Electronic Notes in Theoretical Computer Science 123 (2005) 111–123 

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

Fragments of Monadic Second-Order Logics Over Word Structures

# Yassine Hacha¨ıchi[1](#_bookmark0)

*IPEI-El Manar*

*BP. 244, 2092 El Manar (*Tunisie*).*

*Tel : (+216) 71 757 816*

Abstract

In this paper, we explore the expressive power of fragments of monadic second-order logic enhanced with some generalized quantifiers of comparison of cardinality over finite word structures.

The full monadic second-order fragment of the logics that we study correspond to the famous linear hierarchy, see [[10](#_bookmark15)], and their existential fragments characterize some sequential recognizers. We prove that the first-order closure of the existential fragments of these logics is strictly beyond the existential fragments.

*Keywords:* Descriptive complexity, Monadic Second-Order Logic, Generalized Quantifiers.

# Introduction

In the early sixties, Bu¨chi, Elgot and Trakhtenbrot [[3](#_bookmark11),[7](#_bookmark14),[22](#_bookmark28)] proved that a “finite words” language is recognized by a finite automaton if, and only if, it is the class of “finite word structures” satisfying a *MSO* sentence. Since then, monadic second-order logic (*MSO* for short) have been intensively explored as one of the cornerstones of logic in computer science, see the expository paper

[[21](#_bookmark27)] for historical background and state of art. Analogous results have been proved for infinite words, finite trees, infinite trees, and traces, see [[21](#_bookmark27)]. In all these cases, monadic second-order logic have the same expressive power than its existential fragment (∃*MSO* for short).

1 Email: [Yassine.Hachaichi@ipeiem.rnu.tn](mailto:Yassine.Hachaichi@ipeiem.rnu.tn)

1571-0661 © 2005 Elsevier B.V. Open access under [CC BY-NC-ND license.](http://creativecommons.org/licenses/by-nc-nd/3.0/)

doi:10.1016/j.entcs.2004.05.014

In the finite graphs topic, R. Fagin proved that the existential fragment of *MSO* is strictly less expressive than *MSO*. The famous problem of directed graphs connectivity is one of the problems (queries in databases terminology) that separates these classes, see [[13](#_bookmark16),[5](#_bookmark12)].

In his thesis [[11](#_bookmark17)], the author improved a result of T. Schwentick by proving that either if we allow generalized quantifiers of cardinality comparison (in some restricted form), over ordered graphs, graph connectivity is still not expressible in ∃*MSO*.

Ajtai and *al*, and Matz, in [[1](#_bookmark9),[18](#_bookmark24)], have introduced and studied some frag- ments of *MSO* that contains ∃*MSO*. One of these fragments is the first-order closure of ∃*MSO*. It is the set of prenex formulas whose monadic second-order variables are existentially quantified and alternated by quantified first-order (universally, or existentially) variables. This class is denoted *FO*(∃*MSO*). In [[1](#_bookmark9)], it has been proved that ∃*MSO* Ç *FO*(∃*MSO*) Ç *MSO* over finite graphs. There were also affirmed that *FO*(∃*MSO*) is a “natural” extension of ∃*MSO*. In this paper, we prove analogous results in the aim of finite word structures where we allow generalized quantifiers of cardinality comparison of quantified sets. These logics are monadic second-order logic enhanced by partial order constraints over the cardinality of quantified sets. We will use for this purpose Logical characterizations of some classes of languages and some complexity

results.

This paper is organized as follows: In the next section, we introduce the logics we explore and recall the descriptive complexity results needed in the sequel. In section 3, we study the fragments of *MSO* augmented with some partial order over quantified subsets. In section 4, we study the expressive power of fragments of monadic second-order logic enhanced by the famous Rescher and H¨artig quantifiers over finite word structures. In the conclusion of the paper we give some remarks over the results and some directions to explore.

# Preliminary definitions and results

Let’s first define how to identify words with logical structures, see for example [[21](#_bookmark27),[13](#_bookmark16)].

Definition 2.1 We associate with each word *w* = *w*0 *... wn*−1 over the alpha- bet Σ, the *word structure Sw*, namely the relational structure *Sw* = ([*n*]*,<*

*,* (*Pa*)*a*∈Σ), where [*n*] = {0*,... ,n* − 1}, *<* is the linear order on [*n*], and *Pa* is the unary predicate collecting the positions of *w* labeled *a*:

*Pa* = {*i* ∈ [*n*]|*wi* = *a*}*.*

In the case of binary word structures, i.e. Σ = {0*,* 1}, we need a single predicate collecting the positions labeled 1.

