

On Fragments of Higher Order Logics that on Finite Structures Collapse to Second Order

Flavio Ferrarotti¹(✉), Senén González¹, and José María Turull-Torres^{2,3}

¹ Software Competence Center Hagenberg, Hagenberg im Mühlkreis, Austria
{Flavio.Ferrarotti, Senen.Gonzalez}@scch.at

² Depto. de Ingeniería e Investigaciones Tecnológicas,
Universidad Nacional de La Matanza, San Justo, Argentina
J.M.Turull@massey.ac.nz

³ Massey University, Palmerston North, New Zealand

Abstract. We define new fragments of higher-order logics of order three and above, and investigate their expressive power over finite models. The key unifying property of these fragments is that they all admit inexpensive algorithmic translations of their formulae to equivalent second-order logic formulae. That is, within these fragments we can make use of third- and higher-order quantification without paying the extremely high complexity price associated with them. Although theoretical in nature, the results reported here are more significant from a practical perspective. It turns out that there are many examples of properties of finite models (queries from the perspective of relational databases) which can be simply and elegantly defined by formulae of the higher-order fragments studied in this work. For many of those properties, the equivalent second-order formulae can be very complicated and unintuitive. In particular when they concern properties of complex objects, such as hyper-graphs, and the equivalent second-order expressions require the encoding of those objects into plain relations.

1 Introduction

There are many examples of properties of finite models (queries from the perspective of relational databases) that can be defined by simple and elegant sentences of higher-order logics of order three and above. Take for instance the property of a graph of being an *n-hypercube graph* \mathbf{Q}_n , i.e., an undirected graph whose vertices are binary *n*-tuples and such that two vertices are adjacent iff they differ in exactly one bit. We can build an $(n + 1)$ -hypercube \mathbf{Q}_{n+1} by simply taking

Work supported by **Austrian Science Fund (FWF): [I2420-N31]**. Project: *Higher-Order Logics and Structures*. Initiated during a project sponsored visit of Prof. José María Turull-Torres. The research reported in this paper has been partly supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH.

two isomorphic copies of an n -hypercube \mathbf{Q}_n and adding edges between the corresponding vertices. This strategy can be formally expressed by means of a clear and elegant third-order logic sentence which expresses that \mathbf{G} is an n -hypercube graph for some n iff the following holds:

- There is a sequence \mathcal{S} of graphs, i.e., a *third-order linear digraph* whose nodes are undirected (second-order) graphs.
- The sequence \mathcal{S} starts with a K_2 -graph and ends with \mathbf{G} .
- For every graph \mathbf{G}_{succ} and its immediate predecessor \mathbf{G}_{pred} in the sequence \mathcal{S} , there is a pair of injective functions f_1, f_2 from \mathbf{G}_{pred} to \mathbf{G}_{succ} such that
 - f_1 and f_2 induce in \mathbf{G}_{succ} two isomorphic copies of \mathbf{G}_{pred} ,
 - f_1 and f_2 define a partition in the vertex set of \mathbf{G}_{succ} , and
 - for every edge (x, y) of \mathbf{G}_{succ} , $f_1^{-1}(x) = f_2^{-1}(y)$ or either the edge $(f_1^{-1}(x), f_1^{-1}(y))$ or $(f_2^{-1}(x), f_2^{-1}(y))$ belongs to \mathbf{G}_{pred} .

Yet another example of a property that can be expressed by a simple and elegant third-order sentence is given by the *formula-value query*, consisting on determining whether a propositional formula φ with constants in $\{F, T\}$ evaluates to true. We can express it in third-order logic by writing that there is a sequence of propositional formulae (represented as finite structures) which starts with φ , ends with the formula T , and such that every formula φ_{succ} in the sequence results from applying to exactly one sub-formula of its immediate predecessor φ_{pred} an operations of conjunction, disjunction, or negation which is “ready” to be evaluated (e.g., the conjunction in “ $(T \wedge F)$ ”), or the elimination of a pair of redundant parenthesis (e.g., the parenthesis in “ (T) ”).

The high expressive power of third-order logic is *not* really necessary to characterize hypercube graphs, since they can be recognized in non-deterministic polynomial time (NP) and by Fagin’s theorem [6] existential second-order logic is then powerful enough to define this property. Nevertheless, to define the class of hypercube graphs in second-order logic is certainly more challenging than to define it in third-order logic (see the two strategies for hypercube graphs in [8]). Likewise, we do *not* really need third-order logic to express the formula-value query, since it is in DLOGSPACE [2].

It is then relevant to *distinguish* formulae of order three or higher which *do have* a second-order equivalent formula, from those which (most likely) do not. Beyond the significance of this questions to advance the theory of descriptive complexity, such a development can clearly empower us to write simpler and more intuitive queries, *although still formal*, by taking advantage of the higher level of expressivity of higher-order logics. Provided that those queries can be translated into formal languages with lower complexity of evaluation, this can be done without paying the extremely high complexity price which is associated to higher-order logics. Note that by the results in [6, 13], existential second-order logic captures NP while existential third-order logic already captures $N\text{TIME}(2^{n^{O(1)}})$.

Outline of Contributions. We define new fragments of higher-order logics of order three and above, and investigate their expressive power over finite models.

The key unifying property of these fragments is that they all admit inexpensive algorithmic translations of their formulae to equivalent second-order logic formulae.

We start by defining in Sect. 3 a general *schema* of existential third-order formulae. The schema generalizes the approach described in our previous examples for hypercube graph and formula-value query. It essentially allows us to express an iteration of polynomial length. This iteration is represented by an unfolded sequence of relational structures which can be seen as a computation or derivation. Transitions are then specified by explicitly stating the operations which can be involved in the construction of a given structure in the sequence, when applied to the previous one. As further discussed in Sect. 3, this is a very usual, intuitive, and convenient schema in the expression of properties.

In Sect. 4, we characterize a broader fragment of third-order logic which is no longer restricted to formulae of a fixed schema as in the previous section. We call this fragment TO^P , for *polynomial third-order*, and give a constructive proof of the fact that it collapses to second-order logic. Although the schema of existential third-order formulae proposed in Sect. 3 turned out to be a special case of TO^P , it is still relevant. The translation of the formulae of that schema yields second-order formulae which are more intuitive and clearer than the general translation of TO^P formulae. Moreover, taking into consideration the examples in this paper, the translation proposed in Sect. 3 always results in second-order formulae which use relation variables of considerable smaller arity than the equivalent, but also more general translation proposed in this section. Since the maximum arity of the relation variables in the second-order formulae is *relevant* for the complexity of their evaluation (see [13] among others), it makes sense to study specific schemas of third-order formulae with the aim of finding more efficient translations.

