Bounded Arithmetic and Descriptive Complexity

Achim Blumensath

Mathematische Grundlagen der Informatik
RWTH Aachen, D-52056 Aachen
blume@i7.informatik.rwth-aachen.de

Abstract. We study definability of languages in arithmetic and the free monoid by bounded versions of fixed-point and transitive-closure logics. In particular we give logical characterisations of complexity classes $\mathscr{C}$ by showing that a language belongs to $\mathscr{C}$ if and only if it is definable in either arithmetic or the free monoid by a formula of a certain logic. We investigate in which cases the bounds of fixed-point operators may be omitted. Finally, a general translation of results from descriptive complexity to the approach described in this paper is presented.

Keywords: descriptive complexity, definability, arithmetic

1 Introduction

Descriptive complexity theory studies the connections between definability and complexity classes (see [1, 4, 5] for an overview). The most common approach originates in finite model theory and yields characterisations of the following form: "Some class $\mathscr{K}$ of finite structures belongs to the complexity class $\mathscr{C}$ if and only if $\mathscr{K}$ is the class of finite models of some sentence of the logic $\mathfrak{L}$." More formally, $\mathscr{K} \in \mathscr{C}$ iff $\mathscr{K} = \text{Mod}(\varphi)$ for some $\varphi \in \mathfrak{L}$. Starting with Fagin's famous characterisation of NPTIME descriptions of most of the common complexity classes have been obtained in this way.

Another equally well developed method is based on function algebras and recursion schemes (see [3] for an overview, or [6] for a formulation in terms of proof theory). It originated in recursion theory with characterisations of the recursive and primitive recursive functions and later on was applied by Cobham to describe the class of polynomial time computable functions.

In the present article we will follow a third approach. We fix a model with universe $\{0,1\}^*$, $\mathbb{N}$, or some other countable set with canonical encoding in $\{0,1\}^*$, and investigate which languages are definable within this model using different logics. This approach has mainly been used in recursion theory so far, for instance to define the arithmetic and analytic hierarchy. To the author's knowledge there are only few characterisations of decidable complexity classes using this method. The Büchi-Bruyère Theorem (see [2] for an overview) states that the p-adic encoding of a set of natural numbers is regular if and only if the set is first-order definable in $(\mathbb{N}, +, V_p)$ where $V_p(x) := p^k$ for the greatest k such that $p^k | x$. Wrathall [7] showed that the class of languages definable by Δ^0_0-formulae in

$(\mathbb{N}, +, \cdot)$ is equal to the linear hierarchy. As mentioned in [6], a characterisation of the polynomial hierarchy is obtained if one adds the operator $x\#y = 2^{\log_2 x \log_2 y}$.

Below we will show that by adding fixed-point or transitive-closure operators those results can be extended to characterise many of the usual complexity classes. The results themselves are unsurprising and mirror those of finite model theory. Indeed, the similarity between both approaches enables us to present a translation from the formalism of finite model theory to definability in the binary tree and vice versa. So, what are the differences between them? First, our formalism seems to be more general since by using other structures than the binary tree—e.g., arithmetic—we can capture different classes such as Exptime or Expspace for which there is no "classical" characterisation. A second point is that depending on the circumstances one approach might be more convenient to work with. For instance, from an an algorithmic point of view the classical approach seems to be more suitable since one can speak about, say, graphs directly instead of having to encode them as words. On the other hand when dealing with languages, e.g., in structural complexity, or when thinking of applications in feasible model theory, our formalism might be of advantage. Finally, by changing the formalism a different set of logical and algebraic methods for the investigation of complexity classes becomes available (although whether this is of any help remains to be seen).

The paper is organised as follows. In the next section we give a short overview of classical descriptive complexity theory and list some results for comparison. Furthermore, we introduce the logics used in the rest of the article.

Section 3 considers various structures of natural numbers and investigates which complexity classes can be characterisations within them. We show how to generalise these results to arbitrary linear orderings of type ω, and study in which cases the bounds of fixed-point operators are really needed.

In Section 4 we turn to the free monoid and show that many classical results can be translated to our approach, and vice versa.

2 Preliminaries

We recall the basic definitions of descriptive complexity theory. For simplicity we will consider only languages over a binary alphabet. In the classical approach each word $w \in \{0,1\}^+$ is represented by the word model

$$\underline{w} := \big(\{0, \dots, |w| - 1\}, <, S, \min, \max, P\big)$$

where S is the successor relation of $<$, min and max are the first and last elements, and P is the set of positions carrying the symbol 1. While in descriptive complexity theory one usually allows classes of arbitrary finite models we will only consider word models in the following.

Let $\mathscr{C}$ be a complexity class. We say that the logic $\mathfrak{L}$ *captures* $\mathscr{C}$ (on word models) iff $\mathscr{C} = \{\, L(\varphi) \mid \varphi \in \mathfrak{L} \,\}$, where $L(\varphi) := \{\, w \in \{0,1\} \mid \underline{w} \models \varphi \,\}$.

Logics capturing complexity classes include first-order logic FO and its extensions by transitive-closure or fixed-point operators, and fragments of second-order logic (see Table 1).

Table 1. Logics capturing complexity classes

Class	Logic
AC^0	FO
Logspace	FO(DTC)
Nlogspace	FO(TC)
Ptime	FO(LFP)
— " —	Σ^1_1-Horn
— " —	SO-Horn
Nptime	Σ^1_1
PH	SO
Pspace	FO(PFP)

(Deterministic) transitive-closure logic FO((D)TC) is obtained from FO by adding the operator

$$[(\mathrm{D})\mathrm{TC}_{\bar{x},\bar{y}}\ \varphi(\bar{x},\bar{y},\bar{z})](\bar{u},\bar{v}).$$

The semantics is defined as follows (where for notational convenience we omitted all references to the structure in question). $[\mathrm{TC}_{\bar{x},\bar{y}}\ \varphi](\bar{a},\bar{b})$ holds iff there are tuples $\bar{a}_0 = \bar{a}$, $\bar{a}_1$, … , $\bar{a}_n = \bar{b}$, $n > 0$, such that $\varphi(a_i, a_{i+1})$ holds for all $i < n$. The deterministic version is defined by

$$\begin{aligned}&[\mathrm{DTC}_{\bar{x},\bar{y}}\ \varphi(\bar{x},\bar{y},\bar{z})](\bar{u},\bar{v})\\ &\qquad\equiv \big[\mathrm{TC}_{\bar{x},\bar{y}}\ \varphi(\bar{x},\bar{y},\bar{z}) \wedge \forall\bar{y}'(\varphi(\bar{x},\bar{y}',\bar{z}) \rightarrow \bar{y}' = \bar{y})\big](\bar{u},\bar{v})\end{aligned}$$

Similarly, in least and partial fixed-point logic FO(LFP) and FO(PFP) one adds the operator

$$[\mathrm{L/PFP}_{R,\bar{x}}\ \varphi(R,\bar{x},\bar{z})](\bar{u})$$

where in the case of LFP, R occurs only positive in φ. To define the semantics consider the operator

$$F(R) := \{\, \bar{a} \mid \varphi(R,\bar{a}) \text{ holds} \,\}.$$

$[\mathrm{LFP}_{R,\bar{x}}\ \varphi](\bar{a})$ holds iff $\bar{a}$ is in the least fixed-point of F, and $[\mathrm{PFP}_{R,\bar{x}}\ \varphi](\bar{a})$ holds iff there is some n such that $F^{n+1}(\emptyset) = F^n(\emptyset)$ and $\bar{a} \in F^n(\emptyset)$.

