Primitive Recursiveness vs. Definability by Bounded Formulas

Saeed Salehi

University of Tabriz

http://SaeedSalehi.ir/

Isfahan Seminar on Mathematical Logic

30 Nov. & 01 Dec. 2011

Reminding P.R.'s

- $Z(x) = 0$ • $S(x) = x + 1$ • $P_i^n(x_1, \ldots, x_n) = x_i \in \mathrm{P.R.}$
- $g, h_1, \ldots, h_m \in \mathrm{P.R.} \,\&\, f(\overline{x}) = g(h_1(\overline{x}), \ldots, h_m(\overline{x})) \Rightarrow f \in \mathrm{P.R.}$
- $g, h \in \mathrm{P.R.} \,\&\, \begin{cases} f(\overline{x}, 0) = g(\overline{x}) \\ f(\overline{x}, y+1) = h(\overline{x}, y, f(\overline{x}, y)) \end{cases} \Rightarrow f \in \mathrm{P.R.}$

▶ $P \subseteq \mathbb{N}^k$: $\quad P \in \mathrm{P.R.} \iff \chi_P \in \mathrm{P.R.} \quad \chi_P(\overline{x}) = \begin{cases} 1 & \text{if } \overline{x} \in P \\ 0 & \text{if } \overline{x} \notin P \end{cases}$

Reminding P.R.'s (continued)

$$+, \quad \times, \quad sg(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } x \neq 0 \end{cases}, \quad \overline{sg}(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases} \in \text{P.R.}$$

- $\chi_{P\cap Q} = \chi_P \times \chi_Q$
- $\chi_{P\cup Q} = sg(\chi_P + \chi_Q)$
- $\chi_{P^\complement} = \overline{sg}(\chi_P)$
- $\chi_{P-Q} = \chi_P \times \overline{sg}(\chi_Q)$

$$\begin{cases} \chi_{\forall x\leqslant\alpha\, P(\overline{z},x)}(\overline{z},0) = \chi_P(\overline{z},0) \\ \chi_{\forall x\leqslant\alpha\, P(\overline{z},x)}(\overline{z},\alpha+1) = \chi_{\forall x\leqslant\alpha\, P(\overline{z},x)}(\overline{z},\alpha) \times \chi_P(\overline{z},\alpha+1) \end{cases}$$

$$\begin{cases} \chi_{\exists x\leqslant\alpha\, P(\overline{z},x)}(\overline{z},0) = \chi_P(\overline{z},0) \\ \chi_{\exists x\leqslant\alpha\, P(\overline{z},x)}(\overline{z},\alpha+1) = sg(\chi_{\forall x\leqslant\alpha\, P(\overline{z},x)}(\overline{z},\alpha) + \chi_P(\overline{z},\alpha+1)) \end{cases}$$

All Δ_0-Definable Predicates are Primitive Recursive

- If $P, Q \in \mathrm{P.R.}$ then so are $P \cap Q, P \cup Q, P^{\complement}, P - Q$, etc.
- Also $\forall x \leqslant y\, P(x, y, \overline{z})$ and $\exists x \leqslant y\, Q(x, y, \overline{z})$ are in P.R.

$$\Delta_0 :=| \texttt{ATOMS} \mid \Delta_0 \wedge \Delta_0 \mid \Delta_0 \vee \Delta_0 \mid \Delta_0 \rightarrow \Delta_0 \mid \neg\Delta_0 \mid$$
$$\mid \forall x \leqslant y\, \Delta_0(\overline{z}, x, y) \mid \exists x \leqslant y\, \Delta_0(\overline{z}, x, y) \mid$$

For any Δ_0-formula θ we have $\{\overline{a} \mid \mathbb{N} \models \theta(\overline{a})\} \in \mathrm{P.R.}$

Δ_0-Definable and P.R. Functions

For Functions: Δ_0-definable:

exists $\theta(\overline{x}, y) \in \Delta_0$ s.t. $f(\overline{a}) = b \iff \mathbb{N} \models \theta(\overline{a}, b)$.

⊳ Is Every Δ_0-Definable Function P.R.?

⊳ Is Every P.R. Function Δ_0-Definable?

P.R. Functions and Relations vs. Δ_0−Definability

$$R \in \Delta_0 \Longrightarrow R \in \text{P.R.} \quad \checkmark$$

$$R \in \text{P.R.} \Longrightarrow R \in \Delta_0 \quad ?$$

– – – – – – – – – – – ––

$$f \in \Delta_0 \Longrightarrow f \in \text{P.R.} \quad ?$$

$$f \in \text{P.R.} \Longrightarrow f \in \Delta_0 \quad ?$$

P.R. Functions and Relations vs. Δ_0−Definability (continued)

$f : X \to Y \qquad \Gamma_f \subseteq X \times Y$

$$\bullet\ f \in \Delta_0 \iff \Gamma_f \in \Delta_0$$

▶ $f \in \text{P.R.} \Longrightarrow \Gamma_f \in \text{P.R.}$:

$$\chi_{\Gamma_f}(\overline{a}, b) = \chi_{=}(f(\overline{a}), b) = sg(|f(\overline{a}) - b|)$$

⊳⊳ $\Gamma_f \in \text{P.R.} \Longrightarrow f \in \text{P.R.}$?

