Concerning definable sets

by

N. A. Routledge (Cambridge)

1. We shall take the notions of *function of n variables (over the positive integers) taking only the values of* 0 *and* 1, *predicate with n arguments defined over the positive integers* and *set in R_n, Euclidean space of n dimensions, each coordinate being a positive integer* to be entirely equivalent. (The function will be 0 for a set of arguments $(r_1,\ldots,r_n)$ when the predicate holds, or *is true*, for $(r_1,\ldots,r_n)$, and when $(r_1,\ldots,r_n)$ belongs to the set, and so on). Following Mostowski [1] we define certain classes of sets: $P_n^{(k)}$ is the class of all sets expressible as $(Ex_1)(x_2)(Ex_3)\ldots$ $\ldots T(x_1,\ldots,x_n,y_1,\ldots,y_k)$ with n quantifiers (alternately existential and universal), where $T(x_1,\ldots,y_k)$ is a g. r. (general recursive) predicate, and $Q_n^{(k)}$ is the class of all sets expressible as $(x_1)(Ex_2)(x_3)\ldots T(x_1,\ldots,x_n,y_1,\ldots,y_k)$ under the same conditions. Provided $n>0$, then we lose no generality by restricting $T(x_1,\ldots,y_k)$ to be p. r. (primitive recursive) (see Kleene [2], Theorem V, Corollary). His theorem V can be enunciated as $P_1^{(k)}\cdot Q_1^{(k)}=P_0^{(k)}(=Q_0^{(k)})$.

We enquire: *What is $P_n^{(k)}\cdot Q_n^{(k)}$, for $n>1$?*

2. From Turing [3] we get the idea of an *oracle machine*. We are given a certain set S of integers, about which our knowledge is limited. But there is an oracle which knows all about S. We then set up a machine of the usual Turing type, save that from time to time the machine produces an integer and enquires of the oracle whether this integer belongs to S. The machine then moves on in a manner determined by the reply of the oracle. If a certain function can be calculated by such a machine we shall say that the function is *Turing derivable from, or reducible to*, S. This is exactly the same as for the function to be definable by a general recursive system of equations into which S (or rather its associated function) enters as an already known function. We can now answer the question ending **1**:

$P_n^{(k)}\cdot Q_n^{(k)}$ is exactly those sets in R_k which are Turing derivable from some set in $P_{n-1}^{(1)}$, for $n>0$.

To prove this is the aim of this paper [1]).

[1]) I am informed that A. Janiczak has already obtained this result, in the case $n=2$, in a paper shortly to appear in the Colloquium Mathematicum.

3. Let us assume that a certain predicate $H(n)$ is Turing derivable from a certain predicate $S(n)$. Replace $S(n)$ by an *undetermined* predicate $T(n)$ and set the machine working. After some steps the machine will produce a number r_1, and enquire: *Does $T(r_1)$ hold?* Assign the value of $T(r_1)$ *arbitrarily*, and the machine will move on and eventually perhaps produce another number r_2 and ask whether $T(r_2)$ holds. If $r_2\neq r_1$ (otherwise $T(r_2)$ is already determined) assign the value of $T(r_2)$ arbitrarily and let the machine move on. And so on. Eventually the machine may reach a stage when it claims to have calculated $H(n_0)$, for some n_0 (it would of course do this for every n_0 if the values assigned to $T(r_1),T(r_2),\ldots$ were those of $S(r_1),S(r_2),\ldots$). I say that clearly it is possible to obtain an effective (and even primitive recursive) enumeration of all possible stages in the motion of the machine, for any particular assignment of values to T leading to that stage, but I will discuss this in **5**.

4. We require the notion of the *general multiple* of a set of integers R (we shall denote it by $(G.M.R)$). This contains no numbers which have prime factors of power higher than 2, and $P_{i_1}^{\pi_1}\ldots P_{i_n}^{\pi_n}$ belongs to it if and only if $i_j\in R$ if $\pi_j=1$ and $i_j\notin R$ if $\pi_j=2$ (for $1\leqslant j\leqslant n$).

$(P_0,P_1,P_2,\ldots$ are the primes in ascending order).

Also 1 belongs and 0 does not.

Thus to settle a question:

Is it true that $\alpha_1,\ldots,\alpha_r\in R$ but $\beta_1,\ldots,\beta_s\notin R$? we need merely enquire: *Does*

$$P_{\alpha_1}\ldots P_{\alpha_r}(P_{\beta_1}\ldots P_{\beta_s})^2\in G.M.(R)?$$

(We shall say that an integer is the *correct form* if and only if it is non-zero and has no prime factor of power higher than 2).

