Some Results on Majority Quantifiers over Words

Klaus-Jörn Lange
Wiolhelm–Schickard–Institut für Informatik
Eberhard–Karls–Universität Tübingen
Sand 13, D-72076 Tübingen, Germany
lange@informatik.uni-tuebingen.de

Abstract

The class of languages definable by majority quantifiers using the order predicate is investigated. It is shown that the additional use of first order or counting quantifiers does not increase this class. Further on, addition is in this connection a definable numerical predicate, while the converse does not hold. The emptiness problem for this class turns out to be undecidable.

1. Introduction

Barrington et al. characterized the languages in TC^0 as those definable by first oder and majority quantifiers using as numerical predicates addition, multiplication, and the order rpedicate ([**?**]). Later, Lautemann et al. showed that using only majority quantifiers and the order predicate results in a proper subclass of TC^0 ([**?**]). In this paper we continue the investigation of this class and show:

- Both first order and counting quantifiers can be expressed by majority quantifiers using the order predicate.
- Addition is definable by majority quantifiers using the order predicate.
- The order predicate is not definable by majority quantifiers using addition and multiplication.
- The numerical predicates definable by majority quantifiers using the order predicate are precisely those definable by first order quantifiers using addition.
- Satisfiability over words is undecidable for majority quantifiers using the order predicate.
- There exists a computable function T such that L is in TC^0 iff $T(L)$ is definable by majority quantifiers using the order predicate.

It seems surprising that the situation of definability between the order predicate and addition is reversed when going from first order quantifiers to majority ones.

2. Preliminaries

2.1. Logic formulae over words

Throughout of this paper we consider languages defined by logical formulae. We use the notation as it is presented in the book of Straubing ([**?**]). In general, i, j, k, n will denote positive integers, while x, y, z denote position variables with positive integer values. The integer n will usually denote the lenght of the actual input word. Thus, variables will range over $\{1, 2, \cdots, n\}$. The predicate expressing that the position a variable x is pointing to contains the symbol a is denoted by $Q_a(x)$. For $\Sigma' \subset \Sigma$ we use $Q_{\Sigma'}(x)$ as an abbreviation for the disjunction over $a \in \Sigma'$ of all $Q_a(x)$. The set of words defined by a formula Φ is $L(\Phi) := \{w \in A^* | w \models \Phi\}$. The nonemptiness problem of $L(\Phi)$ is identical to the satisfiability problem of Φ over words.

As usual, a formula Φ with set $\mathcal{V}$ of free variables is interpreted over words as structures $w = (a_1, \mathcal{V}_1)(a_2, \mathcal{V}_2)\cdots(a_n, \mathcal{V}_n)$ such that $\mathcal{V}$ is the union of the $\mathcal{V}_i$ and that the $\mathcal{V}_i$ are pairwise disjoint. A letter $(a, \emptyset)$ is simply denoted by a. If a word w has the set of free variable $\mathcal{V}$ we express this shortly by $w \in \Sigma^* \times \mathcal{V}$. If $w = (a_1, \mathcal{V}_1)(a_2, \mathcal{V}_2)\cdots(a_n, \mathcal{V}_n) \in \Sigma^* \times \mathcal{V}$ and $x \notin \mathcal{V}$ then $w_{x=i}$ denotes the word $(a_1, \mathcal{V}_1)\cdots(a_{i-1}, \mathcal{V}_{i-1})(a_i, \mathcal{V}_i \cup \{x\})(a_{i+1}, \mathcal{V}_{i+1})\cdots(a_n, \mathcal{V}_n) \in \Sigma^* \times (\mathcal{V} \cup \{x\})$.

If $\mathcal{X}$ is a set of quantifier types (eg.: first order) and $\mathcal{P}$ a set of numerical predicates (possibly not containing the order predicate $<$) we denote by $\mathcal{X}[\mathcal{P}]$ the set of all formulae built over the elements of $\mathcal{P}$ and the $Q_a(.)$ predicates as atomic formulae by conjunction, negation and quantification using quantifier types from $\mathcal{X}$. By $\mathcal{L}(\mathcal{X}[\mathcal{P}])$ we denote the class of all languages defined by $\mathcal{X}[\mathcal{P}]$ formulae.

2.1.1. Numerical predicates Formulae not using the $Q_a(.)$ predicates define numerical predicates. It is possible to define in $FO[<]$ the following numerical predicates: $=, \neq, >, \leq, \geq, +1$ (successor), -1 (predecessor), MIN (first position), and MAX (last position). Further on, $<$ can be defined in $FO[+]$, where $+$ is the addition predicate $x + y = z$. We sometimes use expressions like $Q_a(x + y)$ as abbreviation for $\exists_z\ x + y = z \wedge Q_a(z)$. Similarly, $\forall_y\ \exists_{1 \leq x \leq y}\ \Phi$ means $\forall_y\ \exists_x\ 1 \leq x \wedge x \leq y \wedge \Phi$.

Barrington et al. showed that in the presence of the order predicate first order quantifiers can define addition and multiplication by use of the BIT predicate and vice versa ([**?**]).

