A Survey of Classes of Primitive Recursive Functions

Stephen A. Cook

Abstract

This paper is a transcription of mimeographed course notes titled “A Survey of Classes of Primitive Recursive Functions”, by S.A. Cook, for the University of California Berkeley course Math 290, Sect. 14, January 1967. The notes present a survey of subrecursive function classes (and classes of relations based on these classes,) including Cobham's class $\mathcal{L}$ of polynomial time functions, and Bennett's class (denoted here by $\mathcal{L}^+$) of extended positive rudimentary functions. It is noted that $\mathcal{L}^+$ corresponds to those functions computable in nondeterministic polynomial time and that $\mathcal{L} \subseteq \mathcal{L}^+$, and it is conjectured that this inclusion is proper. Relational versions of these classes are also introduced, and a similar inclusion is noted. This is likely the earliest consideration in print of the relationship between the complexity classes P and NP, in both functional and relational forms.

The numbering of sections and theorems corresponds to that in the original notes. However, page numbering does not correspond to the page numbering of the original. Minor typographical errors have been corrected.

–Bruce Kapron, December 15, 2016

I Basic Notions

All functions considered here take tuples of non-negative integers into non-negative integers. We use the notation $\underline{x}$ for $x_1, \ldots, x_p$, where p is usually not specified.

Relations vs. functions If $\mathcal{F}$ is a class of functions, then $\mathcal{R}_{\mathcal{F}}$, the $\mathcal{F}$-relations, is the class of relations whose characteristic function is in $\mathcal{F}$. Conversely, if $\mathcal{R}$ is a class of relations, then it is possible to associate a class $\mathcal{F}$ of functions with $\mathcal{R}$ by saying $f \in \mathcal{F}$ iff first the relation $y = f(\underline{x})$ is in $\mathcal{R}$, and second f is bounded by a function from some appropriate class. For example, in case $\mathcal{R}$ is the class of constructive arithmetic relations the appropriate class of bounding functions turns out to be the class of polynomials. If we start with a class $\mathcal{F}$ of functions which includes the function $x = y$ and is closed under substitution and limited minimalization, then the class of functions associated with $\mathcal{R}_{\mathcal{F}}$ is again precisely $\mathcal{F}$, provided the class of bounding functions is chosen to be cofinal (see below) with $\mathcal{F}$. On the other hand, suppose we start with a class $\mathcal{R}$ of relations which includes the identity relation and is closed under explicit transformation and the Boolean operations. If we pass to the associated class $\mathcal{F}$ of functions using any bounding class which includes the constant function $\mathcal{I}$, then $\mathcal{R}_{\mathcal{F}}$ is precisely $\mathcal{R}$.

Cofinal Classes Two classes $\mathcal{F}_0$ and $\mathcal{F}_1$ are cofinal if for every $f \in \mathcal{F}_i$ there is a $g \in \mathcal{F}_{i-1}$ such that $f(\underline{x}) \leq g(\underline{x})$ for all $\underline{x}$ $(i = 0, 1)$.

Explicit Transformation A class $\mathcal{F}$ of functions is closed under explicit transformation if, whenever $g \in \mathcal{F}$, there is an $f \in \mathcal{F}$ such that $f(\underline{x}) = g(\underline{t})$ holds identically, where each t_i is either an x_j or a constant. For example, perhaps $f(x_1, x_2, x_3) = g(x_3, 2, x_3, x_1)$. Similarly for classes of relations.

Substitution $\mathcal{F}$ is closed under substitution if it is closed under both explicit transformation and composition.

Boolean Operations The three Boolean operations are negation (complementation), finite conjunction (intersection), and finite disjunction (union). These apply to relations.

Bounded (i.e. limited) quantification The two operations $\exists_{\leq}$ and $\forall_{\leq}$ apply to relations, and are defined as follows: $(\exists_{\leq} R)(\underline{x}, y)$ holds iff $R(\underline{x}, z)$ holds for some $z \leq y$, and $(\forall_{\leq} R)(\underline{x}, y)$ holds iff $R(\underline{x}, z)$ holds for all $z \leq y$.

