Intro

Aleksander Wisniewski

July 16, 2024

Filip: Filip: Intro, sekcja 3

1 Introduction

In *Weighted automata and weighted logic* Manfred Droste and Paul Gastin introduced weighted logics. Expressions of weighted logics make it possible to express quantitative properties of finite automata. In particular there is a subclass of weighted logics that can represent weighted automata.

Output of weighted logic expression over some word is a value from a semiring, so it defines a formal power series. We're interested in a subclass of weighted logic expressions on words over one letter alphabet. This way a weighted logic expression will define a sequence of numbers - n-th number in a sequence will be the output of the expression on a word of length n. In particular it's possible to define all linear recursive sequences in weighted logic expressions: linear recursive sequences can be defined using weighted automata, and weighted automata can be defined using weighted logic expressions.

In this work we'll look at sequences that can be defined using weighted logic expressions over one letter alphabet. We know that class of linear recursive sequences is contained in this class and we can easily prove that this containment is strict. We'll compare this class of sequences to class of polynomial recursive sequences. We'll look at possible speed of growth of these sequences and try to analyze its various properties, like behaviour modulo or whether we can easily check if it's constantly equal to zero.

2 Preliminaries

In this section we'll summarize syntax and semantics for Quantitative Monadic Second Order Logic (QMSO). It was introduced in *Quantitative Monadic Second-Order Logic* by Stephen Kreutzer and Cristian Riveros. It has the same expressive power as Weighted Logics of Droste and Gastin, but it's easier to work with. From now on we'll look at Weighted Logics only from perspective of QMSO.

2.1 Monadic Second Order Logic

Let Γ be a finite alphabet. The syntax of MSO over Γ is given by:

$$\varphi := P_a(x) \mid x \leq y \mid x \in X \mid (\varphi \vee \varphi) \mid \neg\varphi \mid \exists x.\varphi \mid \exists X.\varphi$$

where $a \in \Gamma$, x, y are first-order variables, X is a second order variable. Universal quantification can be obtained from existential quantification and negation. We can also use $\wedge$ and $\implies$ as usual.

Let $w = w_1 \dots w_n \in \Gamma*$ be a word such that $|w| = n$. We represent w as a structure $(\{1, \dots, n\}, \leq, (P_a)_{a\in\Gamma})$ where $P_a = \{i \mid w_i = a\}$. We denote by $Dom(w) = \{1, \dots, n\}$ the domain of w as a structure. Given a finite set V of first-order and second-order variables, a (V, w)-assignment σ is a function that maps every first order variable in V to $Dom(w)$ and every second order variable in V to $2^{Dom(w)}$. Furthermore, we denote by $\sigma[x \to i]$ the extension of the (V, w)-assignment σ such that $\sigma[x \to i](x) = i$ and $\sigma[x \to i](y) = \sigma(y)$ for all variables $y \neq x$.The assignment $\sigma[X \to I]$, where X is a second-order variable and $I \subseteq Dom(w)$, is defined analogously. Consider an MSO-formula φ and a (V, w)-assignment σ where V is the set of free variables of φ. We write $(w, \sigma) \models \varphi$ if (w, σ) satisfies φ using the standard MSO-semantics.

2.2 Semirings

A semiring signature $\xi := (\oplus, \odot, 0, 1)$ is a tuple containing two binary function symbols $\oplus, \odot$, where $\oplus$ is called the addition and $\odot$ the multiplication, and two constant symbols 0 and 1. A semiring over the signature ξ is a ξ-structure $\mathbb{S} = (S, \oplus, \odot, 0, 1)$, where $(S, \odot, 0)$ is a commutative monoid, $(S, \odot, 1)$ is a monoid, multiplication distributes over addition, and $0 \odot s = s \odot 0 = 0$ for each $s \in S$. If the multiplication is commutative, then we say that $\mathbb{S}$ is commutative.

