Chapter II

The last word on Leibniz?

1. *“8.12”, again*

The reader should recall—from Chapter I of our notes, “Post’s Theorem and other Tools”—the conventions regarding the use the symbols “☡” and “☡☡”

We show in Chapter II that Gries’s ([5]) Leibniz rules 8.12(a,b)—under a number of different formulations—remain valid derived rules in the E-logic of [10].

☡ Until Section 3 the quantifier “$*$” used in [5] will be the *logical* $\forall$ (equivalently, $\exists$). ☡

The twin rules “Leibniz (8.12)” ([5], p.148) are stated below in their “no-capture” versions, using contextual substitution (carefully defined in [10]) and “standard quantifier notation”.[†]

$$\frac{A \equiv B}{(\forall x)(C[p := A] \Rightarrow D) \equiv (\forall x)(C[p := B] \Rightarrow D)} \tag{8.12a}$$

and

$$\frac{D \Rightarrow (A \equiv B)}{(\forall x)(D \Rightarrow C[p := A]) \equiv (\forall x)(D \Rightarrow C[p := B])} \tag{8.12b}$$

We have proved the “weak” “full-capture” versions (stated below as (1) and (2)) in [10]. As it was remarked there, we cannot do any better: Full-capture “strong”[‡] versions will yield strong generalization which the E-logic of [10] does not support.[§]

$$\vdash A \equiv B \text{ implies } \vdash (\forall x)(C[p \setminus A] \Rightarrow D) \equiv (\forall x)(C[p \setminus B] \Rightarrow D) \tag{1}$$

[†] The translation into standard notation uses the **Trading Axiom** (9.2) of [5] and the definition “$(*x \mid \mathit{true} : A)$ *means* $(*x)A$, where ‘$*$’ is one of $\exists$ or $\forall$”.

[‡] “Weak” means that the premise is an absolute theorem. If this is not required, then we have a “strong” rule.

[§] “Strong” generalization is the—unavailable to our E-logic of [10]—“rule” $\dfrac{A}{(\forall x)A}$. Note that the E-logic of [11] *does* support this rule!

and

$$\vdash A \equiv B \text{ implies } \vdash (\forall x)(D \Rightarrow C[p \setminus A]) \equiv (\forall x)(D \Rightarrow C[p \setminus B]) \qquad (2)$$

In that connection we had also shown that the $D \Rightarrow$-part on the premise side of 8.12b had to be dropped from (2) above, otherwise rule (2) would become invalid ([10, 11]).

Finally, we had shown ([10]) that 8.12b is a valid derived rule if the premise has an absolute proof

$$\vdash D \Rightarrow (A \equiv B)$$

and that it is *not* valid as a "strong" rule (i.e., in the general case when the premise $D \Rightarrow (A \equiv B)$ does *not* have an absolute proof).

Similarly, 8.12a is not valid if the premise $A \equiv B$ is *not* absolute (take B to be *true*, C to be $\neg p$ and D to be *false* to obtain the invalid $A \vdash (\forall x)A$.)

Are there "practical" *conditions*—that are not so restrictive as to render the rules useless—under which 8.12(a, b) are valid with *non absolute* premises?

Rick Ganong has informed me of a (restricted) "strong" version of 8.12(a, b), proposed by Alan Dow. Namely, that the rules 8.12(a, b) remain valid with non absolutely provable premises, on the *condition* that *there is a proof of the premises such that the "assumptions"*† *used in the proof contain no free* x.

The theorem below proves the validity of these strong(er) rules, by proving them to be *derived rules* in E-logic of [10]. As we want to reserve the term "strong" to mean absolutely no restriction on the rules' premises, we call the strengthened rules just "stronger" (!)

1.1 Theorem. ("Stronger" 8.12) *We are in E-logic of* [10]*. Let* Γ *be a finite set of formulas that contain no free* x.

If Γ *can prove the premises of the rules* 8.12(a, b) *above, then it can also prove the consequences of the two rules.*

Pause. Why "finite"? Does this not restrict generality?

Proof. For 8.12a. Here is a Hilbert-style proof that yields the premise $A \equiv B$.

†Nonlogical axioms.

