Electronic Colloquium on Computational Complexity, Report No. 173 (2011)

# Regular Languages in MAJ[<] with three variables

Christoph Behle Andreas Krebs

December 22, 2011

{behlec, krebs}@informatik.uni-tuebingen.de
Universität Tübingen
Wilhelm-Schickard-Institut
Sand 13, D-72076 Tübingen

**Abstract**

We consider first order logic over words and show FO + MOD[<] is contained in MAJ[<] with three variables. It is known that for the classes FO[<], FO + MOD[<], FO + GROUP[<] three variables suffice. In the case of MOD[<] even two variables are sufficient.

As a consequence we know that if $TC^0 \neq NC^1$ then for every regular language describable in MAJ[<] three variables are sufficient.

## 1 Introduction

The power of the first order logic classes FO[<] and FO + MOD[<] is well understood. The languages in FO[<] are exactly the star-free languages by [Sch65] and [MP71]. The languages in FO+MOD[<] are exactly those regular languages where the syntactic monoids do not contain non-solvable groups [STT95]. For both classes it was shown [Kam68], [ST03] that three variables suffice to describe the languages they can describe. We show in this paper that the regular languages where the syntactic monoids do not contain non-solvable groups are contained in MAJ[<] with only three variables. In view of the known results this is not surprising but it did not follow immediately.

A direct translation from a three variable FO + MOD[<] sentence seems to fail, since an extra variable is needed to simulate a $\mathrm{Mod}^{r,q}\ x$ quantifier in FO+MAJ[<]. Further, we show our result for the class MAJ[<] not FO + MAJ[<]. While both classes coincide by [Lan04] the constructions require extra variables. Our proof is based on the observation that two variable logic is sufficient if we allow an additional first order quantifier. This quantifier is similar to the Until operator in LTL logic. A line of research lead to the result that aperiodic regular languages are exactly those that can be described in LTL logic, see [Wil99]. The classes FO[<] and FO + MOD[<] with only two variables are proper subclasses [TW98]. It is known that FO[<] with two variables describe the same languages as LTL without the Until operator [EVW02].

We also want to mention that this can be seen in by different argument. The Krohn-Rhodes theorem shows that every finite monoids divides a block product of monoids which are either simple groups or the monoid $U_1$. The block product is not associative and the theorem uses the strong bracketing. The theorem can be also stated using the weak bracketing when one allows the monoid $U_2$ instead of $U_1$. A quantifier corresponding to $U_2$ in the same way as the existential and universal quantifiers correspond to $U_1$ is similar to the until quantifier.

We define now a new first order quantifier $\mathrm{U}\ x\ \langle \phi_1, \phi_2 \rangle$ that corresponds to the Until operator.



ISSN 1433-8092

**Definition 1.1** (Until Quantifier)**.** The U $x\ \langle \phi_1, \phi_2 \rangle$ quantifier models $w$ iff there exists $i$ such that $w, i \models \phi_2(x)$ and $w, i' \models \phi_1(x)$ for all $i' < i$

It is clear by the definition that we can express U $x\ \langle \phi_1, \phi_2 \rangle$ in first-order logic by $\exists x\ \phi_2(x) \wedge \forall x'\ x' < x \rightarrow \phi_1(x/x')$. Conversely, $\exists x \phi$ is equivalent to U $x\ \langle \text{true}, \phi \rangle$.

By the results mentioned above it is clear that:

**Proposition 1.2.** *The languages described by* $(\text{U})_2[<]$ *are exactly the languages described by* FO[<].

*The languages described by* $(\text{U} + \text{MOD})_2[<]$ *are exactly the languages described by* FO + MOD[<].

Using this proposition we show how to describe every language in FO+MOD[<] within MAJ[<] with three variables.

# 2 Constructions and Results

We always assume that our input is a word $w = w_1 \dots w_n$ over some alphabet $\Sigma$. The first position has the numerical value 1 and the last position has the value $n$. The construction could be adopted if the positions have numerical value 0 to $n - 1$.

We assume familiarity with first order logic over words and refer to [Str94] for an introduction. In the following we will have the variables $x, y, z$. We denote the numerical values of the positions they point to by $i, j, k$.

