Intro

Aleksander Wisniewski

February 26, 2024

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 Quantitative Monadic Second Order Logic

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 gives us the same expressive power as Weighted Logics, but it's easier to work with and explain. From now on we'll look at Weighted Logics only from perspective of QMSO.

QMSO is an extension of Monadic Second Order Logic (MSO) that can express quantitative properties of finite words. Let Γ be a finite alphabet. The syntax of MSO overy Γ is as following: (skopiowane z pracy QMSOL)

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

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 $\rightarrow$ 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 $P_a = \{i | w_i = a\}$. 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..... TODO

Semirings: TODO

QMSO Syntax: TODO

QMSO Semantics: TODO

From now on we'll only look at expressions of QMSO logic over one letter alphabet. Let's call such expressions 1QL expressions for simplification. Let's also establish that our alphabet is simply $\{a\}$. Sequence for a given 1QL expression is value of this expression on words $a, aa, aaa, \dots$. Let's call a class of sequences that can be defined using 1QL expressions 1QL sequences.

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 easily 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 $1, \dots, n$. 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 $1 \cdot 2 \cdot \dots \cdot n$.

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^n}:

Lemma 3.1 (1QL growth rate)**.** 1QL sequences maximal growth rate is bounded by 2^{2^n}.

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 variables evaluations. Constant might be other than 2, but we'll see that it doesn't change growth class in the end.

It's easy to see that if we add or multiple 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:

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

We can do following, quite crude construction which will give us simple expression with greater growth rate:

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

What we did is we increased depth of quantifiers by sum of depth of quantifiers of two inner subexpressions.

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 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:

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

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

Expression

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

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

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 $\Pi_{X_1}\Pi_{X_2}\ldots\Pi_{X_k}\ .\ 2$, and this expression has doubly exponential speed of growth. □

4 1QL sequences modulo eventual periodicity

In this section we'll analyze sequences created with 1QL expressions modulo p, for arbitrary p. In particular we want to answer following question: is every such sequence eventually periodic and how would we go about finding a period for given sequence?

A sequence is eventually periodic if there exists N, t such that for all $n \geq N$ $a(n) = a(n+t)$. That means: at some point sequence will start to be periodic.

Lemma 4.1 (1QL sequences modulo eventual periodicity)**.** 1QL sequences are eventually periodic modulo 2.

Proof. Let's start with 1QL expressions without quantification on semiring level. In general $\square$