# On Boolean closed full trios and rational Kripke frames

**Markus Lohrey**[1] **and Georg Zetzsche**[2]

1 **Department für Elektrotechnik und Informatik, Universität Siegen**
2 **Fachbereich Informatik, Technische Universität Kaiserslautern**

## Abstract

A Boolean closed full trio is a class of languages that is closed under Boolean operations (union, intersection, and complement) and rational transductions. It is well-known that the regular languages constitute such a Boolean closed full trio. It is shown here that every such language class that contains any non-regular language already includes the whole arithmetical hierarchy (and even the one relative to this language).

Our construction also shows that there is a fixed rational Kripke frame such that assigning an arbitrary non-regular language to some variable allows the interpretation of any language from the arithmetical hierarchy in the corresponding Kripke structure.

Another consequence of our result is that no full trio generated by one language is closed under complementation, unless it coincides with the regular languages.

## 1 Introduction

The study of closure properties of language classes has a long tradition, early results have been obtained shortly after the introduction of regular languages [5]. Among other applications, they provide insights about whether languages belong to certain classes and, as far as they are effective, allow the computation of representations of languages. They also often serve as a way to describe language classes without reference to concrete generating or accepting devices: In many cases, a language class can be described as the smallest class of languages that possesses a given collection of closure properties and contains certain generating languages.

Here, we are concerned with Boolean closed full trios, i.e., classes closed under Boolean operations (union, intersection, and complementation) and rational transductions. It is well-known that the class of regular languages constitutes a Boolean closed full trio.

This combination of closure properties is interesting for several reasons. First, in the case of regular languages, this particular collection is exploited, for example, in the theory of automatic structures [10], since it implies that in such structures, every first-order definable relation can be represented by a regular language. Since emptiness is decidable for regular languages, one can therefore decide the first-order theory of these structures.

Second, the languages definable by multimodal logic in a rational Kripke frame, i.e., a Kripke frame in which the worlds are words and the visibility relations are given by rational transductions, are always confined to the Boolean closed full trio generated by the values (that is, languages) assigned to the variables. This was observed by Bekker and Goranko [2] and then used to show that the model checking problem for multimodal logic and rational Kripke frames is decidable if all variables are assigned regular languages.

Third, a wide range of interesting language classes are principal full trios, i.e., full trios that are generated by one language. Since these are always union closed, their closure under complementation is equivalent to the class being a Boolean closed full trio. Examples of principal full trios are the context-free languages, languages accepted by multicounter

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

automata (for a bounded number of counters and blind, partially blind, or with zero test [8]), and the languages accepted by valence automata over a finitely generated monoid [7].

Hence, the question arises whether there are languages classes beyond the regular languages that enjoy these closure properties and still admit decision procedures for simple properties such as emptiness. Our first main result (Theorem 9) states that every Boolean closed full trio that contains any *non-regular language* already includes the whole arithmetical hierarchy (and even the arithmetical hierarchy relative to this language) and thus loses virtually all decidability properties. This is a remarkable fact, because it means that these closure properties are so extremely powerful that even the simplest non-regular languages allow the construction of a very large class of languages. We can also conclude that *other than the regular languages, no principal full trio* is closed under complementation.

Actually, it turns out that three fixed rational transductions, together with the Boolean operations, suffice to construct all arithmetical languages from any non-regular language. Thus, our second main result (Theorem 14) states that there is a fixed rational Kripke frame with three modalities such that assigning any non-regular language to a variable allows the definition of every arithmetical language.

It should be noted that our main results do not mean that there is no way of developing a theory of automatic structures beyond regular languages. It might well be that some smaller collection of closure properties suffices to obtain all first-order definable relations and still admits a decision procedure for the emptiness problem.

A large number of grammar and automata models is easily seen to produce only recursively enumerable languages. Hence, Theorem 9 also implies that the corresponding language classes are never Boolean closed full trios.

Other results of a similar spirit on closure properties of language classes have been known for a long time. For example, Hartmanis and Hopcroft [9] proved that every intersection closed full AFL containing $\{a^nb^n \mid n \in \mathbb{N}\}$ includes the recursively enumerable languages. Here, a full AFL is a full trio that is closed under union and the Kleene star. Furthermore, Book [4] has shown that the arithmetical languages constitute the smallest Boolean closed full trio that is closed under homomorphic replication, the latter of which is a generalization of homomorphisms. Hence, our result means in Book's result one can replace the homomorphic replication by containment of any non-regular language. However, to the best of the authors' knowledge, to date there is no known combination of natural closure properties that are enjoyed by the regular languages but that yield all the recursively enumerable languages (let alone the arithmetical hierarchy) when applied to any non-regular language.

## 2 Preliminaries

Let $\Sigma$ be a fixed countable set of abstract symbols, the finite subsets of which are called *alphabets*. Subsets of $X^*$ for alphabets $X$ are called *languages*. For a language $L$, the smallest alphabet $X$ with $L \subseteq X^*$ is denoted by $\alpha(L)$. The *complement* of $L$ is defined as $\overline{L} = \alpha(L)^* \setminus L$. A *transduction* is a subset of $X^* \times Y^*$ for alphabets $X$, $Y$.

Let $M$ be a monoid. An *automaton over* $M$ is a tuple $A = (Q, M, E, q_0, F)$, in which $Q$ is a finite set of *states*, $E$ is a finite subset of $Q \times M \times Q$ called the set of *edges*, $q_0 \in Q$ is the *initial state*, and $F \subseteq Q$ is the set of *final states*. The *step relation* $\Rightarrow_A$ of $A$ is a binary relation on $Q \times M$, for which $(p, a) \Rightarrow_A (q, b)$ iff there is an edge $(p, c, q)$ such that $b = ac$. The set generated by $A$ is then

$$S(A) = \{a \in M \mid \exists q \in F : (q_0, 1) \Rightarrow_A^* (q, a)\}.$$

A set $R \subseteq M$ is called *rational* if it can be written as $R = S(A)$ for some automaton $A$ over $M$. A rational language is also called *regular*. We use $\mathsf{REG}$ to denote the class of regular languages.

