METAREGULAR LANGUAGES: A LOGICAL POINT OF VIEW

Mikhail Kats and Wolfgang Thomas
Lehrstuhl für Informatik VII, RWTH Aachen
D-52056 Aachen, Germany
e-mail: {kats,thomas}@i7.informatik.rwth-aachen.de

ABSTRACT

In this paper we study, in the framework of mathematical logic, the class $MetaReg$ of metaregular languages, i.e. the class of languages accepted by finite Time-Variant Automata (TVA) in the sense of Agasandyan [1] and Salomaa [6]. We set up a correspondence (in the style of Büchi-Elgot-Trakhtenbrot's Theorem for regular languages) between $MetaReg$ and $MSO[\mathcal{M}]$, i.e. the class of languages definable in Monadic Second Order logic extended by fixed monadic numerical predicates. The number of monadic numerical predicates are used in the language yields an infinite hierarchy inside $MSO[\mathcal{M}]$.

Keywords: time-variant automata; metaregular languages; second-order logic

1. Introduction

Finite time-variant automata (or, shortly, TVA) were introduced in the sixties by Gill [5] and (independenly) Agasandyan [1]. TVA are finite automata whose structure (i.e., transition and/or output relations) depends on the time instant. (For a survey on some early results see [4].)

Beginning since the early nineties, reconfigurable computing systems have been studied. Speaking in general, a reconfigurable device is a computational device that can be "reconfigured" at run-time (see e.g. [8] for the theoretical introduction). A time-variant automaton can be considered as an abstract model of such a device. A related idea appears in recent work of van Leeuwen and Wiedermann [12], [13]; they consider interactive Turing machines with advice, "site machines" and "internet machines".

The connection between finite automata and logic was established in the sixties by Büchi, Elgot and Trakhtenbrot. They showed that finite automata and monadic second-order logic (interpreted over finite words) have the same expressive power, and that the transformation from formulas to automata and vice-versa are effective. Later, such an equivalence was established also between finite automata and monadic second order logic over infinite words and trees. (For detailed information see the survey [11] and the monography [10].)

Agasandyan [1] has first established a necessary and sufficient condition, resembling Myhill-Nerode's Theorem for finite automata, for a language to be *metaregular*, i.e. acceptable by a TVA. Various results concerning the family of metaregular languages were obtained e.g. by Salomaa [6], [7], Baer and Spanier [2], Simovici [9]. In this paper we are interested in the logical description of metaregular languages. We extend the Büchi-Elgot-Trakhtenbrot Theorem to the family of metaregular languages, connecting TVA to monadic second order logic extended by numerical predicates.

2. Time-Variant Automata

The set of all words, including the empty word ϵ, over a finite alphabet Σ is denoted by Σ^*. The set of subsets of a set X is denoted by 2^X and the set of mappings of a set X into a set Y by Y^X. The set of positive integers is denoted by $\mathbf{N}$.

A *nondeterministic finite time-variant automaton* or, shortly, NTVA is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ where Q is a finite set of states, Σ is a finite alphabet, q_0 is the initial state, $\Delta\colon \mathbf{N} \to 2^{Q\times\Sigma\times Q}$ is the *dynamic transition relation*, and $F \subseteq Q$ is the set of final states.

The mapping Δ defines the infinite relation sequence $\Delta_1, \Delta_2, \ldots$, where $\Delta_i := \Delta(i)$ for $i = 1, 2, \ldots$. Given $i \in \mathbf{N}$, we say that $\Delta_i \subseteq Q \times \Sigma \times Q$ is the *transition relation at time* i of the NTVA $\mathcal{A}$.

A word $w = a_1 \ldots a_n \in \Sigma^*$ is *accepted* by $\mathcal{A}$ if there is a state sequence (a *successful run*) $p_1, \ldots, p_n, p_{n+1}$ such that $p_1 = q_0$, $(p_i, a_i, p_{i+1}) \in \Delta_i$ for $i = 1, \ldots, n$, and $p_{n+1} \in F$.