Lee showed that addition is definable by $FO[<, *]$ formulae ([**?**]).

2.1.2. Majority Quantifiers We will use Maj to denote the majority quantifier. $w \models Maj_x\ \Phi$ is fullfilled iff the number of all $1 \leq i \leq |w|$ such that $w_{x=i} \models \Phi$ is larger than $\lfloor |w|/2 \rfloor$. The majority quantifier rejects in case of a draw. If $\tilde{Maj}$ denotes the weak majority quantifier which accepts in case of a draw we have the deMorgan-like relation $\tilde{Maj}_x \Phi = \neg Maj_x \neg \Phi$.

In the case of majority quantifiers it makes a difference whether we quantify over single variables or over pairs. If Maj_2 denotes the majority quantifier qnatifying over pairs, Barrington et al. showed $\mathcal{L}(FO + Maj_2[<]) = \mathcal{L}(FO + Maj[<, BIT]) = \mathcal{L}(FO + Maj[<, +, *])$ ([**?**]).

Lautemann et al. showed that all numerical predicates definable by $Maj[<]$-formulae are even definable by $FO[+]$ formulae. As a consequence they got:

Theorem 2.1 (Lautemann et al. ([?])

$$\mathcal{L}(Maj[<, +, *]) \not\subset \mathcal{L}(Maj[<])$$

2.1.3. Counting Quantifiers The counting quantifier is denoted by$\exists_x^{=y}$. $w \models \exists_x^{=y}\ \Phi$ is fullfilled iff there are exactly j positions $1 \leq i \leq |w|$ such that $w_{x=i} \models \Phi$ where j is the numerical value of variable y, i.e.: y points to the j.-th symbol of w.

Let $k \geq 2$ be a positive integer. We will use Mod^k to denote the counting quantifier modulo k. $w \models Mod_x^k\ \Phi$ is fullfilled iff the number of all $1 \leq i \leq |w|$ such that $w_{x=i} \models \Phi$ is divisible by k. Straubing denotes this quantifier by $\exists^{(k,o)}$.

2.2. circuits

The reader is assumed to be aquainted with language classes defined by uniform circuit classes as they are presented by Barrington et al. ([**?**]). In particular we will use the following relations:

$$\mathrm{AC}^0 = \mathcal{L}(FO[<, +, *]) \text{ and}$$
$$\mathrm{TC}^0 = \mathcal{L}(FO + Maj[<, +, *]).$$

3. Simulation of first order quantifiers

The simulation of and- and or-gates by majority gates in circuits is very simple by adding inputs which are set to the constants one for or-gates and to zero for and-gates. If we translate this method to formulae using majority quantifiers this would mean to modify the input word. Neither we get a direct simulation if we get rid of first order gates in a circuit and then transform the resulting circuit into a formula since the known translations from circuits to logic expressions use first order quantifiers.

We start our investigations by giving some basic construction which are further used throughout the paper. We first express the numerical predicates $i \leq \lceil n/2 \rceil$, $i > \lfloor n/2 \rfloor$, and the fact that the length of the input is odd by majority quantifiers:

Lemma 3.1 **a)** $x \leq \lceil MAX/2 \rceil \iff Maj_y y \geq x$

b) $x > \lfloor MAX/2 \rfloor \iff Maj_y y \leq x$

c) $|w|$ *is odd* $\iff w \models Maj_x\ Maj_y\ x \leq y \iff w \models Maj_x\ Maj_y\ x \geq y$

In this way it is possible to define the predicates $i = \lfloor n/2 \rfloor$ and $i = \lceil n/2 \rceil$.

Hence, from now on we are free to use quantifiers like $\forall_{x > \lfloor n/2 \rfloor}\ \Phi$ as an abbreviation for $\forall_x\ (\neg Maj_y\ y \leq x) \vee \Phi$.

We now can express first order quantifiers directly without using nonregular predicates like the BIT-predicate.

Theorem 3.2 *For each formula Φ in $Maj[<]$ there is a formula in $Maj[<]$ equivalent to $\exists_x\ \Phi$ over words.*

Proof: The simulation is done by asking whether there is a position $x > \lfloor n/2 \rfloor$ or one that is not larger than $\lceil n/2 \rceil$ satisfying Φ. Observe that this works both for an odd and even length of the input.

$$\exists_x\ \Phi \iff$$
$$(Maj_x[x \leq \lfloor MAX/2 \rfloor \vee (x > \lfloor MAX/2 \rfloor \wedge \Phi)]) \vee$$
$$(Maj_x[x > \lceil MAX/2 \rceil \vee (x \leq \lceil MAX/2 \rceil \wedge \Phi)])\,. \quad \square$$

Corollary 3.3 $\mathcal{L}(FO + Maj[<]) = \mathcal{L}(Maj[<])$

Remark 3.4 *If the quantifier depth of $\exists_x \Phi$ is at least 2, than the quantifier depth in the construction of Theorem* **??** *does not grow. On the other hand, the number of quantifiers is doubled and then increased by 4.*

In the following we demonstrate by three examples a basic method used throughout of this paper to count numbers of positions fullfilling certain properties. The idea underlying this *Equivalence Technique* is the fact that for arbitrary $i, j < \lceil n/2 \rceil$ we have $i \geq j$ iff for all k we have that $j + k > \lfloor n/2 \rfloor$ implies $i + k > \lfloor n/2 \rfloor$.

