On Expressive Power of Regular Expressions over Infinite Orders

Alexander Rabinovich

The Blavatnik School of Computer Science, Tel Aviv University, Israel
email: rabinoa@post.tau.ac.il

Abstract. Two fundamental results of classical automata theory are the Kleene theorem and the Büchi-Elgot-Trakhtenbrot theorem. Kleene's theorem states that a language of finite words is definable by a regular expression iff it is accepted by a finite state automaton. Büchi-Elgot-Trakhtenbrot's theorem states that a language of finite words is accepted by a finite-state automaton iff it is definable in the weak monadic second-order logic. Hence, the weak monadic logic and regular expressions are expressively equivalent over finite words. We generalize this to words over arbitrary linear orders.

1 Definitions and Result

A linear ordering $(L, <)$ is a non-empty set L equipped with a total order. A subset I of a linear order $(L, <)$ is convex, if for all $x < y < z$ with $x, z \in I$ also $y \in I$. We use "*interval*" as a synonym for "convex subset."

A linear order $(A, <)$ is *Dedekind complete* if every non-empty subset (of the domain) which has an upper bound has a least upper bound. For example, finite orders, the naturals and reals are Dedekind complete, while the order of the rationals is not.

In this paper a *cut* of a linearly ordered set $(A, <)$ is a downward closed set $C \subseteq A$. A cut C is non-trivial if it is not empty and is a proper subset of A. If $(A, <)$ is Dedekind complete and C is its nontrivial cut, then there is $a \in A$ such that $C := \{c \in A \mid c \leq a\}$ or $C := \{c \in A \mid c < a\}$.

1.1 Extended Regular Expression

We use a generalized notion of a word, which coincides with the notion of a labeled linear ordering. Given a finite alphabet Σ, a word over Σ or Σ-labeled chain is a linear order $(L, <)$ equipped with a function lab from L into Σ. A language over Σ is a class of words over Σ. Whenever Σ is clear from the context or unimportant we will use "word" for "word over Σ" and "language" for "language over Σ."

The concatenation (the lexicographical sum) of two words $w_1 = (L_1, <_1, lab_1)$ and $w_2 = (L_2, <_2, lab_2)$ over the same alphabet (up to renaming, assume that L_1 and L_2 are disjoint) is a word $(L_1 \cup L_2, <, lab)$, where (1) lab coincides with

lab_1 on L_1, with lab_2 on L_2, and (2) $<$ coincides with $<_1$ on L_1, with $<_2$ on L_2, and if $a \in L_1$ and $b \in L_2$ then $a < b$. The concatenation of words w_1 and w_2 is denoted[1] by $w_1 + w_2$.

For languages C_1 and C_2, their concatenation is defined as $\{w_1 + w_2 \mid w_1 \in C_1 \text{ and } w_2 \in C_2\}$ and is denoted by $C_1; C_2$.

The Kleene iteration or the positive concatenation closure of a language C is denoted by C^+ and is defined as $\cup_{k=1}^{\infty}\{w_1 + w_2 + \cdots + w_k \mid w_i \in C\}$.

Extended regular expressions over an alphabet Σ are defined by the following grammar: $E := \emptyset \mid \sigma \mid E \cup E \mid E; E \mid E^+ \mid \neg E$, where $\sigma \in \Sigma$. The semantics assigns to such an expression a language over Σ, as follows: (1) The empty language is assigned to $\emptyset$. (2) A language consisting of one element order labeled by σ is assigned to σ. (3) $\cup$ is interpreted as the union and $\neg$ as the complementation with respect to the class of all words over Σ. (4) $E_1; E_2$ is the concatenation of the languages assigned to E_1 and E_2, and (5) E^+ is the positive concatenation closure of the language assigned to E.

A *regular expression* is an extended regular expression without negation. Note that the semantics assigns to a regular expression only a set of finite words. Usually, in classical automata theory the complementation is taken only with respect to the set of finite words. Clearly, under such finite-words interpretation of complementation only languages of finite words are defined by extended regular expressions.

We conclude this section with examples which illustrate the expressive power of extended regular expressions.

All expressions below are over unary alphabet $\{1\}$; a word over a unary alphabet can be identified with the underlying linear order.

- $All := \neg\emptyset$ - defines the class of all linear orders.
- $Max := 1 \cup All; 1$ - defines the linear orders with a maximal element.
- $Min := 1 \cup 1; All$ - defines the linear orders with a minimal element.
- Dense:= $\neg(Max; Min)$ - defines the dense linear orders.
- Dedekind:=$1 \cup \neg((\neg Max); (\neg Min))$ - defines the Dedekind complete linear orders.
- Dense$^+$ - defines the orders which can be partitioned in a finite set of dense intervals; equivalently the linear order with a finite set of a successor elements where a is a successor if there is $b < a$ such that no element exists between b and a.

1.2 Fragments of MSO

The Monadic second-order logic (MSO) is an extension of first-order logic that allows to quantify over elements as well as over subsets of the domain of the structure.