The language $\mathcal{L}(\mathcal{A})$ *recognized* by $\mathcal{A}$ consists of the words over Σ accepted by $\mathcal{A}$. A language L over Σ is *metaregular* [1] iff $L = \mathcal{L}(\mathcal{A})$ for some NTVA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$.

Remark 1 A *deterministic* finite time-variant automaton or, shortly, DTVA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$, where a function $\delta\colon \mathbf{N} \to Q^{Q\times\Sigma}$ replaces the function Δ (and a mapping $\delta_i\colon Q \times \Sigma \to Q$ replaces the relation Δ_i for $i = 1, 2, \ldots$), can be also defined. It can be proved by a straightforward adaption of the power set construction that a language L is metaregular iff it can be recognized by a DTVA $\mathcal{A}$.

Let us denote by *Reg* (*MetaReg*) the family of all (meta)regular languages. It is clear that $Reg \subset MetaReg$. This inclusion is strict, as e.g. any language over one letter is metaregular, and thus the family *MetaReg* is not denumerable. For some other properties of the family *MetaReg* see Salomaa [6], [7].

3. Referenced Automata

Referenced automata was introduced by Baer and Spanier [2]. They are the same as time-variant automata, but they permit to separate the "static" part of a TVA from the "dynamic" one. Let Γ be a finite set. An *referenced tape* over Γ is an infinite sequence $R = r_1 r_2 r_3 \ldots$, where the r_i are elements of Γ.

A *nondeterministic referenced automaton* or, shortly, NRA is a tuple $\mathcal{B} = (Q, \Sigma, \Gamma, q_0, \Phi, F, R)$, where Q is a finite set of states, Σ is a finite work alphabet, Γ is a finite reference alphabet, q_0 is the initial state, $\Phi \subseteq Q \times \Sigma \times \Gamma \times Q$ is the transition relation, $F \subseteq Q$ is the set of final states, and R is a referenced tape over Γ.

A word $w = a_1 \ldots a_n \in \Sigma^*$ is *accepted* by $\mathcal{B}$ if there is a state sequence (a *successful run*) $p_1, \ldots, p_n, p_{n+1}$ such that $p_1 = q_0$, $(p_i, a_i, r_i, p_{i+1}) \in \Phi$ for $i = 1, \ldots, n$, and $p_{n+1} \in F$.

The language $\mathcal{L}(\mathcal{B})$ *recognized* by $\mathcal{B}$ consists of the words over Σ accepted by $\mathcal{B}$.

Thus the reference automaton $\mathcal{B}$ can be regarded as a finite-state automaton controled by an oracle (the referenced tape R is regarded in this situation as external oracle information).

It is easy to see that a language L is metaregular iff it is recognized by a referenced automaton.

For any Σ let $MetaReg[n]$, $n = 1, 2, \ldots$, be the collection of metaregular languages over Σ defined by referenced automata with the work alphabet Σ and the reference alphabet $\Gamma_n = \{1, \ldots, n\}$.

Proposition 1 (Baer and Spanier [2]) $MetaReg[1] \subseteq MetaReg[2] \subseteq \ldots$, and if Σ has more than one element, each inclusion is proper.

[1] The term "metaregular" is due to Baer and Spanier [2].

4. Extended Monadic Second Order Logic $MSO[\mathcal{M}]$

Let Σ be an alphabet, $w = a_1 \ldots a_n$ be a word over Σ, and $\overrightarrow{M}_m = (\hat{M}_1, \ldots, \hat{M}_m)$ be a sequence of unary predicates $\hat{M}_1, \ldots, \hat{M}_m \subseteq \mathbf{N}$. (We define $\overrightarrow{M}_0$ to be the *empty* tuple.) The pair $(w, \overrightarrow{M}_m)$ defines the *extended word model* $\underline{w}[\overrightarrow{M}_m] := (dom(w), S^w, <^w, (Q_a^w)_{a\in\Sigma}, \hat{M}_1^w, \ldots, \hat{M}_m^w)$, where $dom(w) = \{1, \ldots, n\}$ is the set of (letter) positions of w (the "domain" of w), S^w and $<^w$ are the usual successor relation and order relation on $dom(w)$, $Q_a^w := \{i \in dom(w) \,|\, a_i = a\}$, and $\hat{M}_j^w := dom(w) \cap \hat{M}_j$. The relations $S^w, <^w, \hat{M}_1^w, \ldots, \hat{M}_m^w$ are called *numerical predicates*, while the monadic predicates Q_a^w are *letter predicates.*