The majority quantifier Maj x is defined by $w \models \text{Maj x } \phi(x)$ iff $|\{i \mid w, i \models \phi(x)\}| > n/2$. So $\lfloor n/2 \rfloor$ is the maximal number of positions where the subformula can be true and the majority quantifier is still false. The logic with this quantifier, the query predicate, the usual connectives $\wedge, \vee, \neg$ and the order predicate is denoted by MAJ[<]. The restriction to the case with two or three variables is denoted by $\text{MAJ}_2[<]$ resp. $\text{MAJ}_3[<]$.

The class MAJ[<] is equal to the class FO + MAJ[<, +] and can define the counting quantifier [Lan04].

Let $L$ be a language described by FO + MOD[<], then there is a formula $\phi \in (\text{U} + \text{MOD})_2[<]$ such that $L = L_\phi$. We show by induction on the structure of $\phi$ that there is a formula $\psi$ in MAJ[<] with three variables that describes $L$.

As a first observation we can express existential quantifiers by majority quantifiers with three variables. $\exists x\ \phi(x)$ is equivalent to

$$(\text{Maj x } \phi(x) \vee \neg \text{Maj y } (y \geq x)) \vee (\text{Maj x } \phi(x) \vee \neg \text{Maj y } (y \leq x)) .$$

If there is an $i \leq \lfloor n/2 \rfloor$ with $w, i \models \phi(x)$ then the first part of the formula is valid, if $i > \lfloor n/2 \rfloor$, then the second part is true, and if there is no such $i$, then the formula is never true. Note, that we need only one extra variable and this variable is still available in $\phi(x)$.

Before we start with the proof we introduce some auxiliary (numerical) predicates. The predicate first$(x)$ is true iff $x$ is in the first half of the word, i.e. $w, i \models \text{first}(x) \iff i \leq \lfloor n/2 \rfloor$. This definition is a bit non-symmetric since the second half is one position larger than the first half, if the input length is odd, but we deal with this. Also we have a restricted predicate $\text{mod}^{r,q}(x)$ which is true if $x$ points to the first half and iff $x \equiv r \mod q$, i.e. $w, i \models \text{mod}^{r,q}(x) \iff i \leq \lfloor n/2 \rfloor \wedge i \equiv r \mod q$. We denote this by $\text{mod}^{r,q}(x)$ anyway, since we ensure in following always that $x$ points to the first half.

**Lemma 2.1.** *The predicates* first$(x)$ *and* $\text{mod}^{r,q}(x)$ *are definable in* $\text{MAJ}_3[<]$.

*Proof.* The predicate first$(x)$ can be defined by $\neg \text{Maj y } y \leq x$.

For the $\text{mod}^{r,q}(x)$ predicate we use induction on $q$. Please note $\text{mod}^{r,q}(x)$ is only valid on the first half of the input.

The following formula checks if $x$ points to an even position $i$. This is the case if there exists a variable $y$ pointing to the position $j$ and $2j = i$. So we use an existential

quantifier to place a variable $y$ to a position and check if the number of position before $y$ is the same as the number of positions between $y$ and $x$. In order to check this we use a simple construction. As an example we look at the formula:

$$\begin{aligned}\phi(x) \quad &= \quad \neg\text{Maj y } (y < x) \\ &\quad\ \ \wedge\text{Maj y } (y \leq x)\end{aligned}$$

This formula checks if $x$ points to the position $\lfloor n/2 \rfloor + 1$. Note that we use the subformulas $y < x$ and $y \leq x$ which differ at exactly one position, so the number of positions for $y$ in $y < x$ has to equal $\lfloor n/2 \rfloor$. We will use a similar construction in the following very often.

For $q = 2$ we use the formula:

$$\begin{aligned}\text{mod}^{0,2}(x) = \quad & \quad \text{first}(x) \wedge \\ & \exists y \quad \neg\text{Maj z } (z < y \vee x \leq z \wedge z \leq y + \lfloor n/2 \rfloor) \\ & \quad\ \ \wedge\text{Maj z } (z \leq y \vee x \leq z \wedge z \leq y + \lfloor n/2 \rfloor)\end{aligned}$$

So if $w, i, j$ models the formula inside the existential quantifier then $(j-1)+(j+\lfloor n/2 \rfloor - i + 1) = \lfloor n/2 \rfloor$ (here $z < y$ corresponds to $j - 1$ and $x \leq z \wedge z \leq y + \lfloor n/2 \rfloor$ corresponds to $j + \lfloor n/2 \rfloor - i + 1$), hence $2j = i$. The expression $z \leq y + \lfloor n/2 \rfloor$ can be rewritten as $\neg$Maj x $(y \leq x \wedge x < z)$.

