Abstract. We prove that the existential fragment $\Sigma^2_1\omega$ of the third order logic $TO\omega$ captures the relational complexity class non deterministic exponential time. As a Corollary we have that relational machines can simulate third order relational machines.

1 Introduction

Relational machines (RM) were introduced in [AV,91] (there called loosely coupled generic machines) as abstract machines that compute queries to (finite) relational structures, or relational database instances (dbi’s) as functions from such structures to relations, that are generic (i.e., that preserve isomorphisms), and hence are more appropriate than Turing machines (TM) for query computation. RM’s are TM’s endowed with a relational store that hold the input structure, as well as work relations, and that can be accessed through first order logic (FO) queries (sentences) and updates (formulas with free variables). As the set of those FO formulas for a given machine is fixed, an RM can only distinguish between tuples (i.e., sequences of elements in the domain of the dbi) when the differences between them can be expressed with FO formulas with $k$ variables, where $k$ is the maximum number of variables in any formula in the finite control of the given RM. Note that the same is true for FO queries (i.e., relational calculus), or equivalently relational algebra queries.

On the other hand, it has been proved that RM’s have the same computation, or expressive power, as the (effective fragment of the) well known infinitary logic with finitely many variables $L_{\omega\omega}$ ([AVV,95]), (in the context of Finite Model Theory, i.e., with sentences interpreted by finite relational structures or database instances - dbi’s). This logic extends FO with conjunctions and disjunctions of sets of formulas of arbitrary (infinite) cardinality, while restricting the number of variables in each (infinitary) formula to be finite. This is a very important logic in descriptive complexity theory, in which among other properties, equivalence is characterized by pebble (Ehrenfeucht-Fraisse) games, and on ordered dbi’s it can express all computable queries (see [Lib,04], among others). Hence, a nice characterization of the discerning power of RM’s is also given by those games.
Consequently, $k$-ary RM’s are incapable of computing the size of the input structure though, however, they can compute its size $k$. A $k$-ary RM, for a positive integer $k$, is an RM in which the FO formulas in its finite control have at most $k$ different variables, and the size $k$ of a structure (or dbi) is the number of equivalence classes in the relation $\equiv_k$ of equality of $\text{FO}^k$ types in the set of $k$-tuples of the structure, for $1 \leq k$.

Then, it was a natural consequence to define a new notion of complexity suitable for RM’s. Relational complexity was introduced in [AV,91] as a complexity theory where the (finite relational) input structure $A$ to an algorithm is measured as its size $k$, for some $k \geq 1$, instead of the size of its encoding, as in computational complexity. Roughly, two $k$-tuples in $A$ have the same $\text{FO}^k$ types if they both satisfy in $A$ exactly the same FO formulas with up to $k$ variables, $r$ of them being free, for all $0 \leq r \leq k$. That is, if the two tuples have the same properties in the structure $A$, considering only the properties that can be expressed in $\text{FO}^k$.

In that way, relational complexity classes mirroring computational complexity classes like $P$, $NP$, $PSPACE$, $EXPTIME$ and $\text{NEXPTIME}$, etc., have been defined ([AV,91],[AVV,97]), and denoted as $P_r$, $NP_r$, $PSPACE_r$, $EXPTIME_r$, and $\text{NEXPTIME}_r$, respectively (the class $\text{NEXPTIME}_r$ is actually defined later in this article).

Beyond the study of RM’s as a model of computation for queries to relational databases, relational complexity turned out to be a theoretical framework in which we can characterize exactly the expressive power of the well known fixed point quantifiers (FP) of a wide range of types. Those quantifiers are typically added to first order logic, thus forming the so called fixed point logics, where the different types of fixed point quantifiers add to FO different kinds of iterations of first-order operators ([Lib,04], [AVV,97]).

In [AVV,97], S. Abiteboul, M. Vardi and V. Vianu introduced new fixed point quantifiers, and organized a wide range of them as either deterministic (det), non deterministic (ndet), or alternating (alt), and either inflationary (inf) or non inflationary (ninf), according to the type of iteration implied by the semantics of each such quantifier. In the same article they proved the following equivalences: $\text{det-inf-FP} = P_r$, $\text{ndet-inf-FP} = NP_r$, $\text{alt-inf-FP} = \text{det-ninf-FP} = \text{ndet-ninf-FP} = PSPACE_r$, and $\text{alt-ninf-FP} = EXPTIME_r$ (in the case of ndet FP no negation affecting an FP quantifier is allowed).

Those characterizations of relational complexity classes are actually very interesting and meaningful, given that it was already known that if we restrict the input to only ordered structures, the following equivalences with computational complexity classes hold: $\text{det-inf-FP} = P$, $\text{ndet-inf-FP} = NP$, $\text{det-ninf-FP} = \text{ndet-ninf-FP} = \text{alt-inf-FP} = PSPACE$, and $\text{alt-ninf-FP} = EXPTIME$ ([Lib,04], [AVV,97]).

Regarding the characterization of relational complexity classes with other logics, A. Dawar introduced in [Daw,98] the logic $\text{SO}^\omega$, defining it as a semantic restriction of second order logic (SO) where the valuating relations for the quantified second order variables are “unions” of complete $\text{FO}^k$ types for $r$-tuples for
some constants \(k \geq r \geq 1\), that depend on the quantifiers\(^3\). That is, the relations are closed under the relation \(\equiv_k\) of equality of \(\text{FO}^k\) types in the set of \(r\)-tuples of the structure.