Given alphabets $X$ and $Y$, a *rational transduction* is a rational subset of the monoid $X^* \times Y^*$. For a language $L \subseteq Y^*$ and a rational transduction $R$, we write $RL$ for $\{x \in X^* \mid \exists y \in L : (x, y) \in R\}$.

A *language class* is a set of languages that contains at least one non-empty language. A language class $\mathcal{C}$ is called a *full trio* if it is closed under (arbitrary) homomorphisms, inverse homomorphisms, and regular intersections. It is well-known [3] that a class $\mathcal{C}$ is a full trio if and only if it is closed under rational transductions, i.e., for every $L \in \mathcal{C}$ and every rational transduction $R$, we have $RL \in \mathcal{C}$. We call a language class *Boolean closed* if it is closed under all Boolean operations (union, intersection, and complementation). By the *full trio generated by* the language $L$ we mean the smallest full trio that contains $L$. A full trio is called a *principal full trio* if it is generated by one language.

For any language class $\mathcal{C}$, we write $\mathsf{RE}(\mathcal{C})$ for the class of languages accepted by some Turing machine with an oracle $L \in \mathcal{C}$. Similarly, let $\mathsf{REC}(\mathcal{C})$ be the class of languages accepted by some Turing machine that halts on every input and has access to an oracle $L \in \mathcal{C}$. Furthermore, let $\mathsf{REC}$ denote the class of recursive languages. We also write $\mathsf{REC}(L)$ and $\mathsf{RE}(L)$ for $\mathsf{REC}(\{L\})$ and $\mathsf{RE}(\{L\})$, respectively. Then the *arithmetical hierarchy* (see, for example, [11]) is defined as

$$\Sigma_0 = \mathsf{REC}, \qquad \Sigma_{n+1} = \mathsf{RE}(\Sigma_n) \text{ for } n \geq 0, \qquad \mathsf{AH} = \bigcup_{n \geq 0} \Sigma_n.$$

Languages in $\bigcup_{n \geq 0} \Sigma_n$ are called *arithmetical*. The *arithmetical hierarchy relative to* $L$ is defined as

$$\Sigma_0(L) = \mathsf{REC}(L), \qquad \Sigma_{n+1}(L) = \mathsf{RE}(\Sigma_n(L)) \text{ for } n \geq 0, \qquad \mathsf{AH}(L) = \bigcup_{n \geq 0} \Sigma_n(L).$$

A *valence automaton over* $M$ is an automaton $A$ over the monoid $X^* \times M$, where $X$ is an alphabet. The *language accepted by* $A$ is defined as $L(A) = \{w \in X^* \mid (w, 1) \in S(A)\}$. The class of languages accepted by valence automata over $M$ is denoted by $\mathsf{VA}(M)$.

We will often encode words from an alphabet $X$, $|X| \geq 2$, by words in $\{0, 1\}^*$. If $X = \{a_1, \ldots, a_n\}$, then a homomorphism $g\colon X^* \to \{0, 1\}^*$ with $g(a_i) = 10^i$ will be called a *standard encoding*. For each subset $Y \subseteq X$, the homomorphism $\pi_Y : X^* \to Y^*$ is defined by $\pi_Y(x) = x$ for $x \in Y$ and $\pi_Y(x) = \lambda$ for $x \in X \setminus Y$.

Let $X$ be an alphabet. For languages $L \subseteq X^*$ and words $u, v \in X^*$, we write $u \equiv_L v$ if for each $w \in X^*$, we have $uv \in L$ if and only if $uw \in L$. The equivalence relation $\equiv_L$ is called the *Myhill-Nerode equivalence*. The well-known Myhill-Nerode Theorem states that $L$ is regular if and only if $\equiv_L$ has a finite index.

▶ Remark. In the following, we will make statements about certain languages in $\{0, 1\}^*$ being obtainable from other languages in $\{0, 1\}^*$ either by using a finite set of transductions or by using a finite set of transductions and Boolean operations. It will then always be possible to use larger alphabets with auxiliary symbols for the following reason. Suppose there is a finite set $S$ of transductions, each over the alphabet $X = \{a_1, \ldots, a_n\}$, $\{0, 1\} \subseteq X$. Let $h : X^* \to \{0, 1\}^*$ be a standard encoding. Then we have

$$h(L \cap K) = h(L) \cap h(K), \qquad h(L \cup K) = h(L) \cup h(K),$$
$$h(\overline{L}) = \overline{h(L)} \cap h(\alpha(L))^*, \qquad h(RL) = hRh^{-1}(h(L)).$$

By induction, it follows that for every language $K \subseteq X^*$ that can be obtained from $L \subseteq X^*$ using transductions in $S$ (and Boolean operations), we can obtain $h(K)$ from $h(L)$ by using transductions in

$$S' = \{\rho_Y \mid Y \subseteq X\} \cup \{hRh^{-1} \mid R \in S\}$$

(and Boolean operations), where $\rho_Y$ is the rational transduction in $\{0,1\}^* \times \{0,1\}^*$ that maps $M$ to $M \cap h(Y)^*$. In particular, if $K \subseteq \{0,1\}^*$ can be obtained from $L \subseteq \{0,1\}^*$ using transductions in $S$ (and Boolean operations), we can obtain $K$ from $L$ by using transductions in $S'' = S' \cup \{h\mid_{\{0,1\}^*}, \rho_{\{0,1\}} \circ h^{-1}\}$ (and Boolean operations) by producing $h(L)$, then $h(K)$ using $S'$ (and Boolean operations), and then $\rho_{\{0,1\}}(h^{-1}(h(K))) = K$. ◀

