On the expressive power of existential quantification in polynomial-time computability
(Extended abstract)

Dieter Spreen
Fachbereich Mathematik, Theoretische Informatik
Universität Siegen, 57068 Siegen, Germany
Email: `spreen@informatik.uni-siegen.de`

It is the aim of this paper to study the expressive power of bounded existential quantification in polynomial-time computability. Our goal was to characterize nondeterministic polynomial-time computations in a machine-independent way. The following considerations are intended to make our idea clear.

Let Γ be the finite alphabet $\{0, 1\}$ and IN be the following set of *initial functions* from Γ^* to Γ^*:

- (Constant) ε (the empty word considered as a zero-ary function).

- (Zero function) $Z(x) = \varepsilon$.

- (Successors) $S_0(x) = x0$ and $S_1(x) = x1$.

- (Projections) $U_i^n(x_1, \ldots, x_n) = x_i$, for $1 \leq i \leq n$.

- (Smash function) $x \# y = 1^{|x| \cdot |y|}$, where $|x|$ denotes the *length* of x.

Then it is well known that the smallest set $[\text{IN}; \text{SUB}, \text{LR}]$ of functions from Γ^* to Γ^* containing IN and closed under substitution and limited recursion is exactly the collection $\mathcal{PF}$ of all functions computable in polynomial time on a deterministic Turing transducer [6, 17]. Note that in recent years several other machine-independent characterizations of $\mathcal{PF}$ have been presented [2, 7, 9, 8]. In what follows we will work with the above characterization. But it should be obvious that characterizations in which a different set of initial functions or other closure operations are chosen can be used as well.

Now, for a set $A \subseteq \Gamma^*$, let $\sigma_A \colon \Gamma^* \rightharpoonup \Gamma$, defined by

$$\sigma_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ \text{undefined} & \text{otherwise,} \end{cases}$$

be the *semi-characteristic function* of A. Then

$$A \in \mathcal{NP} \Leftrightarrow \sigma_A \in [\text{IN}; \text{SUB}, \text{LR}, \text{E}^{1/2}],$$

where the *bounded half-quantifier* $\text{E}^{1/2}$ is the following operator. For functions $f \colon (\Gamma^*)^2 \rightharpoonup \Gamma^*$ and $b \colon \Gamma^* \rightharpoonup \Gamma^*$, $\text{E}^{1/2}(f, b)$ is the map with

$$\text{E}^{1/2}(f, b)(x) = \begin{cases} 1 & \text{if } x \in \text{dom}(b) \text{ and there is some } y \in \Gamma^* \\ & \text{such that } |y| \leq |b(x)| \text{ and } f(x, y) = 1, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

is used as an additional closure operation.

This shows that the bounded half-quantifier adds nondeterministic computations to the class $[\text{IN}; \text{SUB}, \text{LR}]$. In the present paper we characterize the function class $[\text{IN}; \text{SUB}, \text{LR}, \text{E}^{1/2}]$. Furthermore, we consider the class $[\text{IN}; \text{SUB}, \text{LR}, \text{E}]$ obtained similarly by adding the *bounded (full) quantifier* E. For functions $f \colon (\Gamma^*)^2 \rightharpoonup \Gamma^*$ and $b \colon \Gamma^* \rightharpoonup \Gamma^*$, $\text{E}(f, b)$ is the function given by

$$\text{E}(f, b)(x) = \begin{cases} 1 & \text{if } x \in \text{dom}(b) \text{ and there is some } y \in \Gamma^* \\ & \text{such that } |y| \leq |b(x)| \text{ and } f(x, y) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Unbounded versions of both quantifiers have been applied in various areas of computability theory with great success. A modification (continuous extension) of the half quantifier has been used by Plotkin to characterize the Ershov-Scott higher type partial computable functionals over the natural numbers [12]. Recursion over structures extended (among others) by the full quantifier can be used to give a natural and elegant treatment of positive, elementary induction. In the case of the natural numbers various classes of functions and relations like the hyperarithmetical functions, the arithmetical relations and the Π_1^1 relations can be identified. (See Moschovakis [10] for a general treatment.) In higher type recursion theory the important class of normal type 2 functionals is just the collection of functionals in which E is recursive [11].

