Characterizing the Exponential-Space Hierarchy Via Partial Fixpoints

Florian Bruse David Kronenberger Martin Lange
University of Kassel, Germany
{florian.bruse|martin.lange}@uni-kassel.de

The characterization of PSPACE-queries over ordered structures as exactly those expressible in first-order logic with partial fixpoints (Vardi'82) is one of the classical results in the field of descriptive complexity. In this paper, we extend this result to characterizations of k-EXPSPACE-queries for arbitrary k, characterizing them as exactly those expressible in order-$k+1$-higher-order logic with partial fixpoints. For $k > 1$, the restriction to ordered structures is no longer necessary due to the high expressive power of higher-order logic.

1 Introduction

Computational complexity studies the difficulty of computation problems with regards to the consumption of computational resources, most prominently time and space. Descriptive complexity, as a subdomain of both computational complexity and formal logic, has taken this study to a more abstract level by characterizing classes of problems, i.e., complexity classes, through logical definability. This achieves the characterization of the difficulty, resp. complexity of problems without resorting to measuring the use of computational resources, as this ultimately depends on the choice of an underlying model of computation like a Turing machine for instance. Descriptive complexity thus manages to characterize the difficulty of problems through the structure of the problem alone, regardless of an underlying model of computation. One can argue, though, that the resources used to measure complexity are logical operators that give the underlying logics their expressiveness, like predicate or fixpoint quantifiers

Descriptive complexity started off with Fagin's seminal result [3] showing that the well-known complexity class NP coincides with $\exists$SO, the set of problems definable in existential second-order logic. Stockmeyer extended this to a characterisation of problems between NP and PSPACE by means of second-order logic (SO), known as the polynomial hierarchy (PH) [9].

An interesting – and still open – question asks for a logical characterisation of the complexity class P. This is believed to open ways to tackle the famous P=NP question. One of the major obstacles here is the lack of a total order on the elements of a structure forming an instance of some computational problem, like a graph for instance. When processing graphs with a computational model like a Turing machine, it can be assumed to be totally ordered due to the way that it needs to be represented as an input. For logical formulas, operating directly on structures and not on string representations thereof, this is not the case. On the other hand, a total order helps immensely; it enables iteration over all elements of the structure. Moreover, a logical characterisation of the complexity class P is known when inputs to its problems are assumed to be explicitly ordered. This is known as the Immerman-Vardi Theorem [5, 10], stating that the complexity class P on ordered structures is captured by the extension of first-order logic with least fixpoint quantifiers (FO+LFP).

Fixpoint quantifiers turned out to be a useful tool in descriptive complexity. Immerman lifted the Immerman-Vardi Theorem to a characterisation of the complexity class EXPTIME by second-order logic

Submitted to:
FICS 2024

with least fixpoint quantifiers (SO+LFP) [6]. Note that the fixpoint quantifiers in SO+LFP are not the same as the ones in FO+LFP. The LFP in FO+LFP refers to least fixpoints of first-order functions mapping tuples of elements to tuples of elements. This can be expressed in SO, i.e. FO+LPF $\subseteq$ SO. The LFP in SO+LFP refers to fixpoints of second-order functions, mapping predicates to predicates. This naturally gives rise to the question after characterisations of classes in the exponential-time hierarchy by means of higher-order logic with fixpoints. Indeed, Freire and Martins [4] showed that for any $k \geq 2$, the class k-EXPTIME of problems solvable in k-fold exponential time is captured by HO^{k+1}+LFP, i.e. higher-order formulas of order at most $k+1$ with corresponding least fixpoint quantifiers.

Given the rather complete picture for time complexity, it is natural to ask whether space complexity is also open to logical characterizations in the same fashion. Another celebrated result in descriptive complexity, made use of in e.g., the Abiteboul-Vianu Theorem [1], is due to Vardi [10] (not to be confused with the Immerman-Vardi Theorem, from [5] and also [10]). It states that the class PSPACE on ordered structures is captured by FO+PFP, i.e., the extension of first-order logic by *partial* fixpoints.

In this paper we extend the descriptive complexity of classes in the exponential space hierarchy with the Vardi's result at the basis, just like Freire and Martins have done for the time hierarchy with the Immerman-Vardi Theorem at its basis. We show that, for any $k \geq 1$, the complexity class k-EXPSPACE of problems solvable using at most k-fold exponential space, is captured by the logic HO^{k+1}+PFP of formulas of order at most $k+1$ with partial fixpoint quantifiers.

2 Preliminaries

Let $n,k \in \mathbb{N}$. We write 2^n_k for the following: $2^n_k = n$ if $k = 0$, and $2^n_{k+1} = 2^{2^n_k}$.

2.1 Space-bounded Turing Machines

A deterministic Turing machine (DTM) is a tuple $\mathscr{M} = (Q,\Sigma,\Gamma,\Box,\delta,q_{\text{init}},q_{\text{acc}},q_{\text{rej}})$ where Q is a finite set of *states*, Σ is a finite, nonempty *input alphabet*, $\Gamma \supseteq \Sigma$ is a finite, nonempty *tape alphabet*, $\Box \in \Gamma \setminus \Sigma$ is the *blank symbol*, $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L,N,R\}$ is the *transition function*, and $q_{\text{init}},q_{\text{acc}},q_{\text{rej}} \in Q$ are the unique *starting*, *accepting* and *rejecting states*.

A *configuration* of a DTM is a tuple (q,h,t) where $q \in Q$ is the current state, $h \in \mathbb{N}$ is the head position, and $t\colon \mathbb{N} \to \Gamma$ is the tape content. The *initial configuration* on input word $w \in \Sigma^*$ is given by $(q_{\text{init}},0,t)$ with $t(i) = w_i$ if $i < |w|$ and $t(i) = \Box$ otherwise. The unique *accepting* and *rejecting configurations* are given by $(q_{\text{acc}},0,t)$, resp. $(q_{\text{rej}},0,t)$ where $t(i) = \Box$ for all i in both cases.

