Notre Dame Journal of Formal Logic
Volume VIII, Numbers 1 and 2, April 1967

LIMITED UNIVERSAL AND EXISTENTIAL QUANTIFIERS IN COMMUTATIVE PARTIALLY ORDERED RECURSIVE ARITHMETICS

M. T. PARTIS

1. In this paper we shall be dealing with the two different types of recursive arithmetics which will be described as V-systems and C-systems. These arithmetics have the following properties.

V-*systems*

(1) Every number $\mathbf{x}$ has n successors, denoted by S_1x, S_2x, . . . , S_nx.
(2) The system has three initial functions, namely, the zero function, $Z(x)$, written 0, the identity function, $I(x)$, written x, and n successor functions, S_vx, with $v = 1, 2, \ldots, n$.
(3) Primitive recursive functions can be defined by using the schema

$$\begin{aligned} F(x, 0) &= a(x) \\ F(x, S_vy) &= b_v(x, y, F(x, y)) \quad v = 1, 2, \ldots, n, \end{aligned}$$

where $a(x)$ and $b_v(x,y)$ are previously defined functions. Functions can also be defined explicitly by substitution.
(4) The system is made commutative by introducing the axiom

$$S_vS_ux = S_vS_ux \qquad u,v = 1, 2, \ldots, n,$$

and by stipulating that the functions used in a defining schema of the type given above satisfy the condition

$$b_v(x,S_uy,b_u(x,y,F(x,y))) = b_u(x,S_vy,b_v(x,y,F(x,y))).$$

C-*systems*

(1) The elements of the system are ordered sets of n natural numbers, written $(x_1, x_2, \ldots, x_n)$.
(2) Functions are defined as ordered sets of n primitive recursive functions in single successor recursive arithmetic, written

$$(f_1(x_1, \ldots, x_n), f_2(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n)).$$

The functions $f_1, f_2, \ldots, f_n$ are called component functions.
(3) Two functions in a C-system are said to be equal if their corresponding component functions are equal, i.e.

Received February 4, 1966

$$(f_1, f_2, \ldots, f_n) = (g_1, g_2, \ldots, g_n)$$
if and only if
$$f_i = g_i \text{ for } i = 1, 2, \ldots, n.$$

It was shown in a previous paper that an isomorphism can be established between a V-system and a C-system. The basis of this isomorphism is a (1-1) correspondence between the numbers of the two systems which is written

$$\mathbf{x} \leftrightarrow (x_1, x_2, \ldots, x_n).$$

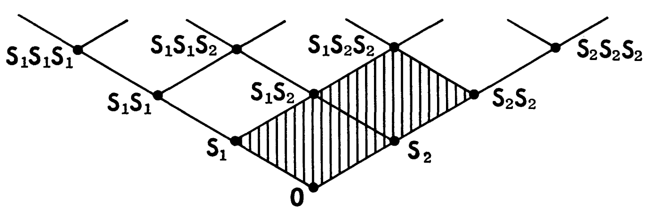
This relationship holds if and only if $\mathbf{x}$ is such that it contains x_1 successors of type $\mathbf{S}_1$, x_2 successors of type $\mathbf{S}_2$, . . . , and x_n successors of type $\mathbf{S}_n$.

Using this correspondence it is possible to establish a complete functional isomorphism between the systems in the sense that for any primitive recursive function in one system there is a corresponding primitive recursive function in the other. It is also possible to establish a deductive isomorphism between the systems. That is to say by introducing suitable rules of inference it can be shown that for every proof in one system there is a corresponding proof in the other.

2. In a V-system the inequality relationship can be defined as follows:

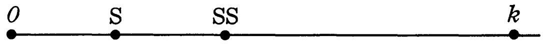
$$\mathbf{a} \leqslant \mathbf{b} \textit{ if and only if } \mathbf{a} \dot{-} \mathbf{b} = \mathbf{0}. \qquad \text{(See [2], p. 214)}$$

The class of numbers $\mathbf{x}$ such that $\mathbf{x} \leqslant \mathbf{k}$ will be written $\ell_{\mathbf{k}}$. For example, consider a V-system with two successors. Then $\ell_{\mathbf{S}_1\mathbf{S}_2\mathbf{S}_2}$ consists of the numbers $\mathbf{0}$, $\mathbf{S}_1$, $\mathbf{S}_2$, $\mathbf{S}_1\mathbf{S}_2$, $\mathbf{S}_2\mathbf{S}_2$, $\mathbf{S}_1\mathbf{S}_2\mathbf{S}_2$. This is illustrated in the diagram below.