Given $m \in \{0, 1, 2, \ldots\}$ and an alphabet Σ, the formulas of Monadic Second Order Logic extended by m monadic numerical predicate symbols $M_1, \ldots, M_m$, $MSO[\mathcal{M}_m]$, are built up in the usual way from first-order variables $x, y, z, \ldots, x_1, x_2, \ldots$, ranging over the positions in extended word models, and second-order variables $X, Y, Z, \ldots, X_1, X_2, \ldots$, ranging over sets of positions, the atomic formulas of the form $x = y$, $x < y$ ("x is less than y"), $S(x, y)$ ("x has y as a successor"), $Q_a(x)$ ("at position x there is an a"), $X(y)$ ("$y \in X$"), and $M_i(y)$ ("$y \in M_i$") by means of usual connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and the quantifiers $\exists$ and $\forall$. The satisfaction relation $\models$ is defined accordingly. The language *defined by* the sentence φ and the tuple $\overrightarrow{M}_m$ is $\mathcal{L}(\varphi, \overrightarrow{M}_m) := \{w \in \Sigma^* \,|\, \underline{w}[\overrightarrow{M}_m] \models \varphi\}$. We denote by $MSO[\mathcal{M}_m]$ the class of laguages definable by pairs of this form, and let $MSO[\mathcal{M}] := \bigcup\{MSO[\mathcal{M}_m] \,|\, m = 0, 1, 2, \ldots\}$.

5. Metaregular Languages and $MSO[\mathcal{M}]$

The following theorem connects finite time-variant automata with logic.

Theorem 1 *A language is metaregular iff it belongs to $MSO[\mathcal{M}]$.*

Proof. The proof follows the pattern of the Büchi-Elgot-Trakhtenbrot Theorem, so we give here only an outline.

To show the direction from left to right, let L be a metaregular language and $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ be a NTVA such that $L = \mathcal{L}(\mathcal{A})$. Assume $Q = \{0, \ldots, k\}$ where $q_0 = 0$, and $\Delta(\mathbf{N}) = \{\rho_1, \ldots, \rho_m\}$. Define a tuple $\overrightarrow{M} = (\hat{M}_1, \ldots, \hat{M}_m)$ of monadic predicates in $\mathbf{N}$, where $\hat{M}_i := \{l \in \mathbf{N} \,|\, \Delta(l) = \rho_i\}$ for $i = 1, \ldots, m$. We have to find a sentence in the logic $MSO[\mathcal{M}_m]$ that expresses in any given extended word model $\underline{w}[\overrightarrow{M}]$ (over Σ) that $\mathcal{A}$ accepts w. Over a word $w = a_1 \ldots a_n$, the sentence will state the existence of a successful run $p_1, \ldots, p_{n+1}$ of $\mathcal{A}$. We may code such a state sequence up to p_n by a tuple $(X_0, \ldots, X_k)$ of pairwise disjoint subsets of $\{1, \ldots, n\}$ such that X_i contains those positions of w where state i is assumed. From the last state p_n the automaton should be able to reach a final state via the words last letter a_n. Thus, $\mathcal{A}$ accepts the nonempty word w iff