In [Daw,98] it was also proved that the existential fragment of \(\Sigma^1_1\omega\), \(\Sigma^1_1\omega\), characterizes exactly the non deterministic fixed point logic (\(\text{FO} + \text{NFP}\)), and hence, by the equivalences mentioned above, it turned out that \(\Sigma^1_1\omega\) captured \(NP\), analogously to the well known relationsship \(\Sigma^1_1 = \text{NP}\) ([Fag,74]). Continuing the analogy, the characterization of the relational polynomial time hierarchy \(\text{PH}\), with full \(\text{SO}\) was stated without proof in [Daw,98], and later proved by the second author jointly with F. Ferrarotti in [FT,08].

In [AT,14], aiming to characterize higher relational complexity classes, and as a natural continuation of the study of the logic \(\text{SO}^\omega\), we defined a variation of third order logic (\(\text{TO}\)) denoted as \(\text{TO}^\omega\), under finite interpretations. We defined it as a semantic restriction of \(\text{TO}\) where the (second order) relations which form the tuples in the third order relations that valuate the quantified third order variables are closed under the relation \(\equiv_k\) as above. In [AT,14] we also introduced a variation of the non deterministic relational machine, which we denoted 3-NRM (for third order NRM), where we allow \(\text{TO}\) relations in the relational store of the machine. We defined the class \(\text{NEXPTIME}_{3,r}\) as the class of 3-NRM’s that work in time exponential in the size \(k\) (see above) of the input dbi. We then proved that the existential fragment of \(\text{TO}^\omega\), denoted \(\Sigma^2_1\omega\), captures \(\text{NEXPTIME}_{3,r}\).

In the present article, we prove a stronger result: we show that the existential fragment of \(\text{TO}^\omega\) also captures the relational complexity class \(\text{NEXPTIME}_{r}\). Then, adding the result proved in this article, we have the following picture regarding the known characterizations of relational complexity classes up to now: \(P_r = (\text{FO} + \text{det-inf-FP})\), \(NP_r = (\text{FO} + \text{ndet-inf-FP}) = \Sigma^1_1\omega\), \(\text{PH}_r = \text{SO}^\omega\), \(\text{PSPACE}_r = (\text{FO} + \text{alt-inf-FP}) = (\text{FO} + \text{det-inf-FP}) = (\text{FO} + \text{ndet-inf-FP})\), \(\text{EXPTIME}_r = (\text{FO} + \text{alt-ninf-FP})\), and \(\text{NEXPTIME}_r = \Sigma^2_1\omega\).

Then, as it turned out that \(\text{NEXPTIME}_r = \text{NEXPTIME}_{3,r}\), an interesting consequence of our result is that RM’s in their original formulation are strong enough as to simulate the existence of \(\text{TO}^\omega\) formulas in their finite control (without \(\text{TO}^\omega\) or \(\text{SO}^\omega\) quantifiers, as in 3-NRM’s in [AT,14], see below).

That is, for every 3-NRM that works in time \(\text{NEXPTIME}_{3,r}\), i.e., relational third order exponential time, in the size \(k\) of their input, there is an NRM that computes the same query, and that works in time \(\text{NEXPTIME}_r\), i.e., relational exponential exponential time in the size \(k\) of their input.

2 Preliminaries

We assume a basic knowledge of Logic and Model Theory (refer to [Lib,04]). We only consider vocabularies of the form \(\sigma = \langle R_1, \ldots, R_s \rangle\) (i.e., purely relational), where the arities of the relation symbols are \(r_1, \ldots, r_s \geq 1\), respectively. We