The structures considered in this paper are expansions of nonempty linear orderings $(A, <^A)$ by subsets $P_1^A, \ldots, P_l^A$. When no confusion arises we cancel the

[1] In algebraic framework to formal languages the concatenation of w_1 and w_2 is called "the product" and is denoted by $w_1 \cdot w_2$.

superscript A, use the abbreviating notation $\overline{P}$ for the set tuple $(P_1^A, \ldots, P_l^A)$, and write $(A, \overline{P})$.

Such a structure is called l-chain. It can be regarded as a labeled ordering (or generalized word) with labels in $\{0,1\}^l$: the element $a \in A$ has the label $(b_1, \ldots, b_l)$ defined by $b_i := 1$ iff $a \in P_i$. When $P_1, \ldots, P_l$ partitions the domain of a linear ordering $(A, <^A)$, such a structure can be regarded as a word with labels in $\{1, \ldots, l\}$: the element $a \in A$ is labeled by i iff $a \in P_i$.

The standard language of *MSO* for structures of this signature is built up as follows, using the relation symbols $<$ and $P_1, \ldots P_m$. We have first-order variables $x, y, \ldots$ for elements of structures, monadic second order variables $X, Y \ldots$ for sets of elements of structures, and the atomic formulas are of the form $x = y$, $x < y$, $P_i(x)$, and $Y(x)$, with the canonical interpretation. Formulas are constructed from atomic formulas by the Boolean connectives, and by applying the first-order quantifier $\exists x$ "there is an element x" to first-order variables, and the monadic second-order quantifier $\exists X$ - "there is a set X" to monadic variables.

The Weak Monadic Second-Order logic is an extension of first-order logic that allows to quantify over elements as well as over finite subsets of the domain of the structure. So, it has the first-order quantifiers, and the quantifier $\exists^{fin} X$ - "there is a finite set X". We denote this logic by $MSO[\exists^{fin}]$.

The logic we are going to consider is denoted by $MSO[\exists^{fin}, \exists^{cut}]$ and it extends the weak monadic logic by the quantifier over cuts: $\exists^{cut} X$ - "there is a cut X."

A language (or a class of chains) *definable* by a formula φ is the class of all chains that satisfy φ.

Note that over Dedekind complete chains $MSO[\exists^{fin}]$ is expressively equivalent to $MSO[\exists^{fin}, \exists^{cut}]$. Both $MSO[\exists^{fin}]$ and $MSO[\exists^{fin}, \exists^{cut}]$ are equivalent to *MSO* over the class of finite words. McNaughton's theorem [10] implies that an ω-language is definable in *MSO* iff it is accepted by a deterministic Muller automaton. For a deterministic automaton "the run on an ω-word is accepting" can be formalized in $MSO[\exists^{fin}]$. Hence, $MSO[\exists^{fin}]$, $MSO[\exists^{fin}, \exists^{cut}]$ and *MSO* are expressively equivalent on the class of ω-words.

1.3 Result

Kleene [7] introduced regular expressions and proved that a language is definable by a regular expression iff it is accepted by a finite state automaton, and that the transformations from expressions to automata and vice versa are computable. The Büchi-Elgot-Trakhtenbrot theorem states that finite-state automata and the Weak Monadic Second-Order Logic (interpreted over finite words) have the same expressive power, and that the transformations from formulas to automata and vice versa are computable [1, 4, 17]. Hence, the classical theorem is:

Theorem 1.1 (Kleene, Büchi, Elgot, Trakhtenbrot) *The following are equivalent for languages of finite words:*

1. *A language is definable by a regular expression.*

2. *A language is accepted by a finite state automaton.*
3. *A language is definable in* $MSO[\exists^{fin}]$.

We generalize the equivalence between (1) and (3) of this classical result to arbitrary words, as follows:

Theorem 1.2 (Main) *A language of labelled orderings is definable by an extended regular expression iff it is definable in* $MSO[\exists^{fin}, \exists^{cut}]$.

Hence, extended regular expressions and $MSO[\exists^{fin}, \exists^{cut}]$ have the same expressive power over the class of all words. The transformations from formulas to extended regular expressions and vice versa are computable and can be easily extracted from the proof.

The paper is organized as follows. The next section provides a logical background and summarizes elements of the composition method. In Section 3 we prove that every $MSO[\exists^{fin}, \exists^{cut}]$ formula is equivalent to an extended regular expression. In Section 4 we prove that every extended regular expression is equivalent to a $MSO[\exists^{fin}, \exists^{cut}]$ formula. Section 5 presents a conclusion and further results.

2 Logical Background

2.1 A variant of $MSO[\exists^{fin}, \exists^{cut}]$

It will be convenient to work with a slightly modified (but expressively equivalent) set-up, in which the first-order variables are canceled. We allow only monadic second-order variables and take as atomic formulas of $MSO[\exists^{fin}, \exists^{cut}]$ the following: $Empty(X)$, $X \subseteq Y$, $Sing(X)$, $X < Y$, $All(X)$, $Finite(X)$ and $Cut(X)$. These are interpreted, respectively, as "X is empty," "X is a subset of Y," "X contains one element," "X contains one element and Y contains one element and the element of X is smaller than the element of Y," "X is the universe," "X is finite," and "X is a cut."

Formulas are constructed from atomic formulas by the Boolean connectives, and by the quantifiers $\exists^{fin}$ and $\exists^{cut}$.