Bounded (i.e. limited) recursion f is defined from g, h, k by limited recursion provided the following holds for all $\underline{x}, y$.

$$\begin{aligned} f(\underline{x}, 0) &= g(\underline{x}) \\ f(\underline{x}, y+1) &= g(\underline{x}, f(\underline{x}, y), y) \\ f(\underline{x}, y) &\leq k(\underline{x}, y) \end{aligned}$$

m-adic notation (We always assume $m \geq 2$ when speaking of m-adic notation). The m-adic notation for the positive integer n is the unique string $d_k d_{k-1} \ldots d_0$ of digits from the alphabet $\{1, 2, \ldots, m\}$ such that

$$n = \sum_{l=0}^{k} d_i m^i.$$

The m-adic notation for 0 is the empty string. Switching back and forth from m-adic to m-ary (radix) notation involves very little computation. m-adic (as opposed to m-ary) notation sets up a one-one correspondence between strings and non-negative integers. The ordering induced on strings by their m-adic value is the one determined first by length, and among strings of the same length, the ordering is lexicographical.

Bounded (i.e. limited) recursion on notation f is defined from $g, h_1, \ldots, h_m$, and k by limited recursion on (m-adic) notation provided the following hold for all $\underline{x}, y$.

$$\begin{aligned}
&f(\underline{x}, 0) = g(\underline{x})\\
&f(\underline{x}, y * i) = h_i(\underline{x}, f(\underline{x}, y), y), i = 1, 2, \ldots, m\\
&f(\underline{x}, y) \leq k(\underline{x}, y)
\end{aligned}$$

Here $*$ is the m-adic concatenation function.

Subpart quantification The two operations $\exists^m$ $\forall^m$ apply to relations and are defined as follows. $(\exists^m R)(\underline{x}, y)$ holds iff $R(\underline{x}, z)$ holds for some z whose m-adic notation is a consecutive substring (possibly all or empty) of the m-adic notation for y, and $(\forall^m R)(\underline{x}, y)$ holds iff $R(\underline{x}, z)$ holds for all such z.

II The Grzegorczyk Hierarchy

(See Grzegorczyk [4]) This is a sequence $\mathcal{E}^0 \subseteq \mathcal{E}^1 \subseteq \ldots$ of classes of functions whose union is precisely the class of prmitive recursive functions. First let us define a sequence $\xi_0(x, y), \xi_1(x, y), \ldots$ of functions by

$$\begin{aligned}
\xi_0(x, y) &= y + 1\\
\xi_{n+1}(x, 0) &= \begin{cases} x & \text{if } n = 0\\ 0 & \text{if } n = 1\\ 1 & \text{if } n > 1 \end{cases} \quad \begin{bmatrix}\text{DEFINE SUCH THAT FUNC-}\\ \text{TION IS ALWAYS THERE}\end{bmatrix}\\
\xi_{n+1}(x, y + 1) &= \xi_n(x, \xi_{n+1}(x, y)).
\end{aligned}$$

Note that $\xi_1(x, y) = x + 1$, $\xi_2(x, y) = xy$ and $\xi_3(x, y) = x^y$. Then $\mathcal{E}^n$ can be defined as the least class including the functions $x + 1$ and ξ_n, and closed under substitution and limited recursion. Grzegorczyk showed that $\mathcal{E}^3$ is just $\mathcal{E}$, the class of elementary functions of Kalmár. Each class is closed under limited minimalization and (at least for $n \geq 2$) limited recursion on notation, and the class of $\mathcal{E}^n$ relations is closed under the Boolean operations and bounded quantification for all n. $\mathcal{E}^{n+1}$ contains $\mathcal{E}^n$ properly for all n, and the class of $\mathcal{E}^{n+1}$ relations contains the class or $\mathcal{E}^n$ relations, at least for $n \geq 2$.

Theorem 1 was proved by Ritchie [6], and theorem 2 was stated by Cobham [3] and can be proved by Ritchie's methods. The theorems provide interesting characterizations for the functions of $\mathcal{E}^n$ in terms of their computation time and storage requirements.

Notation If Z is a Turing machine which computes a function $f(\underline{x})$ in m-adic notation, $\tau_Z(\underline{x})$ (the tape function for Z) is the number of tape squares used by Z in evaluating f at $\underline{x}$, and $\sigma(\underline{x})$ (the time function for Z) is the number of steps required by Z to evaluate f at $\underline{x}$.

