A modal perspective on monadic second-order alternation hierarchies

Antti Kuusisto

abstract. We establish that the quantifier alternation hierarchy of formulae of Second-Order Propositional Modal Logic (*SOPML*) induces an infinite corresponding semantic hierarchy over the class of finite directed graphs. This is a response to an open problem posed in [4] and [8]. We also provide modal characterizations of the expressive power of Monadic Second-Order Logic (*MSO*) and address a number of points that should promote the potential advantages of viewing *MSO* and its fragments from the modal perspective.

Keywords: monadic second-order logic, second-order propositional modal logic, finite model theory, alternation hierarchies

1 Introduction

In this paper we investigate the expressive power of Second-Order Propositional Modal Logic (*SOPML*), which is a modal logic extended with propositional quantifiers ranging over sets of possible worlds. Modal logics with propositional quantifiers have been investigated by a variety of researchers, see [7, 8, 9, 10, 11, 12, 13, 14, 22, 23] for example.

Johan van Benthem [4] and Balder ten Cate [8] raise the question whether the quantifier alternation hierarchy of *SOPML*-formulae induces an ascending corresponding hierarchy of definable classes of Kripke frames. This is an interesting question, especially as ten Cate shows in [8] that formulae of *SOPML* admit a prenex normal form representation. In this paper we prove that the semantic counterpart of the quantifier alternation hierarchy of *SOPML*-formulae is infinite over the class of finite directed graphs. This automatically implies that the semantic hierarchy is infinite over arbitrary Kripke frames.

Alternation hierarchies have received a lot of attention in finite model theory, see [16, 18, 19, 20, 21, 24] for example. As *SOPML* is a semantically natural fragment of *MSO* (see Theorem 6 in [8]), we feel that our result is also relatively interesting from the point of view of finite model theory.

Our main tool in answering the the question of van Benthem and ten Cate is a theorem of Schweikardt [21] which states that the alternation hierarchy of Monadic Second-Order Logic is strict over the class of grids. Inspired by the approach of Matz and Thomas in [19], we employ an approach based on *strong first-order reductions* in order to transfer the result of Schweikardt to a special class of finite directed graphs we define. Over this class the expressive power of *SOPML* coincides with that of *MSO*, whence we easily obtain the desired result that the alternation hierarchy of *SOPML* is infinite over finite directed graphs. The precise definition of strong first-order reductions (found in [18]) is of no particular importance for the present paper, as we give a virtually self-contained exposition of all our results.

As a by-product of our investigations we obtain a simple, effective procedure (inspired by the approach of ten Cate [8]) that translates *MSO*-sentences to equivalent formulae of Second-Order Propositional Modal Logic with Universal Modality (*SOPML*(E)). This implies that the expressive power of *SOPML*(E) on finite/arbitrary relational structures coincides with that of *MSO*, and a trivial adaptation of our argument shows that replacing universal modality E with difference modality D does not change the picture. Such modal perspectives on *MSO* could turn out interesting from the point of view of finite model theory.

The paper is structured as follows: In Section 2 we fix the notation and discuss a number of preliminary issues. In Section 3 we show that *MSO* = *SOPML*(E) with regard to expressive power. Using an approach analogous to that in Section 3, we then define in Section 4 a special class of directed graphs over which *MSO* and *SOPML* coincide in expressive power. In Section 5 we first work with *MSO*, transferring the result of Schweikardt to our special class of directed graphs. Then, using the connection created in Section 4, it is easy to establish that the *SOPML* alternation hierarchy is infinite over directed graphs.

2 Preliminary considerations

In this section we introduce technical notions that occupy a central role in the rest of the discourse.

2.1 Syntax and semantics

With a model we mean a model of predicate logic. We only consider models associated with a relational vocabulary. With a relational vocabulary we mean a vocabulary with relation symbols and constant symbols only.

We fix countable sets VAR_{FO} and VAR_{SO} of first-order and second-order variables, respectively. Naturally we assume that the sets are disjoint. We let $VAR = VAR_{FO} \cup VAR_{SO}$. We let lower-case symbols x, y, z denote first-order variables. Upper-case symbols X, Y, Z denote second-order variables. A union f of two functions $f_{FO} : VAR_{FO} \longrightarrow Dom(M)$ and $f_{SO} : VAR_{SO} \longrightarrow \mathcal{P}(Dom(M))$, where M is a model and $Dom(M)$ its domain, is called an *assignment*. Monadic Second-Order Logic is interpreted in terms of models and assignments in the usual way: We write $M, f \models \varphi$ when model M satisfies *MSO*-formula φ under assignment f.

Let *PROP* denote a countable set of *proposition variables*. We let symbols $p_x, p_y, p_z, p_X, p_Y, p_Z$ denote proposition variables. Let $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3$ be a relational vocabulary with set $\mathcal{S}_0$ of constant symbols and sets $\mathcal{S}_1$ and $\mathcal{S}_2$ of unary and binary relation symbols respectively; set $\mathcal{S}_3$ contains the relation symbols of higher arities. The language $L(\mathcal{S})$ of Second-Order Propositional Modal Logic associated with vocabulary $\mathcal{S}$ is determined by the following recursive definition:

$$\varphi ::= c_i \mid p_x \mid p_j \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond_k \varphi \mid \triangle_l(\varphi_1, ..., \varphi_{n-1}) \mid \exists p_x\, \varphi$$

such that $c_i \in \mathcal{S}_0$, $p_x \in PROP$, $P_j \in \mathcal{S}_1$, $R_k \in \mathcal{S}_2$, and $R_l \in \mathcal{S}_3$ is an n-ary relation symbol.

In order to interpret formulae of Second-Order Propositional Modal Logic, we need the notion of a *pointed model*:

DEFINITION 1. A pointed model is a pair (M, w), where M is a model and $w \in Dom(M)$.

We also need objects that interpret free occurrences of proposition variables $p_x \in PROP$: Any mapping $V : PROP \longrightarrow \mathcal{P}(Dom(M))$, where M is a model, is called a *valuation*.

Let $\mathcal{S}$ be a vocabulary and M an $\mathcal{S}$-model with $w \in Dom(M) = W$. Let V be a related valuation. We let $\Vdash$ denote the modal truth relation, which we define in the following way:

$(M, w), V \Vdash c_i$	$\Leftrightarrow$	$w = c_i^M$
$(M, w), V \Vdash p_j$	$\Leftrightarrow$	$w \in P_j^M$
$(M, w), V \Vdash p_x$	$\Leftrightarrow$	$w \in V(p_x)$
$(M, w), V \Vdash \neg\varphi$	$\Leftrightarrow$	$(M, w), V \nVdash \varphi$
$(M, w), V \Vdash \varphi \wedge \psi$	$\Leftrightarrow$	$(M, w), V \Vdash \varphi$ and $(M, w), V \Vdash \psi$
$(M, w), V \Vdash \exists p_x \varphi$	$\Leftrightarrow$	$\exists U \subseteq W((M, w), V[p_x \mapsto U] \Vdash \varphi)$
$(M, w), V \Vdash \Diamond_k\, \varphi$	$\Leftrightarrow$	$\exists u \in W(wR_k u$ and $(M, u) \Vdash \varphi)$
$(M, w), V \Vdash \triangle_l\, (\varphi_1, ..., \varphi_{n-1})$	$\Leftrightarrow$	$\exists u_1, ..., u_{n-1} \in W$ such that $R_l(w, u_1, ..., u_{n-1})$ and $\forall i < n((M, u_i), V \Vdash \varphi_i)$

If a formula φ does not contain free occurrences of proposition variables, we may drop valuation V and write $(M, w) \Vdash \varphi$. An *SOPML*-formula without free proposition variables is an *SOPML-sentence*. We extend the definition of relation $\Vdash$ to models in the following way:

$$M \Vdash \varphi \;\Leftrightarrow\; \text{for all } w \in W,\; (M, w) \Vdash \varphi$$

We also extend the truth relation of predicate logic to cover pointed models. We define

$$(M, w) \models \varphi(x) \;\Leftrightarrow\; M, [x \mapsto w] \models \varphi(x),$$

where $\varphi(x)$ is a formula with exactly one free variable, x.