A configuration $C = (q',h',t')$ is the, necessarily unique, *successor configuration* of (q,h,t) if (I) $q \notin \{q_{\text{acc}},q_{\text{rej}}\}$, (II) $\delta(q,t(h)) = (q',\gamma,D)$ for some $\gamma \in \Gamma, D \in \{L,N,R\}$, (III) $t'(i) = \gamma$ if $i = h$ and $t'(i) = t(i)$ otherwise, and (IV) $h' = h-1$ if $D = L$ and $h > 0$, $h' = h$ if $D = N$, and $h' = h+1$ if $D = R$. A (partial) *computation* of $\mathscr{M}$ on input w is a finite or infinite sequence of configurations $C_0,C_1,\ldots$ where C_0 is the initial configuration of $\mathscr{M}$ on w, and C_{i+1}, if it exists, is the successor configuration of C_i. Such a computation is *maximal* if it is either infinite or its last configuration is the accepting or the rejecting configuration. Note that each $\mathscr{M}$ has exactly one maximal computation for each w, whence from now on we talk about *the* computation of $\mathscr{M}$ on w. We say that $\mathscr{M}$ *accepts* w if its unique maximal computation on w ends with the accepting configuration, and we write $L(\mathscr{M})$ for the set of words accepted by $\mathscr{M}$. Conversely, $\mathscr{M}$ *rejects* w if its unique maximal computation on w ends in the rejecting configuration or if it is infinite. In the latter case, we say that the computation *diverges*.

We say that a non-diverging computation $C_0,\ldots,C_k$ on input some w consumes space n, written

$\text{space}_{\mathscr{M}}(w) = n$, if $n = 1 + \max\{h_i \mid C_i = (q_i, h_i, t_i)\}$. Obviously, if the head never advances beyond position $n-1$, then $t_i(j) = \square$ for all $0 \le i \le k$ and $j > n-1$. Let $f \colon \mathbb{N} \to \mathbb{N}$ be a function. We say that $\mathscr{M}$ is f-space-bounded if $\mathscr{M}$ has no diverging computations on any input and, for all n, we have $\max\{\text{space}_{\mathscr{M}}(w) \mid w \in \Sigma^n\} \le f(n)$. We say that $\mathscr{M}$ is *k-fold-exponential space bounded* if there is a polynomial $p(n)$ such that $\mathscr{M}$ is $2_k^{p(n)}$-space-bounded.

2.2 Higher-Order Logic with Partial Fixpoints

In order to keeps things notationally simple, we restrict ourselves to the class of labelled transition systems (LTS), or labelled graphs. Let $\mathbf{P} = \{p, q, \dots\}$ be a set of *propositions* and let $\mathbf{A} = \{a, b, \dots\}$ be a set of *actions* or *transition relation* or *edge relations*. An LTS is a tuple $T = (S, A, \ell)$ where $S = \{s, t, \dots\}$ is a finite, nonempty set of *states*, $A \subseteq S \times \mathbf{A} \times S$ is the *transition relation* and $\ell \colon S \to 2^{\mathbf{P}}$ labels each state by the set of propositions valid in it. We write $s \xrightarrow{a} t$ instead of $(s, a, t) \in A$.

Types. The set of types is defined via the grammar

$$\tau, \tau_1, \dots, \tau_n ::= \bullet \mid \tau_1, \dots, \tau_n \mid (\tau)$$

where $\bullet$ is the *ground type* or *type of individuals* of *order* $ord(\bullet) = 1$, $\tau_1, \dots, \tau_n$ is a *compound type* of order $ord(\tau_1, \dots, \tau_n) = \max\{ord(\tau_1), \dots, ord(\tau_n)\}$, and where (τ) is a *set type* of order $ord((\tau)) = 1 + ord(\tau)$.

Given an LTS $T = (S, A, \ell)$, the semantics $[\![\tau]\!]^T$ of a type τ is given by

$$[\![\bullet]\!]^T = S, \qquad [\![\tau_1, \dots, \tau_n]\!]^T = [\![\tau_1]\!]^T \times \dots \times [\![\tau_n]\!]^T \qquad [\![(\tau)]\!]^T = 2^{[\![\tau]\!]^T}.$$

We often compress compound and set types by writing e.g., $(\tau_1, \dots, \tau_n)$.

The following is straightforward to prove by induction on the structure of types.

Lemma 1. *For any τ of order k and any LTS T, with state set S, $|[\![\tau]\!]^T|$ is $k-1$-fold exponential in $|S|$.*

Given an LTS as above, and some $f \colon [\![\tau]\!]^T \to [\![\tau]\!]^T$ for τ of order at least 2, we define its *partial fixpoint* PFP f via

$$\text{PFP } f = \begin{cases} F^i, \text{ if } i \text{ exists such that } F^i = F^{i+1} \\ \emptyset, \text{ otherwise,} \end{cases}$$

where $F^0 = \emptyset$ and $F^{i+1} = f(F^i)$. By an obvious counting argument, if there is i such that a nontrivial partial fixpoint exists, then there already is one bounded by $|[\![\tau]\!]^T|$, which, by Lem. 1, is $k-1$-fold exponential in $|S|$ for τ of order k.