In the case $q = 2$ we placed the variable $y$ at the middle position in general we place $y$ at the position $(q - 1)/q$ relative to $x$. So the positions before $y$ have to be approximately $q - 1$ times the positions between $x$ and $y$. For $q > 2$ we use the following formula:

$$\begin{aligned}\text{mod}^{0,q}(x) = \quad & \quad \text{first}(x) \wedge \\ & \exists y \quad \text{mod}^{0,q-1}(x) \wedge \\ & \quad\ \ \neg\text{Maj z } (z < y \wedge \text{mod}^{0,q-1}(x) \vee x \leq z \wedge z \leq y + \lfloor n/2 \rfloor) \\ & \quad\ \ \wedge\text{Maj z } (z \leq y \wedge \text{mod}^{0,q-1}(x) \vee x \leq z \wedge z \leq y + \lfloor n/2 \rfloor)\end{aligned}$$

As in the previous formula the last two lines induce a linear equation on the position of the variable $y$. This time we only count the positions equivalent to 0 modulo $q - 1$ before $y$, so we count only $1/(q - 1)$ of these positions. So if $w, i, j$ models the formula inside the existential quantifier then $j/(q-1)$ is an integer and $j/(q-1)-1+j+\lfloor n/2 \rfloor - i+1 = \lfloor n/2 \rfloor$, hence $qj/(q - 1) = i$.

Next we define $\text{mod}^{r,q}(x)$ for different $r$ by induction on $r$.

$$\text{mod}^{1,q}(x) = \forall y\ (y + 1 = x \rightarrow \text{mod}^{0,q}(y))$$

Then $x$ either points to the first position, or a position following a position that is 0 modulo $q$. For $r > 1$ we define

$$\text{mod}^{r,q}(x) = \exists y\ (y + 1 = x \wedge \text{mod}^{r-1,q}(y)).$$

As usually, $y + 1 = x$ can be rewritten as $\forall z\ z \leq y \vee x \leq z$. □

We will now predicate an show how we can simulate a modulo quantifier.

**Lemma 2.2.** *Let $\phi(y) = \text{Mod}^{r,q}\ y\ \phi'(x, y)$ be a formula in $(U + MOD)_2$. If there is a formula $\psi'(x, y)$ in $\text{MAJ}_3[<]$ equivalent to $\phi'(x, y)$, then there is a formula $\psi(x)$ in $\text{MAJ}_3[<]$ equivalent to $\phi(x)$.*

*Proof.* We will show how to simulate the $\text{Mod}^{r,q}\ x$ quantifier by a majority formula. Fix a word $w$ and a position $i$ of $w$.

If we show that we can find formulas $\tau_1^{r,q}(x)$ resp. $\tau_2^{r,q}(x)$ that check if $|\{j \leq \lfloor n/2 \rfloor \mid w, i, j \models \phi'(x, y)\}| \equiv r' \mod q$ resp. $|\{j > \lfloor n/2 \rfloor \mid w, i, j \models \phi'(x, y)\}| \equiv r' \mod q$ for

$r' \in \{0, \ldots, q-1\}$, then we can check if $|\{j \mid w, i, j \models \phi'(x,y)\}| \equiv r \mod q$ with a Boolean combination of these formulas.

We construct the formula $\tau_1^{r,q}(x) = \exists z\ \mu_1(x,z) \wedge \mathrm{mod}^{r,q}(z)$, where $w, i, k \models \mu_1(x,z)$ iff $|\{j \leq \lfloor n/2 \rfloor \mid w, i, j \models \phi'(x,y)\}| = k$, and a similar formula for $\tau_2^{r,q}(x)$ later in the proof.