$$f \in \text{Rec.} : f(\overline{a}) = \mu y.\Gamma_f(\overline{a}, y)$$

$R \subseteq X \qquad \chi_R : X \to \{0, 1\}$

$$\bullet\ R \in \text{P.R.} \iff \chi_R \in \text{P.R.}$$

▶ $R \in \Delta_0 \iff \chi_R \in \Delta_0$:

$$\Gamma_{\chi_R}(\overline{x}, y) \equiv \big(y = 1 \wedge R(\overline{x})\big) \vee \big(y = 0 \wedge \neg R(\overline{x})\big)$$

$$R(\overline{x}) \iff \chi_R(\overline{x}) = 1 \iff \Gamma_{\chi_R}(\overline{x}, 1)$$

P.R. Functions and Relations vs. Δ_0–Definability ?

$$R \in \Delta_0 \Longrightarrow R \in \text{P.R.} \qquad \checkmark$$

$$(R \in \text{P.R.} \Longrightarrow R \in \Delta_0) \qquad ?$$

$$\Updownarrow \quad \Updownarrow$$

$$(f \in \text{P.R.} \Longrightarrow f \in \Delta_0) \qquad ?$$

$$f \in \Delta_0 \Longrightarrow f \in \text{P.R.} \qquad ?$$

$$\Gamma_f \in \text{P.R.} \Longrightarrow f \in \text{P.R.} \qquad ?$$

A First Course in Logic

An Introduction to Model Theory, Proof Theory, Computability, and Complexity

Shawn Hedman

Oxford Texts in Logic Volume 1
430 pages - ISBN 0-19-852981-3

Dr. Shawn C. Hedman
Associate Professor

B.S., DePaul University
Ph.D., University of Illinois at Chicago

Office: Polk Science Bldg, #137
Phone: (863) 680-4332
E-mail: shedman@flsouthern.edu
Dr. Hedman's home page

Description: *A First Course in Logic* is a unified introduction to model theory, proof theory, computability, and descriptive complexity. It provides a foundation for further study in any one of these related areas of modern logic. Beginning with the basics, this book takes the reader through many of the landmark results in logic achieved during the past century (including the Löwenhiem-Skolem Theorems, Beth's Definability Theorem, Lindström's Theorem, Trakhtenbrot's Theorem, Fagin's Theorem, and Gödel's Incompleteness Theorems). In addition to classical topics such as quantifier elimination and Herbrand theory, this book contains topics in logic that have recently become prominent such as descriptive complexity and strongly minimal theories. The book stresses the connections between logic and theoretical computer science including such topics as recursive sets, the completeness of resolution, and the P = NP problem.

OXFORD
LOGIC

OXFORD TEXTS IN LOGIC 1

A First Course in Logic

An Introduction to Model Theory, Proof Theory, Computability, and Complexity

SHAWN HEDMAN

A First Course in Logic

An introduction to model theory, proof theory, computability, and complexity

SHAWN HEDMAN
Department of Mathematics, Florida Southern College

OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

Great Clarendon Street, Oxford OX2 6DP

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide in

Oxford New York

Auckland Cape Town Dar es Salaam Hong Kong Karachi
Kuala Lumpur Madrid Melbourne Mexico City Nairobi
New Delhi Shanghai Taipei Toronto

With offices in

Argentina Austria Brazil Chile Czech Republic France Greece
Guatemala Hungary Italy Japan Poland Portugal Singapore
South Korea Switzerland Thailand Turkey Ukraine Vietnam

Published in the United States
by Oxford University Press Inc., New York

First published 2004
Reprinted (with corrections) 2006

A catalogue record for this title is available from the British Library

Library of Congress Cataloging in Publication Data

Data available

Typeset by Newgen Imaging Systems (P) Ltd., Chennai, India

Printed in Great Britain
on acid-free paper by
Biddles Ltd., King's Lynn, Norfolk

ISBN 0–19–852980–5 (Hbk)
ISBN 0–19–852981–3 (Pbk)

10 9 8 7 6 5 4 3 2

A relation is primitive recursive if and only if it is definable by a Δ_0 formula. We presently prove one direction of this fact. The other direction shall become apparent after Section 8.3 of the next chapter and is left as Exercise 8.6.

Proposition 7.28 Let A be a definable subset of $\mathbf{N}_0$. If A is definable by a Δ_0 formula, then A is primitive recursive.

Proof We must add one step to the proof of the previous proposition. Suppose that $\varphi(x, y)$ defines a primitive recursive subset A of $\mathbf{N}_0$. We must show the formula $\exists y(y < x \wedge \varphi(x, y))$ also defines a primitive recursive subset. For convenience, we assume that x and y are the only free variables of $\varphi(x, y)$ (this assumption does not alter the essence of the proof).

Let $\chi_A(x, y)$ be the characteristic function for A. Since this function is primitive recursive, so is the function $sum\chi(x, y) = \sum_{z<y} \chi_A(x, z)$ by Proposition 7.18. It follows that the function $g(x) = sum\chi(x, x)$ is also primitive recursive. Note that $1 \dot{-} g(x)$ equals 0 if $\chi_A(x, z) = 1$ for some $z < x$ and otherwise $1 \dot{-} g(x)$ equals 1. From this observation, we see that the function $1 \dot{-} (1 \dot{-} g(x))$ is the characteristic function for the set defined by $\exists y(y < x \wedge \varphi(x, y))$. It follows that this is a primitive recursive set. □

Propositions 7.26 and 7.28 allow us to succinctly show that certain functions and relations are primitive recursive. The aim for the remainder of this section is twofold. One aim is to demonstrate some of the many familiar functions and relations that are primitive recursive. The other aim is to show that a specific binary function, namely $pf(x, i)$, is primitive recursive. The name "pf" bestowed to this function is an abbreviation for "prime factorization." We shall make use of this function and the fact that it is primitive recursive in Section 7.4.