\(^3\) in the sense of [FPT,10] these relations are redundant relations
assume that they also contain equality. And we consider only finite $\sigma$-structures, denoted as $A = \langle A, R_1^A, \ldots, R_n^A \rangle$, where $A$ is the domain, also denoted $\text{dom}(A)$, and $R_1^A, \ldots, R_n^A$ are (second order) relations in $A^{r_1}, \ldots, A^{r_n}$, respectively. If $\gamma(x_1, \ldots, x_l)$ is a formula of some logic with free FO variables $\{x_1, \ldots, x_l\}$, for some $l \geq 1$, with $\gamma^A$ we denote the $l$-ary relation defined by $\gamma$ in $A$, i.e., the set $\{(a_1, \ldots, a_l) : a_1, \ldots, a_l \in A \land A \models \gamma(x_1, \ldots, x_l)[a_1, \ldots, a_l]\}$. For any $l$-tuple $\vec{a} = (a_1, \ldots, a_l)$ of elements in $A$, with $1 \leq l \leq k$, we define the FO$^k$ type of $\vec{a}$, denoted $\text{Type}_k(A, \vec{a})$, to be the set of FO$^k$ formulas $\varphi \in \text{FO}^k$ with free variables among $x_1, \ldots, x_l$, such that $A \models \varphi[a_1, \ldots, a_l]$. If $\tau$ is an FO$^k$ type, we say that the tuple $\vec{a}$ realizes $\tau$ in $A$, if and only if, $\tau = \text{Type}_k(A, \vec{a})$. Let $A$ and $B$ be $\sigma$-structures and let $\vec{a} \equiv_k \vec{b}$ be two $l$-tuples on $A$ and $B$ respectively, we write $(A, \vec{a}) \equiv_k (B, \vec{b})$, to denote that $\text{Type}_k(A, \vec{a}) = \text{Type}_k(B, \vec{b})$. If $A = B$, we also write $\vec{a} \equiv_k \vec{b}$. We denote as $\text{size}_k(A)$ the number of equivalence classes in $\equiv_k$ in $A$. An $l$-ary relation $R$ in $A$ is closed under $\equiv_k$ if for any two $l$-tuples $\vec{a}, \vec{b}$ in $A^l$, $\vec{a} \in R \land \vec{a} \equiv_k \vec{b} \Rightarrow \vec{b} \in R$. Let $S$ be a set, a binary relation $R$ is a pre-order on $S$ if it satisfies: 1) $\forall a \in S \quad (a, a) \in R$ (reflexive). 2) $\forall a, b, c \in S \quad (a, b) \in R \land (b, c) \in R \Rightarrow (a, c) \in R$ (transitive). 3) $\forall a, b \in S \quad (a, b) \in R \lor (b, a) \in R$ (conex). A pre-order $\preceq$ on $S$ induces an equivalence relation $\equiv$ on $S$ (i.e., $a \equiv b$ if and only if, $a \preceq b$ and $b \preceq a$), and also induces a total order over the set of equivalence classes of $\equiv$. When the equivalence classes induced by a pre-order on $k$-tuples from some structure $A$ agree with the equivalence classes of $\equiv_k$, then the pre-order establishes a total order over the FO$^k$ types for $k$-tuples which are realized on $A$. We denote by $\Sigma_m^\omega[\sigma]$ the class of formulas of the form $\exists^{k_1} Y_{t_1}^{r_1} Y_{t_1}^{k_1} \ldots \exists^{k_{tt}} Y_{t_{tt}}^{r_{tt}} k_{tt} \varphi_{\upsilon \omega} \in \Sigma_m^\omega, \phi_{\upsilon \omega}$, where the quantifiers $Q^{k_1}, \ldots, Q^{k_{tt}}$ are $\forall^{k_1}, \ldots, \forall^{k_{tt}}$, if $t$ is even, or $\exists^{k_1}, \ldots, \exists^{k_{tt}}$, if $t$ is odd, $\phi$ is an FO formula in the vocabulary $\sigma \cup \{Y_{t_1}^{r_1}, \ldots, Y_{t_{tt}}^{r_{tt}}, k_{tt}\}$, with $r_{tt} \leq k_{tt}, \ldots, r_{tt} \leq k_{tt}$, respectively. We define $\text{SO}^\omega = \bigcup_{m \geq 1} \Sigma_m^\omega$. The second order quantifier $\exists^k$ has the following semantics: let $I$ be a $\sigma$-structure; then $I \models \exists^k \varphi$ if there is an $r$-ary (second order) relation $R^{r,k}$ on $I$ that is closed under the relation $\equiv_k$ in $I$, and $(I, R) \models \varphi$.

3 The Restricted Third-Order Logic TO$^w$ and 3-NRM's

A third order relation type is a $w$-tuple $\tau = (r_1, \ldots, r_w)$ where $w, r_1, \ldots, r_w \geq 1$. In addition to the symbols of SO$^\omega$, the alphabet of TO$^w$ ([AT,14]) contains for every $k \geq 1$, a third-order quantifier $\exists^k$, and for every relation type $\tau$ such that $r_1, \ldots, r_w \leq k$ a countably infinite set of third order variables, denoted as $X^\tau_1, k, X^\tau_2, k, \ldots$, and called TO$^w$ variables. We use upper case Roman letters $X_i^r, k$ for SO$^\omega$ variables (in this article we will often drop the superindex $k$, when it is clear from the context), where $r$ is their arity, and lower case Roman letters for individual (i.e., FO) variables. Let $\sigma$ be a relational vocabulary. A TO$^w$ atomic formula of vocabulary $\sigma$, on the TO$^w$ variable $X^\tau_1, k$ is a formula of the form $X^\tau_1, k (V_1, \ldots, V_w)$, where $V_1, \ldots, V_w$ are either second order variables of the form $X^\tau_1, k$, or relation symbols in $\sigma$, and whose arities are re-
spectively \( r_1, \ldots, r_w \leq k \). Note that all the relations that form a \( \sigma \)-structure are closed under \( \equiv_k \), since \( k \) is \( \geq \) than all the arities in \( \sigma \) (see above, and Fact 9 in [FT,08]). Let \( m \geq 1 \). We denote by \( \Sigma^2_m[\sigma] \) the class of formulas of the form 
\[
\exists k_{3,11} X_{11}^{r_{11,k_{3,11}}} \ldots \exists k_{3,1,1} X_{11}^{r_{11,k_{3,1,1}}} \forall k_{3,21} X_{21}^{r_{21,k_{3,21}}} \ldots \forall k_{3,2,2} X_{22}^{r_{22,k_{3,2,2}}} \ldots Q_{k_{3,1,m_{1}}} X_{1m_{1}}^{r_{1m_{1},k_{3,1,m_{1}}}} \ldots Q_{k_{3,m_{m}},m_{m}} X_{m_{m}m_{m}}^{r_{m_{m},m_{m},k_{3,m_{m}},m_{m}}}(\psi),
\]
where for \( i, j \geq 1 \), with \( \tau_{ij} = (r_{ij,1}, \ldots, r_{ij,w}), \) it is \( r_{ij,1}, \ldots, r_{ij,w} \leq k_{3,ij} \), \( Q \) is either \( \exists^k \) or \( \forall^k \), for some \( k \), depending on whether \( m \) is odd or even, respectively, and \( \psi \) is an \( SO^\omega \) formula with the addition of \( TO^\omega \) atomic formulas. As usual, \( \forall^k X^\tau,k(\psi) \) abbreviates \( \exists^k X^\tau,k(\neg \psi) \). We define 
\[
TO^\omega = \bigcup_{m \geq 1} \Sigma^2_m.
\]