Let H_p be a class of *pointed* models. We say that *SOPML*-sentence φ *defines class* C of pointed models *with respect to* H_p if $C = \{(M, w) \in H_p \mid (M, w) \Vdash \varphi\}$. We write $MOD_{H_p}(\varphi) = C$. Similarly, we say that *MSO*-formula $\psi(x)$ *defines class* C of pointed models *with respect to* H_p if $C = \{(M, w) \in H_p \mid (M, w) \models \psi(x)\}$. Formula $\psi(x)$ is required to contain exactly one free first-order variable and no free second-order variables. We write $MOD_{H_p}(\psi(x)) = C$.

Let H be a class of models. We say that *SOPML*-sentence φ *defines class* C of models *with respect to* H if $C = \{M \in H \mid M \Vdash \varphi\}$. This corresponds to the notion of global definability. We write $MOD_H(\varphi) = C$. Similarly, we say that *MSO*-sentence ψ *defines class* C of models *with respect to* H if $C = \{M \in H \mid M \models \psi\}$. We write $MOD_H(\psi) = C$.

When we informally leave out parentheses when writing formulae, the order of preference of logical connectives is such that unary connectives have the highest priority and then come $\wedge, \vee, \rightarrow, \leftrightarrow$ in the given order.

When a subindex of a symbol ($c_i, p_i, R_i, \Diamond_i$ etc.) is irrelevant or understood from the context, we may leave it unwritten.

2.2 Grids and graphs

Two classes of structures have a central role in the considerations that follow:

DEFINITION 2. Let $m, n \in \mathbb{N}_{\geq 1}$ and let $D^n_m = \{1, 2, ..., m\} \times \{1, 2, ..., n\}$. Define binary relations S^{Gr}_1 and S^{Gr}_2 such that S^{Gr}_1 contains exactly the pairs of type $((i, j), (i + 1, j)) \in D^n_m \times D^n_m$ and S^{Gr}_2 exactly the pairs of type $((i, j), (i, j + 1)) \in D^n_m \times D^n_m$. A structure $Gr = (D^n_m, S^{Gr}_1, S^{Gr}_2)$, where $m, n \in \mathbb{N}_{\geq 1}$, is called a *grid*. Grid $Gr = (D^n_m, S^{Gr}_1, S^{Gr}_2)$ is said to *correspond to* an $m \times n$-matrix. Element $(1, 1)$ is referred to as the *top left element*. We let $GRID$ denote the class of grids. Note that this class is not closed under isomorphism.

The other class of structures we shall consider is that of (non-empty) *directed graphs*. We define a directed graph to be a structure of type (W, R), where $W \neq \emptyset$ is a finite set and $R \subseteq W \times W$ a binary relation. When we refer to a graph we always mean a finite directed graph. We let $GRAPH$ denote the class of finite directed graphs.

2.3 Alternation hierarchies

Intuitively, the levels of the *monadic second-order quantifier alternation hierarchy* measure the number of alternations of existential and universal second-order quantifiers of *MSO*-formulae in prenex normal form. (An *MSO*-formula in prenex normal form consists of a vector of second-order quantifiers, followed by a first order part.) It is natural to classify *SOPML*-formulae in an analogous way.

Below, we give formal definitions of alternation hierarchies. We only define levels containing formulae that begin with an existential quantifier, as this suffices for the purposes of this article.

Let $L_{FO}(\mathcal{S} \cup VAR_{SO})$ denote the first-order language associated with relational vocabulary $\mathcal{S} \cup VAR_{SO}$. We define $\Sigma_0(\mathcal{S}) = L_{FO}(\mathcal{S} \cup VAR_{SO})$ and let

$$\Sigma_{n+1}(\mathcal{S}) = \{\exists X_1, ..., \exists X_k \neg\varphi \mid k \in \mathbb{N} \text{ and } \varphi \in \Sigma_n(\mathcal{S})\}.$$

Sets $\Sigma_n(\mathcal{S})$ are levels of the syntactic alternation hierarchy of *MSO*.

We write Σ_n instead of $\Sigma_n(\mathcal{S})$ when the vocabulary is clear from the context. With $[\Sigma_n]$ we refer to the equivalence closure of Σ_n. In other words, $\varphi \in [\Sigma_n]$ iff φ is equivalent to some formula $\varphi' \in \Sigma_n$.

Levels of the syntactic alternation hierarchy are associated with natural semantic counterparts: Let H be a subclass of $\mathcal{S}$-structures. We define

$$\underline{\Sigma_n}(H) = \{C \in \mathcal{P}(H) \mid MOD_H(\varphi) = C \text{ for some sentence } \varphi \in \Sigma_n(\mathcal{S})\}.$$

Similarly, we let

$$\underline{\Sigma_n}(H_p) \\ = \{C \in \mathcal{P}(H_p) \mid MOD_{H_p}(\varphi(x)) = C \text{ for some formula } \varphi(x) \in \Sigma_n(\mathcal{S})\},$$

where H_p is a class of pointed $\mathcal{S}$-models.

We then deal with the quantifier alternation hierarchies of *SOPML*. The zeroeth level of the syntactic hierarchy of *SOPML* contains all quantifier free *SOPML*-formulae, and any formula $\exists p_{x_1}, ..., \exists p_{x_k} \neg\varphi$ belongs to level $n+1$ iff φ belongs to the n-th level. We let $\Sigma_n^{ML}(\mathcal{S})$ denote the n-th level of this hierarchy. On the semantic side we define

$$\underline{\Sigma_n^{ML}}(H) \\ = \{C \in \mathcal{P}(H) \mid MOD_H(\varphi) = C \text{ for some sentence } \varphi \in \Sigma_n^{ML}(\mathcal{S})\},$$

where H is a subclass of the class of $\mathcal{S}$-models. Similarly, we define

$$\underline{\Sigma_n^{ML}}(H_p) \\ = \{C \in \mathcal{P}(H_p) \mid MOD_{H_p}(\varphi) = C \text{ for some sentence } \varphi \in \Sigma_n^{ML}(\mathcal{S})\},$$

where H_p is a class of pointed $\mathcal{S}$-models.

