theory, DLT'13*, volume 7907 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 2013.
11. Jean-Éric Pin. Mathematical foundations of automata theory. In preparation, 2016. URL: `http://www.irif.fr/~jep/MPRI/MPRI.html`.
12. Jean-Éric Pin. *The dot-depth hierarchy, 45 years later*, chapter 8, pages 177–202. World Scientific, 2017.
13. 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.
14. Jean-Éric Pin and Pascal Weil. Polynomial closure and unambiguous product. In *Proceedings of the 22nd International Colloquium on Automata, Languages and Programming, ICALP'95*, volume 944 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 1995.
15. Jean-Éric Pin and Pascal Weil. Polynomial closure and unambiguous product. *Theory of Computing Systems*, 30(4):383–422, 1997.
16. Thomas Place. Separating regular languages with two quantifiers alternations. In *Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS'15*, pages 202–213. IEEE Computer Society, 2015.
17. Thomas Place. Separating regular languages with two quantifiers alternations. Unpublished, a preliminary version can be found at `https://arxiv.org/abs/1707.03295`, 2018.
18. Thomas Place, Lorijn van Rooijen, and Marc Zeitoun. Separating regular languages by piecewise testable and unambiguous languages. In *Proceedings of the 38th International*

Symposium on Mathematical Foundations of Computer Science, MFCS'13, volume 8087 of *Lecture Notes in Computer Science*, pages 729–740. Springer, 2013.

19 Thomas Place and Marc Zeitoun. Going higher in the first-order quantifier alternation hierarchy on words. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming, ICALP'14*, volume 8573 of *Lecture Notes in Computer Science*, pages 342–353. Springer, 2014.

20 Thomas Place and Marc Zeitoun. The covering problem: A unified approach for investigating the expressive power of logics. In *Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science, MFCS'16*, volume 58 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 77:1–77:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.

21 Thomas Place and Marc Zeitoun. Separation for dot-depth two. In *Proceedings of the 32th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS'17*, pages 202–213. IEEE Computer Society, 2017.

22 Thomas Place and Marc Zeitoun. The covering problem. Unpublished, a preliminary version can be found at `https://arxiv.org/abs/1707.03370`, 2018.

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

24 John L. Rhodes. A homomorphism theorem for finite semigroups. *Mathematical Systems Theory*, 1:289–304, 1967.

25 Marcel-Paul Schützenberger. Sur le produit de concaténation non ambigu. *Semigroup Forum*, 13:47–75, 1976.

26 Imre Simon. Piecewise testable events. In *Proceedings of the 2nd GI Conference on Automata Theory and Formal Languages*, volume 33 of *Lecture Notes in Computer Science*, pages 214–222. Springer, 1975.

27 Pascal Tesson and Denis Thérien. Diamonds are forever: The variety DA. In *Semigroups, Algorithms, Automata and Languages*, pages 475–500. World Scientific, 2002.

28 Denis Thérien and Thomas Wilke. Over words, two variables are as powerful as one quantifier alternation. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing, STOC'98*, pages 234–240. ACM, 1998.

29 Wolfgang Thomas. Classifying regular events in symbolic logic. *Journal of Computer and System Sciences*, 25(3):360–376, 1982.