Prior to defining the function $pf(x, i)$, we define the relations $div(x, y)$ and $prime(x)$. For any pair (x, y) of non-negative integers, the relation $div(x, y)$ says that x divides y and $prime(x)$ says that x is prime. The relation $div(x, y)$ holds if and only if there exists a z such that $x \cdot z = y$. Clearly, if such a z exists, then z is at most y. So $div(x, y)$ is definable by the Δ_0 formula

$$\exists z(z < y \wedge x \cdot z = y) \vee x = 1 \vee (y = 0 \wedge \neg x = 0).$$

Likewise, $prime(x)$ is defined by the formula

$$\forall z(z < x \rightarrow (z = 1 \vee \neg div(z, x))) \wedge (\neg x = 1).$$

Since these formulas are Δ_0, the relations $div(x, y)$ and $prime(x)$ are primitive recursive by Proposition 7.28.

There are infinitely many primes. Let $p_1, p_2, p_3, \ldots$ represent the enumeration of the primes in increasing order. So $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, and so forth. We claim that the function $pr(i) = p_i$ is primitive recursive. To make this function

(b) Show that f is a recursive function if and only if the graph of f is a recursively enumerable set.

(c) Show that f is a total recursive function if and only if the graph of f is a recursive set.

7.14 Let $f(x, y) = \lfloor x/y \rfloor$ where $\lfloor x/y \rfloor$ denotes the greatest integer less than x/y. Let $g(x)$ be a primitive recursive function and let

$$h(x, y) = \begin{cases} f(x, y) & y \neq 0 \\ g(x) & y = 0. \end{cases}$$

Show that $h(x, y)$ is primitive recursive. (Use the previous exercise and the fact that a relation is primitive recursive if and only if it is definable by a Δ_0 formula.)

7.15 Let $f(x)$, $g(x)$, and $h(x)$ be primitive recursive functions. Let

$$e(x) = \begin{cases} f(x)^{g(x)} & \text{if } f(x) + g(x) > 0 \\ h(x) & \text{if } f(x) + g(x) = 0. \end{cases}$$

Show that $e(x)$ is primitive recursive.

7.16 Let $\varphi(x, y)$ be a Δ_0 formula and let $f(x)$ be a primitive recursive function. Show that the formula $\exists y(y < f(x) \wedge \varphi(x, y))$ is Δ_0, where $y < f(x)$ is an abbreviation for the $\mathcal{V}_{ar}$-formula $\exists z(y + z = f(x))$.

7.17 Assuming that every primitive recursive set is Δ_0, prove the following.

(a) Every recursively enumerable set is $\sum_1$.

(b) Every recursive set is both $\sum_1$ and Π_1.

7.18 Let A be an infinite set. Prove that A is recursive if and only if it is the range of an increasing recursive function.

7.19 Show that every infinite recursively enumerable set has an infinite recursive subset.

7.20 Let A and B be recursively enumerable sets. Show that there exist recursively enumerable subsets $A_1 \subset A$ and $B_1 \subset B$ such that $A_1 \cap B_1 = \emptyset$ and $A_1 \cup B_1 = A \cup B$.

7.21 Show that the union of two recursively enumerable sets is recursive enumerable. Moreover, show that the function $f(x, y)$ defined by $W_x \cup W_y = W_{f(x,y)}$ is a recursive function.

7.22 Repeat the previous exercise with intersections instead of unions.

7.23 Show that there exists a partial recursive function that cannot be extended to a total recursive function.

Exercises

8.1. Explain why Gödel's Incompleteness theorems do not contradict the Completeness theorem (also proved by Kurt Gödel).

8.2. Encode the following finite sequences as a triple $[l, m, k]$ using the method described in Section 8.2: (a) $(1, 1, 1)$; (b) $(3, 3, 3, 3, 3)$; (c) $(1, 2, 3)$.

8.3. What finite sequence is coded by the triple $[4, 5, 373777]$?

8.4. The Fibonacci sequence is $1, 1, 2, 3, 5, 8, \ldots$ (each number in the sequence is the sum of the previous two.) A number is called a *Fibonacci number* if it is one of the numbers in this sequence. Write a $\mathcal{V}_N$-formula $\phi(x)$ such that $\mathbf{N} \models \phi(a)$ if and only if a is a Fibonacci number.

8.5. (a) Express the formula $1 + 2 + \cdots + x = \frac{x(x+1)}{2}$ as a $\mathcal{V}_N$-formula $\varphi(x)$.

(b) Show that $\Gamma_N \vdash \forall x \varphi(x)$ where Γ_N is the set of axioms from Section 8.1.

8.6. Prove that a definable subset D of $\mathbf{N}$ is definable by a Δ_0 $\mathcal{V}_N$-formula if and only if D is primitive recursive.

8.7. Show that the following sets of natural numbers are primitive recursive by describing a Δ_0 formula that defines the set:

(a) $\mathcal{T} = \{n | n$ is the Gödel code for a $\mathcal{V}_N$-term $\}$

(b) $\mathcal{F} = \{n | n$ is the Gödel code for a $\mathcal{V}_N$-formula $\}$

(c) $\mathcal{S} = \{n | n$ is the Gödel code for a $\mathcal{V}_N$-sentence $\}$.

8.8. Let T be recursive $\mathcal{V}_N$-theory containing Γ_N. Show that the set $\{n | \mathbf{N} \models Pr_T(n)\}$ is not primitive recursive.

8.9. Show that the decision problems corresponding to each of the four sets defined in the previous two problems are in $\mathbf{NP}$. If $\mathbf{P} \neq \mathbf{NP}$, then which these problems are in $\mathbf{P}$?

