Capturing relational NEXPTIME with a Fragment of Existential Third Order Logic

Turull-Torres José Maria[1,2]

[1] Depto. de Ingeniería e Investigaciones Tecnológicas
Universidad Nacional de La Matanza

[2] and Massey University, New Zealand
J.M.Turull@massey.ac.nz

Abstract. We prove that the existential fragment $\Sigma_1^{2,\omega}$ of the third order logic TO^ω 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 $\mathcal{L}^\omega_{\infty\omega}$ ([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-Fraissee) 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 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 $\mathcal{A}$ 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 $\mathcal{A}$ have the same FO^k types if they both satisfy in $\mathcal{A}$ 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 $\mathcal{A}$, considering only the properties that can be expressed in FO^k. In that way, relational complexity classes mirroring computational complexity classes like P, NP, $PSPACE$, $EXPTIME$ and $NEXPTIME$, etc., have been defined ([AV,91],[AVV,97]), and denoted as P_r, NP_r, $PSPACE_r$, $EXPTIME_r$ and $NEXPTIME_r$, respectively (the class $NEXPTIME_r$ 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: det-inf-FP = P_r, ndet-inf-FP = NP_r, alt-inf-FP = det-ninf-FP = ndet-ninf-FP = $PSPACE_r$, and 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: det-inf-FP = P, ndet-inf-FP = NP, det-ninf-FP = ndet-ninf-FP = alt-inf-FP = $PSPACE$, and 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 SO^ω, 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 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 FO^k types in the set of r-tuples of the structure.

In [Daw,98] it was also proved that the existential fragment of SO^ω, $\Sigma_1^{1,\omega}$, characterizes exactly the non deterministic fixed point logic $(\mathrm{FO} + NFP)$, and hence, by the equivalences mentioned above, it turned out that $\Sigma_1^{1,\omega}$ captured NP_r, analogously to the well known relationssip $\Sigma_1^1 = NP$ ([Fag,74]). Continuing the analogy, the characterization of the relational polynomial time hierarchy PH_r with *full* SO^ω 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 SO^ω, we defined a variation of third order logic (TO) denoted as TO^ω, under finite interpretations. We defined it as a semantic restriction of 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 TO relations in the relational store of the machine. We defined the class $\mathrm{NEXPTIME}_{3,r}$ 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 TO^ω, denoted $\Sigma_1^{2,\omega}$, captures $NEXPTIME_{3,r}$.

In the present article, we prove a stronger result: we show that the existential fragment of TO^ω *also captures* the relational complexity class $NEXPTIME_r$. Then, adding the result proved in this article, we have the following picture regarding the known characterizations of relational complexity classes up to now: $P_r = (\mathrm{FO} + \text{ det-inf-FP })$, $NP_r = (\mathrm{FO} + \text{ndet-inf-FP }) = \Sigma_1^{1,\omega}$, $PH_r = \mathrm{SO}^\omega$, $PSPACE_r = (\mathrm{FO} + \text{ alt-inf-FP }) = (\mathrm{FO} + \text{ det-ninf-FP }) = (\mathrm{FO} + \text{ ndet-ninf-FP })$, $EXPTIME_r = (\mathrm{FO} + \text{ alt-ninf-FP })$, and $NEXPTIME_r = \Sigma_1^{2,\omega}$.

Then, as it turned out that $NEXPTIME_r = NEXPTIME_{3,r}$, an interesting consequence of our result is that RM's in their original formulation are strong enough as to *simulate* the existence of TO relations in their relational store and, hence, to also *simulate* the existence of TO^ω formulas in their finite control (without TO^ω or SO^ω quantifiers, as in 3-NRM's in [AT,14], see below).