In Sect. 5 we generalize the result in Sect. 4 by characterizing, for each order $i \geq 4$, a fragment $HO^{i,P}$ of the i -th order logic which collapses to second-order. Again this result has interesting practical applications. As an example, consider a multilevel PERT chart such as those commonly used in engineering for planning and scheduling tasks of complex projects. The encoding of higher-order relations of order ≥ 3 into second-order relations can be exploited as a normal form to store such type of complex multilevel PERT charts into a standard relational database. Under certain conditions, higher-order queries of order ≥ 3 could then be synthesised into efficient SQL queries over such normalized relational database. Notice that, a related approach with synthesisation to efficient algorithms was already taken in [14].

We conclude the paper in Sect. 6 where we discuss in detail the expressive power of different fragments of the $HO^{i,P}$ logics and their relationship with known fragments of second-order logic. In particular, adapting Makowsky and Pnueli [16] approach to prove hierarchies of arity and alternation of second-order formulae, we are able to prove interesting strict hierarchies of $HO^{i,P}$ formulae.

Due to space limitations, in most cases we only present sketches of the proofs. Nevertheless, a technical report with the omitted details in Sects. 3–5 is accessible as a CoRR abstract in [9].

2 Preliminaries

We assume familiarity with the basic concepts of finite model theory [5, 15]. We only consider signatures, or vocabularies, which are purely *relational*. We use the classical Tarski's semantics, except that in the context of finite model theory, *only finite* structures or interpretations are considered. If \mathbf{A} is a structure of vocabulary σ , we denote its finite domain by $\text{dom}(\mathbf{A})$ or A . By $\varphi(x_1, \dots, x_r)$ we denote a formula of some logic whose free variables are exactly $\{x_1, \dots, x_r\}$. We write $\mathbf{A} \models \varphi(x_1, \dots, x_r)[\bar{a}]$ to denote that φ is satisfied by the structure \mathbf{A} under *all* valuations v such that $v(x_i) = a_i$ for $1 \leq i \leq r$.

With HO^i we denote the i -th order logic which extends first-order logic with quantifiers of any order $2 \leq j \leq i$, which in turn bind j -th order relation variables. In particular, HO^2 denotes second-order logic as usually studied in the context of finite model theory [5, 15], and HO^3 denotes third-order logic. A *third-order relation type of width w* is a w -tuple $\tau = (r_1, \dots, r_w)$ where $w, r_1, \dots, r_w \geq 1$, and r_1, \dots, r_w are arities of (second-order) relations. For $i \geq 4$, an *i -th order relation type of width w* is a w -tuple $\tau = (\rho_1, \dots, \rho_w)$ where $w \geq 1$ and ρ_1, \dots, ρ_w are $(i-1)$ -th order relation types. A *second-order relation* is a relation in the usual sense. A *third-order relation* of type $\tau = (r_1, \dots, r_w)$ is a set of tuples of (second-order) relations of arities r_1, \dots, r_w , respectively. For $i \geq 4$, an *i -th order relation* of type $\tau = (\rho_1, \dots, \rho_w)$ is a set of tuples of $(i-1)$ -th order relations, of types ρ_1, \dots, ρ_w , respectively. A^τ denotes the set of all higher-order relations of type τ over the domain of individuals A . We use uppercase calligraphic letters $\mathcal{X}^i, \mathcal{Y}^i, \mathcal{Z}^i, \dots$ to denote i -th order variables of order $i \geq 3$, uppercase letters X, Y, Z, \dots to denote second-order variables, and lower case letters x, y, z, \dots to denote first-order variables. With $\mathcal{X}^{i,\tau}$ we denote an i -th order variable of type τ . If \mathcal{X}^i is a third-order variable, we tend to omit the superscript. We sometimes use X^r to denote that X is a second-order variable or arity r . Second-order variables of arity r are valuated with r -ary relations. For $i \geq 3$, i -th order relation variables are valuated with sets of tuples of $(i-1)$ -th order relations according to their relation types. Thus, if v is a valuation, \mathbf{A} is a structure and $\mathcal{X}^{i,\tau}$ is a higher-order variable, then $v(\mathcal{X}^{i,\tau})$ is a i -th order relation of type τ in A^τ . Independently of the order and type of the variables, we say that two valuations v and v' are \mathcal{X} -equivalent if $v(\mathcal{Y}) = v'(\mathcal{Y})$ for every variable \mathcal{Y} other than \mathcal{X} . For any $i \geq 3$, we define the notion of *satisfaction* in HO^i by extending the usual notion of satisfaction of second-order logic formula as follows: $\mathbf{A}, v \models \exists \mathcal{X}^{i,(\rho_1, \dots, \rho_w)}(\varphi(\mathcal{X}))$, where \mathcal{X} is an i -th order relation variable and φ is a well-formed formula, iff there is a i -th order relation \mathcal{R} of type $\tau = (\rho_1, \dots, \rho_w)$ in A^τ , such that $\mathbf{A}, v' \models \varphi(\mathcal{X})$ whenever v' is \mathcal{X} -equivalent to v and $v'(\mathcal{X}) = \mathcal{R}$. Likewise, $\mathbf{A}, v \models \forall \mathcal{X}^{i,(\rho_1, \dots, \rho_w)}(\varphi(\mathcal{X}))$ iff for all i -th order relation \mathcal{R} in A^τ , it holds that $\mathbf{A}, v' \models \varphi(\mathcal{X})$ whenever v' is \mathcal{X} -equivalent to v and $v'(\mathcal{X}) = \mathcal{R}$.

3 A General Schema of Existential Third-Order Formulae

We define next a *general schema* of \exists TO formulae which consists of existentially quantifying a third-order linear digraph of polynomial length (i.e., a sequence of structures that represents a computation) by explicitly stating which operations are the ones which can be involved in the construction of a given structure in the sequence, when applied to the previous one. The schema is as follows:

$$\begin{aligned} \exists \mathcal{C}^{\bar{s}} \mathcal{O}^{\bar{s}\bar{s}} & (\text{TotalOrder}(\mathcal{C}, \mathcal{O}) \wedge \\ & \forall G (\text{First}(G) \rightarrow \alpha_{\text{First}}(G) \wedge \text{Last}(G) \rightarrow \alpha_{\text{Last}}(G)) \wedge \\ & \forall G_{pred} G_{succ} (\mathcal{C}(G_{pred}) \wedge \mathcal{C}(G_{succ}) \wedge \text{Pred}(G_{pred}, G_{succ}) \\ & \rightarrow \varphi(G_{pred}, G_{succ}))), \end{aligned} \quad (1)$$

where

- \mathcal{C} ranges over third-order relations of type $\bar{s} = (i_1, \dots, i_s)$, i.e., over sets of s -tuples of relations of arities $i_1, \dots, i_s \geq 1$.
- $\text{TotalOrder}(\mathcal{C}, \mathcal{O})$, $\text{First}(G)$, $\text{Last}(G)$ and $\text{Pred}(G_{pred}, G_{succ})$ denote fixed second-order formulae which express that \mathcal{O} is a total order over \mathcal{C} , G is the first relational structure in \mathcal{O} , G is the last relational structure in \mathcal{O} , and G_{pred} is the immediate predecessor of G_{succ} in \mathcal{O} , respectively.
- $\alpha_{\text{First}}(G)$ and $\alpha_{\text{Last}}(G)$ denote arbitrary second-order formulae which define, respectively, the properties that the first and last structure in \mathcal{O} should satisfy.
- $\varphi(G_{pred}, G_{succ})$ denotes an arbitrary second-order formula that expresses the transition from G_{pred} to G_{succ} , i.e., which operations can be used to obtain G_{succ} from G_{pred} .