5. Let us define functions $\Delta(n)$, $\Theta(n)$, $J(u,v)$, $K(n)$ and $L(n)$ as follows:

$$\Delta(n)=\tfrac{1}{2}n(n+1),$$

$$\Theta(n)=\mu x(x\leqslant n+1\ \&\ \Delta(n)>n)-1,$$

(*i. e.* $\Theta(n)$ is the greatest number x such that $\Delta(x)\leqslant n$),

$$J(u,v)=\Delta(u+v)+u,$$

$$K(n)=n-\Delta(\Theta(n)),$$

$$L(n)=\Theta(n)-K(n).$$

The functions J,K,L furnish a 1-1 correspondence between the ordered pairs of integers and the integers, in the order

$$(0,0),(0,1),(1,0),(0,2),(1,1),(2,0),(0,3)\ldots$$

That is, (u,v) is in the $J(u,v)$th place (counting from 0), and the pair in the nth place is $(K(n), L(n))$.

J, K, and L are clearly p. r. (see Péter [4] § 1).

We now return to our machine, and define two functions $\mathrm{Step}(n)$ and $\mathrm{Ans}(n)$.

If $L(n)$ is of the form $P_{a_1}\dots P_{a_r}(P_{\beta_1}\dots P_{\beta_s})^2$, with $a_1,\dots,\beta_s$ all different and $r, s\geqslant 0$, and there is a possible assignment of values to $T(x)$, and a possible notion of the machine such that in the course of the first $K(n)$ steps the machine appeals for the values of $T(a_1),\dots,T(\beta_s)$ and is told that $T(a_1),\dots,T(a_r)$ hold but $T(\beta_1),\dots,T(\beta_s)$ do not, let $\mathrm{Step}(n)=K(n)$ and $\mathrm{Ans}(n)=L(n)$. In all other cases let $\mathrm{Step}(n)=0$ and $\mathrm{Ans}(n)=1$. That these two functions are p. r. may be demonstrated by an arithmetisation, or *Gödelisation*, of the structure of the machine, in the manner of Kleene [5]. The proof is *obvious* but highly tedious and complex.

These functions provide the enumeration required at the end of **3** (with many repetitions of the case when the machine has made no moves, and no values of T have been assigned).

We can also *obviously* define predicates $\mathrm{Fin}(n,x)$ and $\mathrm{Eval}(n)$ (both p. r.), where $\mathrm{Fin}(n,x)$ means that after $\mathrm{Step}(n)$ moves, if we tell it that $\mathrm{Ans}(n)\in G.M.(T)$ (*i. e.* that $T(a_1)$ holds, etc.) the machine finishes the calculation of the (supposed) value of $H(x)$, and, this being the case, $\mathrm{Eval}(n)$ is the value of $H(x)$. (In other cases we are not interested in the value of $\mathrm{Eval}(n)$).

6. We know that, for any x, there is a certain assignment of values to T (viz. putting $T(y)=S(y)$) which makes the machine complete the calculation of $H(x)$ *correctly*, after a finite number of steps (and after only a finite number of values of T have been assigned) *i. e.*

$$(x)(En)\big(\mathrm{Fin}(n,x)\ \&\ \mathrm{Ans}(n)\in G.M.(S)\big).$$

There is for each x clearly only one such n, and, for this, $\mathrm{Eval}(n)\equiv H(x)$. So

LEMMA 1. $H(x)\equiv(En)\big(\mathrm{Fin}(n,x)\ \&\ \mathrm{Eval}(n)\ \&\ \mathrm{Ans}(n)\in G.M.(S)\big)$.

7. LEMMA 2. *If $S(n)$ is of the form $(Ex)R(n,x)$ then $G.M.(S)$ is of the form $(Ex)(y)M(n,x,y)$ where $M(n,x,y)$ is Turing derivable from $R(n,x)$, being derived from R and p. r. fns. by substitution.*

(Derivability was defined only for predicates with *one* variable, but this restriction was for convenience of exposition and is not essential at all).