Example 3.5 *If x, y, z point to positions in the left half of the input word and if $x < y$ is fullfilled, we can test whether*

the number of symbols in subalphabet Σ' between positions x and y of the input is at least z, by the formula

$$\forall_{z'>\lfloor MAX/2\rfloor}\,[Maj_\mu\,(\mu \le z \vee \mu \ge z')] \implies$$

$$[Maj_\mu\,((x \le \mu \le y \wedge Q_{\Sigma'}(\mu)) \vee \mu \ge z')]$$

In the previous example we could have tested for equality by using equivalence instead of implication in the characterizing formula. The following example describes a formula to express the equation $x + x = y$ if $y \neq MAX$:

Example 3.6 *Let $w \in \Sigma^* \times \{x, y\}$ be a word with occurences of the variables x and y and let w fullfil $w \models x < y \wedge y < MAX$. That is, $w = w_1(a, x)w_2(b, y)w_3$ for some $a, b \in \Sigma, w_1, w_2 \in \Sigma^*$ and $w_3 \in \Sigma^+$. Then the following formula expresses the fact that $|w_1| = |w_2|$, i.e.: that $w \models x + x = y$ holds:*

$$\Phi_{??} := \forall_{z>y}\,(Maj_\mu\ \mu \ge z \vee \mu \le x)$$

$$\iff (Maj_\mu\ \mu \ge z \vee (x < \mu \le y))$$

.

Hence we can express in $Maj[<]$ whether a number is even or odd. Thus we can use henceforward an expression like $\forall_{y:even}\ \Phi$ as an abbreviation for $\forall_y\ (\exists_x\ x + x = y \Rightarrow \Phi)$.

The next example shows how to express the *left* and *right half counting quantifier*.

Example 3.7 **a)** *Let $\exists_x^{=y,l}\ \Phi$ denote the quantifier which is true iff there are exactly j positions $1 \le i \le \lfloor n/2\rfloor$ which fullfil $w_{x=i} \models \Phi$ where j is the numerical value of y.*

For each $Maj[<]$ formula Φ there is a $Maj[<]$ formula Ψ which is equivalent to $\exists_x^{=y,l}\ \Phi$. Then

$$\exists_x^{=y,l}\ \Phi \iff$$

$$\big(\forall_{z>\lfloor n/2\rfloor}\ \forall_{\mu\ge z}\ Maj_\nu\ (\nu \le y \vee \nu \ge \mu)$$

$$\iff\ Maj_\nu(\nu \le \lfloor MAX/2\rfloor \wedge \Phi) \vee \nu \ge \mu)$$

.

b) *Let $\exists_x^{=y,r}\ \Phi$ denote the quantifier which is true iff there are exactly j positions $1\lfloor n/2\rfloor + 1 \le i \le n$ which fullfil $w_{x=i} \models \Phi$ where j is the numerical value of y.*

For each $Maj[<]$ formula Φ there is a $Maj[<]$ formula Ψ which is equivalent to $\exists_x^{=y,l}\ \Phi$. Then if the numerical value of y is not larger than $\lceil n/2\rceil$ we have

$$\exists_x^{=y,r}\ \Phi \iff$$

$$\big(\forall_{z\le\lfloor n/2\rfloor}\ \forall_{\mu\le z}\ Maj_\nu\ (\nu \ge MAX + 1 - y \vee \nu \le \mu)$$

$$\iff\ Maj_\nu(\nu \ge \lfloor MAX/2\rfloor + 1 \wedge \Phi) \vee \nu \le \mu)$$

.

4. Addition is definable in $Maj[<]$

In this section we deal with the numerical predicate $x + y = z$, i.e.: of adding two small numbers. Let w be a word in $\Sigma^* \times \{x, y, z\}$, i.e. w has occurences of the variables x, y, and z. Let us express the fact that $x + y = z$ by a $FO + Maj[<]$-formula (and because of Lemma **??** by a $Maj[<]$-formula). By $x + y = z$ we mean the fact, that if e.g. $w = w_1(a, x)w_2(a, y)w_3(a, z)w_4$ and $i := |w_1| + 1, j := |w_1w_2| + 2$ and $k := |w_1w_2w_3| + 3$ then $i + j = k$.