Let us first study the effect of adding the bounded half-quantifier to the closure operations SUB and LR.

Theorem 1 $[\text{IN}; \text{SUB}, \text{LR}, \text{E}^{1/2}]$ *is the class $\mathcal{PNPF}$ of functions from Γ^* to Γ^* computable by a polynomial-time-bounded deterministic oracle Turing transducer with an $\mathcal{NP}$ set as oracle, which works in such a way that the machine in case of a negative oracle answer starts an infinite deterministic computation in the course of which the oracle is never queried again.*

This means that the function algebra $[\text{IN}; \text{SUB}, \text{LR}, \text{E}^{1/2}]$ allows the specification of polynomial-time algorithms in which decision problems can be solved by using the full power of nondeterministic polynomial-time computations, whereas all other kind of computations proceed deterministically.

Obviously, $\mathcal{PF} \subseteq \mathcal{PNPF} \subseteq \mathcal{NPSV}$, where the last set is the collection of all functions from Γ^* to Γ^* computable by a nondeterministic polynomial-time-bounded Turing transducer which on all finite computation paths computes the same result. The function class $\mathcal{NPSV}$ as well as the class $\mathcal{NPMV}$ of multi-valued functions from Γ^* to Γ^* computable by a nondeterministic polynomial-time-bounded Turing transducer have first been studied in [5].

The properness of these inclusions is connected with certain unsolved problems in complexity theory.

Lemma 2 *1. $\mathcal{P} = \mathcal{NP} \Rightarrow \mathcal{PNPF} = \mathcal{NPSV}$.*

2. $\mathcal{PNPF} = \mathcal{NPSV} \Rightarrow \mathcal{P} = \mathcal{NP} \cap \text{co-}\mathcal{NP}$.

3. $\mathcal{PNPF} \neq \mathcal{NPSV} \Leftrightarrow \mathcal{NP}$ contains $\mathcal{P}$-inseparable sets.

Note that two subsets A and B of Γ^* are $\mathcal{P}$-*inseparable* if no set L with the property $A \subseteq L \subseteq \Gamma^* \setminus B$ is in $\mathcal{P}$.

As has been shown in [15], $\mathcal{PNPF}$ is the class of functions from Γ^* to Γ^* which are computed by pairs (N, D) consisting of a nondeterministic polynomial-time-bounded Turing acceptor N and a deterministic polynomial-time-bounded Turing transducer D in such a way that first the acceptor N starts its computation and only if it halts in an accepting state after polynomial many steps, the transducer starts to work. In any other case the combined machine diverges.

Deterministic polynomial-time-bounded *parallel*, or *nonadaptive*, oracle Turing computations, where all queries are listed before any of them is made, have been studied in [16, 13]. But in that case also the information obtained from a negative oracle answer is used, which means that the computation is no longer purely nondeterministic.

Theorem 1 and the above characterization of $\mathcal{PNPF}$ implies that functions in $[\mathrm{IN}; \mathrm{SUB}, \mathrm{LR}, \mathrm{E}^{1/2}]$ can be brought into a normal form.

Theorem 3 *For every function* $f \in [\mathrm{IN}; \mathrm{SUB}, \mathrm{LR}, \mathrm{E}^{1/2}]$ *there are total functions* $g, h, b \in [\mathrm{IN}; \mathrm{SUB}, \mathrm{LR}]$ *such that* $f = \mathrm{SUB}(\mathrm{pr}_1^{(2)}, h, \mathrm{E}^{1/2}(g, b))$.

The next result exhibits some useful properties of the class $\mathcal{PNPF}$. Let to this end for a set $A \subseteq \Gamma^*$, $\chi_A \colon \Gamma^* \to \Gamma$, defined by

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{otherwise} \end{cases}$$

be the *characteristic function* of A. Moreover, call a function $f \colon \Gamma^* \rightharpoonup \Gamma^*$ *polynomial-time invertible* if there is a function $g \in \mathcal{DPF}$ such that $f(g(y)) = y$, for all $y \in \mathrm{range}(f)$.