**First half.** In the construction of $\mu_1(x,z)$, we first fix the variable $y$ to the position $\lfloor n/2 \rfloor + 1 + |\{j \leq \lfloor n/2 \rfloor \mid w, i, j \models \phi'(x,y)\}|$ so the positions greater or equal than $y$ do not interfere with the positions where we want to check $\phi'(x,y)$. Then we flip the value to obtain the position of $z$.

Again we use a trick similar to the previous lemma to place a variable $y$ at a certain position. We first state the formula and then show the two equalities induces by the first two lines of the formula and the last two lines.

$$\begin{aligned}\mu_1(x,z) \quad = \exists y \quad & \neg\text{Maj z } (z > y \vee (z \leq \lfloor n/2 \rfloor \wedge \phi'(x,z))) \\ & \wedge\text{Maj z } (z \geq y \vee (z \leq \lfloor n/2 \rfloor \wedge \phi'(x,z))) \\ & \wedge\neg\text{Maj x } (x < z \vee x \geq y) \\ & \wedge\text{Maj x } (x \leq z \vee x \geq y)\end{aligned}$$

Now if $w, i, j, k$ models the formula inside the existential quantifier than $n - j + |\{j \leq \lfloor n/2 \rfloor \mid w, i, j \models \phi'(x,y)\}| = \lfloor n/2 \rfloor$ by the first two lines, so $n - j - \lfloor n/2 \rfloor = |\{j \leq \lfloor n/2 \rfloor \mid w, i, j \models \phi'(x,y)\}|$. And be the last two lines we have $k - 1 + n - j + 1 = \lfloor n/2 \rfloor$ so $z = n - j - \lfloor n/2 \rfloor$, hence $k = |\{j \leq \lfloor n/2 \rfloor \mid w, i, j \models \phi'(x,y)\}|$.

**Second half.** The second half consists of $\lfloor n/2 \rfloor + 1$ letters if $n$ is odd, so we need to account for the extra position separately. If the word length is odd and $w, i, \lfloor n/2 \rfloor + 1 \models \phi'(x,y)$ we correct the modulo count.

$$\begin{aligned}\tau_2^{r,q}(x) \quad = \quad & \exists z\ \mu_2(x,z) \wedge ( \\ & \exists y\ (\neg\text{Maj x } x < y \wedge \neg\text{Maj x } x > y \wedge \phi'(x,y)) \wedge \mathrm{mod}^{r-1,q}(z) \\ & \vee\neg\exists\ y(\neg\text{Maj x } x < y \wedge \neg\text{Maj x } x > y \wedge \phi'(x,y)) \wedge \mathrm{mod}^{r,q}(z))\end{aligned}$$

Here the construction of $\mu_2(x,z)$ is simpler, since we only need to check validity of $\phi'(x,y)$ in the second half, so no need to flip the variable.

$$\begin{aligned}\mu_2(x,z) \quad = \exists z \quad & \neg\text{Maj y } (y > z \wedge y \leq \lfloor n/2 \rfloor \vee (z \leq \lfloor n/2 \rfloor \wedge \phi'(x,z))) \\ & \text{Maj y } (y \geq z \wedge y \leq \lfloor n/2 \rfloor \vee (z \leq \lfloor n/2 \rfloor \wedge \phi'(x,z)))\end{aligned}$$

Now if $w, i, k$ models the formula $\mu_2$ then $\lfloor n/2 \rfloor - k + |\{j > \lfloor n/2 \rfloor \mid w, i, j \models \phi'(x,y)\}| = \lfloor n/2 \rfloor$, and so $k = |\{j > \lfloor n/2 \rfloor \mid w, i, j \models \phi'(x,y)\}|$. □

Next we show how we can simulate the until quantifier.

**Lemma 2.3.** *Let $\phi(y) = \mathrm{U}\ y\ \langle \phi'_1(x,y), \phi'_2(x,y) \rangle$ be a formula in $(\mathrm{U} + \mathrm{MOD})_2$. If there are formulas $\psi'_1(x,y)$ and $\psi'_2(x,y)$ in $\mathrm{MAJ}_3[<]$ equivalent to $\phi'_1(x,y)$ resp. $\phi'_2(x,y)$, then there is a formula $\psi(x)$ in $\mathrm{MAJ}_3[<]$ equivalent to $\phi(x)$.*

*Proof.* Since we can simulate all first-order quantifiers by majority quantifiers we can simply rewrite the until quantifier.

$$\psi(y) = \exists x\ \psi'_2(x,y) \wedge \forall z\ (z < y \implies \psi'_1(z,y))$$

□

Since we have the same predicates, connectives and the same numerical predicate, by induction on the depth of the formula and the previous two lemmas we get our main theorem.