That is, for every 3-NRM that works in time $NEXPTIME_{3,r}$, 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 $NEXPTIME_r$, i.e., relational 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* σ-structures, denoted as $\mathbf{A} = \langle A, R_1^{\mathbf{A}}, \ldots, R_s^{\mathbf{A}} \rangle$, where A is the domain, also denoted $dom(\mathbf{A})$, and $R_1^{\mathbf{A}}, \ldots, R_s^{\mathbf{A}}$ are (second order) relations in $A^{r_1}, \ldots, A^{r_s}$, 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 $\gamma^{\mathbf{A}}$ we denote the l-ary relation defined by γ in $\mathbf{A}$, i.e., the set $\{(a_1, \ldots, a_l) : a_1, \ldots, a_l \in A \wedge \mathbf{A} \models \gamma(x_1, \ldots, x_l)[a_1, \ldots, a_l]\}$. For any l-tuple $\bar{a} = (a_1, \ldots, a_l)$ of elements in A, with $1 \leq l \leq k$, we define the FO^k type of $\bar{a}$, denoted $Type_k(\mathbf{A}, \bar{a})$, to be the set of FO^k formulas $\varphi \in FO^k$ with free variables among $x_1, \ldots, x_l$, such that $\mathbf{A} \models \varphi[a_1, \ldots, a_l]$. If τ is an FO^k type, we say that the tuple $\bar{a}$ *realizes* τ in $\mathbf{A}$, if and only if, $\tau = Type_k(\mathbf{A}, \bar{a})$. Let $\mathbf{A}$ and $\mathbf{B}$ be σ-structures and let $\bar{a}$ and $\bar{b}$ be two l-tuples on $\mathbf{A}$ and $\mathbf{B}$ respectively, we write $(\mathbf{A}, \bar{a}) \equiv_k (\mathbf{B}, \bar{b})$, to denote that $Type_k(\mathbf{A}, \bar{a}) = Type_k(\mathbf{B}, \bar{b})$. If $\mathbf{A} = \mathbf{B}$, we also write $\bar{a} \equiv_k \bar{b}$. We denote as $size_k(\mathbf{A})$ the number of equivalence classes in $\equiv_k$ in $\mathbf{A}$. An l-ary relation R in $\mathbf{A}$ is *closed under* $\equiv_k$ if for any two l-tuples $\bar{a}$, $\bar{b}$ in A^l, $\bar{a} \in R \wedge \bar{a} \equiv_k \bar{b} \Rightarrow \bar{b} \in R$. 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) $\forall a, b, c \in S \quad (a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R$ (transitive). 3) $\forall a, b \in S$ $(a, b) \in R \vee (b, a) \in R$ (conex). A *pre-order* $\preceq$ on S induces an equivalence relation $\equiv$ on S (i.e., $a \equiv b \Leftrightarrow a \preceq b \wedge b \preceq a$), 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 $\mathbf{A}$ 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 $\mathbf{A}$. We denote by $\Sigma_m^{1,\omega}[\sigma]$ the class of formulas of the form $\exists^{k_{11}} Y_{11}^{r_{11},k_{11}} \ldots \exists^{k_{1l_1}} Y_{1l_1}^{r_{1l_1},k_{1l_1}} \forall^{k_{21}} Y_{21}^{r_{21},k_{21}} \ldots \forall^{k_{2l_2}} Y_{2l_2}^{r_{2l_2},k_{2l_2}} \ldots Q^{k_{t1}} Y_{t1}^{r_{t1},k_{t1}} \ldots$ $Q^{k_{tl_t}} Y_{tl_t}^{r_{tl_t},k_{tl_t}}(\phi)$, where the quantifiers $Q^{k_{t1}}, \ldots, Q^{k_{tl_t}}$ are $\forall^{k_{t1}}, \ldots, \forall^{k_{tl_t}}$, if t is even, or $\exists^{k_{t1}}, \ldots, \exists^{k_{tl_t}}$, if t is odd, ϕ is an FO formula in the vocabulary $\sigma \cup \{Y_{11}^{r_{11},k_{11}}, \ldots, Y_{tl_t}^{r_{tl_t},k_{tl_t}}\}$, with $r_{11} \leq k_{11}$, ..., $r_{tl_t} \leq k_{tl_t}$, respectively. We define $\mathrm{SO}^\omega = \bigcup_{m \geq 1} \Sigma_m^{1,\omega}$. The second order quantifier $\exists^k$ has the following semantics: let $\mathbf{I}$ be a σ-structure; then $\mathbf{I} \models \exists^k Y^{r,k} \varphi$ if there is an r-ary (second order) relation $R^{r,k}$ on I that is closed under the relation $\equiv_k$ in $\mathbf{I}$, and $(\mathbf{I}, R) \models \varphi$.