The use of the unary relation symbols P_i will be avoided by taking free set variables X_i instead. Thus, we shall use labeled chains $(A, <, \overline{P})$ as interpretations of monadic formulas $\varphi(\overline{X})$.

The quantifier rank of a formula φ, denoted $\mathrm{qr}(\varphi)$, is the maximum depth of nesting of quantifiers in φ. For $r, l \in \mathbb{N}$ we denote by $\mathfrak{Form}_l^r$ the set of formulas of quantifier rank $\leq r$ and with free variables among $X_1, \ldots, X_l$.

2.2 Elements of the composition method

Our proofs use a technique known as the composition method [9, 14]. To fix notations and to aid the reader unfamiliar with this technique, we briefly review those definitions and results that we require. A more detailed presentation can be found in [16] or in [5].

2.2.1 Hintikka formulas and r-types

Definition 2.1 *Let $r, l \in \mathbb{N}$ and $\mathfrak{A}, \mathfrak{B}$ l-chains. The* r-theory *of $\mathfrak{A}$ is*

$$\mathrm{Th}^r(\mathfrak{A}) := \{\varphi \in \mathfrak{Form}_l^r \mid \mathcal{M} \models \varphi\}.$$

If $\mathrm{Th}^r(\mathfrak{A}) = \mathrm{Th}^r(\mathfrak{B})$, we say that $\mathfrak{A}$ and $\mathfrak{B}$ are r-equivalent *and write $\mathfrak{A} \equiv^r \mathfrak{B}$.*

Clearly, $\equiv^r$ is an equivalence relation. For any $r, l \in \mathbb{N}$, the set $\mathfrak{Form}_l^r$ is infinite. However, it contains only finitely many semantically distinct formulas. So, there are finitely many $\equiv^r$-classes of l-chains. In fact, we can compute "representatives" for these classes:

Lemma 2.2 (Hintikka Lemma) *For $r, l \in \mathbb{N}$, we can compute a* finite *set $H_l^r \subseteq \mathfrak{Form}_l^r$ such that:*

(*a*) *For* distinct *$\tau, \tau' \in H_l^r$, $\tau \wedge \tau'$ is* not *satisfiable.*
(*b*) *If $\tau \in H_l^r$ and $\varphi \in \mathfrak{Form}_l^r$, then either $\tau \models \varphi$ or $\tau \models \neg\varphi$. Furthermore, there is an algorithm that, given such τ and φ, decides which of these two possibilities holds.*
(*c*) *For every l-structure $\mathfrak{A}$, there is a* unique *$\tau \in H_l^r$ such that $\mathfrak{A} \models \tau$.*

Any member of H_l^r we call an (r, l)-Hintikka formula[2] *or a formal (r, l)-type.*

Definition 2.3 (r-type) *For $r, l \in \mathbb{N}$ and $\mathfrak{A}$ an l-chain, we denote by $\mathrm{Tp}^r(\mathfrak{A})$ the unique member of H_l^r satisfied by $\mathfrak{A}$ and call it the* r-type *of $\mathfrak{A}$.*

Thus, $\mathrm{Tp}^r(\mathfrak{A})$ determines $\mathrm{Th}^r(\mathfrak{A})$ and, indeed, $\mathrm{Th}^r(\mathfrak{A})$ is computable from $\mathrm{Tp}^r(\mathfrak{A})$.

Lemma 2.4 (Projection) *For $r, l \in \mathbb{N}$, there is an operation Pr_l^r from H_l^r into H_{l-1}^r such that if $\mathrm{Tp}_l^r(A, <, P_1^{\mathfrak{A}}, \cdots, P_{l-1}^{\mathfrak{A}}, P_l^{\mathfrak{A}}) = \tau$, then $\mathrm{Tp}_{l-1}^r(A, <, P_1^{\mathfrak{A}}, \cdots, P_{l-1}^{\mathfrak{A}}) = \mathrm{Pr}_l^r(\tau)$.*

2.2.2 The lexicographical sum of chains and of r-types

Let $\mathfrak{A} := (A, <^{\mathfrak{A}}, P_1^{\mathfrak{A}}, \ldots, P_l^{\mathfrak{A}})$ and $\mathfrak{B} := (B, <^{\mathfrak{B}}, , P_1^{\mathfrak{B}}, \ldots, P_l^{\mathfrak{B}})$ be l-chains with disjoint domains. The lexicographical sum (or concatenation) of $\mathfrak{A}$ and $\mathfrak{B}$ is denoted $\mathfrak{A} + \mathfrak{B}$ and is defined as the l-chain $(A \cup B, <, P_1^{\mathfrak{A}} \cup P_1^{\mathfrak{B}}, \ldots, P_l^{\mathfrak{A}} \cup P_l^{\mathfrak{B}})$ where $a < b$ if $a \in A$ and $b \in B$ or $a, b \in A$ and $a <_{\mathfrak{A}} b$ or $a, b \in B$ and $a <_{\mathfrak{B}} b$.

As usual, we do not distinguish between isomorphic structures. So, if the domains of $\mathfrak{A}$ and $\mathfrak{B}$ are not disjoint, replace them with isomorphic l-chains that have disjoint domains, and proceed as before.