Theorem 1. *$f(\underline{x}) \in \mathcal{E}^2$ iff there are constants C_1 and C_2 and a Turing machine Z which computes f in m-adic notation such that*

$$* \qquad \tau_Z(\underline{x}) \leq C_1(\sum_i \ell(x_i)) + C_2$$

for all $\underline{x}$. (Here $\ell(x_i)$ is the length of the m-adic notation for x_i).

Theorem 2.

(a) If $n \geq 3$, then $f(\underline{x}) \in \mathcal{E}^n$ iff there is a Turing Machine Z which computes f and a function $g \in \mathcal{E}^n$ such that $\tau_Z(\underline{x}) \leq g(\underline{x})$ for all $\underline{x}$.

(b) Same as (a), with $\sigma_Z(\underline{x})$ replacing $\tau_Z(\underline{x})$.

Remark 1. Of course Theorem 2 is equally valid if the functions g are chosen from some class of functions cofinal iwth $\mathcal{E}^n$ instead of $\mathcal{E}^n$ itself. For example, the class

$$\{\xi_{n_1}(\max \underline{x}, c_1) + c_2 \mid c_1, c_2 \text{ positive integers}\}$$

is cofinal with $\mathcal{E}^n$, $n \geq 0$.

Remark 2. Shepherdson-Sturgis machines. Theorems 1 and 2 remain valid when other computer models are used besides Turing machines, provided τ_Z and σ_Z are defined properly; and in particular the Turing machines may have several taps and several read/write heads per tape. The unlimited register machines of Shepherdson and Sturgis [7] will do as the computer model, provided $\tau_Z(\underline{x})$ is taken to be the length of the m-adic notation of the maximum number occurring in any register during the course of the computation with input $\underline{x}$. Then (*) in Theorem 1 is equivalent to requiring that the numbers in each register of the machine be bounded by polynomials[1] in $\underline{x}$, which in turn is equivalent (since $\mathcal{E}^2$ is cofinal with the polynomials) to requiring that the members of each register be bounded by some member of $\mathcal{E}^2$. In fact, in general for $n \geq 2$, $\mathcal{E}^n$ consists exactly of those functions computable by some Shepherdson-Sturgis machine in which *the numbers in all* registers are bounded by some member of $\mathcal{E}^n$.

Remark 3. There is no known characterization of $\mathcal{E}^2$ in terms of $\sigma_Z(\underline{x})$ analogous to the characterizations of $\mathcal{E}^n$ for $n > 2$ state in theorem 2. This is one reason for introducing the class $\mathcal{L}$ defined in the next section.

III Computation Time and Limited Recursion on Notation

Cobham [3] introduced the class $\mathcal{L}$ of functions which is defined in terms of computation time. A function $f(\underline{x})$ is in $\mathcal{L}$ iff there is a Turing machine Z which computes f in m-adic

[1] Robert Elschlager points out that the numbers can always be bounded by max $\underline{x}$ if the function computed is a characteristic function

notation and a polynomial $P(\underline{t})$ such that

$$\sigma_Z(x_1, \ldots, x_n) \leq P(\ell(x_1), \ldots, \ell(x_n))$$

for all $\underline{x}$. Cobham stated the following characterization of $\mathcal{L}$.

Theorem 3. *$\mathcal{L}$ is the least class of functions containing $S_i(x)$, $1 = 1, \ldots, m$, $x^{\ell(y)}$ and closed under substitution and limited recursion on notation. Here $S_i(x)$ is $x * i$ (the ith m-adic successor function.)*

The proof is similar to the proof of theorem 1. The class $\mathcal{L}$ is independent of the choice of m since a Turing machine can convert from m-adic to n-adic notation sufficiently rapidly.

The class $\mathcal{L}$, characterizable in terms of computation time requirements, is a natural analog of $\mathcal{E}^2$, characterizable in terms of storage requirements. Note the parallel between theorems 1 and 3, contrasting limited recursion with limited recursion on notation.

