Set graphs are not definable in MSO1

Bruno Bandeira[1], **Márcia R. Cerioli**[2], **Petrucio Viana**[3]

[1]Programa de Engenharia de Sistemas e Computação – COPPE
Universidade Federal do Rio de Janeiro (UFRJ) – Rio de Janeiro – RJ – Brazil

[2]Instituto de Matemática e
Programa de Engenharia de Sistemas e Computação – COPPE
Universidade Federal do Rio de Janeiro (UFRJ) – Rio de Janeiro – RJ – Brazil

[3]Instituto de Matemática e Estatística
Universidade Federal Fluminense (UFF) – Niterói – RJ – Brazil

brunobm@cos.ufrj.br, marcia@cos.ufrj.br, petrucio_viana@id.uff.br

Abstract. *We prove that the property of being a set graph cannot be expressed in MSO*$_1$*, the language of Monadic Second Order Logic with quantification over sets of vertices, but not sets of edges. Hence, Courcelle's Theorem relating MSO*$_1$ *and clique-width cannot be applied to the set graph recognition problem.*

1. Introduction

A graph is a *set graph* if it admits an *extensional acyclic orientation* [Tomescu 2011], where an orientation of a graph is extensional if each vertex has a unique out-neighborhood, and is acyclic if it does not have any directed cycle. The graph in Figure 1 is a set graph, as it admits an extensional acyclic orientation, shown in Figure 2. The smallest connected graph that is not a set graph is the complete bipartite graph $K_{1,3}$ [Milanič and Tomescu 2013] shown in Figure 3.

The set graph recognition problem, denoted by EAO, is NP-complete even when restricted to bipartite graphs [Milanič et al. 2014] and split graphs [Monteiro 2022]. On the other hand, EAO can be solved in polynomial time when restricted to claw-free graphs [Milanič and Tomescu 2013], unicyclic graphs [Milanič and Tomescu 2013], graphs admitting Hamiltonian paths [Milanič and Tomescu 2013], and cographs [Monteiro 2022]. Moreover, EAO can be solved in linear time when restricted to graphs of bounded *treewidth* [Milanič and Tomescu 2013]. To prove this, Milanič and Tomescu expressed the property of being a set graph in MSO$_2$, the language of Monadic Second Order Logic with quantification over sets vertices and sets of edges, and applied Courcelle's Theorem [Courcelle 1990].

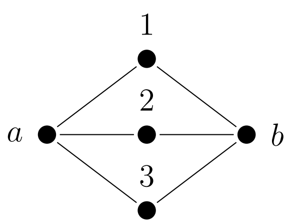

Figure 1. $K_{2,3}$

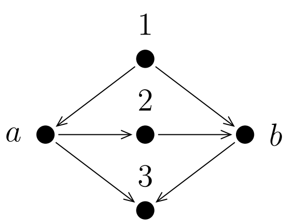

Figure 2. Oriented $K_{2,3}$

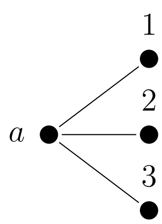

Figure 3. $K_{1,3}$

A similar theorem states that any graph property that can be expressed in the more restrictive language MSO_1, of Monadic Second Order Logic with quantification over sets of vertices, but not sets of edges, can be verified in polynomial time in graph classes with bounded *clique-width* [Courcelle et al. 2000]. Milanič and Tomescu posed the problem of determining if EAO can also be solved in polynomial time when restricted to graph classes with bounded clique-width [Milanič and Tomescu 2013]. A strategy to positively answer this question would be to express EAO in MSO_1, i.e., without quantification over sets of edges. However, as we prove in this work, EAO cannot be expressed in MSO_1.

In Section 2, we present the syntax and semantics of the language of the Monadic Second Order Logic over a vocabulary τ, and define the language MSO_1 for expressing graph properties without quantification over sets of edges. Moreover, in Subsection 2.3 we present the tools used for proving the main result of this text: translations and transductions on MSO. In Section 3 we present a proof that MSO_1 cannot express the property of being a set graph.

2. Monadic Second Order Logic

In this section we review the syntax and the semantics of monadic second order languages. Then, we define *translations schemes* between such languages, simplifying the definitions found in [Courcelle et al. 2000]. We present these notions with more generality to suggest how other inexpressibility results can be proven with similar strategies, using the same tools.