## 3 Boolean closed full trios

▶ **Lemma 1.** *Let $X = \{0,1\}$. There is a finite set $F$ of rational transductions in $X^* \times X^*$ such that each regular language $K \subseteq X^*$ can be obtained from any non-empty $L \subseteq X^*$ using transductions in $F$.*

**Proof.** It suffices to prove the lemma for $K \subseteq X^*$ with $\lambda \notin K$: If $\lambda \in K$, $K \setminus \{\lambda\} \neq \emptyset$, we can use the rational transduction $\Lambda = \{(w,w) \mid w \in X^*\} \cup \{\lambda\} \times X^*$, which maps $w \in X^*$ to $\{w, \lambda\}$, to obtain $K$ from $K \setminus \{\lambda\}$. If $K = \{\lambda\}$, we can use $\Lambda' = \{\lambda\} \times X^*$, which maps every $w \in \{0,1\}^*$ to $\lambda$, to obtain $K$ directly from $L$. We may therefore assume that $K$ is accepted by an automaton $A = (Q, X, E, q, Q_f)$, where $Q = \{0, \ldots, k\}$, $q = 0$, $Q_f = \{1\}$, and $E \subseteq Q \times X \times Q$.

Our goal is to produce the language $T_A$ of all words $10^{i_0}1x_110^{i_1} \cdots x_n10^{i_n}$, such that $i_0 = 0$, $i_n = 1$, and $x_j \in \{0,1\}$ and $(i_j, x_{j+1}, i_{j+1}) \in E$ for $0 \leq j < n$. Then, clearly, the rational transduction $P$ that outputs only the $x_j$ will satisfy $PT_A = K$. By the above remark, it suffices to provide transductions over the extended alphabet $Y = \{0, 1, \#_1, \#_2\}$. The additional symbols $\#_1, \#_2$ are called *markers*.

First we use the initial transduction $I = 1(1\{0,1\}10^*)^*1\{0,1\}10 \times \{0,1\}^*$ to produce the set $1(1\{0,1\}10^*)^*1\{0,1\}10$ from $L$. In the following, a word $10^{i_0}1x_110^{i_1} \cdots x_n10^{i_n}$ is called an *encoding*. Its factors $0^{i_j}$ are called *state blocks* and its factors $0^{i_j}1x10^{i_{j+1}}$ are called *transition blocks*. The transduction $I$ already guarantee that the leftmost and the rightmost state block correspond to the initial and the final state, respectively. We now wish to remove all words that contain a state block of length greater than $k$. In order to do this, we use the transduction $S_1$, which inserts the marker $\#_1$ in the beginning of every state block. Furthermore, we have the transduction $M_1$, which moves each occurrence of the marker one position to the right (i.e. outputs $0\#_1$ on input $\#_10$) if its right neighbor is a 0, and drops the occurrence otherwise. We also have the transduction $R$, which rejects all inputs that have a factor $\#_10$. All other words are accepted by $R$ but stripped of their occurrences of $\#_1$ in the output. Then applying $RM_1^kS_1$ yields the set of encodings with state blocks of length at most $k$.

In the next step, we wish to remove from the language all encodings that contain a transition block $10^\ell x10^m$ with $x \in \{0,1\}$, $0 \leq \ell, m \leq k$, and $(\ell, x, m) \notin E$. To this end, we have the transductions $S_2$ and $M_2$, which behave analogously to $S_1$ and $M_1$ by using $\#_2$ instead of $\#_1$. We assume that $S_1$ and $S_2$ are defined so as to add their marker and leave the other marker in place. We assume further that $M_1$ and $M_2$ move their marker so as to overtake the other marker if necessary. Finally, we have for each $x \in \{0,1\}$ the transition $R_x$, which rejects every word containing a transition block in which $\#_1$ is on the right end of the left state block, $\#_2$ is on the right end of the right state block, and the input letter is $x$. All other words are accepted by $R_x$ but stripped of all occurrences of markers. Applying $R_x M_2^m S_2 M_1^\ell S_1$ clearly yields the set of encodings that do not contain the transition block $10^\ell 1x10^m$. Therefore, we apply this sequence of transductions for each triple $(\ell, x, m)$ with $0 \le \ell, m \le k$, $x \in \{0,1\}$, and $(\ell, x, m) \notin E$. This clearly produces the language $T_A$ and hence $K = PT_A$ is obtained. Since we only used the transductions $\{\Lambda, \Lambda', P, I, S_1, S_2, M_1, M_2, R, R_0, R_1\}$, the lemma is proven. ◀

▶ **Lemma 2.** *Let $X$ be an alphabet with $|X| \ge 2$. For each finite set $F$ of rational transductions in $X^* \times X^*$, there are rational transductions $R, S, T$ in $X^* \times X^*$ such that every composition of transductions from $F$ can be written in the form $T^n S^m R$ with $m, n \in \mathbb{N}$.*

**Proof.** Let $0, 1 \in X$ be distinct letters and for $x \in \{0,1\}$, let $A_x$ be the transduction that appends $x$ to each input word, hence $A_x = \{(wx, w) \mid w \in X^*\}$. Furthermore, let $F = \{U_0, \dots, U_{k-1}\}$, $b = k+1$, and let $U_i'$ be the rational transduction

$$U_i' = \{(u10^m, v10^{bm+i}) \mid (u,v) \in U_i, m \in \mathbb{N}\}, \quad U_k' = \{(w, w10^k) \mid w \in X^*\}$$

for each $0 \le i < k$. We shall prove that $R = A_1$, $S = A_0$, and $T = \bigcup_{0 \le i \le k} U_i'$ have the desired property. Let $U_{i_n} \cdots U_{i_0}$ be a composition of elements of $F$ and let $i_{n+1} = k$. We claim that

$$U_{i_n} \cdots U_{i_0} = T^{n+2} S^m R \quad \text{for } m = \sum_{j=0}^{n+1} i_j b^j.$$

Applying $S^m R$ appends $10^m$ to each input word. Then, each application of $T$ to a word $w10^\ell$ chooses one of the $U_j'$, but this choice will only lead to a valid computation of the transducer if $\ell$ is congruent to $i_j$ modulo $b$. Hence, applying $T^{n+1}$ has the same effect as applying $U_{i_n}' \cdots U_{i_0}'$. Since the most significant digit in the $b$-ary representation of $m$ is $i_{n+1} = k$, applying $T$ once more means applying $U_k'$ and hence removing the $10^k$ suffix of the input word. In the end, we applied $U_{i_n} \cdots U_{i_0}$. ◀

Lemmas 1 and 2 together immediately imply the following byproduct, which might be of independent interest.