We continue it until we get the conclusion of 8.12a.

$$
\begin{array}{ll}
 & \Gamma \quad \Big\langle \text{A finite number of "assumptions" used. None has a free } x \Big\rangle \\
 & \vdots \\
(1) & A \equiv B \quad \Big\langle \text{Proved from } \Gamma \Big\rangle \\
(2) & (C[p := A] \Rightarrow D) \equiv (C[p := B] \Rightarrow D) \quad \Big\langle (1) \text{ and SLCS} \Big\rangle \\
(3) & (C[p := A] \Rightarrow D) \Rightarrow (C[p := B] \Rightarrow D) \quad \Big\langle (2) \text{ and } \models_{\mathbf{Taut}} \Big\rangle \\
(4) & (C[p := A] \Rightarrow D) \Leftarrow (C[p := B] \Rightarrow D) \quad \Big\langle (2) \text{ and } \models_{\mathbf{Taut}} \Big\rangle \\
(5) & (\forall x)\Big((C[p := A] \Rightarrow D) \Rightarrow (C[p := B] \Rightarrow D)\Big) \quad \Big\langle (3), \text{ gen., and cond. on } \Gamma \Big\rangle \\
(6) & (\forall x)\Big((C[p := A] \Rightarrow D) \Leftarrow (C[p := B] \Rightarrow D)\Big) \quad \Big\langle (4), \text{ gen., and cond. on } \Gamma \Big\rangle \\
(7) & (\forall x)(C[p := A] \Rightarrow D) \Rightarrow (\forall x)(C[p := B] \Rightarrow D) \quad \Big\langle (5), \mathbf{Ax4} \text{ and MP} \Big\rangle \\
(8) & (\forall x)(C[p := A] \Rightarrow D) \Leftarrow (\forall x)(C[p := B] \Rightarrow D) \quad \Big\langle (6), \mathbf{Ax4} \text{ and MP} \Big\rangle \\
(9) & (\forall x)(C[p := A] \Rightarrow D) \equiv (\forall x)(C[p := B] \Rightarrow D) \quad \Big\langle (7), (8) \text{ and } \models_{\mathbf{Taut}} \Big\rangle
\end{array}
$$

For 8.12b. Let some (finite set) of premises, Γ, where x does not occur free, manage to prove $D \Rightarrow (A \equiv B)$.

Thus we have:

$$
\begin{array}{ll}
 & \Gamma \quad \Big\langle \text{A finite number of "assumptions" used. None has a free } x \Big\rangle \\
 & \vdots \\
(1) & D \Rightarrow (A \equiv B) \quad \Big\langle \text{Proved from } \Gamma \Big\rangle \\
(2) & D \quad \Big\langle \text{Add as an "assumption"} \Big\rangle \\
(3) & A \equiv B \quad \Big\langle (1), (2) \text{ and MP} \Big\rangle \\
(4) & C[p := A] \equiv C[p := B] \quad \Big\langle (3) \text{ and SLCS} \Big\rangle
\end{array}
$$

By the Deduction theorem,

$$\Gamma \vdash D \Rightarrow (C[p := A] \equiv C[p := B]) \tag{$*$}$$

Thus,

$$
\begin{array}{ll}
 & \Gamma \\
 & \vdots \\
(1) & D \Rightarrow (C[p := A] \equiv C[p := B]) \qquad \big\langle \text{by } (*) \text{ above} \big\rangle \\
(2) & (D \Rightarrow C[p := A]) \equiv (D \Rightarrow C[p := B]) \qquad \big\langle (1), \text{ distr. of } \Rightarrow \text{ over } \equiv \text{ and EQN} \big\rangle \\
(3) & (D \Rightarrow C[p := A]) \Rightarrow (D \Rightarrow C[p := B]) \qquad \big\langle (2) \text{ and } \models_{\textbf{Taut}} \big\rangle \\
(4) & (D \Rightarrow C[p := A]) \Leftarrow (D \Rightarrow C[p := B]) \qquad \big\langle (2) \text{ and } \models_{\textbf{Taut}} \big\rangle \\
(5) & (\forall x)\Big((D \Rightarrow C[p := A]) \Rightarrow (D \Rightarrow C[p := B])\Big) \qquad \big\langle (3) \text{ and gen.} \big\rangle \\
(6) & (\forall x)\Big((D \Rightarrow C[p := A]) \Leftarrow (D \Rightarrow C[p := B])\Big) \qquad \big\langle (4) \text{ and gen.} \big\rangle \\
(7) & (\forall x)(D \Rightarrow C[p := A]) \Rightarrow (\forall x)(D \Rightarrow C[p := B]) \qquad \big\langle (5), \textbf{Ax4} \text{ and MP} \big\rangle \\
(8) & (\forall x)(D \Rightarrow C[p := A]) \Leftarrow (\forall x)(D \Rightarrow C[p := B]) \qquad \big\langle (6), \textbf{Ax4} \text{ and MP} \big\rangle \\
(9) & (\forall x)(D \Rightarrow C[p := A]) \equiv (\forall x)(D \Rightarrow C[p := B]) \qquad \big\langle (7), (8) \text{ and } \models_{\textbf{Taut}} \big\rangle
\end{array}
$$