It is clear that the sum of chains is associative. We will use the notation $\mathfrak{A}_1 + \mathfrak{A}_2 + \cdots + \mathfrak{A}_k$ for the sum of k chains.

The next Lemma says that $\equiv^r$ is a congruence with respect to the sum.

Lemma 2.5 *The r-types of l-chains $\mathfrak{A}$, $\mathfrak{B}$ determine the r-type of $\mathfrak{A} + \mathfrak{B}$.*

[2] Hintikka formulas made their first appearance in [6], in the framework of first-order logic.

The Lemma justifies the notation $\tau_1 + \tau_2$ for the r-type of an l-chain which is the sum of two l-chains of r-types τ_1 and τ_2, respectively. The composition theorem states that $+$ can be extended to a (uniformly) computable operation on the formal types.

Theorem 2.6 (Composition Theorem) *For $r, l \in \mathbb{N}$, there is an associative operation $+ : H^r_l \times H^r_l \to H^r_l$ such that for every l-chains $\mathfrak{A}$, $\mathfrak{B}$ if $\mathrm{Tp}^r(\mathfrak{A}) = \tau_1$ and $\mathrm{Tp}^r(\mathfrak{B}) = \tau_2$ then $\mathrm{Tp}^r(\mathfrak{A} + \mathfrak{B}) = \tau_1 + \tau_2$. Furthermore, the sum of (r, l)-formal types is (uniformly) computable.*

The reader may wonder why we do not say: "$\tau_1 + \tau_2$ is the *unique* element of H^r_l such that ...". The reason is that by Hintikka's construction [6] there are in H^r_l formulas that are not satisfied in any structure.

3 From Logic to Expressions

In this section we prove that for every formula φ in $MSO[\exists^{fin}, \exists^{cut}]$ there is an equivalent extended regular expression E_φ.

We proceed by induction on the quantifier rank of formulas.

For a quantifier free formula the corresponding equivalent expression is easily constructed.

If φ_1 is equivalent to E_{φ_i} for $i = 1, 2$, then $\varphi_1 \vee \varphi_2$ is equivalent to $E_{\varphi_1} \cup E_{\varphi_1}$, and $\neg\varphi_1$ is equivalent to $\neg E_{\varphi_1}$

The only interesting case is for quantifiers:

3.1 Translation for $\exists^{cut}$ quantifier

Assume that the inductive assumption holds for r. In particular, for every Hintikka formula τ of quantifier rank r there is an equivalent expression E_τ.

Let $\varphi(X_1, \cdots, X_l)$ be a formula and assume that $\mathrm{qr}(\varphi) = r$.

$\exists^{cut} X_l \varphi$ is equivalent to a disjunction of

1. $\varphi_0 := \exists^{cut} X_l \mathit{Empty}(X_l) \wedge \varphi$
2. $\varphi_1 := \exists^{cut} X_l \mathit{All}(X_l) \wedge \varphi$
3. $\varphi_2 := \exists^{cut} X_l \neg \mathit{Empty}(X_l) \wedge \neg \mathit{All}(X_l) \wedge \varphi$

Let $S_0 \subseteq H^r_{l-1}$ be defined as $\{\mathrm{Pr}^r_l(\tau_1) \mid \tau_1 \in H^r_l \text{ and } \tau_1 \models \varphi \wedge \mathit{Empty}(X_l)\}$, where Pr^r_l was defined in Lemma 2.4. Then $\mathfrak{A} \models \varphi_0$ iff $\mathrm{Tp}^r_{l-1}(\mathfrak{A}) \in S_0$. Therefore, φ_0 is equivalent to $\cup_{\tau \in S_0} E_\tau$ (where E_τ are defined by the inductive assumption). For φ_1 an equivalent expression E_{φ_1} is defined in a similar way as $E_{\varphi_1} := \cup_{\tau \in S_1} E_\tau$, where $S_1 := \{\mathrm{Pr}^r_l(\tau_1) \mid \tau_1 \in H^r_l \text{ and } \tau_1 \models \varphi \wedge \mathit{All}(X_l)\}$.

In order to translate φ_2 into an equivalent expression we will use the composition theorem and an observation that every non-empty proper downward closed subset P of the domain of $\mathfrak{A}$ induces a representation of $\mathfrak{A}$ as the sum $\mathfrak{A}_1 + \mathfrak{A}_2$ where $\mathfrak{A}_1$ (respectively, $\mathfrak{A}_2$) is the substructure of $\mathfrak{A}$ over P (respectively, the complement of P).

Set $\psi_2 := \neg \mathit{Empty}(X_l) \wedge \neg \mathit{All}(X_l) \wedge \mathit{Cut}(X_l) \wedge \varphi$. Hence, $\varphi_2 := \exists^{cut} X_l \psi_2$.