If for all $n \in \mathbb{N}$ there exists a $k \in \mathbb{N}$ such that $\underline{\Sigma_n}(K) \neq \underline{\Sigma_k}(K)$, we say that *the alternation hierarchy of MSO is infinite on* K. We define infinity of *SOPML* alternation hierarchies analogously.

3 SOPML(E) = MSO

In this section we show that Second-Order Propositional Modal Logic with Universal Modality (*SOPML*(E)) has the same expressive power as *MSO*. This result is closely related to the fact that hybrid logic $\mathcal{H}(\downarrow, E)$ is expressively complete for first-order logic (see [3] and the references therein). In fact, in the light of the results in [1, 2, 8], the result is not surprising.

In order to establish that *SOPML*(E) is expressively complete for *MSO*, we define a simple translation from the set of *MSO*-formulae to the set of *SOPML*(E)-formulae. The translation was inspired by a very similar translation defined in [8].

We begin with a formal definition of logic *SOPML*(E) (cf. *SOEPDL* in [22]): Let $\mathcal{S}$ be a relational vocabulary and let $L(\mathcal{S})$ denote the related second-order propositional modal language. We extend language $L(\mathcal{S})$ to a new language $L^E(S)$ in the following way:

$$\begin{array}{lcl} \varphi \in L(S) & \Rightarrow & \varphi \in L^E(S) \\ \varphi \in L^E(S) & \Rightarrow & \langle E\rangle\varphi \in L^E(S) \end{array}$$

The truth definition of *SOPML* is extended by setting

$$(M,w),V \Vdash \langle E\rangle\varphi \;\Leftrightarrow\; \exists u \in Dom(M)((M,u),V \Vdash \varphi).$$

Next we prepare ourselves for an important auxiliary result (Lemma 3), which we then prove.

Let M be a model and $f : VAR \longrightarrow Dom(M) \cup \mathcal{P}(Dom(M))$ a related assignment. Define $PROP = \{p_x \mid x \in VAR_{FO}\} \cup \{p_X \mid X \in VAR_{SO}\}$. We let V_f denote the valuation mapping from $PROP$ to $\mathcal{P}(Dom(M))$ such that $V_f(p_x) = \{f(x)\}$ and $V_f(p_X) = f(X)$ for all $p_x, p_X \in PROP$.

Consider the following formula:

$$uniq(p_x) = \langle E\rangle p_x \wedge \forall p_y(\langle E\rangle(p_y \wedge p_x) \rightarrow [E](p_x \rightarrow p_y)),$$

where $[E]$ stands for $\neg\langle E\rangle\neg$. The formula states that proposition p_x is satisfied by exactly one point of the model.

We define the following translation TR from the set of *MSO*-formulae to the set of *SOPML*(E)-formulae:

$$\begin{array}{lcl}
TR(P(y)) & = & \langle E\rangle(p \wedge p_y) \\
TR(Y(z)) & = & \langle E\rangle(p_Y \wedge p_z) \\
TR(R_i(y,z)) & = & \langle E\rangle(p_y \wedge \Diamond_i p_z) \\
TR(R_j(x_1,...,x_n)) & = & \langle E\rangle(p_{x_1} \wedge \triangle_j(p_{x_2},...,p_{x_n})) \\
TR(y = z) & = & \langle E\rangle(p_y \wedge p_z) \\
TR(c = y) & = & \langle E\rangle(c \wedge p_y) \\
TR(y = c) & = & \langle E\rangle(p_y \wedge c) \\
TR(c_{i_1} = c_{i_2}) & = & \langle E\rangle(c_{i_1} \wedge c_{i_2}) \\
TR(\neg\psi) & = & \neg TR(\psi) \\
TR(\psi \wedge \varphi) & = & TR(\psi) \wedge TR(\varphi) \\
TR(\exists z(\psi)) & = & \exists p_z(uniq(p_z) \wedge TR(\psi)) \\
TR(\exists Z(\psi)) & = & \exists p_Z(TR(\psi))
\end{array}$$

LEMMA 3. *For all MSO-formulae* φ*,*

$$M, f[x \mapsto w] \models \varphi \quad \Leftrightarrow \quad (M,w), V_f[p_x \mapsto \{w\}] \Vdash TR(\varphi)$$

for all models $M = (W,R)$*, all points* $w \in W$ *and all assignments* $f : VAR \longrightarrow W \cup \mathcal{P}(W)$*.*

Proof. We prove the claim by induction on the structure of formula φ. The basis of the induction is established by a straightforward argument. The case where $\varphi = \neg\psi$ for some formula ψ is trivial, as is the case where φ has a conjunction as its main connective. Therefore we may proceed directly to the case where $\varphi = \exists z(\psi)$.

Assume first that $M, f[x \mapsto w] \models \exists z(\psi)$ (we assume w.l.o.g. that $z \neq x$). Thus $M, f[z \mapsto u, x \mapsto w] \models \psi$ for some $u \in W$. Therefore $(M,w), V_f[p_z \mapsto \{u\}, p_x \mapsto \{w\}] \Vdash TR(\psi)$ by the induction hypothesis. Thus $(M,w), V_f[p_x \mapsto \{w\}] \Vdash \exists p_z(uniq(p_z) \wedge TR(\psi))$, as required.

Assume then that $(M, w), V_f[p_x \mapsto \{w\}] \Vdash \exists p_z(uniq(p_z) \wedge TR(\psi))$. Therefore $(M, w), V_f[p_z \mapsto U, p_x \mapsto \{w\}] \Vdash uniq(p_z) \wedge TR(\psi)$ for some set $U \subseteq W$. As $(M, w), V_f[p_z \mapsto U, p_x \mapsto \{w\}] \Vdash uniq(p_z)$, we have $U = \{u\}$ for some $u \in W$. Therefore $(M, w), V_f[p_z \mapsto \{u\}, p_x \mapsto \{w\}] \Vdash TR(\psi)$. Thus $M, f[z \mapsto u, x \mapsto w] \models \psi$ by the induction hypothesis, and therefore $M, f[x \mapsto w] \models \exists z(\psi)$, as required.

Finally, the argument for the case where formula φ is of type $\exists Z(\psi)$, for some formula ψ, is straightforward. ■

We are now ready for the main results of this section:

THEOREM 4. *A subclass K of a class C of pointed models is definable w.r.t. C by an MSO-formula if and only if K is definable w.r.t. C by an SOPML(E)-sentence.*

Proof. Let $\varphi(x)$ be an arbitrary *MSO*-formula with exactly one free variable. Let $M = (W, R, ...)$ be an arbitrary model and $f : VAR \longrightarrow W \cup \mathcal{P}(W)$ an arbitrary assignment. We have the following equivalence by Lemma 3:

$$M, f[x \mapsto w] \models \varphi \quad \Leftrightarrow \quad (M, w), V_f[p_x \mapsto \{w\}] \Vdash TR(\varphi)$$

We also have the following equivalence:

$$\begin{array}{ll} & (M, w), V_f[p_x \mapsto \{w\}] \Vdash TR(\varphi) \\ \Leftrightarrow & \\ & (M, w) \Vdash \exists p_x(p_x \wedge uniq(p_x) \wedge TR(\varphi)) \end{array}$$

By the two equivalences, it is clear that $\exists p_x(p_x \wedge uniq(p_x) \wedge TR(\varphi))$ is the desired *SOPML*(E)-sentence equivalent to φ.

