Non-Emptiness Test for Automata over Words Indexed by the Reals and Rationals⋆

Bernard Boigelot[0009−0009−4721−3824], Pascal Fontaine[0000−0003−4700−6031], and Baptiste Vergain[0009−0003−5545−4579]

Montefiore Institute, B28, University of Liège, Belgium
{Bernard.Boigelot, Pascal.Fontaine, BVergain}@uliege.be

Abstract. Automata have been defined to recognize languages of words indexed by linear orderings, which generalize the usual notions of finite, infinite, and ordinal words. The reachability problem for these automata has already been solved for scattered linear orderings.
In this paper, we design an analogous procedure that solves reachability over the specific domains $\mathbb{R}$ and $\mathbb{Q}$. Given an automaton on linear orderings, this procedure decides in polynomial time whether this automaton accepts at least one word indexed by $\mathbb{R}$ or by $\mathbb{Q}$. We claim that this algorithm constitutes an essential step to designing effective decision procedures for the first-order monadic theory of order interpreted over $\mathbb{R}$ or $\mathbb{Q}$.

Keywords: Automata · Linear Orderings · Real Domain · Rational Domain · Emptiness Test · Reachability

1 Introduction

In [3], Bruyère and Carton introduce automata on words indexed by linear orderings, which generalize the concept of word, and notably encapsulate the usual notions of finite, infinite, and ordinal words. Although initially restricted to scattered linear orderings (i.e., orderings that do not contain a dense subordering), these automata have later been extended to deal with all linear orderings [2]. Notably, a Kleene-like theorem asserting the equivalence between languages accepted by automata on linear orderings and languages described by an extended form of rational expressions is proved in [3, 2]. The strong connections between rational languages and the monadic second-order theory of linear orderings MSO($<$) is investigated in [1], namely, every language accepted by an automaton on linear orderings is definable in MSO($<$). The converse however only holds when the orderings are countable and scattered.

We are interested in designing effective decision procedures for monadic first-order theories of order MFO(D, $<$) interpreted over a fixed dense domain D, in particular $\mathbb{R}$ and $\mathbb{Q}$. The decidability of MFO($\mathbb{Q}$, $<$) derives directly from the

⋆ This work is partially supported by the FNRS-DFG PDR Weave (SMT-ART) grant 40019202.

decidability of MSO($\mathbb{Q}$, $<$) [15]. For $\mathbb{R}$ however, Shelah [19] showed that MSO($\mathbb{R}$, $<$) is undecidable. The decidability of MFO($\mathbb{R}$, $<$) has been established in [6]. Some theoretical bounds for the complexity of decision procedures for MFO($\mathbb{Q}$, $<$) and MFO($\mathbb{R}$, $<$) have been obtained in [17, 16]. Our long-term goal consists in designing an automata-based decision procedure in the spirit of the one introduced by Büchi to deal with S1S [4], which roughly corresponds to MSO($\mathbb{N}$, $<$), later extended to deal with all words indexed by countable ordinals [5]. The first step would consist in building an automaton recognizing the models of a given formula without any restriction on the domain. Then, a dedicated procedure would decide the emptiness of the language over a specific linear ordering, equal to $\mathbb{R}$ or $\mathbb{Q}$ in our case. This work focuses on the latter part.

Testing for the non-emptiness of the language accepted by an automaton on linear orderings reduces to deciding whether an accepting state is reachable from an initial state. Reachability for automata on scattered linear orderings is discussed in [9]. A generalization over every linear ordering is sketched in [11]. In this paper, we investigate how to design an analogous procedure that decides reachability over the domains $\mathbb{R}$ and $\mathbb{Q}$.

The paper is structured as follows. We first recall in Section 2 the useful definitions relating to linear orderings, rational expressions, and automata on linear orderings. In Section 3, we discuss the state of the art regarding the reachability problem. In Section 4, we provide an algorithm for deciding emptiness over $\mathbb{R}$ in polynomial time. Finally, in Section 5, we show how to adapt the previous algorithm to deal with $\mathbb{Q}$.

2 Preliminaries

2.1 Linear Orderings

We first give basic definitions and results about linear orderings, and refer to [18] for further details.

A *linear ordering* J is a totally ordered set, i.e., a set equipped with a binary relation $<_J$ that is irreflexive, asymmetric, transitive, and total. Two linear orderings J and K — respectively associated with the order relations $<_J$ and $<_K$ — are *order-isomorphic* if there exists an order-preserving bijection between J and K. Formally, let b be such a bijection (from J to K), then $b(j_1) <_K b(j_2)$ iff $j_1 <_J j_2$ for all $j_1, j_2 \in J$. We denote by $-J$ the *backwards* linear ordering that corresponds to J with its ordering reversed. The class of all linear orderings is denoted by $\mathcal{L}$.

The *order type* of a linear ordering J is the class of all linear orderings order-isomorphic to J. The order types of a singleton, the set composed of the N first natural numbers, $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ are respectively denoted by 1, N, ω, ζ, η, and λ. Notice that the order type of any non-empty open interval of $\mathbb{R}$ is λ, and that the order type of any non-empty open interval of $\mathbb{Q}$ is η.

The concatenation of two linear orderings J and K (with respective order relations $<_J$ and $<_K$) is denoted by $J + K$. It corresponds to the linear ordering

Fig. 1: The linear ordering $\mathbb{N}+\mathbb{Z}$.

composed of the set of pairs $\{(j,1)\,|\,j \in J\} \cup \{(k,2)\,|\,k \in K\}$, equipped with the order relation $<$ defined by $(j_1,1) < (j_2,1)$ if $j_1 <_J j_2$, $(k_1,2) < (k_2,2)$ if $k_1 <_K k_2$, and $(j,1) < (k,2)$ for every $j \in J$ and $k \in K$.

More generally, let K and J_k for all $k \in K$ be linear orderings. The linear ordering $\Sigma_{k\in K} J_k$ is obtained by concatenating the orderings J_k w.r.t. the ordering K. Formally, the *sum* $\Sigma_{k\in K} J_k$ is the linear ordering defined over the set of pairs (j,k) such that $j \in J_k$ and $k \in K$, with the order relation $(j_1,k_1) < (j_2,k_2)$ iff either $k_1 <_K k_2$, or $k_1 = k_2$ and $j_1 <_{J_{k_1}} j_2$. When $J_k = J$ for every $k \in K$, we may write J^K in place of $\Sigma_{k\in K} J_k$. These operators naturally extend to order types. For instance, the order type ω^ω is the class of all linear orderings order-isomorphic to the linear ordering $\mathbb{N}^\mathbb{N}$.