A \( TO^\omega \) relation \( R_i^{\tau,k} \) of type \( \tau \) and closed under \( \equiv_k \) on a \( \sigma \)-structure \( I \) is a set of \( \tau \)-tuples \((R_i^{r_{i1},k}, \ldots, R_i^{r_{iw},k})\) of (second order) relations on \( I \) with respective arities \( r_1, \ldots, r_w \leq k \), closed under \( \equiv_k \). The third order quantifier \( \exists^k \) has the following semantics: let \( I \) be a \( \sigma \)-structure; then \( I \models \exists^k X^\tau,k \varphi \) if there is a \( TO^\omega \) relation \( R_i^{\tau,k} \) of type \( \tau \) on \( I \) closed under the relation \( \equiv_k \) in \( I \), such that \( (I, R) \models \varphi \). Here \((I, R)\) is the third order \((\sigma \cup \{X^\tau,k\})\) structure expanding \( I \), in which \( X \) is interpreted as \( R \). Note that a valuation in this setting also assigns to each second order variable \( X^\tau,k \) a (second order) relation on \( I \) of arity \( r \) that is closed under \( \equiv_k \) in \( I \), and to each third order variable \( X^\tau,k \) a third order relation \( R_i^{\tau,k} \) on \( I \) of type \( \tau \), closed under \( \equiv_k \) in \( I \). We don’t allow free second or third order variables in the logics \( SO^\omega \) and \( TO^\omega \). Note that allowing elements (from the domain of the structure) in a third order relation type would change the semantics of \( TO^\omega \), since we could use a third order relation of such type to simulate a second order relation not closed under \( \equiv_k \). See [AT,14] for an example of a non trivial query in \( \Sigma^2_1 \).

A third order non-deterministic relational machine ([AT,14]), denoted as 3-NRM, of arity \( k \), for \( k \geq 1 \), is a 11-tuple \((Q, \Sigma, \delta, q_0, b, F, \sigma, \tau, T, \Omega, \Phi)\) where: \( Q \) is the finite set of internal states; \( q_0 \in Q \) is the initial state; \( \Sigma \) is the finite tape alphabet; \( b \in \Sigma \) is the symbol denoting blank; \( F \subseteq Q \) is the set of accepting states; \( \tau \) is the finite vocabulary of the \( rs \) (its relational store), with finitely many \( TO^\omega \) relation symbols \( R_i^{\tau,k'} \) of any arbitrary type \( \tau_i = (r_{i1}, \ldots, r_{iw}) \), with \( 1 \leq r_{i1}, \ldots, r_{iw} \leq k' = k \), and finitely many \( SO^\omega \) relation symbols \( R_i^{r',k''} \) of arities \( r_i \leq k'' = k \); \( T \in \tau \) is the output relation; \( \sigma \) is the vocabulary of the input structure; \( \Omega \) is a finite set of \( TO^\omega \) formulas with up to \( k \) \( FO \) variables, with \( noSO^\omega \) or \( TO^\omega \) quantifiers, and with no free variables of any order (i.e., all the \( SO^\omega \) and \( TO^\omega \) relation symbols are in \( \tau \)); \( \Phi \) is a finite set of \( TO^\omega \) formulas with up to \( k \) \( FO \) variables, that are not sentences, with \( noSO^\omega \) or \( TO^\omega \) quantifiers, and where the free variables are either \( allFO \) variables, or \( allSO^\omega \) variables; \( \delta : Q \times \Sigma \times \Omega \to P(\Sigma \times Q \times \{R, L\} \times \Phi \times \tau) \) is the transition function. In any pair in \( \delta \), if \( \varphi \), \( S \) occur in the 5-tuple of its second component, for \( \Phi \) and \( \tau \), then either \( S \) is a \( TO^\omega \) relation symbol \( R_i^{\tau,k'} \) in \( rs \) and \( \varphi \) has \( |\tau_i| \) \( SO^\omega \) free variables \( X_1^{r_1,k''}, \ldots, X_{|\tau_i|}^{r_{|\tau_i|},k'''} \) with arities according to \( \tau_i \), and \( 1 \leq r_1, \ldots, r_{|\tau_i|} \leq k''' = k' = k \), or \( S \) is an \( SO^\omega \) relation symbol \( R_i^{r',k''} \) in \( rs \) and \( \varphi \) has \( 1 \leq r_1 \leq k'' = k \) \( FO \) free variables. At any stage of the computation of a 3-NRM on an input \( \sigma \)-structure \( I \), there is one relation in its \( rs \) of the corresponding relation type
(or arity) in \(I\) for each relation symbol in \(\tau\), so that in each transition there is a (finite) \(\tau\)-structure \(A\) in the \(rs\), which we can query and/or update through the formulas in \(\Omega\) and \(\Phi\), respectively, and a finite \(\Sigma\) string in its tape, which we can access as in Turing machines. The concept of computation is analogous to that in the Turing machine. We define the complexity class \(\text{NEXPTIME}_{3,r}\) as the class of the relational languages or Boolean queries (i.e., sets of finite structures of a given relational vocabulary, closed under isomorphisms) that are decidable by 3-NRM machines of some arity \(k'\), that work in non deterministic exponential time in the number of equivalence classes in \(\equiv_{k'}\) of the input structure. In symbols:

\[
\text{NEXPTIME}_{3,r} = \bigcup_{c \in \mathbb{N}} \text{NTIME}_{3,r}(2^c(\text{size}_{k'}))
\]

(as usual, this notation does not mean that the arity of the 3-NRM must be \(k\)).

A non-deterministic relational machine, i.e., an NRM in its classical formulation, denoted as NRM, of arity \(k\), for \(k \geq 1\), is a 11-tuple as above, where the formulas in \(\Omega\) and \(\Phi\) are FO formulas with up to \(k\) FO variables, in the vocabulary \(\tau\), and where all the relations in the relational store are second or higher order relations of arity at most \(k\). The relational complexity class \(\text{NEXPTIME}_r\) is the class of the relational languages or Boolean queries that are decidable by NRM machines of some arity \(k'\), that work in non deterministic exponential time in the number of equivalence classes in \(\equiv_{k'}\) of the input structure. In symbols:

\[
\text{NEXPTIME}_r = \bigcup_{c \in \mathbb{N}} \text{NTIME}_r(2^c(\text{size}_{k'})).
\]

In [AT,14] we proved the following results:

**Theorem 1.** ([AT,14]) \(\text{NEXPTIME}_{3,r} \subseteq \Sigma^2_{1,\omega}\). That is, given a 3-NRM \(M\) in \(\text{NTIME}_{3,r}(2^c(\text{size}_{k'}))\), for some positive integer \(c\), and with input vocabulary \(\sigma\) that computes a Boolean query \(q\) we can build a formula \(\varphi_M \in \Sigma^2_{1,\omega}\) such that, for every \(\sigma\)-structure \(I\), \(M\) accepts \(I\) iff \(I \models \varphi_M\).

**Theorem 2.** ([AT,14]) \(\Sigma^2_{1,\omega} \subseteq \text{NEXPTIME}_{3,r}\). That is, every class of relational structures definable in \(\Sigma^2_{1,\omega}\) is in \(\bigcup_{c \in \mathbb{N}} \text{NTIME}_{3,r}(2^c(\text{size}_{k'}))\).

### 4 Existential \(\text{TO}^\omega\) captures \(\text{NEXPTIME}_r\)

**Corollary 3** \(\text{NEXPTIME}_r \subseteq \Sigma^2_{1,\omega}\). That is, given an NRM \(M\) that works in \(\text{NTIME}_r(2^c(\text{size}_{k'}))\), for some positive integer \(c\), and with input vocabulary \(\sigma\) that computes a Boolean query \(q\) we can build a formula \(\varphi_M \in \Sigma^2_{1,\omega}\) such that, for every \(\sigma\)-structure \(I\), \(M\) accepts \(I\) iff \(I \models \varphi_M\).

**Proof.** This is a consequence of Theorem 1 by the following two immediate facts: 1) an NRM is a special case of a 3-NRM, with no third order relations in its \(rs\), and 2) an NRM \(M\) is in \(\text{NEXPTIME}_r\) iff \(M\), as a 3-NRM, it is in \(\text{NEXPTIME}_{3,r}\). \(\square\)

**Theorem 4.** \(\Sigma^2_{1,\omega} \subseteq \text{NEXPTIME}_r\). That is, every class of relational structures definable in \(\Sigma^2_{1,\omega}\) is in \(\bigcup_{c \in \mathbb{N}} \text{NTIME}_r(2^c(\text{size}_{k'}))\).
Proof. Let \( \sigma \) be a relational vocabulary, let \( \varphi \) be a \( \Sigma^{2,\omega}_1[\sigma] \) sentence of the form 
\[ \exists x_1, \ldots, x_s \exists^\forall_{t_1,\ldots,t_s} (\psi), \]
where \( \psi \) is a \( \Sigma^{1,\omega}_t \) formula, for some \( t \geq 1 \), with atomic \( TO^\omega \) formulas formed with the \( TO^\omega \) variables \( X_1, \ldots, X_s \). For the sake of a simpler presentation we assume w.l.o.g. that for \( 1 \leq i \leq s \) the type of the relation \( X_i \) is \( \tau_i = (r_{3,i}, \ldots, r_{3,i}) \) of cardinality \( r_{3,i} \), with \( r_{3,i} \leq k_{3,i}. \)