Semirings we'll be interesed in in this paper include:

- semiring of natural numbers: $\mathbb{N} = (\mathbb{N}, +, \cdot, 0, 1)$
- semiring of rational numbers: $\mathbb{Q} = (\mathbb{Q}, +, \cdot, 0, 1)$
- finite semirings of numbers modulo arbitrary k: $(\{0, 1, \dots, k-1\}, +, \cdot, 0, 1)$ where operations are executed modulo k

2.3 Quantitative Monadic Second-Order Logic

(Syntax) The formulas of Quantitative Monadic Second-Order Logic (QMSO) over semiring $\mathbb{S}$ and alphabet Γ (QMSO$[\mathbb{S}, \Gamma]$) are defined by the following grammar:

$$\theta := \varphi \mid s \mid (\theta \oplus \theta) \mid (\theta \odot \theta) \mid \Sigma_x \, . \, \theta \mid \Pi_x \, . \, \theta \mid \Sigma_X \, . \, \theta \mid \Pi_X \, . \, \theta$$

where $\varphi \in MSO[\leq, (P_a)_{a\in\Gamma}]$, $s \in \mathbb{S}$, x is first-order variable and X is second-order variable.

(Semantics) Let $w = w_1 \dots w_n \in \Gamma^*$ where $n = |w|$. For the Boolean level φ, the semantics is the usual semantics of MSO, i.e. for any assignment σ,

$$[\![\varphi]\!](w,\sigma) = \begin{cases} 1 & \text{if } (w,\sigma) \models \varphi \\ 0 & \text{otherwise} \end{cases}$$

The semantics of the semiring level is defined as follows:

$$\begin{aligned}
[\![s]\!](w,\sigma) &:= s \\
[\![(\theta_1 \oplus \theta_2)]\!](w,\sigma) &:= [\![\theta_1]\!](w,\sigma) \oplus [\![\theta_2]\!](w,\sigma) \\
[\![(\theta_1 \odot \theta_2)]\!](w,\sigma) &:= [\![\theta_1]\!](w,\sigma) \odot [\![\theta_2]\!](w,\sigma) \\
[\![\Sigma_x \,.\, \theta]\!](w,\sigma) &:= \oplus_{i=1}^{n} [\![\theta]\!](w,\sigma[x \to i]) \\
[\![\Pi_x \,.\, \theta]\!](w,\sigma) &:= \odot_{i=1}^{n} [\![\theta]\!](w,\sigma[x \to i]) \\
[\![\Sigma_X \,.\, \theta]\!](w,\sigma) &:= \oplus_{I \subseteq [1,n]} [\![\theta]\!](w,\sigma[X \to I]) \\
[\![\Pi_X \,.\, \theta]\!](w,\sigma) &:= \odot_{I \subseteq [1,n]} [\![\theta]\!](w,\sigma[X \to I])
\end{aligned}$$

2.4 QMSO over one letter alphabet

In this work we focus on expressions of QMSO logic over one letter alphabet. Let's call these expressions **1QL expressions**. In this case we only use a small fragment of MSO logic, in particular we don't care about letter predicates $(P_a)_{a \in \Gamma}$, as there's only one such predicate and it's true on every element of structure. For simplification let's assume that our one letter alphabet will always be following $\Gamma = \{a\}$.

1QL expressions with no free variables (1QL sentences) generate sequences of numbers. Suppose we have a 1QL sentence φ, then we can generate corresponding sequence $a(n) = [\![\varphi]\!](a^n,\sigma)$, where σ is an empty valuation function (there are no free variables). Class of sequenecs definable 1QL expressions is called **1QL sequences**.

2.5 Weighted automata

TODO??

2.6 Polynomial Recursive Sequences

TODO??

3 1QL sequences rate of growth

It is easy to see that linear recursive sequences (LRS) have rate of growth bounded by $2^{O(n)}$. In 1QL we can define sequence $n!$, so it already proves that LRS are contained strictly in 1QL (as it's not possible to define $n!$ in LRS) and that 1QL can have higher rate of growth - $n!$ grows faster than $2^{O(n)}$. 1QL expression for $n!$ looks like following:

$$\Pi_{x_1}\Sigma_{x_2}\ .\ (x_2 \leq x_1) \cdot 1$$

For given n it works as following: x_1 iterates over $1, \ldots, n$. For some $x_1 = k$ expression $\Sigma_{x_2}\ .\ (x_2 \leq x_1) \cdot 1$ will have value exactly k. So the whole expression will be $1 \cdot 2 \cdot \ldots \cdot n$.