For, $n\in G.M.(S).\equiv.n$ is of the *correct form* (see **4**)

$$\&\ (a)\big(P_a \text{ divides } n \text{ but } P_a^2 \text{ does not}.\to(Ex)R(a,x)\big)$$
$$\&\ (\beta)\big(P_\beta^2 \text{ divides } n\to(x)\sim R(\beta,x)\big).$$

Now let $n=P_{a_1}\dots P_{a_r}(P_{\beta_1}\dots P_{\beta_s})^2$ and if $x_1\dots x_r$ are the (hypothetical) x_is such that $R(x_i,a_i)$ holds for $1\leqslant i\leqslant r$ and if we put

$$y=P_{a_1}^{x_1}\cdot P_{a_2}^{x_2}\dots P_{a_r}^{x_r}$$

we see that $n\in G.M.(S).\equiv.n$ is of the correct form

$\&(Ey)(a)$ (P_a divides n but P_a^2 does not. $\to R(a,z)$, where z is the power of P_a in y),

$\&(\beta)(x)\big(P_\beta^2$ divides $n\to\sim R(\beta,x)\big)$.

Thus $n\in G.M.(S).\equiv.(Ey)(a)(\beta)(x)\,N(n,y,a,\beta,x)$,

where N is a predicate obtained from p. r. fns. and $R(n,x)$ simply by substitution.

Now the functions $J(J(p,q),r)$, and $K(K(n))$, $L(K(n))$, $L(n)$ furnish a 1-1 correspondence between the ordered triplets and the integers.

Thus $n\in G.M.(S).\equiv.(Ey)(z)N\big(n,y,K(K(z)),\ L(K(z)),\ L(z)\big)$ as was required (since p. r. fns. and substitution are all part of Turing derivability).

THEOREM 1. *If $G(n)$ is a predicate, Turing derivable from $(Ex)R(n,x)$, then*

(i) $$G(n).\equiv.(Ex)(y)M_1(n,x,y)$$

(ii) $$G(n).\equiv.(x)(Ey)M_2(n,x,y)$$

where M_1 and M_2 are obtained from p. r. fns. and $R(n,x)$ by substitution, and so, certainly, are Turing derivable from $R(n,x)$.

(i) follows immediately from lemmas 1 and 2, and the general result

$$(Ea)(E\beta)X(a,\beta,\dots)\equiv(E\gamma)X\big(K(\gamma),L(\gamma),\dots\big)$$

for any predicate $X(a,\beta,\dots)$.

For (ii), note that $\sim G(n)$ is also Turing derivable from $(Ex)R(n,x)$ and so, by (i),

$$\sim G(n).\equiv.(Ex)(y)\,M_1'(n,x,y)$$

and thus, at once, (ii).

This theorem clearly holds true when n stands for not one variable but many.

COROLLARY. *If $G(n)$ is Turing reducible to a $P_1^{(1)}$ set* (see **1**) *then $G(n)\in P_2^{(1)}\cdot Q_2^{(1)}$.*

8. THEOREM 2. *For $n>0$, $P_n^{(k)}\cdot Q_n^{(k)}=$ the class of sets in R_k Turing derivable from sets in $P_{n-1}^{(1)}$, $=R_k\cdot\mathfrak{D}(P_{n-1}^{(1)})$, say* (where $\mathfrak{D}(X)$ means the class of all sets Turing derivable from sets in the class of sets X).

First, two lemmas:

LEMMA 3.

$$P_n^{(k)} \cdot Q_n^{(k)} \subseteq \mathfrak{D}(P_{n-1}^{(1)}), \quad \textit{if} \quad n > 0.$$

For if

$$H(y_1, \ldots, y_k) \,.\!\equiv.\, (Ex_1)(x_2) \ldots A(x_1, \ldots, x_n, y_1, \ldots, y_k)$$
$$.\!\equiv.\, (x_1)(Ex_2) \ldots B(x_1, \ldots, y_k),$$

where A and B are g. r., then for all $y_1, \ldots, y_k$ either there is an x_1 such that (I) $(x_2)(Ex_3) \ldots A(x_1, \ldots, x_n, y_1, \ldots, y_k)$, or there is an x_1 such that $(x_2)(Ex_3) \ldots \sim B(x_1, \ldots, y_k)$ but never both.

Let

$$C(x_1, \ldots, x_n, y_1, \ldots, y^k) \equiv \begin{cases} A(x_1, \ldots, x_n, \tfrac{1}{2}y_1, y_2, \ldots, y_k) & \text{if } y_1 \text{ be even}, \\ \sim B(x_1, \ldots, x_n, \tfrac{1}{2}(y_1 - 1), y_2, \ldots, y_k) & \text{if } y_1 \text{ be odd}. \end{cases}$$

Let

$$D(x_1, y_1, \ldots, y_k) \equiv (x_2)(Ex_3) \ldots C(x_1, \ldots, x_n, y_1, \ldots, y_k).$$

Then

$$D(x_1, \ldots, y_k) \in Q_{n-1}^{(k+1)},$$

and clearly

$$\mu x_1\big(D(x_1, 2y_1, y_2, \ldots, y_k) \,.\vee.\, D(x_1, 2y_1 + 1, y_2, \ldots, y_k)\big)$$

exists, by (I), and is Turing derivable from D.