A *cut* of a linear ordering J is a partition of J into two sets (K,L) such that for every pair $(k,\ell) \in K \times L$, one has $k < \ell$. The set of all cuts of an ordering J is denoted by $\widehat{J}$. Notice that $\widehat{J}$ is a linear ordering as well, with $(K_1,L_1) < (K_2,L_2)$ iff $K_1 \subsetneq K_2$, and that $\widehat{J}$ always has a first element $(\emptyset, J)$ and a last element $(J,\emptyset)$. For every $j \in J$, the *consecutive* cuts $(\{k \mid k < j\}, \{\ell \mid j \leq \ell\})$ and $(\{k \mid k \leq j\}, \{\ell \mid j < \ell\})$ are (respectively) denoted by j^- and j^+. Let $c = (K,L)$ be a cut that is neither the first nor the last one of J, i.e., $K \neq \emptyset$ and $L \neq \emptyset$. If K admits a greatest element and L admits a smallest element, then c is called a *jump*. If K does not admit a greatest element and L does not admit a smallest element, then c is called a *gap*.

A linear ordering J is *(Dedekind) complete* if the set $\widehat{J}$ does not contain any gap. For instance, $\mathbb{N}$ and $\mathbb{R}$ are complete, while $\mathbb{Q}$ is incomplete. A linear ordering J is *dense* if between any two distinct elements lies another one, i.e., for every $j_1, j_2 \in J$ such that $j_1 <_J j_2$, there exists j_3 such that $j_1 <_J j_3 <_J j_2$. A linear ordering is *scattered* if it does not contain any dense sub-ordering. Given a dense linear ordering J, a sub-ordering $J' \subseteq J$ is said to be *dense in* J if between any two distinct elements of J lies an element of J'. For instance, $\mathbb{Q}$ and $\mathbb{R} \setminus \mathbb{Q}$ are both dense in $\mathbb{R}$.

Example 1. Consider the linear ordering $\mathbb{N}+\mathbb{Z}$ corresponding to the set of natural numbers followed by the set of integers. A depiction of this set and its cuts is given in Figure 1. The elements of this ordering are represented as dots, while the cuts are represented as vertical lines. Notice that this ordering admits a single gap, hence it is incomplete. It is also scattered.

2.2 Words and Rational Expressions

We now define the notions of words and rational expressions on linear orderings, as introduced in [3, 2].

A *word indexed by a linear ordering* is a totally ordered sequence of letters. Formally, given an alphabet Σ and a linear ordering J, a word $w = (\alpha_j)_{j\in J}$ indexed by J is a function $w : J \to \Sigma$. The *length* $|w|$ of w is J itself. The *empty word* ε denotes the word indexed by the empty set. Two words $w_1 = (\alpha_j)_{j\in J}$ and $w_2 = (\beta_k)_{k\in K}$ are equal if there exists an order-preserving bijection b from J to K such that for all $j \in J$, $\alpha_j = \beta_{b(j)}$. Informally, for a given word w, only the order type of its underlying linear ordering $|w|$ is relevant, not the linear ordering $|w|$ itself.

We now define a product operator on words. Let K and J_k for all $k \in K$ be linear orderings. Let $w_k = (\alpha_{j,k})_{j\in J_k}$ be a word of length J_k, for every $k \in K$. The *product* $\Pi_{k\in K} w_k$ is the word w of length $|w| = \Sigma_{k\in K} J_k$ equal to $(\alpha_{j,k})_{(j,k)\in|w|}$. The product of two words w_1 and w_2 is denoted by $w_1 \cdot w_2$.

A notion of rational expressions has been introduced in [3, 2] to describe languages of words indexed by linear orderings. Recall that in this paper, we consider a fixed domain that is either $\mathbb{R}$ or $\mathbb{Q}$. For this reason, we only introduce the rational operators that are relevant to describe words indexed by $\mathbb{R}$ or $\mathbb{Q}$.

Let X and Y be two sets of words on linear orderings. We define the following operators:

- *Concatenation*: $X \cdot Y = \{x \cdot y \mid x \in X, y \in Y\}$,
- *Infinite repetition*: $X^\omega = \{\Pi_{i\in\mathbb{N}}\ x_i \mid \forall i \in \mathbb{N},\ x_i \in X\}$,
- *Reverse infinite repetition*: $X^{-\omega} = \{\Pi_{i\in(-\mathbb{N})}\ x_i \mid \forall i \in -\mathbb{N},\ x_i \in X\}$.

We also define a *shuffle* operator that generates density. Let $\Sigma^\diamond$ denote the set of all the words indexed by any linear ordering[1] over the alphabet Σ, i.e., $\Sigma^\diamond = \{w \mid |w| \in \mathcal{L}\}$.

Let $X_1, \ldots, X_n \subseteq \Sigma^\diamond$, for some $n \geq 1$. We denote by $\mathrm{sh}(X_1, \ldots, X_n)$ the set of words of $\Sigma^\diamond$ that can be written as $\Pi_{r\in R}\ x_r$ such that (1) R is a non-empty, dense, and complete linear ordering without first or last element, and (2) there exists a partition $\{R_1, \ldots, R_n\}$ of R such that every R_i is dense in R, and for every $r \in R$ and $i \in [1, n]$, if $r \in R_i$ then $x_r \in X_i$.

Some remarks are in order. Notice that the order type of the dense and complete ordering R mentioned in the shuffle definition can be different from λ. This will be further discussed in Section 4.1. Also notice that, as mentioned in [2], although the shuffle operator relies on a complete ordering R, if there exists $i \in [1, n]$ such that $\varepsilon \in X_i$ (but $\bigcup_{i\in[1,n]} X_i \neq \{\varepsilon\}$), then there exists $w \in \mathrm{sh}(X_1, \ldots, X_n)$ such that $|w|$ is incomplete. Intuitively, every element of R_i corresponding to ε does not belong to $|w|$, but matches a gap in $\widehat{|w|}$.

2.3 Automata

We now recall definitions initially introduced in [3].

[1] This definition slightly differs from what is commonly found in the literature: it implies that $\varepsilon \in \Sigma^\diamond$ holds, while the usual definition excludes the empty word from the set $\Sigma^\diamond$.

