

## Contents

1 Terminology: quantifier rank, type	1
2 Büchi's theorem on strings	3

## 1 Terminology: quantifier rank, type

We introduce some important terminology, which will be also used for proving Theorem 7.

**Definition 1** (Quantifier rank). *A quantifier rank of a formula  $\psi$  is the maximum depth of its nested quantifiers. That is,*

- *an atomic formula has quantifier rank 0,*
- *the quantifier rank of a boolean combination  $(\wedge, \vee, \neg)$  of formulas is the maximum quantifier rank over the formulas,*
- *one existential / universal quantification increases the quantifier rank by exactly 1.*

*The set of all MSO-formulas of quantifier rank at most  $k$  is denoted by  $\text{MSO}[k]$*

One can consider all MSO-sentences over  $\tau$  satisfied by some  $\tau$ -structure. There are infinitely many MSO-sentences and such a set consisting of all true MSO-sentences in some  $\tau$ -structure could be infinitely large. However, when you restrict to sentences of quantifier rank up to  $k$ , the set becomes finite, size bounded by a function of  $k$ .

**Definition 2** (Rank- $k$   $(\ell, m)$ -type). *For a relational structure  $\mathbb{A}$  over  $\tau$ ,  $\ell$ -tuple  $\vec{v} = (v_1, \dots, v_\ell) \in A^\ell$  of elements of  $A$  and  $m$ -tuple  $\vec{V} = (V_1, \dots, V_m) \in (2^A)^m$  of subsets of  $A$ , we define the MSO rank- $k$   $\ell, m$ -type of  $(\mathbb{A}, \vec{v}, \vec{V})$  as the set of all MSO-formulas with  $\ell$  free individual variables and  $m$  free set variables satisfied by  $(\mathbb{A}, \vec{v}, \vec{V})$ . That is,*

$$\text{mso-type}_k(\mathbb{A}, \vec{v}, \vec{V}) = \{\psi \in \text{MSO}[k] \mid \mathbb{A} \models \psi(\vec{v}, \vec{V})\}.$$

*When  $\ell = 0, m = 0$ , the MSO rank- $k$   $\ell, m$ -type of a structure  $\mathbb{A}$  is simply called the rank- $k$  type of  $\mathbb{A}$ . Notice that  $\text{mso-type}_k(\mathbb{A})$  is the set of all MSO-sentences of quantifier rank at most  $k$  which holds on  $\mathbb{A}$ .*

*An MSO rank- $k$   $\ell, m$ -type (when a  $\tau$ -structure  $\mathbb{A}$ ,  $\ell$ -tuple of elements and  $m$ -tuple of subsets of  $A$  is not specified) is the set  $S$  of MSO-formulas in  $\text{MSO}[k]$  with  $\ell$  free individual variables and  $m$  free set variables such that*

- **CONSISTENCY:** *there exist a  $\tau$ -structure  $\mathbb{A}$  and an  $\ell$ -tuple  $\vec{v}$  and an  $m$ -tuple of sets  $\vec{V}$  over  $A$  which satisfies all the formulas in  $S$ , and*

- **COMPLETENESS:** for any MSO-formula  $\psi$  with  $\ell$  free individual variables and  $m$  free set variables, exactly one of  $\psi$  or  $\neg\psi$  is included in the set

It is not difficult to see that for each fixed  $k, \ell, m$ , there are finitely many (bounded by a function of  $k$ ) MSO-formulas of quantifier rank up to  $k$  with  $\ell$  free individual variables and  $m$  free set variables. This is particularly because in the base case, i.e. when a formula is quantifier-free, the formula is a boolean combination of atomic formulas on  $\ell + m + k$  free variables. The number of atomic formulas is bounded by a function of  $\ell, m$  and  $\tau$ , and the number of their boolean combinations (up to logical equivalence!) are again bounded. We state this observation without proof.

**Observation 3.** For any fixed  $k, \ell, m$ , the number of MSO rank- $k$   $\ell, m$ -types is finite, determined solely by  $k, \ell, m$  and the vocabulary  $\tau$ .