The problem we shall be concerned with in this paper is to show that limited universal and existential quantifiers can be introduced to cover all members of a class $\ell_{\mathbf{k}}$ and further that these quantifiers have primitive recursive representing functions. This problem has been solved by V. Vučković in [3], but an alternative solution can be found using the functional isomorphism between V-systems and C-systems.

In single successor recursive arithmetic the limited universal and existential quantifiers are arrived at by way of sum and product functions taken over all values less than or equal to some designated number k. The process can be illustrated diagramatically by representing the numbers of a single successor recursive arithmetic as points on a straight line



Then the expression,

(2.1) *For all x less than or equal to k*, $f(x) = 0$,

is equivalent to the expression,

(2.2) $f(0) + f(\mathrm{S}) + f(\mathrm{SS}) + \ldots + f(k) = 0.$

By introducing the function $\Sigma_f(x)$, defined by

$$\Sigma_f(0) = f(0)$$
$$\Sigma_f(\mathrm{S}x) = \Sigma_f(x) + f(\mathrm{S}x),$$

expression (2.2) can be written

(2.3) $$\Sigma_f(k) = 0.$$

The limited existential quantifier is introduced in a similar way.

In a commutative partially ordered recursive arithmetic the problem is more difficult but the approach is fundamentally the same. Suppose we are working in a V-system with two successors and we wish to say

(2.4) *For all* $\mathbf{x}$ *less than or equal to* $\mathbf{k}$, $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.

The first stage in tackling this expression is to write it in an equivalent form in the C-system, i.e.

(2.5) *For all* (x_1, x_2) *such that* x_1 *is less than or equal to* k_1 *and* x_2 *is less than or equal to* k_2, $(f_1(x_1, x_2), f_2(x_1, x_2)) = (0, 0)$.

By analogy with single successor recursive arithmetic the next step is to introduce a sum function which runs systematically through the values of $(f_1(x_1, x_2), f_2(x_1, x_2))$ for all values of (x_1, x_2) such that $x_1 \leq k_1$ and $x_2 \leq k_2$. This is done by using the remainder and quotient functions of single successor recursive arithmetic. These are defined as follows:

$$\mathrm{r}(0, b) = 0$$
$$\mathrm{r}(\mathrm{S}a, b) = \mathrm{Sr}(a, b) \,.\, \alpha(\mathrm{Sr}(a, b), b),$$

where $\alpha(x, y) = 0$ if $x = y$ and 1 otherwise,

$$\mathrm{q}(0, b) = 0$$
$$\mathrm{q}(\mathrm{S}a, b) = \mathrm{q}(a, b) + (1 \dot{-} \alpha(\mathrm{Sr}(a, b), b)).$$

These functions have the following properties (see [1], pages 86-89).

(2.61) $a = \mathrm{r}(a, b) + b \,.\, \mathrm{q}(a, b)$
(2.62) $0 = b \rightarrow \mathrm{r}(a, b) < b$
(2.63) $\{(a = b \,.\, c + d) \;\&\; (d < b)\} \rightarrow \{c = \mathrm{q}(a, b) \;\&\; d = \mathrm{r}(a, b)\}$

3. The class of ordered sets $(x_1, \ldots, x_n)$ such that $x_i \leq k_i$ for $i = 1, \ldots, n$ will be represented by D_k. It is clear that there will be a (1-1) correspondence between the members of D_k and the members of class $\boldsymbol{b}_{\mathrm{k}}$ defined early in section 2.

We next introduce a primitive recursive C-function

$$(\mathrm{h}_1(y, k_1, \ldots, k_n), \ldots, \mathrm{h}_n(y, k_1, \ldots, k_n)).$$

We want this function to have the properties that

(3.1) $(h_1, \ldots, h_n)$ *is a member of* D_k.

(3.2) *If* $(a_1, \ldots, a_n)$ *is a member of* D_k, *then a* y *can be found such that* $(h_1, \ldots, h_n) = (a_1, \ldots, a_n)$.

The functions $h_1, \ldots, h_n$ are defined as follows.