Definition 1. *An* automaton on linear orderings *is a tuple* $\mathcal{A} = (Q, \Sigma, I, F, \Delta^s, \Delta^\ell)$*, where Q is a finite set of states, Σ is a finite alphabet, $I \subseteq Q$ is a set of initial states, $F \subseteq Q$ is a set of accepting states, $\Delta^s \subseteq Q \times \Sigma \times Q$ is a set of* successor transitions*, and $\Delta^\ell \subseteq (2^Q \times Q) \cup (Q \times 2^Q)$ is a set of (*left *and* right*)* limit transitions*.*

The set Δ^s contains *successor transitions* of the form (q, α, q'), with $q, q' \in Q$ and $\alpha \in \Sigma$, alternatively written $q \xrightarrow{\alpha} q'$, which are similar to the transitions of finite and infinite-word automata. Notice however that automata on linear orderings do not admit ε-transitions. The set Δ^ℓ is composed of *left-limit transitions* of the form (P, q), with $P \subseteq Q$ and $q \in Q$, written $P \to q$, and *right-limit transitions* of the form (q, P), with $q \in Q$ and $P \subseteq Q$, written $q \to P$. We sometimes write sets of limit transitions in a more compact form, e.g., we write $p, q \to \{p, q\} \to p$ in place of the set $\{p \to \{p, q\}, q \to \{p, q\}, \{p, q\} \to p\}$.

We now define the notion of *path* in such an automaton. Let J be a linear ordering, Q a finite set of states, and $\pi = (q_c)_{c \in \widehat{J}}$ a word over the alphabet Q (i.e., each cut $c \in \widehat{J}$ is associated with an element of Q). The *left-limit set* and *right-limit set* of π at c are the two subsets $\lim_c^- \pi$ and $\lim_c^+ \pi$ of Q defined as follows:

- $\lim_c^- \pi = \{q \in Q \mid \forall i < c,\ \exists k : i < k < c \wedge q = q_k\}$,
- $\lim_c^+ \pi = \{q \in Q \mid \forall c < i,\ \exists k : c < k < i \wedge q = q_k\}$.

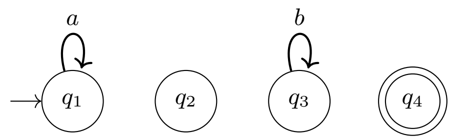
The left limit at c is therefore the set of states that can be found before c and infinitely close to c in π. If c has a predecessor, then this set is empty (intuitively it means that a *successor transition*, and not a *limit transition*, has to be taken to reach the state mapped to c). The case of a right limit is handled symmetrically. For the first cut c_{min} and the last cut c_{max} of J, we set $\lim_{c_{min}}^- \pi = \emptyset$, and $\lim_{c_{max}}^+ \pi = \emptyset$, although the definition above implies that both sets are equal to Q.

Definition 2. *Let $\mathcal{A} = (Q, \Sigma, I, F, \Delta^s, \Delta^\ell)$ be an automaton and $w = (\alpha_j)_{j \in J}$ a word of length J. A* path *π labeled by w is a sequence of states $\pi = (q_c)_{c \in \widehat{J}}$ of length $\widehat{J}$ such that*

- *For every $j \in J$, $(q_{j^-}, \alpha_j, q_{j^+})$ is a successor transition in Δ^s.*
- *For every cut $c \in \widehat{J}$ that is not the first cut and does not have a predecessor, $lim_c^- \pi \to q_c$ is a left-limit transition in Δ^ℓ.*
- *For every cut $c \in \widehat{J}$ that is not the last cut and does not have a successor, $q_c \to lim_c^+ \pi$ is a right-limit transition in Δ^ℓ.*

A path is *accepting* if it starts in a state $p \in I$ and ends in a state $q \in F$. A word is accepted by $\mathcal{A}$ if it labels an accepting path.

Note that a *finite-word automaton* can be seen as an automaton on linear orderings that has an empty set of limit transitions (i.e., $\Delta^\ell = \emptyset$). Infinite-word automata [4, 14] and automata on ordinal words [5, 10] also correspond to particular cases of automata on linear orderings.



$$\{q_1\} \to q_2$$
$$q_2 \to \{q_3\} \to q_4$$

Fig. 2: An automaton accepting $a^\omega \cdot b^{-\omega} \cdot b^\omega$.

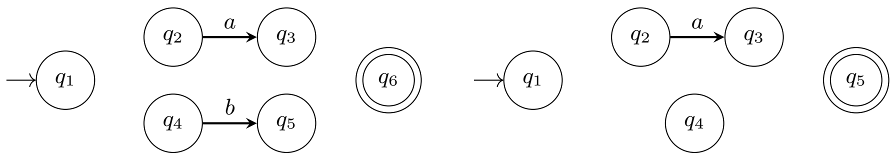
Example 2. Consider the word $w = a^\omega \cdot b^{-\omega} \cdot b^\omega$ defined over the alphabet $\{a, b\}$. Notice that w is indexed by the linear ordering considered in Example 1. The automaton depicted in Figure 2 accepts $\{w\}$. An accepting path can be described as follows: it maps the cuts (including the first one) of the first sub-ordering of order type ω to the state q_1, the gap to the state q_2, the cuts of the sub-ordering of order type $-\omega + \omega$ (or equivalently ζ) to the state q_3, and the last cut to the state q_4.

Example 3. Consider the automata depicted in Figures 3a and 3b. Both accept a non-empty language. The automaton in Figure 3a accepts words indexed by $\mathbb{R}$, and more generally by any non-empty dense and complete ordering J with no first and no last element, as long as the set of indices labeled by the letter a and those labeled by b partition J, and are both dense in J. For instance, consider the word $w : \mathbb{R} \to \{a, b\}$, such that $w(x) = a$ if $x \in \mathbb{Q}$, and $w(x) = b$ if $x \in \mathbb{R} \setminus \mathbb{Q}$. Let us describe an accepting path π labeled by w. The first cut $(\emptyset, \mathbb{R})$ of $\mathbb{R}$ is abbreviated by $-\infty$, and the last cut $(\mathbb{R}, \emptyset)$ by $+\infty$. We set $\pi(-\infty) = q_1$ and $\pi(+\infty) = q_6$. Since $\mathbb{R}$ is complete, every other cut of $\mathbb{R}$ is of the form r^- or r^+ for some $r \in \mathbb{R}$. If $r \in \mathbb{Q}$, then we set $\pi(r^-) = q_2$ and $\pi(r^+) = q_3$. If $r \in \mathbb{R} \setminus \mathbb{Q}$, then we set $\pi(r^-) = q_4$ and $\pi(r^+) = q_5$. The automaton depicted in Figure 3a does not however accept words indexed by $\mathbb{Q}$, or any incomplete ordering, since this automaton does not contain any state that can be mapped to a gap. The language accepted by this automaton is described by the rational expression $\mathrm{sh}(a, b)$.