$$
\begin{aligned}
\underline{w}[\overrightarrow{M}] \models \exists X_0 \ldots \exists X_k \; & (\textstyle\bigwedge_{i\neq j} \neg\exists x(X_i(x) \wedge X_j(x)) \\
& \wedge\forall x(\mathrm{first}(x) \rightarrow X_0(x)) \\
& \wedge \textstyle\bigwedge_{l\in\{1,\ldots,m\}} \forall x \forall y (S(x, y) \wedge M_l(x) \rightarrow \\
& \qquad\qquad \textstyle\bigvee_{(i,a,j)\in\rho_l} (X_i(x) \wedge Q_a(x) \wedge X_j(y))) \\
& \wedge \textstyle\bigwedge_{l\in\{1,\ldots,m\}} \forall x(\mathrm{last}(x) \wedge M_l(x) \rightarrow \textstyle\bigvee_{\exists j\in F:(i,a,j)\in\rho_l} (X_i(x) \wedge Q_a(x)))
\end{aligned}
$$

(Here first(x) and last(x) stand for the formulas $\neg\exists y S(y, x)$ and $\neg\exists y S(x, y)$, respectively.) The empty word satisfies this sentence (with $X_i = \emptyset$). Thus, if $\mathcal{A}$ does not accept ϵ, a corresponding clause (such as $\exists x\; x = x$) should be added.

Let us briefly sketch the proof from right to left. Let $L = \mathcal{L}(\varphi, \overrightarrow{M})$ for a $MSO[\mathcal{M}_m]$-sentence φ and a tuple $\overrightarrow{M} = (\hat{M}_1, \ldots, \hat{M}_m)$. Let us now regard the predicate symbols $M_1, \ldots, M_m$ occuring in the formula φ as free monadic second-order variables. Then, following the method from [11, p. 396], we can construct for "the formula $\varphi(M_1,\dots,M_m)$ with the free occuring of $M_1,\dots,M_m$" a finite nondeterministic automaton $\mathcal{C} = (Q, \Sigma\times\Gamma, q_0, \Phi, F)$ over the alphabet $\Sigma\times\Gamma$, where $\Gamma := \{0,1\}^m$, which accepts precisely those words $W \in (\Sigma\times\Gamma)^*$ which satisfy "the formula $\varphi(M_1,\dots,M_m)$". (Such words are represented by extended word models $(\underline{w}, P_1,\dots,P_m)$ where $w \in \Sigma^*$, $P_1,\dots,P_m \subseteq dom(w)$, and are regarded as words over the extended alphabet $\Sigma\times\Gamma$.) Let us now define the reference tape $R = r_1 r_2 \dots$ over Γ, where $r_i = (r_i^1,\dots,r_i^m)$ and $r_i^j = 1 :\iff i \in \hat{M}_j$ for $i = 1,2,\dots$ and $j = 1,\dots,m$. The language $\mathcal{L}(\mathcal{C}(R))$ recognized by the referenced automaton $\mathcal{C}(R) = (Q,\Sigma,\Gamma,q_0,\Phi,F,R)$ is exactly $\mathcal{L}(\varphi,\overrightarrow{M})$. Indeed, for any word $w = a_1\dots a_n \in \Sigma^*$ we have: $w \in \mathcal{L}(\mathcal{C}(R)) \iff (a_1,r_1)\dots(a_n,r_n) \in \mathcal{L}(\mathcal{C}) \iff \underline{w}[\overrightarrow{M}] \models \varphi \iff w \in \mathcal{L}(\varphi,\overrightarrow{M})$. □

Let us now compare the power of the logics $MSO[\mathcal{M}_m]$, where $m = 0,1,2,\dots$.