Note that for  $\tau$ -structure  $\mathbb{A}$ ,  $\text{mso-type}_k(\mathbb{A})$  is an MSO rank- $k$  type; the consistency of the set  $\text{mso-type}_k(\mathbb{A})$  is witnessed by the very structure  $\mathbb{A}$ . The completeness of  $\text{mso-type}_k(\mathbb{A})$  is clear from the fact that  $\text{qr}(\psi) = \text{qr}(\neg\psi)$  for any formula  $\psi$  and exactly one of  $\mathbb{A} \models \psi$  and  $\mathbb{A} \models \neg\psi$  holds.

The consistency and completeness of MSO rank- $k$  type implies that the converse also holds. That is, for any MSO rank- $k$  type  $q$  there exists a  $\tau$ -structure  $\mathbb{A}$  such that  $q = \text{mso-type}_k(\mathbb{A})$ .

**Lemma 4.** Let  $\mathcal{Q}$  be the set of all MSO rank- $k$  types in  $\text{MSO}[k]$ . Then for every  $Q \in \mathcal{Q}$ , there exists a  $\tau$ -structures  $\mathbb{A}$  such that  $\text{mso-type}_k(\mathbb{A}) = Q$ .

**Proof:** Choose an arbitrary  $Q \in \mathcal{Q}$ . By consistency of MSO rank- $k$  type, there exists a  $\tau$ -structure  $\mathbb{A}$  which satisfies all sentences in  $Q$ . This implies that  $Q \subseteq \text{mso-type}_k(\mathbb{A})$ . We want to show  $Q = \text{mso-type}_k(\mathbb{A})$ . Suppose  $\psi \in \text{mso-type}_k(\mathbb{A}) \setminus Q$ . Because  $Q$ , as an MSO rank- $k$  type, is complete and  $Q$  contains  $\neg\psi$ . Then  $\text{mso-type}_k(\mathbb{A})$  contains both  $\neg\psi$  and  $\psi$ , which is impossible.  $\square$

**Definition 5** (Disjoint union on  $\tau$ -structures). When the vocabulary  $\tau$  contains only the predicates and no constant symbols<sup>1</sup>, the disjoint union  $\mathbb{A} \cup \mathbb{B}$  of  $\tau$ -structures  $\mathbb{A}$  and  $\mathbb{B}$  with disjoint universe is defined as:

- the universe of  $\mathbb{A} \cup \mathbb{B}$  is  $A \cup B$
- the interpretation  $R^{\mathbb{A} \cup \mathbb{B}}$  of  $R$  is  $R^{\mathbb{A}} \cup R^{\mathbb{B}}$  for each predicate  $R \in \tau$ .

The so-called *compositionality* of MSO logic is of central importance. It is rather loosely defined and needs to be appropriately formulated in relevant settings. Informally speaking, it says that whether a given MSO-sentence (or formula) holds on a relational structure is determined by whether MSO-sentences hold on relational substructures, when the original structure is formed from the substructures with well-regulated combination rule. The next lemma observes the simplest case of MSO compositionality. We postpone the proof till we learn *Ehrenfeucht-Fraïssé game*.

**Lemma 6.** Let  $\mathbb{A}, \mathbb{A}', \mathbb{B}, \mathbb{B}'$  be  $\tau$ -structures such that  $\text{mso-type}_k(\mathbb{A}) = \text{mso-type}_k(\mathbb{A}')$  and  $\text{mso-type}_k(\mathbb{B}) = \text{mso-type}_k(\mathbb{B}')$ . Then it holds that  $\text{mso-type}_k(\mathbb{A} \cup \mathbb{B}) = \text{mso-type}_k(\mathbb{A}' \cup \mathbb{B}')$ .

---

<sup>1</sup>Why do we need this restriction?

## 2 Büchi's theorem on strings

We explore the surprising connection between MSO logic on strings and regular languages.