Addition is easily expressible with $\exists_x^{=y}$ quantifiers. In the following we show that majority quanitifier are enough to do this job. We proceed stepwise:

- The easiest case for us is when the input word w fullfils $w \models (x = y) \wedge z = MAX$ because we only have to test whether w has even length n and the value of x (and hence of y) equals $n/2$ which is equivalent with $x \le \lfloor MAX/2\rfloor$ and $x + 1 > \lceil MAX/2\rceil$. Hence it is represented by the formula:

  $$\Phi_1(x) := \neg[Maj_\mu\ \mu \ge (x + 1)] \wedge \neg[Maj_\mu\ \mu \le x]$$

- We now deal with the case that z still points to the MAX-position, but x and y have different values: Case $x \neq y$ and $z = MAX$: Subcase 1: $x < y$: The following formula checks $x = MAX - y$ by checking whether $x + i > \lfloor n/2\rfloor$ if and only if $MAX - y > \lfloor n/2\rfloor$. If x and y do not fullfil this relation we find counterexamples $x < z' \le z'' \le y$ such that $x + z'' - z' \neq MAX - y + z'' - z'$. Thus, $\Phi_2(x, y) :=$

  $$\forall_{x<z',z''\le y}\,(Maj_\mu[\mu \le x \vee z' \le \mu \le z''] \iff$$

  $$Maj_\mu[y < \mu \le MAX \vee z' \le \mu \le z''])$$

  Subcase 2: $y < x$: Use $\Phi_2(y, x)$.

- The case $x = y$ and $z < MAX$ has been dealt with in example **??**: Hence we define: $\Phi_3(x, z) :=$

  $$\forall_{z'>z}\,(Maj_\mu\ \mu \ge z' \vee \mu \le x) \iff$$

  $$(Maj_\mu\ \mu \ge z' \vee (x < \mu \le z))$$

  .

- Finally, we treat the most general case $x \neq y$ and $z < MAX$: We are left with the two subcases:

  Subcase 1: $x < y$: This is just a small generalisation of the case treated in formula Φ_2. Thus, $\Phi_4(x, y, z) :=$

  $$\forall_{x<z'\le z''\le z}\forall_{z'''>z}$$

  $$(Maj_\mu\ \mu \le x \vee z' \le \mu \le z'' \vee \mu \ge z''')$$

  $$\iff (Maj_\mu\ y < \mu \le z \vee z' \le \mu \le z'' \vee \mu \ge z''')$$

  Subcase 2: $y < x \wedge z < MAX$: Use $\Phi_4(y, x, z)$.

Combining these cases we can now express $x + y = z$ by $Maj[<]$ formulae:

Theorem 4.1 *Let w be a word in $\{a\}^* \times \{x, y, z\}$. Then $x + y = z$ holds if and only if $w \models \Phi_+(x, y, z)$, where Φ_+ is defined by:*

$$\Phi_+(x,y,z) := (x = y \wedge z = MAX \wedge \Phi_1(x)) \vee$$
$$(x < y \wedge z = MAX \wedge \Phi_2(x,y)) \vee$$
$$(x > y \wedge z = MAX \wedge \Phi_2(y,x)) \vee$$
$$(x = y \wedge z < MAX \wedge \Phi_3(x,z)) \vee$$
$$(x < y \wedge z < MAX \wedge \Phi_4(x,y,z)) \vee$$
$$(x > y \wedge z < MAX \wedge \Phi_4(y,x,z))$$

□

It should be noted that Lautemann et al. also expressed addition using only the order predicate, but they used a specially designed context-free quantifier and not pure majorities ([**?**, **?**]).

By Lautemann et al. we know that all numerical predicates definable in by $Maj[<]$, i.e.: those not using the Q-predicates, are definable in $FO[+]$ and hence can only describe semilinear sets. Since we now can represent addition by $Maj[<]$ we have:

Corollary 4.2 *The numerical predicates definable in $Maj[<]$ are exactly those definable in $FO[+]$.*

By the results of the previous section we know that

Corollary 4.3 $\mathcal{L}(FO[+]) \subset \mathcal{L}(FO + Maj[<,+]) = \mathcal{L}(Maj[<])$.

4.1. Simulation of Counting Quantifiers

A consequence of Theorem **??** is the expressibility of the counting quantifier, which answers a question posed by Schweickardt ([**?**])[1].

Corollary 4.4 *For arbitrary Φ we have:*

$$\exists_x^{=y} \Phi \iff$$

$$\left(\exists_{y_1,y_2 \le \lfloor MAX/2 \rfloor + 1} y_1 + y_2 = y \wedge \exists_x^{=y_1,l} \Phi \wedge \exists_x^{=y_2,r} \Phi\right),$$

where $\exists_x^{=y,l}$ and $\exists_x^{=y,r}$ have been conststructed in Example **??**.

Since $\mathrm{TC}^0 = \mathcal{L}(FO + \exists^{=y}[<,+,*])$we get as another consequence:

Corollary 4.5 *$TC^0 = \mathcal{L}(Maj[<,*])$*

1 In the notation of Schweickardt we get $FOunM[<] = FOunM[<,+] = FOunC[<]$

As another consequence, we can express for each fixed k the modular counting quantifier Mod_x^k in $Maj[<]$. For instance, we have

$$Mod_x^3\ \Phi \iff \exists_{z,y}\ z + z + z = y\ \wedge\ \exists_x^{=y}\ \Phi.$$

Hence the parity language is in $\mathcal{L}(Maj[<])$:

Corollary 4.6 $\{w \in \{a,b\}^* | \#_a(w) is\ \ odd\} = L(Mod_x^2\ Q_a(x)) \in \mathcal{L}(Maj[<])$.