*P* = {*i* ∈ [*n*]|*wi* = 1}*.*

In the rest of this section, we define the enhanced monadic second-order logics we study in the next sections. We suppose the reader familiar with monadic second-order logic and its existential fragment, see [[5](#_bookmark12),[13](#_bookmark16),[21](#_bookmark27)] for de- tailed definitions. We after recall some descriptive complexity results that we need in the separation results.

Definition 2.2 Let *MSO*(≤*g,* =*g*) be the monadic-second order logic over word structures where atomic formulas are of the one of the forms *x* = *y, x < y, P* (*x*)*,U* (*x*)*,U* =*g V* and *U* ≤*g V* , for some individual variables or constants *x, y* and set variables *U, V* .

The semantic of this logic is the natural one for monadic second-order logic, and the interpretations of *X* =*g Y* and *X* ≤*g Y* are *partial orders between subsets of the universe* [*n*], as introduced in [[19](#_bookmark25)]:

([*n*]*,* ·· ·) |= *X* ≤*g Y* iff for all *m < n,* |*X* ∩ [*m*]| ≤ |*Y* ∩ [*m*]|*,*

and

([*n*]*,* ·· ·) |= *X* =*g Y* iff *X* ≤*g Y* and |*X*| = |*Y* |*.*

Example 2.3 Let Σ = {(*,* )}. A word *w* = *u*0 ··· *un*−1 over Σ is a sequence of well balanced parentheses (a word of the two symbols Dyck language, see [[12](#_bookmark18)]) if, and only if:

*Sw* |= {*i < n*|*ui* = ‘)'} =*g* {*i < n*|*ui* = ‘('}*.*

Let *MSO*(*Qr*) (resp. *MSO*(*Qh*)) be monadic second-order logic where atomic formulas are of the form *x* = *y, x < y, U* (*x*)*,P* (*x*) and *Qr*(*U, V* ) (resp. *Qh*(*U, V* )), for individual variables or constants *x, y* and set variables *U, V* . The semantics of these logics are the same as monadic second-order logic where:

* *Qr* is interpreted as the *Rescher quantiﬁer Qr*, or the majority of cardinality quantifier, defined by:

*Qr*(*X, Y* ) ≡ |*X*| *<* |*Y* |*.*

* *Qh* is the *H¨artig quantiﬁer Qh*, also called equicardinality quantifier, defined by:

*Qh*(*X, Y* ) ≡ |*X*| = |*Y* |*.*

Petri nets were introduced in the aim to study concurrency. This model is also studied as a model of sequential computing. It is this last point of view that interests us in this paper. Naturally, this sequential model has been compared to classical models of the Chomsky hierarchy, we will denote this class *PNL*. It has been proved that:

*Reg* Ç *PNL* Ç *CS*

where *Reg* (resp *CS*) denotes the class of regular languages (resp context sensitive). For a detailed introduction, and motivations that led to the study of the sequential behavior of Petri nets, see for example, chapter 6 of [[20](#_bookmark26)], the article [[19](#_bookmark25)] or [[11](#_bookmark17)].

Result 2.4 (Parigot and Pelz [[19](#_bookmark25)]) *Let L be a language over an alphabet*

Σ*. The following are equivalent:*

1. *L is a Petri net language;*
2. *L is deﬁned by a sentence of MSO*(=*g,* ≤*g*) *of the form* ∃*Xφ*(*X*)*, where*

*φ*(*X*) *is MSO*(=*g,* ≤*g*)*-ﬁrst-order formula over* Σ ∪ *X;*

1. *L is deﬁned by a sentence of MSO*(=*g,* ≤*g*) *of the form* ∃*Xφ*(*X*)*, where φ*(*X*) *is a positive combination* [2](#_bookmark1) *of formulas of the form X* =*g Y and ﬁrst-order formulas in which* =*g and* ≤*g do not occur.*

In order to define non regular languages, we have to use a strictly more power- ful logic than *MSO*. In the other hand, if we add a quantified binary predicate expresses all context free and some *NP* -complete languages, following a re- cent result of Eiter, Gottlob and Gurevich [[6](#_bookmark13)]. In [[6](#_bookmark13)], the authors proved that a prefix class of second-order logic, either expresses only regular languages, or defines some *NP* -complete problem. Furthermore, they proved that *NP* - hardness is present in formulas of the form ∃*R φ*, for some binary predicate *R* and a first-order formula *φ* of the appropriate prefix. Lautemann, Schwentick and Th´erien [[16](#_bookmark22)] chose a semantic approach in order to characterize the class of context free languages. They confined the binary second-order predicate to be a matching, i.e. an order preserving, non-crossing relation.