On the other hand, the automaton depicted in Figure 3b accepts words indexed by $\mathbb{Q}$, but does not accept any word indexed by $\mathbb{R}$. Indeed, any accepting path necessarily visits the state q_4, which can only correspond to a gap, since it does not admit any incoming or outgoing successor transition. The language accepted by this automaton is described by the rational expression $\mathrm{sh}(a, \varepsilon)$.

3 State of the Art

Given an automaton $\mathcal{A}$ and two states p and q, the reachability problem consists in deciding whether $\mathcal{A}$ admits a path starting at p and ending at q. In [9],



$q_1, q_3, q_5 \to \{q_2, q_3, q_4, q_5\} \to q_2, q_4, q_6$ $\quad$ $q_1, q_3, q_4 \to \{q_2, q_3, q_4\} \to q_2, q_4, q_5$

(a) A complete shuffle of a and b. (b) An incomplete shuffle of a.

Fig. 3: Automata accepting words indexed by dense orderings.

Carton describes a polynomial procedure for deciding reachability on scattered linear orderings only, i.e., deciding whether there exists a path labeled by a word indexed by such an ordering. Given an input automaton on linear orderings $\mathcal{A}$, Carton's algorithm generates a finite-word automaton $\mathcal{A}_F$ sharing the same set of states as $\mathcal{A}$, that accounts for every possible path in $\mathcal{A}$ while storing the set of states visited in the labels read along the paths. More precisely, the existence of a transition (p, P, q) in $\mathcal{A}_F$ implies that there exists in $\mathcal{A}$ a path $p \to q$ labeled by a word indexed by a scattered linear ordering that visits exactly the set of states P. Reciprocally, the existence in $\mathcal{A}$ of a path $p \to q$ labeled by a word indexed by a scattered linear ordering implies the existence of a path $p \to q$ in $\mathcal{A}_F$, although it may not necessarily be composed of a single transition. By not explicitly building a transition for every possible path, Carton's solution remains polynomial. Reachability in $\mathcal{A}$ then reduces to reachability in $\mathcal{A}_F$.

A generalization of this algorithm over arbitrary linear orderings is sketched in [11], where the author introduces a specific additional rule for dealing with shuffles. To the best of our knowledge, a proof of this generalization has not been published, and it seems that the additional rule is actually only able to deal with complete shuffles. Actually, a small modification of this rule (somewhat similar to what will be done in Section 5) is enough to deal with both complete and incomplete shuffles altogether, so as to accurately cover any linear ordering.

4 Reachability over the Reals

4.1 Characterization

Properties like completeness, density, and not having first or last elements can be expressed by automata on linear orderings. However, there is no equivalence between accepting words indexed by complete and dense orderings without first or last elements, and accepting words indexed by $\mathbb{R}$. In particular, we have the following result.

Theorem 1. *There exists an automaton on linear orderings $\mathcal{A}_{\mathbb{R}}^{\emptyset}$ that accepts words indexed by non-empty, dense, and complete linear orderings that do not have first or last elements, such that $\mathcal{A}_{\mathbb{R}}^{\emptyset}$ does not accept words indexed by $\mathbb{R}$.*

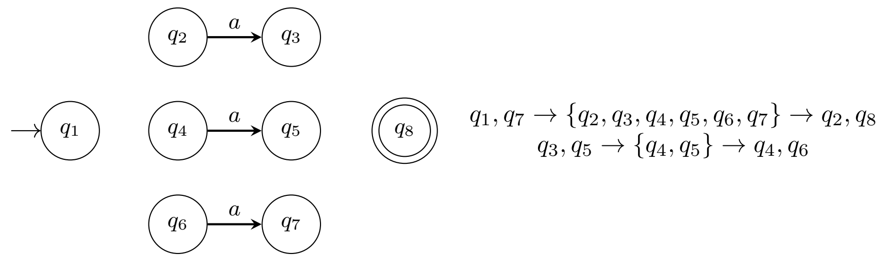


Fig. 4: The automaton $\mathcal{A}^{\emptyset}_{\mathbb{R}}$, that does not accept words indexed by $\mathbb{R}$.

The automaton $\mathcal{A}^{\emptyset}_{\mathbb{R}}$ depicted in Figure 4 matches the description given in Theorem 1. The proof of Theorem 1 ensues from the two following lemmas.

Lemma 1. *The automaton $\mathcal{A}^{\emptyset}_{\mathbb{R}}$ does not accept any word indexed by $\mathbb{R}$.*

Proof sketch. Let w be a word accepted by $\mathcal{A}^{\emptyset}_{\mathbb{R}}$. By studying the structure of the automaton $\mathcal{A}^{\emptyset}_{\mathbb{R}}$, one first proves that the ordering $|w|$ must be of the form $\Sigma_{k \in K} J_k$, where for all $k \in K$, J_k is an infinite, dense, and complete ordering with a first and a last element, and K is an infinite, dense, and complete ordering without first and last elements. Intuitively, each J_k is recognized by a sub-path that starts with $q_2 \xrightarrow{a} q_3$, then follows a dense mix of $q_4 \xrightarrow{a} q_5$ transitions, and finally ends with $q_6 \xrightarrow{a} q_7$.

Then, one shows that $\mathbb{R}$ cannot be expressed as such an ordering $\Sigma_{k \in K} J_k$. By contradiction, assume that $\mathbb{R}$ can be partitioned into an infinite, dense, and complete set I of pairwise disjoint closed intervals $[a_i, b_i]$ with $a_i < b_i$. On the one hand, thanks to Cantor's isomorphism theorem [8], there does not exist an infinite, dense, and complete ordering that is also countable, therefore the set I must be uncountable. On the other hand, I must be countable since each interval $[a_i, b_i] \in I$ contains a rational number that is not shared with any other, which yields a contradiction. □

Lemma 2. *The automaton $\mathcal{A}^{\emptyset}_{\mathbb{R}}$ accepts a word indexed by a non-empty, dense, and complete linear ordering that does not have a first or a last element.*