$\mathcal{E}^2$ does not contain $\mathcal{L}$ because the function $x^{\ell(y)}$ in $\mathcal{L}$ grows too fast to be in $\mathcal{E}^2$, but it is not known whether $\mathcal{L} \supseteq \mathcal{E}^2$. Cobham points out that the function $f(n) =$ the nth prime is known to be in $\mathcal{E}^2$, but suggests that it is too time consuming to compute to be in $\mathcal{L}$, that $\mathcal{L}$ and $\mathcal{E}^2$ are probably incomparable. Similarly, it is a good guess that the $\mathcal{L}$-relations and $\mathcal{E}^2$-relations are incomparable, or at least the former should not include the latter.

Extended Positive Rudimentary Functions, $\mathcal{L}^+$. These were introduced by Bennett [1], p. 67, and can be defined as the class of functions associated with the extended positive rudimentary relations (see section V), where the bounding functions are taken to be those of the form $x^{(\ell(y))^n+c}$ for arbitrary constants n and c. Bennett states that this class of functions, which we might call $\mathcal{L}^+$, is closed under substitution and limited recursion on notation, and since $\mathcal{L}^+$ certainly contains $x+1$ and $x^{\ell(y)}$, we can conclude

$$\mathcal{L} \subseteq \mathcal{L}^+$$

Whereas $\mathcal{L}$ can be characterized as consisting of those functions whose Turing machine computation time is bounded by a polynomial in the lengths of the arguments, $\mathcal{L}^+$ has the same characterization except that we must allow the Turing machines to be non-deterministic. It seems likely that this non-determinism increases the computing power of the machine, but this may be difficult to prove.

IV The Ritchie Hierarchy

Ritchie [6] introduced a sequence $\langle F_i \rangle$ of classes of functions satisfying

$$\mathcal{E}^2 \subseteq F_1 \subseteq F_2 \subseteq \cdots \subseteq \mathcal{E}^3$$

F_i consists of just those functions computable by a Turing machine Z whose tape function τ_Z satisfies

$$(**) \qquad \tau_Z(\underline{x}) \leq f_{i-1}(K(\max \underline{x}, 1))$$

for some constant K, where $f_0(x) = x$, $f_1(x) = 2^x$, and in general $f_{i+1}(x) = 2^{f_i(x)}$. For $i \geq 2$, (**) is equivalent to requiring $\tau_Z(\underline{x})$ be bounded by some member of F_{i-1}, since for each i the class of functions $\{f_i(K(\max \underline{x}, 1)) \mid K \text{a positive integer}\}$ is cofinal with F_i. Because of the latter characterization, Ritchie called the functions in F_i the "predicatively computable functions"; i.e., one always knows that the tape function for a member of F_i is bounded by some member of the class F_{i-1}, which has already been "obtained".

Since the class $\{f_i(K(\max \underline{x}, 1)) \mid i, K \text{positive integers}\}$ is cofinal with $\mathcal{E}^3$, it follows from theorem 2 that

$$\bigcup_{i=1}^{\infty} F_i = \mathcal{E}^3 \quad \text{(elementary functions)}$$

Also, by theorem 1, each F_i contains $\mathcal{E}^2$ properly. Ritchie used a diagonal argument to show that the F_i-relations are properly contained in the F_{i+1}-relations for all i.

Each class F_i is closed under explicit transformation, but none is closed under composition or limited recursion. Bennett [1], p. 74 points out the following characterization of F_1 follows from Ritchie's work: $f \in F_i$ iff $f(\underline{x}) = g(\underline{x}, f_1(K \max \underline{x})$ for some $g \in \mathcal{E}^2$ and integer K

V Other Classes

All of the following classes except the context-sensitive relations (and languages) are discussed by Bennett [1]. Smullyan [8] introduced the m-rudimentary relations and the constructive arithmetic relations.

Notation $x \overset{m}{*} y$ is the number whose m-adic notation is the concatenation of the m-adic notations for x and y.

The Strictly m-rudimentary relations are the least class of containing the three place relation $x \overset{m}{*} y = z$ and closed under explicit transformation, the Boolean operations, and subpart quantification. Bennett states that these are distinct for each m.

The positive m-rudimentary relations are the least class containing $x \overset{m}{*} y = z$ and closed under explicit transformation, conjunction, disjunction, subpart quantification, and $\exists_{\leq}$. Bennett shows these form the same class for each m.

The Strongly m-rudimentary relations are those relations R such that both R and $\neg R$ are positive m-rudimentary.