Syntax. Let $\mathscr{V} = \{X, Y, \dots\}$ be a set of typed variables, tacitly assumed to contain infinitely many variables for each type. The set of HO+PFP-formulas is defined by the grammar

$$\varphi ::= \mathtt{tt} \mid p(X) \mid a(X,Y) \mid X(Y_1, \dots, Y_n) \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists(X \colon \tau).\ \varphi \mid (\text{PFP}(X \colon \tau).\ \varphi)(Y_1, \dots, Y_n)$$

where $p \in \mathbf{P}$, $a \in \mathbf{A}$, and $X, Y_1, \dots, Y_n$, are variables. A formula φ is *well-formed* if the following are true for φ: (I) The variables in terms of the form $p(X)$ or $a(X,Y)$ are of type $\bullet$, and (II) in a term of the form $X(Y_1, \dots, Y_n)$ or $(\text{PFP}(X \colon \tau).\ \varphi)(Y_1, \dots, Y_n)$, the variable X has type $(\tau_1, \dots, \tau_n)$ if Y_i has type τ_i for $1 \le i \le n$. If they are not important, we omit type annotations of the form $(X \colon \tau)$, and we use compressed notation such as $\exists(X, Y, Z \colon \bullet).\ \varphi$ or $\exists(X, Y, Z \colon \tau, \tau', \tau'')$ where appropriate.

Other derived operators such as $\wedge, \rightarrow, \forall, \mathtt{ff}$ etc. can be added in the usual way. The notions of subformula, formula size etc. are also standard. Free and bound variables are defined as usual, with X being a bound variable in $(\mathrm{PFP}(X\colon \tau).\ \varphi)(Y_1,\ldots,Y_n)$. We use notation such as $\varphi(X\colon \tau, Y\colon \tau')$ etc. to communicate the names and types of the free variables of a formula, with shorthands as above used if appropriate.

We say that φ has order k if the highest order of a variable that occurs freely or as X in a formula of the form $\exists X.\ \psi$ is at most k, and the highest order of a variable X in a subformula of the form $(\mathrm{PFP}(X\colon \tau).\ \varphi)(Y_1,\ldots,Y_n)$ is at most $k+1$. We write HO^k+PFP for the collection of all formulas of order at most k.

Semantics. Let $T = (S, A, \ell)$ be an LTS. A variable assignment η is a function that assigns, to each variable $X \in \mathscr{V}$ of type τ, an element of $[\![\tau]\!]^T$. Given some X of type τ and some $f \in [\![\tau]\!]^T$, the update $\eta[X \mapsto f]$ is defined as $\eta[X \mapsto f](X) = f$ and $\eta[X \mapsto f](Y) = \eta(Y)$ if $Y \neq X$.

The semantics of a HO+PFP formula is defined as follows:

$$\begin{aligned}
&T,\eta \models \mathtt{tt}, \text{ always}\\
&T,\eta \models p(X), \text{ iff } p \in \ell(\eta(x))\\
&T,\eta \models a(X,Y), \text{ iff } \eta(X) \xrightarrow{a} \eta(Y)\\
&T,\eta \models X(Y_1,\ldots,Y_n), \text{ iff } (\eta(Y_1),\ldots,\eta(Y_n)) \in \eta(X)\\
&T,\eta \models \neg\varphi, \text{ iff } T,\eta \not\models \varphi\\
&T,\eta \models \varphi_1 \vee \varphi_2, \text{ iff } T,\eta \models \varphi_1 \text{ or } T,\eta \models \varphi_2\\
&T,\eta \models \exists(X\colon \tau).\varphi, \text{ iff there is } f \in [\![\tau]\!]^T \text{ s.t. } T,\eta[X \mapsto f] \models \varphi\\
&T,\eta \models (\mathrm{PFP}(X\colon \tau)\varphi)(Y_1,\ldots,Y_n), \text{ iff } (\eta(Y_1),\ldots,\eta(Y_n)) \in \mathrm{PFP}\ \varphi^T_\eta
\end{aligned}$$

where φ^T_η is the function that maps $g \in [\![\tau]\!]^T$ to

$$\{(g_1,\ldots,g_n) \in 2^{[\![\tau_1]\!]^T \times \cdots \times [\![\tau_n]\!]^T} \mid T,\eta[X \mapsto g, Y_1 \mapsto g_1,\ldots,Y_n \mapsto g_n] \models \varphi\} \in [\![\tau]\!]^T$$

if $\tau = (\tau_1,\ldots,\tau_n)$.

2.3 Queries

Let **P** and **A** be fixed. A (boolean) *query* (over **P** and **A**) is a function $\mathscr{Q}$ that maps, to each finite LTS $T = (S, A, \ell)$ a truth value, i.e., either true or false. Alternatively, such a boolean query is just a set of finite LTS over (over **P** and **A**), which we shall identify with $\mathscr{Q}$.

A closed HO+PFP formula φ naturally defines a query via

$$\mathscr{Q}_\varphi = \{T \mid T \models \varphi\}.$$

Conversely, queries can be decided by space-bounded Turing machines. For this, the machine receives the LTS in question as an input, and either accepts or rejects. The LTS has to be encoded into some word of the input alphabet for this. Naturally, this introduces a total order on the set of states of the LTS. It is known that e.g., the expressive power of first-order logic increases in the presence of an order (this is a classic exercise when introducing Ehrenfeucht-Fraïssé games). However, order is not an issue in our setting. The classical first-order characterization due to Abiteboul and Vianu is explicitly restricted to ordered structures, and characterizations for logics beyond existential second-order logic can be done with an order in mind, as existential second-order logic is strong enough to simply guess an order. This includes HO^k+PFP for $k \geq 2$, i.e., the topic of this paper.

Given an LTS T, let $\langle T \rangle$ be some form of polynomial encoding of T into a given input alphabet Σ, e.g., using adjacency matrices or the like. We say that a Turing machine $\mathscr{M}$ decides a query $\mathscr{Q}$ if $\mathscr{M}$ halts on any input of the form $\langle T \rangle$, where T is necessarily finite, and accepts exactly those codings where $T \in \mathscr{Q}$. A query is a k-EXPSPACE-query if there is $\mathscr{M}$ that is k-fold-exponential space bounded and decides $\mathscr{Q}$.