2.1. Syntax

Let τ be a *vocabulary*, that is, a finite set of relational symbols equipped with a function $\rho : \tau \to \mathbb{N}$ that associates each $r \in \tau$ with its arity $\rho(r)$. The *monadic second order language over* τ, denoted MSO_τ, has two sets of variables $Var_1 = \{x_1, x_2, x_3, \dots\}$, the set of *individual variables*, and $Var_2 = \{X_1, X_2, X_3, \dots\}$, the set of *set variables*. The MSO_τ-*formulas* are defined by the following grammar:

$$\varphi ::= Xx \mid x = y \mid rx_{i_1}x_{i_2}\dots x_{i_{\rho(r)}} \mid (\varphi_1 \wedge \varphi_2) \mid \neg\varphi_1 \mid \forall x\varphi_1 \mid \forall X\varphi_1,$$

where $x, y, x_{i_1}, x_{i_2}, \dots, x_{i_{\rho(r)}} \in Var_1$, $X \in Var_2$ and $r \in \tau$.

The connectives $\vee$, $\to$, $\leftrightarrow$ and the quantifier $\exists$ are introduced by the usual definitions. The absence of variables for relations of arity greater than 1 is what makes MSO_τ *monadic*. When $\tau = \emptyset$, MSO_τ is denoted simply as MSO.

We denote by $FV(\varphi)$ the set of variables that occur free in an MSO_τ-formula φ. We say that an MSO_τ-formula φ is an *MSO_τ-sentence* if and only if $FV(\varphi) = \emptyset$. We denote by $\varphi(x_{i_1}, \dots, x_{i_n}, X_{j_1}, \dots, X_{j_m})$ the MSO_τ-formula obtained from φ by substituting each free occurrence of x_k with x_{i_k}, for every $k \in \{1, \dots, n\}$, and each free occurrence of X_k with X_{j_k}, for each $k \in \{1, \dots, m\}$. We assume these substitutions of variables are accompanied by the possibly necessary renaming of bound variables (i.e., quantified occurrences) in φ.

2.2. Semantics

Given a language MSO_τ over a vocabulary $\tau = \{r_1, \dots, r_n\}$, a *$\tau$-structure* is a sequence $\mathcal{S} = \langle D, R_1, \dots, R_n \rangle$, where D is a non-empty set and $R_i \subseteq D^{\rho(r_i)}$, for each

$i \in \{1, \ldots, n\}$. Given a τ-structure $\mathcal{S}$, a pair (a, A), where $a : Var_1 \to D$ and $A : Var_2 \to \mathcal{P}(D)$, is called a *valuation pair on* $\mathcal{S}$.

Given a τ-structure $\mathcal{S}$ with a valuation pair (a, A), an element $u \in D$ and a set $U \subseteq D$, we denote by $a[u \to x_i]$ the valuation that agrees with a in all individual variables, except possibly in x_i, that $a[u \to x_i]$ assigns to u; and we denote by $A[U \to X_i]$ the valuation that agrees with A in all set variables, except possibly in X_i, that $A[U \to X_i]$ assigns to U.

An MSO$_\tau$-*formula* φ *to be satisfied on a* τ*-structure* $\mathcal{S}$ *with a valuation pair* (a, A), denoted $\mathcal{S}, a, A \models \varphi$, is defined recursively as follows: $\mathcal{S}, a, A \models Xx$ iff $a(x) \in A(X)$; $\mathcal{S}, a, A \models x = y$ iff $a(x)$ is $a(y)$; $\mathcal{S}, a, A \models r_j x_{i_1} \ldots x_{i_{\rho(r_j)}}$ iff $(a(x_{i_1}), \ldots, a(x_{i_{\rho(r_j)}})) \in R_j$; $\mathcal{S}, a, A \models (\varphi \wedge \psi)$ iff $\mathcal{S}, a, A \models \varphi$ and $\mathcal{S}, a, A \models \psi$; $\mathcal{S}, a, A \models \neg\varphi$ iff $\mathcal{S}, a, A \not\models \varphi$; $\mathcal{S}, a, A \models \forall x\varphi$ iff $\mathcal{S}, a[u \to x], A \models \varphi$, for each $u \in D$; $\mathcal{S}, a, A \models \forall X\varphi$ iff $\mathcal{S}, a, A[U \to X] \models \varphi$, for each $U \in \mathcal{P}(D)$. The rules for $\vee$, $\to$, $\leftrightarrow$, and $\exists$ follow from the definitions, as usual. To simplify the notation, if $FV(\varphi) \subseteq \{x_1, \ldots, x_n\}$, we write $\mathcal{S} \models \varphi(u_1, \ldots, u_n)$ instead of $\mathcal{S}, a[u_1 \to x_1] \ldots [u_n \to x_n], A \models \varphi$.

2.3. Translations between MSO languages and structures

Given two MSO vocabularies $\tau = \{r_1, \ldots, r_n\}$ and $\sigma = \{s_1, \ldots, s_m\}$, a τ*-*σ*-translation scheme* is a sequence of MSO$_\sigma$-formulas $\Phi = \langle \delta, \phi_1, \ldots, \phi_n \rangle$ where $FV(\delta) = \{x_1\}$, and $FV(\phi_i) = \{x_1, \ldots, x_{\rho(r_i)}\}$. Given a τ-σ-translation scheme Φ, we define two transformations—also denoted by Φ: one that transforms each MSO$_\tau$-formula φ into an MSO$_\sigma$-formula $\Phi(\varphi)$; and another that transforms each σ-structure $\mathcal{S}$ into a τ-structure $\Phi(\mathcal{S})$.