Proof. Consider the linear ordering $S = [0, 1]^{\mathbb{R}}$, where $[0, 1]$ denotes the closed interval of $\mathbb{R}$ between 0 and 1. Intuitively, S corresponds to the real line in which every real number has been replaced by a copy of the closed interval $[0, 1]$. The ordering S is dense, complete, and does not admit a first or a last element. We can represent elements of S as pairs $(x, y) \in [0, 1] \times \mathbb{R}$, ordered by the relation $<_S$ defined by $(x_1, y_1) <_S (x_2, y_2)$ if either $y_1 < y_2$, or $y_1 = y_2$ and $x_1 < x_2$, where $<$ denotes the usual order relation on both $\mathbb{R}$ and $[0, 1]$. Now, consider the word w indexed by S (i.e., $|w| = S$) obtained by labeling each element of S with the symbol a. Let us describe an accepting run of $\mathcal{A}^{\emptyset}_{\mathbb{R}}$ reading w. We denote by $\widehat{s}_{min}$ (resp. $\widehat{s}_{max}$) the first (resp. last) cut of S. Consider the mapping $\pi : \widehat{S} \to Q$ defined as follows:

– $\pi(\widehat{s}_{min}) = q_1$,
– $\pi(\widehat{s}_{max}) = q_8$,
– for all $y \in \mathbb{R}$, $\pi((0, y)^-) = q_2$ and $\pi((0, y)^+) = q_3$,
– for all $y \in \mathbb{R}$, $\pi((1, y)^-) = q_6$ and $\pi((1, y)^+) = q_7$,
– for all $x \in (0, 1)$ and for all $y \in \mathbb{R}$, $\pi((x, y)^-) = q_4$,
– for all $x \in (0, 1)$ and for all $y \in \mathbb{R}$, $\pi((x, y)^+) = q_5$.

It is immediate to establish that π describes an accepting run. □

This concludes the proof of Theorem 1.

4.2 Algorithm

We now introduce an algorithm that decides reachability over $\mathbb{R}$.

Let $\mathcal{A} = (Q, \Sigma, I, F, \Delta^s, \Delta^\ell)$ be an automaton and $q_I, q_F \in Q$ two states. We consider the problem of deciding reachability between q_I and q_F. The first step of the algorithm consists in constructing a sequence of finite-word automata $\mathcal{A}_0, \ldots, \mathcal{A}_M$, where $M = \max\{|P| \mid \exists q \in Q : q \to P \in \Delta^\ell \vee P \to q \in \Delta^\ell\}$. For every $j \in [0, M]$, the automaton $\mathcal{A}_j$ is of the form $(Q, 2^Q, I, F, \Delta_j, \emptyset)$, i.e., the successor transitions in Δ_j (which we define later) are labeled by subsets of Q. The purpose of each automaton $\mathcal{A}_j$ is to characterize every path in $\mathcal{A}$ that is labeled by a word indexed by $\mathbb{R}$, and that only involves limit transitions on sets of cardinality less than or equal to j.

We introduce the following useful definitions. An *open* $\mathbb{R}$*-path* $\overline{r}$ from q_1 to q'_n in $\mathcal{A}_j$ of label $P = P_1 \cup \cdots \cup P_n$ is a finite sequence of transitions (q_1, P_1, q'_1), …, $(q_n, P_n, q'_n) \in \Delta_j$ (of length $n \geq 1$) such that, for every $i \in [1, n-1]$, there exists a successor transition $(q'_i, \alpha_i, q_{i+1}) \in \Delta^s$ for some $\alpha_i \in \Sigma$. Intuitively, such a path corresponds to a finite concatenation of open intervals of $\mathbb{R}$ (corresponding to transitions in $\mathcal{A}_j$), connected by single elements (corresponding to successor transitions in $\mathcal{A}$), resulting in an open interval of $\mathbb{R}$. Analogously, a *closed* $\mathbb{R}$*-path* $\overline{r}$ from p to p' in $\mathcal{A}_j$ of label $P \cup \{p, p'\}$ is the combination of an open $\mathbb{R}$-path from a state q_1 to a state q'_n (of label P) with two transitions $(p, \alpha, q_1), (q'_n, \alpha', p') \in \Delta^s$, for some $\alpha, \alpha' \in \Sigma$.

The transition relation Δ_j is defined for $j \in [0, M]$ by recursion over j. For $j > 0$, $\Delta_j = \Delta_{j-1} \cup S_j \cup L_j \cup R_j$, where S_j, L_j, and R_j are sets of transitions constructed from $\mathcal{A}_{j-1}$ as described below, and $\Delta_0 = \emptyset$. For every $j \in [1, M]$, the sets S_j, L_j, and R_j are initially empty. They are then filled according to the following rules.

Shuffle rule: For every pair of transitions $q \to P \in \Delta^\ell$ and $P \to q' \in \Delta^\ell$ such that $|P| = j$, the transition $(q, P \cup \{q, q'\}, q')$ is added to S_j if and only if:

– There exists a (possibly empty) set $\{\overline{r}_1, \ldots, \overline{r}_n\}$ of closed $\mathbb{R}$-paths in $\mathcal{A}_{j-1}$ with $n \geq 0$, where each $\overline{r}_i$ starts in a state q_i such that $P \to q_i \in \Delta^\ell$, and ends in a state q'_i such that $q'_i \to P \in \Delta^\ell$. The label of the $\mathbb{R}$-path $\overline{r}_i$ is denoted by Q_i.

– There exists a (necessarily non-empty) set of successor transitions $\{(s_1,\gamma_1,s'_1), \ldots, (s_m, \gamma_m, s'_m)\} \subseteq \Delta^s$, with $m > 0$, such that, for each $i \in [1, m]$, $s_i, s'_i \in P$, $P \to s_i \in \Delta^\ell$, and $s'_i \to P \in \Delta^\ell$.
– $\bigcup_{i\in[1,m]} \{s_i, s'_i\} \cup \bigcup_{i\in[1,n]} Q_i = P$.

We refer to the case $n = 0$ as a *simple shuffle*, and to the case $n > 0$ as a *Cantor shuffle*. Intuitively, a Cantor shuffle represents a dense combination of intervals of order type $1 + \lambda + 1$ (corresponding to closed $\mathbb{R}$-paths in $\mathcal{A}_{j-1}$) and single elements (corresponding to successor transitions in $\mathcal{A}$). A simple shuffle represents a dense combination of single elements only.

Infinite repetition rule: For every transition $P \to q \in \Delta^\ell$ such that $|P| = j$, and for every state $s \in P$, the transition $(s, P \cup \{q\}, q)$ is added to L_j if and only if there exist $p, p' \in P$ such that:

– $\mathcal{A}_{j-1}$ admits an open $\mathbb{R}$-path from p to p' labeled by P of the form $(r_1, P_1, r'_1), \ldots, (r_n, P_n, r'_n)$ such that $s = r_i$ for some $i \in [1, n]$, and
– $\mathcal{A}$ admits a successor transition of the form $(p', \alpha, p) \in \Delta^s$.