Using similar expression it's possible to define a sequence n^n, which differentiates 1QL from polynomially recursive class of sequences:

$$\Pi_{x_1}\Sigma_{x_2}\ .\ 1$$

It's also quite simple to create a sequence 2^{2^n} - doubly exponential. It would look as following:

$$\Pi_{X_1}\ .\ 2$$

For given n, there are 2^n valuations of X_1. So the value is 2 multiplied by itself 2^n times - 2^{2^n}.

We'll argue that 1QL sequences rate of growth is actually bounded by $2^{2^{kn}}$:

Lemma 3.1 (1QL growth rate)**.** 1QL sequences maximal growth rate is bounded by $2^{2^{kn}}$, where k can be an arbitrary big constant.

Proof. We want to create 1QL expressions giving maximal possible rate of growth. Let's simplify this problem a bit. We'll analyze 1QL expressions of form:

$$Q_{1Y_1}Q_{2Y_2}\ldots Q_{kY_k}\ .\ 2$$

Where $Q_i \in \{\Sigma, \Pi\}$ and each Y_i is either first-order or second-order variable. Innermost expression is just a constant 2 - it doesn't make sense to have any MSO expressions here, as those only filter out some variable evaluations. Constant might be other than 2, it doesn't affect rate of growth class.

Let's explain why it is enough to consider only expressions in simplified form. If we add or multiply 1QL expressions, we won't increase growth class. The other thing is that our simple expression is in a form resembling prenex normal form. Let's look at some expression that is not in such a form:

$$Q_{1Y_1}Q_{2Y_2}\ldots Q_{jY_j}\ .\ ((Q_{j'Y_{j'}}\ldots Q_{k'Y_{k'}}\ .\ 2) + (Q_{j''Y_{j''}}\ldots Q_{k''Y_{k''}}\ .\ 2))$$

Mark $E_1 = (Q_{j'Y_{j'}}\ldots Q_{k'Y_{k'}}\ .\ 2)$, $E_2 = (Q_{j''Y_{j''}}\ldots Q_{k''Y_{k''}}\ .\ 2)$. Without loss of generality assume that $E_1 \geq E_2$. Then $Q_x E_1 \geq E_1 + E_2$. Thus we can get rid of such addition by nesting quantification more, getting faster growing sequence as a result. The same argument would work for multiplication instead of addition, but we would need to use Π_x, Π_X quantifiers instead of arbitrary quantifiers.

Now we see that in order to analyze 1QL sequences rate of growth we can just analyze maximal rate of growth of expressions as specified on beginning of this proof. Let's do it.

Suppose we have quantification of depth k. There are four possible kinds of quantification: Σ_x, Σ_X, Π_x, Π_X. We can argue that it only makes sense to consider Π_X quantifications. No matter which Q_{iY_i} quantifier we consider, its result will be sum (for Σ) or multiplication (for Π) of n (for first-order variable) or 2^n (for second-order variable) identical subexpressions, each having value at least 2. We'll always achieve largest values multiplying 2^n subexpressions.

From this we see that for arbitrary 1QL expression Ψ we can create an expression Φ of form:

$$\Pi_{X_1}\Pi_{X_2}\dots\Pi_{X_k}\ .\ 2$$

for some k, such that Φ has greater rate of growth than Ψ.

Expression

$$\Pi_{X_1}\Pi_{X_2}\dots\Pi_{X_k}\ .\ 2$$

gives us a sequence defined by $a(n) = 2^{2^{kn}}$. It can be easily seen by unfolding quantifiers starting from the innermost one.