This is a very usual, intuitive, and convenient schema in the expression of natural properties of finite models. For a start, it can clearly be used to express the hypercube and formula-value query as described in the introduction. Significant additional examples are provided by the different relationships between pairs of undirected graphs (G, H) that can be defined as orderings of special sorts. Using schema (1) these relationships can be expressed by defining a set of possible operations that can be applied repeatedly to H , until a graph which is isomorphic to G is obtained. In particular, the following relationships fall into this category: (a) $G \leq_{\text{immersion}} H$: G is an *immersion* in H (see [1, 4, 12]); (b) $G \leq_{\text{top}} H$: G is *topologically embedded* or *topologically contained* in H (see [1, 4, 12]); (c) $G \leq_{\text{minor}} H$: G is a *minor* of H (see [4, 11]); (d) $G \leq_{\text{induced-minor}} H$: G is an *induced minor* of H (see [4]). Interestingly, in all these cases the length of the sequence is at most linear. The operations on graphs needed to define those orderings are: (E) delete an edge, (V) delete a vertex, (C) contract an edge, (T) degree 2 contraction, or subdivision removal, and (L) lift an edge. In particular the set of allowable operations for each of those orderings are: $\{E, V, L\}$ for $\leq_{\text{immersion}}$, $\{E, V, C\}$ for \leq_{minor} , $\{E, V, T\}$ for \leq_{top} , and $\{V, C\}$ for $\leq_{\text{induced-minor}}$ (see [4]).

The classical Kuratowski definition of *planarity*, provides yet another example of a property that can be defined using our schema (1) and also results in a

polynomially bounded sequence of structures. By Wagner's characterization [3]: a graph is planar if and only if it contains neither K_5 nor $K_{3,3}$ as a minor.

Provided some simple conditions are met, every third-order formula of the schema (1) can be translated into an equivalent second-order formula. Note that the proof of this results directly implies that the translation can be done by means of a simple and inexpensive algorithm.

Theorem 1. *Every third-order formulae $\Psi \equiv \exists \mathcal{C}^{\bar{s}} \mathcal{O}^{\bar{s}\bar{s}} \psi(\mathcal{C}, \mathcal{O})$ of the schema (1) can be translated into an equivalent second-order formula Ψ' whenever the following conditions hold.*

- i. *The sub formulae α_{First} , α_{Last} and φ of Ψ are second-order formulae.*
- ii. *There is a $d \geq 0$ such that for every valuation v with $v(\mathcal{C}) = \mathcal{R}$, if $\mathbf{A}, v \models \exists \mathcal{O}^{\bar{s}\bar{s}} \psi(\mathcal{C}, \mathcal{O})$, then $|\mathcal{R}| \leq |\text{dom}(\mathbf{A})|^d$.*

Proof (Sketch). Let us first consider the case in which \mathcal{C} is valuated with sets of non-empty graphs. Let t be the degree of a polynomial bounding the size of those graphs. Our strategy consists on encoding \mathcal{C} as a pair of second-order variables C and E_C of arities $d+t$ and $2(d+t)$, respectively. Notice that every formula that complies with schema (1) stipulates that \mathcal{O} is a linear order of the graphs in \mathcal{C} which represents the stages (or steps) of a computation. Consequently the number of stages needed is bounded by n^d , where n is the size of the structure. Since in turn each stage has a bound on the number of elements it adds or changes (at most n^t), we have to consider a set of $(d+t)$ -tuples.

The encoding into second-order is completed by a total relation $R \subseteq ST \times C$, where C is the union of the domains of all the structures in the sequence. Every node in ST represents one stage, and through the forest R defines a subset of nodes, which is the vertex set of a sub graph (not necessarily connected) of the whole graph (C, E_C) . We use $C|_{R(\bar{x})}$, $E_C|_{R(\bar{x})}$ to denote the restriction of C and E_C , respectively, to $R(\bar{x})$, i.e., $C|_{R(\bar{x})} = \{\bar{y} \mid C(\bar{y}) \wedge R(\bar{x}, \bar{y})\}$ and $E_C|_{R(\bar{x})} = \{(\bar{v}, \bar{w}) \in E_C \mid R(\bar{x}, \bar{v}) \wedge R(\bar{x}, \bar{w})\}$. The sub graph of (C, E_C) which corresponds to the stage $ST(\bar{x})$ is denoted as $(C|_{R(\bar{x})}, E_C|_{R(\bar{x})})$. In this way, Ψ can then be translated into an equivalent second-order formula Ψ' as follows:

$$\begin{aligned} & \exists C^{d+t} E_C^{2(d+t)} ST^d E_{ST}^{2d} R^{2d+t} (\text{Linear}(ST, E_{ST}) \wedge R \subseteq ST \times C \wedge \text{Total}(R) \wedge \\ & \quad \forall \bar{x} \forall \bar{y} ((\text{First}(\bar{x}) \rightarrow \hat{\alpha}_{\text{First}}) \wedge (\text{Last}(\bar{x}) \rightarrow \hat{\alpha}_{\text{Last}}) \wedge \\ & \quad ((ST(\bar{x}) \wedge ST(\bar{y}) \wedge \text{Pred}(\bar{x}, \bar{y})) \rightarrow \\ & \quad \quad \hat{\varphi}((C|_{R(\bar{x})}, E_C|_{R(\bar{x})}, (C|_{R(\bar{y})}, E_C|_{R(\bar{y})}))))), \end{aligned} \tag{2}$$

where

- $\text{Linear}(ST, E_{ST})$, $\text{First}(\bar{x})$, $\text{Last}(\bar{x})$ and $\text{Pred}(\bar{x}, \bar{y})$ denote second-order formulae which express that (ST, E_{ST}) is a linear digraph, \bar{x} is the first node in (ST, E_{ST}) , \bar{x} is the last node in (ST, E_{ST}) , and \bar{x} is the immediate predecessor of \bar{y} in (ST, E_{ST}) , respectively.
- $R \subseteq ST \times C$ and $\text{Total}(R)$ are shorthands for $\forall \bar{x} \bar{y} (R(\bar{x}, \bar{y}) \rightarrow (ST(\bar{x}) \wedge C(\bar{y})))$ and $\forall \bar{x} (ST(\bar{x}) \rightarrow \exists \bar{y} (R(\bar{x}, \bar{y})))$, respectively.