▶ **Corollary 3.** *Let $X = \{0,1\}$. There are rational transductions $R, S, T$ over $X^*$ such that every regular language $K \subseteq X^*$ can be written as $T^n S^m R X^*$ for some $m, n \in \mathbb{N}$.*

Let $\Delta = \{+, -, z\}$, whose elements will represent the operations *increment*, *decrement*, and *zero test*, respectively.

▶ **Definition 4.** Let $C \subseteq \Delta^*$ be the set of words $\delta_1 \cdots \delta_m$, $\delta_1, \dots, \delta_m \in \Delta$, for which there are numbers $x_0, \dots, x_m \in \mathbb{N}$ such that for $1 \le i \le m$:

1. if $\delta_i = +$, then $x_i = x_{i-1} + 1$,
2. if $\delta_i = -$, then $x_i = x_{i-1} - 1$, and
3. if $\delta_i = z$, then $x_i = x_{i-1} = 0$.

We shall prove that from $L$ we can construct the following language $\hat{C}_L$ using a fixed finite set of rational transductions and Boolean operations.

▶ **Definition 5.** Let $\hat{C}_L$ be the set of all words

$$v_0\delta_1v_1\cdots\delta_mv_m\#u_0\#\cdots u_n\#$$

with $\delta_i \in \Delta$, $v_i \in X^*$, $u_j \in X^*$, such that $u_k \not\equiv_L u_\ell$ for $k \neq \ell$ and for each $1 \leq i \leq m$ there is a $1 \leq j \leq n$ with

1. if $\delta_i = +$, then $v_{i-1} \equiv_L u_{j-1}$, $v_i \equiv_L u_j$
2. if $\delta_i = -$, then $v_{i-1} \equiv_L u_j$, $v_i \equiv_L u_{j-1}$
3. if $\delta_i = z$, then $j = 1$ and $v_{i-1} \equiv_L v_i \equiv_L u_0 = u_{j-1}$.

▶ **Lemma 6.** *If $L$ is not regular, then $\pi_\Delta(\hat{C}_L) = C$.*

**Proof.** In order to prove the inclusion "$\supseteq$", let $x_0, \ldots, x_m \in \mathbb{N}$ be numbers as in Definition 4 and suppose $\{x_0, \ldots, x_m\} \subseteq \{0, \ldots, n\}$. Since $L$ is not regular, we can find words $u_0, \ldots, u_n \in X^*$ such that $u_k \not\equiv_L u_\ell$ for $k \neq \ell$. Now for each $0 \leq i \leq m$, let $v_i = u_{x_i}$. Then it can be checked straightforwardly that $v_0\delta_1v_1\cdots\delta_mv_m\#u_0\#\cdots u_n\# \in \hat{C}_L$ and hence $\delta_1\cdots\delta_m \in \pi_\Delta(\hat{C}_L)$.

For the inclusion "$\subseteq$", let $\delta_1\cdots\delta_m \in \pi_\Delta(\hat{C}_L)$. Then there are words $v_0, \ldots, v_m \in X^*$, $u_0, \ldots, u_n \in X^*$ with $v_0\delta_1v_1\cdots\delta_mv_m\#u_0\#\cdots u_n\# \in \hat{C}_L$. By the definition of $\hat{C}_L$, this means for each $1 \leq i \leq m$, there is a $1 \leq j \leq n$ such that 1–3 of Definition 5 hold. Hence, we can pick for each $1 \leq i \leq m$ an $x_i \in \{1, \ldots, n\}$ such that 1–3 of Definition 5 hold with $j = x_i$. Note that since this implies $v_{i-1} \equiv_L u_{j-1}$ for $\delta_i \in \{+, z\}$ and $v_{i-1} \equiv_L u_j$ for $\delta_i = -$ and the $u_k$ are pairwise incongruent w.r.t. $\equiv_L$, this choice of $x_i$ is unique. It can now be verified by induction on $i$ that the conditions 1–3 of Definition 4 are satisfied. ◀

The following lemma is the central ingredient in our proof. The idea is to construct $\hat{C}_L$, which by Lemma 6 allows us to obtain $C$.

▶ **Lemma 7.** *Let $X = \{0, 1\}$. There is a finite set $F$ of rational transductions such that for any non-regular $L \subseteq X^*$, the language $C$ can be obtained from $L$ using transductions in $F$ and Boolean operations.*

**Proof.** We will use the alphabet $Y = X \cup \{\#\} \cup \Delta$. We prove the lemma by constructing $C$ from $L$ using a sequence of Boolean operations and transductions $T_1, \ldots, T_{19}$ over $Y^*$ for which it will be clear that they do not depend on $L$.

