A note on omitting the replacement schema

Notre Dame Journal of Formal Logic
Volume XIV, Number 1, January 1973
NDJFAM

A NOTE ON OMITTING THE REPLACEMENT SCHEMA

A. BUNDY

In [1] Heath considers a formalisation of primitive recursive arithmetic similar to that given in Goodstein [2], in which the replacement schema (Goodstein's $\mathbf{Sb}_2$) is deduced from special cases of itself, using a double recursive uniqueness rule. The deduction of $\mathbf{Sb}_2$ given in [1] is, however, incomplete. This is rectified in the present note. The special cases of $\mathbf{Sb}_2$ taken by Heath are:

(i) $A = B \vdash \mathrm{S}A = \mathrm{S}B$
(ii) $A = B \vdash x + A = x + B$
(iii) $A = B \vdash A + x = B + x$
(iv) $A = B \vdash x \dot{-} A = x \dot{-} B$
(v) $A = B \vdash A \dot{-} x = B \dot{-} x$

Remark In fact either (ii) or (iii) can be omitted since $x + y = y + x$ can be proved without using (ii) or (iii) and then one can be derived from the other.

In order to derive the full $\mathbf{Sb}_2$, i.e., $A = B \vdash f(A) = f(B)$, for any primitive recursive function f, it is necessary to show that the substitution theorem, $x = y \rightarrow f(x) = f(y)$, persists under definition by a primitive recursive schema. Heath shows that it persists under the recursion without parameter, which I shall call $\mathbf{R}$,

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

i.e., that from $x = y \;\&\; w = z \rightarrow g(x, w) = g(y, z)$ we can deduce $x = y \rightarrow f(x) = f(y)$. He then quotes a theorem of R. M. Robinson that all primitive recursive functions are generated from $0, x, \mathrm{S}x, x + y$ and $x \dot{-} y$ by substitution and the recursion $\mathbf{R}$. To complete the proof it would be sufficient to show that Robinson's reduction of primitive recursion can be carried out in the restricted primitive recursive arithmetic (i.e., without full $\mathbf{Sb}_2$). This would involve defining the pairing functions $\mathrm{J}(x, y)$, $\mathrm{K}(x)$ and $\mathrm{L}(x)$ given by Robinson, deriving their main properties, e.g. $\mathrm{L}(\mathrm{S}x) \neq 0 \rightarrow \mathrm{K}(\mathrm{S}x) = \mathrm{K}(x) \;\&\; \mathrm{L}(\mathrm{S}x) = \mathrm{S}(\mathrm{L}x)$, and checking that the substitution theorem is satisfied by them. This part was omitted by Heath, and it is not clear that this programme could be carried out.

Received October 7, 1971

However it is fairly easy to check that the substitution theorem persists under full recursion, by a simple adaptation of Heath's proof for the recursion scheme **R**, as the following theorem shows.

Theorem *Suppose f is defined by primitive recursion from h and g, i.e.,*

$$f(u_0, \ldots, u_n, 0) = h(u_0, \ldots, u_n) \quad \text{(a)}$$
$$f(u_0, \ldots, u_n, \mathrm{S}x) = g(u_0, \ldots, u_n, x, f(u_0, \ldots, u_n, x)) \quad \text{(b)}$$

and the substitution theorem has already been proved for h and g, i.e.,

$$u_0 = v_0 \ \& \ \ldots \ \& \ u_n = v_n \rightarrow h(u_0, \ldots, u_n) = h(v_0, \ldots, v_n) \quad \text{(c)}$$

and

$$u_0 = v_0 \ \& \ \ldots \ \& \ u_{n+2} = v_{n+2} \rightarrow g(u_0, \ldots, u_{n+2}) = g(v_0, \ldots, v_{n+2}) \quad \text{(d)}$$

Then the substitution theorem holds for f, i.e.,

$$u_0 = v_0 \ \& \ \ldots \ \& \ u_{n+1} = v_{n+1} \rightarrow f(u_0, \ldots, u_{n+1}) = f(v_0, \ldots, v_{n+1})$$

Proof

Lemma I $u_0 = v_0 \ \& \ \ldots \ \& \ u_n = v_n \rightarrow f(u_0, \ldots, u_n, x) = f(v_0, \ldots, v_n, x)$

By induction on x, prove the basis

$$u_0 = v_0 \ \& \ \ldots \ \& \ u_n = v_n \rightarrow f(u_0, \ldots, u_n, 0) = f(v_0, \ldots, v_n, 0)$$

by hypotheses (a) and (c)

and the step

$$u_0 = v_0 \ \& \ \ldots \ \& \ u_n = v_n \ \& \ (u_0 = v_0 \ \& \ \ldots \ \& \ u_n = v_n \rightarrow$$
$$f(u_0, \ldots, u_n, x) = f(v_0, \ldots, v_n, x)) \rightarrow f(u_0, \ldots, u_n, \mathrm{S}x) = f(v_0, \ldots, v_n, \mathrm{S}x)$$

by hypotheses (b) and (d).

Lemma II $x = y \rightarrow f(u_0, \ldots, u_n, x) = f(u_0, \ldots, u_n, y)$

By double induction on x and y, prove

$$x = 0 \rightarrow f(u_0, \ldots, u_n, x) = f(u_0, \ldots, u_n, 0)$$

and

$$0 = y \rightarrow f(u_0, \ldots, u_n, 0) = f(u_0, \ldots, u_n, y)$$

by schema **F** on x and y respectively. Then use the deduction theorem to prove

$$(x = y \rightarrow f(u_0, \ldots, u_n, x) = f(u_0, \ldots, u_n, y)) \rightarrow$$
$$(\mathrm{S}x = \mathrm{S}y \rightarrow f(u_0, \ldots, u_n, \mathrm{S}x) = f(u_0, \ldots, u_n, \mathrm{S}y))$$

Assume $x = y \rightarrow f(u_0, \ldots, u_n, x) = f(u_0, \ldots, u_n, y)$ and $\mathrm{S}x = \mathrm{S}y$ and without using $\mathbf{Sb}_1$ on any of the variables $u_0, \ldots, u_n, x, y$, deduce, in turn,

$x = y$

$f(u_0, \ldots, u_n, x) = f(u_0, \ldots, u_n, y)$ by modus ponens

$g(u_0, \ldots, u_n, x, f(u_0, \ldots, u_n, x)) = g(u_0, \ldots, u_n, y, f(u_0, \ldots, u_n, y))$

by hypothesis (d).

Therefore

$$f(u_0, \ldots, u_n, \mathrm{S}x) = f(u_0, \ldots, u_n, \mathrm{S}y) \quad \text{by hypothesis (b).}$$

The theorem follows from Lemmas I and II.

REFERENCES

[1] Heath, I. J., "Omitting the replacement schema in recursive arithmetic," *Notre Dame Journal of Formal Logic*, vol. VIII (1967), pp. 234-238.

[2] Goodstein, R. L., "Logic-free formalization of recursive arithmetic," *Mathematica Scandinavia*, vol. 2 (1954), pp. 247-261.

University of Leicester,
Leicester, England