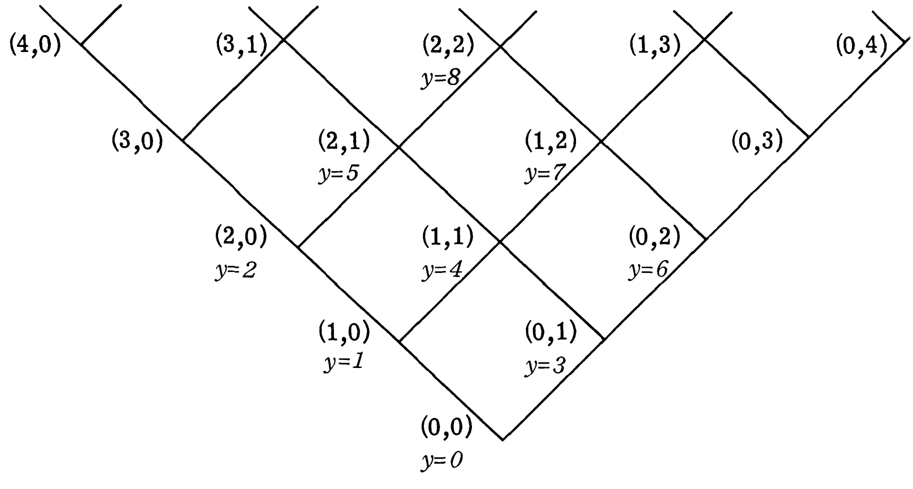
$$
\begin{aligned}
&h_1(y,k_1, .., k_n) = r(y,Sk_1)\\
&h_2(y,k_1, .., k_n) = r(q(y,Sk_1), Sk_2)\\
(3.3)\quad &h_3(y,k_1, .., k_n) = r(q(q(y, Sk_1), Sk_2), Sk_3)\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&h_n(y,k_1, .., k_n) = r(q(..q(y,Sk_1),Sk_2), ..), Sk_n)
\end{aligned}
$$

The reasons for selecting the above combinations of the remainder and quotient functions to define $h_1, \ldots, h_n$ will not be immediately apparent. The functions are chosen in this particular form in order to satisfy (3.1) and (3.2). That they do satisfy these conditions is shown below.

An example will serve to clarify the nature of the functions defined by (3.3). Suppose $K \leftrightarrow (2, 2)$; then since the first component place can be filled in three different ways and the second component place also in three different ways, it follows that D_k will have 9 members. The table below shows the value of (h_1, h_2) as y varies from *0* to *8*.

y	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
(h_1,h_2)	(0,0)	(1,0)	(2,0)	(0,1)	(1,1)	(2,1)	(0,2)	(1,2)	(2,2)

The lattice diagram below illustrates the same example.



A useful way of thinking of the values of $(h_1, \ldots, h_n)$ in the general case is to suppose that as y increases we count in the scale of Sk_1 in the first component place, in the scale of Sk_2 in the second component place, and so on.

(3.4) THEOREM. *The functions* $(h_1, \ldots, h_n)$ *defined by* (3.3) *satisfy conditions* (3.1) *and* (3.2).

PROOF. From (2.62) and (3.3) it follows that

$$h_i(y, k_1, \ldots, k_n) \leqslant k_i \text{ for } i = 1, \ldots, n.$$

(3.41) $\therefore (h_1, \ldots, h_n)$ is a member of D_k.

Thus condition (3.1) is satisfied.

To show that condition (3.2) is satisfied is more involved. From (2.61) and (3.3),

$$\begin{aligned} y &= r(y, Sk_1) + Sk_1 . q(y, Sk_1) \\ &= h_1 + Sk_1 . q(y, Sk_1) \end{aligned} \tag{3.42}$$

Then, again by (2.61),

$$\begin{aligned} q(y, Sk_1) &= r(q(y, Sk_1), Sk_2) + Sk_2 . q(q(y, Sk_1), Sk_2) \\ &= h_2 + Sk_2 . q(q(y, Sk_1), Sk_2) \end{aligned}$$

Hence, from (3.42),

$$y = h_1 + Sk_1 . h_2 + Sk_1 . Sk_2 . q(q(y, Sk_1), Sk_2).$$

By continuing this process the following expression is obtained.

$$\begin{aligned} y &= h_1 + Sk_1 . h_2 + Sk_1 . Sk_2 . h_3 + \ldots + Sk_1 . \ldots\ldots . Sk_{n-1} . h_n \\ &\quad + Sk_1 . \ldots\ldots . Sk_n . q(. . q(y, Sk_1), \ldots, Sk_n) \end{aligned} \tag{3.43}$$

Hence, for any member of D_k, say $(a_1, \ldots, a_n)$, a value of y such that $h_i = a_i$, for $i = 1, \ldots, n$, is given by

$$y = a_1 + Sk_1 . a_2 + \ldots + Sk_1 . \ldots\ldots . Sk_{n-1} . a_n \tag{3.44}$$

It can be deduced from (3.43) that there is more than one value of y such that $h_1 = a_1, \ldots, h_n = a_n$. However, (3.44) does in fact give the least such value of y though this will not be proved since it is not relevant to the main discussion.