For the converse, if φ is an *SOPML*(E)-sentence, the desired *MSO*-formula is $St_x(\varphi)$, where St_x denotes the required trivial generalization of the standard translation operator (see [5] for the definition of standard translation). ■

THEOREM 5. *A subclass K of a class C of models is definable w.r.t. C by an MSO-sentence if and only if K is definable w.r.t. C by an SOPML(E)-sentence.*

Proof. Let φ be an arbitrary *MSO*-sentence. Notice that $TR(\varphi)$ does not contain any free proposition variables. We have the following equivalences:

$$\begin{array}{lll} M \models \varphi & \Leftrightarrow & M, f[x \mapsto w] \models \varphi \text{ for all } w \in W \\ & \Leftrightarrow & (M, w), V_f[p_x \mapsto \{w\}] \Vdash TR(\varphi) \text{ for all } w \in W \\ & \Leftrightarrow & (M, w) \Vdash TR(\varphi) \text{ for all } w \in W \\ & \Leftrightarrow & M \Vdash TR(\varphi) \end{array}$$

where the second equivalence follows from Lemma 3.

For the converse, $\forall x St_x(\psi)$ is the desired *MSO*-sentence equivalent to *SOPML*(E)-sentence ψ. ■

A trivial adaptation of the approach in this section leads to the realization that with regard to expressive power, *SOPML*(D) $=$ *MSO*, where D is the difference modality.

4 Simulating globality

The local nature of *SOPML* (cf. Proposition 4 of [8]) limits its expressive power. In this section we define a class of structures over which this is not the case. The key point is to insist that each structure contains a point which connects to every point of the structure:

DEFINITION 6. Let $S = (W, R, ...)$ be a structure with a binary relation R. Assume there is a point $w \in W$ such that wRu for all $u \in W$. We call such a point u a *localizer*. Structures with a localizer are called *localized*. If (M, w) is a pointed model where w is a localizer, we say that (M, w) is *l-pointed*.

The notions of a localizer and a localized model resemble the notions of a *spypoint* and a *spypoint model* applied in the hybrid logic literature (see [2, 6]).

We then prepare ourselves for the next result (Lemma 7) by defining local analogues of formula $uniq(p_x)$ and translation TR defined in Section 3.

Let $uniq'(p_x)$ be the following formula:

$$\Diamond p_x \wedge \forall p_y(\Diamond(p_y \wedge p_x) \rightarrow \Box(p_x \rightarrow p_y)),$$

where $\Box$ stands for $\neg\Diamond\neg$. It is easy to see that if (W, R) is a directed graph with a localizer $w \in W$, then $((W, R), w), [p_x \mapsto U] \Vdash uniq'(p_x)$ if and only if $U = \{u\}$ for some $u \in W$.

Consider translation TR defined in Section 3. Replace the occurrences of the universal diamond $\langle E \rangle$ by $\Diamond$, and also replace $uniq(p_z)$ by $uniq'(p_z)$. Denote this new translation by LTR.

The following lemma is a local analogue of Lemma 3:

LEMMA 7. *For all MSO-formulae* φ*,*

$$M, f[x \mapsto w] \models \varphi \quad \Leftrightarrow \quad (M, w), V_f[p_x \mapsto \{w\}] \Vdash LTR(\varphi)$$

for all localized models $M = (W, R)$*, all localizers* $w \in W$ *and all assignments* $f : VAR \longrightarrow W \cup \mathcal{P}(W)$*.*

Proof. The proof is essentially the same as that of Lemma 3. ∎

The following lemma is a local analogue of Theorem 4:

LEMMA 8. *Let C be a class of l-pointed models. A class $K \subseteq C$ of l-pointed models is definable w.r.t. C by an MSO-formula if and only if K is definable w.r.t. C by an SOPML-sentence.*

Proof. Let *MSO*-formula φ define K w.r.t. C. Formula $\exists p_x(p_x \wedge uniq'(p_x) \wedge LTR(\varphi))$ is the desired *SOPML*-sentence equivalent to φ. The proof is essentially the same as that of Theorem 4. Instead of using Lemma 3, however, we apply the analogous lemma that fits the framework without universal modality, i.e., Lemma 7. ∎

Let C be a class of localized models. Let φ be an *SOPML*-sentence such that for each model $M \in C$ there exists at least one point $w \in Dom(M)$ that satisfies φ, and moreover, every point w that satisfies φ is a localizer. We say that φ *fixes localizers on* C.

The following lemma is a local analogue of Theorem 5:

LEMMA 9. *Let C be a class of localized models and assume there exists some SOPML-sentence φ that fixes localizers on C. A class $K \subseteq C$ of localized models is definable w.r.t C by an MSO-sentence if and only if K is definable w.r.t. C by an SOPML-sentence.*

Proof. Let ψ be an *MSO*-sentence that defines K w.r.t. C. Let $M \in C$ and let $U \subseteq Dom(M)$ be the set of points $w \in Dom(M)$ such that $(M, w) \Vdash \varphi$. We have the following equivalences:

$$\begin{aligned} M \models \psi \quad &\Leftrightarrow \quad \forall w \in U(M, f[x \mapsto w] \models \psi) \\ &\Leftrightarrow \quad \forall w \in U((M, w), V_f[p_x \mapsto \{w\}] \Vdash LTR(\psi)) \\ &\Leftrightarrow \quad \forall w \in U((M, w) \Vdash LTR(\psi)) \\ &\Leftrightarrow \quad \forall w \in U((M, w) \Vdash \varphi \rightarrow LTR(\psi)) \\ &\Leftrightarrow \quad M \Vdash \varphi \rightarrow LTR(\psi), \end{aligned}$$

where the second equivalence follows from Lemma 7.

For the converse, $\forall x St_x(\pi)$ is the desired *MSO*-sentence equivalent to *SOPML*-sentence π. ■

5 Alternation hierarchy of SOPML is infinite

In this section we prove that the *SOPML* alternation hierarchy is infinite over the class of finite directed graphs. The following theorem from [21] is the most important tool we shall use in the elaborations below:

THEOREM 10. *For all $n \in \mathbb{N}_{\geq 1}$ we have $\underline{\Sigma_n}(GRID) \neq \underline{\Sigma_{n+1}}(GRID)$.*

While a similar result holds for directed graphs[1], on *words* and *labelled trees*, for example, the alternation hierarchy of *MSO* is known to collapse to level Σ_1 (see [17] for a recent survey of related results). This explains why we use grids in the elaborations below.

In Subsection 5.1 we show how to encode grids as *localized grid graphs* (see Definition 11). In Subsection 5.2 we then transfer the result of Theorem 10 to localized grid graphs (Proposition 16) and l-pointed localized grid graphs (Proposition 17). The transferred results will be needed in Subsection 5.3, where we show that the alternation hierarchy of *SOPML* is infinite over pointed directed graphs (Theorem 18) and ordinary directed graphs (Theorem 19).

[1]See [18, 19]. In [18], the result for directed graphs is established via a reduction from the class of grids to a certain subclass of directed graphs. Let us call this class C. While we could prove Proposition 16 via a reduction from class C, we instead prove it via a direct reduction from the class of grids. The two alternative approaches are very similar, but the approach via a direct reduction from the class of grids has presentational advantages over the approach via a reduction from class C.