We now say that a logic $\mathscr{L}$ captures a complexity class $\mathscr{C}$ over a class of structures (LTS) $\mathscr{S}$ if, for each $\mathscr{L}$-query there is a $\mathscr{C}$-query that yields the same set when restricted to $\mathscr{S}$, and vice versa.

Remark 2. Non-boolean queries are quite common in e.g., the field of database theory. A d-query is then not a function that maps an LTS to a truth value, but rather one that maps an LTS and a d-tuple of states to a truth value, or, equivalently, maps every LTS to a set of d-tuples. On the logical side, one now deals with formulas with free first-order variables. We choose to stick to boolean queries here in order to avoid the extra coding required to get said free variables encoded into DTM.

2.4 Vardi's Characterization of PSPACE

We briefly sketch the classical result due to Vardi [10] that first-order logic with partial fixpoints, i.e., HO^1+PFP, captures PSPACE over the class of ordered LTS. One direction is rather straightforward since first-order queries can be evaluated in polynomial *time*, the individual stages of a partial fixpoint only take polynomial space, and the next stage can be computed from the previous one also in polynomial time. Since such a partial fixpoint either does not stabilize, or stabilizes after at most exponentially many iterations, it is sufficient to keep a counter for the number of iterations, which takes polynomially many bits if it is encoded in binary.

For the other direction, let $\mathscr{M} = (Q, \Sigma, \Gamma, \square, \delta, q_{\text{init}}, q_{\text{acc}}, q_{\text{rej}})$ be a $p(n)$-space-bounded DTM that decides a query $\mathscr{Q}$ over LTS, i.e., it accepts those $w = \langle T \rangle$ such that $T \in \mathscr{Q}$.

Since $\mathscr{M}$ is $p(n)$-space-bounded, the tape contents and the head position of each configuration of a computation of $\mathscr{M}$ on an input of length n can be represented by a number of at most $p(n)$ and a word of length $p(n)$ over the tape alphabet of $\mathscr{M}$. The proof rests on three key observations:

- In sufficiently large, ordered LTS, a configuration of $\mathscr{M}$ can be represented as a second-order relation of sufficient arity,
- the operator that computes from such a representation of a configuration its successor configuration, if it exists, can be expressed as a first-order formula, and
- the initial and accepting configurations can be pinned down using first-order logic.

The capturing result is then obtained by observing that $\mathscr{M}$ accepts its input w, derived from T, iff the partial fixpoint obtained by feeding a representation of the initial configuration into the operator mentioned above is nonempty and contains exactly a representation of the accepting configuration.

3 HO^k+PFP-Queries are in $k-1$-EXPSPACE

We begin with the simpler part of the capturing result. We will show that queries definable in HO^k+PFP can be evaluated using at most $(k-1)$-fold exponential space. This does not even need any special tricks. Alg. 1 essentially just computes the semantics of an HO^k+PFP query φ w.r.t. an LTS T and a variable evaluation η, i.e., it decides whether or not $T, \eta \models \varphi$ holds.

Algorithm 1 Evaluating HO^k+PFP queries in $(k-1)$-fold exponential space.

procedure $\textsc{Eval}(T, \eta, \varphi)$
  case φ **of**
  $\mathtt{tt}$: **return** $\mathtt{true}$
  $p(X)$: **return** $p \in \ell(\eta(X))$
  $a(X,Y)$: **return** $\eta(X) \xrightarrow{a} \eta(Y)$
  $X(Y_1,\ldots,Y_n)$: **return** $(\eta(Y_1),\ldots,\eta(Y_n)) \in \eta(X)$
  $\neg\psi$: **return** $\neg\textsc{Eval}(T,\eta,\psi)$
  $\psi_1 \vee \psi_2$: **return** $\textsc{Eval}(T,\eta,\psi_1) \vee \textsc{Eval}(T,\eta,\psi_2)$
  $\exists(X : \tau).\ \psi$:
    for all $f \in [\![\tau]\!]^T$ **do**
      if $\textsc{Eval}(T, \eta[X \mapsto f], \psi)$ **then**
        return $\mathtt{true}$
      end if
    end for
    return $\mathtt{false}$
  $(\mathrm{PFP}(X : (\tau_1,\ldots,\tau_k)).\ \psi)(Y_1,\ldots,Y_n)$:
    $f \leftarrow \emptyset$
    $cnt \leftarrow 0$
    while $cnt < |[\![(\tau_1,\ldots,\tau_k)]\!]^T|$ **do**
      $f' \leftarrow f$
      $f \leftarrow \emptyset$
      for all $(M_1,\ldots,M_k) \in [\![\tau_1]\!]^T \times \cdots \times [\![\tau_k]\!]^T$ **do**
        if $T, \eta[Y_1 \mapsto M_1,\ldots,Y_k \mapsto M_k, X \mapsto f'] \models \psi$ **then**
          $f \leftarrow f \cup \{(M_1,\ldots,M_k)\}$
        end if
      end for
      if $f = f'$ **then**
        return $(\eta(Y_1),\ldots,\eta(Y_n)) \in f$
      end if
      $cnt \leftarrow cnt + 1$
    end while
    return $\mathtt{false}$
  end case
end procedure

Theorem 3. *Let $k \geq 2$. Evaluating an HO^k+PFP query is in $(k-1)$-EXPSPACE.*

Proof. It is not hard to see that algorithm EVAL correctly evaluates an HO+PFP query, as it closely follows the semantics of HO+PFP. It remains to be seen that the space needed by this procedure is bounded by a function that is at most $(k-1)$-fold exponential in the size of the underyling $|T|$ and $|\varphi|$.