- $\hat{\alpha}_{\text{First}}$ and $\hat{\alpha}_{\text{Last}}$ are second-order formulae built from α_{First} and α_{Last} , respectively, by modifying them to talk about the graph described by \bar{x} through $ST(\bar{x})$, E_{ST} and R .
- $\hat{\varphi}$ is an second-order formula built from φ by modifying it to talk about the graphs described by \bar{x} and \bar{y} through $ST(\bar{x})$, $ST(\bar{y})$, E_{ST} and R .

For the case of relations of arbitrary arity, say S of arity $r \geq 1$, we simply need to consider E_C as an r -ary relation (denoted E_C^S). Thus $E_C^S|_{R(\bar{x})} = \{(\bar{v}_1, \dots, \bar{v}_r) \in E_C^S : R(\bar{x}, \bar{v}_1) \wedge \dots \wedge R(\bar{x}, \bar{v}_r)\}$. If we have a tuple of relations, say $l \geq 1$ relations of arities $r_1, \dots, r_l \geq 1$, respectively, then we have to consider similarly $E_{C_1}^{S_1}, \dots, E_{C_l}^{S_l}$. \square

Remark 1. Every property definable by a third-order formula of the schema (1), where α_{First} , α_{Last} and φ are *existential* second-order formulae and condition (ii) in Theorem 1 also holds, can be checked in NP exactly as it happens for every property definable in existential second-order. It suffices to additionally guess polynomial-sized valuations for the existentially quantified third-order variables. Then, by Fagin’s theorem, we get that every property definable by such kind of third-order formulae can also be defined in existential second-order. Our approach however is fundamentally different. Instead of producing a non deterministic Turing machine, we produce a clear and intuitive second-order formula.

4 TO^P : A Restricted Third Order Logic

We define the logic TO^P as third-order logic restricted to third-order quantification ranging over third-order relations of cardinality bounded by a polynomial in the size of the structure. By contrast, the cardinality of an arbitrary third-order relation \mathcal{R} over a structure \mathbf{A} is exponentially bounded by $2^{|\text{dom}(\mathbf{A})|^{O(1)}}$.

Beyond the usual symbols, the alphabet of TO^P includes a third-order quantifier $\exists^{P,d}$ and countably many third order variable symbols $\mathcal{X}^{d,\bar{r}}$ for every $d \geq 0$ and third-order type \bar{r} . Whenever it is clear from the context, we avoid the superscript d in the TO^P variables. A *valuation* in a structure \mathbf{A} assigns to each variable $\mathcal{X}^{d,\bar{r}}$ a third-order relation \mathcal{R} in $A^{\bar{r}}$, such that $|\mathcal{R}| \leq |\text{dom}(\mathbf{A})|^d$. The quantifier $\exists^{P,d}$ has the following semantics: $\mathbf{A} \models \exists^{P,d} \mathcal{X}^{d,\bar{r}} \varphi(\mathcal{X})$ iff there is TO relation $\mathcal{R}^{\bar{r}}$ of type \bar{r} , such that $\mathbf{A} \models \varphi(\mathcal{X})[\mathcal{R}]$ and $|\mathcal{R}| \leq |\text{dom}(\mathbf{A})|^d$.

The following result shows that the expressive power of TO^P collapses to second-order logic. Same as in the previous section, the proof is constructive and directly implies that the translation can be done algorithmically.

Theorem 2. *Every TO^P formula α can be translated into an equivalent second-order formula α' .*

Proof (Sketch). Let \mathbf{A} be a structure and $\mathcal{R}^{(r_1, \dots, r_s)}$ be a TO^P relation of type (r_1, \dots, r_s) in $A^{(r_1, \dots, r_s)}$ which is bounded by a polynomial of degree $d \geq 0$, i.e., such that $|\mathcal{R}| \leq |\text{dom}(\mathbf{A})|^d$. Assuming that all relations which

appear in the tuples of \mathcal{R} are non-empty, we can use a second-order relation $R_{\mathcal{R}}^{d+r_1+\dots+r_s}$ of arity $(d+r_1+\dots+r_s)$ to encode $\mathcal{R}^{(r_1,\dots,r_s)}$. More precisely, we can use d -tuples from $\text{dom}(\mathbf{A})^d$ as identifiers of tuples of second-order relations in \mathcal{R} , so that whenever a tuple $(a_1, \dots, a_d, a_{d+1}, \dots, a_{d+r_1}, \dots, a_{d+r_1+\dots+r_{s-1}+1}, \dots, a_{d+r_1+\dots+r_s}) \in R_{\mathcal{R}}$, then there is a tuple in \mathcal{R} which can be identified by (a_1, \dots, a_d) , which consists of s second-order relations $S_1^{r_1}, \dots, S_s^{r_s}$, of arities r_1, \dots, r_s , respectively, such that $(a_{d+1}, \dots, a_{d+r_1}) \in S_1, \dots, (a_{d+r_1+\dots+r_{s-1}+1}, \dots, a_{d+r_1+\dots+r_s}) \in S_s$.

The actual translation can be done by structural induction on the TO^P -formula α . We present next the two non-trivial cases.

Atomic Formulae. Let α be of the form $\mathcal{X}^{d,(r_1,\dots,r_s)}(X_1^{r_1}, \dots, X_s^{r_s})$, where $s, r_1, \dots, r_s \geq 1$ and \mathcal{X} is a TO^P variable.

Note that there are 2^s possible patterns of empty and non-empty relations in an s -tuple of second-order relations. We denote by $\omega = (i_1, \dots, i_{|\omega|})$ the pattern of *empty* relations, with $1 \leq i_1 < i_2 < \dots < i_{|\omega|} \leq s$ being the indices of the components which are empty. Correspondingly, we denote by $\bar{\omega} = (j_1, \dots, j_{|\bar{\omega}|})$ the pattern of *non-empty* relations. By abuse of notation, we denote as $\{\omega\}$ and $\{\bar{\omega}\}$ the sets of indices in ω and $\bar{\omega}$, respectively. In particular if $\{\omega\} = \emptyset$ and $\bar{\omega} = (1, \dots, s)$, then all the components of the s -tuple of second-order relations are non-empty.