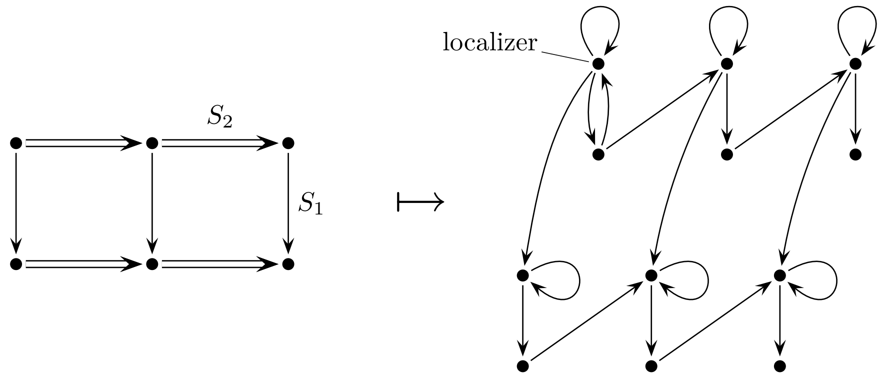

Figure 1. The figure shows a grid and its encoding. The localizer connects to each point of the graph; for the sake of clarity, most arrows originating from the localizer have not been drawn.

5.1 Encoding grids as localized grid graphs

In this subsection we define a map that encodes grids as localized directed graphs.

DEFINITION 11. Mapping $\alpha : GRID \longrightarrow GRAPH$ transforms grid Gr to a directed graph (W, R) such that $W = (Dom(Gr) \times \{0\}) \cup (Dom(Gr) \times \{1\})$ and

$$\begin{array}{rcl} R & = & \{\ ((a,0),(a,0)) \mid a \in Dom(Gr)\} \\ & \cup & \{\ ((a,0),(a,1)) \mid a \in Dom(Gr)\} \\ & \cup & \{\ ((a,0),(b,0)) \mid (a,b) \in S_1^{Gr}\} \\ & \cup & \{\ ((a,1),(b,0)) \mid (a,b) \in S_2^{Gr}\} \\ & \cup & \{\ ((t,0),(a,i)) \mid a \in Dom(Gr),\ i \in \{0,1\}\ \} \\ & \cup & \{\ ((t,1),(t,0))\ \}, \end{array}$$

where $t = (1,1)$ is the top left element of grid Gr. We call structures in the isomorphism closure of $\alpha(GRID)$ *localized grid graphs*. We let LGG denote this class of structures. We let LGG_p denote the corresponding class of l-pointed grid graphs. See Figure 1 for an example of a grid and the corresponding localized grid graph.

Point $(t,0)$ sees every point in graph $\alpha(Gr)$, i.e., it is a localizer. This property enables us to overcome difficulties resulting from the local nature of *SOPML*. We define the following formula:

$$\psi_{t_0}(x) = xRx \wedge \exists y(xRy \wedge yRx \wedge x \neq y)$$

The formula asserts that $x = (t,0)$. Insisting that $(t,1)R(t,0)$ will help us with a number of technical issues, such as defining formula

$$\psi_{t_1}(x) = \neg xRx \wedge \exists y(xRy \wedge yRx),$$

which asserts that $x = (t,1)$.

We then show that encoding $\alpha : GRID \longrightarrow GRAPH$ is injective:

LEMMA 12. *Encoding $\alpha : GRID \longrightarrow GRAPH$ is injective in the following sense: If $\alpha(Gr)$ and $\alpha(Gr')$ are isomorphic, then $Gr = Gr'$.*

Proof. Let $\alpha(Gr) = (W, R) = G$ and $\alpha(Gr') = (W', R') = G'$ for some grids Gr and Gr'. Assume $f : W \longrightarrow W'$ is an isomorphism between the graphs. Let k be the number of elements $w \in W$ with a reflexive loop. It is clear that Gr corresponds to an $m \times n$-matrix such that $m \cdot n = k$ (cf. Definition 2). The number of points $w' \in W'$ with a reflexive loop must also be k, as the two graphs are isomorphic. Thus grid Gr' corresponds to an $m' \times n'$-matrix such that $m' \cdot n' = k$. To conclude the proof it suffices to show that $n = n'$.

We shall show that for each $i \in \mathbb{N}_{\geq 1}$ there is a first-order formula φ_i such that for all $M \in GRID$ we have $\alpha(M) \models \varphi_i$ iff M corresponds to a $j \times i$-matrix for some j. The claim of the lemma follows from this: As $G \cong G'$, they satisfy the same first-order sentences. Thus there is some i such that both graphs G and G' satisfy sentence φ_i. Thus $n = i = n'$.

We then show how to define formulae φ_i. We deal with the case where $i = 1$ separately: We let $\varphi_1 = \exists x(\psi_{t_1}(x) \wedge \exists^{=1} y(xRy))$, where $\exists^{=1} y$ stands for "there exists exactly one y". We then consider the cases where $i \geq 2$. We first define the following formulae:

$$\begin{array}{lcl} \pi_2(x) & = & \exists y \exists z(\psi_{t_1}(y) \wedge yRz \wedge \neg zRy \wedge zRx \wedge \neg xRx) \\ succ(x,y) & = & \exists z(xRz \wedge zRy \wedge \neg yRy) \end{array}$$

We then define φ_i (where $i \geq 2$) in the following way:

$$\varphi_i = \exists x_2, ..., x_i \Big(\pi_2(x_2) \wedge \big(\bigwedge_{2 \leq r < i} succ(x_r, x_{r+1})\big) \wedge \neg \exists y(x_i R y)\Big)$$

It is relatively easy to see that formulae φ_i have the desired meaning. ■

5.2 MSO alternation hierarchy over localized grid graphs

In this subsection we show that results analogous to Theorem 10 hold for localized grid graphs (Proposition 16) and l-pointed grid graphs (Proposition 17).

We begin by showing how to transform any grid-formula $\varphi_1 \in \Sigma_n$ into a graph-formula $\varphi_2 \in \Sigma_n$ that says the same about localized grid graphs as φ_1 says about grids:

LEMMA 13. *For every grid-formula φ_1 there exists a graph-formula φ_2 such that for all grids Gr and all assignments $f : VAR \rightarrow Dom(Gr) \cup \mathcal{P}(Dom(Gr))$,*

$$Gr, f \models \varphi_1 \;\Leftrightarrow\; \alpha(Gr), f' \models \varphi_2,$$

where valuation f' is defined such that for all $x, X \in VAR$, all $a \in Dom(Gr)$ and all $A \subseteq Dom(Gr)$ we have $f'(x) = (a, 0) \;\Leftrightarrow f(x) = a$ and $f'(X) = A \times \{0\} \;\Leftrightarrow f(X) = A$. Furthermore, for all $n \in \mathbb{N}$, if $\varphi_1 \in \Sigma_n$, then $\varphi_2 \in \Sigma_n$.