Intuitively, we look for a cycle that alternates between open intervals of order type λ (corresponding to open $\mathbb{R}$-paths in $\mathcal{A}_{j-1}$) and single elements (corresponding to successor transitions in $\mathcal{A}$) such that the set of states visited by the whole cycle is exactly P. If such a cycle exists, it can be repeated infinitely many times, therefore allowing to follow the limit transition $P \to q$. A similar principle applies to the *reverse infinite repetition rule* described below.

Reverse infinite repetition rule: For every transition $q \to P \in \Delta^\ell$ such that $|P| = j$, and for every state $s \in P$, the transition $(q, P \cup \{q\}, s)$ is added to R_j if and only if there exist $p, p' \in P$ such that:

– $\mathcal{A}_{j-1}$ admits an open $\mathbb{R}$-path from p to p' labeled by P of the form $(r_1, P_1, r'_1), \ldots, (r_n, P_n, r'_n)$ such that $s = r'_i$ for some $i \in [1, n]$, and
– $\mathcal{A}$ admits a successor transition of the form $(p', \alpha, p) \in \Delta^s$.

The final step of the algorithm consists in searching whether there exists an open $\mathbb{R}$-path in $\mathcal{A}_M$ that starts in q_I and ends in q_F. In the positive case, the algorithm returns *yes*, otherwise *no*.

The definitions above ensure that for every $j \in [0, M]$, if $\mathcal{A}$ admits a path from a state q to a state q' that visits a set of states P, is labeled by a word indexed by $\mathbb{R}$, and only involves limit transitions on sets of cardinality less than or equal to j, then $\mathcal{A}_j$ admits an open $\mathbb{R}$-path from q to q' labeled by P.

4.3 Correctness

We have the two following results, asserting the correctness of the algorithm of Section 4.2.

Theorem 2. *If there exists an open $\mathbb{R}$-path in $\mathcal{A}_M$ from q_I to q_F of label P, then there exists a path in $\mathcal{A}$ from q_I to q_F, labeled by a word indexed by $\mathbb{R}$, that visits exactly the states in P.*

Proof sketch. The proof proceeds by explicitly building a word indexed by $\mathbb{R}$ or, equivalently, by any open interval (x, y) of $\mathbb{R}$, that labels a path in $\mathcal{A}$.

Let $\overline{r}$ be an open $\mathbb{R}$-path in $\mathcal{A}_M$ corresponding to the sequence of transitions $(q_1, P_1, q_1'), \ldots, (q_n, P_n, q_n') \in \Delta_M$. The proof is by induction on the sequence of rules involved in the generation of the transitions (q_i, P_i, q_i'). The base case corresponds to an open $\mathbb{R}$-path $\overline{r}$ composed of a single transition $(q, P, q') \in \Delta_M$ that stems from the *simple shuffle* rule, since this rule does not rely on preexisting $\mathbb{R}$-paths. Let $(s_1, \gamma_1, s_1'), \ldots, (s_m, \gamma_m, s_m') \in \Delta^s$ be the successor transitions that triggered the application of the simple shuffle rule. The domain $\mathbb{R}$ can be partitioned into m non-empty disjoint subsets $R_1, \ldots, R_m$ such that each R_i is dense in $\mathbb{R}$. The word $w : \mathbb{R} \to \Sigma$ indexed by $\mathbb{R}$ and defined by $w(x) = \gamma_i$ if $x \in R_i$ labels a path from q to q'.

For the inductive case, we first consider the *infinite repetition* rule. It is handled by considering an infinite sequence of words indexed by consecutive intervals of $\mathbb{R}$, e.g., $(-\infty, 0)$, $(0, 1)$, $(1, 2)$, $\ldots$ and concatenating them to form a word indexed by the full set $\mathbb{R}$. The case of the *reverse infinite repetition* is handled symmetrically.

To reason about the remaining rule, that is, the *Cantor shuffle*, we consider a Cantor set construction [7, 12] that partitions the interval $(0, 1)$ into a countable set I of disjoint open intervals, together with an uncountable set S of isolated points. For every $n > 0$, the set I can be partitioned into n subsets $I_1, \ldots, I_n$ such that for all $k \in [1, n]$, an interval in I_k lies between any two intervals in I. Similarly, for every $m > 0$, the set S can be partitioned into m subsets $S_1, \ldots, S_m$ such that for all $j \in [1, m]$, the set S_i is dense in S. By associating each $\mathbb{R}$-path $\overline{r}_i$ involved in the Cantor shuffle with the set of intervals I_i, and each successor transition (s_j, γ_j, s_j') with the set of isolated points S_j, we obtain a word indexed by $(0, 1)$ (and equivalently $\mathbb{R}$) that labels a path of the desired form. $\square$

Theorem 3. *If the automaton $\mathcal{A}$ accepts a word indexed by $\mathbb{R}$, then there exists an open $\mathbb{R}$-path in $\mathcal{A}_M$ from a state $q_I \in I$ to a state $q_F \in F$.*

The proof is essentially based on mechanisms introduced in [13] for establishing the decidability of the first-order theory of order, and on ideas introduced in [9].

Proof sketch. The proof consists in showing that for all $j \in [0, M]$, if $\mathcal{A}$ admits a path from a state q to a state q' that visits a set of states P, labeled by a word indexed by $\mathbb{R}$, and that only follows limit transitions on sets of cardinality less than or equal to j, then $\mathcal{A}_j$ admits an open $\mathbb{R}$-path from q to q' labeled by P.