Suppose the \( \Sigma^{1,\omega}_t \) formula \( \psi \) is of the form 
\[ \exists x_1, \ldots, x_s \exists^\forall_{r_{2,1}, \ldots, r_{2,1}} (\phi), \]
where the quantifiers \( Q^{r_{2,1}} \), \( \ldots, Q^{r_{2,1}} \), are \( \forall^{r_{2,1}} \), \( \ldots, \forall^{r_{2,1}} \), if \( t \) is even, or \( \exists^{r_{2,1}} \), \( \ldots, \exists^{r_{2,1}} \), if \( t \) is odd, \( \phi \) is an \( FO \) formula in the vocabulary \( \sigma \cup \{I_{r_{1,1}}, I_{r_{2,1}}, \ldots, I_{r_{2,1}} \} \), with atomic \( TO^\omega \) formulas, and \( r_{2,1} \leq k_{2,1}, \ldots, r_{2,1} \leq k_{2,1}, \) respectively. We now build an NRM \( M_\sigma \) which accepts a given \( \sigma \) structure \( I \) iff \( I \models \varphi \). It is known that for every \( \sigma \), and every \( k \geq 1 \), a formula \( \gamma (\bar{x}, \bar{y}) \) with \( |\bar{x}| \geq 2k \) variables of the fixed point logic \((FO + LFP)\) can be built s.t. on any \( \sigma \) structure \( J \), \( \gamma \) defines a pre-order \( \preceq_k \) in the set of \( k \)-tuples of \( J \), whose induced equivalence relation is \( \equiv_k \) (see T.11.20 in [Lib.04]). On the other hand, it is known that \((FO + LFP)\) captures relational polynomial time \( P_r \) ([AVV.97]).

Hence, an \( RM \) \( M_\sigma \) of some arity \( k' \geq 2k \) can be built, that constructs, on input \( J \), the pre-order \( \preceq_k \) in \( J \), in time polynomial in \( size_\varphi (J) \). We define the arity of \( M_\sigma \) as \( k = \max (\{k_{3,1}, \ldots, k_{3,1}, k_{2,1}, \ldots, k_{2,1} \}) \), where the \( \langle i \rangle \)'s are the arities of the \( RM \)'s \( M_{\preceq_{k_{3,1}}, \ldots, M_{\preceq_{k_{2,1}}}} \), respectively.

Let \( I \) be the input structure. \( M_\sigma \) works as follows: 1) \( M_\sigma \) simulates the \( RM \)'s \( M_{\preceq_{k_{3,1}}, \ldots, M_{\preceq_{k_{2,1}}}} \), to build the pre-orders \( \preceq_{k_{3,1}}, \ldots, \preceq_{k_{2,1}} \), respectively. \( M_\sigma \) builds those pre-orders in time polynomial in \( size_{\varphi_{k_{3,1}} (I)}, \ldots, size_{k_{2,1}} (I) \), respectively. As all these arities are \( \leq k \) (see above), that time is also polynomial in \( size_\varphi (I) \) (see [FT.08]). 2) By stepping through the equivalence classes of the relation \( \equiv_{k_{3,1}} \), in the order given by \( \preceq_{k_{3,1}} \), \( M_\sigma \) computes \( size_{k_{3,1}} (I) \), and the same process is followed to compute \( size_{k_{3,2}} (I), \ldots, size_{k_{2,1}} (I) \) by using the equivalence relations \( \equiv_{k_{3,2}}, \ldots, \equiv_{k_{2,1}} \) and the pre-orders \( \preceq_{k_{3,2}}, \ldots, \preceq_{k_{2,1}} \), respectively (recall that the pre-orders \( \preceq_{k_{3,1}}, \ldots, \preceq_{k_{2,1}} \) induce total orders in the equivalence classes of the corresponding equivalence relations \( \equiv_{k_{3,1}}, \ldots, \equiv_{k_{2,1}} \)).