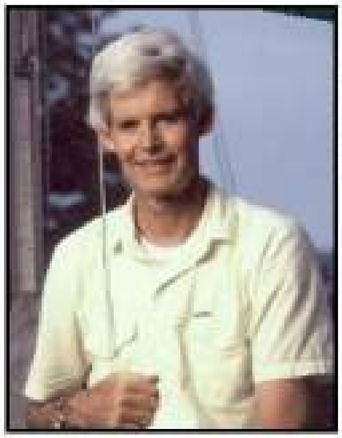
8.10. Consider the structure $\mathbf{R} = \{\mathbb{R} | +, \cdot, 0, 1\}$. The theory of $\mathbf{R}$ is decidable. For each $n \in \mathbb{N}$ the set $\{1, 2, 3, \ldots, n\}$ is a definable subset of $\mathbf{R}$. Let $\mathbf{R}_e$ be an expansion of $\mathbf{R}$ in which the natural numbers is a definable subset. Show that the theory of $\mathbf{R}_e$ is undecidable.

8.11. Let $\mathcal{V}$ be a finite vocabulary and let T be a $\mathcal{V}$-theory. Let $D = \{t_1, t_2, t_3, \ldots\}$ be a set of $\mathcal{V}$-terms. A subset B of D^2 is *recursive* if $B = \{(t_i, t_j) | (i, j) \in I\}$ for some recursive subset I of $\mathbb{N}^2$. Suppose that

- for some $M \models T$, each recursive subset of D^2 is a definable subset of M.
- for each $m \in \mathbb{N}$ there exists a term $t_n \in D$ such that n is more than m times the length of t_n. (i.e., there exist terms $t_n \in D$ that are arbitrarily short relative to n.) Prove that T is undecidable.

A Workshop in Honour of Stephen A. COOK
"Steve Cook at 60" -- April 28 - 29, 2000

Speakers	Schedule	Invited Participants Abstracts



Stephen Cook was born in Buffalo, New York. He received his B.Sc. degree from the University of Michigan in 1961 and his SM and PhD degrees from Harvard University in 1962 and 1966 respectively. From 1966 to 1970 he was an Assistant Professor at the University of California, Berkeley. He joined the University of Toronto in 1970 as an Associate Professor and was promoted to a Professor in 1975.

Dr. Cook's principal research area is computational complexity, with excursions into programming language semantics, parallel computation and especially the interaction between logic and complexity theory. He has authored over 50 research papers, including his famous 1971 paper, "The Complexity of Theorem Proving Procedures", which introduced the theory of NP completeness. Dr. Cook was the 1982 recipient of the Turing award, a Killam Research Fellowship in 1982, and a Steacie Fellowship in 1977. He received computer science teaching awards in 1989 and 1995. He is a fellow of the Royal Society of Canada and was elected to membership in the National Academy of Sciences (U.S.) and the American Academy of Arts and Sciences.

CSC 438F/2404F: Computability and Logic
Fall, 2008

Example: The relation $x|y$ is a Δ_0 relation.

Side Remark: All Δ_0 relations can be recognized in linear space on a Turing machine, when input numbers are represented in binary notation.

Lemma: The Δ_0 relations are closed under $\wedge, \vee, \neg$ and the bounded quantifiers $\forall \leq, \exists \leq$.

Proof: Notice that in this lemma, the operations in question are semantic operations, since they operate on relations (semantic objects). The boolean operations $\wedge, \vee, \neg$, for example, are discussed in the context of primitive recursive relations on page 62 of the notes, and the operations of bounded quantification are discussed on page 63 of the notes.

However each of these semantic operations on relations corresponds to a syntactic operation on formulas. For example, suppose that R and S are n-ary Δ_0 relations. Then by definition of Δ_0, there are bounded formulas A and B which represent R and S, respectively. Then the formula $(A \wedge B)$ is a bounded formula which represents the relation $R \wedge S$. Therefore $R \wedge S$ is a Δ_0 relation. A similar argument applies to each of the other operations mentioned in the lemma.

Lemma: Every Δ_0 relation is primitive recursive.

Proof: Structural Induction on bounded formulas in the vocabulary $\mathcal{L}_{A,\leq}$. We use the fact that the primitive recursive relations (i.e. predicates) are closed under the boolean operations and bounded quantification, as discussed on pages 62 and 63 in the notes. $\square$

Remark: The converse of the above lemma is false, as can be shown by a diagonal argument. For those familiar with complexity theory, we can clarify things as follows. As noted in the Side Remark above, all Δ_0 relations can be recognized in linear space on a Turing machine. On the other hand, it follows from the Ritchie-Cobham Theorem that all relations recognizable in space bounded by a primitive recursive function of the input length are primitive recursive. In particular, space $O(n^2)$ relations are primitive recursive, and a straightforward diagonal argument shows that there are relations recognizable in n^2 space which are not recognizable in linear space, and hence are not Δ_0 relations.

Definition: A $\exists\Delta_0$ formula (also called a Σ_1 formula) is one of the form $\exists y A$, where A is a Δ_0 formula.

Definition: R is a $\exists\Delta_0$-relation iff R is represented by a $\exists\Delta_0$ formula.

Notice that we are applying the same adjective "$\exists\Delta_0$" to both relations and formulas. Of course all $\exists\Delta_0$ relations are arithmetical.