**Theorem 2.4.** *Every languages described by* FO+MOD[<] *can be described by a* MAJ[<] *formula with only three variables.*

Since every regular language outside of FO + MOD[<] is $NC^1$ complete we get:

**Corollary 2.5.** *If* $TC^0 \neq NC^1$*, then every regular languages described by* MAJ[<] *can be described using only three variables.*

# 3 Discussion

We have shown that three variables suffice for all regular language currently known to be in MAJ[<]. On the other hand by [BKR09b] we know that two variables do not suffice. Completely unaffected by our result is the question if $L_{A_5} \in$ MAJ[<], i.e. the language whose syntactic monoid is the alternating group on five elements. We can only say that if $\mathrm{MAJ}_4[<] \cap REG \supsetneq \mathrm{MAJ}_3[<] \cap REG$ this would imply that $TC^0 = NC^1$.

By [BKR09a] it is known that $L_{A_5} \notin$ FO + MOD + $\mathrm{MAJ}_2$[reg]. Could this result be extended to three variables? The same proof idea does not seem to work so this might be hard to show. This classes are strictly weaker than (DLOGTIME)-uniform $TC^0$ so such a result would not imply $TC^0 \neq NC^1$. Hence, this problem might be tractable.

A hindrance seems to be the fact that MAJ[<] = MAJ[<, +]. While we know that for FO[<, +] three and four variables differ on non-regular languages the proof ideas do not work in the presence of the majority quantifier.

# References


[BKR09a] Christoph Behle, Andreas Krebs, and Stephanie Reifferscheid. Non-solvable groups are not in FO+MOD+MÂJ2[reg]. In Adrian Horia Dediu, Armand-Mihai Ionescu, and Carlos Martín-Vide, editors, *LATA*, volume 5457 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 2009.

[BKR09b] Christoph Behle, Andreas Krebs, and Stephanie Reifferscheid. Regular languages definable by majority quantifiers with two variables. In Volker Diekert and Dirk Nowotka, editors, *Developments in Language Theory*, volume 5583 of *Lecture Notes in Computer Science*, pages 91–102. Springer, 2009.

[EVW02] Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. First-order logic with two variables and unary temporal logic. *Inf. Comput.*, 179(2):279–295, 2002.

[Kam68] Johan Anthony Willem Kamp. Tense logic and the theory of linear order. *Ph.D. thesis, University of California, Berkeley*, 1968.

[Lan04] Klaus-Jörn Lange. Some results on majority quantifiers over words. In *IEEE Conference on Computational Complexity*, pages 123–129, 2004.

[MP71] Robert McNaughton and Seymour Papert. *Counter-free automata. With an appendix by William Henneman.* Research Monograph No.65. Cambridge, Massachusetts, and London, England: The M. I. T. Press. XIX, 163 p., 1971.

[Sch65] Marcel Paul Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8(2):190–194, 1965.

[ST03] Howard Straubing and Denis Thérien. Regular languages defined by generalized first-order formulas with a bounded number of bound variables. *Theory Comput. Syst.*, 36(1):29–69, 2003.

[Str94] Howard Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser, Boston, 1994.

[STT95] Howard Straubing, Denis Thérien, and Wolfgang Thomas. Regular languages defined with generalized quanifiers. *Inf. Comput.*, 118(2):289–301, 1995.

[TW98] Denis Thérien and Thomas Wilke. Over words, two variables are as powerful as one quantifier alternation. In *STOC*, pages 234–240, 1998.

[Wil99] Thomas Wilke. Classifying discrete temporal properties. In Christoph Meinel and Sophie Tison, editors, *STACS*, volume 1563 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 1999.



ECCC ISSN 1433-8092

http://eccc.hpi-web.de