There are clearly rational transductions $T_1$ and $T_2$ with

$$W_1 = \{u\#v\#w \mid u, v, w \in X^*,\ uw \in L\} = T_1L,$$
$$W_2 = \{u\#v\#w \mid u, v, w \in X^*,\ vw \in L\} = T_2L,$$

which means we can construct $W_1$ and $W_2$. Hence,

$$\begin{aligned} W' =& \{u\#v\#w \mid u, v, w \in X^*, (uw \in L, vw \notin L) \text{ or } (uw \notin L, vw \in L)\} \\ =& (W_1 \cap \overline{W_2}) \cup (\overline{W_1} \cap W_2) \end{aligned}$$

can also be constructed. We can clearly find a rational transduction $T_3$ with

$$W = \{u\#v \mid u, v \in X^*, u \not\equiv_L v\} = \{u\#v \mid u\#v\#w \in W' \text{ for some } w \in X^*\} = T_3W'.$$

This means $P = \{u\#v \mid u \equiv_L v\} = X^*\#X^* \setminus W = T_4\overline{W}$, for some $T_4$, can be constructed. With suitable rational transductions $T_5, T_6$, we have

$$\begin{aligned} S &= \{u_0\#u_1\#\cdots u_n\# \mid u_i \not\equiv_L u_j \text{ for all } i \neq j\} \\ &= (X^*\#)^* \setminus \{ru\#sv\#t \mid r, s, t \in (X^*\#)^*, u\#v \in P\} = T_6\overline{T_5P}, \end{aligned}$$

meaning $S$ can be constructed as well. Let $M$ (*matching*) be the set of all words $v_1\delta v_2\#u_1\#u_2$, $v_1, v_2, u_1, u_2 \in X^*$, with

- if $\delta = +$, then $v_1 \equiv_L u_1$ and $v_2 \equiv_L u_2$,
- if $\delta = -$, then $v_1 \equiv_L u_2$ and $v_2 \equiv_L u_1$, and
- if $\delta = z$, then $v_1 \equiv_L v_2 \equiv_L u_1$.

Since

$$\begin{aligned} M &= \{v_1{+}v_2\#u_1\#u_2 \mid v_1\#u_1 \in P,\ v_2\#u_2 \in P\} \\ &\quad \cup \{v_1{-}v_2\#u_1\#u_2 \mid v_1\#u_2 \in P,\ v_2\#u_1 \in P\} \\ &\quad \cup \{v_1 z v_2\#u_1\#u_2 \mid v_1\#v_2 \in P,\ v_1\#u_1 \in P,\ u_2 \in X^*\} \\ &= (T_7P \cap T_8P) \cup (T_9P \cap T_{10}P) \cup (T_{11}P \cap T_{12}P) \end{aligned}$$

for suitable rational transductions $T_7, \ldots, T_{12}$, we can also construct $M$.

Let $E$ (*error*) be the set of words $v_1\delta v_2\#u_0\# \cdots u_n\#$ such that for every $1 \le j \le n$, we have $v_1\delta v_2\#u_{j-1}\#u_j \notin M$ or we have $\delta = z$ and $v_1 \not\equiv_L u_0$. Since

$$E' = \{v_1\delta v_2\#ru_1\#u_2\#s \mid v_1\delta v_2\#u_1\#u_2 \in M, r, s \in (X^*\#)^*\} = T_{13}M$$

for some rational transduction $T_{13}$, we can construct $E'$. Furthermore, since

$$E = \{v_1 z v_2\#u_0 r \mid v_1 \not\equiv_L u_0,\ r \in (X^*\#)^*\} \cup [(X^*\Delta X^*\#(X^*\#)^* \setminus E'] = T_{14}\overline{P} \cup T_{15}\overline{E'},$$

for some rational transductions $T_{14}, T_{15}$, we can construct $E$.

Let $N$ (*no error*) be the set of words $v_0\delta_1 v_1 \cdots \delta_m v_m\#u_0\# \cdots u_n\#$ such that for every $1 \le i \le m$, there is a $1 \le j \le n$ with $v_{i-1}\delta v_i\#u_{j-1}\#u_j \in M$ and if $\delta_i = z$, then $v_{i-1} \equiv_L u_0$. Since

$$\begin{aligned} N' &= \{w \in (X^*\Delta)^* v_1\delta v_2(\Delta X^*)^*\#u_0\# \cdots u_n\# \mid v_1\delta v_2\#u_0\# \cdots u_n\# \in E\} = T_{16}E, \\ N &= (X^*\Delta)^+X^*\#(X^*\#)^* \setminus N' = T_{17}\overline{N'} \end{aligned}$$

for some rational transductions $T_{16}, T_{17}$, we can construct $N$.

Now we have $\hat{C}_L = N \cap (X^*\Delta)^*X^*\#S = N \cap T_{18}S$ for some rational transduction $T_{18}$, meaning we can construct $\hat{C}_L$. By Lemma 6, we have $C = T_{19}\hat{C}_L$ for some rational transduction $T_{19}$. This proves our claim and hence the lemma. ◀

▶ **Lemma 8.** *Let $X = \{0, 1\}$. There is a finite set $F$ of rational transductions in $X^* \times X^*$ such that for any non-regular $L \subseteq X^*$, each $K \in \mathsf{RE}$, $K \subseteq X^*$, can be obtained from $L$ using transductions in $F$ and Boolean operations.*

**Proof.** Let $F'$ contain the set of rational transductions provided by Lemma 1 and the one provided by Lemma 7. We will use the alphabet $Y = X \cup \Delta \cup \{\#\}$ and a standard encoding $g : Y^* \to X^*$.

Suppose $K \subseteq X^*$ is recursively enumerable and let $A = (Q, X, E, q_0, Q_f)$ be a 2-counter machine, $E \subseteq Q \times X^* \times \Delta \times \Delta \times Q$, accepting $K$ and with $Q = \{0, \ldots, k\}$ and $Q_f = \{k\}$. Here, we assume that the machine operates on both counters in each step. Let $R$ be the regular language of all words $0^{m_0} \prod_{i=1}^{n} \#w_i\#\delta_i^{(0)}\delta_i^{(1)}0^{m_i}$ with $(m_{i-1}, w_i, \delta_i^{(0)}, \delta_i^{(1)}, m_i) \in E$ for every $1 \le i \le n$, $m_0 = 0$, and $m_n = k$. We can obtain $g(R)$ from $L$ using only transductions in $F'$. Thus, we can obtain $R = g^{-1}(g(R))$. Clearly, there are rational transductions $T_1$ and $T_2$ such that

$$U = \left\{ 10^0 \prod_{i=1}^{n} \#w_i\#\delta_i^{(0)}\delta_i^{(1)}10^{m_i} \in R \,\middle|\, \delta_1^{(k)} \cdots \delta_n^{(k)} \in C \text{ for } k = 0, 1 \right\} = R \cap T_1C \cap T_2C,$$

meaning that we can also obtain $U$. Finally, applying to $U$ the transduction $T_3$ that outputs all occurrences of $X$ after odd occurrences of $\#$ up to the next occurrence of $\#$ clearly yields $K$. If we let $F$ consist of $F'$ and $g^{-1}$, $T_1, T_2, T_3$, the lemma is proven. ◀