Claim 1 *Let $\mathfrak{B}$ be an l-chain. $\mathfrak{B} \models \psi_2$ iff there are $\tau_1, \tau_2 \in H^r_l$ and $\mathfrak{B}_1$ and $\mathfrak{B}_2$ such that*

1. *$\mathfrak{B} = \mathfrak{B}_1 + \mathfrak{B}_2$ and $\tau_i = \mathrm{Tp}^r(\mathfrak{B}_1)$ for $i = 1, 2$.*
2. *$\tau_1 + \tau_2 \models \psi_2$.*
3. *$\tau_1 \models All(X_l)$ and $\tau_2 \models Empty(X_l)$.*

Proof. $\Leftarrow$ is immediate.

$\Rightarrow$ Take as $\mathfrak{B}_1$ (respectively, $\mathfrak{B}_2$) the substructure of $\mathfrak{B}$ over P_l (respectively, over the complement of P_l), and as τ_i the r-type of $\mathfrak{B}_i$. □

Let S be the set of pairs $\langle \tau_1, \tau_2 \rangle$ of H^r_l formulas, which satisfy conditions (2) and (3) of Claim 1.

Define $\widehat{S} \subseteq H^r_{l-1} \times H^r_{l-1}$ as $\widehat{S} := \{\langle \mathrm{Pr}^r_l(\tau_1), \mathrm{Pr}^r_l(\tau_2)\rangle \mid \langle \tau_1, \tau_2 \rangle \in S\}$, where Pr^r_l was defined in Lemma 2.4. Thus we obtain:

Claim 2 *$\mathfrak{A} \models \varphi_2$ if and only if there are $\mathfrak{A}_1$ and $\mathfrak{A}_2$ such that $\mathfrak{A} = \mathfrak{A}_1 + \mathfrak{A}_2$ and $\langle \mathrm{Tp}^r(\mathfrak{A}_1), \mathrm{Tp}^r(\mathfrak{A}_2)\rangle \in \widehat{S}$.*

By the inductive assumption each formula of quantifier rank r is equivalent to an expression. In particular, each Hintikka formula τ of quantifier rank r is equivalent to an expression E_τ. Finally, Claim 2 implies that φ_2 is equivalent to $\cup_{\langle \tau_1, \tau_2 \rangle \in \widehat{S}} E_{\tau_1}; E_{\tau_2}$.

3.2 Translation for $\exists^{fin}$ quantifier

In order to translate $\exists^{fin} X_l \varphi$ into an equivalent expression we will use the composition theorem and an observation that every finite subset of the domain of $\mathfrak{A}$ induces a natural representation of $\mathfrak{A}$ as a finite sum of its subchains.

Claim 3 *$\mathfrak{B} \models \varphi \wedge Finite(X_l)$ iff there is a sequence $\tau_1, \ldots, \tau_k$ of H^r_l formulas and a sequence $\mathfrak{B}_1 \ldots, \mathfrak{B}_k$ of l-chains such that*

1. *$\mathfrak{B} = \mathfrak{B}_1 + \mathfrak{B}_2 + \cdots + \mathfrak{B}_k$ and $\tau_i = \mathrm{Tp}^r(\mathfrak{B}_i)$ for $i = 1, \ldots, k$.*
2. *$\tau_1 + \tau_2 + \cdots + \tau_k \models \varphi$ and*
3. *if $\tau_i \models \neg Empty(X_l)$ then $\tau_i \models Sing(X_l) \wedge All(X_l)$, i.e., τ_i holds only on singleton chains.*

Proof. $\Leftarrow$ is immediate.

$\Rightarrow$ Assume $\mathfrak{B} \models \varphi \wedge Finite(X_l)$. Hence, P_l is finite. Define an equivalence $\sim$ as follows: $a_1 \sim a_2$ iff either $a_1 = a_2 \in P_l$ or there is no element of P_l in the interval $[\min(a_1, a_2), \max(a_1, a_2)]$. It is clear that $\sim$ is an equivalence relation. It has finitely many equivalence classes, and each $\sim$ equivalence class is an interval of the domain of $\mathfrak{B}$. Let $I_1 < \cdots < I_k$ be the $\sim$-equivalence classes. For $j = 1, \ldots, k$, define $\mathfrak{B}_j$ as the substructure of $\mathfrak{B}$ over I_j and $\tau_j := \mathrm{Tp}^r(\mathfrak{B}_j)$. It is clear that $\mathfrak{B}_j$ and τ_j satisfy the requirements of the claim. □

Let S be the set of finite sequences of H_l^r formulas, which satisfy conditions (2) and (3) of Claim 3.

Define a set $\widehat{S}$ of finite sequences of H_{l-1}^r formulas as $\widehat{S} := \{\langle \mathrm{Pr}_l^r(\tau_1), \ldots, Pr_l^r(\tau_k)\rangle \mid \langle \tau_1, \ldots, \tau_k\rangle \in S\}$. Therefore, Claim 3 implies:

Claim 4 *$\mathfrak{A} \models \exists^{fin} X_l \varphi$ iff there is a sequence $\langle \tau_1, \ldots, \tau_k\rangle \in \widehat{S}$ and a sequence $\mathfrak{A}_1 \ldots, \mathfrak{A}_k$ of $(l-1)$-chains such that $\mathfrak{A} = \mathfrak{A}_1 + \mathfrak{A}_2 + \cdots + \mathfrak{A}_k$ and $\tau_i = \mathrm{Tp}^r(\mathfrak{A}_i)$ for $i = 1, \ldots, k$.*

Claim 5 *There is a regular expression E which defines $\widehat{S}$.*

Proof. We will construct a finite state automaton $\mathcal{A}$ which accepts $\widehat{S}$. The set $Q_{\mathcal{A}}$ of its states is $Q_{\mathcal{A}} := \{q_i\} \cup H_l^r$, where $q_i \notin H_l^r$ is a fresh state.

q_i is the initial state of $\mathcal{A}$. The set Acc of accepting states is defined as $Acc := \{\tau \in H_l^r \mid \tau \models \varphi \wedge \mathit{Finite}(X_l)\}$.