The proof relies on the property that for all $j \in [1, M]$, the construction of $\mathcal{A}_j$ only relies on transitions generated in $\mathcal{A}_{j-1}$. This ensues from the fact that at Step j, every application of a rule generates a transition of one of the two following forms. The first case is that the generated transition is labeled by a set of states of cardinality strictly greater than j. In that case, this transition can clearly only be involved in a rule at a later Step $j+k$ for some $k \geq 1$. The other possibility is to have a transition labeled by a set of cardinality of size exactly equal to j. In that case, using this transition to apply another rule in $\mathcal{A}_j$ (this time not only based on transitions in $\mathcal{A}_{j-1}$, but also transitions generated at Step j itself) would only generate new transitions that are redundant w.r.t. the existence of open $\mathbb{R}$-paths in $\mathcal{A}_j$. □

4.4 Example

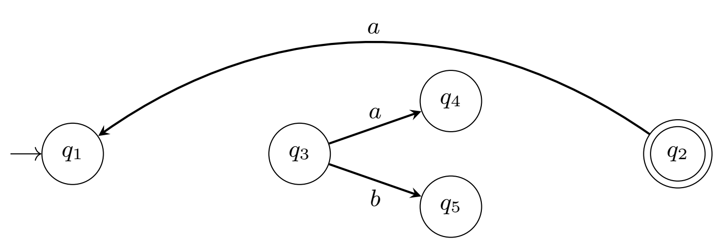
We now give a short example to illustrate our algorithm, and more specifically the construction of the automata $\mathcal{A}_0, \ldots, \mathcal{A}_M$. Consider the automaton $\mathcal{A}$ depicted in Figure 5. The automaton $\mathcal{A}_0$ is shown in Figure 6, and does not contain any transition. Since no limit transition involves a set of cardinality 1 or 2, we have $\mathcal{A}_2 = \mathcal{A}_1 = \mathcal{A}_0$. In the automaton $\mathcal{A}_3$ given in Figure 7, the transition $(q_4, \{q_3, q_4, q_5\}, q_3)$ (resp. $(q_5, \{q_3, q_4, q_5\}, q_3)$) results from the application of the shuffle rule on the set $\{q_3, q_4, q_5\}$, with $q_I = q_4$ (resp. $q_I = q_5$) and $q_F = q_3$. For the same reason as before, we have $\mathcal{A}_4 = \mathcal{A}_3$. The last automaton $\mathcal{A}_5$ is illustrated in Figure 8. The new transitions result from the application of the shuffle rule on the set $\{q_1, q_2, q_3, q_4, q_5\}$, for every $q_I \in \{q_1, q_4, q_5\}$ and $q_F \in \{q_2, q_3\}$.

4.5 Complexity

The algorithm described in Section 4.2 runs in polynomial time w.r.t. the size of $\mathcal{A}$, i.e. $|Q| + |\Delta^s| + |\Delta^\ell|$. More precisely, we have the following result.

Proposition 1. *The automaton $\mathcal{A}_M$ can be computed in polynomial time in the size of $\mathcal{A}$.*

Proof sketch. A first argument is that for all $j \in [0, M]$, each rule is called only polynomially many times during the construction of the automaton $\mathcal{A}_j$. Indeed, the shuffle rule is only considered for every combination of two limit transitions in $\mathcal{A}$, and the (reverse) infinite repetition is only considered for every combination of a limit transition in $\mathcal{A}$ with one state of $\mathcal{A}$.



$$q_4, q_5 \to \{q_3, q_4, q_5\} \to q_3$$
$$q_1, q_4, q_5 \to \{q_1, q_2, q_3, q_4, q_5\} \to q_2, q_3$$

Fig. 5: The input automaton $\mathcal{A}$.

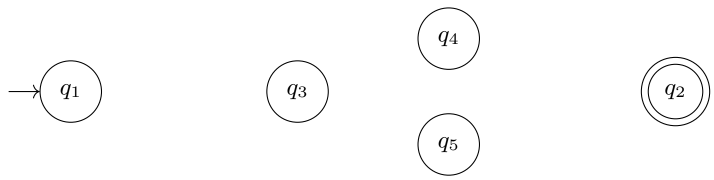


Fig. 6: The finite-word automata $\mathcal{A}_0$, $\mathcal{A}_1$, and $\mathcal{A}_2$.

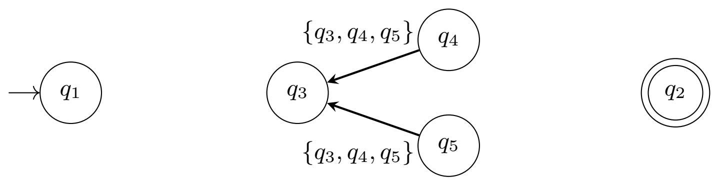
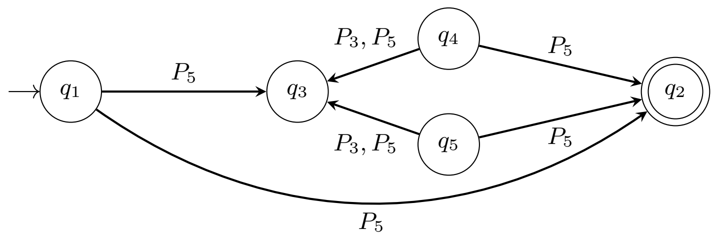


Fig. 7: The finite-word automata $\mathcal{A}_3$ and $\mathcal{A}_4$.

It remains to show that each rule can be applied in polynomial time. The case of the simple shuffle rule is immediate. The other rules require to either be able to check the existence of an open $\mathbb{R}$-path labeled by a given set of states P (to deal with the infinite and reverse infinite repetition rules), and to be able to check the existence of a set of closed $\mathbb{R}$-paths and successor transitions such that their combined labels and visited states are equal to P (to deal with the shuffle rule). To perform these two operations in polynomial time, in this specific context, we rely on a polynomial procedure that, given a set of states P, two states $q, q' \in P$, and the automaton $\mathcal{A}_j$, computes the union of the labels of every possible (open or closed) $\mathbb{R}$-path in $\mathcal{A}_j$ from q to q' labeled by a subset of P. More precisely, this procedure computes the set $S_{\text{open}}(\mathcal{A}_j, P, q, q') = \{r \in P \mid \mathcal{A}_j$ admits an open $\mathbb{R}$-path from q to q', of label $R \subseteq P$, s.t. $r \in R\}$, or analogously the set $S_{\text{closed}}(\mathcal{A}_j, P, q, q')$ that deals with closed $\mathbb{R}$-paths. Let $\delta_{P,j,q}^{\rightarrow} \subseteq \Delta_j$ (resp. $\delta_{P,j,q'}^{\leftarrow}$) be the set of transitions occurring in (open or closed) $\mathbb{R}$-paths of $\mathcal{A}_j$ starting in q (resp. ending in q'), and labeled by a subset of P. The sets $\delta_{P,j,q}^{\rightarrow}$ and $\delta_{P,j,q'}^{\leftarrow}$ can be computed in polynomial time by performing a (reverse) traversal of the automaton $\mathcal{A}_j$ that respects the structure of $\mathbb{R}$-paths, i.e., alternating between transitions in Δ_j and transitions in Δ^s. The set $S_{\text{open}}(\mathcal{A}_j, P, q, q')$ (resp. $S_{\text{closed}}(\mathcal{A}_j, P, q, q')$) is then obtained by taking the union of the labels of the transitions in $\delta_{P,j,q}^{\rightarrow} \cap \delta_{P,j,q'}^{\leftarrow}$.