Hence $\mathcal{L}(Maj[<])$ is not contained in $\mathcal{L}(FO[+])$ or $\mathcal{L}(FO[+,*])$.

5. Order is not definable in $Maj[+]$

We now indicate how to proof that the order predicate is not definable by addition and multiplication. The main idea is that predicates like $x + y = z$ or $x * y = z$ are *simple* in comparison with the order predicate: given x and y there is exactly one value for z which fullfils $x+y = z$ or $x*y = z$. In contrast, for given x there are in general many different values for y which fullfil $x < y$.

For the proof we use the following notation: if Φ is a formula with free variables $\{x_1, x_2, \cdots, x_k\}$ and $w \in \Sigma^* \times \{x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_k\}$ of length $n := |w|$ then $T_{\Phi,x_i}(w) := (w_{x_i:=1} \models \Phi, \cdots, w_{x_i:=n} \models \Phi) \in \{0,1\}^n$.

For a positive integer r we say that a 0-1-vector v is r^+-simple if $\#_1(v) \le r$ and v is r^--simple if $\#_0(v) \le r$.

If $\Phi(x_1, \cdots, x_k)$ is a numerical predicate over k free variables (i.e. not using the $Q_a(.)$ predicates), then we say that Φ is *r-simple* iff for all $1 \le i \le k$ and all $w \in a^* \times \{x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_k\}$ the vector $T_{\Phi,x_i}(w)$ is r^+-simple or if all these vectors are r^--simple. We call Φ *simple* iff Φ is r-simple for some positive integer r. If Φ is r-simple, then so is $\neg\Phi$.

In the following we proof that simple numerical predicates together with majority quantifiers can only define simple predicates. The proof is inspired by the methods used in Lautemann et al. ([**?**] Theorem 4.16).

Theorem 5.1 *Let $\mathcal{N}$ be a set of simple numerical predicates. Then every numerical predicate definable in $Maj[\mathcal{N}]$ is simple.*

Proof: Let Φ be a $Maj[\mathcal{N}]$ formula not using the $Q_a(.)$ predicates. The proof that Φ is simple is done via induction over the term structure of Φ.

- $\Phi \in \mathcal{N}$: if Φ is atomic then Φ is simple by the assumptions.
- $\Phi = \neg\Psi$: the negation of a simple predicate is simple.
- $\Phi = \Psi_1 \wedge \Psi_2$: Let Ψ_j be r_j-simple. Since $T_{\Phi,x_i}(w)$ is the bitwise conjunction of $T_{\Psi_1,x_i}(w)$ and $T_{\Psi_2,x_i}(w)$ is is easy to see that Φ is $r_1 + r_2$-simple.

$\Phi = Maj_z\ \Psi$: by induction Ψ is t-simple for some positive integer t. If the variable z doesn't occurr freely in Ψ then Φ is equivalent to Ψ and we are done. Otherwise we have that $T_{\Psi,z}(w_{x_i:=j})$ is t^+-simple for every w or t^--simple for every w. Hence $L(May_z\ \Psi)$ is either finite or cofinite since it contains either all words of length larger than $2t$ or only words of length bounded by $2t$. Hence Φ is $2t$-simple.

□

Since addition and multiplication[2] are simple numerical predicates, while the order predicate is not, we get the following consequence:

Corollary 5.2 *The order predicate $<$ is not definable in $Maj[+,*]$.*

6. Properties of $Maj[<]$

In this section we investigate the class $\mathcal{L}(Maj[<])$ under three aspects. First we consider the role of numerical predicates. Then it is shown, that the emptiness problem (which is here the satisfiability problem of $Maj[<]$ formulae over words) is undecidable. Finally, a close relationship to TC^0 languages is exhibited.

6.1. Crane Beach properties of majority logics

Barrigton et al. investigated the role of numerical predicates and introduced the *Crane Beach Conjecture* ([**?**]). A language $L \subseteq \Sigma^*$ is said to have a neutral letter if there is a $c \in \Sigma$ such that for all $x, y \in \Sigma^*$ we have $xy \in L$ iff $xcy \in L$.