Proposition 4 *Let A be a subset of Γ^*. Then the following statements hold:*

1. $A \in \mathcal{NP} \cap \mathrm{co}\text{–}\mathcal{NP} \Leftrightarrow \chi_A \in \mathcal{PNPF}$.

2. $A \in \mathcal{NP} \Leftrightarrow (\exists f \in \mathcal{PNPF}) A = \mathrm{range}(f) \wedge f$ *is polynomial-time invertible.*

3. $A \in \mathcal{NP} \cap \mathrm{co}\text{–}\mathcal{NP}$
   $\Leftrightarrow (\exists f \in \mathcal{PNPF}) A = \mathrm{range}(f) \wedge f$ *is total* $\wedge f$ *is polynomial-time invertible.*

4. $A \in \mathcal{P} \Leftrightarrow (\exists f \in \mathcal{PF}) A = \mathrm{range}(f) \wedge f$ *is total* $\wedge f$ *is polynomial-time invertible.*

The remark preceding Theorem 3 seems to indicate that machines computing functions in $\mathcal{PNPF}$ are less powerful than machines computing functions in $\mathcal{NPSV}$. Let us now study the expressive power of the bounded full quantifier.

Theorem 5 $[\mathrm{IN}; \mathrm{SUB}, \mathrm{LR}, \mathrm{E}]$ *is the class* $\mathcal{PF}(\mathcal{PH})$ *of functions from* Γ^* *to* Γ^* *computable by a polynomial-time-bounded oracle Turing transducer which may consult a set in the polynomial-time hierarchy $\mathcal{PH}$ as oracle.*

As is well known $\mathcal{PH} = \bigcup_{i \in \omega} \Sigma_i^p$, where the classes Σ_i^p are inductively defined by $\Sigma_0^p = \mathcal{P}$ and $\Sigma_{i+1}^p = \mathcal{NP}(\Sigma_i^p)$, for $i \geq 0$. Theorem 5 is a consequence of the following lemma.

Lemma 6 *Let $L_0 = [\mathrm{IN}; \mathrm{SUB}, \mathrm{LR}]$ and for $i \geq 0$,*

$$L_{i+1} = [\mathrm{IN} \cup \{ \mathrm{E}(f, b) \mid f, b \in L_i \}; \mathrm{SUB}, \mathrm{LR}].$$

Then $L_i = \mathcal{PF}(\Sigma_i^p)$.

The classes $\mathcal{PF}(\Sigma_i^p)$ have also been considered in [4]. Moreover, they have been characterized by Bellantoni [1] with the help of bounded minimization: A function is in $\mathcal{PF}(\Sigma_i^p)$ just if it can be obtained from initial functions in IN by substitution, limited recursion and bounded minimization, where the last operation can be nested at most i-times. This result should be compared with the above lemma, where instead of minimization the weaker operation of existential quantification is used.

Note that $\mathcal{NPSV} \subseteq \mathcal{PF}(\mathcal{PH})$. In [15] it is shown that

$$\mathcal{NPSV} \neq \mathcal{PF}(\mathcal{PH}) \Leftrightarrow \mathcal{NP} \neq \text{co-}\mathcal{NP}.$$

Though the half quantifier has turned out to be quite powerful in higher type computations, using its bounded version as a further closure operator in addition to substitution and limited recursion for the generation of functions out of the initial functions in IN results in exactly the restrictions of the polynomial-time computable functions to sets in $\mathcal{NP}$. The computational model of this function class seems to be rather weak. On the other hand, employing the bounded version of the full quantifier instead leads to the set of functions computable by a polynomial-time-bounded Turing machine that may consult a set in the polynomial-time hierarchy as oracle, which is quite a powerful model of computation, probably stronger than single-valued nondeterministic polynomial-time computations. If one strengthens the bounded half-quantifier by allowing parallel existential queries, one obtains a machine-independent characterization of the class $\mathcal{NPMV}$ of multi-valued nondeterministic polynomial-time computable functions.