Definition 2.5 *A binary relation M over a word structure is called a* match- ing *if it satisﬁes the following conditions :*

1. ∀*ij*[(*i, j*) ∈ *M* ⇒ *i < j*]*.*

2 This means that we only use ∧ and ∨ in the construction of formulas.

1. ∀*ij*[(*i, j*) ∈ *M* ⇒ ∀*k* /= *i, j*((*i, k*)*,* (*k, i*)*,* (*j, k*)*, and* (*k, j*) are not in *M* )]*.*
2. ∀*ijkl*[(*i, j*)*,* (*k, l*) ∈ *M* ⇒ (*i < k < j* → *i < l < j*)]*.*

*Let* Match *denote the class of matchings on word structures.*

Let *S* be any word structure, *S* |= ∃*MatchM φ* means : there exists a relation

*M* ∈ *M atch* such that (*S, M* ) |= *φ*.

Example 2.6 Suppose that the positions in *P* are opening parentheses. The formula:

∃*MatchM* ∀*x*∀*y*∃*z*((*M* (*x, z*) ∨ *M* (*z, x*)) ∧ (*M* (*x, y*) → (*P* (*x*) ∧ ¬*P* (*y*)))) defines the two letters Dyck language.

Result 2.7 *A ﬁnite word language is context free if, and only if, it is the*

*class of models of a formula of the form* ∃*MatchM φ, where φ is a ﬁrst-order formula using M.*

By combining this result with a result of Book and Greibach [[2](#_bookmark10)] which states: *“A language is in NTime*[*n*] *if, and only if, it is the projection of a ﬁnite intersection of context free languages”,* Lautemann, Schwentick et Schweikardt

[[15](#_bookmark21)] cited the following result:

Result 2.8 *Over binary word structures:*

*NTime*[*n*]= ∃*MatchM*1 *... Mk*∃*R*(*φ*1∧*.. .*∧*φk*)= ∃*MatchM*1*M*2*M*3∃*R*(*φ*1∧*φ*2∧*φ*3)

*where the M* '*s are restricted to be matchings and the only binary relations in*

*i*

*φi are Mi and <.*

Proof. (of ⊇) Let *L* be a language defined by a formula of the form:

∃*MatchM*1 ··· *Mk*∃*R*(*φ*1 ∧ ··· ∧ *φk*)

with the condition that the only binary relations in *φi* are *Mi* and *<*, and all

the *Rj*’s are unary variables. We first guess the unary relations *R*¯ in non-

deterministic linear time. We assume these sets as part of the augmented signature. *We treat the Ri’s as new letters of the alphabet, with minor modi- ﬁcations in order to have that one position have at most one label.*

We finally evaluate the formulas ∃*Miφi*, which describe context free lan- guages by result [2.7](#_bookmark2), in non-deterministic linear time. The other direction is close to the one exhibited in [[15](#_bookmark21)] for a slightly modified logic.

Definition 2.9 *We deﬁne the* first-order closure of ∃*MSO*(=*g,* ≤*g*) *(resp. of*

∃*MSO*(*Qh, Qr*)*), and we note: FO*(∃*MSO*(=*g,* ≤*g*)) *(resp. FO*(∃*MSO*(*Qh, Qr*)*),*

*as the set of prenex formulas of MSO*(=*g,* ≤*g*) *(resp. of MSO*(*Qh, Qr) in which we authorize alternations between ﬁrst-order quantiﬁers and existential monadic second-order ones. These are formulas of the form:*

∃*X*∀*x*∃*Y* ··· *θ*

*where upper case letters stand for set variables, lower case one for individual* *variables and θ is a quantiﬁer free formula.*

1. *FO*(∃*MSO*(=*g,* ≤*g*)) versus ∃*MSO*(=*g,* ≤*g*)

In this section, we prove that *FO*(∃*MSO*(=*g,* ≤*g*)) is strictly more expressive than ∃*MSO*(=*g,* ≤*g*) over finite word structures. To this aim, we prove that *FO*(∃*MSO*(=*g,* ≤*g*)) defines all context free languages, while it is known, see [[20](#_bookmark26)], that there are context free languages that are not Petri net languages (which corresponds to ∃*MSO*(=*g,* ≤*g*) by [[19](#_bookmark25)]).