First note that the recursion depth in EVAL is bounded by $|\varphi|$. Hence, it suffices to check that the space needed within each recursive call is bounded in this way. It is only the last two cases in which this may not be obvious. So consider the case of $\varphi = \exists(X : \tau).\ \psi$. Enumerating all elements of $[\![\tau]\!]^T$ requires space for one of these elements, plus space either for a counter to abort the enumeration after all elements have been constructed, or for a second of these elements in case the enumeration can construct, from one of these elements, a uniquely determined succesor (in a lexicographic ordering for instance). In both cases, the space needed is logarithmic in $|[\![\tau]\!]^T|$ which is at most $(k-1)$-fold exponential in $|T|$ according to Lemma 1.

The argument for the last case of $\varphi = (\mathrm{PFP}(X : (\tau_1,\dots,\tau_k))\,\psi)(Y_1,\dots,Y_n)$ is similar. We write τ for $(\tau_1,\dots,\tau_k)$. Note that the order of τ may be up to $k+1$, so $|[\![\tau]\!]^T$ is k-fold exponential in S. The space needed to evaluate the partial fixpoint formula is determined by a counter with values up to $|[\![\tau]\!]^T|$ and by the two elements $f, f' \in [\![\tau]\!]^T$. Using binary coding, the space needed for the counter is logarithmic in $|[\![\tau]\!]^T$, and individual elements of $[\![\tau]\!]^T$ take $k-1$-fold exponential space, too. Hence, the space needed in this case is also at most $(k-1)$-fold exponential in $|T|$. $\square$

4 $k-1$-EXPSPACE-Queries are Expressible in HO^k+PFP

4.1 Ordering Higher-Order Relations

Since we want to encode runs of k-fold-exponentially space-bounded Turing machines into formulas of HO^{k+1}+PFP, we have to be able to encode the tape contents of the Turing machine in question. For such a space-bounded machine, the tape can be represented by a Γ-word of k-fold exponential length, where Γ is the tape alphabet of the machine in question. Hence, we have to be able to somehow represent such a large word or, in other words, we must be able to count to large numbers.

Let $p(n)$ be a polynomial, for the time being one of the form n^c for some $c \geq 2$. Let $\mathbf{A}$ contain a relation $<$, and let the types $\tau_0,\dots,\tau_k$ be the types defined via $\tau_1 = \bullet^c$, i.e. $\bullet \times \dots \times \bullet$ with c many repetitions of $\bullet$, and $\tau_{i+1} = (\tau_i)$. We define formulas $\varphi^1_<$, $\varphi^2_<$ and $\varphi^{i+1}_<$ for $i \geq 2$ via:

$$\varphi^1_<(X_1,\dots,X_c,Y_1,\dots,Y_c : \bullet) = \bigvee_{i=1}^{c} <(X_i,Y_i) \wedge \bigwedge_{j=1}^{i-1} \neg <(Y_j,X_j)$$

$$\begin{aligned}\varphi^2_<(X,Y : \tau_1) = \exists(Z_1,\dots,Z_c : \bullet).\,& Y(Z_1,\dots,Z_c) \wedge \neg X(Z_1,\dots,Z_c)\\ &\wedge \forall(Z'_1,\dots,Z'_c : \bullet).\,\big(\varphi^1_<(Z'_1,\dots,Z'_c,Z_1,\dots,Z_c)\\ &\qquad \to X(Z'_1,\dots,Z'_c) \to Y(Z'_1,\dots,Z'_c)\end{aligned}$$

$$\varphi^{i+1}_<(X,Y : \tau_{i+1}) = \exists(Z : \tau_i).\,Y(Z) \wedge \neg X(Z) \wedge \forall(Z' : \tau_i).\,\varphi^i_<(Z',Z)) \to \big(X(Z') \to Y(Z')\big)$$

Here, $\varphi^1_<$ defines a total order on $[\![\bullet^c]\!]^T$ via the lexicographical ordering induced by $<$. For $i \geq 1$, the formula $\varphi^{i+1}_<$ then totally orders $[\![\tau_{i+1}]\!]^T$ via lexicographical ordering of sets w.r.t. the membership of elements of τ_i.

Lemma 4. *Let $< \in \mathbf{A}$ and let $T = (S, A, \ell)$ be an LTS over $\mathbf{A}$ and some $\mathbf{P}$ such that $<$ is a total order on S. Let τ_k for $k \geq 1$ be defined as above. Then the following are true for all $k \geq 1$: (I) $|[\![\tau_k]\!]^T| = 2^{|S|}_{k-1}$, (II) $\varphi^k_<$ defines a total order on $[\![\tau_k]\!]^T$.*

Additionally, let $\varphi^1_=, \varphi^i_=$ and $\varphi^1_{\mathrm{succ}}, \varphi^i_{\mathrm{succ}}$ for $i > 1$ be defined as

$$\begin{aligned}
\varphi^1_=(X_1,\ldots,X_c,Y_1,\ldots,Y_c\colon \bullet) &= \neg\varphi^1_<(X_1,\ldots,X_c,Y_1,\ldots,Y_c) \wedge \neg\varphi^1_<(Y_1,\ldots,Y_c,X_1,\ldots,X_c)\\
\varphi^i_=(X,Y\colon \tau_i) &= \neg\varphi^i_<(X,Y) \wedge \neg\varphi^i_<(Y,X)\\
\varphi^1_{\mathrm{succ}}(X_1,\ldots,X_c,Y_1,\ldots,Y_c\colon \bullet) &= \varphi^1_<((X_1,\ldots,X_c,Y_1,\ldots,Y_c) \wedge \forall(Z_1,\ldots,Z_c\colon \bullet).\, \varphi^1_=(X_1,\ldots,X_c,Y_1,\ldots,Y_c)\\
&\qquad \to (\varphi^1_=(X_1,\ldots,X_c,Z_1,\ldots,Z_c) \vee \varphi^1_<(Z_1,\ldots,Z_c,X_1,\ldots,X_c))\\
\varphi^i_{\mathrm{succ}}(X,Y\colon \tau_i) &= \varphi^i_<(X,Y) \wedge \forall(Z\colon \tau_i).\, \varphi_<(Z,Y) \to \varphi_=(X,Z) \vee \varphi_<(Z,X)
\end{aligned}$$

expressing equality between elements of $[\![\tau_i]\!]^T$ or the fact that the second argument is the immediate successor of the first one w.r.t. the total order induced by $\varphi^i_<$.