This class is independent of m and is closed under explicit transformation, the Boolean operations, and subpart quantification (Bennett).

The m-rudimentary relations are the least class of relations containing $x \overset{m}{*} y = z$ and closed under explicit transformation, the Boolean operations and bounded quantification.

The constructive arithmetic relations are the least class containing the two three-place relations $x + y = z$ and $x - y = z$ and closed under explicit transformation, the Boolean operations, and bounded quantification.

Bennett's main result in chapter I of [1] is that this class is the same as the class of m-rudimentary relations for each m.

The extended positive m-rudimentary relations are those of the form

$$(\exists y \leq g_k(\underline{x}))R(\underline{x}, y)$$

where $R(\underline{x}, y)$ is positive m-rudimentary, k is an integer, and $g_k(\underline{x}) = m^{(\ell(\max \underline{x}))^k}$, $\ell(z)$ is the length of the m-adic notation for z.

Using Theorem 3 in section III and Bennett's discussion pp. 62-67 it is easily shown that this class is (independent of m) precisely the closure of Cobham's class of $\mathcal{L}$-relations under the operation $\exists_{\leq}$. The class is also closed under disjunction, conjunction, explicit transformation, and subpart quantification.

A context-sensitive language is the set of strings over some alphabet generated by a semi-Thue system of whose productions $u \rightarrow v$ satisfy $\ell(u) \leq \ell(v)$. Kuroda [5] characterized these languages as those recognizable by some non-deterministic Turing machine whose tape function is bounded by some linear function of the length of the input string. This characterization suggests that a context sensitive relation be defined as one recognizable in the same way. Using m-adic notation we can consider these context senstive relations to be relations on integers, and it is easy to see the resulting class of relations will not depend on m. Then by Theorem 1, we find that the $\mathcal{E}^2$ relations are a subclass of the context sensitive relations.

Spectra The spectrum of a formula of the first order predicate calculus with equaliity is the set of all cardinalities of its finite models. Bennet generalized the notion of spectrum of formula from such a theory to a many-sorted theory of types by defining the spectrum of a formula from such a theory as the relation which is true on those types of integers which are the cardinalties of the basic domains of individuals for some finite model of the sentence. He denotes by $\mathcal{S}^n$ the class of spectra of n^{th} order formulas.

Bennett's main result is the following (p. 116, 125).

Theorem 4.

(a) *For each $n \geq 1$ and $m \geq 2$, $\mathcal{S}^{2n-1}$ is the class of relations of the form $(\exists y \leq g(\underline{x}))R(\underline{x}, y)$ where $g(\underline{x}) = f_n((\max(\underline{x}))^j)$ for some $j \geq 1$ (see Sec. IV for f_n) and R is strictly m-rudimentary.*

(b) *The same as (a) except R may be chosen to be any extended positive rudimentary relation.*

(c) *For each $n \geq 1$, $\mathcal{S}^{2n}$ is the class of relations of the forms $(\exists y \leq g(\underline{x}))R(\underline{x}, y)$ where R is constructive arithmetic and $g(\underline{x}) = f_n((\max(\underline{x}))^j)$ for some $j \geq 1$.*

(d) *For each* $n \geq 1$, $\mathcal{S}^n$ *is a subclass of* $\mathcal{S}^{n+1}$ *and a proper subclass of* $\mathcal{S}^{n+2}$.

(e) $\bigcup_{n=1}^{\infty} \mathcal{S}^n$ *is the class of* $\mathcal{E}^3$ *(elementary)-relations.*

(f) *The* F_1*-relations are a subclass of* $\mathcal{S}^3$, *and for each* $n \geq 2$, $\mathcal{S}^{2n-2}$ *is a subclass of the* F_n*-relations, which in turn are a subclass of* $\mathcal{S}^{2n+1}$. *Moreover, for no* n, $p \geq 1$ *is* $\mathcal{S}^p$ *identical with the class of* F_n*-relations.*

(g) *The constructive arithmetic relations form a proper subclass of* $\mathcal{S}^2$, *and the extended positive rudimentary relations form a proper subclass of* $\mathcal{S}^1$.