Theorem 3.1 *Over ﬁnite word structures:*

∃*MSO*(=*g,* ≤*g*) Ç *FO*(∃*MSO*(=*g,* ≤*g*))*.*

In order to prove this result, let’s prove first that the context free languages are definable in *FO*(∃*MSO*(=*g,* ≤*g*)).

Theorem 3.2 *Over ﬁnite word structures:*

*CFL* ⊆ *FO*(∃*MSO*(=*g,* ≤*g*))*.*

Proof. By a result of [[16](#_bookmark22)], characterizing *CFL* by sentences of ∃*M atch FO*, it suffices to prove that the formulas in ∃*M atch FO* are expressible in *FO*(∃*MSO*(=*g*

*,* ≤*g*)). Let *L* a context free language definable by the formula:

Φ*L* ≡ ∃*MatchMφ*(*M* )*,*

The formula *αM* :

∃*X*1*X*2(*X*1 =*g X*2 ∧ ∀*x*¬(*X*1(*x*) ∧ *X*2(*x*)))*,*

of *FO*(∃*MSO*(=*g,* ≤*g*)), ensure us of the existence of two disjoint subsets of positions that could be interpreted as, the set of opening parentheses:

*X*2 ≡ {*x*|∃*yM* (*x, y*)}*,*

and the set of closing parentheses:

*X*1 ≡ {*x*|∃*yM* (*y, x*)}*,*

of the matching *M* . We after express in *FO*(∃*MSO*(=*g,* ≤*g*)), the fact that two positions , *x* and *y*, are linked by the matching *M* , represented by the sets *X*1 et *X*2. This is made by the formula *µ*(*x, y*):

∃*Y*1*Y*2∀*z*[(*Y*1(*z*) ↔ (*X*1(*z*) ∧ *x < z < y*))

∧(*Y*2(*z*) ↔ (*X*2(*z*) ∧ *x < z < y*)) ∧ (*X*2(*x*) ∧ *X*1(*y*) ∧ *Y*1 =*g Y*2)]*.*

In fact, this formula mimic an elementary algorithm of finding the closing parenthese of each opening one. In order to have uniquely existential second- order quantifiers, the occurences of the atomic formulas of the form *M* (*x, y*) must be all positive. For this aim, we express ¬*M* (*x, y*) by a positive formula in *M* .

¬*M* (*x, y*) ≡ ¬*X*2(*x*) ∨ ¬*X*1(*y*) ∨ ∃*z*(*z* /= *y* ∧ *M* (*x, z*))*.*

Our translation will be as follows:

We begin by replacing the negative occurrences of *M* (*x, y*) in the initial formula by formulas where *M* occur only positively as given above.

After, we replace ∃*M* by *αM* , and the occurences of *M* (*x, y*) by *µ*(*x, y*). We can easily check that this produces a logically equivalent *FO*(∃*MSO*(=*g,* ≤*g*))- formula.

As a direct consequence of this result and the one of Book and Greibach [[2](#_bookmark10),[15](#_bookmark21)], see also result [2.8](#_bookmark3):

Corollary 3.3 *Over binary ﬁnite word structures:*

*NTime*[*n*] ⊆ *FO*(∃*MSO*(=*g,* ≤*g*))*.*

Proof. (of corollary [3.3](#_bookmark6)) In result [2.8](#_bookmark3), it is proved that:

*NTime*[*n*]= ∃*MatchM*1*, M*2*, M*3∃*R*¯(*φ*1 ∧ *φ*2 ∧ *φ*3)*,*

where, for *i* = 1*,* 2*,* 3, *φi* ∈ *FO*[*Mi, R*¯], and the *Ri* are unary. By the previ- ous theorem, and the closure of *FO*(∃*MSO*(=*g,* ≤*g*)) by conjunction (which expresses intersection) and by existential monadic second-order quantification (which expresses projection), we obtain the required claim.