Multi-valued functions form Γ^* to Γ^* are partial maps from Γ^* to 2^{Γ^*}. For multi-valued functions $k_1, \ldots, k_n$ of arity m and h of arity n the *substitution* $\text{SUB}(h, k_1, \ldots, k_n)$ of $k_1, \ldots, k_n$ in h is defined by

$$\text{SUB}(h, k_1, \ldots, k_n)(\bar{x}) = h(k_1(\bar{x}), \ldots, k_n(\bar{x}))$$
$$= \bigcup \{ h(y_1, \ldots, y_n) \mid y_i \in k_i(\bar{x}), \text{for } i = 1, \ldots, n \},$$

for all $\bar{x} \in (\Gamma^*)^m$ such that $k_i(\bar{x})$ is defined, for $i = 1, \ldots, n$. In any other case the function $\text{SUB}(h, k_1, \ldots, k_n)$ remains undefined.

The operation of *limited recursion* is lifted to multi-valued functions as follows. For functions g of arity n, g_0, g_1 of arity $n+2$ and b of arity $n+1$, $\text{LR}(g_0, g_1, g_2, b)$ is the uniquely determined function h of arity $n+1$ which satisfies the following three conditions:

1. $h(\bar{x}, \varepsilon) = \{ z \in g(\bar{x}) \mid |z| \leq |b(\bar{x}, \varepsilon)| \}$, if $g(\bar{x})$ and $b(\bar{x}, \varepsilon)$ are both defined and this set is nonempty.

2. For $i = 0, 1$, $h(\bar{x}, yi) = \{ z \in g_i(\bar{x}, y, h(\bar{x}, y)) \mid |z| \leq |b(\bar{x}, yi)| \}$, if $h(\bar{x}, y)$, $g_i(\bar{x}, y, h(\bar{x}, y))$ and $b(\bar{x}, yi)$ are defined and this set is nonempty.

3. In any other case $h(\bar{x}, y)$ is undefined.

Note that for a set $A \subseteq \Gamma^*$ and an element $z \in \Gamma^*$ we write $|z| \leq |A|$ to mean that $|z| \leq |x|$, for some $x \in A$.

The *parallel quantifier* $\text{E}^{\parallel}$ is defined in the following way. For functions f of arity 2 and b of arity 1, $\text{E}^{\parallel}(f, b)$ is the function of arity 1 with $\text{E}^{\parallel}(f, b)(x) \subseteq \{0, 1\}$ such that the following three conditions hold:

1. $0 \in \text{E}^{\parallel}(f, b)(x)$, if $x \in \text{dom}(b)$ and there is some $y \in \Gamma^*$ such that $|y| \leq b(|0x|)$ and $1 \in f(0x, y)$.

4

2. $1 \in \mathrm{E}^{\|}(f, b)(x)$, if $x \in \mathrm{dom}(b)$ and there is some $y \in \Gamma^*$ such that $|y| \leq b(|1x|)$ and $1 \in f(1x, y)$.

3. In any other case $\mathrm{E}^{\|}(f, b)(x)$ is undefined.

Single-valued functions like the initial functions in IN are considered as special multi-valued functions by identifying x with $\{x\}$.

Theorem 7 $[\mathrm{IN}; \mathrm{SUB}, \mathrm{LR}, \mathrm{E}^{\|}] = \mathcal{NPMV}$.