**Theorem 7** (Büchi'60, Elgot'61, Trakhtenbrot'62). *[1, 2] A language is regular if and only if it is definable in MSO.*

Theorem 7 crucially relies on the compositionality of MSO logic on strings under concatenation<sup>2</sup>. We defer the proof of Lemma 8 for now.

**Lemma 8** (MSO is compositional under concatenation). *Let  $s_i, s'_i$  for  $i = 1, 2$  be two strings over the alphabet  $\Sigma$ . If*

$$\text{mso-type}_k(s_1) = \text{mso-type}_k(s_2) \quad \text{and} \quad \text{mso-type}_k(s'_1) = \text{mso-type}_k(s'_2),$$

*then it holds that  $\text{mso-type}_k(s_1 \cdot s'_1) = \text{mso-type}_k(s_2 \cdot s'_2)$*

### Proof of Theorem 7:

◇ Forward implication. We use the fact that any regular language has a regular expression. To establish that a regular language is MSO-definable, it suffices to prove that a regular language  $L$  are MSO-definable for each of the following cases by induction on the length of the regular expression  $R$  generating  $L$ :

- $R = a$  for some letter  $a \in \Sigma$ :

$$\varphi_a := \exists x(P_a(x) \wedge \forall z(x = z)).$$

- $R = \epsilon$  :

$$\varphi_\epsilon := \neg \exists x(x = x).$$

- $R = \emptyset$  :

$$\varphi_\emptyset := \exists x(x \neq x)$$

- $R = R_1 \cup R_2$ : by induction hypothesis, there exists MSO-sentences  $\varphi_1$  and  $\varphi_2$  such that  $s \models \varphi_i$  if and only if  $s \in L(R_i)$  for  $i = 1, 2$ . Now  $\varphi := \varphi_1 \vee \varphi_2$  is the desired MSO-sentence.
- $R = \bar{R}'$ : by induction hypothesis, there exists MSO-sentences  $\varphi'$  such that  $s \models \varphi'$  if and only if  $s \in L(R')$ . Now  $\varphi := \neg \varphi'$  is the desired MSO-sentence.

- $R = R_1 \cdot R_2$ : Again by induction hypothesis, there exists MSO-sentences  $\varphi_1$  and  $\varphi_2$  such that  $s \models \varphi_i$  if and only if  $s \in L(R_i)$  for  $i = 1, 2$ .

If we simply take the conjunction  $\varphi_1 \wedge \varphi_2$  to define  $L(R)$ , then for the evaluation  $\varphi_1 \wedge \varphi_2$  on a given string  $s$  we consider an interpretation of a variable of  $\varphi_1$  in  $s$ . But what we actually want is to evaluate  $\varphi_1$  on the substring  $s[1 : z]$ , i.e. *up to some position  $z$* . Likewise, we want to evaluate  $\varphi_2$  on the substring  $s[z + 1, n]$ . For this, we need to modify the original sentence  $\varphi_1$  defining  $L(R_1)$  so that, even when the variables are interpreted in  $s$ , in practice its interpretation is confined to the prefix (likewise  $\varphi_2$  for the suffix of  $s$ ). We can achieve this effect by replacing every occurrence

- $\exists x \psi$  by  $\exists x(x \leq z) \wedge \psi(x)$ , and
- $\exists X \psi$  by  $\exists X(\forall x(x \in X \rightarrow x \leq z)) \wedge \psi(X)$

---

<sup>2</sup>The compositionality of MSO logic on strings under concatenation is a special case of Feferman-Vaught Theorem.

in  $\varphi_1$  using  $z$  as a free individual variable. A similar relativization can be applied to the universal quantifiers in  $\varphi_1$ .

- $\forall x \psi(x)$  by  $\forall x(x \leq z \rightarrow \psi)$ , and
- $\forall X \psi(X)$  by  $\forall X(\forall x(x \in X \rightarrow x \leq z)) \rightarrow \psi(X)$  (read as: “if  $X \leq z$  then  $\psi$  holds”).