Proof. (of theorem [3.1](#_bookmark4)) It is known that *CFL* /⊂ *PNL*, see for example [[20](#_bookmark26)]. This result is equivalent to ∃*M atch FO* /⊂ ∃*MSO*(=*g,* ≤*g*), by results of

[[16](#_bookmark22)] and [[19](#_bookmark25)]. By theorem [3.2](#_bookmark5)

*CFL* = ∃*M atch FO* ⊆ *FO*(∃*MSO*(=*g,* ≤*g*))*.*

We conclude:

∃*MSO*(=*g,* ≤*g*) Ç *FO*(∃*MSO*(=*g,* ≤*g*))*.*

# Fragments of *MSO*(*Qh, Qr*) over word structures

In this section, we first investigate the expressive power of ∃*MSO*(*Qh, Qr*). We give an upper bound for this class which is *NTime*[*n*], we after give it a lower bound which is the union of the class of bounded context free languages and the regular ones.

After that, we prove that this class is strictly included in its first-order closure.

Theorem 4.1 *Over binary ﬁnite word structures:*

∃*MSO*(*Qh, Qr*) ⊆ *NTime*[*n*]*.*

Lemma 4.2 *Each formula of* ∃*MSO*(*Qh, Qr*) *is equivalent to a one in which*

*Qh and Qr occur only positively.*

Proof. We replace, in each formula, the occurrences of ¬*Qh*(*X, Y* ) by

*Qr*(*X, Y* ) ∨ *Qr*(*Y, X*)

and we replace the occurrences of ¬*Qr*(*X, Y* ) by

*Qh*(*X, Y* ) ∨ *Qr*(*Y, X*)*.*

After this procedure, we obtain the desired result.

Proof. (of theorem [4.1](#_bookmark7)) We begin by replacing each formula of ∃*MSO*(*Qh, Qr*) by an equivalent one in ∃*MSO*(*Qh, Qr*) in which *Qh* and *Qr* appear only pos- itively, as in the previous lemma.

Next, we replace the occurrences of atomic formulas of the form *Qh*(*Z, T* ) by the sentence: ∃*M* ∀*xy*[(*M* (*x, y*) ∨ *M* (*y, x*)) → ((*Z*(*x*) ∧ ¬*T* (*x*) ∧ *T* (*y*) ∧

¬*Z*(*y*))∨((*T* (*x*)∧¬*Z*(*x*)∧*Z*(*y*)∧¬*T* (*y*)))]∧∀*x*((*Z*(*x*)∧¬*T* (*x*)) → ∃*y*(*M* (*x, y*)∨

*M* (*y, x*))) ∧ ∀*x*((*T* (*x*) ∧ ¬*Z*(*x*)) → ∃*y*(*M* (*x, y*) ∨ *M* (*y, x*)))*.* This is possible because we always can find a bijection whose graph edges do not cross (as in the construction of parentheses). We associate with each element satisfying *Z*(*x*) ∧ ¬*T* (*x*) the least element such that:

*T* (*y*) ∧ ¬*Z*(*y*) and there is as many elements satisfying *Z* and *T* between these positions and *vice versa*.

We after replace the occurrences of atomic formulas of the form *Qr*(*Z, T* ) by the sentence: ∃*M* ∀*xy*[(*M* (*x, y*) ∨ *M* (*y, x*)) → ((*Z*(*x*) ∧ ¬*T* (*x*) ∧ *T* (*y*) ∧

¬*Z*(*y*))∨((*T* (*x*)∧¬*Z*(*x*)∧*Z*(*y*)∧¬*T* (*y*)))]∧∀*x*((*Z*(*x*)∧¬*T* (*x*)) → ∃*y*(*M* (*x, y*)∨

*M* (*y, x*)))∧∃*x*((*T* (*x*)∧¬*Z*(*x*))∧∀*y*¬(*M* (*x, y*)∨*M* (*y, x*)))*.* Who corresponds to a non crossing bijection between a proper subset of *T* ∧¬*Z* and *Z* ∧¬*T* . Because,

*M* appear only in a single translation, we can, via an appropriate renaming, put the ∃*Mi*’s in the beginning of the prenex formula. This formula is so in the form given in Result [2.8](#_bookmark3). We conclude that ∃*MSO*(*Qh, Qr*) ⊆ *NTime*[*n*].