Proof. We begin by showing how to define φ_2 in the case where φ_1 is atomic. If φ_1 is of type $x = y$ or type $X(y)$, we let $\varphi_2 = \varphi_1$. If φ_1 is of type xS_1y, we let φ_2 be the following formula:

$$\begin{array}{ll} & \psi_{t_0}(x) \wedge \psi_{t_0}(y) \rightarrow \bot \\ \wedge & \psi_{t_0}(x) \wedge \neg\psi_{t_0}(y) \rightarrow \forall z(zRy \rightarrow (\psi_{t_0}(z) \vee z = y)) \\ \wedge & \neg\psi_{t_0}(x) \wedge \psi_{t_0}(y) \rightarrow \bot \\ \wedge & \neg\psi_{t_0}(x) \wedge \neg\psi_{t_0}(y) \rightarrow xRy \wedge x \neq y \end{array}$$

If φ_1 is of type xS_2y, we define φ_2 to be the following formula:

$$\begin{array}{ll} & \psi_{t_0}(x) \wedge \psi_{t_0}(y) \rightarrow \bot \\ \wedge & \psi_{t_0}(x) \wedge \neg\psi_{t_0}(y) \rightarrow \exists u(\psi_{t_1}(u) \wedge uRy) \\ \wedge & \neg\psi_{t_0}(x) \wedge \psi_{t_0}(y) \rightarrow \bot \\ \wedge & \neg\psi_{t_0}(x) \wedge \neg\psi_{t_0}(y) \rightarrow \exists z(xRz \wedge \neg zRz \wedge zRy) \end{array}$$

For the sake of induction, assume $\varphi_1 = \neg\pi_1$. By the induction hypothesis there exists a graph-formula π_2 such that $Gr, f \models \pi_1 \;\Leftrightarrow\; \alpha(Gr), f' \models \pi_2$ for all grids Gr and related assignments f. Let $\varphi_2 = \neg\pi_2$. Similarly, in the case where $\varphi_1 = \pi_1 \wedge \pi_1'$, let $\varphi_2 = \pi_2 \wedge \pi_2'$, where graph-formulae π_2, π_2' are again chosen by the induction hypothesis. In the case where $\varphi_1 = \exists x(\pi_1)$, let $\varphi_2 = \exists x(xRx \wedge \pi_2)$. Finally, in the case $\varphi_1 = \exists X(\pi_1)$, let $\varphi_2 = \exists X(\forall x(X(x) \rightarrow xRx) \wedge \pi_2)$. ■

Our next aim is to show that for each graph-sentence $\varphi_2 \in \Sigma_n$, there exists a grid-sentence $\varphi_1 \in \Sigma_n$ that says the same about grids as φ_2 says about localized grid graphs. In order to establish this, we first need to address a number of technical issues.

We define a new set of symbols $VAR' = VAR_{FO} \cup (VAR_{SO} \times \{0\}) \cup (VAR_{SO} \times \{1\}) \cup (VAR_{SO} \times \{t_0\}) \cup (VAR_{SO} \times \{t_1\})$. Naturally we choose our symbols such that the above five sets making up VAR' are disjoint. We associate each first-order variable with an index such that $VAR_{FO} = \{x_1, x_2, ...\}$. We denote the new second-order variables of type $(X, 0)$, $(X, 1)$, (X, t_0) and (X, t_1) by X^0, X^1, X^{t_0} and X^{t_1} respectively.

Let Gr be a grid. We partition the domain of grid graph $\alpha(Gr)$ into four sets:

$$\begin{array}{lll} V_{t_0} & = & \{\ ((1,1),0)\ \} \\ V_{t_1} & = & \{\ ((1,1),1)\ \} \\ V_0 & = & \{\ ((x,y),0) \in Dom(\alpha(Gr)) \mid (x,y) \neq (1,1)\} \\ V_1 & = & \{\ ((x,y),1) \in Dom(\alpha(Gr)) \mid (x,y) \neq (1,1)\} \end{array}$$

Now let $\kappa : \mathbb{N}_{\geq 1} \longrightarrow \{0, 1, t_0, t_1\}$ be a function. We say that assignment $f : VAR \longrightarrow Dom(\alpha(Gr)) \cup \mathcal{P}(Dom(\alpha(Gr)))$ is *of type* κ if $f(x_i) \in V_{\kappa(i)}$ for all $i \in \mathbb{N}_{\geq 1}$. We call function κ an *assignment type*.

Each assignment $f : VAR \longrightarrow Dom(\alpha(Gr)) \cup \mathcal{P}(Dom(\alpha(Gr)))$ is associated with a related assignment $f_{Gr} : VAR' \longrightarrow Dom(Gr) \cup \mathcal{P}(Dom(Gr))$ defined in the following way:

$$\forall a \in Dom(Gr)\Big(f_{Gr}(x) = a \;\Leftrightarrow\; \big(f(x) = (a,0) \text{ or } f(x) = (a,1)\big)\Big)$$

for first-order variables $x \in VAR'$. For second-order variables $X^0, X^1 \in VAR'$ we let

$$\begin{array}{rcl} f_{Gr}(X^0) & = & \{a \in Dom(Gr) \mid (a,0) \in f(X)\} \setminus \{t\} \\ f_{Gr}(X^1) & = & \{a \in Dom(Gr) \mid (a,1) \in f(X)\} \setminus \{t\}, \end{array}$$

where $t = (1,1)$ is the top left element of grid Gr. For second-order variables X^{t_i}, where $i \in \{0,1\}$, we let

$$f_{Gr}(X^{t_i}) = \begin{cases} \{t\} & \text{if } (t,i) \in f(X), \text{ where } t = (1,1) \in Dom(Gr) \\ \emptyset & \text{otherwise} \end{cases}$$

We are now ready for the following lemma:

LEMMA 14. *For every graph-formula φ_2 and every assignment type κ there exists a grid-formula φ_1^κ such that for all grid graphs $\alpha(Gr)$ and assignments $f : VAR \to Dom(\alpha(Gr)) \cup \mathcal{P}(Dom(\alpha(Gr)))$ of type κ,*

$$Gr, f_{Gr} \models \varphi_1^\kappa \;\Leftrightarrow\; \alpha(Gr), f \models \varphi_2.$$

Furthermore, for all $n \in \mathbb{N}$, if $\varphi_2 \in \Sigma_n$, then also $\varphi_1^\kappa \in \Sigma_n$.

Proof. First assume that φ_2 is atomic. If φ_2 is $x_i = x_j$, then we let

$$\varphi_1^\kappa = \begin{cases} x_i = x_j & \text{when } \kappa(i) = \kappa(j) \\ \bot & \text{when } \kappa(i) \neq \kappa(j) \end{cases}$$

If $\varphi_2 = x_i R x_j$, we define φ_1^κ according to the following table:

$(\kappa(i),\kappa(j))$	φ_1^κ	$(\kappa(i),\kappa(j))$	φ_1^κ
$(0,0)$	$x_i = x_j \vee x_i S_1 x_j$	$(0,t_0)$	$\bot$
$(0,1)$	$x_i = x_j$	$(t_0,0)$	$\top$
$(1,0)$	$x_i S_2 x_j$	$(0,t_1)$	$\bot$
$(1,1)$	$\bot$	$(t_1,0)$	$\exists z(topleft(z) \wedge z S_2 x_j)$
$(\kappa(i),\kappa(j))$	φ_1^κ	$(\kappa(i),\kappa(j))$	φ_1^κ
$(1,t_0)$	$\bot$	(t_0,t_0)	$\top$
$(t_0,1)$	$\top$	(t_0,t_1)	$\top$
$(1,t_1)$	$\bot$	(t_1,t_0)	$\top$
$(t_1,1)$	$\bot$	(t_1,t_1)	$\bot$

where $topleft(z)$ denotes formula $\neg\exists x(x S_1 z \vee x S_2 z)$. Finally, if $\varphi_2 = X(x_i)$, we let $\varphi_1^\kappa = X^{\kappa(i)}(x_i)$. We now have a basis for an argument by induction.