Theorem: Every $\exists\Delta_0$ relation is r.e. (defined page 75)

Proof: Suppose that $R(\vec{x})$ is a $\exists\Delta_0$ relation. Then R is represented by a formula $\exists y A(\vec{x}, y)$, where $A(\vec{x}, y)$ is a bounded formula. Then A represents a Δ_0 relation $S(\vec{x}, y)$, such that $R(\vec{x}) = \exists y S(\vec{x}, y)$. By the previous lemma, S is primitive recursive, and hence recursive, and therefore R is r.e., by the definition of r.e. $\square$

Theoretical Computer Science 193 (1998) 129–148

Theoretical Computer Science

Rudimentary relations and primitive recursion: A toolbox

Henri-Alex Esbelin[a,*], Malika More[b]

[a] LLAIC1, IUT Clermont 1, BP 86, 63172 Aubière Cedex, France

[b] Département de Mathématiques, Université de Caen, 14032 Caen, France

Received October 1996; revised December 1996

Communicated by M. Nivat

Abstract

Rudimentary relations are those relations over natural integers that are defined by a first-order arithmetical formula, in which all quantifications are bounded by some variables. The question of whether a given primitive recursive relation is rudimentary is in some cases difficult and related to several well-known open questions in theoretical computer science. In this paper, we present systematic tools to study this question, and various applications. One of them gives a sufficient condition of the collapsing of the first classes of the Grzegorczyk's hierarchy.

Keywords: Rudimentary relations; counting; primitive recursion; coding

1. Introduction

Rudimentary relations are those relations over integers which are definable by a Δ_0 formula. They have been studied for a long time (see [1, 9, 20, 24]) and are still interesting because they form the following *robust*, *large* and *intriguing* class of relations.

Definition 1.1. Let us denote by $\mathfrak{R}$ the smallest class of relations over integers containing the graphs of addition and multiplication (seen as ternary relations) and closed under the following operations:

- boolean operations ($\neg, \wedge, \vee, \rightarrow$);
- explicit transformations, i.e. adding, cancelling, renaming, permuting and confusing variables. (see a precise definition in [24]);
- variable bounded quantifications (i.e. $\forall x < y \ldots$ meaning $\forall x\ (x < y \rightarrow \ldots)$ and $\exists x < y \ldots$ meaning $\exists x\ (x < y \wedge \ldots)$).

$\mathfrak{R}$ is *robust*: there are several different definitions of this class of relations. Rudimentary relations were first introduced by Smullyan in [24], following the ideas of Quine

* Corresponding author. E-mail: esbelin@llaic.univ-bpclermont.fr.

PII S0304-3975(97)00062-9

(cf. [20]) using dyadic concatenation $x = y * z$ as a basic relation for arithmetic instead of $x = y + z$ and $x = y.z$. In the same paper, Smullyan shows that any semi-recursive relation can be defined by a formula of type as $\exists w P(x, w)$, where P is a rudimentary relation, and that all rudimentary relations are definable by Δ_0-formulas. Then, Bennett shows in [1] that the class of rudimentary relations does not depend on what (fixed) alphabet is used to represent integers, as long as it contains at least two letters, and that all Δ_0-formulas define rudimentary relations.

In [9], Harrow proved that $\mathfrak{R}$ is closed under substitution of a polynomial to a variable. In particular, this means we can make use of *polynomially* bounded quantifications (such as $\forall y < x^2$) instead of bounded variable quantifications.

Also, $\mathfrak{R}$ corresponds to a computational complexity class, to a descriptive complexity class and to a weak recursion class as we recall below.

$\mathfrak{R}$ is *large*: most natural arithmetical relations are rudimentary. For instance, the following formula defines the set of prime numbers:

$$x > 1 \wedge \forall y < x \forall z < x \neg(x = y.z).$$

In Section 2, we list some other relations that are proved to be rudimentary. In some cases, we use the fact that the ternary relation $x = y^z$ is rudimentary, as Bennett has proved in [1].

Mainly there are two types of relations whose status toward rudimentary relations is unknown. The first type corresponds to graphs of recursive primitive functions, in particular to counting relations. For instance, *is the binary relation "y is the xth prime number" rudimentary?* Indeed, the question of whether $\mathfrak{R}$ is closed under counting is still open (see [13, 19] and Section 7). In this paper, we are concerned in this first type of relations.

The second type corresponds to relations obtained from rudimentary relations by substituting an exponential to a variable. For instance, *is the unary relation "x verifies $2^x + 1$ is prime" rudimentary?* Note that it is known that $\mathfrak{R}$ is not closed under substitution of an exponential to a variable (see [9]), so that the answer is "no" in general.

However, it is difficult to exhibit a *natural* arithmetical relation which can be proved not to be rudimentary.

The origin of this paper is a previous proof of the fact that the sequence of Fibonacci's numbers is rudimentary (see [6, 16] and Section 6). With this aim in view, we had to use various coding devices which are presented in this paper. This paper is an attempt to systematize the use of these tools for proving that various primitive recursive relations are rudimentary (or counting rudimentary, see Section 7).

$\mathfrak{R}$ is *intriguing*: rudimentary relations are linked with a lot of well-known open questions in computational complexity, finite model theory, weak arithmetics and recursivity theory. Let us denote by $R(\mathfrak{R})$ the class of rudimentary sets (i.e. unary rudimentary relations) and by $Ev(R(\mathfrak{R}))$ the class of their dyadic representations ("rudimentary languages"). Wrathall proved that reasonable encoding of k-ary way of exhibiting a primitive recursive relation which is not rudimentary is to choose it in $\mathfrak{E}^2_* \setminus \mathfrak{E}^0_*$. Although it is true that infinitly many relations exist, we know no natural example.

