Axiomatising Weighted Monadic Second-Order Logic on Finite Words

Antonis Achilleos and Mathias Ruggaard Pedersen

ICE-TCS, Department of Computer Science, Rekjavík University

1 Introduction

Weighted automata constitute a popular and useful framework for specifying and modelling the behaviour of quantitative systems. In order to reason about such systems, a logical description language is often used. One such logic is monadic second-order logic (MSO), which is known to describe exactly the behaviour of unweighted automata by a theorem of Büchi, Elgot, and Trakhtenbrot [1, 4, 10]. A more recent result by Droste and Gastin [2] showed that a weighted extension of MSO captures the behaviour of weighted automata in a similar fashion. While this result has been extended in many different ways, a proper analysis of this logic in the form of axiomatisation, satisfiability, and model checking issues has not yet surfaced.

One of the difficulties of such an analysis is that the weighted MSO (wMSO) is interpreted over an arbitrary set of values, rather than a standard Boolean setting. This means that each formula is a function which takes a model and returns a value, rather than something which may or may not be satisfied by a given model. We therefore investigate how to extend familiar concepts such as completeness, validity, and satisfiability to this non-Boolean, real-valued setting.

We document here some of our on-going work on these issues, including presenting equational systems that give a complete axiomatisation of a fragment of weighted MSO as well as algorithms for some of the variants of satisfiability checking in this setting.

2 Syntax and Semantics of wMSO

Following [5], we define the syntax and semantics of wMSO as follows. Consider a finite set of first-order variables $\mathcal{V}_{\mathsf{FO}}$, a finite set of second-order variables $\mathcal{V}_{\mathsf{MSO}}$, a finite alphabet Σ, and an arbitrary set $\mathcal{R}$ of weights. Note that we assume no structure on $\mathcal{R}$, it does not even have to be a semiring. The syntax of wMSO is given by the following grammar, divided into two layers.

$$\varphi ::= \top \mid P_a(x) \mid x \le y \mid x \in X \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \forall x \varphi \mid \forall X \varphi \qquad (\mathsf{MSO})$$
$$\Psi ::= r \mid \varphi \,?\, \Psi_1 : \Psi_2 \qquad (\mathsf{wMSO})$$

Here, $a \in \Sigma$, $r \in \mathcal{R}$, $x, y \in \mathcal{V}_{\mathsf{FO}}$, and $X \in \mathcal{V}_{\mathsf{MSO}}$. The first layer, MSO, is simply MSO on finite words. The second layer, wMSO, allows one to condition on MSO formulas, and choose different values of $\mathcal{R}$ depending on the truth of the MSO formulas.

The MSO formulas are interpreted in the standard way over words $w \in \Sigma^+$ together with a valuation of this word σ, which assigns to each first-order variable a position in the word and to each second-order variable a set of such positions. We denote by Σ^+_σ the set of pairs (w, σ) where $w \in \Sigma^+$ and σ is a valuation of w, and we denote by $[\![\varphi]\!]$ the set of all pairs $(w, \sigma) \in \Sigma^+_\sigma$ that satisfies φ.

$$\Gamma \vdash r \approx r \qquad \Gamma \vdash \Psi \approx \varphi \,?\, \Psi : \Psi \qquad \Gamma \vdash \neg\varphi \,?\, \Psi_1 : \Psi_2 \approx \varphi \,?\, \Psi_2 : \Psi_1$$
$$\Gamma \cup \{\varphi\} \vdash \Psi_1 \approx \Psi_2, \quad \text{if } \Gamma \vdash \Psi_1 \approx \Psi_2 \qquad \Gamma \vdash \varphi \,?\, \Psi_1 : \Psi_2 \approx \Psi_1, \quad \text{if } \Gamma \vdash \varphi \leftrightarrow \top$$
$$\Gamma \vdash \varphi \,?\, \Psi_1 : \Psi_2 \approx \Psi, \quad \text{if } \Gamma \cup \{\varphi\} \vdash \Psi_1 \approx \Psi \text{ and } \Gamma \cup \{\neg\varphi\} \vdash \Psi_2 \approx \Psi$$

Table 1: Axioms for wMSO.

The semantics of formulas Ψ of wMSO is given by a function $[\![\cdot]\!] : \Sigma_\sigma^+ \to \mathcal{R}$, defined by

$$[\![r]\!]\,(w,\sigma) = r \qquad [\![\varphi \,?\, \Psi_1 : \Psi_2]\!]\,(w,\sigma) = \begin{cases} [\![\Psi_1]\!]\,(w,\sigma) \text{ if } w,\sigma \models \varphi \\ [\![\Psi_2]\!]\,(w,\sigma) \text{ otherwise} \end{cases}$$

For a formula $\Psi \in$ wMSO, a given value $r \in \mathcal{R}$ can be represented by an MSO formula that describes all the strings on which Ψ returns the value r.

Definition 1. *For $\Psi \in$ wMSO and $r \in \mathcal{R}$, we define $\varphi(\Psi, r)$ recursively: $\varphi(r,r) = \top$ and $\varphi(r', r) = \neg\top$, when $r \neq r'$; and $\varphi(\psi ? \Psi_1 : \Psi_2, r) = (\psi \wedge \varphi(\Psi_1, r)) \vee (\neg\psi \wedge \varphi(\Psi_2, r))$.*

Lemma 1. *$(w,\sigma) \in [\![\phi(\Psi, r)]\!]$ iff $[\![\Psi]\!]\,(w,\sigma) = r$.*

3 Axioms

The main concern of our on-going work is to give a complete axiomatisation of wMSO. Our axiomatisation relies on an axiomatisation of MSO (or FO) on finite strings. Since satisfiability is decidable for both these logics, they have recursive and complete axiomatisations. For the case of MSO, such an axiomatisation has been given in [6], although we are not aware of a similar axiomatization for FO.