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$^\omega$ ([AT,14]) contains for every $k \geq 1$, a *third-order quantifier* $\exists^k$, and for every relation type τ such that $r_1, \ldots, r_w \leq k$ a countably infinite set of third order variables, denoted as $\mathcal{X}_1^{\tau,k}, \mathcal{X}_2^{\tau,k}, \ldots$, and called TO$^\omega$ *variables*. We use upper case Roman letters $X_i^{r,k}$ for SO^ω 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 σ be a relational vocabulary. A TO$^\omega$ *atomic formula* of vocabulary σ, on the TO$^\omega$ variable $\mathcal{X}^{\tau,k}$ is a formula of the form $\mathcal{X}^{\tau,k}(V_1, \ldots, V_w)$, where $V_1, \ldots, V_w$ are either second order variables of the form $X_i^{r_i,k}$, or relation symbols in σ, and whose arities are respectively $r_1, \dots, r_w \leq k$. Note that all the relations that form a σ-structure are closed under $\equiv_k$, since k is $\geq$ than all the arities in σ (see above, and Fact 9 in [FT,08]). Let $m \geq 1$. We denote by $\Sigma^{2,\omega}_m[\sigma]$ the class of formulas of the form $\exists^{k_{3,11}} \mathcal{X}^{\tau_{11},k_{3,11}}_{11} \dots \exists^{k_{3,1s_1}} \mathcal{X}^{\tau_{1s_1},k_{3,1s_1}}_{1s_1} \forall^{k_{3,21}} \mathcal{X}^{\tau_{21},k_{3,21}}_{21} \dots \forall^{k_{3,2s_2}} \mathcal{X}^{\tau_{2s_2},k_{3,2s_2}}_{2s_2} \dots Q^{k_{3,m1}} \mathcal{X}^{\tau_{m1},k_{3,m1}}_{m1} \dots Q^{k_{3,ms_m}} \mathcal{X}^{\tau_{ms_m},k_{3,ms_m}}_{ms_m}(\psi)$, where for $i, j \geq 1$, with $\tau_{ij} = (r_{ij,1}, \dots, r_{ij,w_{ij}})$, it is $r_{ij,1}, \dots, r_{ij,w_{ij}} \leq k_{3,ij}$, Q is either $\exists^k$ or $\forall^k$, for some k, depending on whether m is odd or even, respectively, and ψ is an SO^ω formula with the addition of TO^ω atomic formulas. As usual, $\forall^k \mathcal{X}^{\tau,k}(\psi)$ abbreviates $\neg\exists^k \mathcal{X}^{\tau,k}(\neg\psi)$. We define $\mathrm{TO}^\omega = \bigcup_{m\geq 1} \Sigma^{2,\omega}_m$.

A TO^ω *relation* $\mathcal{R}^{\tau,k}$ *of type* τ *and closed under* $\equiv_k$ on a σ structure $\mathbf{I}$ is a set of w tuples $(R^{r_1,k}_1, \dots, R^{r_w,k}_w)$ of (second order) relations on $\mathbf{I}$ with respective arities $r_1, \dots, r_w \leq k$, closed under $\equiv_k$. The third order quantifier $\exists^k$ has the following semantics: let $\mathbf{I}$ be a σ-structure; then $\mathbf{I} \models \exists^k \mathcal{X}^{\tau,k}\varphi$ if there is a TO^ω relation $\mathcal{R}^{\tau,k}$ of type τ on I closed under the relation $\equiv_k$ in $\mathbf{I}$, such that $(\mathbf{I}, \mathcal{R}) \models \varphi$. Here $(\mathbf{I}, \mathcal{R})$ is the *third order* $(\sigma \cup \{\mathcal{X}^{\tau,k}\})$ structure expanding $\mathbf{I}$, in which $\mathcal{X}$ is interpreted as $\mathcal{R}$. Note that a *valuation* in this setting also assigns to each second order variable $X^{r,k}$ a (second order) relation on I of arity r that is closed under $\equiv_k$ in $\mathbf{I}$, and to each third order variable $\mathcal{X}^{\tau,k}$ a third order relation $\mathcal{R}^{\tau,k}$ on I of type τ, closed under $\equiv_k$ in $\mathbf{I}$. We don't allow free second or third order variables in the logics SO^ω and TO^ω. Note that allowing elements (from the domain of the structure) in a third order relation type would change the semantics of TO^ω, 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 $\Sigma^{2,\omega}_1$.