The first author attempts to characterize $\mathfrak{R}$ as a recursion class in [5]. Let $\mathfrak{I}^{-1}$ be the smallest class of functions containing constants, projections and predecessor and closed under composition and bounded iteration ($f(x,0)=g(x)$, $f(x,i+1)=h(x,f(x,i))$, $f(x,i)\leqslant k(x,i)$). The corresponding class of relations $\mathfrak{I}^{-1}_*$ contains $\mathfrak{R}$ whereas any class obtained by cancelling some basic operations or by replacing bounded iteration by bounded pure iteration is strictly contained in $\mathfrak{R}$. But, once again, *the question of equality remains open.*

Hence, given a primitive recursive relation, the question of its belonging to $\mathfrak{R}$ is nontrivial, and methodic tools for studying this type of questions are worth being developed.

In Section 2, we write down several easy rudimentary definitions that are used in the sequel. In Section 3, we present the classical encoding of calculus for a primitive recursive function, and study in which case this tool is strong enough to prove that the graph of a given (primitive recursive) function is rudimentary. This method was used by Paris and Wilkie in [19], and by Woods in [26]; his result corresponds to our Lemma 3.6, and proves that some "short and small" recursively defined sequences of integers are rudimentary. He extended this lemma to larger and longer sequences, but only when they are defined by *summation*. We generalize it to larger and longer and *recursively defined* sequences. Then we can obtain results, (some of these have already been proved by Meloul in [15] via LOGSPACE machines), in a straightforward arithmetical way. Then we prove that linear and polynomial recurrences are rudimentary in Section 6. Finally, in Section 7, we prove that the set of the counting rudimentary relations is closed under polynomial substitution, and to obtain a sufficient condition of the collapsing of the Grzegorczyk's hierarchy. Sections 6 and 7 can be read independently.

2. Easy rudimentary relations

It will be convenient for the last section to generalize the results to classes which contains the rudimentary relations. So let us introduce the following definition:

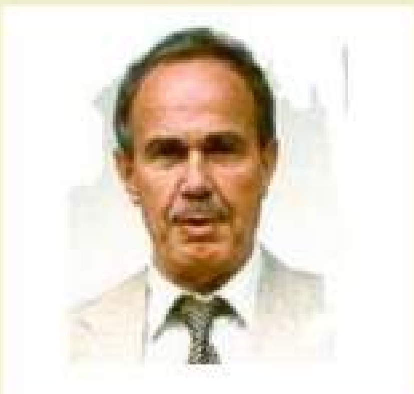
Definition 2.1. Let us denote by $\mathfrak{R}^{+}$ any class of primitive recursive relations over integers containing the graph of addition and multiplication and closed under the following operations:

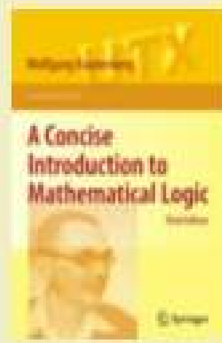
- boolean operations;
- explicit transformations;
- variables bounded quantifications.

It clearly appears that the class $\mathfrak{R}$ is the smallest among the $\mathfrak{R}^{+}$ classes.

A Concise Introduction to Mathematical Logic

Third Edition



You may read more about the textbook 

at the link Springer.

Wolfgang Rautenberg

UNIVERSITEXT

A Concise Introduction to Mathematical Logic

Third Edition

Springer

Wolfgang Rautenberg

A Concise Introduction to Mathematical Logic

Third Edition

Springer

Prof. Dr. Wolfgang Rautenberg
Fachbereich Mathematik und Informatik
14195 Berlin
Germany
raut@math.fu-berlin.de

Cover illustration: Photographer unknown. Courtesy of The Shelby White and Leon Levy Archives Center, Institute for Advanced Study, Princeton, NJ, USA.

ISBN 978-1-4419-1220-6 e-ISBN 978-1-4419-1221-3
DOI 10.1007/978-1-4419-1221-3
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2009932782

Mathematics Subject Classification (2000): 03-XX, 68N 17

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

formulas. φ is called Δ_0, Σ_1, or Π_1 also if it is equivalent to an original Δ_0-, Σ_1-, or Π_1-formula, respectively. In this sense, if α is Δ_0 then so too are $(\exists x{\leqslant}t)\,\alpha\ \big(\equiv \neg(\forall x{\leqslant}t)\neg\alpha\big)$ and $(\forall x{<}t)\alpha\ \big(\equiv (\forall x{\leqslant}t)(x = t \vee \alpha)\big)$.

Clearly, Π_1 consists of the complements of the $P \in \Sigma_1$. The $P \in \Delta_1$ are both Σ_1- and Π_1-definable, with possibly distinct formulas. By Exercise 3 in **2.4**, Σ_1 and Π_1 are closed under union and intersection of predicates of the same arity, and Δ_1 like Δ_0 moreover under complements. If $P \in \mathbb{N}^m$ and $g_1, \ldots, g_m \in \mathbf{F}_n$ are Σ_1, so too is $Q = P[g_1, \ldots, g_m]$, simply because $Q\vec{a} \Leftrightarrow \exists\vec{y}(\bigwedge_{i=1}^{n} y_i{=}g_i\vec{a}\ \ \&\ P\vec{y})$. Note also that if graph f is Σ_1 then it is automatically Δ_1, for $f\vec{a}{\neq}b \Leftrightarrow \exists y(f\vec{a}{=}y\ \&\ y{\neq}b)$, so that the complement of graph f is again Σ_1. Here some examples of Δ_0- and Σ_1-formulas and sentences. Interesting Π_1-sentences are found at the end of **6.5**.