To see that the value of y given by (3.44) does have the required property we first observe that $a_i < Sk_i$ since $(a_1, \ldots, a_n)$ is a member of D_k. Then, from (2.63) and (3.44),

$$a_1 = r(y, Sk_1) = h_1$$

and $a_2 + Sk_2 . a_3 + \ldots + Sk_2 . \ldots\ldots Sk_{n-1} . a_n = q(y, Sk_1)$. Hence, applying (2.63) again,

$$a_2 = r(q(y, Sk_1), Sk_2) = h_2$$

and $a_3 + Sk_3 . a_4 + \ldots + Sk_3 . \ldots\ldots . Sk_{n-1} . a_n = q(q(y, Sk_1), Sk_2)$.
By repeated application of (2.63) we obtain

$$a_1 = h_1,\ a_2 = h_2, \ldots, a_n = h_n.$$

Hence condition (3.2) is satisfied. This result together with (3.41) establishes the theorem.

The maximum value of y which can be obtained from (3.44) will occur when $a_1 = k_1,\ a_2 = k_2, \ldots, a_n = k_n$. That is to say,

$$y_{\max} = k_1 + Sk_1 . k_2 + \ldots + Sk_1 . Sk_2 . \ldots\ldots . Sk_{n-1} . k_n$$
$$= \phi(k_1, \ldots, k_n), \text{ say.}$$

Hence, as y varies from 0 to $\phi(k_1, \ldots, k_n)$, $(\mathrm{h}_1, \ldots, \mathrm{h}_n)$ is equal in turn to each member of D_k.

4. The functions $\mathrm{h}_1, \ldots, \mathrm{h}_n$ which were introduced and discussed in section 3 can now be used to define a summation function.

(4.1) THEOREM. *If* $\mathbf{F(x)}$ *is a primitive recursive* V-*function, then there exists a primitive recursive* V-*function,* $\boldsymbol{\Sigma}_{\mathbf{F}}\mathbf{(x)}$, *such that* $\boldsymbol{\Sigma}_{\mathbf{F}}\mathbf{(k)} = \mathbf{0}$ *if and only if* $\mathbf{F(x)} = \mathbf{0}$ *for all* $\mathbf{x} \leqslant \mathbf{k}$.

PROOF. Let $\mathbf{F(x)} \leftrightarrow (f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n))$.
The n primitive recursive functions,

$$\Sigma_{f_1}(y, k_1, \ldots, k_n), \ldots, \Sigma_{f_n}(y, k_1, \ldots, k_n)$$

are defined by the schema

$$\Sigma_{f_i}(0, k_1, \ldots, k_n) = f_i(0, 0, \ldots, 0)$$

(4.11) $\Sigma_{f_i}(\mathrm{S}y, k_1, \ldots, k_n) = \Sigma_{f_i}(y, k_1, \ldots, k_n) + f_i(\mathrm{h}_1(\mathrm{S}y, k_1, \ldots, k_n), \ldots,$
$\mathrm{h}_n(\mathrm{S}y, k_1, \ldots, k_n))$ for $i = 1, \ldots, n$.

Hence as y varies from 0 to $\phi(k_1, \ldots, k_n)$, $f_i(\mathrm{h}_1, \ldots, \mathrm{h}_n)$ takes all possible values for $(x_1, \ldots, x_n)$ in D_k.

Hence the ordered set of n primitive recursive functions defined by (4.11) will equal $(0, 0, \ldots, 0)$ if and only if $(f_1, f_2, \ldots, f_n)$ is equal to $(0, 0, \ldots, 0)$ for all members of D_k.

Since $\mathbf{F(x)} \leftrightarrow (f_1, f_2, \ldots, f_n)$ this will be the case if and only if $\mathbf{F(x)} = \mathbf{0}$ for all $\mathbf{x} \leqslant \mathbf{k}$. By the isomorphism theorem proved in [2], there exists a primitive recursive V-function, $\boldsymbol{\Sigma}_{\mathbf{F}}\mathbf{(x)}$, such that

$$\boldsymbol{\Sigma}_{\mathbf{F}}\mathbf{(x)} \leftrightarrow (\Sigma_{f_1}(\phi(x_1, \ldots, x_n), x_1, \ldots, x_n), \ldots, \Sigma_{f_n}(\phi(x_1, \ldots, x_n), x_1, \ldots, x_n)).$$

Hence $\boldsymbol{\Sigma}_{\mathbf{F}}\mathbf{(k)} = \mathbf{0}$ if and only if $\mathbf{F(x)} = \mathbf{0}$ for all $\mathbf{x} \leqslant \mathbf{k}$.