If $\mathcal{X}$ is a logic and $\mathcal{P}$ a set of numerical predicates we say that $\mathcal{X}[<,\mathcal{P}]$ fullfils the *Crane-Beach-Condition* iff every language L in $\mathcal{L}(\mathcal{X}[<,\mathcal{P}])$ which has a neutral letter is a member of $\mathcal{L}(\mathcal{X}[<])$

As a consequence of the results of Lautemann et al. we get:

Proposition 6.1 *The Crane Beach condition fails for $Maj[<,+,*]$.*

Proof: we know by [**?**] $L := \{a^{n^2} | n \geq 1\}$ is in $\mathcal{L}(Maj[<,*]) \setminus \mathcal{L}(Maj[<])$. Now let L' be the language L shuffled with a neutral letter c. Obviously, L' is in uniform $TC^0 = \mathcal{L}(Maj[<,*])$. On the other hand, we know by the previous lemma, that we can express first-order quantifiers by majority quantifiers. Hence we can intersect any language subset of some Σ^* with the set Σ'^* for any subalphabet Σ'. Since $L = L' \cap a^*$ the language L' cannot be a member of $\mathcal{L}(Maj[<])$. □

2 If the variables would range over $\{0, \cdots, n-1\}$ instead of $\{1, \cdots, n\}$ multiplication would no longer be simple since $0 \cdot i = 0$ for all i. But the following expresibility result still would hold.

Observe that this counterexample is over a binary alphabet. In contrast, $FO[<,\mathcal{P}]$ fullfils the crane beach condition with respect to languages over a binary alphabet for arbitrary set $\mathcal{P}$ of numerical predicates ([**?**]).

On the other hand we have $Maj[<,+] = Maj[<]$ and hence

Proposition 6.2 *$Maj[<,+]$ fullfils the Crane Beach condition.*

6.2. Satisfiability is undecidable

In this subsection it will be shown that the satisfiability of $Maj[<]$ formulae is undecidable. It is possible to do this directly, e.g. by a reduction from the emptiness problem for two-counter automata. Instead, in the following this is shown for $FO[+]$ formulae. This result stresses the importance of the $Q_a(i)$ predicates, since without them the decidability follows from the decidability of Presburger Arithmetic.

Lemma 6.3 *Satisfiability for $FO[+]$ over words is undecidable.*

Proof: The predicates $<, +1, MIN, MAX$ are definable in $FO[+]$ and thus can be used freely. We use the undecidability of the emptiness problem of deterministic linear bounded automata (DLBA). We consider the configurations of an accepting computation of some DLBA A. The symbol at position i in configuration at time $t+1$ is uniquely determined by the three symbols at position $i-1, i$, and $i+1$ at time t. We assume the positions 0 and $n+1$ to contain a special marker symbol \$. Since the simulated machine is deterministic there is mapping $f : \Gamma^3 \longrightarrow \Gamma$ which determins for every triple of possible symbols at some position $i-1, i, i+1$ the validsymbol at position i in next configuration. Now we consider the language of all valid computations $L_A \subset \$(\Gamma^*\$)^*$. We have $w \in L_A$ if and only if w fulfills the following conditions:

1. w starts and ends with a \$-symbol.
2. There is a natural number i such that between any two adjacent \$-symbols there are precisely i non-\$-symbols, That is $xw = \$w_1\$w_2\$\cdots w_k\$$ for some k and some $w_j \in \Gamma^i$.
3. w_1 is the coding of an initial configuration of A beginning with the starting state q_0.
4. w_k is the coding of an accepting configuration of A containing the accepting state q_f.
5. For each $1 \leq j < k$ w_{j+1} is the coding of the successor configuration of the configuration coded by w_j.

This can be expressed by the following formula:

$$Q_\$(Min) \wedge Q_\$(Max) \wedge$$

$$\exists_x \Big(\forall_y (Q_\$(y) \Rightarrow y = Max \vee (Q_\$(y+x) \wedge$$

$$\forall_{y<z<y+x} \ \neg Q_\$(z)) \wedge$$

$$\exists_y \ (y = MIN + 1 \wedge Q_{q_0}(y)) \wedge$$

$$\exists_{y,z} \ (y + x = MAX \wedge y < z < MAX \wedge Q_{q_f}(z)) \wedge$$

$$\bigwedge_{(A,B,C)\in\Gamma^3} (Q_A(z-1) \wedge Q_B(z) \wedge Q_C(z+1)$$

$$\Rightarrow Q_{f(A,B,C)}(z+x)\Big)$$

□

Since the methods proving Corollary **??** were consstructive we get:

Corollary 6.4 *Satisfiability is undecidable for* $Maj[<]$ *over words.*

Observe that the satisfiability of $FO[<]$ formulae is decidable by their constructive equivalence to starfree regular sets.

The results of this subsection indicate that $\mathcal{L}(FO[+])$ behave like a typical low–level complexity class: the morphic images of $\mathcal{L}(FO[+])$ are the recursively enumerable sets and the nonerasing morphic images are contained in *NP* and contain *NP*-complete problems.

6.3. Translating from TC^0 to $\mathcal{L}(Maj[<])$

In this section the relationship between TC^0 and $\mathcal{L}(Maj[<])$ will be investigated and a translational result between these two classes established. Since $TC^0 = \mathcal{L}(Maj[<,*])$ this means to simulate the multiplication predicate $x * y = z$. We do this by attaching to the input word an advice string which provides enough information to do multiplication.