A *third order non-deterministic relational machine* ([AT,14]), denoted as 3-NRM, of *arity* k, for $k \geq 1$, is a 11-tuple $\langle Q, \Sigma, \delta, q_0, \mathfrak{b}, F, \sigma, \tau, T, \Omega, \Phi \rangle$ where: Q is the finite set of internal states; $q_0 \in Q$ is the initial state; Σ is the finite tape alphabet; $\mathfrak{b} \in \Sigma$ is the symbol denoting blank; $F \subseteq Q$ is the set of accepting states; τ is the finite vocabulary of the *rs* (its *relational store*), with finitely many TO^ω relation symbols $\mathcal{R}^{\tau_i,k'}_i$ of any arbitrary type $\tau_i = (r_{i1}, \dots, r_{iw})$, with $1 \leq r_{i1}, \dots, r_{iw} \leq k' = k$, and finitely many SO^ω relation symbols $R^{r_i,k''}_i$ of arities $r_i \leq k'' = k$; $T \in \tau$ is the output relation; σ is the vocabulary of the input structure; Ω is a finite set of TO^ω formulas with up to k FO variables, with *no* SO^ω *or* TO^ω *quantifiers*, and with no free variables of any order (i.e., all the SO^ω and TO^ω relation symbols are in τ); Φ is a finite set of TO^ω formulas with up to k FO variables, that are not sentences, with *no* SO^ω *or* TO^ω *quantifiers*, and where the free variables are either *all* FO variables, or *all* SO^ω variables; $\delta : Q \times \Sigma \times \Omega \to \mathcal{P}(\Sigma \times Q \times \{R, L\} \times \Phi \times \tau)$ is the transition function. In any pair in δ, if φ, S occur in the 5-tuple of its second component, for Φ and τ, then either S is a TO^ω relation symbol $\mathcal{R}^{\tau_i,k'}_i$ in rs and φ has $|\tau_i|$ SO^ω free variables $X^{r_1,k''}_1, \dots, X^{r_{|\tau_i|},k''}_{|\tau_i|}$ with arities according to τ_i, and $1 \leq r_1, \dots, r_{|\tau_i|} \leq k'' = k' = k$, or S is an SO^ω relation symbol $R^{r_i,k''}_i$ in rs and φ has $1 \leq r_i \leq k'' = k$ FO free variables. At any stage of the computation of a 3-NRM on an input σ-structure $\mathbf{I}$, there is one relation in its *rs* of the corresponding relation type (or arity) in $\mathbf{I}$ for each relation symbol in τ, so that in each transition there is a (finite) τ-structure $\mathbf{A}$ in the *rs*, which we can *query* and/or *update* through the formulas in Ω and Φ, respectively, and a finite Σ 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 $NEXPTIME_{3,r}$ 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: $NEXPTIME_{3,r} = \bigcup_{c\in N} NTIME_{3,r}(2^{c\cdot(size_k)})$ (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 Ω and Φ are FO formulas with up to k FO variables, in the vocabulary τ, and where all the relations in the *relational store* are second order relations of arity at most k. The relational complexity class $NEXPTIME_r$ 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: $NEXPTIME_r = \bigcup_{c\in N} NTIME_r(2^{c\cdot(size_k)})$.

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

Theorem 1. *([AT,14]) $NEXPTIME_{3,r} \subseteq \Sigma_1^{2,\omega}$. That is, given a 3-NRM M in $NTIME_{3,r}(2^{c\cdot(size_k)})$, for some positive integer c, and with input vocabulary σ that computes a Boolean query q we can build a formula $\varphi_M \in \Sigma_1^{2,\omega}$ such that, for every σ-structure $\boldsymbol{I}$, M accepts $\boldsymbol{I}$ iff $\boldsymbol{I} \models \varphi_M$.*

Theorem 2. *([AT,14]) $\Sigma_1^{2,\omega} \subseteq NEXPTIME_{3,r}$. That is, every class of relational structures definable in $\Sigma_1^{2,\omega}$ is in $\bigcup_{c\in N} NTIME_{3,r}(2^{c\cdot(size_k)})$.*