▶ **Theorem 9.** *Let $X = \{0,1\}$. There are rational transductions $R, S, T$ over $X^*$ such that for any non-regular $L \subseteq X^*$, each $K \in \mathsf{AH}(L)$, $K \subseteq X^*$, can be obtained from $L$ using $R, S, T$ and Boolean operations.*

**Proof.** We shall prove that there is a finite set $F$ of rational transductions in $X^* \times X^*$ such that for any $K \subseteq X^*$, we can obtain each $M \in \mathsf{RE}(K)$, $M \subseteq X^*$, from $K$ and $L$ using transductions in $F$ and Boolean operations. This clearly implies that we can obtain all of $\Sigma_1(L) = \mathsf{RE}(L)$ from $L$ and hence by induction on $i$ all of $\Sigma_i(L)$ from $L$. According to Lemma 2 we can then find transductions $R, S, T$ that have the desired property.

Let $F'$ be the set of transductions provided by Lemma 8 and let $K \subseteq X^*$ be arbitrary and $M \in \mathsf{RE}(K)$, $M \subseteq X^*$. This means there is an oracle Turing machine $A$ such that $M$ is accepted by $A^K$. We will use the extended alphabet $Y = \{0, 1, \#_1, \#_2\}$ and a standard encoding $g : Y^* \to \{0,1\}^*$. Let $M' \subseteq Y^*$ be the set of words

$$u_1\#_1 \cdots u_n\#_1 v_1\#_2 \cdots v_m\#_2 w$$

such that there is an accepting computation in $A$ with input $w$ and in which oracle queries about $u_1, \ldots, u_n$ are made with a positive result and oracle queries about $v_1, \ldots, v_m$ are made with a negative result. Note that this does not mean that $u_i \in K$ or $v_i \notin K$, we collect all computations that $A$ could make and what inputs would be accepted provided that an oracle answered as specified. Then $M'$ is clearly recursively enumerable. Therefore, $g(M')$ can be obtained from $L$ by transductions in $F'$ and Boolean operations.

Hence, we can obtain $M' = g^{-1}(g(M'))$ from $L$. Furthermore, since

$$(K\#_1)^* = \overline{(X^*\#_1)^*\overline{K}\#_1(X^*\#_1)} = \overline{T_1\overline{K}}, \qquad (\overline{K}\#_2)^* = \overline{(X^*\#_2)^*K\#_2(X^*\#_2)} = \overline{T_2K}$$

for some rational transductions $T_1, T_2$, we can construct $(K\#_1)^*$ and $(\overline{K}\#_2)^*$ from $K$. Moreover, since

$$\begin{aligned} M'' &= \{u_1\#_1 \cdots u_n\#_1 v_1\#_2 \cdots v_m\#_2 w \in M' \mid u_1, \ldots, u_n \in K, \quad v_1, \ldots, v_m \in \overline{K}\} \\ &= M' \ \cap\ (K\#_1)^*(X^*\#_2)^*X^* \ \cap\ (X^*\#_1)^*(\overline{K}\#_2)^*X^*, \\ &= M' \ \cap\ T_3(K\#_1)^* \ \cap\ T_4(\overline{K}\#_2)^*, \end{aligned}$$

for suitable rational transductions $T_3, T_4$, we can construct $M''$ from $K$ and $L$. If we now apply a transduction $T_5$ that for an input from $Y^*$ outputs the longest suffix in $X^*$, we obtain $M$ from $K$ and $L$. Since, apart from the transductions in $F'$, we only used $g^{-1}$ and $T_1, \ldots, T_5$, the lemma follows. ◀

▶ **Corollary 10.** *Let $L \subseteq X^*$ be a non-regular language. Then $\mathsf{AH}(L)$ is the smallest Boolean closed full trio containing $L$.*

**Proof.** Let $\mathcal{T}$ be the smallest Boolean closed full trio containing $L$. If $|X| \leq 2$, Theorem 9 implies that $\mathcal{T}$ includes $\mathsf{AH}(L)$. If $|X| > 2$, let $g : X^* \to \{0,1\}^*$ be a standard encoding. Then $g(L)$ is non-regular as well and we have $\mathsf{AH}(L) = \mathsf{AH}(g(L))$. Hence, according to Theorem 9, $\mathcal{T}$ includes $\mathsf{AH}(L) = \mathsf{AH}(g(L))$. The fact that $\mathsf{AH}(L)$ is a Boolean closed full trio concludes the proof. ◀

The following corollary applies to a wide range of language classes. A *full semi-AFL* is a union closed full trio. Although the authors are not aware of any particular full semi-AFL for which it is not known whether complementation closure is available, the following fact is interesting because of its generality.

▶ **Corollary 11.** *Other than the regular languages, no full semi-AFL $\mathcal{C} \subseteq \mathsf{RE}$ is closed under complementation.*

**Proof.** Suppose $\mathcal{C}$ were a complementation closed full semi-AFL that contains a non-regular language. According to Theorem 9, it would already include $\mathsf{AH}$ and thus not be included in $\mathsf{RE}$. ◀

▶ **Corollary 12.** *A principal full trio is closed under complementation if and only if it coincides with the regular languages.*

**Proof.** Let $\mathcal{T}$ be a principal full trio generated by the language $L$. If $L$ is regular, $\mathcal{T}$ coincides with the regular languages and is therefore closed under complementation.