Finally, for each $j \in \mathbb{N}$ and $i > 1$, define the formulas $\varphi^1_{=j}$, $\varphi^i_{=j}$ via

$$\begin{aligned}
\varphi^1_{=0}(X_1,\ldots,X_c\colon \bullet) &= \forall(Y_1,\ldots,Y_c\colon \bullet).\, \varphi^1_<(X_1,\ldots,X_c,Y_1,\ldots,Y_c) \vee \varphi^1_=(X_1,\ldots,X_c,Y_1,\ldots,Y_c)\\
\varphi^1_{=j+1}(X_1,\ldots,X_c\colon \bullet) &= \exists(Y_1,\ldots,Y_c\colon \bullet).\, \varphi^1_{=j}(Y_1,\ldots,Y_c) \wedge \varphi^1_{\mathrm{succ}}(X_1,\ldots,X_c,Y_1,\ldots,Y_c)\\
\varphi^i_{=0}(X\colon \tau_i) &= \forall(Y\colon \tau_i).\, \varphi^i_<(X,Y) \vee \varphi^i_=(X,Y)\\
\varphi^i_{=j+1}(X\colon \tau_i) &= \exists(Y\colon \tau_i).\, \varphi^i_{=j}(X,Y) \wedge \varphi^i_{\mathrm{succ}}(X,Y)
\end{aligned}$$

where $\varphi^0_{=j}$ and $\varphi^i_{=j}$ express that $(X_1,\ldots,X_c)$, resp. X is the $j+1$st element of the total order induced by $\varphi^{0]}_<$, resp. $\varphi^i_<$, if such an element exists. Clearly, the size of these formulas is linear in j.

4.2 The Reduction

Let $k \geq 1$ and let $\mathscr{M} = (Q, \Sigma, \Gamma, \Box, \delta, q_{\mathrm{init}}, q_{\mathrm{acc}}, q_{\mathrm{rej}})$ be a $2^k_{p(n)}$-space-bounded DTM that decides a query $\mathscr{Q}_{\mathscr{M}}$ over ordered LTS, i.e., it accepts those $w = \langle T \rangle$ for which $T \in \mathscr{Q}$. W.l.o.g. $p(n) = c \cdot n^{c-1}$ for some c, whence also for $n \geq c$ we have $p(n) \leq n^c$. We also assume that $\mathscr{M}$ rejects all inputs that do not encode an LTS ordered by a relation $<$.

We have to build a HO^{k+1}+PFP formula $\varphi(X_1,\ldots,X_d)$ such that $T \models \varphi$ iff $T \in \mathscr{Q}_{\mathscr{M}}$ and T is ordered by $<$.

Encoding Configurations. Let $\tau = \bullet, \tau_{k+1}, \tau_{k+1}, \bullet$. Let T be an LTS ordered by $<$ such that its state set satisfies $|S| \geq \max\{c, |Q|, |\Gamma|\}$. Hence, $|S|^c \geq p(|S|)$. W.l.o.g. Q and Γ are ordered, i.e, $Q = \{q_0, \ldots, q_{|Q|-1}\}$ and $\Gamma = \{\gamma_0, \ldots, \gamma_{|\Gamma|-1}\}$. Since S is ordered by $<$, for each $q_i \in Q$ and for each $\gamma_j \in \Gamma$, there are unique states s_q and s_γ, given as the $i+1$st, resp. $j+1$st states in the total order $<$. An element of $[\![\tau]\!]^T$ has the form (s, H, I, s') with $s, s' \in [\![\bullet]\!]^T = S$ and $H, I \in [\![\tau_{k+1}]\!]^T$.

Definition 5. Let $M \in [\![(\tau)]\!]^T$. We say that M encodes a configuration $C = (q, h, t)$ of $\mathscr{M}$ if the following are true:

1. For all $(s, H, I, s') \in M$, we have that $s = s_q$.
2. For all $(s, H, I, s'), (t, H', I', t') \in M$, we have $s = t$ and $H = H'$ and H is the $h+1$st element in the total order induced by $\varphi^{k+1}_<$.

3. For each $I \in [\![\tau_{k+1}]\!]^T$, there is exactly one tuple of the form (s,H,I,s') in M.
4. If $j \le 2_k^{p(|S|)}$, if I is the $j+1$st element in the total order induced by $\varphi_<^{k+1}$, and if $(s,H,I,s') \in M$, then $s' = s_\gamma$ for some $\gamma \in \Gamma$ and $t(j) = \gamma$.