The idea for the translation is to replace \mathcal{X} with 2^s second-order variables, one for each pattern ω of empty second-order relations. We use $X_{\mathcal{X},e,\omega}$ to denote the second-order variable that encodes those tuples of second-order relations (in the TO^P relation that valuates \mathcal{X}) which follow ω . The arity of the second-order relation $X_{\mathcal{X},e,\omega}$ is $d+r_{j_1}+\dots+r_{j_{|\bar{\omega}|}}$ where $\bar{\omega} = (j_1, \dots, j_{|\bar{\omega}|})$. In what follows we use $\bar{f}_{\bar{\omega}} = \bar{f}_{j_1} \dots \bar{f}_{j_{|\bar{\omega}|}}$ to denote a tuple of first-order variables formed by the concatenation of the tuples of first-order variables $\bar{f}_{j_1} = (f_{j_1 1}, \dots, f_{j_1 r_{j_1}}), \dots, \bar{f}_{j_{|\bar{\omega}|}} = (f_{j_{|\bar{\omega}|} 1}, \dots, f_{j_{|\bar{\omega}|} r_{j_{|\bar{\omega}|}}})$.

Let $\Omega = \{\omega \mid \omega = (i_1, \dots, i_{|\omega|}); 1 \leq i_1 < i_2 < \dots < i_{|\omega|} \leq s; 0 \leq |\omega| \leq s; \bar{\omega} = (j_1, \dots, j_{|\bar{\omega}|}); \{\bar{\omega}\} \cup \{\omega\} = \{1, \dots, s\}; \{\bar{\omega}\} \cap \{\omega\} = \emptyset\}$. The translation to second-order of $\mathcal{X}^{d,(r_1,\dots,r_s)}(X_1^{r_1}, \dots, X_s^{r_s})$ is as follows:

$$\begin{aligned} & \bigvee_{\omega \in \Omega} \left(\left(\left(X_{i_1} = \emptyset \wedge \dots \wedge X_{i_{|\omega|}} = \emptyset \right) \wedge \left(X_{j_1} \neq \emptyset \wedge \dots \wedge X_{j_{|\bar{\omega}|}} \neq \emptyset \right) \wedge \right. \right. \\ & \quad \left. \left. \exists v_1 \dots v_d \left(X_{\mathcal{X},e,\omega}(v_1, \dots, v_d, \bar{f}_{\bar{\omega}}) \vee \right. \right. \right. \\ & \quad \left. \left. \left. \forall \bar{f}_{\bar{\omega}} \left(X_{\mathcal{X},e,\omega}(v_1, \dots, v_d, \bar{f}_{\bar{\omega}}) \leftrightarrow \bigwedge_{l \in \{j_1, \dots, j_{|\bar{\omega}|}\}} X_l(f'_{l1}, \dots, f'_{lr_l}) \right) \right) \right) \right) \end{aligned}$$

Existential Case. Let α be of the form $\exists^{P,d} \mathcal{X}^{d,(r_1,\dots,r_s)}(\varphi)$. In the translation we simply replace the existentially quantified \mathcal{X} by its corresponding 2^s second-order variables and state that no d -tuple can be in more than one of the different second-order relations that encode the value of \mathcal{X} . The formula is as follows:

$$\hat{\alpha} \equiv \{\exists X_{\mathcal{X},e,\omega}^{d+|\bar{f}_{\bar{\omega}}|}\}_{\omega \in \Omega} (\forall z_1 \dots z_d [\bigwedge_{\substack{\omega=(i_1, \dots, i_{|\omega|}) \\ 0 \leq i_1 < i_2 < \dots < i_{|\omega|} \leq s \\ 1 \leq |\omega| \leq s}} \forall \bar{f}_{\bar{\omega}} [X_{\mathcal{X},e,\omega}(z_1, \dots, z_d, \bar{f}_{\bar{\omega}}) \rightarrow \\ (\bigwedge_{\substack{\omega'=(i'_1, \dots, i'_{|\omega'}|) \\ 1 \leq i'_1 < i'_2 < \dots < i'_{|\omega'}| \leq s \\ 0 \leq |\omega'| \leq s; \omega' \neq \omega}} \forall \bar{f}'_{\bar{\omega}'} (\neg X_{\mathcal{X},e,\omega'}(z_1, \dots, z_d, \bar{f}'_{\bar{\omega}'})))]]) \wedge \hat{\varphi},$$

where Ω is as before and φ' is the second-order formula equivalent to the TO^P formula φ , obtained by applying inductively the described translations. \square

5 $\text{HO}^{i,P}$: Restricted Higher Order Logics

We say that a third-order relation \mathcal{R} is *downward polynomially bounded* by d in a structure \mathbf{A} if $|\mathcal{R}| \leq |\text{dom}(\mathbf{A})|^d$. Likewise, we say that a relation \mathcal{R} of order $i > 3$ is *downward polynomially bounded* by d in \mathbf{A} if $|\mathcal{R}| \leq |\text{dom}(\mathbf{A})|^d$ and further every relation \mathcal{R}_i^j of order $3 \leq j < i$ which appears in a tuple of \mathcal{R} is downward polynomially bounded by d .

For $i = 4$, we define $\text{HO}^{4,P}$ as the extension of TO^P with quantifiers that range over downward polynomially bounded relations of order 4. More general, for $i \geq 5$ we define $\text{HO}^{i,P}$ as the extension of $\text{HO}^{i-1,P}$ with quantifiers that range over downward polynomially bounded relations of order i .

Beyond the symbols in the alphabet of TO^P , the alphabet of $\text{HO}^{i,P}$ includes a j -th order quantifier $\exists^{j,P,d}$ for every $i \geq j \geq 4$, as well as countably many variable symbols $\mathcal{X}^{j,d,\tau}$ for every j -th order type τ . We sometimes avoid the superscripts d and τ for clarity.

A *valuation* in a structure \mathbf{A} assigns to each variable $\mathcal{X}^{j,d,\tau}$ a j -th order relation \mathcal{R} in A^τ which is downward polynomially bounded by d . The quantifier $\exists^{j,P,d}$ has the following semantics: $\mathbf{A} \models \exists^{j,P,d} \mathcal{X}^{j,d,\tau} \varphi(\mathcal{X})$ iff there is a j -th order relation \mathcal{R} of type τ , such that $\mathbf{A} \models \varphi(\mathcal{X})[\mathcal{R}]$ and \mathcal{R} is downward polynomially bounded by d in \mathbf{A} .

Same as with TO^P , for every order $i \geq 4$ the expressive power of $\text{HO}^{i,P}$ collapses to second-order logic.