Construction 6.5 *Let* Σ *be a finite alphabet and* $0, 1, \$$, *and* c *be new symbols not in* Σ. *For every nonnegative integer* n *we define the following words: let* a_{ij} *be 1 iff* i *divides* j, *and 0 otherwise for* $1 \leq i, j \leq n$. *Set* $M_{ni} := a_{i1}a_{i2}\cdots a_{in}$ *and* $M_n := \$M_{n1}\$M_{n2}\$\cdots\$M_{nn}\$$. *For* $w \in \Sigma^*$ *of length* $n := |w|$ *define the transformation* $T : \Sigma^* \longrightarrow \Sigma^*\{0,1,\$\}^*c^*$ *by* $T(w) := wM_{|w|}c^{|M_{|w|}|} \in \Sigma^*\{0,1,\$\}^+c^+$. *In particular, the empty word is mapped to* $\$c$. *Then* $T(w)$ *is of length* $2n^2 + 3n + 2$ *for* $n := |w|$.

Lemma 6.6 *The set* $T(\Sigma^*)$ *is in* $\mathcal{L}(FO[+])$

Proof: Using the abbreviation explainend in the preliminaries we will build a $FO[+]$ formula $\Psi_{??}$ expressing the following facts, which characterize the words in $T(\Sigma^*)$: There is aposition i in the input word to which the variable x points such that the input word is in $\Sigma^{i-1}\$\{0,1,\$,c\}^*$. We then demand that between every two \$-symbols there is a $0/1$-word of length $i-1$ and that this word is of the form $(0^j1)^*0^*$. And finally, we require, that this word, if it is not the last one, is followed by one beginning with $0^{j+1}1$. Thus, the formula has the following form: $\Psi_{??}$:=

$$\exists_x \ \Big(\forall_{y<x} \ Q_\Sigma(y) \wedge Q_\$(x) \wedge \forall_{y>x} \ \neg Q_\Sigma(y) \wedge$$

$$\forall_y \ Q_\$(y) \Rightarrow \Big\langle (Q_c(y+x) \wedge \forall_{z\geq y+x} \ Q_c(z)) \vee$$

$$\Big(Q_\$(y+x) \wedge \forall_{1\leq z<x} \ Q_{\{0,1\}}(z) \wedge$$

$$\exists_{z<x} \ Q_1(y+z) \wedge \forall_{1\leq\mu<z} \ Q_0(y+\mu) \wedge$$

$$\forall_{1\leq\mu<x-z} \ (Q_1(y+\mu) \Rightarrow Q_1(y+\mu+z)) \wedge$$

$$(\neg Q_c(y+x+1) \Rightarrow (Q_1(y+x+z+1)) \wedge$$

$$\forall_{1\leq\mu<z+1} \ Q_0(y+x+\mu)\Big)\Big\rangle\Big)$$

□

We now construct the announced tranlation of $FO + Maj[<,*]$ formulae into $FO + Maj[<]$ formulae:

Theorem 6.7 *There is a computable function* $\mathcal{T} : FO + Maj[<,+,*] \longrightarrow FO + Maj[<,+]$ *such that for each* $FO + Maj[<,*]$ *formula* Φ *we have*

$$\forall_w \ w \models \Phi \Longleftrightarrow T(w) \models \mathcal{T}(\Phi).$$

In addition, we have $L(\Phi) \subseteq T(\Sigma^*)$. *Hence* $L(\mathcal{T}(\Phi)) = T(L(\Phi))$.

Proof: The mapping $\mathcal{T}$ will be defined inductively over the termstructure of Φ. During the translation we have to replace the multiplication predicate by some $Maj[<]$ formula using the advice and we have to make quantifiers to work now over the word $T(w)$.

We first describe how to simulate multiplication. Let $T(w) = wM_nc^{|M_n|}$ be the input word and let the variable y point to the \$-symbol in $T(w)$ where the subword M_{ni} begins. Then we can express that fact that x points to position i (in short $x = I(y)$) by the formula $Q_\$(y) \wedge Q_1(y+x) \wedge \forall_{1\leq z<x} \ Q_0(y+z)$. Using this new predicate I we can express the relation $x * y = z$ by a formula $\Xi(x,y,z)$ as follows: we require the existence of a z' such that $x = I(z')$, that is with z' we are looking in the relevant row of the advice. By $Q_1(z'+z)$ we require, that x

divides z and we make sure that $\lfloor z/x \rfloor = y$ holds by requiring that $|\{1 \leq \mu \leq z \mid Q_1(z' + \mu)\}| = y$. The last equation is assured by:

$$\bigforall_{z''} Q_c(z'') \Rightarrow (Maj_\mu\ (\mu \leq y \vee \mu \geq z'') \Leftrightarrow$$
$$Maj_\mu\ ((z' < \mu \leq z' + z \wedge Q_1(\mu)) \vee \mu \geq z''))$$

.