The intuition here is the following: Since all tuples in M agree on s_q and H, this uniquely determines q and h. Moreover, since for each $I \in [\![\tau_{k+1}]\!]^T$, there is exactly one tuple of the form (s,H,I,s') in M, this defines a function $[\![\tau_{k+1}]\!]^T \to \Gamma$, and since $[\![\tau_{k+1}]\!]^T$ is linearly ordered via $\varphi_<^{k+1}$ and has cardinality $2_k^{p(|S|)}$ due to Lem. 4, this yields a function $\{0,\dots,2_k^{|S|^c}-1\} \to \Gamma$. Since $\mathscr{M}$ is $2_k^{p(n))}$-space-bounded, all configurations of a run of $\mathscr{M}$ on input $\langle T,(s_1,\dots,s_d)\rangle$ have a head position less than $2_k^{p(|S|)} \le |[\![\tau_{k+1}]\!]^T|$ and, consequently, all tape cells of such a configuration with index at least $2_k^{p(|S|)}$ must contain $\square$. Hence, such a set in $[\![(\tau)]\!]^T$ can encode any configuration $\mathscr{M}$ may enter during its run on input $\langle T,(s_1,\dots,s_d)\rangle$.

Now let $w = \langle T\rangle$. Consider the HO^{k+1}+PFP formula

$$\varphi_{\mathrm{init}}^w(Y_q\colon \bullet, H\colon \tau_{k+1}, I\colon \tau_{k+1}, Y_\gamma\colon \bullet) = \varphi_{=0}^{k+1}(H) \land Y_q = s_{q_{\mathrm{init}}} \land \bigwedge_{i=0}^{|w|-1} \varphi_{=i}^{k+1}(I) \to Y_\gamma = s_{w_i}$$
$$\land \exists(Z\colon \tau_{k+1})\ \varphi_{=|w|-1}^{k+1}(Z) \land \varphi_<^{k+1}(Z,I) \to Y_\gamma = s_\square.$$

It is of polynomial size and expresses that the tuple encoded in the variables Y_q,H,I,Y_γ is in the unique set that encodes the initial configuration of $\mathscr{M}$ on input w. We use shorthand such as $Y_q = s_{q_{\mathrm{init}}}$ to abbreviate $\varphi_{=j}^1(Y_q)$ if q_{init} is the $j+1$st state w.r.t. $<$ on S.

The Partial Fixpoint. Consider the formula

$$\begin{aligned}
\psi_{\mathrm{trans}}(X,Y_q,H,I,Y_\gamma) = {} & \exists(Y_q',H',I',Y_\gamma'\colon \bullet,\tau_{k+1},\tau_{k+1},\bullet).\\
& \exists(Y_q'',H'',I'',Y_\gamma''\colon \bullet,\tau_{k+1},\tau_{k+1},\bullet).\\
& X(Y_q',H',I',Y_\gamma') \land X(Y_q'',H'',I'',Y_\gamma'') \land \varphi_=^{k+1}(H',I'') \land \varphi_=^{k+1}(I,I')\\
& \land \neg\varphi_=^{k+1}(H,I) \to \varphi_=^1(Y_\gamma,Y_\gamma')\\
& \land \bigwedge_{(q',\gamma,q'',\gamma',L)\in\delta} Y_q' = s_{q'} \land Y_\gamma' = s_\gamma \to Y = s_{q''} \land \varphi_{\mathrm{succ}}^{k+1}(H,H') \land \varphi_=^{k+1}(H,I) \to Y_\gamma = s_{\gamma'}\\
& \land \bigwedge_{(q',\gamma,q'',\gamma',N)\in\delta} Y_q' = s_{q'} \land Y_\gamma' = s_\gamma \to Y = s_{q''} \land \varphi_=^{k+1}(H,H') \land \varphi_=^{k+1}(H,I) \to Y_\gamma = s_{\gamma'}\\
& \land \bigwedge_{(q',\gamma,q'',\gamma',R)\in\delta} Y_q' = s_{q'} \land Y_\gamma' = s_\gamma \to Y = s_{q''} \land \varphi_{\mathrm{succ}}^{k+1}(H',H) \land \varphi_=^{k+1}(H,I) \to Y_\gamma = s_{\gamma'}
\end{aligned}$$

where by abuse of syntax we write $(q',\gamma,q'',\gamma',L) \in \delta$ instead of $\delta(q',\gamma) = (q'',\gamma',L)$ etc.

Lemma 6. *Assume that $M \in [\![(\tau)]\!]^T$ with $\tau = (\bullet,\tau_{k+1},\tau_{k+1},\bullet)$ as before encodes some configuration C of the computation of $\mathscr{M}$ on input w, and assume that M' of the same type encodes the successor configuration of C.*

Let $s,s' \in S$ and $M_H,M_I \in [\![\tau_{k+1}]\!]^T$. Then

$$T,\eta[X \mapsto M, Y_q \mapsto s, H \mapsto M_H, I \mapsto I_H, Y_\gamma \mapsto s'] \models \psi_{trans} \quad \textit{iff} \quad (s,M_H,M_I,s') \in M'.$$

The intuition here is that ψ_{trans} defines the encoding of a successor of some configuration C from the encoding of C itself. The first existential quantifier requires the existence of a tuple in X that encodes the value of the tape at the same position as the new tuple will, i.e., they both must have the same third component, and the second quantifier requires the existence of a tuple that encodes the content of the tape at the head position. The third line enforces these properties. The fourth line fixes tape contents not under the head. The last three lines, separated for the ease of notation, enforce that both the state transition and the new content of the tape at the old head position obey the transition function.

We now have the required machinery to encode a computation of $\mathscr{M}$ on input w into HO^{k+1}+PFP. Let

$$\begin{aligned}\psi(X,Y_q,H,I,Y_\gamma) &= \varphi_{\text{trans}}(X,Y_q,H,I,Y_\gamma)\\ &\qquad \vee \forall(Y_q',H',I',Y_\gamma'\colon \bullet,\tau_{k+1},\tau_{k+1},\bullet).\, \neg X(Y_q',H',I',Y_\gamma') \wedge \varphi_{init}^{w}(Y_q,H,I,Y_\gamma)\\ \varphi_{\mathscr{M}} &= \varphi^{<} \wedge \exists(Y_q',H',I',Y_\gamma'\colon \bullet,\tau_{k+1},\tau_{k+1},\bullet).\, Y_q' = s_{q_{\text{acc}}} \wedge (\text{PFP}(X\colon (\tau).\,\psi)(Y_q',H',I',Y_\gamma').\end{aligned}$$

where $\varphi^{<}$ expresses that $<$ is a total order.