VI Summary of Facts and Open Questions

The chart on the next page indicates the inclusion relationships among most of the classes of relations discussed earlier. A line from one class to one above it indicates the higher class contains the lower. If the inclusion is known to be proper, the line is so labelled. The numbers on the lines refer to the following list of sources for the proofs of inclusion.

Sources

1, 8. Stated by Bennett, p. 13

2, 3. Bennett, p. 13 Immediate from definitions

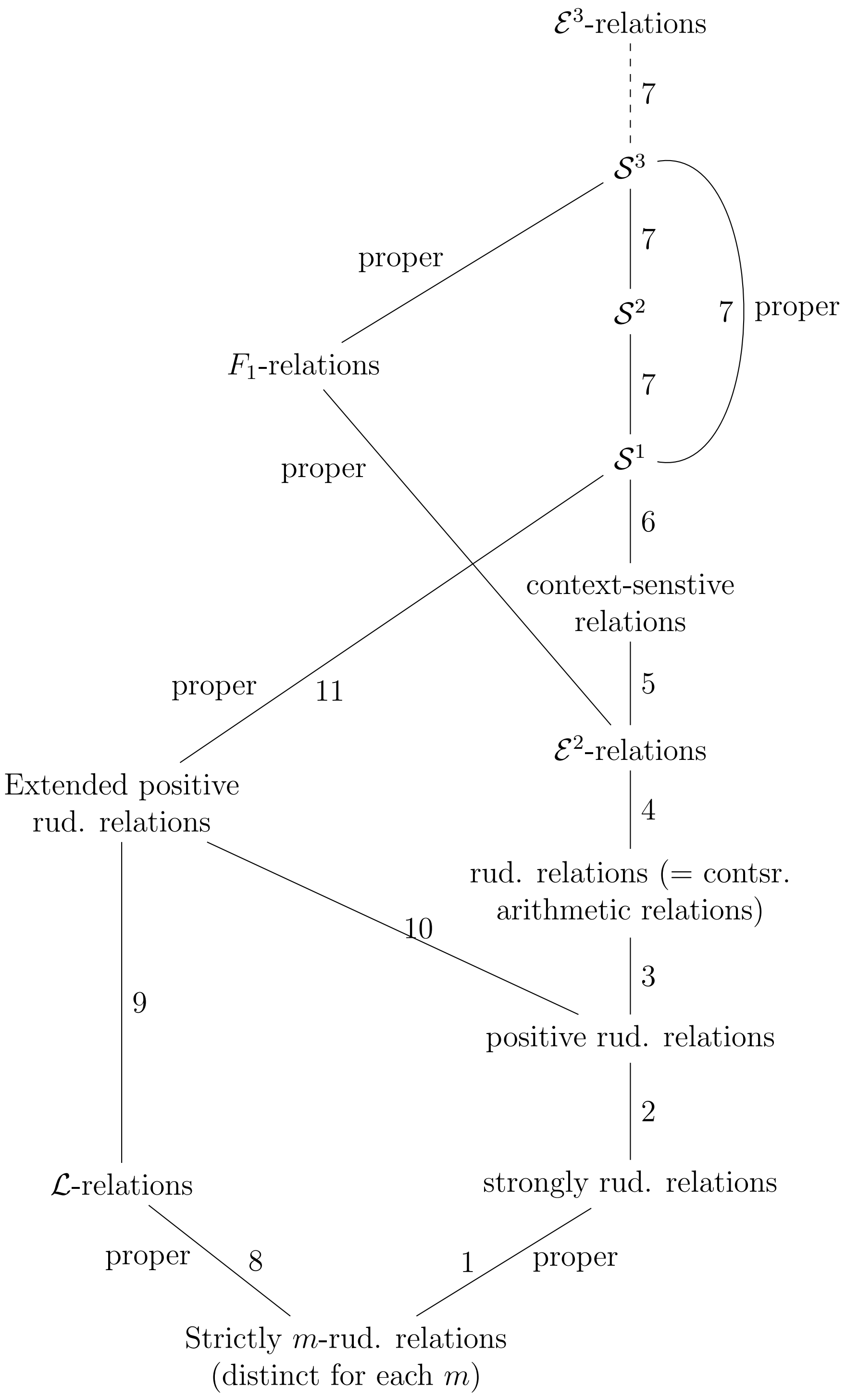
4 Bennett, p. 75 Follows immediately form the definitions and the fact that $\mathcal{E}^2$-relations are closed under explicit transformation, the Boolean operations, and limited quantification

5 Kuroda [5] and Theorem 1.

6 Theorem 4, part (a), Kuroda's characterization of context-senstive languages, and an easy argument.

7,11 Theorem 4 (Bennett).