If $\varphi_2 = \neg\pi_2$, we use π_2 and the induction hypothesis to find π_1^κ. We then let $\varphi_1^\kappa = \neg\pi_1^\kappa$. Similarly, if $\varphi_2 = \pi_2 \wedge \chi_2$, we use the induction hypothesis to find π_1^κ and χ_1^κ and let $\varphi_1^\kappa = \pi_1^\kappa \wedge \chi_1^\kappa$.

In the case where $\varphi_2 = \exists x(\pi_2)$ we apply the induction hypothesis to π_2 in order to find formulae $\pi_1^{\kappa[x\mapsto i]}$, where $i \in \{0,1,t_0,t_1\}$, such that

$$Gr, f_{Gr} \models \pi_1^{\kappa[x\mapsto i]} \;\Leftrightarrow\; \alpha(Gr), f \models \pi_2$$

holds for all grid graphs $\alpha(Gr)$ and valuations f of type $\kappa[x \mapsto i]$. We then use these four formulae and define φ_1^κ to be the following formula:

$$\begin{array}{lrl} \exists x & \Big(topleft(x) & \wedge\ \pi_1^{\kappa[x\mapsto t_0]} \\ \vee & topleft(x) & \wedge\ \pi_1^{\kappa[x\mapsto t_1]} \\ \vee & \neg topleft(x) & \wedge\ \pi_1^{\kappa[x\mapsto 0]} \\ \vee & \neg topleft(x) & \wedge\ \pi_1^{\kappa[x\mapsto 1]} \Big) \end{array}$$

Finally, if $\varphi_2 = \exists X(\pi_2)$, we find a grid formula π_1^κ corresponding to π_2 by the induction hypothesis and set $\varphi_1^\kappa = \exists X^0 \exists X^1 \exists X^{t_0} \exists X^{t_1}(\chi \wedge \pi_1^\kappa)$, where χ is the conjunction of formulae $\forall x(X^0(x) \vee X^1(x) \rightarrow \neg topleft(x))$ and $\forall x(X^{t_0}(x) \vee X^{t_1}(x) \rightarrow topleft(x))$. ■

COROLLARY 15. *For every graph-sentence φ_2 there exists a grid-sentence φ_1 such that for all grid graphs $\alpha(Gr)$,*

$$Gr \models \varphi_1 \ \Leftrightarrow\ \alpha(Gr) \models \varphi_2.$$

Sentence φ_1 can be chosen such that it is on the same level of the second-order quantifier alternation hierarchy as φ_2.

Proof. Choose an arbitrary κ and apply Lemma 14. ■

The next two propositions will be needed later on, but they are also interesting in their own right as they characterize the *MSO* alternation hierarchy with respect to *localized* graphs.

PROPOSITION 16. *For all $n \in \mathbb{N}_{\geq 1}$ we have $\underline{\Sigma_n}(LGG) \neq \underline{\Sigma_{n+1}}(LGG)$.*

Proof. Fix an arbitrary positive integer n. By Theorem 10 there is a class of grids $C \in \underline{\Sigma_{n+1}}(GRID) \setminus \underline{\Sigma_n}(GRID)$. Let $\varphi_1 \in \Sigma_{n+1}$ define C w.r.t. class $GRID$. We apply Lemma 13 to find a graph-sentence $\varphi_2 \in \Sigma_{n+1}$ such that $Gr \models \varphi_1 \Leftrightarrow \alpha(Gr) \models \varphi_2$ for all grids Gr. It is clear that φ_2 defines, with respect to the class of localized grid graphs, the isomorphism closure of class $\alpha(C)$.

We then show that there exists no graph-sentence $\psi_2 \in \Sigma_n$ that defines the isomorphism closure of class $\alpha(C)$ w.r.t. class LGG. For assume ψ_2 exists. Use Corollary 15 to choose the related grid-sentence ψ_1. Now, since α is injective, grid-sentence $\psi_1 \in \Sigma_n$ defines class C w.r.t. the class of grids. This is a contradiction. ■

PROPOSITION 17. *For all $n \in \mathbb{N}_{\geq 1}$ we have $\underline{\Sigma_n}(LGG_p) \neq \underline{\Sigma_{n+1}}(LGG_p)$.*

Proof. Fix an arbitrary $n \in \mathbb{N}_{\geq 1}$. By Proposition 16 there exists some sentence $\pi \in \Sigma_{n+1}$ that defines some class $C \in \underline{\Sigma_{n+1}}(LGG) \setminus \underline{\Sigma_n}(LGG)$ w.r.t LGG. Thus the l-pointed version C_p of C is definable w.r.t. LGG_p by formula $(x = x) \wedge \pi$, which is obviously in $[\Sigma_{n+1}]$.

Assume that C_p is definable w.r.t. LGG_p by some formula $\varphi(x) \in \Sigma_n$. Let $\varphi(x) = \overline{Q}\psi(x)$, where $\overline{Q}$ is a vector of second-order quantifiers and $\psi(x)$ is a first-order formula. Sentence $\overline{Q}(\exists x(\psi_{t_0}(x) \wedge \psi(x))) \in \Sigma_n$ defines class C w.r.t. LGG. This contradicts our assumption. ■

5.3 Alternation hierarchy of SOPML over directed graphs

We now prove that the alternation hierarchy of *SOPML* is infinite. We first show this for pointed graphs and then for graphs.

THEOREM 18. *The alternation hierarchy of SOPML over pointed directed graphs is infinite.*

Proof. Fix an arbitrary $n \in \mathbb{N}_{\geq 1}$. Then apply Proposition 17 in order to find some class $H_p \in \underline{\Sigma_{n+1}}(LGG_p) \setminus \underline{\Sigma_n}(LGG_p)$ of l-pointed grid graphs. By Lemma 8 there exists an *SOPML*-sentence that defines class H_p w.r.t. class LGG_p.

Now, class H_p cannot be definable w.r.t. class LGG_p by any *SOPML*-sentence on the n-th level of the alternation hierarchy of *SOPML*. For assume that $\varphi \in \Sigma_n^{ML}$ defines H_p w.r.t LGG_p. Now $St_x(\varphi)$ is an *MSO*-formula in Σ_n that defines H_p w.r.t. LGG_p. ∎

THEOREM 19. *The alternation hierarchy of SOPML over directed graphs is infinite.*

Proof. Fix an arbitrary $n \in \mathbb{N}$. By Proposition 16 there exists a class $H \in \underline{\Sigma_{n+3}}(LGG) \setminus \underline{\Sigma_{n+2}}(LGG)$ of localized grid graphs. We shall first establish that class H is *SOPML*-definable w.r.t. *LGG*.