Theorem 4.3 *Over binary ﬁnite word structures:*

∃*MSO*(*Qh, Qr*) /⊆ *CFL.*

Proof. Let *L* = {*anbncn*|*n* ∈ N} be the non context free language over Σ =

{*a, b, c*}, see [[12](#_bookmark18)]. The ∃*MSO*(*Qh, Qr*)-formula:

∃*XY Z*∃*xy*∀*t*[(*X*(*t*) ↔ *t* ≤ *x*) ∧ (*X*(*t*) ↔ *Pa*(*t*))∧

(*Y* (*t*) ↔ *x < t* ≤ *y*) ∧ (*Y* (*t*) ↔ *Pb*(*t*))∧

(*Z*(*t*) ↔ *y < t*) ∧ (*Z*(*t*) ↔ *Pc*(*t*)) ∧ *Qh*(*X, Y* ) ∧ *Qh*(*Y, Z*)]*.*

defines *L*. We conclude that *CFL* does not contain ∃*MSO*(*Qh, Qr*).

Definition 4.4 *The class of bounded context free languages, denoted BCFL, introduced by Ginsburg in 1966 (see [*[*20*](#_bookmark26)*] pages 181-182) is the least class such that:*

* 1. *Finite languages are in BCFL;*
  2. *If L*1 *and L*2 *are in BCFL, then so are L*1*L*2 *and L*1 ∪ *L*2*;*
  3. *If L is in BCFL, and u, v are ﬁnite words of* Σ∗*, then:*

{*uiLvi*|*i* ≥ 0}*,*

*is in BCFL.*

Remark: The class *BCFL* does not contain all regular languages because Σ∗ is not in *BCFL* if Σ contains at least two letters. The class *BCFL* is not included in the class of regular languages because *L* = {*anbn*|*n* ∈ N}, which is not regular, is in *BCFL*.

Theorem 4.5 *Over ﬁnite word structures,* ∃*MSO*(*Qh, Qr*) *contains regular languages and BCFL.*

Proof. Over finite word structures, the class of regular languages correspond to ∃*MSO*, see [[3](#_bookmark11),[21](#_bookmark27)], so obviously, *Reg* ⊆ ∃*MSO*(*Qh, Qr*). Let us show that *BCFL* ⊆ ∃*MSO*(*Qh, Qr*) by structural induction over the construction of *BCFL*.

Basis. Let *φw* be the first-order sentence which is satisfied by the word *w* = *w*1 ··· *wn*. So finite languages are definable by a finite disjunction of such formulas.

Closure operations. For union, it is easy to check whether the dis- junction of sentences in ∃*MSO*(*Qh, Qr*) is a sentence in ∃*MSO*(*Qh, Qr*). For

concatenation, it is a simple relativization of variables. Let *X* be a set (a unary predicate). We define *SX*(*i, j*), the successor relation relative to *X* by:

*i < j* ∧ *X*(*i*) ∧ *X*(*j*) ∧ ∀*k*(*i < k < j* → ¬*X*(*k*))*.*

Let *M axX* (*i*) the predicate stating that *i* is the greatest element satisfying *X*, and let *M inX* (*i*) the predicate stating that *i* is the least element satisfying *X*. Let Φ*L* ∈ ∃*MSO*(*Qh, Qr*) the formula defining *L*. We will mark the elements forming the last letter of *u* by a predicate *X*, and those forming the first letter of *v* by *Y* . The formula defining {*uiLvi*|*i* ∈ N} expresses that there are such sets *X, Y* and positions *x, y* such that *M axX* (*x*) and *M inY* (*y*), and if *SX*(*i, j*) then *φu*(*i* + 1*, j*), and if *SY* (*i, j*) then *φv*(*i, j* − 1).

∃*XY* ∃*xy*[(*x* ≤ *y*) ∧ *M axX* (*x*) ∧ *M inY* (*y*)∧

∀*ij*(*SX* (*i, j*) → *φu*(*i* + 1*, j*)) ∧ *φu*(*min, M inX* )

∧∀*ij*(*SY* (*i, j*) → *φv*(*i, j* − 1)) ∧ *φv*(*M axY , max*)]

*Qh*(*X, Y* ) states that the word is of the form *uimvi* with *m* ∈ Σ∗. We finally relativize the formula Φ*L* to the integer interval [*x* + 1*,y* − 1].