Note that by the choice of \( k \), all these computations are done by \( M_\sigma \) in time polynomial in \( size_k (I) \). 3) \( M_\sigma \) needs to guess the \( TO^\omega \) relations \( S^\tau_{i}, \ldots, S^\tau_{i} \), as interpretations of the \( TO^\omega \) variables \( X^\tau_{i}, \ldots, X^\tau_{i} \), respectively. Each \( S^\tau_{i} \) is a set of \( r_{3,1} \)-tuples of \( r_{3,1} \)-ary \((SO)\) relations closed under \( \equiv_{k_{3,1}} \). To represent \( S^\tau_{i} \) we use three sorts of bit strings as follows: a) each bit string of sort \( b^3_{R_{r_{3,1}, k_{3,1}}} \) of size \( size_{k_{3,1}} (I) \) represents one of the possible \( r_{3,1} \)-ary \((SO)\) relations on \( I \), closed under \( \equiv_{k_{3,1}} \); note that each bit represents one equivalence class in \( \equiv_{k_{3,1}} \), following from left to right the total order induced by \( \preceq_{k_{3,1}} \); b) each bit string of sort \( b^2_{R_{r_{3,1}, k_{3,1}}} \) of size \( r_{3,1} \cdot size_{k_{3,1}} (I) \) represents one of the possible \( r_{3,1} \)-tuples of \( r_{3,1} \)-ary \((SO)\) relations on \( I \), closed under \( \equiv_{k_{3,1}} \); c) each bit string of sort \( b^1_{S^\tau_{i}, k_{3,1}} \) of size \( 2^{\tau_{i} \cdot size_{k_{3,1}} (I)} \) represents one of the possible sets of \( r_{3,1} \)-tuples of \( r_{3,1} \)-ary \((SO)\) relations on \( I \), closed under \( \equiv_{k_{3,1}} \), i.e., one of the possible \( TO^\omega \) relations on \( I \) of type \( \tau_i \), closed under \( \equiv_{k_{3,1}} \). Let \( b \) be a bit string of sort \( b^1 \).
Each bit in $b$ represents one of the possible bit strings of sort $b^2_{\Sigma_3,i,k_3,i}(I)$. The leftmost bit in $b$ represents a bit string of type $b^2_{\Sigma_3,i,k_3,i}$ that has all its bits 0, i.e., it is the bit string that corresponds to the $r_{3,i}$-tuple formed by $r_{3,i}$ empty $r_{3,i}$-ary relations. The following bits in $b$ represent the bit strings of sort $b^2_{\Sigma_3,i,k_3,i}$ that correspond to the order in all the possible bit strings of sort $b^2_{\Sigma_3,i,k_3,i}$ according to their binary value. And so on, up to the rightmost bit in $b$, which represents a bit string of sort $b^2_{\Sigma_3,i,k_3,i}$ that has all its bits 1 (i.e., it is the bit string that corresponds to the $r_{3,i}$-tuple formed by $r_{3,i}$ copies of the $r_{3,i}$-ary relation that has the $r_{3,i}$-tuples in all the equivalence classes in the relation $\equiv_{k_3,i}$). Then, $M_\varphi$ guesses $s$ bit strings of sort $b^1_{\Sigma_3,i}$ of size $2^{3,i\cdot size_{k_3,i}(I)}$, one for each one of the relations $\Sigma^3_i$. Note that this is done in time $2^{c\cdot size_{k_3,i}(I)}$, and hence also in time $2^{d\cdot size_{k_3,i}(I)}$, since $k_{3,i} \leq k$ (see above), for some constants $c, d$.

4): Regarding the SO$^\omega$ variables quantified in the $\Sigma^1_3$ formula $\psi$, to interpret each of them we build all the possible SO$^\omega$ relations of the corresponding arity and closed under the corresponding equivalence class in the the rs of $M_\varphi$. We build those relations by stepping in the equivalence classes of tuples $\equiv_{k_2,ij}$ according to the total orders induced by the corresponding pre-orders $\preceq_{k_2,ij}$. The details on how to do that are equal to the algorithm used in [FT, 08] to prove $\Sigma^1_3 \subseteq \text{NTIME}(\text{size}_k^c)$. Note that we can afford to do that because for each variable $Y_{ij}^{r_{2,ij},k_{2,ij}}$ the number of such relations is bounded by $2^{d\cdot size_{k_2,ij}(I)}$, and hence also by $2^{d\cdot size_{k_3}(I)}$, since $k_{2,ij} \leq k$ (see above), for some constant $d$ that depends on the arity. Then, for each SO$^\omega$ variable $Y_{ij}^{r_{2,ij},k_{2,ij}}$ we will require that either for all the generated relations, or for at least one of them, depending on the corresponding quantifier being $\forall$ or $\exists$, respectively, the formula $\phi$ is true. 5): Evaluation of $\phi$: Recall that $\phi$ is an FO formula with atomic TO$^\omega$ formulas. To evaluate $\phi$ we consider the syntax tree of $\phi$, $T_\phi$, and evaluate one node of it at a time in the finite control of $M_\varphi$, in a bottom up direction. To that end, for every node $\alpha$ in $T_\phi$, that represents a sub-formula with $r \geq 1$ free FO variables, we define in the rs an $r$-ary relation variable $R_\alpha$. And for every node $\alpha$ in $T_\phi$, that represents a sub-formula with no free FO variables, we define in the rs a 1-ary relation variable $B_\alpha$ that represents a Boolean variable, which we interpret as True if $B_\alpha = \text{dom}(I)$, and as False if $B_\alpha = \emptyset$. Note that all the SO$^\omega$ relations that appear in the nodes in $T_\phi$ are in the rs of $M_\varphi$. Every node in $T_\phi$ is of one of the following kinds: i) an atomic FO formula with a relation symbol either in $\sigma$ or quantified by an SO$^\omega$ quantifier in $\psi$, ii) a $\lor$ connective, iii) a $\land$ connective, iv) a $\neg$ connective, v) an existential FO quantifier, or vi) an atomic TO$^\omega$ formula with a relation symbol quantified by a TO$^\omega$ quantifier in $\varphi$. We omit the details on how to evaluate the nodes of the first 5 kinds, since they are straightforward, and focus on the nodes that correspond to atomic TO$^\omega$ formulas. Suppose a given node $\alpha$ in $T_\phi$ corresponds to the sub-formula $\chi^r_{\tau_i,k_3,i}(V_{1}^{r_3,i,k_3,i}, \ldots, V_{r_3,i}^{r_3,i,k_3,i})$ with $\tau_i = (r_{3,i}, \ldots, r_{3,i})$ of cardinality $r_{3,i}$, with $r_{3,i} \leq k_{3,i}$ as stated in the beginning of the proof, and where $V_{1}^{r_3,i,k_3,i}, \ldots, V_{r_3,i}^{r_3,i,k_3,i}$ are either relation symbols in $\sigma$ or quantified by an SO$^\omega$ quantifier in $\psi$. We check whether or not the $r_{3,i}$-tuple
of relations \((V_{r_3,1,k_3,i}, \ldots, V_{r_3,1,k_3,i})\) is in the TO\(^\omega\) relation \(S_{i}^{r_3,k_3,i}\) guessed above for the variable \(\chi_{i}^{r_3,k_3,i}\), using the (guessed) bit string \(b^1_{S_{i}^{r_3,k_3,i}}\) that represents \(S_{i}^{r_3,k_3,i}\), with the following algorithm, that clearly runs in time \(2^{d \cdot \text{size}_k,i}\) (1), and hence also in time \(2^{d \cdot \text{size}_k,i}\) (1), since \(k_{3,i} \leq k\) (see above), for some constants \(c, d\):