Examples. Diophantine equations are the simplest Δ_0-formulas. To these belong the formulas $y = t(\vec{x})$ with $y \notin \mathit{var}\,t$, which define the term functions $\vec{a} \mapsto t^{\mathcal{N}}(\vec{a})$. Since $a \mid b \Leftrightarrow (\exists c{\leqslant}b)(a \cdot c = b)$, divisibility and thus also the predicate prim are Δ_0. Because $\wp(a,b) = c \Leftrightarrow 2c = (a+b)^2 + 3a + b$, graph$_\wp$ is Δ_0. The same holds for the relation of being *coprime*, denoted by $\perp$ and defined by $a \perp b :\Leftrightarrow (\forall c \leqslant a+b)(c \mid a, b \Rightarrow c = 1)$. Diophantine predicates are trivially Σ_1. Surprisingly, by Theorem 5.6 the converse holds as well, although it had originally been conjectured that, for instance, the set $\{a \in \mathbb{N} \mid (\forall p{\leqslant}a)(\mathsf{prim}\,p\ \&\ p \mid a \Rightarrow p = 2)\}$ of all powers of 2 was not Diophantine. This set is Δ_0. Even the graph of $n \mapsto 2^n$ is Δ_0.

Remark 1. More generally, the predicate '$a^b = c$' is Δ_0, though it is difficult to prove this fact. Indeed, even the proof in **6.4** that this predicate is arithmetical requires effort. Earlier results from Bennet, Paris, Pudlak, among others, are generalized in [BA] as follows: if $f \in \mathbf{F}_{n+1}$ (more precisely, graph f) is Δ_0 then so is $g\colon(\vec{a}, n) \mapsto \prod_{i \leqslant n} f(\vec{a}, i)$, and the recursion equation $g(\vec{x}, \mathsf{S}y) = g(\vec{x}, y) \cdot f(\vec{x}, y)$ is provable in $I\Delta_0$. This theory is an important weakening of PA. It results from Q by adjoining the induction schema restricted to Δ_0-formulas. $I\Delta_0$ plays a role in various questions, e.g., in complexity theory ([Kra] or [HP]). Induction on the Δ_0-formulas readily shows that all Δ_0-predicates are p.r. The converse does not hold; an example is the graph of the very rapidly growing *hyperexponentiation*, recursively defined by $\mathrm{hex}(a, 0) = 1$ and $\mathrm{hex}(a, Sb) = a^{\mathrm{hex}(a,b)}$. Stated more suggestively, $\mathrm{hex}(a, n) = \underbrace{a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}}_{n}$.

According to Theorem 3.1 below, already the weak theory Q is Σ_1-complete, i.e., each Σ_1-sentence true in $\mathcal{N}$ is provable in Q. This can be

P.R. & Δ_0-Definable Functions

- Kurt Gödel: defining $y = 2^x$ in the language $\langle 0, S, +, \times, \leqslant \rangle$
- a Δ_0-definition for $y = 2^x$ ... `is possible` ...

  through `efficient coding`:

  code of $\langle a_1, \ldots, a_k \rangle \leqslant$ polynomial of $(\prod_{j=1}^{k} a_j)$

- for $y = 2^{2^x}$: code of $\langle 2, 2^2, 2^{2^2}, \ldots, 2^{2^i}, \ldots, 2^{2^x} \rangle \leqslant$ polynomial of $\prod_{j=0}^{x} 2^{2^j} \leqslant P(2^{\sum_{j=0}^{x} 2^j}) \leqslant P(2^{2^{x+1}}) = P\big((2^{2^x})^2\big) = Q(2^{2^x})$.
- So, $y = 2^{2^x} \iff$
  $\exists s \leqslant Q(y) \big[\mathrm{seq}(s) \wedge (s)_0 = 2 \wedge \forall i < x [(s)_{i+1} = (s)_i \cdot (s)_i] \wedge (s)_x = y\big]$.

- For $y = h-exp(a, x)$ if we have $\mathfrak{exp}_{\Delta_0}(a, x, y)$ for $y = a^x$ then $\exists s \leqslant y^2 \big[\mathrm{seq}(s) \wedge (s)_0 = 1 \wedge \forall i < x [\mathfrak{exp}_{\Delta_0}(a, (s)_i, (s)_{i+1})] \wedge (s)_x = y\big]$.

Primitive Recursiveness vs. Definability by Bounded Formulas Isfahan Seminar on Math. Logic 2011

P.R. & Δ_0-Definable Functions (continued)

- Proskurin, A.V.; "Positive Rudimentarity of the Graphs of Ackermann's and Grzegorczyk's Functions", Zap. Nauch. Semin. Leningr. Otd. Mat. Inst. Steklova 88, 186–191 (1979). Russian!
- Pudlák, Pavel, "A Definition of Exponentiation by a Bounded Arithmetical Formula", Commentat. Math. Univ. Carol. 24, 667–671 (1983).
- Calude, Cristian; "Super-Exponentials Non-Primitive Recursive, but Rudimentary", Inf. Process. Lett. 25, 311–315 (1987).

Non-P.R. but Δ_0-Definable Functions

- Kleene's Normal Form Theorem:

– A Primitive Recursive (indeed Δ_0-definable) Predicate $\mathtt{T}(e,x,z)$: the program with code e has (a halting) configuration z on input x

– A Primitive Recursive Function $\mathtt{U}(z)$

such that for every *Recursive* function f there exists some e:

$$f(\overline{x}) = \mathtt{U}\big(\mu z.[\mathtt{T}(e,\overline{x},z)]\big).$$

- For non-P.R. total e:

the function $x \mapsto \mu z.\mathtt{T}(e,x,z)$ is Δ_0-definable but non-P.R.!