Definition 4.6 *Let’s deﬁne the logic* ∃*MSO*+(*Qh, Qr*)*. The formulas of this logic are the same as those of* ∃*MSO*(*Qh, Qr*) *with the authorization to have atomic formulas of the form:*

*Qh*(*φ*(*x*)*, ψ*(*x*)) and *Qr*(*φ*(*x*)*, ψ*(*x*))*,*

*for ﬁrst order formulas φ and ψ, over the signature augmented by the monadic predicates, having a single free variable and parameters.*

Theorem 4.7 *Over ﬁnite word structures:*

*NTime*[*n*] ⊆ ∃*MSO*+(*Qh, Qr*)*.*

Proof. The formula :

*Qh*(0 ≤ *i* ≤ *x, y* ≤ *i* ≤ *z*)*,*

defines *z* = *x* + *y*. In this formula, *x, y* and *z* are parameters, they are so free in this atomic formula. We then have ∃*MSO*(+) ⊆ ∃*MSO*+(*Qh, Qr*)*.* Using a result of Lynch in [[17](#_bookmark23)], we obtain:

*NTime*[*n*] ⊆ ∃*MSO*(+) ⊆ ∃*MSO*+(*Qh, Qr*)*.*

Here is a state of art of the results involved and proved in this section:

∃*MSO*(*Qh, Qr*) ⊆ *NTime*[*n*] ⊆ *N LIN* ⊆ ∃*MSO*(+) ⊆ ∃*MSO*+(*Qh, Qr*) The inclusion *N LIN* ⊆ *MSO*(+) is a result of Grandjean and Olive [[9](#_bookmark19)].

Theorem 4.8 *Over ﬁnite word structures:*

∃*MSO*+(*Qh, Qr*) ⊆ *FO*(∃*MSO*(*Qh, Qr*))

Proof. The first step, will be to replace the formulas of ∃*MSO*+(*Qh, Qr*) by formulas in which *Qh* and *Qr* appear only positively, because in the proof of lemma [4.2](#_bookmark8) we can replace ∃*MSO*(*Qh, Qr*) by ∃*MSO*+(*Qh, Qr*) without changing the proof. After we replace the occurrences of *Qh*(*φ, ψ*) by

∃*X*∃*Y* ∀*x*((*X*(*x*) ↔ *φ*(*x*)) ∧ (*Y* (*x*) ↔ *ψ*(*x*)) ∧ *Qh*(*X, Y* ))

for new (not previously used) variables *X, Y* and *x*. We do the same for

*Qr*(*φ, ψ*) by

∃*X*∃*Y* ∀*x*((*X*(*x*) ↔ *φ*(*x*)) ∧ (*Y* (*x*) ↔ *ψ*(*x*)) ∧ *Qr*(*X, Y* ))

Because these variables do not occur in the other subformulas, we can put the quantifiers ∃*X*∃*Y* ∀*x* in front of the first-order formula and we obtain a formulas in *FO*(∃*MSO*(*Qh, Qr*)). For example, let the formula

∀*x*∀*y*∃*z*(¬*Qh*(0 ≤ *i* ≤ *x, y* ≤ *i* ≤ *z*))

that states that for any *x* and *y*, there is a *z* which is different from *x* + *y*. We begin by replacing the negative occurrence of *Qh* and obtain:

∀*x*∀*y*∃*z*(*Qr*(0 ≤ *i* ≤ *x, y* ≤ *i* ≤ *z*) ∨ *Qr*(*y* ≤ *i* ≤ *z,* 0 ≤ *i* ≤ *x*)) By the second procedure, we have:

∀*x*∀*y*∃*z*∃*X*∃*Y* ∀*i*((*X*(*i*) ↔ 0 ≤ *i* ≤ *x*) ∧ (*Y* (*i*) ↔ *y* ≤ *i* ≤ *z*)

∧(*Qr*(*X, Y* ) ∨ *Qr*(*Y, X*)))

It is also proved in [[14](#_bookmark20)] that:

Result 4.9 ∃*MSO*(*Qh, Qr*) *does not express Petri Net languages.*

We then conclude that

∃*MSO*(*Qh, Qr*) Ç *FO*(∃*MSO*(*Qh, Qr*))

# Conclusion

In [[1](#_bookmark9)], the authors proved that in the presence of a binary predicate in the signature:

∃*MSO* Ç *FO*(∃*MSO*) Ç *MSO*

In a recent paper [[10](#_bookmark15)], the author proved that over binary word structures:

*MSO*(+) = *MSO*(=*g,* ≤*g*)= *MSO*(*Qr*)= *MSO*(*Qh*)= *MSO*(*M aj*)= *LinH*

It is an easy exercise to prove that :

*FO*(∃*MSO*(+)) = *FO*(∃*MSO*(=*g,* ≤*g*)) = *FO*(∃*MSO*(*Qr, Qh*))

We then have that the existential fragments and all alternation classes of these logics correspond to “natural” complexity classes, which legitimate their study as “natural” classes. Does the classes *FO*(∃*MSO*) also correspond to some machine model? By the definition of these fragments, in all cases:

∃*MSO* ⊆ *FO*(∃*MSO*) ⊆ *MSO*

It will be interesting for further works, to investigate in which cases these inclusions are strict. Some consequences over collapse of complexity classes can be given.

In his PhD [[18](#_bookmark24)], Oliver Matz studied a finer hierarchy than *MSO* and compared it to the classical *MSO* hierarchy. We can try to set analogous results in the context of this work.

# References

1. M. Ajtai, R. Fagin & L. Stockmeyer, *The Closure of Monadic NP ,* J. Computer and System Sciences, 60, 2000, 660-716.
2. R. Book and S. Greibach, *Quasi Realtime Languages*, Math. System Theory, 4, 1970, 97-111.
3. J. R. Bu¨chi, *Weak Second-Order Arithmetic and Finite Automata*, Zeit.Math.Logik und Grund. Math., 6, 1960, 66-92.
4. J. E. Doner, *Tree Acceptors and Some of their Applications*, Journal of Computer and System Science, 4, 1970, 406-451.
5. H. D. Ebbinghaus & J. Flum, *Finite Model Theory*, Springer, 1995.
6. T. Eiter, G. Gottlob & Y. Gurevich, *Existential Second order Logic Over Strings*, in Proc. of the 13*th* Symp. of Logic In Computer Science, 1998, 16-27.
7. C. C. Elgot, *Decision Problems of Finite Automata and related Arithmetics,* Trans. of the AMS, 98, 1961, 21-52.
8. R. Fagin, *Generalized First-order Spectra and Polynomial-time Recognizable Sets,* R. Karp editor, The Complexity of Computation, SIAM-AMS Proceedings, Vol 7, 1974, 27-41.
9. E. Grandjean & F. Olive, *Monadic Logical Deﬁnability of NP-Complete Problems*, in CSL’94, LNCS 933, 1995, 190-204.
10. Y. Hacha¨ıchi, *A descriptive complexity approach to the linear hierarchy,* Theoretical Computer Science, 304, 2003, 421-429.
11. Y. Hacha¨ıchi, *Contibutions a` la th´eorie des mod`eles ﬁnis et `a la complexit´e descriptive,* PhD, Universit´e Denis Diderot (Paris 7), July 2001.
12. M. A. Harrison, *Introduction to Formal Language Theory*, Addison Wesley, 1978.
13. N. Immerman, *Descriptive Complexity*, Graduate Texts in Computer Science, Springer, New York, 1999.
14. F. Klaedtke and H. Ruess, *WS1S with cardinality constraints,* SRI Technical report, SRI-CSL- 04-01.
15. C. Lautemann, N. Schweikardt and T. Schwentick, *A Logical Characterization of Nondeterministic Linear Time on Turing Machines,* STACS ’99, Lecture Notes in Computer Science 1563, 143-152.
16. C. Lautemann, T. Schwentick and D. Th´erien, *Logics for Context Free Languages*, CSL’95, Lecture Notes in Computer Science, Volume 933, 205-216.
17. J. Lynch, *Complexity Classes and Theories of Finite Models,* Math. System Theory, 15, 1982, 127-144.
18. O. Matz, *Dot-Depth and Monadic Quantiﬁer Alternation Over Pictures*, PhD, RWTH Aachen, june 1999.
19. M. Parigot & E. Pelz, *A Logical Approch of Petri Net Languages*, Theoretical Computer Science, 39, 1985, 155-169.
20. J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice Hall, 1981.
21. W. Thomas, *Languages, Automata and Logic*, in Handbook of formal languages, vol 3, Springer Verlag, 1997, 389-455.
22. B. A. Trakhtenbrot, *Finite Automata and the Logic of Monadic Second Order Predicates*, Dokl. Akad. Nauk SSSR, 140, in russian, 1961, 326-329.