Consider the following *SOPML*-sentence:

$$\psi = \forall p_x(p_x \rightarrow \Diamond p_x) \wedge \forall p_x(p_x \rightarrow \exists p_y(\neg p_y \wedge \Diamond(p_y \wedge \Diamond p_x)))$$

To see that ψ fixes localizers on *LGG*, notice that the only point u of a localized grid graph that satisfies conditions uRu and $\exists v(v \neq u \wedge uRv \wedge vRu)$ is the localizer. As sentence ψ fixes localizers on *LGG*, Lemma 9 implies that class H is definable w.r.t *LGG* by some *SOPML*-sentence.

Assume then, for contradiction, that $H \in \underline{\Sigma_n^{ML}}(LGG)$. Thus there exists an *SOPML*-sentence $\pi \in \Sigma_n^{ML}$ that defines class H w.r.t *LGG*. It is easy to see that therefore *MSO*-sentence $\varphi = \forall x(St_x(\pi))$ defines H w.r.t *LGG*. To conclude the proof, it now suffices to show that there is an *MSO*-sentence in Σ_{n+2} that is equivalent to φ.

Let $\pi = \overline{\exists p_y ... p_v}(\pi')$, where $\overline{\exists p_y ... p_v}$ is a vector of proposition quantifiers and π' a quantifier-free formula. Consider the following sentence:

$$\forall X \overline{\exists P_y ... P_v}\Big(\ \forall x\Big(X(x) \wedge \forall z(X(z) \rightarrow x = z) \rightarrow\ St_x(\pi')\Big)\Big)$$

It is easy to see that this sentence is equivalent to φ and in Σ_{n+2}. ∎

As the class of Kripke frames is a superclass of the class of finite directed graphs, we immediately obtain the following corollary:

COROLLARY 20. *The alternation hierarchy of SOPML over Kripke frames is infinite.*

6 Concluding remarks

We have shown that the quantifier alternation hierarchy of *SOPML* induces an infinite corresponding semantic hierarchy over the class of finite directed graphs (Theorem 19). While establishing the result, we have defined the notion of a localized structure and characterized the *MSO* alternation hierarchy over localized (finite directed) graphs. Theorem 19 answers a longstanding open problem from [4] (also addressed in [8]). The result is also relatively interesting from the point of view of finite model theory, as *SOPML* is a semantically natural fragment of *MSO* (cf. Theorem 6 in [8]).

In addition to obtaining the results related to alternation hierarchies, we have observed that with regard to expressive power, $MSO = SOPML(E) = SOPML(D)$. Connections of this kind offer an interesting modal perspective on *MSO*. For example, they suggest alternative approaches to *MSO*-games (see [15] for the definition).

Finally, our techniques do not directly yield *strictness* of the hierarchy of *SOPML*. The reason for this is that an *MSO*-formula $\varphi \in \Sigma_n$ cannot necessarily be translated to an *SOPML*-formula in Σ_n^{ML}, as in the general case the first-order quantifiers of φ translate to second-order quantifiers. Therefore, it remains to be investigated whether the *SOPML* alternation hierarchy is strict over finite directed graphs.

Acknowledgements. The author was supported by the Graduate School in Mathematical Logic and Algebra, MALJA, grant no. 118815 from the Academy of Finland. The author wishes to thank Professor Lauri Hella and the referees for their feedback.

BIBLIOGRAPHY

[1] C. Areces, P. Blackburn, and M. Marx. Hybrid logic is the bounded fragment of first order logic. In *Proc. of WOLLIC*, pages 33-50, 1999.
[2] C. Areces, P. Blackburn, and M. Marx. Hybrid Logics: Characterization, Interpolation and Complexity. *Journal of Symbolic Logic*, 66(3):977-1010, 2001.
[3] C. Areces and B. ten Cate. Hybrid Logics. In P. Blackburn, F. Wolter, and J. van Benthem, editors, *Handbook of Modal Logic*. Elsevier, 2006.
[4] J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, 1983.
[5] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
[6] P. Blackburn and J. Seligman. Hybrid Languages. *Journal of Logic, Language and Information*, 4(3):251-272, 1995.
[7] R.A. Bull. On Modal Logic with Propositional Quantifiers. *Journal of Symbolic Logic*, 34(2):257-263, 1969.
[8] B. ten Cate. Expressivity of Second Order Propositional Modal Logic. *Journal of Philosophical Logic*, 35(2):209-223, 2006.
[9] K. Fine. Propositional Quantifiers in Modal Logic. *Theoria*, 36:336-346, 1970.
[10] M. Fitting. Interpolation for First Order S5. *Journal of Symbolic Logic*, 67(2):621-634, 2002.
[11] D. Kaplan. S5 with Quantifiable Propositional Variables. *Journal of Symbolic Logic*, 35(2):352-363, 1970.
[12] P. Kremer. Defining Relevant Implication in a Propositionally Quantified S4. *Journal of Symbolic Logic*, 62(4):1057-1069, 1997.

[13] P. Kremer. On the Complexity of Propositional Quantification in Intuitionistic Logic. *Journal of Symbolic Logic*, 62(2):529-544, 1997.

[14] S. Kripke. A Completeness Theorem in Modal Logic. *Journal of Symbolic Logic*, 21:1-14, 1959.

[15] L. Libkin. *Elements of Finite Model Theory.* Springer, 2004.

[16] J.A. Makowsky and Y.B. Pnueli. Arity and Alternation in Second-Order Logic. *Annals of Pure and Applied Logic*, 78(1-3):189-202, 1996.

[17] O. Matz and N. Schweikardt. Expressive power of monadic logics on words, trees, pictures, and graphs. In J. Flum, E. Grädel and T. Wilke, editors, *Logic and Automata: History and Perspectives*, Texts in Logic and Games, pages 531-552. Amsterdam University Press, 2007.

[18] O. Matz, N. Schweikardt, and W. Thomas. The Monadic Quantifier Alternation Hierarchy over Grids and Graphs. *Information and Computation*, 179(2):356-383, 2002.

[19] O. Matz and W. Thomas. The Monadic Quantifier Alternation Hierarchy over Graphs is Infinite. In *Proc. of LICS*, pages 236-244, 1997.

[20] N. Schweikardt. The Monadic Quantifier Alternation Hierarchy over Grids and Pictures. In *Proc. of CSL*, pages 441-460, 1997.

[21] N. Schweikardt. The Monadic Second-Order Quantifier Alternation Hierarchy over Grids and Pictures. Diplomarbeit, Johannes Gutenberg-Universität Mainz, 1997.

[22] N.V. Shilov and K. Yi. On Expressive and Model Checking Power of Propositional Program Logics. In *Ershov Memorial Conference*, pages 39-46, 2001.

[23] C. Stirling. Games and Modal Mu-Calculus. In *Proc. of TACAS*, pages 298-312, 1996.

[24] L.J. Stockmeyer. The Polynomial-Time Hierarchy. *Theoretical Computer Science*, 3(1):1-22, 1976.

Antti Kuusisto

Department of Mathematics and Statistics

University of Tampere

Kanslerinrinne 1 A, 33014 Tampere,

Finland

antti.j.kuusisto@uta.fi