For the proof of this result we use the fact that for every polynomial-time-bounded nondeterministic Turing transducer a deterministic polynomial-time computable function f can be constructed such that $f(x) \in \mathrm{SAT}$, for $x \in \Gamma^*$, exactly if the Turing transducer on input x stops after polynomially many steps with the output written on the output tape [3]. The function f can be constructed in such a way that from any sequence y witnessing that $f(x) \in \mathrm{SAT}$ one can compute in polynomial time a path in the computation tree of the Turing transducer on input x. From every such path the result of the computation can easily be read off. Moreover, given a query $f(x)$ to SAT one can readily compute the number $n(x)$ of its propositional variables. Since a formula $\varphi(x_1, \ldots, x_n)$ is satisfiable if and only if at least one of the two *shorter* formulas $\varphi(0, x_2, \ldots, x_n)$ or $\varphi(1, x_2, \ldots, x_n)$ is satisfiable, a witness y for $f(x) \in \mathrm{SAT}$ can be computed by $n(x)$ applications of the parallel quantifier $\mathrm{E}^{\|}$.

A variety of other characterizations of $\mathcal{NPMV}$ has been presented in [14].

References

[1] S. Bellantoni, Predicative recursion and the polytime hierarchy, in: P. Clote and J. Remmel, eds., *Feasible Mathematics II*, Birkhäuser, Boston, 1995.

[2] S. Bellantoni and S. Cook, A new recursion-theoretic characterization of the polytime functions, *Computational Complexity* 2 (1992) 97–110.

[3] S. A. Cook, The complexity of theorem-proving procedures, in: *Proc. 3rd Ann. ACM Symp. on Theory of Computing*, Association for Computing Machinery, New York, 1971, 151–158.

[4] S. R. Buss, *Bounded Arithmetic*, Bibliopolis, Napoli, 1986.

[5] R. V. Book, T. J. Long and A. L. Selman, Quantitative relativizations of complexity classes, *SIAM J. Comput.* 13 (1984) 461–487.

[6] A. Cobham, The intrinsic computational difficulty of functions, in: Y. Bar-Hillel, ed., *Logic, Methodology and Philosophy of Science II*, North-Holland, Amsterdam, 1965, 24–30.

[7] J.-Y. Girard, A. Scedrov and P. Scott, Bounded linear logic: a modular approach to polynomial time computability, *Theoret. Comput. Sci.* 97 (1992) 1–66.

[8] D. Leivant, A foundational delineation of poly-time, *Inform. and Computation* 110 (1994) 391–420.

[9] D. Leivant and J.-Y. Marion, Lambda calculus characterizations of poly-time, *Fund. Inf.* 19 (1993) 167–184.

[10] Y. N. Moschovakis, Abstract recursion as a foundation for the theory of algorithms, in: M. M. Richter et al., eds., *Computation and Proof Theory, Proc., Logic Colloq. Aachen 1983, Part II*, Lec. Notes Math. 1104, Springer-Verlag, Berlin, 1984, 289–364.

[11] D. Normann, *Recursion on the Countable Functionals*, Lec. Notes Math. 811, Springer-Verlag, Berlin, 1980.

[12] G. D. Plotkin, LCF considered as a programming language, *Theoret. Comput. Sci.* 5 (1977) 223–255.

[13] A. L. Selman, A taxonomy of complexity classes of functions, *J. Comput. Systems Sci.* 48 (1994) 357–381.

[14] D. Spreen, On functions computable in nondeterministic polynomial time: some characterizations, in: E. Börger et al., eds, *CSL'87, 1st Workshop on Computer Science Logic, Proc.*, Lec. Notes Comput. Sci. 329, Springer-Verlag, Berlin, 1988, 289–303.

[15] D. Spreen and H. Stahl, On the power of single-valued nondeterministic polynomial time computations, in: E. Börger, ed., *Computation Theory and Logic*, Lec. Notes Comput. Sci. 270, Springer-Verlag, Berlin, 1987, 403–414.

[16] K. Wagner, On restricting the access to an NP-oracle, in: T. Lepistö and A. Salomaa, eds., *Automata, Languages and Programming, 15th Intern. Conf., Proc.*, Lec. Notes Comput. Sci. 317, Springer-Verlag, Berlin, 1988, 682–696.

[17] K. Weihrauch, Teilklassen primitiv-rekursiver Wortfunktionen, Bericht Nr. 91, Gesellschaft f. Mathematik u. Datenverarbeitung, St. Augustin, 1973.