The Φ-*translation* of MSO$_\tau$-formulas into MSO$_\sigma$-formulas is defined recursively as follows. $\Phi(\varphi) = \varphi$ for $\varphi \in \{Xx, x = y\}$; $\Phi(r_j x_{i_1} \ldots x_{i_{\rho(r_j)}}) = \phi_j(x_{i_1}, \ldots, x_{i_{\rho(r_j)}})$; $\Phi((\psi \wedge \theta)) = (\Phi(\psi) \wedge \Phi(\theta))$; $\Phi(\neg\psi) = \neg\Phi(\psi)$; $\Phi(\forall x\psi) = \forall x(\delta(x) \to \Phi(\psi))$; $\Phi(\forall X\psi) = \forall X(\forall x(Xx \to \delta(x)) \to \Phi(\psi))$.

The Φ-*transduction* of σ-structures into τ-structures is defined as follows. Given a σ-structure $\mathcal{S} = \langle D, S_1, \ldots, S_m \rangle$, we set $\Phi(\mathcal{S}) = \langle D_\Phi, R_1, \ldots, R_n \rangle$, where $D_\Phi = \{u \in D : \mathcal{S} \models \delta(u)\}$ and $R_i = \{(u_1, \ldots, u_{\rho(r_i)}) \in D^{\rho(r_i)} : \mathcal{S} \models \phi_i(u_1, \ldots, u_{\rho(r_i)})\}$, for each $i \in \{1, \ldots, n\}$.

Lemma 1. *If* Φ *is a* τ*-*σ*-translation scheme,* φ *is an MSO*$_\tau$*-sentence and* $\mathcal{S}$ *is a* σ*-structure, then* $\mathcal{S} \models \Phi(\varphi)$ *if and only if* $\Phi(\mathcal{S}) \models \varphi$.

3. Set graphs and MSO1

One among many ways to represent graphs as τ-structures is to take $\tau_1 = \{E\}$ where $\rho(E) = 2$, and to represent each graph $G = (V, E)$ with the τ_1-structure $G^* = \langle V, E^* \rangle$, where $E^* = \{(u, v) \in V^2 : \{u, v\} \in E\}$. The language MSO$_{\tau_1}$ is known as the Monadic Second Order Logic of graphs without quantification over sets of edges, and is denoted simply as MSO$_1$.

A graph property P is *MSO*$_\tau$*-expressible* if there is an MSO$_\tau$-sentence φ such that G has the property P if and only if $G^* \models \varphi$, for every graph G. In this section, we prove

Theorem 1. *The property of being a set graph is not MSO*$_1$*-expressible.*

To prove Theorem 1, we show that any MSO_1-sentence φ_{sg} that expresses being a set graph could be transformed into an MSO-formula φ_{qeq} expressing *quasi-equinumerosity*, i.e., $\mathcal{S}, a, A \models \varphi_{qeq}(X, Y)$ if and only if $||A(X)| - |A(Y)|| \leq 1$. Then, we show that such an MSO-formula φ_{qeq} could be used to define an MSO-formula φ_{eq} expressing *equinumerosity*, i.e., $\mathcal{S}, a, A \models \varphi_{eq}(X, Y)$ if and only if $|A(X)| = |A(Y)|$. However, it is widely known that equinumerosity is not MSO-expressible (cf. [Courcelle and Engelfriet 2012], Chapter 5).

3.1. From set graphs to quasi-equinumerosity

Suppose φ_{sg} is an MSO_1-sentence that expresses the property of being a set graph.

Let $\sigma = \{p_1, p_2\}$ be a vocabulary with $\rho(p_1) = \rho(p_2) = 1$. Let $\Phi = \langle \delta, \phi \rangle$ be the τ_1-σ-translation scheme where δ is $((p_1 x_1 \wedge \neg p_2 x_1) \vee (p_2 x_1 \wedge \neg p_1 x_1))$, and ϕ is $(p_1 x_1 \leftrightarrow p_2 x_2)$.

Given a σ-structure $\mathcal{S} = \langle D, P_1, P_2 \rangle$, if $P_1 \neq P_2$, the structure $\Phi(\mathcal{S}) = \langle D_\Phi, R \rangle$ represents the complete bipartite graph with bipartition $\{P_1 \setminus P_2, P_2 \setminus P_1\}$. However, if $P_1 = P_2$, then $D_\Phi = \emptyset$, so $\Phi(\mathcal{S})$ is not a well-defined τ_1-structure.