- \(B_\alpha \leftarrow \emptyset\) (i.e., \(B_\alpha \leftarrow \text{FALSE}\));
- for all bit strings of sort \(b^2_{R_{r_3,i,k_3,i}}\) \((a_1, \ldots, a_{2^r_{r_3,i,k_3,i}})\), counting in binary, varying \(n\) from 1 through \(2^3_{r_3,i,\text{size}_{k_3,i}}\) (i.e., for all \(r_3,i\)-ary (SO) relations closed under \(\equiv_{k_3,i}\));
  - for \(j = 1\) through \(r_3,i\) (i.e., the \(j\)-th component in the tuple of (SO) rel.);
    - \(S_{r_3,i,k_3,i} \leftarrow \emptyset\);
    - for \(l = 1\) through \(\text{size}_{k_3,i}\) (I) (i.e., bit \(l\) in bit substring of sort \(b^3_{R_{r_3,i,k_3,i}}\));
      - if bit \(m\) of bit string \(a_m\) is 1, where \(m = (j-1) \cdot \text{size}_{k_3,i}(I) + l\), (i.e., bit \(m\) in a bit string of sort \(b^2_{R_{r_3,i,k_3,i}}\))
        - add to \(S_{r_3,i,k_3,i}\) the \(l\)-th equivalence class in \(\equiv_{k_3,i}\), according to pre-order \(\leq_{k_3,i}\) (i.e., all the \(r_3,i\)-tuples of elements in that class);
  - end \(l\);
  - end \(j\);
- if bit \(n\) in bit string \(b^1_{S_{r_3,i,k_3,i}} = 1\)
  - if \((V_{r_3,1,k_3,i} = S_{r_3,1,k_3,i} \land \ldots \land V_{r_3,1,k_3,i} = S_{r_3,1,k_3,i})\)
    - \(B_\alpha \leftarrow \text{dom}(I)\) (i.e., \(B_\alpha \leftarrow \text{TRUE}\));
- end all; \(\Box\)

5 Conclusions

From Theorems 1, 2, 4, and Corollary 3, we have the following result:

**Corollary 5** Let \(M_3\) be a 3-NRM that works in \(NTIME_{3,r}(2^{c \cdot \text{size}_k})\), for some positive integer \(c\), that computes a Boolean query \(q\). Then, there is a NRM \(M_2\) that works in \(NTIME_{r}(2^{d \cdot \text{size}_k})\), for some positive integer \(d\), that also computes \(q\). \(\Box\)

This is very interesting, since in the general case it is much easier to define an NRM using TO relations in its rs, and TO formulas to access it, than restricting the machine to SO relations in its rs, and SO formulas. Then, to prove that a given query is computable by an NRM it is enough with showing that it can be computed by a 3-NRM. Note however, that we think that we still need 3-NRM’s as well as the third order relational complexity class \(NEXPTIME_{3,r}\), if we need to work with oracle NRM’s with third order relations, since as the oracle cannot access the tape of the base machine (see [FT,08]), there seems to be no way to pass the bit strings that represent TO relations from the base to the oracle.

Recall that it has been proved that RM’s have the same computation, or expressive power, as the (effective fragment of the) well known infinitary logic
with finitely many variables $L_{\omega}^\omega$ ([AVV,95]). On the other hand, analogously to the well known result that states that the computation power of deterministic and non deterministic Turing machines is the same, it is straightforward to see that any NRM $M_n$ can be simulated by a (deterministic) RM $M_d$ working in relational time exponentially higher, just by checking in $M_d$ all possible transitions instead of guessing one in each non deterministic step of the transition relation of $M_n$. Then, the following is immediate:

**Corollary 6** $\Sigma_1^{2,\omega} \subseteq \{\text{effective fragment of}\} \ L_{\omega}^\omega$.

Finally, in [GT,10], the logic $SO^F$ was introduced and defined as a semantic restriction of SO where the valuating $r$-ary relations for the quantified SO variables are closed under the relation $\equiv_F$ of equality of FO types in the set of $r$-tuples of the structure. It was shown there that its existential fragment $\Sigma_1^{1,F}$ is not included in $L_{\omega}^\omega$, as opposite to $\Sigma_1^{1,\omega}$ which is. Then, we have the following result:

**Corollary 7** $\Sigma_1^{1,F} \subsetneq \Sigma_1^{2,\omega}$.

**References**