Suppose $L$ is not regular. $\mathcal{T}$ consists of all languages of the form $RL$, where $R$ is a rational transduction. Hence, $\mathcal{T}$ is contained in $\mathsf{RE}(L)$ and closed under union. If $\mathcal{T}$ were closed under complementation, it would be closed under all Boolean operations and thus, by Theorem 9, contain $\mathsf{AH}(L)$. Since $\mathsf{RE}(L) \subsetneq \mathsf{AH}(L)$, this is a contradiction. ◀

▶ **Corollary 13.** *For finitely generated monoids $M$, the following are equivalent:*

1. $\mathsf{VA}(M)$ *is closed under complementation.*
2. $\mathsf{VA}(M) = \mathsf{REG}$.
3. $M$ *has finitely many right-invertible elements.*

**Proof.** Let $L$ be the identity language corresponding to some finite generating set of $M$. Since $\mathsf{VA}(M)$ is the principal full trio generated by $L$, Corollary 12 yields the equivalence between 1 and 2. The equivalence between 2 and 3 has been shown in [14] (and independently in [16]). ◀

## 4 Rational Kripke frames

Theorem 9 can be also restated in terms of multimodal logic. A *Kripke structure* (or edge- and node-labeled graph) is a tuple

$$\mathcal{K} = (V, (E_a)_{a \in A}, (U_p)_{p \in P}),$$

where $V$ is a set of nodes (also called worlds), $A$ and $P$ are finite sets of actions and propositions, respectively, for every $a \in A$, $E_a \subseteq V \times V$, and for every $p \in P$, $U_p \subseteq V$. The tuple $\mathcal{F} = (V, (E_a)_{a \in A})$ is then also called a *Kripke frame*. We say that $\mathcal{K}$ (and $\mathcal{F}$) is *word-based* if $V = X^*$ for some finite alphabet $X$. Formulas of *multimodal logic* are defined by the following grammar, where $p \in P$ and $a \in A$:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Box_a \varphi \mid \Diamond_a \varphi.$$

The semantics $[\![\varphi]\!]_{\mathcal{K}} \subseteq V$ of formulas $\varphi$ in $\mathcal{K}$ is defined inductively as follows:

$$\begin{aligned}
[\![p]\!]_{\mathcal{K}} &= U_p, \\
[\![\neg\varphi]\!]_{\mathcal{K}} &= V \setminus [\![\varphi]\!]_{\mathcal{K}}, \\
[\![\varphi \wedge \psi]\!]_{\mathcal{K}} &= [\![\varphi]\!]_{\mathcal{K}} \cap [\![\psi]\!]_{\mathcal{K}}, \\
[\![\varphi \vee \psi]\!]_{\mathcal{K}} &= [\![\varphi]\!]_{\mathcal{K}} \cup [\![\psi]\!]_{\mathcal{K}}, \\
[\![\Box_a\varphi]\!]_{\mathcal{K}} &= \{v \in V \mid \forall u \in V : (v,u) \in E_a \to u \in [\![\varphi]\!]_{\mathcal{K}}\}, \\
[\![\Diamond_a\varphi]\!]_{\mathcal{K}} &= \{v \in V \mid \exists u \in V : (v,u) \in E_a \wedge u \in [\![\varphi]\!]_{\mathcal{K}}\}.
\end{aligned}$$

A word-based Kripke frame $\mathcal{F} = (X^*, (E_a)_{a\in A})$ is called *rational* if every $E_a$ is a rational transduction. Rational Kripke frames with a single relation are also known as *rational graphs* and have been studied intensively [6, 12, 13]. A word-based Kripke structure $\mathcal{K} = (X^*, (E_a)_{a\in A}, (U_p)_{p\in P})$ is called *rational* if every relation $E_a$ is a rational transduction and every $U_p$ is a regular language. The closure properties of regular languages imply that for every rational Kripke structure $\mathcal{K}$ and every multimodal formula $\varphi$, the set $[\![\varphi]\!]_{\mathcal{K}}$ is a regular language that can be effectively constructed from $\varphi$ and (automata describing the structure) $\mathcal{K}$. Using this fact, Bekker and Goranko [2] proved that the *model-checking problem* for rational Kripke structures and multimodal logic is decidable. This problem has as input a rational Kripke structure $\mathcal{K}$ (given by a tuple of automata and transducers), a word $w \in X^*$ (where $X^*$ is the node set of $\mathcal{K}$), and a multimodal formula $\varphi$, and it is asked whether $w \in [\![\varphi]\!]_{\mathcal{K}}$ holds. In contrast, there exist rational graphs (even acyclic ones) with an undecidable first-order theory [6, 15], but every rational tree has a decidable first-order theory [6]. Rational Kripke structures and frames were also considered in the context of querying graph databases [1].

Our reformulation of Theorem 9 in terms of multimodal logic is:

▶ **Theorem 14.** *Let $X = \{0,1\}$. There are rational transductions $E_r, E_s, E_t$ in $X^*$ such that the rational Kripke frame $\mathcal{F} = (X^*, E_r, E_s, E_t)$ has the following property: For every non-regular language $U_p \subseteq X^*$ and every language $K \in \mathsf{AH}(U_p)$, $K \subseteq X^*$, there exists a multimodal formula $\varphi$ such that $K = [\![\varphi]\!]_{\mathcal{K}}$, where $\mathcal{K} = (X^*, E_r, E_s, E_t, U_p)$.*