For every $\tau \in H_{l-1}^r$ define two sets $D(\tau), F(\tau) \subseteq H_l^r$ as $D(\tau) := \{\tau' \in H_l^r \mid \tau' \models \tau \wedge \mathit{Empty}(X_l)\}$ and $F(\tau) := \{\tau' \in H_l^r \mid \tau' \models \tau \wedge \mathit{All}(X_l) \wedge \mathit{Sing}(X_l)\}$.

The transition relation $\rightarrow_{\mathcal{A}} \subseteq Q_{\mathcal{A}} \times H_{l-1}^r \times Q_{\mathcal{A}}$ is defined as follows:

1. $\langle q_i, \tau, \tau'\rangle \in \rightarrow_{\mathcal{A}}$ iff $\tau' \in D(\tau) \cup F(\tau)$.
2. $\langle \tau_1, \tau, \tau_2\rangle \in \rightarrow_{\mathcal{A}}$ iff there is $\tau' \in D(\tau) \cup F(\tau)$ such that $\tau_2 = \tau_1 + \tau'$.

It is straightforward to check that $\mathcal{A}$ accepts $\widehat{S}$. Therefore, by Theorem 1.1, $\widehat{S}$ is definable by a regular expression. □

By the inductive assumption each formula of quantifier rank r is equivalent to an expression. In particular, each Hintikka formula τ of quantifier rank r is equivalent to an expression E_τ.

Finally, let E_φ be obtained from a regular (complementation free) expression E of Claim 5, by replacing each letter $\tau \in H_{l-1}^r$ with an equivalent extended regular expression E_τ. Claims 4 and 5 imply that φ is equivalent to E_φ.

4 From Expressions to Logic

We are going to prove that for every expression E over an alphabet Σ there is an equivalent $MSO[\exists^{fin}, \exists^{cut}]$ formula φ.

We proceed by the structural induction on expressions.

It is straightforward to write a formula for $\emptyset$ and for a letter $\sigma \in \Sigma$.

If E_i are equivalent to φ_i for $i = 1, 2$, then $E_1 \cup E_2$ is equivalent to $\varphi_1 \vee \varphi_2$ and $\neg E_1$ is equivalent to $\neg\varphi_1$.

Below we will treat concatenation and iteration.

First, let us introduce notations and state a standard "relativization" lemma which will be used several times.

Notation 4.1 *Let $l \in \mathbb{N}$, $\mathfrak{A} := (A, <, P_1, \ldots, P_l)$ an l-chain and D a non-empty subset of A. The* restriction *of $\mathfrak{A}$ to D is the l-chain $\mathfrak{A}_{\restriction D}$ defined as $\mathfrak{A}_{\restriction D} := (D, <, P_1 \cap D, \ldots, P_l \cap D)$.*

Lemma 4.2 (Relativization) *Let $\varphi(\overline{Y})$ be a formula, U a variable not appearing in φ. There is a formula $\varphi_{\restriction U}(\overline{Y}, U)$ such that for every chain $(A, <, \overline{P})$ and every non-empty $D \subseteq A$,*

$$(A, <, \overline{P}, D) \models \varphi_{\restriction U}(\overline{Y}, U) \text{ iff } (A, <, \overline{P})_{\restriction D} \models \varphi(\overline{Y}).$$

When this is the case, we say that φ holds in $(A, <, \overline{P})$ *relativized* to D.

4.1 Concatenation

Assume φ_i is equivalent to E_i for $i = 1, 2$.

Then $E_1; E_2$ is equivalent to $\exists^{cut} X \varphi$ where φ is the conjunction of the following:

1. X is a non-empty proper downward closed subset of the domain of $\mathfrak{A}$.
2. φ_1 holds in $\mathfrak{A}$ relativized to X.
3. φ_2 holds in $\mathfrak{A}$ relativized to the complement of X.

(1)-(3) are easily formalized in $MSO[\exists^{fin}, \exists^{cut}]$. Moreover, if φ_1 and φ_2 are $MSO[\exists^{cut}]$ formulas, then (1)-(3) are easily formalized in $MSO[\exists^{cut}]$.

4.2 Kleene Iteration

Assume that E is equivalent to φ.