Finally, denote full second-order logic by SO, existential second-order logic by Σ^1_1, and (existential) second-order horn logic by SO-Horn and Σ^1_1-Horn, respectively. Here, SO-Horn consists of second-order formulae in prenex-normalform where the first-order part is universal, in conjunctive normalform, and each clause contains at most one positive literal $X\bar{x}$ for second-order variables X.

In this article we want to ask which languages can be defined within some fixed structure $\mathfrak{A}$. Of course, in order to do so the universe of $\mathfrak{A}$ should either consists of $\{0,1\}^*$ or we have to choose some encoding of the elements of $\mathfrak{A}$ by words.

Definition 1. *Let $\mathfrak{A}$ be a countable structure and suppose $e : A \rightarrow \{0,1\}^*$ is bijective. Let $\mathscr{C}$ be a complexity class. We say that the logic $\mathfrak{L}$* captures *$\mathscr{C}$* on *$\mathfrak{A}$ iff*

$$\mathscr{C} = \{\, e(\varphi^{\mathfrak{A}}) \mid \varphi(x) \in \mathfrak{L} \,\},$$

where $\varphi^{\mathfrak{A}} := \{\, a \in A \mid \mathfrak{A} \models \varphi(a) \,\}$.

Obviously this definition may be generalised to relations of arbitrary arity. As there are pairing functions definable in all structures considered below the arity can w.l.o.g. assumed to be one. So, for simplicity, we will only deal with this case.

Below we will investigate which classes are captured on several variants of arithmetic and the free monoid. Since the first-order theory of arithmetic is highly undecidable, we can only hope to capture decidable complexity classes by fragments of FO. In particular we will try to ensure that all variables only range over finite sets. The following definition was motivated by the observation that in recursion theory "bounded quantifiers come for free."

Definition 2. *Fix some structure $\mathfrak{A}$. A* bounded guard *on $\mathfrak{A}$ is a quantifier-free formula $\alpha(\bar{x};\bar{y})$ such that for all $\bar{b} \in A^m$ the set $\{\, \bar{a} \in A^n \mid \mathfrak{A} \models \alpha(\bar{a};\bar{b}) \,\}$ is finite. Here, $\bar{x}$ are called the* bounded variables *of α, and $\bar{y}$ are the* free variables *or* parameters *of α.*

The bounded fragment BFO *on $\mathfrak{A}$ is defined like* FO *where all quantifiers are guarded, i.e., of the form $(Q\bar{x}.\alpha)\varphi$ for $Q \in \{\exists, \forall\}$ and some bounded guard α with bounded variables $\bar{x}$.*

For $O \in \{\mathrm{DTC}, \mathrm{TC}, \mathrm{LFP}, \mathrm{PFP}\}$ we define the bounded version BO *by restricting the syntax to*

$$[(\mathrm{D})\mathrm{TC}_{\bar{x},\bar{y}}\ \alpha(\bar{y};\bar{z}) \wedge \varphi(\bar{x},\bar{y},\bar{z})] \quad \textit{and} \quad [\mathrm{L/PFP}_{R,\bar{x}}\ \alpha(\bar{x};\bar{z}) \wedge \varphi(R,\bar{x},\bar{z})]$$

for some bounded guard α. Let BFO(O) *be the logic obtained by adding the operator O to* BFO*. Similarly, bounded second-order logic* BSO *is obtained by adding (unrestricted) second-order quantifiers to* BFO*.*

Definition 3. *Let $\varphi(\bar{x})$ be a formula of some bounded logic, and let y be a variable appearing bound in φ (w.l.o.g. assume that no variable is quantified twice). For values $\bar{c}$ of $\bar{x}$, the* domain of y at $\bar{c}$ *is defined inductively as follows. Let $(Qy.\alpha(y;\bar{x},\bar{z}))\psi$ be the subformula where y is bound.*

$$\mathrm{dom}(y) := \{\, a \mid \textit{there are } \bar{b} \textit{ in the domains of } \bar{z} \textit{ such that } \alpha(a;\bar{c},\bar{b}) \textit{ holds} \,\}.$$

Intuitively, the domain contains all values y may have. Note that, by induction the domains of bound variables are finite.

Remark 4. Regarding the expressive power the following inclusions hold:

$$\begin{array}{ccccccccc}
\mathrm{FO} & \subseteq & \mathrm{FO(DTC)} & \subseteq & \mathrm{FO(TC)} & \subseteq & \mathrm{FO(LFP)} & \subseteq & \mathrm{FO(PFP)} \\
\cup\!| & & \cup\!| & & \cup\!| & & \cup\!| & & \cup\!| \\
\mathrm{BFO} & \subseteq & \mathrm{BFO(BDTC)} & \subseteq & \mathrm{BFO(BTC)} & \subseteq & \mathrm{BFO(BLFP)} & \subseteq & \mathrm{BFO(BPFP)}
\end{array}$$

3 Arithmetic and high complexity classes

In this section we will consider $(\mathbb{N}, <, \mathscr{F})$, the natural numbers with order and some additional functions $f \in \mathscr{F}$ where $\mathscr{F}$ is allowed to be empty. Note that in this case we can w.l.o.g. assume that all guards are of the form

$$\bar{x} < t(\bar{y}) \;:=\; x_0 < t(\bar{y}) \wedge \cdots \wedge x_{n-1} < t(\bar{y})$$

for some $\mathscr{F}$-term t. The expressive power of bounded logics mainly depends on the growth-rate of the bounds. In order to compare such rates we define $\mathscr{F}_0 \leq \mathscr{F}_1$ for classes $\mathscr{F}_0$ and $\mathscr{F}_1$ of functions on $\mathbb{N}$ iff for all $f_0 \in \mathscr{F}_0$ there is some $f_1 \in \mathscr{F}_1$ such that $f_0(n, \ldots, n) \leq f_1(n, \ldots, n)$ for all $n \in \mathbb{N}$, and we write $\mathscr{F}_0 \equiv \mathscr{F}_1$ iff both $\mathscr{F}_0 \leq \mathscr{F}_1$ and $\mathscr{F}_1 \leq \mathscr{F}_0$.

