Contents

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 ψ 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* MSO$[k]$

For a τ-structure $\mathbb{A}$ one can consider all MSO-sentences over τ satisfied by $\mathbb{A}$. There are infinitely many MSO-sentences and such a set consisting of all true MSO-sentences in $\mathbb{A}$ 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. Such a set is one of the crucial defining features of the relational structure $\mathbb{A}$, two τ-structures which are equivalent in this way have a powerful 'exchangeabiliy' property. This is the key property for proving results regarding MSO-logic.

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

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

Notice that if $\textsf{mso-type}_k(\mathbb{A})$ *is the set of all* MSO*-sentences of quantifier rank at most k which holds on $\mathbb{A}$.*

An MSO rank-k ℓ, m-type *is a set S of* MSO*-formulas in* MSO$[k]$ *with ℓ free individual variables and m free set variables such that*

- CONSISTENCY*: there exist a τ-structure $\mathbb{A}$ and an ℓ-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 ψ with ℓ free individual variables and m free set variables, exactly one of ψ or $\neg\psi$ is included in the set*

It is not difficult to see that for each fixed k, l, m, there are finitely many (bounded by a function of k) MSO-formulas of quantifier rank up to k with ℓ 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. The number of atomic formulas is bounded by a function of ℓ, m and τ, 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, l, m, the number of* MSO *rank-k ℓ, m-types is finite, determined solely by k, ℓ, m and the vocabulary τ.*

Note that for τ-structure $\mathbb{A}$, $\textsf{mso-type}_k(\mathbb{A})$ is an MSO rank-k type; the consistency of the set $\textsf{mso-type}_k(\mathbb{A})$ is witnessed by the very structure $\mathbb{A}$. The completeness of $\textsf{mso-type}_k(\mathbb{A})$ is clear from the fact that $\textsf{qr}(\psi) = \textsf{qr}(\neg\psi)$ for any formula ψ 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 τ-structure $\mathbb{A}$ such that $q = \textsf{mso-type}_k(\mathbb{A})$.

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

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

Definition 5 (Disjoint union on τ-structures)**.** *When the vocabulary τ contains only the predicates and no constant symbols*[1]*, the* disjoint union $\mathbb{A} \cup \mathbb{B}$ *of τ-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 τ-structures such that* $\textsf{mso-type}_k(\mathbb{A}) = \textsf{mso-type}_k(A')$ *and* $\textsf{mso-type}_k(\mathbb{B}) = \textsf{mso-type}_k(\mathbb{B}')$*. Then it holds that* $\textsf{mso-type}_k(\mathbb{A} \cup \mathbb{B}) = \textsf{mso-type}_k(\mathbb{A}' \cup \mathbb{B}')$.

[1]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[2]. 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 Σ. If*

$$\textsf{mso-type}_k(s_1) = \textsf{mso-type}_k(s_2) \quad \textit{and} \quad \textsf{mso-type}_k(s_1') = \textsf{mso-type}_k(s_2'),$$

then it holds that $\textsf{mso-type}_k(s_1 \cdot s_1') = \textsf{mso-type}_k(s_2 \cdot s_2')$

Proof of Theorem 7:
$\diamond$ 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 singleton consisting of a symbol from $\Sigma \cup \epsilon$:

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

- $R = \{\epsilon\}$:

$$\varphi_a := \neg\exists x(x = x).$$

- $R = \emptyset$:

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

- $R = R_1 \cup R_2$: by induction hypothesis, there exists MSO-sentences φ_1 and φ_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 φ' 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 φ_1 and φ_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 φ_1 in s. But what we actually want is to evaluate φ_1 on the substring $s[1 : z]$, i.e. *up to some position* z. Likewise, we want to evaluate φ_2 on the substring $s[z + 1, n]$. For this, we need to modify the original sentence φ_1 defining $L(R_1)$ so that, even when the variables are interpreted in s, in practice its interpretation is confined to the prefix. We can achieve this effect by replacing every occurrence

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

[2]In a more genera form, Feferman-Vaught Theorem is known.

in φ_1 using z as a free individual variable. A similar relativization can be applied to the universal quantifiers in φ_1.

- $\forall x\ \psi$ by $\forall x(\neg(x \leq z) \vee \psi)$, and (read: "if $x \leq z$ then ψ holds")
- $\forall X\ \psi$ by $\forall X(\exists x(x \in X \wedge x > z)) \vee \psi$ (read: "if $X \leq z$ then ψ holds").

A symmetric modification applies to φ_2. Apparently, the free variable z point at the last position of the prefix to match a string from $L(R_1)$. Let $\varphi_1^{pf}(z)$ and $\varphi_2^{sf}(z)$ be the respective formulas obtained in this way. Then 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 ψ such that $s \models \varphi'$ 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 lead to arbitrarily many delimiters. However, using set variables, we can designate the delimiters *simultaneously* no matter how many substrings you need so as to be generated by R'. Let us introduce a free set variable Z, which shall be interpreted as the set of last positions of $s_i's$ when s is written as $s_1 \cdot s_2 \cdot \dots \cdot s_n$, each $s_i \in L(R')$.

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 a free set variable 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. Let $\varphi_{int}(I)$ be such a formula.

Now

$$\varphi := \exists Z\ \forall I\ \varphi_{int}(I, Z) \rightarrow \psi^{int}(I)$$

and here, $\psi^{int}(I)$ is a *relativization of* ψ 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 φ only when the interpretation is confined to the interval I. The implementation of $\varphi_{int}(I)$ and $\psi^{int}(I)$ with an actual MSO-sentence is an easy exercise.

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

Claim 1. *Let ψ is an MSO-sentence with $k := \mathsf{qr}(\psi)$ 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 ψ has quantifier rank at most k, it holds that $s \models \psi$, or equivalently the sentence ψ is contained in $\text{mso-type}_k(s)$. Therefore, $\psi \in \text{mso-type}_k(w)$, finishing the proof. $\diamondsuit$

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 $\{\text{mso-type}_k(s) \mid s \in L\}$.

- δ is a function from $Q \times \Sigma \to Q$ defined as

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

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 $\mathsf{mso\text{-}type}_k(s) = \mathsf{mso\text{-}type}_k(s')$, then we have $\mathsf{mso\text{-}type}_k(sa) = \mathsf{mso\text{-}type}_k(s'a)$ for any $a \in \Sigma$ by Lemma 8. This means that, if $q = \mathsf{mso\text{-}type}_k(s)$ 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 $q = \mathsf{mso\text{-}type}_k(s)$ 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 $\mathsf{mso\text{-}type}_k(sa) \in Q$, thus $\delta(q, a) \in Q$, whenever $\mathsf{mso\text{-}type}_k(s) = q$. 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* $\mathsf{mso\text{-}type}_k(s)$.

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, $\mathsf{mso\text{-}type}_k(s')$ is the last state of the run of M on s'. Now after reading the symbol a, the transition function δ updates the state from $\mathsf{mso\text{-}type}_k(s')$ to $\mathsf{mso\text{-}type}(s'a)$ by construction of δ. $\Diamond$

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

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.