Thanks to the (reverse) infinite repetition rule, given a set of states P and two states $p, p' \in P$ such that $S_{\text{open}}(\mathcal{A}_j, P, p, p') = P$, there exists an open $\mathbb{R}$-path from p to p' in $\mathcal{A}_j$ labeled by P. Such a path can be constructed by first choosing, for each $q \in P$, an open $\mathbb{R}$-path $\overline{r}_q$ labeled by $R \subseteq P$ such that $q \in R$, and then concatenating these paths by means of transitions of the form $(p', \alpha, p) \in \Delta^s$, which must exist for the infinite repetition rule to be applicable.

Now, regarding the shuffle rule, consider two set of states P, S such that $S \subseteq P$. We define $L_P = \{p \in P \mid P \rightarrow p \in \Delta^\ell\}$, and $R_P = \{p \in P \mid p \rightarrow P \in \Delta^\ell\}$.



with $P_5 = \{q_1, q_2, q_3, q_4, q_5\}$, and $P_3 = \{q_3, q_4, q_5\}$.

Fig. 8: The finite-word automaton $\mathcal{A}_5$.

Using a similar argument as before, one shows that checking the existence of a set of closed $\mathbb{R}$-paths $\overline{r}_1, \ldots, \overline{r}_n$, each labeled by Q_i, starting from a state in L_P, and ending in a state in R_P, such that $(P \setminus S) \subseteq \bigcup_{i \in [1,n]} Q_i$, reduces to checking that the inclusion $(P \setminus S) \subseteq \bigcup_{(q,q') \in L_P \times R_P} S_{\text{closed}}(\mathcal{A}_j, P, q, q')$ holds. □

5 Reachability over the Rationals

The algorithm introduced in Section 4.2 can be adapted to solve reachability over $\mathbb{Q}$. The incompleteness of $\mathbb{Q}$ requires the rules to be modified, in order to account for the presence of gaps.

We first define a notion of valid path over $\mathbb{Q}$, analogous to the notion of $\mathbb{R}$-path. A $\mathbb{Q}$-path from q_1 to q'_n in $\mathcal{A}_j$ of label $P = P_1 \cup \cdots \cup P_n$ is a finite sequence of transitions (q_1, P_1, q'_1), …, $(q_n, P_n, q'_n) \in \Delta_j$ such that for each $i \in [1, n-1]$, either $q'_i = q_{i+1}$, or there exists a successor transition $(q'_i, \alpha_i, q_{i+1}) \in \Delta^s$. Intuitively, such a path corresponds to a finite concatenation of open intervals of $\mathbb{Q}$ (corresponding to transitions in $\mathcal{A}_j$) connected either by single elements (corresponding to successor transitions in $\mathcal{A}$, if the shared bound between the two consecutive interval belongs to $\mathbb{Q}$), or by gaps (if the shared bound belongs to $\mathbb{R} \setminus \mathbb{Q}$). For a given set of states P appearing in a limit transition, both the simple and Cantor shuffle rules additionally require a non-empty set of states $(g_1, \ldots, g_k)$ such that $g_i \to P \to g_i$, for all $i \in [1, k]$. Finally, the left and right infinite repetition rules are modified to allow for the required $\mathbb{Q}$-path to be cyclic, i.e., $q'_n = q_1$.

References

1. Bedon, N., Bès, A., Carton, O., Rispal, C.: Logic and rational languages of words indexed by linear orderings. Theory of Computing Systems **46**, 737–760 (2010)
2. Bès, A., Carton, O.: A Kleene theorem for languages of words indexed by linear orderings. International Journal of Foundations of Computer Science **17**(03), 519–541 (2006)
3. Bruyère, V., Carton, O.: Automata on linear orderings. Journal of Computer and System Sciences **73**(1), 1–24 (2007)

4. Büchi, J.R.: On a decision method in restricted second order arithmetic. In: Proc. Intl. Congress on Logic, Methodology and Philosophy of Science, 1960 (1962)
5. Büchi, J.R.: Transfinite automata recursions and weak second order theory of ordinals. In: Proc. Intl. Congress on Logic, Methodology, and Philosophy of Science, 1965 (1965)
6. Burgess, J.P., Gurevich, Y.: The decision problem for linear temporal logic. Notre Dame Journal of Formal Logic **26**(2), 115–128 (1985)
7. Cantor, G.: Über unendliche, lineare Punktmannichfaltigkeiten. Mathematische Annalen **20**(1), 113–121 (1882)
8. Cantor, G.: Beiträge zur Begründung der transfiniten Mengenlehre. Mathematische Annalen **46**, 481–512 (1895)
9. Carton, O.: Accessibility in automata on scattered linear orderings. In: Proc. Intl. Symp. on Mathematical Foundations of Computer Science. pp. 155–164. Springer (2002)
10. Choueka, Y.: Finite automata, definable sets, and regular expressions over ω^n-tapes. Journal of Computer and System Sciences **17**(1), 81–97 (1978)
11. Cristau, J.: Automata and temporal logic over arbitrary linear time. In: Proc. Intl. Conf. on Foundations of Software Technology and Theoretical Computer Science. LIPIcs, vol. 4, pp. 133–144 (2009)
12. DiBenedetto, E.: Real analysis. Springer (2002)
13. Läuchli, H., Leonard, J.: On the elementary theory of linear order. Fundamenta Mathematicae **59**(1), 109–116 (1966)
14. Muller, D.E.: Infinite sequences and finite machines. In: Proc. Annual Symp. on Switching Circuit Theory and Logical Design. pp. 3–16. IEEE Computer Society (1963)
15. Rabin, M.O.: Decidability of second-order theories and automata on infinite trees. Transactions of the American Mathematical Society **141**, 1–35 (1969)
16. Rabinovich, A.: Temporal logics over linear time domains are in PSPACE. Information and Computation **210**, 40–67 (2012)
17. Reynolds, M.: The complexity of temporal logic over the reals. Annals of Pure and Applied Logic **161**(8), 1063–1096 (2010)
18. Rosenstein, J.G.: Linear orderings. Academic press (1982)
19. Shelah, S.: The monadic theory of order. Annals of Mathematics **102**(3), 379–419 (1975)