Let $\mathcal{T}\mathscr{F}$ be the set of terms built from functions of $\mathscr{F}$. We will see that it mainly depends on the growth-rate of $\mathcal{T}\mathscr{F}$ which complexity classes can be captured on $(\mathbb{N}, <, \mathscr{F})$.

In order to define the complexity of a set of natural numbers it is assumed that numbers are coded by their binary encoding in reversed order, i.e., with the least significant bit first. Note that the number of bits of n is $\lceil \log_2(n+1) \rceil$. Thus, given a monotone function $f : \mathbb{N}^k \to \mathbb{N}$ and a tuple $\bar{n} \in \mathbb{N}^k$ where n_i has l_i bits, the number of bits of $f(\bar{n})$ is

$$(\mathcal{B}f)(\bar{l}) := \left\lceil \log_2\bigl(f(2^{l_0 - 1}, \ldots, 2^{l_{k-1}-1}) + 1\bigr) \right\rceil.$$

Our first result is preceeded by some two lemmas. Let $f : \mathbb{N}^k \to \mathbb{N}$ be a function such that $f(\bar{a}) \geq a_i$ for all $i < k$. We call a formula $\varphi(\bar{x})$ *f-bounded* iff for all $\bar{a} \in \mathbb{N}^k$ and every term $t(\bar{x}, \bar{y})$ in $\varphi(\bar{x})$ the inequality $t(\bar{a}, \bar{b}) \leq f(\bar{a})$ holds for values $\bar{b}$ in the domains of $\bar{y}$. Note that for all formulae $\varphi(\bar{x})$ of bounded logics defined above there is some $\mathscr{F}$-term $t(\bar{x})$ such that φ is t-bounded.

Lemma 5. *Let $\mathscr{F}$ be a set of functions whose graphs are decidable in linear space, and let $\varphi(\bar{x}) \in$ BFO be f-bounded. The question whether $(\mathbb{N}, <, \mathscr{F}) \models \varphi(\bar{a})$ can be decided in space $\mathcal{O}(|\varphi| \log_2 f(\bar{a}))$.*

Proof. Since φ is f-bounded we need to consider only values less than $f(\bar{a})$. These can be stored in space $\mathcal{O}(\log_2 f(\bar{a}))$. The claim is proved by induction on φ. To evaluate a function $h(\bar{b})$ we can enumerate all numbers c and check if the tuple $(\bar{b}, c)$ belongs to the graph. Thus, since both the arguments and the values of functions are less than $f(\bar{a})$, atoms can be evaluated in space $\mathcal{O}(\log_2 f(\bar{a}))$. The induction step for boolean connectives is trivial. So consider a formula of the form $(Qy < t(\bar{x}))\psi(y, \bar{x})$. To decide whether it holds we can iterate over all values for y. The only space needed to do so is the storage of y. Thus, it is sufficient to have space $\mathcal{O}(\log_2 f(\bar{a}))$ for each variable appearing in φ. $\square$

The second lemma shows that in many cases we can assume that addition and multiplication is available.

Lemma 6. *The graphs of addition and multiplication are* BFO(BDTC)*-definable in* $(\mathbb{N}, <)$.

Proof. Clearly, 0 and the successor relation S are definable. $+$ and $\cdot$ are defined via the usual recurrence.

$$\begin{aligned}
x + y = z &:= (y = 0 \land x = z) \\
&\quad \lor [\mathrm{DTC}_{uv,u'v'}\ u' < y \land v' < z \land Su'u \land Sv'v](yz, 0x) \\
x \cdot y = z &:= (y = 0 \land z = 0) \\
&\quad \lor [\mathrm{DTC}_{uv,u'v'}\ u' < y \land v' < z \land Su'u \land v' + x = v](yz, 00)
\end{aligned}$$
$\square$

The previous lemma indicates that it does not matter much which functions are present since many of them are definable if the logic is at least as expressible as BFO(BDTC). In deed, for such logics, our next result shows that the only thing which matters is the growth-rate of the available functions.

Theorem 7. *Let $\mathscr{R}$ and $\mathscr{F}$ be sets of functions such that $\mathscr{R} \equiv \mathcal{BT}\mathscr{F}$, the graphs of functions in $\mathscr{F}$ are computable in linear space, $\mathcal{O}(n) \subseteq \mathscr{R}$, and $\mathcal{O}\mathscr{R} \subseteq \mathscr{R}$. Let $X \subseteq \mathbb{N}$.*

(i) $X \in \mathrm{DSPACE}[\mathscr{R}]$ iff X *is* BFO(BDTC)*-definable in* $(\mathbb{N}, \mathscr{F}, <)$.

(ii) $X \in \mathrm{NSPACE}[\mathscr{R}]$ iff X *is* BFO(BTC)*-definable in* $(\mathbb{N}, \mathscr{F}, <)$.

(iii) $X \in \mathrm{DTIME}[2^{\mathscr{R}}]$ iff X *is* BFO(BLFP)*-definable in* $(\mathbb{N}, \mathscr{F}, <)$
iff X *is* $\mathrm{B}\Sigma^1_1$-HORN*-definable in* $(\mathbb{N}, \mathscr{F}, <)$
iff X *is* BSO-HORN*-definable in* $(\mathbb{N}, \mathscr{F}, <)$.

(iv) $X \in \mathrm{NTIME}[2^{\mathscr{R}}]$ iff X *is* $\mathrm{B}\Sigma^1_1$*-definable in* $(\mathbb{N}, \mathscr{F}, <)$.

(v) $X \in \mathrm{DSPACE}[2^{\mathscr{R}}]$ iff X *is* BFO(BPFP)*-definable in* $(\mathbb{N}, \mathscr{F}, <)$.

Proof. In the formulae defined below we will use addition and multiplication whose graphs are definable in all logics mentioned above. In order to keep them readable we will use not only their graphs but also the functions themselves. This can be done since we only use equations of the form $x = t$ for some variable x and term t. Thus all intermediate results are less than or equal to x and we can reduce t by introducing new variables y by bounded quantification $(\exists y \leq x)$.

Below the following model of Turing machine is used. A k-tape Turing machine M is given by a tuple $(Q, \Sigma, \Delta, q_0, F)$ where Q is the set of states, $\Sigma = \{0, 1\}$ is both the input and the working alphabet, q_0 is the initial state, F is the set of final states, and

$$\Delta \subseteq Q \times \Sigma \times \Sigma^k \times \Sigma^k \times Q \times \{-1, 0, 1\}^{k+1}$$

is the transition relation with components: old state, symbol on the input tape, symbols on the working tapes, symbols to write on the working tapes, new state, and movement of the heads.

We prove only two items. The other proofs are similar.