□

While we are at it, we strengthen WLUS of [10].

1.2 Theorem. ("Stronger" WLUS) *If* Γ*, a finite set of assumptions, proves* $A \equiv B$*, then it also proves* $C[p\backslash A] \equiv C[p\backslash B]$—provided that *all* free occurrences of variables that *are* captured in the two substitutions above ***do not occur free*** in Γ.

Proof. The proof is a simple amendment of that for WLUS (Metatheorem 4.2 in [10]).

The induction step involves an induction on the formula C. The interesting case is when C is $(\forall x)D$. By I.H. we have $\Gamma \vdash D[p \setminus A] \equiv D[p \setminus B]$. By the Tautology Theorem $\Gamma \vdash D[p\backslash A] \Rightarrow D[p\backslash B]$, hence $\Gamma \vdash (\forall x)\big(D[p\backslash A] \Rightarrow D[p\backslash B]\big)$ by generalization—applicable due to the restriction on Γ.

Thus $\Gamma \vdash (\forall x)D[p \setminus A] \Rightarrow (\forall x)D[p \setminus B]$ by **Ax4** followed by MP.

Similarly we obtain $\Gamma \vdash (\forall x)D[p \setminus A] \Leftarrow (\forall x)D[p \setminus B]$ and are done by the Tautology Theorem. □

If the reader will pardon the pedantry, we note that "stronger" WLUS is weaker than "strong" [W]LUS (or SLUS). The latter—which we *cannot* have in our particular E-logic of [10], as already remarked—requires absolutely no restrictions on what free variables Γ has or does not have. In particular Γ could just be the set $\{A \equiv B\}$.

2. *About "3.83"*

On p.60 of [5] we find

(3.83) **Axiom, Leibniz:** $(e = f) \Rightarrow (E_e^z = E_f^z)$ (E any expression)

In the context of *Boolean* expressions, that is, *well-formed formulas* (which is *the* context of Chapter 3 in [5]), the above is *not* a new axiom, but follows by techniques of Chapter 4 (*loc cit*), namely the Deduction theorem applied to the instance of the *Rule* Leibniz below—where, in this case, the "=" above is an alias for "≡":

$$e \equiv f \vdash E_e^z \equiv E_f^z \tag{1}$$

for any *Boolean* expressions e, f, E and propositional variable z. The notation "E_e^z" is an abbreviation of "$E[z := e]$".

Of course, since (1) above holds in Predicate Calculus *as well* (by SLCS), so does 3.83.

However, there is a version of 3.83 that *is* different, and is worth emphasizing. This is when e, f and E are *non*-Boolean expressions (not *formulas*, that is), i.e., when they are *terms*, in which case z is an object variable.

In class and in [10] we are using $\approx$ as equality between terms ("objects") to avoid confusion with "=" which also means (!) "≡".

Thus we set here to explore (3.83′) below, which still is *not* an axiom in our ([10]) setting! It is a theorem (schema).