We now define for each $\Phi \in Maj[<,+,*]$ inductively over the term structure of Φ a $FO + Maj[<,+]$ formula $\mathcal{T}'(\Phi)$ which is modelled by $T(w)$ iff Φ is fullfilled by w. The idea is to simulate multiplication with the help of the $\{0,1,\$\}^+c^+$–suffix of $T(w)$ and to keep all predicates working on the Σ^*–prefix. Define $\mathcal{T}'(\Phi)$ by the following case distinctions according to the structure of Φ:

$x < y$: $\mathcal{T}'(\Phi) := x < y$ remains unchanged.

$x + y = z$: $\mathcal{T}'(\Phi) := x + y = z$ remains unchanged.

$x * y = z$: $\mathcal{T}'(\Phi) := \Xi(x,y,z)$ as defined above.

$Q_a(x)$: $\mathcal{T}'(\Phi) := Q_a(x)$ remains unchanged.

$\neg\Psi$: $\mathcal{T}'(\Phi) := \neg\mathcal{T}'(\Psi)$.

$\Psi_1 \wedge \Psi_2$: $\mathcal{T}'(\Phi) := \mathcal{T}'(\Psi_1) \wedge \mathcal{T}'(\Psi_2)$

$\exists_x\ \Psi$: $\mathcal{T}'(\Phi) := \exists_x\ Q_\Sigma(x) \wedge \mathcal{T}'(\Psi)$

$\forall_x\ \Psi$: $\mathcal{T}'(\Phi) := \forall_x\ \neg\, Q_\Sigma(x) \vee \mathcal{T}'(\Psi)$

$Maj_x\ \Psi$: $\mathcal{T}'(\Phi) := Maj_x\ (Q_c(x) \vee (\mathcal{T}'(\Psi) \wedge Q_\Sigma(x)))$

The idea of the last entry is that exactly half of the positions in the advice carry the symbol c. Thus a majority for Φ over w is transformed into a majority for $\mathcal{T}'(\Phi)$ over $T(w)$. Our anticipated formula is now defined by

$$\mathcal{T}(\Phi) := \mathcal{T}'(\Phi) \wedge \Psi_{\mathbf{??}}.$$

□

Because of $L(\mathcal{T}(\Phi)) = T(L(\Phi))$ we have

$$L(\Phi) \neq \emptyset \Longleftrightarrow L(\mathcal{T}(\Phi)) \neq \emptyset$$

i.e.: Φ is satisfiable iff $\mathcal{T}(\Phi)$ is satisfiable. Hence we get another proof for the undecidability of the satisfiability problem of $Maj[<]$ formulae.

If $T(L) \in \mathcal{L}(Maj[<]) \subset \mathrm{TC}^0$ we can construct a TC^0-circuit accepting L since T is obviously a TC^0-computable function. Hence we have:

Corollary 6.8 *For each language L we have*

$$L \in TC^0 \Leftrightarrow T(L) \in \mathcal{L}(Maj[<]).$$

Observe, that this construction does not work to give a corresponding translation from $FO[+,*]$ into $FO[+]$ since we make essential use of the majority quantifier when simulating multiplication by the $\Xi(x,y,z)$ predicate.

7. Open Questions

In the first order framework it is possibl to represent circuit depth by alternation depth of first oder quantifiers. This doesn't seem to work with majority gates and quantifiers, since these are not idempotent, i.e.: majority over pairs are more powerful than nested majorities over single variables. So the question remains, how to represent circuit depth of threshold circuits in the majority logic framework.

Another question concerns lower bounds. The separation of uniform circuit classes like NC^1 and TC^0 is one of the major open problems of theoretical computer science. These classes are defined with the help of the BIT-predicate, or equivalently with addition and multiplication. What happens if we drop multiplication? It should be possible to separate modular counting from majority and the later from NC^1-circuits (resp. arbitrary finite group quantifiers) if we use only $FO[+]$-uniformity (resp. $FO + \mathcal{X}[<,+]$ logic).

Acknowledgements

I thank Andreas Krebs for pointing out Corollary **??** to me. Further I thank Pascal Tesson and Denis Therien for fruitful discussions about the topic.

References

[1] D.A. Barrington, N. Immerman, and H. Straubing. On uniformity within NC^1. *J. Comp. System Sci.*, 41:274–306, 1990.

[2] D.A. Barrington, Immerman N., Lautemann C., Schweikardt N., and Therien D. The crane beach conjecture. In *In IEEE Symposium on Logic in Computer Science*, pages 187–196, 2001.

[3] C. Lautemann, P. MCKenzie, T. Schwentick, and H. Vollmer. The descriptive complexity approach to logcfl. *J. Comp. System Sci.*, 62:629–652, 2001.

[4] C. Lautemann, T. Schwentick, and D. Therien. Logics for context-free languages. In *In the 8th International Worship on Computer Science Logic*, number 933 in LNCS, pages 205–216. Springer, 1994.

[5] Troy Lee. Is multiplication harder than addition? arithmetical definability over finite structures. Master's thesis, Institute for Logic, Language, and Computation, 2001. (in Amsterdam).

[6] N. Schweickardt. On the Expressive Power of First–Order Logic with Built–In Predicates. Dissertation, Johannes Gutenberg Universität in Mainz, 2001.

[7] H. Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser, 1994.