(i) ($\Rightarrow$) Let $M = (Q, \Sigma, \Delta, q_0, F)$ be an f space-bounded k-tape Turing machine recognising X. W.l.o.g. assume that $Q = \{0, \dots, n\}$, $\Sigma = \{0, 1\}$, and $q_0 = 0$. Choose some $\mathscr{F}$-term $r(x)$ such that $f(x) \leq \mathcal{B}r(x)$ for all $x \in \mathbb{N}$. Configurations of M can be stored in tuples $(q, \bar{w}, \bar{p})$ where each component is less than $2r(x)$. If there is a formula $\mathrm{TRANS}(\bar{c}, \bar{c}')$ expressing that the configuration stored in $\bar{c}'$ is the successor of $\bar{c}$, we can determine whether a final configuration can be reached from the initial one using an DTC-operator.

$$\begin{aligned}\varphi_X(x) := (\exists w_1 \cdots w_k p_0 \cdots p_k < 2r(x)) \\ \bigvee_{q_f \in F} [\mathrm{DTC}_{q\bar{w}\bar{p}, q'\bar{w}'\bar{p}'}\ q`\bar{w}'\bar{p}' < 2r(x) \wedge \mathrm{TRANS}(q\bar{w}\bar{p}, q'\bar{w}'\bar{p}')] \\ (0\underbrace{0 \cdots 0}_{k}\underbrace{1 \cdots 1}_{k+1}, q_f w_1 \cdots w_k p_0 \cdots p_k).\end{aligned}$$

TRANS is defined by

$$
\begin{aligned}
&\mathrm{TRANS}(q\bar{w}\bar{p}, q'\bar{w}'\bar{p}') := \\
&\qquad \bigvee_{(i,a_0\bar{a},\bar{b},j,\bar{m})\in\varDelta} \Big(q = i \wedge q' = j \wedge \mathrm{bit}_{a_0}(x,p_0) \wedge \bigwedge_{l=0}^{k} \mathrm{MOVE}_{m_l}(p_l,p'_l) \\
&\qquad\qquad \wedge \bigwedge_{l=1}^{k} (\exists s < w_l)(\exists s' < p_l)\,(w_l = (2s+a_l)p_l + s' \wedge \\
&\qquad\qquad\qquad\qquad w'_l = (2s+b_l)p_l + s')\Big),
\end{aligned}
$$

where

$$
\begin{aligned}
\mathrm{bit}_d(x,p) &:= (\exists s < x)(\exists s' < p)(x = (2s+d)p + s'), \\
\mathrm{MOVE}_m(p,p') &:= \begin{cases} p = 2p' & \text{if } m = -1, \\ p' = p & \text{if } m = 0, \\ p' = 2p & \text{if } m = 1. \end{cases}
\end{aligned}
$$

($\Leftarrow$) Let X be defined by $\varphi(x)$, and let $\varphi(x)$ be $t(x)$-bounded. Since $\mathscr{R} \equiv \mathcal{BT}\mathscr{F}$ there is some $r \in \mathscr{R}$ with $\log_2 t(2^n) \leq r(n)$ for all $n \in \mathbb{N}$. Thus it is sufficient to prove that $X \in \mathrm{DSPACE}[\mathcal{O}(\log_2 t(2^n))]$. For BFO-formulae this was proved in the above lemma. It remains to consider the evaluation of a DTC-operator $[\mathrm{DTC}_{\bar{x},\bar{y}}\ \bar{z} < s(\bar{z}) \wedge \psi(\bar{x},\bar{y},\bar{z})]$ which can be done by calculating the sequence $\bar{x}_0, \bar{x}_1, \bar{x}_2, \dots$ of tuples such that $\psi(\bar{x}_i, \bar{x}_{i+1}, \bar{z})$ holds for all i. By induction we can assume that this condition can be checked in $\mathrm{DSPACE}[\mathcal{O}(\log_2 t(2^n))]$. In order to compute $\bar{x}_{i+1}$ we only need to remember $\bar{x}_i$. Thus the space to store two such tuples is sufficient.

(iii) ($\Rightarrow$) Let $M = (Q, \varSigma, \varDelta, q_0, F)$ be an f time-bounded k-tape Turing machine recognising X. W.l.o.g. assume that $Q = \{0,\dots,n\}$, $\varSigma = \{0,1\}$, and $q_0 = 0$. Choose some $\mathscr{F}$-term $t(x)$ such that $f(x) \leq \mathcal{B}t(x)$ for all $x \in \mathbb{N}$, and let $r(x) = 2t(x) + n$. Using least-fixed points we inductively define relations Q, $\bar{W}$, $\bar{P}$ containing the whole run of M on input x. For instance, $(q,t) \in Q$ means that M is in state q at time t. W.l.o.g. we define those relations by a simultaneous fixed-point which can always be transformed into a normal one.

$$
\begin{aligned}
\varphi_X(x) := (\exists t < r(x)) \bigvee_{q_f \in F} [\mathrm{LFP}_{Q,qt;\bar{W},apt;\bar{P},pt}\ & qt < r(x) \wedge \psi_Q \\
& apt < r(x) \wedge \psi_{W_1} \\
& \dots \\
& apt < r(x) \wedge \psi_{W_k} \\
& pt < r(x) \wedge \psi_{P_0} \\
& \dots \\
& pt < r(x) \wedge \psi_{P_k}]_0(q_f t)
\end{aligned}
$$

where

$$\begin{aligned}
&\mathrm{CONF}_{q,a_0\bar{a}}(t) := \\
&\quad Qqt \land (\exists p < r(x))(\exists s < x)(\exists s' < p)(P_0pt \land x = (2s + a_0)p + s') \\
&\land \bigwedge_{l=1}^{k} (\exists p < r(x))(P_lpt \land W_la_lpt) \\
&\psi_Q(q,t) := (q = 0 \land t = 0) \lor \bigvee_{(i,a_0\bar{a},\bar{b},j,\bar{m})\in\Delta} (\mathrm{CONF}_{i,a_0\bar{a}}(t-1) \land q = j) \\
&\psi_{W_l}(a,p,t) := (a = 0 \land t = 0) \\
&\qquad \lor \bigvee_{(i,a_0\bar{a},\bar{b},j,\bar{m})\in\Delta} [\mathrm{CONF}_{i,a_0\bar{a}}(t-1) \land \\
&\qquad\qquad (\exists p' < r(x))(P_lp'(t-1) \land [(p \neq p' \land W_lap(t-1)) \\
&\qquad\qquad\qquad \lor (p = p' \land a = b_l)])] \\
&\psi_{P_l}(p,t) := (p = 1 \land t = 0) \\
&\qquad \lor \bigvee_{(i,a_0\bar{a},\bar{b},j,\bar{m})\in\Delta} [\mathrm{CONF}_{i,a_0\bar{a}}(t-1) \land \\
&\qquad\qquad (\exists p' < r(x))(P_lp'(t-1) \land \mathrm{MOVE}_{m_l}(p,p'))]
\end{aligned}$$