Recall that $\mathfrak{A}$ is in E^+ iff there is $k > 0$ and a partition of the domain of $\mathfrak{A}$ into intervals $I_1, \ldots, I_k$ such that $\mathfrak{A}_{\restriction I_j}$ are in E. In the case when all I_j are intervals with endpoints in $\mathfrak{A}$ this can be easily formalized. However, $\mathfrak{A}$ is not necessarily Dedekind complete, and not all intervals have end-points in $\mathfrak{A}$. To overcome this problem we use the following Lemma:

Lemma 4.3 *Let $\varphi(\overline{X})$ be a formula. Then there are formulas $\psi^i_{\leq}(\overline{X})$ and $\psi^i_{\geq}(\overline{X})$ $(i = 0, \ldots, m)$ such that for every $\mathfrak{A}$, element $a \in \mathfrak{A}$, and intervals $I_{\leq a} := \{b \in \mathfrak{A} \mid b \leq a\}$ and $I_{\geq a} := \{b \in \mathfrak{A} \mid b \geq a\}$:*

$$\mathfrak{A} \models \varphi \text{ iff there is } i \text{ such that } \mathfrak{A}_{\restriction I_{\leq a}} \models \psi^i_{\leq} \text{ and } \mathfrak{A}_{\restriction I_{\geq a}} \models \psi^i_{\geq}$$

The Lemma is easily obtained from Lemma 2.5 (one can take as $\psi^i_{\leq}(\overline{X})$, $\psi^i_{\geq}(\overline{X})$ formulas of quantifier rank smaller than $\mathrm{qr}(\varphi) + 3$).

The Lemma implies that "$\mathfrak{A}$ is in E^+" can be rephrased as:

> there is a partition of the domain of $\mathfrak{A}$ into intervals $I_1, \ldots, I_k$ and there are $a_j \in I_j$ and a function $F : \{1, \ldots, k\}$ into $\{0, \ldots, m\}$ such that for every $j \in \{1, \ldots, k\}$ and $s := F(j)$
> 1. the substructure of $\mathfrak{A}$ over the interval $\{b \in I_j \mid b \geq a_j\}$ satisfies $\psi^s_{\geq}$
> 2. the substructure of $\mathfrak{A}$ over the interval $\{b \in I_j \mid b \leq a_j\}$ satisfies $\psi^s_{\leq}$

The above is equivalent to

there is a non-empty finite subset P of the domain of $\mathfrak{A}$ and a function $F : P \to \{0, \ldots, m\}$ such that

1. if a is the maximal element of P and $s = F(a)$ then the substructure of $\mathfrak{A}$ over the interval $\{b \mid b \geq a\}$ satisfies $\psi^s_{\geq}$
2. if a is the minimal element of P and $s = F(a)$ then the substructure of $\mathfrak{A}$ over the interval $\{b \mid b \leq a\}$ satisfies $\psi^s_{\leq}$, and
3. If $a < c$ are successive elements of P and $s = F(a)$ and $p = F(c)$, then there is a downward closed set D such that
   (a) $a \in D$, $c \notin D$
   (b) the substructure of $\mathfrak{A}$ over the interval $\{b \in D \mid b \geq a\}$ satisfies $\psi^s_{\geq}$
   (c) the substructure of $\mathfrak{A}$ over the interval $\{b \notin D \mid b \leq c\}$ satisfies $\psi^p_{\leq}$

Observe that F cannot be represented by a single monadic predicate. However, since F is a mapping from a finite set P to a set of size $m + 1$ (m is defined in Lemma 4.3 depens on φ, but is independent of P), it can be represented by a tuple of finite sets and the conditions (1)-(3) can be easily formalized in $MSO[\exists^{fin}, \exists^{cut}]$.

5 Conclusion

The classical automata theory establishes equivalence (over finite words) between three fundamental formalisms: the monadic second-order logic, regular expressions and finite state automata. The cornerstones of automata theory on infinite objects are Büchi's and Rabin's theorems. The Büchi theorem states that *MSO* and finite automata are equivalent over ω-words [2] and the Rabin theorem states that *MSO* and finite automata are equivalent over labeled binary trees [12].

MSO and its fragment have a natural interpretation over arbitrary (even partial) orders. Regular expressions have a natural interpretation over arbitrary linear orders. We proved expressive equivalence (over arbitrary words) between the extended regular expressions and $MSO[\exists^{fin}, \exists^{cut}]$. It seems that there is no natural notion of automata which has the same expressive power as the above formalisms. Usually, automata correspond to logical formulas of a fixed quantifier alternation depth. However, Thomas Colcombet pointed out that the quantifier alternation hierarchy does not collapse for $MSO[\exists^{fin}, \exists^{cut}]$.

Below we comment about some extensions of our results.

5.1 Words over linear orders of a bounded cardinality

Let $\aleph$ be an infinite cardinal. A linear order $(L, <)$ is an $\aleph^<$-order if the cardinality of L is less than $\aleph$. Given a finite alphabet Σ, an $\aleph^<$-word over Σ or Σ-labeled $\aleph^<$-chain is an $\aleph^<$-linear order $(L, <)$ equipped with a function *lab* from L into Σ. A $\aleph^<$-language over Σ is a set of $\aleph^<$-words over Σ. Whenever Σ

is clear from the context or unimportant we will use "$\aleph^<$-word" for "$\aleph^<$-word over Σ" and "$\aleph^<$-language" for "$\aleph^<$-language over Σ."