For wMSO, we first have to consider what it means to axiomatize a real-valued, non-Boolean logic. On the syntactic side, it seems natural to give an axiomatisation in terms of an equational system, denoted $\vdash \Psi_1 \approx \Psi_2$, and on the semantic side to equate two formulas that give the same value on all models, denoted $\Psi_1 \sim \Psi_2$. This also agrees with the work by Mio et al. [7] on axiomatising Riesz modal logic, which is the only other work on axiomatising real-valued logics that we know of. We augment this definition slightly by considering a set of MSO formulas Γ which we think of as assumptions. Then we write $\Gamma \vdash \Psi_1 \approx \Psi_2$ if Ψ_1 and Ψ_2 can be derived from the axioms under the assumptions Γ and $\Psi_1 \sim_\Gamma \Psi_2$ if Ψ_1 and Ψ_2 give the same values on all models that satisfy all the formulas of Γ.

We propose an axiomatization that includes the usual axioms for equality and the axioms in Table 1. The main part of the axiomatisation is concerned with axiomatising the behaviour of the conditional operator $\varphi \,?\, \Psi_1 : \Psi_2$.

Theorem 1 (Completeness of wMSO)**.** $\Psi_1 \sim_\Gamma \Psi_2$ *if and only if* $\Gamma \vdash \Psi_1 \approx \Psi_2$.

4 Further Concerns

Satisfiability For a real-valued, non-Boolean logic such as wMSO, familiar notions such as satisfiability need to be redefined, since we no longer have a satisfaction relation, but each formula is instead a function. We therefore wish to discuss and investigate how to generalise the notion of satisfiability to the real-valued setting, and determine the decidability and complexity of such notions. Plausible candidates for an extension of satisfiability is asking if a formula can return a specific value, if two formulae can return the same value, or if a formula can return a value other than a given one.

Complexity Each of these notions, as well as provability for the equational theory, can be reduced to MSO satisfiability and are thus decidable. For instance, Ψ_1 returns the same value as Ψ_2 iff

$$\bigvee_{\substack{r \text{ in } \Psi_1 \\ \text{and } \Psi_2}} \varphi(\Psi_1, r) \wedge \varphi(\Psi_2, r)$$

is satisfiable. However, one can not really hope for efficient algorithms, since the satisfiability problem for MSO on finite strings is already non-elementary [8], and MSO-satisfiability can be reduced to any of the three variants of wMSO satisfiability discussed above — for instance, φ is satisfiable iff $\varphi \mathbin{?} 1 : 0$ can take the value 1.

Another interesting theory would be the one of *inequalities*. Similarly to the above, we can see that $\Psi_1 \leq \Psi_2$ can be described by the MSO formula

$$\bigvee_{\substack{r_1 \text{ in } \Psi_1 \\ r_2 \text{ in } \Psi_2 \\ r_1 \leq r_2}} \varphi(\Psi_1, r_1) \wedge \varphi(\Psi_2, r_2),$$

and therefore, the same decidability and complexity observations can be made in this setting.

Variations Instead of using MSO formulas for conditions, one can use formulas of another logic, such as first-order logic (FO), resulting in wFO, which leads to different representation results [3]. Our complete axiomatization is agnostic with regard to the logic that one uses for conditions, as long as this logic has a complete axiomatization. One may hope to obtain better complexity results with respect to the decision problems discussed previously, by making a different choice with regards to the base logic. However, this seems unlikely in the case of FO, since, similarly to MSO, FO-satisfiability is non-elementary [8], and model checking for FO is PSPACE-complete [9, 11].

The full wMSO logic described in [5] includes a third layer called core-wMSO, which allows one to form multisets of values. This layer includes a sum over formulas indexed by a second-order variable, which corresponds to a kind of union over multisets. This sum behaves like a quantifier, so we hope that one can add axioms reminiscent of those for quantifiers of MSO in order to obtain a complete axiomatisation, although this is on-going work.

References

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

[2] Manfred Droste and Paul Gastin. Weighted automata and weighted logics. *Theor. Comput. Sci.*, 380(1-2):69–86, 2007.

[3] Manfred Droste and Paul Gastin. Aperiodic weighted automata and weighted first-order logic. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *MFCS*, volume 138 of *LIPIcs*, pages 76:1–76:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[4] Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98(1):21–51, 1961.

[5] Paul Gastin and Benjamin Monmege. A unifying survey on weighted logics and weighted automata - core weighted logic: minimal and versatile specification of quantitative properties. *Soft Comput.*, 22(4):1047–1065, 2018.

[6] Amélie Gheerbrant and Balder ten Cate. Complete axiomatizations of fragments of monadic second-order logic on finite trees. *Logical Methods in Computer Science*, 8(4), 2012.

[7] Matteo Mio, Robert Furber, and Radu Mardare. Riesz modal logic for markov processes. In *LICS*, pages 1–12. IEEE Computer Society, 2017.

[8] Klaus Reinhardt. The complexity of translating logic to finite automata. In Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors, *Automata Logics, and Infinite Games: A Guide to Current Research*, pages 231–238. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[9] Larry J. Stockmeyer. *The Complexity of Decision Problems in Automata Theory*. PhD thesis, MIT, 1974.

[10] Boris A. Trakhtenbrot. Finite automata and the logic of monadic predicates. *Doklady Akademii Nauk SSSR*, 149:326–329, 1961.

[11] Moshe Y. Vardi. The complexity of relational query languages (extended abstract). In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 137–146. ACM, 1982.