4 Existential $\mathbf{TO^\omega}$ captures $\mathbf{NEXPTIME}_r$

Corollary 3 *$NEXPTIME_r \subseteq \Sigma_1^{2,\omega}$. That is, given an NRM M that works in $NTIME_r(2^{c\cdot(size_k)})$, for some positive integer c, and with input vocabulary σ that computes a Boolean query q we can build a formula $\varphi_M \in \Sigma_1^{2,\omega}$ such that, for every σ-structure $\boldsymbol{I}$, M accepts $\boldsymbol{I}$ iff $\boldsymbol{I} \models \varphi_M$.*

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 $NEXPTIME_r$ iff M, as a 3-NRM, it is in $NEXPTIME_{3,r}$. □

Theorem 4. *$\Sigma_1^{2,\omega} \subseteq NEXPTIME_r$. That is, every class of relational structures definable in $\Sigma_1^{2,\omega}$ is in $\bigcup_{c\in N} NTIME_r(2^{c\cdot(size_k)})$.*

Proof. Let σ be a relational vocabulary, let φ be a $\Sigma_1^{2,\omega}[\sigma]$ sentence of the form $\exists^{k_{3,1}}\mathcal{X}_1^{\tau_1,k_{3,1}}\ldots\exists^{k_{3,s}}\mathcal{X}_s^{\tau_s,k_{3,s}}(\psi)$, where ψ is a $\Sigma_t^{1,\omega}$ formula, for some $t\geq 1$, with atomic TO^ω formulas formed with the TO^ω variables $\mathcal{X}_1,\ldots,\mathcal{X}_s$. 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 $\mathcal{X}_i$ is $\tau_i=(r_{3,i},...,r_{3,i})$ of cardinality $r_{3,i}$, with $r_{3,i}\leq k_{3,i}$.
Suppose the $\Sigma_t^{1,\omega}$ formula ψ is of the form $\exists^{k_{2,11}}Y_{11}^{r_{2,11},k_{2,11}}\ldots\exists^{k_{2,1l_1}}Y_{1l_1}^{r_{2,1l_1},k_{2,1l_1}}$ $\forall^{k_{2,21}}Y_{21}^{r_{2,21},k_{2,21}}\ldots\forall^{k_{2,2l_2}}Y_{2l_2}^{r_{2,2l_2},k_{2,2l_2}}\ldots Q^{k_{2,t1}}Y_{t1}^{r_{2,t1},k_{2,t1}}\ldots Q^{k_{2,tl_t}}Y_{tl_t}^{r_{2,tl_t},k_{2,tl_t}}(\phi)$, where the quantifiers $Q^{k_{2,t1}},\ldots,Q^{k_{2,tl_t}}$ are $\forall^{k_{2,t1}},\ldots,\forall^{k_{2,tl_t}}$, if t is even, or $\exists^{k_{2,t1}},\ldots,\exists^{k_{2,tl_t}}$, if t is odd, ϕ is an FO formula in the vocabulary $\sigma\cup\{Y_{11}^{r_{2,11},k_{2,11}},\ldots,Y_{tl_t}^{r_{2,tl_t},k_{2,tl_t}}\}$, with atomic TO^ω formulas, and $r_{2,11}\leq k_{2,11},\ldots,r_{2,tl_t}\leq k_{2,tl_t}$, respectively. We now build an NRM M_φ which accepts a given σ structure $\mathbf{I}$ iff $\mathbf{I}\models\varphi$. It is known that for every σ, and every $k\geq 1$, a formula $\gamma(\bar{x},\bar{y})$ with $k''\geq 2k$ variables of the fixed point logic $(FO+LFP)$ can be built s. t. on any σ structure $\mathbf{J}$, γ defines a pre-order $\preceq_k$ in the set of k-tuples of $\mathbf{J}$, 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 $M_{\preceq_k}$ of some arity $k'\geq 2k$ can be built, that constructs, on input $\mathbf{J}$, the pre-order $\preceq_k$ in $\mathbf{J}$, in time polynomial in $size_{k'}(\mathbf{J})$. We define the arity of M_φ as $k=max(\{k'_{3,1},\ldots,k'_{3,s},k'_{2,11},\ldots,k'_{2,tl_t}\})$, where the k'_{ij}'s are the arities of the RM's $M_{\preceq_{k_{3,1}}}$, …, $M_{\preceq_{k_{2,tl_t}}}$, respectively.