Theorem 3. *For every order $i \geq 3$, every $\text{HO}^{i,P}$ formula α can be translated into an equivalent second-order formula α' .*

The actual proof of this theorem is quite long and cumbersome. Due to space limitations we omit it here. The details can nevertheless be consulted in the technical report in [9]. To gain some intuition on how this translations works, let us consider the case of $\text{HO}^{4,P}$. The general idea is to represent the fourth-order relations as a *normalized relational database*. Assume w.l.o.g. that the type of every relations of order 3 and 4 has width $s \geq 1$, and that every such relation is downward polynomially bounded by $d \geq 1$. In the case of a formula of the form $\mathcal{X}^{4,d,\tau}(\mathcal{Y}_1^3, \dots, \mathcal{Y}_s^3)$ of $\text{HO}^{4,P}$ we can represent the fourth-order variable $\mathcal{X}^{4,d,\tau}$

using 2^s second-order variables $X_{\mathcal{X}^4, \omega_{3, \mathcal{X}}}$ of arities between d and $d + (s \cdot (d + s))$, depending on the pattern $\omega_{3, \mathcal{X}}$ of non-empty relations. Thus, each $X_{\mathcal{X}^4, \omega_{3, \mathcal{X}}}$ can encode the tuples of (non-empty) third-order relations whose pattern is $\omega_{3, \mathcal{X}}$. In turns, each \mathcal{Y}_j^3 can be represented by 2^s second-order variables as explained in Sect. 4 for the case of TO^P .

6 Fragments of $\text{HO}^{i,P}$ Formulae

The aim of this section is to gain a better understanding of the syntactic restrictions relevant as to the expressive power of the $\text{HO}^{i,P}$ logics.

In [17] we showed that for any $i \geq 3$ the deterministic inflationary fixed-point quantifier (IFP) in HO^i (i.e., where the variable which is bound by the IFP quantifier is an $(i + 1)$ -th order variable) is expressible in $\exists \text{HO}^{i+1}$. Let $IFP|_P$ denote the restriction of IFP where there is a positive integer d such that in every structure \mathbf{A} , the number of stages of the fixed-point is bounded by $|\text{dom}(\mathbf{A})|^d$. And let $(SO + IFP)$ denote second-order logic extended with the deterministic inflationary fixed-point quantifier, where the variable which is bound by the IFP quantifier is a third order variable. Note that the addition of such IFP quantifier to second-order means that we can express iterations of length exponential in $|\text{dom}(\mathbf{A})|$, so that it is strongly conjectured that $(SO + IFP)$ strictly includes second-order logic as to expressive power. However, as a consequence of Theorem 2 this is not the case with $IFP|_P$.

Corollary 1. *For every formula in $(SO + IFP|_P)$ there is an equivalent second-order formula.*

Let us define ΣTO_n^P as the restriction of TO^P to prenex formulae of the form $Q_1 V_1 \dots Q_k V_k (\varphi)$ such that:

- $Q_1, \dots, Q_k \in \{\forall^{P,d}, \exists^{P,d}, \forall, \exists\}$.
- V_i for $1 \leq i \leq k$ is either a second-order or TO^P variable (depending on Q_i).
- φ is a TO^P formula free of TO^P as well as second-order quantifiers (first-order quantifiers as well as free TO^P and second-order variables are allowed).
- The prefix $Q_1 V_1 \dots Q_k V_k$ starts with an existential block of quantifiers and has at most n alternating (between universal and existential) blocks.

By the well known Fagin-Stockmeyer characterization [18] of the polynomial-time hierarchy, for every $n \geq 1$ the prenex fragment Σ_n of second-order logic captures the level Σ_n^{poly} of the polynomial-time hierarchy. From the proof of Theorem 2 is immediate that every formula in ΣTO_n^P can be translated into an equivalent second-order formula in Σ_n .

Corollary 2. *ΣTO_n^P captures Σ_n^{poly} .*

Consider the $AA(r, m)$ classes of second-order logic formulae where all quantifiers of *whichever* order are grouped together at the beginning of the formula, forming up to m alternating blocks of consecutive existential and universal quantifiers, and such that the arity of the second-order variables is bounded by r . Note that,

the order of the quantifiers in the prefix may be mixed. As shown by Makowsky and Pnueli [16], the $AA(r, m)$ classes constitute a strict hierarchy of arity and alternation. Their strategy to prove this result consisted in considering the set $AUTOSAT(AA(r, m))$ of formulae of $AA(r, m)$ which, encoded as finite structures, satisfy themselves. As the well known diagonalization argument applies, it follows that $AUTOSAT(AA(r, m))$ is not definable by any formulae of $AA(r, m)$, but it is definable in a higher level of the same hierarchy. Similarly to Makowsky and Pnueli arity and alternation hierarchy of second-order formulae, we can we define hierarchies of HO^i and $HO^{i,P}$ formulae as follows.

Definition 1 (*AA^i - and AAD^i -hierarchies*). *The maximum-width of a type $\tau = (\rho_1, \dots, \rho_s)$ of order $i \geq 3$ (denoted as $\max\text{-width}(\tau)$) is defined as follows:*

- $\max\text{-width}(\tau) = \max(\{s, \rho_1, \dots, \rho_s\})$ if $i = 3$.
- $\max\text{-width}(\tau) = \max(\{s, \max\text{-width}(\rho_1), \dots, \max\text{-width}(\rho_s)\})$ if $i > 3$.

For $r, m, i \geq 1$, we define the level $AA^i(r, m)$ of the AA^i -hierarchy as the class of formulae $\varphi \in HO^{i+1}$ of the form $Q_1V_1 \dots Q_kV_k(\psi)$ such that:

- i* ψ is a quantifier-free HO^i -formula.
- ii* For $j = 1, \dots, k$, each Q_j is either an existential or universal quantifier and each V_j is a variable of order $\leq i + 1$.
- iii* The prefix $Q_1V_1 \dots Q_kV_k$ has at most m alternating blocks of quantifiers.
- iv* If V_j is a second-order variable, then its arity is bounded by r .
- v* If V_j is of order ≥ 3 , then the maximum-width of the type τ of V_j is $\leq r$.

For $r, m, i, d \geq 1$, the level $AAD^i(r, m, d)$ of the AAD^i -hierarchy is obtained by adding the following condition to the definition of $AA^i(r, m)$.

- *If V_j is a variable of order ≥ 3 , then the quantifier Q_j has a superscript $d_j \leq d$, which denotes the degree of the polynomial bounding the size of the valuations of V_j (recall definition of $HO^{i,P}$).*

Note that in the formulae of the AA^i and AAD^i hierarchies the quantifiers of the highest order do not necessarily precede all the remaining quantifiers in the prefix, as it is the case in the Σ_m^i hierarchies of higher-order logics.

As proven in [7, 10], it is possible to generalise Makowsky and Pnueli result regarding the AA -hierarchy [16] to every higher-order logic of order $i \geq 2$.

Theorem 4 (Theorem 4.28 in [7]). *For every $i, r, m \geq 1$, there are Boolean queries not expressible in $AA^i(r, m)$ but expressible in $AA^i(r+c(r), m+6)$, where $c(r) = 1$ for $r > 1$ and $c(r) = 2$ for $r = 1$.*

The proof of the previous result in [7] also follows the strategy introduced in [16]. That is, it uses the diagonalization argument to prove the lower bound for the definability of $AUTOSAT(AA^i(r, m))$, and shows a formula in $AA^i(r+c(r), m+6)$ that defines $AUTOSAT(AA^i(r, m))$ to prove the upper bound.