A symmetric modification applies to  $\varphi_2$ . Apparently, the free variable  $z$  points at the last position of the prefix so that it matches a string from  $L(R_1)$ . Let  $\varphi_1^{pf}(z)$  and  $\varphi_2^{sf}(z)$  be the respective formulas obtained as above. It is not difficult to see that

$$\varphi := \exists z \varphi_1^{pf}(z) \wedge \varphi_2^{sf}(z)$$

is an MSO-sentence defining  $L(R)$ .

•  $R = (R')^*$ : by induction hypothesis, there exists MSO-sentences  $\psi$  such that  $s \models \psi$  if and only if  $s \in L(R')$ . The trouble of using *delimiters* as in the case of concatenation using individual variables does not work as the operation  $*$  may require arbitrarily many delimiters. However, using set variables, we can designate the delimiters *simultaneously* no matter how many substrings, each of which is generated by  $R'$ . Let us introduce a free set variable  $Z$ , which shall be interpreted as the set of last positions of substrings  $s_i$  when  $s$  is written as  $s_1 \cdot s_2 \cdots s_n$ , each  $s_i \in L(R')$ . Clearly,  $s$  is a string generated by  $R = (R')^*$  if and only if  $s$  can be written in this way for some  $n \geq 1$  or  $s = \epsilon$ .

Next, we want to talk about the *interval between the delimiters*. Specifically, we want to define the substring  $s_i$  for each  $i$ . For this, we need a formula with free set variable  $Z$  and  $I$  which tests if (i)  $I$  is indeed an interval, i.e. contiguous, (ii) the *maximum* element in  $I$  is a delimiter, i.e. belongs to  $Z$ , and (iii) there is a unique element in  $I$  that belongs to  $Z$ . Let  $\varphi_{good}(I, Z)$  be such a formula.

We also define a formula which says that an interval  $I$  is maximally good; for this we can use the following formula:

$$\varphi_{maximal}(I, Z) := \forall x (x < I \rightarrow \exists z (z \notin I \wedge z \in Z \wedge x \leq z))$$

We define a formula  $\varphi_{\max}(z, I)$  which says that  $z$  is the maximum element in the set  $I$ . The following formula serves this purpose:

$$\varphi_{\max}(z, I) := \forall x (x \in I \rightarrow x \leq z).$$

Using  $\varphi_{\max}(z, I)$  and  $\varphi_{maximal}(I, Z)$ , it is an easy exercise to write an MSO-formula  $\varphi_{dlm}(Z)$  which evaluates to TRUE for some delimiters  $Z$  if and only if each *maximal interval*  $I$  w.r.t  $Z$  satisfies the formula  $\psi$ :

Now

$$\varphi_{dlm}(Z) := \forall I (\varphi_{good}(I, Z) \wedge \varphi_{maximal}(I, Z) \rightarrow \psi^{int}(I))$$

and here,  $\psi^{int}(I)$  is a *relativization of  $\psi$*  with respect to  $I$ . The idea is the same as in the case of concatenation. We want to *activate* an interpretation of a variable in  $\varphi$  only when the interpretation is confined to the interval  $I$ . The implementation of  $\psi^{int}(I)$  with an actual MSO-sentence is an easy exercise.

Finally, we write a sentence which defines the language generated by the regular expression  $(R')^*$ :

$$\varphi := \nexists x (x = x) \vee (\exists Z (\varphi_{dlm}(Z) \wedge \forall z \varphi_{\max}(z) \rightarrow z \in Z))$$

where  $\varphi_{\max}(z)$  is a formula which checks if  $z$  is the maximum element in the entire universe; one can obtain the formula as a simple modification of  $\varphi_{\max}(z, I)$ .

◇ Backward implication. Suppose that  $\psi$  is an MSO-sentence which defines a language  $L \subseteq \Sigma^*$  and let  $k := \text{qr}(\psi)$ .