Let $\mathbf{I}$ be the input structure. M_φ works as follows: **1):** M_φ simulates the RM's $M_{\preceq_{k_{3,1}}}$, …, $M_{\preceq_{k_{2,tl_t}}}$, to build the pre-orders $\preceq_{k_{3,1}}$, …, $\preceq_{k_{2,tl_t}}$, respectively. M_φ builds those pre-orders in time polynomial in $size_{k'_{3,1}}(\mathbf{I}),\ldots,size_{k'_{2,tl_t}}(\mathbf{I})$, respectively. As all these arities are $\leq k$ (see above), that time is also polynomial in $size_k(\mathbf{I})$ (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_φ computes $size_{k_{3,1}}(\mathbf{I})$, and the same process is followed to compute $size_{k_{3,2}}(\mathbf{I})$, …, $size_{k_{2,tl_t}}(\mathbf{I})$ by using the equivalence relations $\equiv_{k_{3,2}}$, …, $\equiv_{k_{2,tl_t}}$, and the pre-orders $\preceq_{k_{3,2}}$, …, $\preceq_{k_{2,tl_t}}$, respectively (recall that the pre-orders $\preceq_{k_{3,1}}$, …, $\preceq_{k_{2,tl_t}}$, induce total orders in the equivalence classes of the corresponding equivalence relations $\equiv_{k_{3,1}}$, …, $\equiv_{k_{2,tl_t}}$). Note that by the choice of k, all these computations are done by M_φ in time polynomial in $size_k(\mathbf{I})$. **3):** M_φ needs to *guess* the TO^ω relations $\mathcal{S}_1^{\tau_1},\ldots,\mathcal{S}_s^{\tau_s}$, as interpretations of the TO^ω variables $\mathcal{X}_1^{\tau_1,k_{3,1}},\ldots,\mathcal{X}_s^{\tau_s,k_{3,s}}$ respectively. Each $\mathcal{S}_i^{\tau_i}$ is a set of $r_{3,i}$-tuples of $r_{3,i}$-ary (SO) relations closed under $\equiv_{k_{3,i}}$. To represent $\mathcal{S}_i^{\tau_i}$ we use three sorts of bit strings as follows: **a)** each bit string of sort $b^3_{R^{r_{3,i},k_{3,i}}}$ of size $size_{k_{3,i}}(\mathbf{I})$ represents one of the possible $r_{3,i}$-ary (SO) relations on $\mathbf{I}$, closed under $\equiv_{k_{3,i}}$; note that each bit represents one equivalence class in $\equiv_{k_{3,i}}$, following from left to right the total order induced by $\preceq_{k_{3,i}}$; **b)** each bit string of sort $b^2_{\bar{R}^{r_{3,i},k_{3,i}}}$ of size $r_{3,i}\cdot size_{k_{3,i}}(\mathbf{I})$ represents one of the possible $r_{3,i}$-tuples of $r_{3,i}$-ary (SO) relations on $\mathbf{I}$, closed under $\equiv_{k_{3,i}}$; **c)** each bit string of sort $b^1_{\mathcal{S}_i^{\tau_i,k_{3,i}}}$ of size $2^{r_{3,i}\cdot size_{k_{3,i}}(\mathbf{I})}$ represents one of the possible sets of $r_{3,i}$-tuples of $r_{3,i}$-ary (SO) relations on $\mathbf{I}$, closed under $\equiv_{k_{3,i}}$, i.e., one of the possible TO^ω relations on $\mathbf{I}$ of type τ_i, closed under $\equiv_{k_{3,i}}$. Let b be a bit string of sort b^1.