For an extended regular expression E over Σ the $\aleph^<$-semantics assigns an $\aleph^<$-language over Σ. The $\aleph^<$-semantics is defined exactly like the semantics of extended regular expressions in Sect. 1.1 with the only exception that complementation is taken with respect to the set of $\aleph^<$-word over Σ. Namely, the $\aleph^<$-semantics is defined as follows: (1) The empty language is assigned to $\emptyset$. (2) A language consisting of one element order labeled by σ is assigned to σ. (3) $\cup$ is interpreted as the union and $\neg$ as the complementation with respect to the set of all $\aleph^<$-words over Σ. (4) $E_1; E_2$ is the concatenation of the languages assigned to E_1 and E_2. (5) E^+ is the positive concatenation closure of the language assigned to E.

Note

(1) For the first infinite cardinal $\aleph_0$, the $\aleph_0^<$-semantics assigns to an extended regular expression the same language (of finite words) as the classical semantics does.
(2) If C is the class of words assigned to E by the semantics defined in Sect. 1.1, then $\aleph^<$-semantics assigns to E the set of all $\aleph^<$-words in C.

We say that a language C is $\aleph^<$-*definable* by an expression E if $\aleph^<$-semantics assigns C to E. We say that an $\aleph^<$-language is *definable* by an MSO formula φ iff it is the set of all $\aleph^<$-words that satisfy φ.

Our main theorem and (2) imply the following Theorem:

Theorem 5.1 *Let $\aleph$ be an infinite cardinal. An $\aleph^<$-language is definable by an extended regular expression iff it is definable by an MSO$[\exists^{fin}, \exists^{cut}]$ formula.*

From our proof it is also easy to extract that a language of labelled Dedekind complete orderings is definable by an extended regular expression iff it is definable by an $MSO[\exists^{fin}]$ formula.

5.2 Star-Free Expressions

McNaughton and Papert introduced star-free regular expressions. These are extended regular expressions without the Kleene iteration. Namely, given an alphabet Σ, the star-free expressions over Σ are built up from $\emptyset$ and the letters in Σ by union, concatenation and complementation. A famous theorem of McNaughton and Papert [11] states that a language of finite words is definable by a star-free expression if and only if it is definable in first-order logic. This theorem was extended to ω-languages in Ladner [8] and Thomas [15], and to languages over the real order by Rabinovich [13]. The following generalization to Dedekind complete orders was proved in [13]:

Theorem 5.2 *A language of labelled Dedekind complete orderings is definable by a star-free regular expression iff it is definable by a first-order formula.*

Our proof of Theorem 1.2 can be easily modified to show that:

Theorem 5.3 *A language of labelled orderings is definable by a star-free regular expression iff it is definable by an MSO$[\exists^{cut}]$ formula.*

References

1. J. R. Büchi. Weak second-order arithmetic and finite automata. Zeit. Math. Logik und Grundl. Math. 6, pp. 66-92, 1960.
2. J. R. Büchi. On a decision method in restricted second order arithmetic. In: Logic, Methodology and Philosophy of Science, Stanford University Press, pp. 1-11, 1962.
3. T. Colcombet. Personal communication. September 2012.
4. C. Elgot. Decision problems of finite-automata design an d related arithmetics. Trans. Amer. Math. Soc. 98, pp. 21-51, 1961.
5. Y. Gurevich. Monadic second order theories. In J. Barwise and S. Feferman eds. Model Theoretic Logics pp. 479-506, Springer Verlag, 1986.
6. J. Hintikka, Distributive normal forms in the calculus of predicates, Acta Philos. Fennica 6, 1953.
7. S. Kleene, Representation of events in nerve nets and finite automata, Automata Studies, Princeton University Press, pp. 3-41, 1956.
8. R. E. Ladner Application of model theoretical games to linear orders and finite automata theory. Information and Control 9, pp. 521-530, 1977.
9. H. Läuchli and J. Leonard, On the elementary theory of linear order, Fund. Math. 59 pp. 109-116, 1966.
10. R. McNaughton, Testing and generating infinite sequences by a finite automaton, Information and Control 9, pp. 521-530, 1966.
11. R. McNaughton and S. Papert. Counter-free automata. The MIT Press, 1971.
12. M. O. Rabin. Decidability of second-order theories and automata on infinite trees. Transactions of the American Mathematical Society, vol. 141, pp. 1-35, 1969.
13. A. Rabinovich Star Free Expressions over the Reals. Theoretical Computer Science, Vol. 233, pp. 233-245, 2000.
14. S. Shelah, The monadic theory of order, Annals of Mathematics, Ser. 2, Vol. 102, pp. 379-419, 1975.
15. W. Thomas. Star Free regular sets of ω-sequences. Information and Control 42, pp. 148-156, 1979.
16. W. Thomas, Ehrenfeucht games, the composition method, and the monadic theory of ordinal words. In *Structures in Logic and Computer Science, A Selection of Essays in Honor of A. Ehrenfeucht*. LNCS 1261, Springer, pp. 118-143, 1997.
17. B. A. Trakhtenbrot. The synthesis of logical nets whose operators are described in terms of one-place predicate calculus. Doklady Akad. Nauk SSSR 118 (4), pp. 646-649, 1958.