Lemma 7. *Let $p(n) = cn^{c-1}$ and let $\mathscr{M}$ be a $2_k^{p(n)}$-space-bounded DTM that decides a query over ordered LTS. Let Q be its state set and let Γ be its tape alphabet. Let T be an LTS ordered by $<$ and such that its state set satisfies $|S| \geq \max\{|Q|,|\Gamma|,c\}$. Let $w = \langle T\rangle$. Then*

$$T \models \varphi_{\mathscr{M},w} \text{ iff } w \in L(\mathscr{M}).$$

This follows from the previous lemmas. ψ stipulates that either X is empty, and a tuple is in its "return value" iff it is in the encoding of the initial configuration, using $\varphi_{\text{init}}^{w}$, or it defers to φ_{trans}. The formula $\varphi_{\mathscr{M}}$ then encodes the unique run of $\mathscr{M}$ on input w, by asking whether a tuple containing the accepting state is contained in the partial fixpoint of ψ. This is the case if and only if the machine halts in the accepting state, due to Lem. 6 and our observations on $\varphi_{\text{init}}^{w}$.

We omit the tedious, but standard argument that $\varphi_{\text{init}}^{w}$ can be rewritten into some φ_{init} not depending on w that internalizes the translation from T to $\langle T\rangle$.

Theorem 8. *HO^{k+1}+PFP captures k-EXPSPACE over ordered LTS for $k \geq 2$.*

One direction is by Thm. 3, the other direction is by the previous Lem. 7 plus the observation that LTS that are smaller than in the requirements of the lemma can be enumerated in a constant-size formula.

5 Conclusion

We have shown that, over ordered structures, the queries expressible in HO^{k+1}+PFP are exactly those decided by a $2_k^{p(n)}$-space-bounded DTM, i.e., that HO^{k+1}+PFP captures k-EXPSPACE over ordered structures for $k \geq 0$, extending the same result by Vardi for $k = 0$ [10].

It should be noted that the requirement that the structures in question be ordered can be removed for $k \geq 1$, as HO^2+PFP and above possess sufficient expressive power to "guess" an order, cf. Fagin's Theorem [3].

Our result has applications in descriptive complexity. Otto's Theorem [8] characterizes *bisimulation-invariant* P-queries as exactly those expressible in the polyadic modal *mu*-calculus. Contrary to Immerman's and Vardi's charaterization [5, 10] of PTIME, the crucial requirement that the LTS be ordered is absent from this result, since an order can be recuperated in the bisimulation-invariant setting. However, the result makes use of the Immerman-Vardi Theorem. We have extended this result to a characterization of bisimulation-invariant k-EXPTIME [2] using Freire and Martin's characterization of k-EXPTIME [4], i.e., their generalization of the Immerman-Vardi Theorem. The results of this paper open up a similar characterization of the bisimulation-invariant exponential-space hierarchy, following from the second author's Master's thesis [7].

References

[1] S. Abiteboul & V. Vianu (1995): *Computing with First-Order Logic*. J. Comput. Syst. Sci. 50(2), pp. 309–335, doi:10.1006/JCSS.1995.1025. Available at https://doi.org/10.1006/jcss.1995.1025.

[2] F. Bruse, D. Kronenberger & M. Lange (2022): *Capturing Bisimulation-Invariant Exponential-Time Complexity Classes*. In P. Ganty & D. Della Monica, editors: Proc. 13th Int. Symp. on Games, Automata, Logics and Formal Verification, GandALF 2022, EPTCS 370, pp. 17–33, doi:10.4204/EPTCS.370.2. Available at https://doi.org/10.4204/EPTCS.370.2.

[3] R. Fagin (1974): *Generalized First-Order Spectra and Polynomial-Time Recognizable Sets*. Complexity and Computation 7, pp. 43–73.

[4] C. M. Freire & A. T. Martins (2011): *The Descriptive Complexity of the Deterministic Exponential Time Hierarchy*. In: Proc. 5th Workshop on Logical and Semantic Frameworks with Applications, LSFA'10, 269, pp. 71–82, doi:10.1016/j.entcs.2011.03.006.

[5] N. Immerman (1986): *Relational Queries Computable in Polynomial Time*. Information and Control 68(1-3), pp. 86–104, doi:10.1016/S0019-9958(86)80029-8.

[6] N. Immerman (1987): *Languages That Capture Complexity Classes*. SIAM Journal of Computing 16(4), pp. 760–778.

[7] D. Kronenberger (2018): *Capturing Bisimulation-Invariant Complexity Classes by Polyadic Higher-Order Fixpoint Logic*. Master's thesis, University of Kassel.

[8] M. Otto (1999): *Bisimulation-invariant PTIME and higher-dimensional μ-calculus*. Theor. Comput. Sci. 224(1-2), pp. 237–265, doi:10.1016/S0304-3975(98)00314-4.

[9] L. J. Stockmeyer (1976): *The polynomial-time hierarchy*. TCS 3(1), pp. 1–22.

[10] M. Y. Vardi (1982): *The Complexity of Relational Query Languages (Extended Abstract)*. In H. R. Lewis, B. B. Simons, W. A. Burkhard & L. H. Landweber, editors: Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA, ACM, pp. 137–146, doi:10.1145/800070.802186.