We finally get that maximal rate of growth of 1QL expressions is doubly exponential: all 1QL expressions can be upper bounded by expression of form $\Pi_{X_1}\Pi_{X_2}\dots\Pi_{X_k}\ .\ 2$, and this expression has doubly exponential rate of growth. □

4 1QL sequences over finite semirings

In order to compare 1QL sequences with other classes of sequences, it's useful to consider what is the behavior of these sequences modulo arbitrary natural numbers.

Lemma 4.1 (1QL sequences modulo)**.** Given a sequence $a(n)$ defined by 1QL expression φ over natural semiring $(\mathbb{N}, +, \cdot, 0, 1)$, a sequence $a(n) \mod c$, $c \in \mathbb{N} - \{0\}$, can be defined by 1QL expression φ', such that:

1. φ' is a 1QL expression over semiring $(\{0, \dots, c-1\}, +, \cdot, 0, 1)$, where addition and multiplication are done modulo c
2. every constant k appearing in φ is replaced by $k \mod c$ in φ'

Proof. Proof!! □

Knowing this we can characterize behavior of 1QL sequences modulo - those are sequences defined by 1QL expressions over finite semirings. It would be useful to know if it's possible to define inverse images of finite semirings. For this, a definition of recognizable step function is useful (FROM DROSTE PAPER).

Definition 4.1 (Recognizable step function)**.** Series $S : A^* \to K$ is a *recognizable step function*, if $S = \sum_{i=1}^{n} . \varphi_{L_i} \cdot k_i$ for some $n \in \mathbb{N}$, $k_i \in K$ and regular languages $L_i \subseteq A^*$ $(i = 1, \dots, n)$. φ_{L_i} is an MSO formula recognizing language L_i.

It would be simpler if languages $\{L_i : i \in \{1, \dots, n\}\}$ were disjoint, so that we don't need to do any addition. For this we can define *simple recognizable step functions*:

Definition 4.2 (Simple recognizable step function)**.** A *simple recognizable step function* is a step function $S = \sum_{i=1}^{n} . \varphi_{L_i} \cdot k_i$ in which all languages $\{L_i : i \in \{1, \dots, n\}\}$ are disjoint.

A recognizable step function can be transformed into simple recognizable step function:

Lemma 4.2. Recognizable step functions and simple recognizable step functions are the same functions.

Proof. Suppose we have a recognizable step function $S = \sum_{i=1}^{n} . \varphi_{L_i} \cdot k_i$. We can create a simple recognizable step function defining the same function in following way: ... TODO □

We can now characterize 1QL sequences modulo behavior:

Lemma 4.3. 1QL sequences modulo are simple recognizable step functions

Proof. Droste says:

1. Series definable by weighted automata over finite semirings are recognizable step functions
2. Series definable by QMSO sentences over finite semirings are the same as series definable by weighted automata over finite semirings

We know that 1QL sequences modulo are defined by 1QL expressions over finite semirings, so in particular those are recognizable step functions. So they are simple recognizable step functions. □

Corollary 4.1. Inverse images of values of 1QL sequences modulo are regular languages

Corollary 4.2. 1QL sequences modulo are ultimately periodic

Corollary 4.3. Catalan numbers sequence is not 1QL-definable

5 1QL sequences modulo constant 0

6 1QL on first order logic

It might be interesting to look into following restriction of 1QL expressions: we can only use first order variables on semiring level and logical level (i.e. we move from MSO logic to FO logic).

First, let's focus on behavior of these sequences modulo.

As stated in Droste paper, first order definable series coincide with series definable by weighted automata when working with aperiodic semirings. Unfortunately, semirings we're working with - semirings for modulo operations - aren't aperiodic. It can be easily with sequence defined by following expression:

$$\Sigma_x \ . \ 1$$

It defines the following sequence modulo 2: $\{1, 0, 1, 0, 1, \ldots\}$, which is not aperiodic. What's even more interesting is that inverse images of values of this sequence are not FO-definable languages. For value 0 inverse image is a language of words of even length, which is only definable in second order logic.

We might want to ask if it's possible to define following sequence in 1QL modulo 2 using FOL: $\{0, 0, 1, 0, 0, 1, \ldots\}$, that is: we have ones on positions divisible by 3.