($\Leftarrow$) Let X be defined by $\varphi(x)$, and let $\varphi(x)$ be $t(x)$-bounded. Since $\mathscr{R} \equiv \mathcal{BT}\mathscr{F}$ there is some $r \in \mathscr{R}$ with $\log_2 t(2^n) \leq r(n)$ for all $n \in \mathbb{N}$. Thus, it is sufficient to prove that $X \in \mathrm{DTIME}[2^{\mathcal{O}(\log_2 t(2^n))}] = \mathrm{DTIME}[\mathcal{O}(t(2^n)^{\mathcal{O}(1)})]$. To evaluate a fixed-point operator $[\mathrm{LFP}_{R,\bar{x}}\ \bar{x} < t(\bar{y}) \land \psi(\bar{x},\bar{y})]$ we calculate its stages R^0, R^1, R^2, ... where by boundedness we only need to consider the part $\tilde{R}^i := R^i \cap \{0,\dots,t(a)-1\}^n$. Thus, $\tilde{R}^{i+1}$ can be computed in $t(a)^n$ steps from $\tilde{R}^i$ each of which takes time $\mathcal{O}(t(a)^{\mathcal{O}(1)}$ (by induction). Since the fixed-point is reached after at most $t(a)^n$ stages we obtain a bound of $\mathcal{O}(t(a)^{\mathcal{O}(1)} \cdot t(a)^n \cdot t(a)^n)$. □

To apply this theorem we need to define functions of appropriate growth. Let $x \# y := 2^{\lceil \log_2 x\rceil \lceil \log_2 y\rceil}$. (Note that # is associative and commutative.) Since

$$\begin{aligned}
\mathcal{BT}\{+,\cdot\} &\equiv \mathcal{O}(n), \\
\mathcal{BT}\{+,\cdot,\#\} &\equiv \mathcal{O}(n^{\mathcal{O}(1)}), \\
\mathcal{BT}\{+,\cdot,2^n\} &\equiv \mathcal{T}\{2^n\}
\end{aligned}$$

we obtain the results in Table 2. What happens when no functions are present?

Theorem 8. *Let $X \subseteq \mathbb{N}$. The results of the previous theorem also hold for $\mathscr{F} = \emptyset$ and $\mathscr{R} = \mathcal{O}(n)$.*

Proof. The only place where the proofs above fail is the existence of a term $r(x)$ providing a bound large enough to store either the complete contents of a tape or the position of a cell on the tape. For $\mathscr{R} = \mathcal{O}(n)$ this term would be $r(x) := x^c$

Table 2. Logics capturing complexity classes on arithmetic

Class	Logic	Structure
Dspace[$\mathcal{O}(n)$]	BFO(BDTC)	$(\mathbb{N}, <, +, \cdot)$
Nspace[$\mathcal{O}(n)$]	BFO(BTC)	$(\mathbb{N}, <, +, \cdot)$
Pspace	BFO(BTC)	$(\mathbb{N}, <, +, \cdot, \#)$
Dtime[$2^{\mathcal{O}(n)}$]	BFO(BLFP)	$(\mathbb{N}, <, +, \cdot)$
Ntime[$2^{\mathcal{O}(n)}$]	BΣ^1_1	$(\mathbb{N}, <, +, \cdot)$
Exptime	BFO(BLFP)	$(\mathbb{N}, <, +, \cdot, \#)$
Nexptime	BΣ^1_1	$(\mathbb{N}, <, +, \cdot, \#)$
Dspace[$2^{\mathcal{O}(n)}$]	BFO(BPFP)	$(\mathbb{N}, <, +, \cdot)$
Expspace	BFO(BPFP)	$(\mathbb{N}, <, +, \cdot, \#)$
Elementary	BFO(BDTC)	$(\mathbb{N}, <, +, \cdot, 2^n)$

for some c. Though such an r is not available we can handle values of this size by storing each in c variables. Using the (BFO-definable) lexicographic order on c-tuples we can then define addition and multiplication as above. □

The only property of $(\mathbb{N}, <, \mathscr{F})$ used in the proofs above was the order type and the growth-rate of $\mathscr{F}$-terms. This enables us to generalise the results to arbitrary structures as follows. Let $\mathfrak{A} = (A, <, R_0, \dots, R_r, f_0, \dots, f_s)$ be a linearly ordered structure of order type ω. For $a \in A$ let $|a| := \{\, b \in A \mid b < a \,\}$. If we identify elements $a \in A$ by the natural number $|a|$ we get the isomorphic structure $(\mathbb{N}, <, R'_0, \dots, R'_r, f'_0, \dots, f'_s)$ to which we can apply our capturing results. If the complexity of subsets $X \subseteq A$ is measured with regard to the encoding $a \mapsto |a|$ we obtain

Theorem 9. *Let $\mathfrak{A} = (A, <, R_0, \dots, R_r, f_0, \dots, f_s)$ be a linearly ordered structure of order type ω such that R_i, $i \leq r$, and the graphs of f_i, $i \leq s$, are computable in linear space. Let $\mathscr{F} := \{f_0, \dots, f_s\}$ and let $\mathscr{R}$ be a set of functions such that if $\mathscr{F}$ is empty then $\mathscr{R} = \mathcal{O}(n)$, otherwise $\mathscr{R} \equiv \mathcal{BT}\mathscr{F}$, $\mathcal{O}(n) \subseteq \mathscr{R}$, and $\mathcal{O}\mathscr{R} \subseteq \mathscr{R}$. Let $X \subseteq A$.*

(i) $X \in$ Dspace[$\mathscr{R}$] iff X *is* BFO(BDTC)*-definable in* $\mathfrak{A}$.
(ii) $X \in$ Nspace[$\mathscr{R}$] iff X *is* BFO(BTC)*-definable in* $\mathfrak{A}$.
(iii) $X \in$ Dtime[$2^{\mathscr{R}}$] iff X *is* BFO(BLFP)*-definable in* $\mathfrak{A}$
iff X *is* BΣ^1_1-Horn*-definable in* $\mathfrak{A}$
iff X *is* BSO-Horn*-definable in* $\mathfrak{A}$.
(iv) $X \in$ Ntime[$2^{\mathscr{R}}$] iff X *is* BΣ^1_1*-definable in* $\mathfrak{A}$.
(v) $X \in$ Dspace[$2^{\mathscr{R}}$] iff X *is* BFO(BPFP)*-definable in* $\mathfrak{A}$.

So far, we only considered extensions of first-order logic. Next we look at the expressive power of BFO. An old result provides an answer in the case of the structure $(\mathbb{N}, <, +, \cdot)$.