The above theorem shows that the limited universal quantifier has a primitive recursive representing function in the V-system. In other words the expression

$$\mathbf{A}_{\mathbf{x}}^{\mathbf{k}}(\mathbf{F(x)} = \mathbf{0})$$

which is to be read 'for all $\mathbf{x}$ less than or equal to $\mathbf{k}$, $\mathbf{F(x)} = \mathbf{0}$', has the primitive recursive representing function $\boldsymbol{\Sigma}_{\mathbf{F}}\mathbf{(k)}$.

5. The method used in the previous section to produce the limited universal quantifier can be applied, in a slightly modified form, to produce the limited existential quantifier.

(5.1) THEOREM. *If* $\mathbf{F(x)}$ *is a primitive recursive* V-*function, then there exists a primitive recursive* V-*function,* $\boldsymbol{\Pi}_{\mathbf{F}}\mathbf{(x)}$, *such that* $\boldsymbol{\Pi}_{\mathbf{F}}\mathbf{(k)} = \mathbf{0}$ *if and only if* $\mathbf{F(x)} = \mathbf{0}$ *for some* $\mathbf{x} \leqslant \mathbf{k}$.

PROOF. Let $\mathsf{F}(\mathsf{x}) \leftrightarrow (f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n))$.
Define the primitive recursive function, $\Pi_f(y, k_1, \ldots, k_n)$, by the schema

$$\Pi_f(0, k_1, \ldots, k_n) = f_1(0, 0, \ldots, 0) + \ldots + f_n(0, 0, \ldots, 0)$$
$$\Pi_f(Sy, k_1, \ldots, k_n) = \Pi_f(y, k_1, \ldots, k_n) . [f_1(h_1(Sy, k_1, \ldots, k_n), \ldots, h_n(Sy, k_1, \ldots, k_n)) + \ldots + f_n(h_1(Sy, k_1, \ldots, k_n), \ldots, h_n(Sy, k_1 \ldots, k_n))]$$

Then $\Pi_f(\phi(k_1, \ldots, k_n), k_1, \ldots, k_n) = 0$ if and only if there is an ordered set $(x_1, \ldots, x_n)$ in D_k such that $(f_1, \ldots, f_n) = (0, 0, \ldots, 0)$.

Now consider the C-function

$$\Pi_f(\phi(k_1, \ldots, k_n), k_1, \ldots, k_n), 0, \ldots, 0).$$

This, too, will equal $(0, 0, \ldots, 0)$ if and only if $(f_1, \ldots, f_n) = (0, \ldots, 0)$ for some $(x_1, \ldots, x_n)$ in D_k.

By the isomorphism theorem between the V-system and the C-system there exists a primitive recursive V-function, $\Pi_{\mathsf{F}}(\mathsf{x})$, such that $\Pi_{\mathsf{F}}(\mathsf{x}) \leftrightarrow (\Pi_f(\mathsf{k})(\phi(x_1, \ldots, x_n), x_1, \ldots, x_n), 0, \ldots, 0)$. Then $\Pi_{\mathsf{F}}(\mathsf{k}) = 0$ if and only if $\mathsf{F}(\mathsf{x}) = 0$ for some $\mathsf{x} \leqslant \mathsf{k}$.

The above theorem shows that the limited existential quantifier has a primitive recursive representing function in the V-system. In other words the expression

$$\mathsf{E}_{\mathsf{x}}^{\mathsf{k}}(\mathsf{F}(\mathsf{x}) = 0),$$

which is to be read 'for some x less than or equal to k, $\mathsf{F}(\mathsf{x}) = 0$', has the primitive recursive representing function $\Pi_{\mathsf{F}}(\mathsf{k})$.

REFERENCES

[1] R. L. Goodstein, *Recursive number theory*, Amsterdam, 1957.

[2] M. T. Partis, Commutative partially ordered recursive arithmetics, *Mathematica Scandinavica*, v. 13, 1963, pages 199-216.

[3] V. Vučković, Einfuhrung von $\Sigma_f(x)$ und $\Pi_f(x)$ in der Rekursiven Gitterpunktarithmetik, *Bulletin of the Institute of Mathematics, Bulgarian Academy of Sciences*, 1962(6), pages 15-25.

University of Leicester
Leicester, England