Given a σ-structure $\mathcal{S}$ such that $P_1 \neq P_2$, we have $\mathcal{S} \models \Phi(\varphi_{sg})$ if and only if the complete bipartite graph with bipartition $\{P_1 \setminus P_2, P_2 \setminus P_1\}$ is a set graph, which holds if and only if $||P_1 \setminus P_2| - |P_2 \setminus P_1|| \leq 1$ (cf. [Milanič and Tomescu 2013]). Moreover, note that $||P_1| - |P_2|| = ||P_1 \cap P_2| + |P_1 \setminus P_2| - (|P_2 \cap P_1| + |P_2 \setminus P_1|)| = ||P_1 \setminus P_2| - |P_2 \setminus P_1||$. Let $p_1 \equiv p_2$ be an abbreviation of $\forall x_1 (p_1 x_1 \leftrightarrow p_2 x_1)$. Hence, for every σ-structure $\mathcal{S}$, the sets P_1 and P_2 are quasi-equinumerous if and only if $\mathcal{S} \models (p_1 \equiv p_2 \vee \Phi(\varphi_{sg}))$.

Assuming, wlog, that X_1 and X_2 do not occur in $\Phi(\varphi_{sg})$, let us define φ_{qeq} by substituting in $(p_1 \equiv p_2 \vee \Phi(\varphi_{sg}))$ the constants p_1 and p_2 with the variables X_1 and X_2, respectively. Then, $\mathcal{S}, a, A \models \varphi_{qeq}(X_1, X_2)$ if and only if $||A(X_1)| - |A(X_2)|| \leq 1$.

3.2. From quasi-equinumerosity to equinumerosity

Suppose $\varphi_{qeq}(X_1, X_2)$ is an MSO-formula expressing quasi-equinumerosity.

Note that for all $n, m \in \mathbb{N}$, we have $n = m$ if and only if both inequalities $|(n-1) - m| \leq 1$ and $|n - (m-1)| \leq 1$ hold. So, given two non-empty finite sets A and B, we have $|A| = |B|$ if and only if $||A \setminus \{a\}| - |B|| \leq 1$ and $||A| - |B \setminus \{b\}|| \leq 1$ where $a \in A$ and $b \in B$.

For all $i, j, k \in \mathbb{N}$, let $X_i \equiv X_j \setminus \{x_k\}$ be an abbreviation of $\forall x_{k+1} (X_i x_{k+1} \leftrightarrow (X_j x_{k+1} \wedge \neg x_{k+1} = x_k))$; and let $X_i \equiv \emptyset$ be an abbreviation for $\forall x_1 \neg X_i x_1$. Now, we define the MSO-formula $\varphi_{eq}(X_1, X_2)$ as follows:

$$\begin{aligned}((X_1 \equiv \emptyset \wedge X_2 \equiv \emptyset) \vee (\exists x_1 (X_1 x_1 \wedge \forall X_3 (X_3 \equiv X_1 \setminus \{x_1\} \rightarrow \varphi_{qeq}(X_3, X_2))) \wedge \\ \exists x_1 (X_2 x_1 \wedge \forall X_3 (X_3 \equiv X_2 \setminus \{x_1\} \rightarrow \varphi_{qeq}(X_1, X_3)))))\end{aligned}$$

If φ_{qeq} expresses quasi-equinumerosity, φ_{eq} expresses equinumerosity. Since equinumerosity is not MSO-expressible, the property of "being a set graph" is not MSO_1-expressible.

Hence, we conclude that the usual path to prove that a property can be verified by a polynomial-time algorithm on graphs of bounded clique-width cannot be followed for the property of being a set graph.

References

Courcelle, B. (1990). Graph rewriting: An algebraic and logic approach. In van Leeuwen, J., editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, pages 193–242. Elsevier and MIT Press.

Courcelle, B. and Engelfriet, J. (2012). *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*, volume 138 of *Encyclopedia of mathematics and its applications*. Cambridge University Press.

Courcelle, B., Makowsky, J. A., and Rotics, U. (2000). Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150.

Milanič, M. and Tomescu, A. I. (2013). Set graphs. I. Hereditarily finite sets and extensional acyclic orientations. *Discrete Applied Mathematics*, 161(4-5):677–690.

Milanič, M., Rizzi, R., and Tomescu, A. I. (2014). Set graphs. II. Complexity of set graph recognition and similar problems. *Theoretical Computer Science*, 547:70–81.

Monteiro, B. B. (2022). On the complexity of the set graph recognition problem restricted to cographs and split graphs. Master's thesis, UFRJ.

Tomescu, A. I. (2011). *Sets as Graphs*. PhD thesis, Università degli Studi di Udine, Italy.