Theorem 10 (Wrathall [7]). *X belongs to the linear hierarchy iff X is* BFO-*definable in* $(\mathbb{N}, <, +, \cdot)$.

As mentioned in [6], by adding the operator # characterisation of PH is obtained.

Theorem 11. *$X \in$ PH iff X is* BFO-*definable in* $(\mathbb{N}, <, +, \cdot, \#)$.

Proof. ($\Leftarrow$) Let X be defined by

$$\varphi(x) = (Q_0 y_0 < t_0) \cdots (Q_{n-1} y_{n-1} < t_{n-1}) \psi(x, \bar{y})$$

where ψ is quantifier-free. There is some $k \in \mathbb{N}$ such that $\varphi(x)$ is $(2^{\log_2^k x})$-bounded. Hence each y_i $(i < n)$ can be encoded in $(\log_2 x)^k$ bits. Obviously, quantifier-free formulae $\psi(\bar{a})$ can be evaluated in polynomial time with respect to the length of $\bar{a}$. Thus,

$$X := \{\, x \mid Q_0^{\mathrm{p}} y_0 \cdots Q_{n-1}^{\mathrm{p}} y_{n-1} R(x, \bar{y}) \,\}$$

where all quantifiers are polynomial bounded and

$$R(x, \bar{y}) := y_0 < t_0 \wedge \cdots \wedge y_{n-1} < t_{n-1} \wedge \psi(x, \bar{y})$$

is a Ptime-predicate. Hence, $X \in$ PH.

($\Rightarrow$) By a corollary to Fagin's characterisation of Nptime, there is some $\varphi \in$ SO such that $x \in X$ iff $\underline{x} \models \varphi$ for all $x \in \{0,1\}^+$ where $\underline{x}$ is the word model of x. We construct a formula $\tilde{\varphi}(x) \in$ BFO with

$$\underline{x} \models \varphi \text{ iff } (\mathbb{N}, <, 0, 1, +, \cdot, \#) \models \tilde{\varphi}(\mathrm{val}(x1))$$

where $\mathrm{val}(y)$ is the number whose binary encoding in reversed order is y. Define

$$\tilde{\varphi}(x) := (\exists p < x + 1)(P_2 p \wedge x < 2p \wedge \varphi^*(x, p))$$

where p denotes the position of the final digit,

$$P_2 x := x = 1 \vee (\forall y < x + 1)(y \mid x \wedge y \neq 1 \to 2 \mid y)]$$

defines the powers of 2, and φ^* is constructed such that

$$\begin{aligned} &\underline{x} \models \psi(U_0, \ldots, U_{n-1}, y_0, \ldots, y_{m-1}) \\ \text{iff } &(\mathbb{N}, <, 0, 1, +, \cdot, \#) \models \psi^*(x, p, u_0, \ldots, u_{n-1}, 2^{y_0}, \ldots, 2^{y_{m-1}}) \end{aligned}$$

where $u_i := \sum \big\{\, 2^{l_0 + l_1 |x| + \cdots + l_{k-1} |x|^{k-1}} \;\big|\; (l_0, \ldots, l_{k-1}) \in U_i \,\big\}$. Define

$$\begin{aligned}
(y_0 = y_1)^* &:= y_0 = y_1 \\
(y_0 < y_1)^* &:= y_0 < y_1 \\
(Py)^* &:= \mathrm{bit}(x, y) \\
(U y_0 \ldots y_{k-1})^* &:= \mathrm{bit}(u, y_0 (y_1 \# p) \cdots (y_{k-1} \# p \# \cdots \# p)) \\
(\neg \psi)^* &:= \neg \psi^* \\
(\psi \vee \vartheta)^* &:= \psi^* \vee \vartheta^* \\
(\exists y \psi)^* &:= (\exists y < p)(P_2 y \wedge \psi^*) \\
(\exists U \psi)^* &:= (\exists u < p \# \cdots \# p) \psi^*
\end{aligned}$$

where $\text{bit}(x, y)$ expresses that the bit of x at position y is 1

$$\text{bit}(x, y) := (\exists s < x)(\exists s' < y)(x = (2s + 1)y + s'). \qquad \square$$

Theorem 12. *$X \subseteq \mathbb{N}$ is of elementary complexity iff X is* BFO*-definable in* $(\mathbb{N}, <, +, \cdot, 2^n)$.

The proof is done by directly coding computations of Turing machines. In this case fixed-points are not needed since numbers large enough to code whole runs are available.

Remark 13. The results of Theorems 7 (i), (ii), and 10–12 also hold for oracle machines if one adds the orcale set as unary predicate to the structure.

Unbounded fixed-points. Above we met the boundedness requirement for the logics considered by an ad hoc definition of bounded fixed-points. Next we will investigate under which conditions this can be avoided by using normal (unbounded) operators instead. The first result shows that in many situations it can not.

Proposition 14. *Any relation which is* FO(DTC)*-definable in* $(\mathbb{N}, <, +, \cdot)$ *(in particular any arithmetic relation) is already* BFO(DTC)*-definable in* $(\mathbb{N}, <)$.

Proof. Since addition and multiplication are FO(DTC)-definable it is sufficient to show how to emulate unbounded quantifiers by DTC-operators. To simulate $\exists x\varphi$ we can enumerate all numbers until some n with $\varphi(n)$ is found. Formally,

$$\exists x\varphi \equiv [\text{DTC}_{x,x'}\ (\neg\varphi(x) \land x' = x + 1) \lor (\varphi(x) \land x' = 0)](0, 0). \qquad \square$$

In contrast, for purely relational structures a positive result is obtained. Note that the proof above shows that it does not hold for transitive-closure operators.

Proposition 15. *Let $\mathfrak{A} = (A, <, R_0, \dots, R_m)$ be a relational structure of order type ω, and let $X \subseteq A^k$.*

(i) *X is* BFO(LFP)*-definable if and only if it is* BFO(BLFP)*-definable.*
(ii) *X is* BFO(PFP)*-definable if and only if it is* BFO(BPFP)*-definable.*

Proof. ($\Leftarrow$) is trivial. For ($\Rightarrow$) consider the stages R^0, R^1, $\dots$ of the fixed-point induction of $\psi(\bar{y}, \bar{z}) := [\text{LFP}_{R,\bar{x}}\ \varphi(\bar{x}, \bar{y})](\bar{z})$. Since all bounds are of the form $u < v$ for variables u and v the decision whether $\bar{x} \in R^{i+1}$ depends only on values of R^i for arguments less than

$$t := \max\{x_0, \dots, x_n, y_0, \dots y_m, z_0, \dots, z_l\}.$$

In particular, the value at position $\bar{z}$ only depends on lower positions. Therefore we can replace the operator by an bounded one.