**Proof.** Take the rational transductions $R, S, T$ provided by Theorem 9. Let $U_p \subseteq X^*$ be a non-regular language and take the Kripke structure $\mathcal{K} = (X^*, E_r, E_s, E_t, U_p)$, where $E_r = R$, $E_s = S$, and $E_t = T$. By induction, we can construct for every language $K$ obtainable from $U_p$ by the transductions $R, S, T$ and Boolean operations a multimodal formula $\varphi$ with $K = [\![\varphi]\!]_{\mathcal{K}}$. For instance, if $K = [\![\psi]\!]_{\mathcal{K}}$, then $RK = [\![\Diamond_r\psi]\!]_{\mathcal{K}}$. The theorem follows immediately. ◀

The question arises whether an analogous statement holds when we allow choosing an arbitrary non-rational transduction instead of an arbitrary non-regular language. In other words: Are there rational transductions $R_1, \ldots, R_n$ and regular languages $L_1, \ldots, L_m$ over an alphabet $X$ such that for any non-rational transduction $T$, the Kripke structure $(X^*, R_1, \ldots, R_n, T, L_1, \ldots, L_m)$ allows to define every arithmetical language in multimodal logic? The answer is no, since there are non-rational transductions $T$ that preserve regularity, i.e., for which $TL$ is regular whenever $L$ is regular. Take, for example, the transduction $T = \{(w, ww) \mid w \in X^*\}$. It is clearly not rational, since $T^{-1}X^* = \{ww \mid w \in X^*\}$ is not regular. However, it is not hard to see that $TL$ is effectively regular for regular languages $L$ [17]. In particular, for every choice of $R_1, \ldots, R_n$ and $L_1, \ldots, L_m$ as above, every language definable in $(X^*, R_1, \ldots, R_n, T, L_1, \ldots, L_m)$ is regular and effectively constructible, implying that the model-checking problem is decidable.

## 5 Open problems

An interesting open problem is whether in Theorem 9 one can replace the rational transductions by suitable synchronized rational relations. A relation $R \subseteq X^* \times X^*$ is synchronized rational if the set of all convolutions $u \otimes v$ with $(u, v) \in R$ is a rational language. The convolution of two words $u = a_1 a_2 \cdots a_n$ and $v = b_1 b_2 \cdots b_m$ is the word $(a_1, b_1)(a_2, b_2) \cdots (a_k, b_k)$ where $k = \max\{n, m\}$, $a_i = \#$ for $i > n$, and $b_i = \#$ for $i > m$. Here, $\#$ is a fresh symbol not appearing in any pair from $R$. In other words, $R$ can be recognized by an automaton on two tapes where both heads move synchronously. Synchronized rational relations underly the definition of automatic structures [10]. Note that the rational transductions used in the proof of Theorem 9 are not synchronized rational.

Another open question is whether the number of rational transductions in Theorem 9 can be reduced to 1 or 2.

## References

1 P. Barceló, D. Figueira, and L. Libkin. Graph logics with rational relations and the generalized intersection problem. In *LICS*, pages 115–124. IEEE, 2012.

2 W. Bekker and V. Goranko. Symbolic model checking of tense logics on rational Kripke models. In M. Archibald, V. Brattka, V. Goranko, and B. Löwe, editors, *ILC*, volume 5489 of *Lecture Notes in Computer Science*, pages 2–20. Springer, 2007.

3 J. Berstel. *Transductions and context-free languages.* Teubner, Stuttgart, 1979.

4 R. V. Book. Simple representations of certain classes of languages. *Journal of the ACM*, 25(1):23–31, 1978.

5 J. Brzozowski. Properties of regular expressions and state diagrams. *Princeton University, Dept. of Electrical Engineering, Digital Systems Laboratory, Technical Report*, (15), 1962.

6 A. Carayol and C. Morvan. On rational trees. In Z. Ésik, editor, *CSL*, volume 4207 of *Lecture Notes in Computer Science*, pages 225–239. Springer, 2006.

7 H. Fernau and R. Stiebe. Sequential grammars and automata with valences. *Theoretical Computer Science*, 276:377–405, 2002.

8 S. A. Greibach. Remarks on blind and partially blind one-way multicounter machines. *Theoretical Computer Science*, 7(3):311–324, 1978.

9 J. Hartmanis and J. Hopcroft. What makes some language theory problems undecidable. *Journal of Computer and System Sciences*, 4(4):368–376, 1970.

10 B. Khoussainov and A. Nerode. Automatic presentations of structures. In *LCC: International Workshop on Logic and Computational Complexity*, volume 960 of *Lecture Notes in Computer Science*, pages 367–392. Springer, 1995.

11 D. C. Kozen. *Automata and computability.* Springer-Verlag, New York, 1997.

12 C. Morvan. On rational graphs. In J. Tiuryn, editor, *Proceedings of the 3rd International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2000), Berlin (Germany)*, number 2303 in Lecture Notes in Computer Science, pages 252–266. Springer, 2000.

13 C. Morvan and C. Stirling. Rational graphs trace context-sensitive languages. In J. Sgall, A. Pultr, and P. Kolman, editors, *Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science (MFCS 2001), Marianske Lazne (Czech Republic)*, number 2136 in Lecture Notes in Computer Science, pages 548–559. Springer, 2001.

14 E. Render. *Rational Monoid and Semigroup Automata.* PhD thesis, University of Manchester, 2010.

15 W. Thomas. A short introduction to infinite automata. In W. Kuich, G. Rozenberg, and A. Salomaa, editors, *Proceedings of the 5th International Conference on Developments in Language Theory (DLT 2001), Vienna (Austria)*, number 2295 in Lecture Notes in Computer Science, pages 130–144. Springer, 2001.

16 G. Zetzsche. On the capabilities of grammars, automata, and transducers controlled by monoids. In L. Aceto, M. Henzinger, and J. Sgall, editors, *Automata, Languages and Programming 38th International Colloquium, ICALP 2011, Zürich, Switzerland, July 4-8, 2011, Proceedings, Part II*, volume 6756 of *Lecture Notes in Computer Science*, pages 222–233. Springer, 2011.

17 G.-Q. Zhang. Automata, boolean matrices, and ultimate periodicity. *Information and Computation*, 152(1):138–154, 1999.