Each bit in b represents one of the possible bit strings of sort $b^2_{\bar{R}^{r_{3,i},k_{3,i}}}$ of size $r_{3,i} \cdot size_{k_{3,i}}(\mathbf{I})$. The *leftmost* bit in b represents a bit string of type $b^2_{\bar{R}^{r_{3,i},k_{3,i}}}$ 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 $b^2_{\bar{R}^{r_{3,i},k_{3,i}}}$ that correspond to the order in all the possible bit strings of sort $b^2_{\bar{R}^{r_{3,i},k_{3,i}}}$ according to their binary value. And so on, up to the *rightmost* bit in b, which represents a bit string of sort $b^2_{\bar{R}^{r_{3,i},k_{3,i}}}$ 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_φ guesses s bit strings of sort $b^1_{\mathcal{S}_i^{\tau_i}}$ of size $2^{r_{3,i} \cdot size_{k_{3,i}}(\mathbf{I})}$, one for each one of the relations $\mathcal{S}_i^{\tau_i}$. Note that this is done in time $2^{c \cdot size_{k_{3,i}}(\mathbf{I})}$, and hence also in time $2^{d \cdot size_k(\mathbf{I})}$, since $k_{3,i} \leq k$ (see above), for some constants c, d. **4):** Regarding the SO$^\omega$ variables quantified in the $\Sigma^{1,\omega}_t$ formula ψ, 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_φ. 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 $\Sigma^{1,\omega}_1 \subseteq NTIME_r((size_k)^c)$. 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 $2^{d \cdot size_{k_{2,ij}}(\mathbf{I})}$, and hence also by $2^{d \cdot size_k(\mathbf{I})}$, 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 ϕ is true. **5): Evaluation of ϕ:** Recall that ϕ is an FO formula with atomic TO$^\omega$ formulas. To evaluate ϕ we consider the syntax tree of ϕ, T_ϕ, and evaluate one node of it at a time in the finite control of M_φ, in a bottom up direction. To that end, for every node α in T_ϕ, that represents a sub-formula with $r \geq 1$ free FO variables, we define in the rs an r-ary relation variable R_α. And for every node α in T_ϕ, that represents a sub-formula with *no* free FO variables, we define in the rs a 1-ary relation variable B_α that represents a Boolean variable, which we interpret as True if $B_\alpha = dom(\mathbf{I})$, and as False if $B_\alpha = \emptyset$. Note that all the SO$^\omega$ relations that appear in the nodes in T_ϕ are in the rs of M_φ. Every node in T_ϕ is of one of the following kinds: i) an atomic FO formula with a relation symbol either in σ or quantified by an SO$^\omega$ quantifier in ψ, ii) a $\vee$ connective, iii) a $\wedge$ 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 φ. 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 α in T_ϕ corresponds to the sub-formula $\mathcal{X}_i^{\tau_i,k_{3,i}}(V_1^{r_{3,i},k_{3,i}}, \ldots, V_{r_{3,i}}^{r_{3,i},k_{3,i}})$

with $\tau_i = (r_{3,i}, ..., r_{3,i})$ 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 σ or quantified by an SO$^\omega$ quantifier in ψ. We check whether or not the $r_{3,i}$-tuple of relations $(V_1^{r_{3,i},k_{3,i}}, \ldots, V_{r_{3,i}}^{r_{3,i},k_{3,i}})$ is in the TO$^\omega$ relation $\mathcal{S}_i^{\tau_i,k_{3,i}}$ guessed above for the variable $\mathcal{X}_i^{\tau_i,k_{3,i}}$, using the (guessed) bit string $b^1_{\mathcal{S}_i^{\tau_i,k_{3,i}}}$ that represents $\mathcal{S}_i^{\tau_i,k_{3,i}}$, with the following algorithm, that clearly runs in time $2^{c\cdot size_{k_{3,i}}(\mathbf{I})}$, and hence also in time $2^{d\cdot size_k(\mathbf{I})}$, since $k_{3,i} \leq k$ (see above), for some constants c, d:

- $B_\alpha \leftarrow \emptyset$ (*i.e.*, $B_\alpha \leftarrow FALSE$);
- **for all** bit strings of sort $b^2_{\bar{R}^{r_{3,i},k_{3,i}}}$, $(a_1, \ldots, a_{2^{(r_{3,i}\cdot size_{k_{3,i}}(\mathbf{I}))}})$, counting in binary, varying n from 1 through $2^{r_{3,i}\cdot size_{k_{3,i}}(\mathbf{I})}$ (*i.e., for all $r_{3,i}$-tuples of $r_{3,i}$-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.*);
    - $S_{i,j}^{r_{3,i},k_{3,i}} \leftarrow \emptyset$;
    - **for** $l = 1$ through $size_{k_{3,i}}(\mathbf{I})$ (*i.e., bit l in bit substring of sort $b^3_{R_j^{r_{3,i},k_{3,i}}}$*);
      - **if** bit m of bit string a_n is 1, where $m = (j-1)\cdot size_{k_{3,i}}(\mathbf{I}) + l$, (*i.e., bit m in a bit string of sort $b^2_{\bar{R}^{r_{3,i},k_{3,i}}}$*)
        - add to $S_{i,j}^{r_{3,i},k_{3,i}}$ the l-th equivalence class in $\equiv_{k_{3,i}}$, according to pre-order $\preceq_{k_{3,i}}$ (*i.e., all the $r_{3,i}$-tuples of elements in that class*);
    - **end** l;
  - **end** j;
  - **if** bit n in bit string $b^1_{\mathcal{S}_i^{\tau_i,k_{3,i}}} = 1$
    - **if** $(V_1^{r_{3,i},k_{3,i}} = S_{i,1}^{r_{3,i},k_{3,i}} \wedge \ldots \wedge V_{r_{3,i}}^{r_{3,i},k_{3,i}} = S_{i,r_{3,i}}^{r_{3,i},k_{3,i}})$
      - $B_\alpha \leftarrow$ $dom(\mathbf{I})$ (*i.e.*, $B_\alpha \leftarrow TRUE$);
- **end all**; $\square$