$$\psi(\bar{y}, \bar{z}) \equiv \bigvee_i \big(\max(y_i, \bar{y}, \bar{z}) \land \chi(y_i)\big) \lor \bigvee_i \big(\max(z_i, \bar{y}, \bar{z}) \land \chi(z_i)\big)$$

Table 3. Logics capturing complexity classes on arithmetic

Class	Logic	Structure
LinH	BFO	$(\mathbb{N}, <, +, \cdot)$
PH	BFO	$(\mathbb{N}, <, +, \cdot, \#)$
Dspace$[\mathcal{O}(n)]$	BFO(BDTC)	$(\mathbb{N}, <)$
Nspace$[\mathcal{O}(n)]$	BFO(BTC)	$(\mathbb{N}, <)$
Pspace	BFO(BTC)	$(\mathbb{N}, <, \#)$
Dtime$[2^{\mathcal{O}(n)}]$	BFO(BLFP)	$(\mathbb{N}, <)$
Ntime$[2^{\mathcal{O}(n)}]$	$\mathrm{B}\Sigma^1_1$	$(\mathbb{N}, <)$
Exptime	BFO(BLFP)	$(\mathbb{N}, <, \#)$
Nexptime	$\mathrm{B}\Sigma^1_1$	$(\mathbb{N}, <, \#)$
Dspace$[2^{\mathcal{O}(n)}]$	BFO(BPFP)	$(\mathbb{N}, <)$
Expspace	BFO(BPFP)	$(\mathbb{N}, <, \#)$
Elementary	BFO	$(\mathbb{N}, <, +, \cdot, 2^n)$

where $\max(u, \bar{y}, \bar{z}) := \bigwedge_k y_k \leq u \wedge \bigwedge_k z_k \leq u$ says that u is a maximal element, and

$$\chi(u) := [\mathrm{LFP}_{R,\bar{x}}\ x_0 \leq u \wedge \cdots \wedge x_n \leq u \wedge \varphi(\bar{x}, \bar{y})](\bar{z})$$

is the bounded version of the LFP-operator. The proof for BFO(PFP) is identical. □

The characterisations of standard complexity classes we have obtained is summarised in the table above. The results remain valid if we add any relations or functions computable in the respective class. In particular we may add 0, 1, $+$, and $\cdot$. Also the structure $(\mathbb{N}, <)$ may be replaced by any linear order $(A, <)$ of the same order type. Similarly, $(\mathbb{N}, <, \#)$ may be replaced by $(A, <, f)$ where $2^{\log_2^c |a|} \leq |f(a)| \leq 2^{\log_2^d |a|}$ for some c, $d > 1$.

4 The free monoid and low complexity classes

So far, we have obtained only characterisations of high (above Ptime) complexity classes. Intuitively, this was caused by the fact that, in arithmetic with the usual order, numbers of n bits have about 2^n predecessors. If we are interested in low complexity classes we thus have to choose a different order. In the classical approach variables can range over n positions in a word model of length n. Therefore, we next consider the free monoid with prefix-ordering where words of length n have n predecessors.

Definition 16. *Let* $\mathfrak{T} := (\{0,1\}^*, \sigma_0, \sigma_1, \prec)$ *where*

$$\sigma_i xy \ :\text{iff}\ \ y = xi, \qquad \textit{and} \qquad x \prec y \ :\text{iff}\ \ y = xz \textit{ for some } z \neq \varepsilon.$$

It turns out that this choice enables us to translate many of the classical results to our setting and vice versa. Let $\mathscr{L}$ consists of the following logics: FO, FO(DTC), FO(TC), FO(LFP), FO(PFP), Σ^1_1-Horn, Σ^1_1, SO-Horn, and SO. For $\mathfrak{L} \in \mathscr{L}$ denote by B$\mathfrak{L}$ the corresponding bounded version.

Theorem 17. *Let $X \subseteq \{0,1\}^+$ and $\mathfrak{L} \in \mathscr{L}$. The following statements are equivalent:*

(i) *There is some $\varphi \in \mathfrak{L}$ such that $w \in X$ iff $\underline{w} \models \varphi$.*
(ii) *There is some $\varphi(x) \in \mathrm{B}\mathfrak{L}$ such that $w \in X$ iff $\mathfrak{T} \models \varphi(w)$.*

The familiar results of descriptive complexity theory can thus be stated as

Corollary 18. *Let $X \subseteq \{0,1\}^*$.*

(i) $X \in$ Logspace iff X *is* BFO(BDTC)*-definable in* $\mathfrak{T}$
(ii) $X \in$ Nlogspace iff X *is* BFO(BTC)*-definable in* $\mathfrak{T}$
(iii) $X \in$ Ptime iff X *is* BFO(BLFP)*-definable in* $\mathfrak{T}$
iff X *is* BΣ^1_1-Horn*-definable in* $\mathfrak{T}$
iff X *is* BSO-Horn*-definable in* $\mathfrak{T}$.
(iv) $X \in$ Nptime iff X *is* BΣ^1_1*-definable in* $\mathfrak{T}$
(v) $X \in$ PH iff X *is* BSO*-definable in* $\mathfrak{T}$
(vi) $X \in$ Pspace iff X *is* BFO(BPFP)*-definable in* $\mathfrak{T}$

The proof of Theorem 17 is divided into two propositions.

Lemma 19. *For every $\varphi \in \mathfrak{L}$ there is some $\varphi^*(x) \in \mathrm{B}\mathfrak{L}$ such that, for all $w \in \{0,1\}^+$, $\underline{w} \models \varphi$ iff $\mathfrak{T} \models \varphi^*(w)$.*

Proof. We construct $\varphi^*(x)$ such that for all subformulae ψ the following condition is satisfied:

$$\underline{x} \models \psi(X_0, \dots, X_n, y_0, \dots, y_m) \text{ iff } \mathfrak{T} \models \psi^*(x, X^*_0, \dots, X^*_n, y^*_0, \dots, y^*_m)$$

where y^*_i is the prefix of x of length y_i, and X^*_i contains a tuple of prefixes of x iff the tuple of their lengths is in X_i. In the following definition variables named y, y_0, etc. are bounded, whereas x is the only free variable.

$$\begin{array}{ll}
(y_0 = y_1)^* := y_0 = y_1 & (\neg\psi)^* := \neg\psi^* \\
(Py)^* := (\exists y' \preceq x)\sigma_1 y y' & (\psi \vee \vartheta)^* := \psi^* \vee \vartheta^* \\
(y_0 < y_1)^* := y_0 \prec y_1 & (\exists y \psi)^* := (\exists y \prec x)\psi^* \\
(X\bar{y})^* := X\bar{y} & (\exists X \psi)^* := (\exists X)\psi^*
\end{array}$$

$$\big([(\mathrm{D})\mathrm{TC}_{\bar{u},\bar{v}}\ \psi](\bar{t},\bar{t}')\big)^* := [(\mathrm{D})\mathrm{TC}_{\bar{u},\bar{v}}\ v_0 \prec x \wedge \dots \wedge v_{k-1} \prec x \wedge \psi^*](\bar{t},\bar{t}')$$
$$\big([X\mathrm{FP}_{R,\bar{u}}\ \psi](\bar{t})\big)^* := [X\mathrm{FP}_{R,\bar{u}}\ \bar{u} \prec x \wedge \dots \wedge u_{k-1} \prec x \wedge \psi^*](\bar{t})$$