Interestingly, the formula $\psi_A \in AA^i(r+c(r), m+6)$ used in the proof of Proposition 4.27 in [7] to define $AUTOSAT(AA^i(r, m))$ can be straightforwardly

translated to a formula ψ_A^d that defines $AUTOSAT(AAD^i(r, m, d))$. We simply need to qualify with an appropriate superscript $d_j \leq d$ the existential and universal quantifiers associated to each higher-order variable V_j of order ≥ 3 which appears in ψ_A , so that V_j becomes restricted to range over higher-order relations which are downward polynomially bounded by d_j . Since d as well as the order and the maximum-width of the higher-order types are bounded, a finite set of variables is still sufficient to encode the valuations for the different variables that might appear in any arbitrary sentence in $AAD^i(r, m, d)$. The resulting ψ_A^d formula is clearly in $AAD^i(r + c(r), m + 6, d)$. It no longer defines $AUTOSAT(AA^i(r, m))$ when $i \geq 2$, since the size of the higher-order relations that interpret the higher-order variables of order ≥ 3 in ψ_A^d are downward polynomially bounded by d and thus insufficient to encode every possible valuation for variables of order ≥ 3 . This is clearly not a problem if we only consider sentences in $AAD^i(r, m, d)$. In fact, in this latter case we need higher-order variables that encode only those valuations which are downward polynomially bounded by some positive integer $\leq d$.

The previous observation together with the fact that by using the same diagonalization argument we can prove that $AUTOSAT(AAD^i(r, m, d))$ is not definable in $AAD^i(r, m, d)$, gives us the following strict hierarchies of $HO^{i,P}$ formulae for every order i .

Theorem 5. *For every $i \geq 2$ and $r, m, d \geq 1$, there are Boolean queries not expressible in $AAD^i(r, m, d)$ but expressible in $AAD^i(r + c(r), m + 6, d)$, where $c(r) = 1$ for $r > 1$ and $c(r) = 2$ for $r = 1$.*

Lemma 1 is a direct consequence of the translations in Theorems 2 and 3.

Lemma 1. *Let $r, m, d \geq 1$ and $i \geq 2$. For every sentence φ in $AAD^i(r, m, d)$, there is an equivalent sentence φ' in $AA^1(r^i + d \cdot (i - 1), m + 2)$.*

The following result suggest that the exact converse of the previous lemma is unlikely to hold. It further shows the relationship between the levels of the arity and alternation hierarchy of Makowsky and Pnueli and the AAD^i hierarchies.

Lemma 2. *Let $r, m \geq 1$. For every second-order sentence φ in $AA^1(r, m)$, there are three sentences φ', φ'' and φ''' , each of them equivalent to φ , such that:*

- i. $\varphi' \in AAD^{\lceil \frac{r}{2} \rceil + 1}(2, m, 2)$.
- ii $\varphi'' \in AAD^2(2, m, r)$.
- iii $\varphi''' \in AAD^2(\lceil \sqrt[2]{r} \rceil, m, \lceil \sqrt[2]{r} \rceil (\lceil \sqrt[2]{r} \rceil - 1))$.

A Sketch of the proof of Lemma 2 is included in Appendix A.

Our final result gives a fine grained picture of the effect, as to expressive power, of simultaneously bounding the arity, alternation and maximum degree of the $HO^{i,P}$ -sentences.

Theorem 6. *For every $r, m, d \geq 1$, there are Boolean queries not expressible in $AAD^2(r, m, d)$ but expressible in $AAD^c(2, m + 8, 2)$ as well as in $AAD^2(2, m + 8, (r + 1)^2 + d)$ and in $AAD^2(q, m + 8, q(q - 1))$, where $c = \left\lceil \frac{((r+1)^2+d}{2} \right\rceil + 1$ and $q = \left\lceil \sqrt[2]{((r+1)^2 + d)} \right\rceil$.*

Proof. By Lemma 1, we get that $AAD^2(r, m, d) \subseteq AA^1(r^2 + d, m + 2)$, i.e., the class of Boolean queries definable by TO^P -sentences in $AAD^2(r, m, d)$ is included in those definable by second-order sentences in $AA^1(r^2 + d, m + 2)$. In turns, by Makowsky and Pnueli [16] result (first level ($i = 1$) in Theorem 4), we get that $AA^1(r^2 + d, m + 2) \subseteq AA^1(r^2 + d + 1, m + 8)$. Finally, by Lemma 2 we get that the class of Boolean queries definable in $AA^1(r^2 + d + 1, m + 8)$ is included in $AAD^c(2, m + 8, 2)$ as well as in $AAD^2(2, m + 8, (r + 1)^2 + d)$ and in $AAD^2(q, m + 8, q(q - 1))$. \square

Appendix A Proof Sketch of Lemma 2

All three sentences φ' , φ'' and φ''' can be defined by structural induction on φ . We show only the non trivial cases.

If φ is an atomic formula of the form $X(x_1, \dots, x_s)$ where $s \leq r$. Then

- φ' is $\mathcal{X}^c(\mathcal{X}_1^{c-1}, \mathcal{X}_2^{c-1}) \wedge \mathcal{X}_1^{c-1}(\mathcal{X}_1^{c-2}) \wedge \mathcal{X}_2^{c-1}(\mathcal{X}_2^{c-2}, \mathcal{X}_3^{c-2}) \wedge \dots \wedge \mathcal{X}_1^3(X_1) \wedge \mathcal{X}_2^3(X_2) \wedge \dots \wedge \mathcal{X}_{c-2}^3(X_{c-2}, X_{c-1}) \wedge X_{c-1}(x_{s-1}, x_s) \wedge X_{c-2}(x_{s-3}, x_{s-2}) \wedge \dots \wedge X_1(\bar{x})$, where $\bar{x} = (x_1)$ or $\bar{x} = (x_1, x_2)$ depending on whether s is odd or even, respectively, and $c = \lceil \frac{r}{2} \rceil + 1$.
- φ'' is $\bigwedge_{\bar{w} \in W} \left(\left(\bigwedge_{1 \leq j < l \leq s} \alpha_{i,j} \right) \rightarrow \psi_{\bar{w}} \right)$, where
 - $W = \{(i_1, i_2, \dots, i_s) \mid 1 \leq j \leq s \text{ and } 1 \leq i_j \leq j\}$
 - $\alpha_{i,j} = \begin{cases} x_i = x_j & \text{if } i=j \\ x_i \neq x_j & \text{if } i \neq j \end{cases}$
 - $\psi_{\bar{w}}$ is $X_{1_{\bar{w}}}(x_1, x_1)$ if $i_u = i_v$ for every $i_u, i_v \in \bar{w}$.
Otherwise $\psi_{\bar{w}}$ is $\bigwedge_{(u,v) \in A_{\bar{w}}} X_{1_{\bar{w}}}(x_u, x_v)$, where $A_{\bar{w}} = \{(u, v) \mid 1 \leq u < v \leq s, i_u, i_v \text{ are the } u\text{-th and } v\text{-th elements of } \bar{w}, \text{ respectively, } i_u = u, i_v = v, \text{ and } i_t < t \text{ for all } v < t < u\}$.
- φ''' is $\mathcal{X}(X_1, \dots, X_t) \bigwedge_{1 \leq i \leq t-1} X_i(x_{k_i+1}, \dots, x_{k_i+u}) \wedge X_t(x_{s-t}, \dots, x_s)$
Where $t = \lceil \sqrt{r} \rceil$, $k_i = ((i - 1)t) + \min((i - 1), (s - t) \bmod (t - 1))$, $u = \lfloor \frac{s-t}{t-1} \rfloor + c$, and $c = 1$ if $i \leq (s - t) \bmod (t - 1)$, $c = 0$ otherwise.