Theorem 2 *Let Σ be an alphabet with more than one element, and for $m \geq 0$ let $MSO[\mathcal{M}_m]$ be the class of languages over Σ defined as in the section 4 above. Then $MSO[\mathcal{M}_0] \subset MSO[\mathcal{M}_1] \subset MSO[\mathcal{M}_2] \subset \dots$.*

Proof. Let us first note that $MSO[\mathcal{M}_m] = MetaReg[2^m]$ for $m = 0,1,2,\dots$. Indeed, for $m = 0$ we have the case of Büchi-Elgot-Trakhtenbrot's Theorem because $MetaReg[1] = Reg$. Let now $m > 0$. The inclusion $MSO[\mathcal{M}_m] \subseteq MetaReg[2^m]$ follows straightforwardly from the second part of the proof of the Theorem 1. To prove the inclusion $MetaReg[2^m] \subseteq MSO[\mathcal{M}_m]$, let us note that in the first part of the proof of the Theorem 1, $\lceil \log_2 m \rceil$ monadic predicates suffice (instead of $\hat{M}_1,\dots,\hat{M}_m$). Now the statement of the theorem follows from the Proposition 1. □

6. Additional Remarks

Theorem 1 can be transferred without essential change to the domain of ω-words, assuming a canonical definition of ω-TVA, e.g. using nondeterministic automata with Büchi acceptance condition. In this context, the evolving systems of van Leeuwen and Wiedermann [12], [13] can be directly represented, as far as finite-state systems are concerned.

Another field where numerical predicates enter as extensions of first-order or monadic second order logic is circuit complexity (see [10], [3]). In this work, also binary predicates are included (such as $BIT(i,j)$, saying that the i-th bit in the binary representation of j is 1). The meaning of such predicates in the TVA-model, in particular over ω-sequences, is not yet analyzed and understood.

References

[1] G. A. Agasandyan, Variable-structure automata. *Dokl. Akad. Nauk SSSR* **174** (1967), 529–530. (in Russian)

[2] R. M. Baer, E. H. Spanier, Referenced automata and metaregular families. *J. Comput. Syst. Sci.* **3** (1969), 423–446.

[3] D. A. M. Barrington, N. Immerman, H. Straubing, On uniformity within NC^1. *J. Comput. Syst. Sci.* **41** (1990), 274–306.

[4] W. Dauscha, G. Nürnberg, P. H. Starke, K. D. Winkler, Theorie der determinierten zeitvariablen Automaten. *Elektron. Informationsverarbeit. Kybernetik* **9** (1973), 455–511.

[5] A. Gill, Time-varying sequential machines. *J. Franklin Inst.* **276** (1963), 519–539.

[6] A. Salomaa, On finite automata with a time-variant structure. *Inform. and Control* **13** (1968), 85–98.

[7] A. Salomaa, On finite time-variant automata with monitors of different types. *Ann. Univ. Turku*, Ser. A I, **118** (1968), no. 3. 12 p.

[8] R. Sidhu, V. K. Prasanna, Efficient metacomputation using self-reconfiguration. In: M. Glesner, P. Zipf and M. Renovell (eds.) *Field-Programmable Logic and Applications. Reconfigurable Computing Is Going Mainstream*, Proc. of 12th International Conference, FPL 2002, Montpellier, France September 2-4, 2002, LNCS 2438, Springer, 2002, 698–709.

[9] D. A. Simovici, On the behaviour of time-variant semiautomata. *An. Ştiinţ. Univ. "Al. I. Cuza" Iaşi*, Secţ. Ia. **25** (1979), 187–192.

[10] H. Straubing, *Finite Automata, Formal Logic, and Circuit Complexity.* Birkhäuser, Boston, Basel, Berlin, 1994.

[11] W. Thomas, Languages, automata, and logic. In: G. Rozenberg and A. Salomaa (eds.), *Handbook of Formal Languages*, Vol. III, Springer, New York, 1997, 389–455.

[12] J. van Leeuwen, J. Wiedermann, The Turing machine paradigm in contemporary computing. In: B. Enquist and W. Schmidt (eds.), *Mathematics Unlimited - 2001 and Beyond*, Springer, Berlin, 2001, 1139–1155.

[13] J. van Leeuwen, J. Wiedermann, Beyond the Turing limit: Evolving interactive systems. In: L. Pacholski and P. Ružička (eds.), SOFSEM 2001, LNCS 2234, Springer, Berlin, Heidelberg, 2001, 90–109.