□

Lemma 20. *For every $\varphi(x) \in \mathrm{B}\mathfrak{L}$ there is some $\varphi' \in \mathfrak{L}$ such that, for all $w \in \{0,1\}^+$, $\mathfrak{T} \models \varphi(w)$ iff $\underline{w} \models \varphi'$.*

Proof. Note that, since $\varphi(x)$ has only one free variable x, all bounded variables are prefixes of x. Thus, it again is sufficient to ensure that

$$\mathfrak{T} \models \psi(x, X_0, \dots, X_n, y_0, \dots, y_m) \text{ iff } \underline{x} \models \psi'(X'_0, \dots, X'_n, |y_0|, \dots, |y_m|)$$

where the definition of X'_i is slightly more involved since there are $|x|+1$ prefixes of x, but only $|x|$ elements in $\underline{x}$. Therefore, we double the arity of X_i and define

$$X'_i := \{\, (u'_{00}u'_{01}, \dots, u'_{k0}u'_{k1}) \mid (u_0, \dots, u_k) \in X_i \,\}$$

where

$$(u'_{j0}, u'_{j1}) := \begin{cases} (0, u_j) & \text{if } u_j \prec x \\ (|x|-1, |x|-1) & \text{if } u_j = x. \end{cases}$$

In the following definition variables named y, y_0, etc. are bounded, whereas x is the only free variable. t stands for an arbitrary term which can either be on of y, x, or one of y, min, max. Furthermore, $P^i t$ is Pt for $i = 1$, and $\neg Pt$ for $i = 0$.

$$\begin{array}{llll}
(y_0 = y_1)' & := y_0 = y_1 & (\sigma_i y_0 y_1)' & := Sy_0y_1 \wedge P^i y_0 \\
(y = x)' & := \text{false} & (\sigma_i x t)' & := \text{false} \\
(x = x)' & := \text{true} & (\sigma_i y x)' & := y = \max \wedge P^i y \\
(y_0 \prec y_1)' & := y_0 < y_1 & (y \prec x)' & := \text{true} \\
(x \prec t)' & := \text{false} & & \\
(\neg\psi)' & := \neg\psi' & ((\exists y_0 \prec y_1)\psi)' & := \exists y_0 (y_0 < y_1 \wedge \psi') \\
(\psi \vee \vartheta)' & := \psi' \vee \vartheta' & ((\exists y \prec x)\psi)' & := \exists y \psi' \\
(\exists X^k \psi)' & := \exists X^{2k} \psi' & (Xt_0 \dots t_{k-1})' & := X\tilde{t}_0 \dots \tilde{t}_{k-1}
\end{array}$$

where

$$\tilde{t} := \begin{cases} \min y & \text{if } t = y, \\ \max\max & \text{if } t = x. \end{cases}$$

$$\big([(\mathrm{D})\mathrm{TC}_{\bar{u},\bar{v}}\; v_0 \prec t_0 \wedge \dots \wedge v_{n-1} \prec t_{n-1} \wedge \psi](\bar{t}', \bar{t}'')\big)' :=$$
$$[(\mathrm{D})\mathrm{TC}_{\bar{u},\bar{v}}\; v_0 < t_0 \wedge \dots \wedge v_{n-1} < t_{n-1} \wedge \psi'](\bar{t}', \bar{t}'')$$
$$\big([X\mathrm{FP}_{R,\bar{u}}\; u_0 \prec t_0 \wedge \dots \wedge u_{n-1} \prec t_{n-1} \wedge \psi](\bar{t}')\big)' :=$$
$$[X\mathrm{FP}_{R,\bar{u}}\; u_0 < t_0 \wedge \dots \wedge u_{n-1} < t_{n-1} \wedge \psi'](\bar{t}')$$

□

Remark 21. (i) The preceding results can be generalised to formulae with several free variables.

(ii) Nothing changes if we replace σ_0, σ_1 by the corresponding functions or even add concatenation. For the last part note that for all variables y appearing

Table 4. Logics capturing complexity classes on the free monoid

Class	Logic	Structure
AC^0	BFO	$(\mathfrak{T}, \mathrm{bit})$
Logspace	BFO(BDTC)	$\mathfrak{T}$
Nlogspace	BFO(BTC)	$\mathfrak{T}$
Ptime	BFO(LFP)	$\mathfrak{T}$
Nptime	$\mathrm{B}\Sigma^1_1$	$\mathfrak{T}$
PH	BSO	$\mathfrak{T}$
Pspace	BFO(PFP)	$\mathfrak{T}$

in a formula $\varphi(x)$ there is some k such that y ranges over values of the form $y_0 \cdots y_j$, $j < k$, where the y_i are prefixes of x. Hence the value of each y can be stored in a fixed number of variables and we can eliminate concatenation by its BFO(BDTC)-definition.

(iii) Since $\mathfrak{T}$ is relational bounded LFP- and PFP-operators can be replaced by unbounded ones as in the case of arithmetic.

(iv) If one adds to $\mathfrak{T}$ either the relations $|x| + |y| = |z|$ and $|x| \cdot |y| = |z|$, or the relation $\mathrm{bit}(x, y)$ saying that the $|y|^{\mathrm{th}}$ bit of $|x|$ is 1, and considers word models with analogous predicates, we also can characterise the class AC^0, i.e., $X \subseteq \{0,1\}^*$ is in AC^0 iff X is BFO-definable in $(\mathfrak{T}, \mathrm{bit})$.

References

1. S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*, Addison-Wesley, 1995.
2. V. Bruyère, G. Hansel, C. Michaux, and R. Villemaire, *Logic and p-recognizable sets of integers*, Bull. Belg. Math. Soc., 1 (1994), pp. 191–238.
3. P. Clote, *Computation models and function algebras*, in Handbook of Computability Theory, E. R. Griffor, ed., North-Holland, 1999.
4. H.-D. Ebbinghaus and J. Flum, *Finite Model Theory*, Springer, 1995.
5. N. Immerman, *Descriptive Complexity*, Springer, New York, 1998.
6. J. Krajíček, *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, Cambridge University Press, 1995.
7. C. Wrathall, *Rudimentary predicates and relative computation*, SIAM Journal on Computing, 7 (1978), pp. 194–209.