If φ is a formula of the form $\exists X(\psi)$ where X is a second-order variable of arity $s \leq r$. Then

- φ' is $\exists^{c,P,2} \mathcal{X}^c \exists^{c-1,P,2} \mathcal{X}_1^{c-1} \mathcal{X}_2^{c-1} \exists^{c-2,P,2} \mathcal{X}_1^{c-2} \mathcal{X}_2^{c-2} \mathcal{X}_3^{c-2} \dots \exists^{P,2} \mathcal{X}_1^3 \mathcal{X}_2^3 \dots \mathcal{X}_{c-2}^3 \exists X_1 X_2 \dots X_{c-1}(\psi')$, where again $c = \lceil \frac{s}{2} \rceil + 1$, and ψ' is the formula in $AAD^c(2, m, 2)$ equivalent to ψ , obtained by applying the translation inductively.
- φ'' is $\exists^{P,s} \mathcal{X}_{\bar{w}_1} \mathcal{X}_{\bar{w}_2} \dots \mathcal{X}_{\bar{w}_{|W|}} \exists X_{1_{\bar{w}_1}} X_{2_{\bar{w}_1}} X_{1_{\bar{w}_2}} X_{2_{\bar{w}_2}} \dots X_{1_{\bar{w}_{|W|}}} X_{2_{\bar{w}_{|W|}}}(\psi'')$, where $\bar{w}_1, \bar{w}_2, \dots, \bar{w}_{|W|}$ is an arbitrary lexicographic order of W and ψ'' is the formula in $AAD^2(2, m, s)$ equivalent to ψ , obtained by applying the translation inductively.

- φ''' is $\exists^{P,h} \mathcal{X} \exists X_1 X_2 \cdots X_t (\psi''')$ where $t = \lceil \sqrt[2]{s} \rceil$, $h = t(t-1)$ and ψ''' is the formula in $AAD^2(t, m, t(t-1))$ equivalent to ψ , obtained by applying the translation inductively.

It is not difficult to show by structural induction that φ' , φ'' and φ''' are equivalent to φ . We only need to see that every second-order relation of arity $s \leq r$ can be encoded as a higher-order relation of the type used in the translation and with the required polynomial bound.

References

1. Abu-Khzam, F.N., Langston, M.A.: Graph coloring and the immersion order. In: Warnow, T., Zhu, B. (eds.) COCOON 2003. LNCS, vol. 2697, pp. 394–403. Springer, Heidelberg (2003). doi:[10.1007/3-540-45071-8_40](https://doi.org/10.1007/3-540-45071-8_40)
2. Beaudry, M., McKenzie, P.: Ciuits, matrices, and nonassociative computation. In: Proceedings of the Seventh Annual Structure in Complexity Theory Conference, Boston, Massachusetts, USA, 22–25 June 1992, pp. 94–106 (1992)
3. Bollobás, B.: Modern Graph Theory. Graduate Texts in Mathematics, vol. 184. Springer, Heidelberg (2002). Corrected edition
4. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science. Springer, Heidelberg (1999)
5. Ebbinghaus, H.D., Flum, J.: Finite Model Theory. Perspectives in Mathematical Logic, 2nd edn. Springer, Heidelberg (1999)
6. Fagin, R.: Generalized first-order spectra and polynomial-time recognizable sets. In: Karp, R. (ed.) Complexity of Computations. SIAM-AMS Proceedings, vol. 7, pp. 27–41. American Mathematical Society (1974)
7. Ferrarotti, F.: Expressibility of higher-order logics on relational databases: proper hierarchies. Ph.D. thesis, Department of Information Systems, Massey University, Wellington, New Zealand (2008). <http://hdl.handle.net/10179/799>
8. Ferrarotti, F., Ren, W., Turull-Torres, J.M.: Expressing properties in second- and third-order logic: hypercube graphs and SATQBF. Logic J. IGPL **22**(2), 355–386 (2014)
9. Ferrarotti, F., Tec, L., Torres, J.M.T.: On higher order query languages which on relational databases collapse to second order logic. CoRR abs/1612.03155 (2016). <http://arxiv.org/abs/1612.03155>
10. Ferrarotti, F., Turull-Torres, J.M.: Arity and alternation: a proper hierarchy in higher order logics. Ann. Math. Artif. Intell. **50**(1–2), 111–141 (2007)
11. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag New York, Inc., Secaucus (2006)
12. Grohe, M., Kawarabayashi, K., Marx, D., Wollan, P.: Finding topological subgraphs is fixed-parameter tractable. In: Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing, STOC 2011, pp. 479–488. ACM, New York (2011)
13. Hella, L., Turull-Torres, J.M.: Computing queries with higher-order logics. Theor. Comput. Sci. **355**(2), 197–214 (2006)
14. Itzhaky, S., Gulwani, S., Immerman, N., Sagiv, M.: A simple inductive synthesis methodology and its applications. SIGPLAN Not. **45**(10), 36–46 (2010)
15. Libkin, L.: Elements of Finite Model Theory. Texts in Theoretical Computer Science, EATCS. Springer, Heidelberg (2004)

16. Makowsky, J.A., Pnueli, Y.B.: Arity and alternation in second-order logic. *Ann. Pure Appl. Logic* **78**(1–3), 189–202 (1996)
17. Schewe, K.D., Turull-Torres, J.M.: Fixed-point quantifiers in higher order logics. In: *Proceedings of the 2006 Conference on Information Modelling and Knowledge Bases XVII*, pp. 237–244. IOS Press, Amsterdam (2006)
18. Stockmeyer, L.J.: The polynomial-time hierarchy. *Theoret. Comput. Sci.* **3**(1), 1–22 (1976)