Call it $f(y_1, \ldots, y_k)$, then $H(y_1, \ldots, y_k) \equiv. D(f(y_1, \ldots, y_k), 2y_1, y_2, \ldots, y_k)$ clearly.

We have now merely to show that D is Turing derivable from a $P_{n-1}^{(1)}$ set. Now clearly a $Q_{n-1}^{(k+1)}$ set is derivable from a $Q_{n-1}^{(1)}$ set by using the enumeration of ordered $(k+1)$-tuples which can be got from J, K and L, and any $Q_{n-1}^{(1)}$ set is clearly Turing derivable from the $P_{n-1}^{(1)}$ set which is its complement. Hence the lemma.

LEMMA 4.

$$R_k \cdot \mathfrak{D}(P_{n-1}^{(1)}) \subseteq P_n^{(k)} \cdot Q_n^{(k)}, \quad \textit{if} \quad n > 0.$$

We prove this by induction on n. It is true for $n=1$, obviously, and for $n=2$ by Theorem 1.

If $n \geqslant 2$, assume the lemma to be true for n. Then if $G(n_1, \ldots, n_k)$ is derivable from some $(Ex)R(x, m)$, where $R \in Q_{n-1}^{(2)}$, by Theorem 1 (extended for many variables)

$$G(n_1, \ldots, n_k) \equiv. (Ex)(y)\, M_1(n_1, \ldots, n_k, x, y)$$
$$\equiv. (x)(Ey)\, M_2(n_1, \ldots, y),$$

where M_1 and M_2 are derivable from R.

Now, as in Lemma 3, a $Q_{n-1}^{(2)}$ set is derivable from a $P_{n-1}^{(1)}$ set, and so M_1 and M_2 are derivable from a $P_{n-1}^{(1)}$ set, and so, by hypothesis, $M_1 \in Q_n^{(k+2)}$ and $M_2 \in P_n^{(k+2)}$. That is, there are g. r. predicates $M_1'(z_1, \ldots, z_n, n_1, \ldots, n_k, x, y)$ and $M_2'(z_1, \ldots, z_n, n_1, \ldots, n_k, x, y)$ such that

$$G(n_1, \ldots, n_k) \equiv (Ex)(y)(z_1)(Ez_2)(z_3) \ldots M_1'(z_1, \ldots, y)$$
$$\equiv (x)(Ey)(Ez_1)(z_2)(Ez_3) \ldots M_2'(z_1, \ldots, y)$$
$$\equiv (Ex)(t)(Ez_2)(z_3) \ldots M_1'\big(K(t), z_2, \ldots, x, L(t)\big)$$
$$\equiv (x)(Et)(z_2)(Ez_3) \ldots M_2'\big(K(t), z_2, \ldots, x, L(t)\big).$$

Thus $G \in P_{n+1}^{(k)} \cdot Q_{n+1}^{(k)}$.

Hence the lemma, and hence Theorem 2, since

$$P_n^{(k)} \cdot Q_n^{(k)} \quad \text{certainly} \quad \subseteq R_k.$$

Added in proof. This paper was written in August 1952, and since then a result, essentially the same as my Theorem 2, has appeared in S. C. Kleene's *Introduction to Metamathematics*, (North Holland Publishing Co.), (see Theorem XI, p. 293).

References

[1] A. Mostowski, *On definable sets of positive integers*, Fundamenta Mathematicae 34 (1946), p. 81-112.

[2] S. C. Kleene, *Recursive predicates and quantifiers*, Transactions of the American Mathematical Society 53 (1943), p. 41-73.

[3] A. M. Turing, *Systems of logic based on ordinals*, Proceedings of the London Mathematical Society 45 (2nd series) (1939), p. 161-228.

[4] R. Péter, *Rekursive Funktionen*, Akademiai Kiado, Budapest 1951.

[5] S. C. Kleene, *General recursive functions*, Mathematische Annalen 112 (1936), p. 727-742.

Reçu par la Rédaction le 15. 9. 1952