9,10 See under definition of extended positive rudimentary relations.


$\mathcal{E}^3$-relations
7
$\mathcal{S}^3$
proper
7
$\mathcal{S}^2$
7
proper
F_1-relations
7
proper
$\mathcal{S}^1$
6
context-senstive relations
proper
11
5
$\mathcal{E}^2$-relations
Extended positive rud. relations
4
rud. relations (= contsr. arithmetic relations)
10
9
3
positive rud. relations
2
$\mathcal{L}$-relations
strongly rud. relations
proper
8
1
proper
Strictly m-rud. relations (distinct for each m)

Closure under operations of relation classes All the classes of relations discussed previously are closed under expliicit transformation, subpart quantification, disjunction, and conjunction. This follows from the definitions either directly or by easy arguments. The following table indicates which classes are known to be closed under negation, $\exists \leq$ and $\forall \leq$. It is tempting to argue that where a "?" appears the answer is probably most often "no", partly from intuition, and partly because plenty of techniques for proving positive results are known, but very few for proving negative results are known.

One of the more interesting questions is whether the positive m-rudimentary relations are closed under negation. If the answer is yes, then this class is the same as the class of constructive arithmetic relations, so $\mathcal{S}^{2n-1} = \mathcal{S}^{2n}$ for all $n \geq 1$, so $\mathcal{S}^n$ would be closed under negation for all n.

	negation	$\exists \leq$	$\forall \leq$	Source
$\mathcal{E}^n$-relations, $n \geq 0$	yes	yes	yes	Grzegorczck [4]
F_n-relations, $n \geq 1$	yes	yes	yes	See section IV
$\mathcal{L}$-relations	yes	?	?	See section III
Strictly m-rudimentary relations	yes	no	no	
Positive m-rudimentary relations	?	yes	?	Bennett, p. 13
Constructive arithmetic relations	yes	yes	yes	
Extended positive rudimentary relations	?	yes	?	Bennett, p. 62
Context sensitive relations	?[2]	yes	yes	See definition, Sec. V
$\mathcal{S}^n$, n odd	?	yes	yes	Bennett, p. 124
$\mathcal{S}^n$, n even	yes	yes	yes	

Functions Each of the classes $\mathcal{E}^n$, $n \geq 0$, F_n, $n \geq 1$, $\mathcal{L}$, $\mathcal{L}^+$ is closed under explicit transformation, composition, and limited recursion on notation. In addition, all except F_n and possibly $\mathcal{L}$ and $\mathcal{L}^+$ are closed under limited recursion.

References

[1] Bennet, James H., *On Spectra.* Doctoral Dissertation, Princeton University, 1962.

[2] Chomsky, Noam, On certain formal properties of grammars, *Information and Control*, vol. 2, 1959, pp. 137-167.

[3] Cobham, Alan, The intrinsic computational difficulty of functions. *Proc. of the 1964 International Congress for Logic, Methodology, and the Philosophy of Science.* North Holland Publishing Co., Amsterdam, pp. 24-30.

[4] Grzegorczyk, Andrez, Some classes of recursive functions. *Rozprawy Matematyczne*, Warsaw, 1953.

[5] Kuroda, S.Y., Classes of languages and linear-bound automata. *Information and Control*, vol. 3, 1964, pp. 207-223.

[2] (Addition to original text): This is now known to be "yes", by the Immerman-Szelepcsényi Theorem.

[6] Ritchie, Robert W., Classes of predictably computable functions. *Trans. Amer. Math. Soc.*, Jan. 1963.

[7] Shepherdson, J.C. and H.E. Sturgis, Computability of recursive functions, *J. Assoc. for Computing Machinery*, 10, 1963, pp. 217-255.

[8] Smullyan, Raymond M., *Theory of Formal Systems, Revised Edition*, Annals of Mathematical Studies No. 47, Princeton, 1961.