P.R. Functions and Relations vs. Δ_0-Definability ?

$$R \in \Delta_0 \Longrightarrow R \in \text{P.R.} \qquad \checkmark$$

$$(R \in \text{P.R.} \Longrightarrow R \in \Delta_0) \qquad ?$$

$$\Updownarrow \quad \Updownarrow$$

$$(f \in \text{P.R.} \Longrightarrow f \in \Delta_0) \qquad ?$$

$$f \in \Delta_0 \Longrightarrow f \in \text{P.R.} \qquad \mathcal{X}$$

$$\Gamma_f \in \text{P.R.} \Longrightarrow f \in \text{P.R.} \qquad \mathcal{X}$$

Primitive Recursiveness vs. Definability by Bounded Formulas Isfahan Seminar on Math. Logic 2011

P.R. $\Longrightarrow$ Δ_0-Definability ?

- A Non-Δ_0-Definable Property? Truth of Δ_0-Formulae!

▷ There can exist no Δ_0-formula $\mathrm{Sat}_{\Delta_0}(x,y)$ such that $\mathbb{N}\models\mathrm{Sat}_{\Delta_0}(x,y) \iff x$ is the Gödel code of a Δ_0-formula with one free variable $\theta(\xi)$ which is satisfied by y (i.e., $\mathbb{N}\models\theta(y)$).

- Otherwise, let $\theta(x)=\neg\mathrm{Sat}_{\Delta_0}(x,x)$ and $m=\ulcorner\theta(x)\urcorner$.
  Then $\mathrm{Sat}_{\Delta_0}(m,m) \iff \theta(m) \iff \neg\mathrm{Sat}_{\Delta_0}(m,m)$.

[Tarski's Theorem on Non-Definability of Truth]

Δ_0-Satisfiability is P.R.!

- Lessan, H. (Hamid), *Models of Arithmetic*, Ph.D. Thesis, University of Manchester (1978).
- Hájek, Petr & Pudlák, Pavel, *Metamathematics of First-Order Arithmetic*, Springer (1998), $2^{\rm nd}$ printing.

► There exists a Δ_0-formula $\Gamma_{\Delta_0}(x,y,z)$ such that the bounded formula with code x is satisfied by y $\iff$ for some $z \geqslant 2^{2^{2^{(x+y+c)}}}$ (c is a constant) $\Gamma_{\Delta_0}(x,y,z)$ holds.

▷ So, $\exists z \leqslant 2_4^{x+y+c} \Gamma_{\Delta_0}(x,y,z)$ is a non-Δ_0, P.R. predicate !

Δ_0-Satisfiability is P.R. – EASIER!

- The Properties of

being a term:	P.R. (also Δ_0)
being a bounded formula:	P.R. (also Δ_0)
the value of a term:	P.R. (not Δ_0 ?)
satisfaction of a bounded formula:	P.R. (not Δ_0 !)

- For a Δ_0-formula

$\theta(\overline{y}) = Q_1 x_1 \leqslant t_1(\overline{y}) \ldots Q_n x_n \leqslant t_n(\overline{y}, x_1, \ldots)\, \varrho(\overline{y}, \overline{x})$, $(Q_i \in \{\forall, \exists\})$
there exists a polynomial $\mathtt{P}_{\varphi}$ (P.R. function) such that
$\forall \overline{a}: \quad \mathbb{N} \models \theta(\overline{a}) \iff \{0, 1, \cdots, \mathtt{P}_{\varphi}(\overline{a})\} \models \theta(\overline{a})$.

  - ►► Thus, $\{(\ulcorner\theta\urcorner, \overline{a}) \mid \theta \in \Delta_0 \ \& \ \mathbb{N} \models \theta(\overline{a})\} \in \mathrm{P.R.} - \Delta_0$!

Finally ...

$$R \in \Delta_0 \Longrightarrow R \in \text{P.R.} \qquad \checkmark$$

$$(R \in \text{P.R.} \Longrightarrow R \in \Delta_0) \qquad \mathcal{X}$$

$$\Updownarrow \quad \Updownarrow$$

$$(f \in \text{P.R.} \Longrightarrow f \in \Delta_0) \qquad \mathcal{X}$$

$$f \in \Delta_0 \Longrightarrow f \in \text{P.R.} \qquad \mathcal{X}$$

$$\Gamma_f \in \text{P.R.} \Longrightarrow f \in \text{P.R.} \qquad \mathcal{X}$$

Finally ...

$$R \in \Delta_0 \Longrightarrow R \in \text{P.R.}$$

$$R \in \text{P.R.} \not\Longrightarrow R \in \Delta_0$$

$$f \in \text{P.R.} \not\Longrightarrow f \in \Delta_0$$

$$f \in \Delta_0 \not\Longrightarrow f \in \text{P.R.}$$

$$\Gamma_f \in \text{P.R.} \not\Longrightarrow f \in \text{P.R.}$$

Finally ...

- Shawn Hedman's assertion in "A First Course in Logic" that every P.R. predicate is Δ_0, is wrong!
- Wolfgang Rautenberg's counterexample for a non$-\Delta_0$ but P.R. functions in "A Concise Introduction to Mathematical Logic" is wrong!
- Stephen Cook's too advanced argument for existence of a non$-\Delta_0$ but P.R. predicate can be considerably simplified!
- I believe that "satisfaction" is a natural predicate for being non$-\Delta_0$ and P.R., contrary to Henri-Alex Esbelin & Malika More's claim in "Rudimentary Relations and Primitive Recursion: A Toolbox"!

Thank You!

Thanks to

The Participants ....For Listening...

and

The Organizers ....For Taking Care of Everything...

SaeedSalehi.ir