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 $NTIME_{3,r}(2^{c\cdot(size_k)})$, for some positive integer c, that computes a Boolean query q. Then, there is a NRM M_2 that works in $NTIME_r(2^{d\cdot(size_k)})$, for some positive integer d, that also computes q.* $\square$

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 $\mathcal{L}^{\omega}_{\infty\omega}$ ([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 $\Sigma^{2,\omega}_1 \subseteq$ *(effective fragment of)* $\mathcal{L}^{\omega}_{\infty\omega}$. $\square$

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 $\Sigma^{1,F}_1$ is *not* included in $\mathcal{L}^{\omega}_{\infty\omega}$, as opposite to $\Sigma^{1,\omega}_1$ which is. Then, we have the following result:

Corollary 7 $\Sigma^{1,F}_1 \subsetneq \Sigma^{2,\omega}_1$. $\square$

References

1. [AT,14] J. Arroyuelo, J. M. Turull-Torres, "The Existential Fragment of Third Order Logic and Third Order Relational Machines", in Proceedings of the XIX Argentine Conference on Computer Science CACIC 2014", ISBN 978-987-3806-05-6, Buenos Aires, October 20-24, p. 324-333, 2014.
2. [AV,91] Abiteboul, S., Vianu, V., "Generic Computation and its Complexity", STOC 1991.
3. [AVV,95] Abiteboul, S., Vardi, M., Vianu, V., "Computing with Infinitary Logic", Theoretical Computer Science 149, 1, pp. 101-128, 1995.
4. [AVV,97] Abiteboul, S., Vardi, M. Y., Vianu, V., "Fixpoint logics, relational machines, and computational complexity", JACM 44 (1997) 30-56.
5. [Daw,98] Dawar, A., "A restricted second order logic for finite structures". Information and Computation 143 (1998) 154-174.
6. [Fag,74] Fagin, R., "Generalized First-Order Spectra and Polynomial-Time Recognizable Sets", in "Complexity of Computations", edited by R. Karp, SIAM-AMS Proc., American Mathematical Society, Providence, RI, pp. 27–41, 1974.
7. [FPT,10] F. A. Ferrarotti, A. L. Paoletti, J. M. Turull-Torres, "Redundant Relations in Relational Databases: A Model Theoretic Perspective", Journal of Universal Computer Science, Vol. 16, No. 20, pp. 2934-2955, 2010. http://www.jucs.org/jucs_16_20/redundant_relations_in_relational
8. [FT,08] Ferrarotti, F. A., Turull-Torres, J. M., "The Relational Polynomial-Time Hierarchy and Second-Order Logic", invited for "Semantics in Databases", edited by K-D. Schewe and B. Thalheim, Springer LNCS 4925, 29 pages, 2008.
9. [GT,10] Grosso, A. L., Turul-Torres J. M., "A Second-Order Logic in which Variables Range over Relations with Complete First-Order Types", 2010 XXIX International Conference of the Chilean Computer Science Society (SCCC) IEEE, p. 270-279, 2010.
10. [Lib,04] Libkin, L., "Elements of Finite Model Theory", Springer, 2004.
11. [Tur,06] J. M. Turull-Torres, "Relational Databases and Homogeneity in Logics with Counting", Acta Cybernetica, Vol 17, number 3, pp. 485-511, 2006.