A note on omitting the replacement schema

Citation for published version:

Published In:
Notre Dame Journal of Formal Logic
A NOTE ON OMITTING THE REPLACEMENT SCHEMA

A. BUNDY

In [1] Heath considers a formalisation of primitive recursive arithme-
tic similar to that given in Goodstein [2], in which the replacement
schema (Goodstein’s $Sb_2$) is deduced from special cases of itself, using a
double recursive uniqueness rule. The deduction of $Sb_2$ given in [1] is,
however, incomplete. This is rectified in the present note. The special
cases of $Sb_2$ taken by Heath are:

(i) $A = B \vdash SA = SB$
(ii) $A = B \vdash x + A = x + B$
(iii) $A = B \vdash A + x = B + x$
(iv) $A = B \vdash x - A = x - B$
(v) $A = B \vdash x = B - x$

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 $Sb_2$, 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 param-
eter, which I shall call $R$,

\[ f(0) = (0), \]
\[ f(Sx) = g(x, f(x)), \]

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, $x, Sx, x + y$ and $x - y$ by substi-
tution and the recursion $R$. 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 $Sb_2$). This
would involve defining the pairing functions $J(x, y), K(x)$ and $L(x)$ given by
Robinson, deriving their main properties, e.g. $L(Sx) \neq 0 \rightarrow K(Sx) = K(x)$ &
$L(Sx) = S(Lx)$, 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.,

\begin{align*}
f(u_0, \ldots, u_n, 0) &= h(u_0, \ldots, u_n) & (a) \\
f(u_0, \ldots, u_n, s x) &= g(u_0, \ldots, u_n, x, f(u_0, \ldots, u_n, x)) & (b)
\end{align*}

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

\begin{align*}
u_0 = v_0 \& \ldots \& u_n = v_n \rightarrow h(u_0, \ldots, u_n) = h(v_0, \ldots, v_n) & (c) \\
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}) & (d)
\end{align*}

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

\begin{align*}
u_0 = v_0 \& \ldots \& u_n \& u_{n+1} = v_n \& u_{n+1} \rightarrow f(u_0, \ldots, u_{n+1}) = f(v_0, \ldots, v_{n+1})
\end{align*}

**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

\( u_0 = v_0 \& \ldots \& u_n = v_n \& (u_0 = v_0 \& \ldots \& u_n = v_n \rightarrow f(u_0, \ldots, u_n, x) = f(v_0, \ldots, v_n, x)) \rightarrow f(u_0, \ldots, u_n, s x) = f(v_0, \ldots, v_n, s x) \)

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

\( (x = y \rightarrow f(u_0, \ldots, u_n, x) = f(u_0, \ldots, u_n, y)) \rightarrow (s x = s y \rightarrow f(u_0, \ldots, u_n, s x) = f(u_0, \ldots, u_n, s y)) \)

Assume \( x = y \rightarrow f(u_0, \ldots, u_n, x) = f(u_0, \ldots, u_n, y) \) and \( s x = s y \) and without using $Sb_1$ 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

\[ f(u_0, \ldots, u_n, Sx) = f(u_0, \ldots, u_n, Sy) \]

by hypothesis (b).

The theorem follows from Lemmas I and II.

REFERENCES


*University of Leicester,*  
*Leicester, England*