**Claim 1.** *Let  $\psi$  is an MSO-sentence with  $k := \text{qr}(\psi)$  which defines  $L \subseteq \Sigma^*$ . Then, for any string  $w \in \Sigma$ , it holds that  $\text{mso-type}_k(w) \in \{\text{mso-type}_k(s) \mid s \in L\}$  if and only if  $w \models \psi$ .*

PROOF OF THE CLAIM: Recall that  $w \in L$  if and only if  $w \models \psi$ , which implies the backward implication. To see the forward direction, suppose that  $\text{mso-type}_k(w) = \text{mso-type}_k(s)$  for some  $s \in L$ . As  $\psi$  has quantifier rank at most  $k$ , it holds that  $s \models \psi$ , or equivalently the sentence  $\psi$  is contained in  $\text{mso-type}_k(s)$ . Therefore,  $\psi \in \text{mso-type}_k(w)$ , finishing the proof. ◇

Consider the 5-tuple  $M = (Q, \Sigma, \delta, q_0, F)$ , where

- $Q$  is the set of all MSO rank- $k$  types (0, 0-types).
- $q_0$  is  $\text{mso-type}_k(\epsilon)$ .
- $F$  is the set of  $\{\text{mso-type}_k(s) \mid s \in L\}$ .
- $\delta$  is a function from  $Q \times \Sigma \rightarrow Q$  defined as

$$\delta(q, a) = \text{mso-type}_k(sa) \quad \text{if there is a string } s \in \Sigma^* \text{ s.t. } \text{mso-type}_k(s) = q$$

First, we claim that  $M$  defines a deterministic finite automaton. For this, it suffices to show that (i) for every  $q \in Q$  and for every  $a \in \Sigma$ ,  $\delta(q, a)$  has a unique value and it belongs to  $Q$ , and (ii) the number of states  $Q$  is finite. Indeed, if  $\text{mso-type}_k(s) = \text{mso-type}_k(s')$ , then we have  $\text{mso-type}_k(sa) = \text{mso-type}_k(s'a)$  for any  $a \in \Sigma$  by Lemma 8. This means that, if  $q = \text{mso-type}_k(s)$  for some string  $s$ , then  $\delta(q, a)$  is uniquely defined for every  $a \in \Sigma$ . By Lemma 4, there exists a string  $s$  such that  $q = \text{mso-type}_k(s)$  for every  $q \in Q$ . Therefore,  $\delta(q, a)$  has a unique value for every  $(q, a) \in Q \times \Sigma$ . To see that  $\delta(q, a) \in Q$ , notice that the string  $sa$  certifies  $\text{mso-type}_k(sa) \in Q$ , thus  $\delta(q, a) \in Q$ , whenever  $\text{mso-type}_k(s) = q$ . This proves (i). That (ii) holds is immediate from Observation 3.

Secondly, we want to show that  $L(M) = L$ . For this we use the following claim.

**Claim 2.** *The run of  $M$  on a string  $s \in \Sigma^*$  ends in  $\text{mso-type}_k(s)$ .*

PROOF OF THE CLAIM: We prove by induction on  $|s|$ . When  $s = \epsilon$ , then the claim trivially holds by definition of  $q_0$ , the start state. Let  $a \in \Sigma$  be the symbol such that  $s = s'a$ . By induction hypothesis,  $\text{mso-type}_k(s')$  is the last state of the run of  $M$  on  $s'$ . Now after reading the symbol  $a$ , the transition function  $\delta$  updates the state from  $\text{mso-type}_k(s')$  to  $\text{mso-type}_k(s'a)$  by construction of  $\delta$ . ◇

It remains to observe that Claim 2 and Claim 1 establish  $L(M) = L$ . □

An alternative proof of the forward implication in Theorem 7 which constructs an MSO-sentence *simulating* an accepting run of the automaton  $M$  recognizing  $L$  is presented in [3].

## References

- [1] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960.

- [2] Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98(1):21–51, 1961.
- [3] Leonid Libkin. *Elements of Finite Model Theory*. Springer, 2004.