(3.83′) $e \approx f \Rightarrow E_e^z \approx E_f^z$ (e, f, E are terms, z an object variable)

We will see that (3.83′) is a consequence of **Ax6** of [10], namely

$$x \approx t \Rightarrow (A \equiv A[x := t]), \text{ for all terms } t \text{ and formulas } A$$

Ax6 is kind of a "mixed-type 3.83" ($\approx$ to the left and $\equiv$ to the right of $\Rightarrow$). It is the axiom most Logicians ([3, 4, 6, 7] and [2, 8][†]) take (along with **Ax5**) to characterize equality of *objects*, and call it "the Leibniz axiom" since it was invented by Leibniz, albeit in a "2nd-order version".[‡]

A word for the hedging "kind of": In the interest of "elegance", **Ax6** is given in a simple, somewhat user-unfriendly form.[§] Note the fact that to the left of $\equiv$ we have just the original A, *no substitution took place*, and the first term (to

[†] Actually, these two authors *split* **Ax6** into *two* axioms, one for predicate symbols and one for function symbols. They give [I state the unary case for convenience], $x \approx y \Rightarrow (P(x) \equiv P(y))$ and $x \approx y \Rightarrow (f(x) \approx f(y))$ for all variables x, y, predicates P and functions f.

[‡] That is, with a quantifier over predicates: $(t \approx s) \equiv (\forall P)(P_t^z \equiv P_s^z)$, where t, s are terms, z an object variable and P a predicate variable. Note the two "≡". We are not allowed to quantify over predicates in our logic.

[§] [4] offers even more elegance, and takes away a bit more from friendliness, by restricting A to be *atomic*.

the left of $\approx$) is just a variable. A distant cry from the user-friendly form (the one actually used in [3])

$$(3.83'') \qquad t \approx s \Rightarrow (A[x := t] \equiv A[x := s])$$

for any terms t, s and formulas A.

We show that we can have our pie (elegant axiom) and eat it too (user-friendliness, via theorems) in Theorem 2.6 below.

2.1 Lemma. *$\vdash x \approx y \Rightarrow y \approx x$ and $\vdash x \approx y \Rightarrow y \approx z \Rightarrow x \approx z$*

Proof. Easy exercise using **Ax6** and **Ax5** (the latter is "$x \approx x$" and all its other "partial generalizations"—see [10]) . $\square$

2.2 Lemma. *For any function symbol f of arity n,*

$$\vdash x \approx y \Rightarrow f(z_1, \ldots, z_i, x, z_{i+2}, \ldots, z_n) \approx f(z_1, \ldots, z_i, y, z_{i+2}, \ldots, z_n)$$

where the variable y is different from all the z_i.

Proof. Let A stand for the formula

$$f(z_1, \ldots, z_i, x, z_{i+2}, \ldots, z_n) \approx f(z_1, \ldots, z_i, y, z_{i+2}, \ldots, z_n)$$

Then, by **Ax6**,

$$\vdash x \approx y \Rightarrow \Big(f(z_1, \ldots, z_i, x, z_{i+2}, \ldots, z_n) \approx f(z_1, \ldots, z_i, y, z_{i+2}, \ldots, z_n) \equiv \\ f(z_1, \ldots, z_i, y, z_{i+2}, \ldots, z_n) \approx f(z_1, \ldots, z_i, y, z_{i+2}, \ldots, z_n) \Big)$$

The subformula "$f(z_1, \ldots, z_i, y, z_{i+2}, \ldots, z_n) \approx f(z_1, \ldots, z_i, y, z_{i+2}, \ldots, z_n)$" can be dropped. Why? By (**Ax5**) $(\forall w) w \approx w$ is an axiom, hence, by **Ax2** and MP we get

$$\vdash f(z_1, \ldots, z_i, y, z_{i+2}, \ldots, z_n) \approx f(z_1, \ldots, z_i, y, z_{i+2}, \ldots, z_n)$$

"Redundant true" does the rest. $\square$

2.3 Corollary. *For any function symbol f of arity n, and distinct variables x_i and y_i,*

$$\vdash x_1 \approx y_1 \Rightarrow \cdots \Rightarrow x_n \approx y_n \Rightarrow \Big(f(x_1, \ldots, x_n) \approx f(y_1, \ldots, y_n) \Big) \qquad (**)$$

Proof. (Sketch) Move all the $x_i \approx y_i$ to the left of $\vdash$ (invoking Deduction theorem). Then, using Lemma 2.2, we deduce

$$f(x_1, \ldots, x_n) \approx f(y_1, x_2, \ldots, x_n) \text{ from } x_1 \approx y_1$$

$$f(y_1, x_2, \ldots, x_n) \approx f(y_1, y_2, x_3, \ldots, x_n) \text{ from } x_2 \approx y_2$$
$$f(y_1, y_2, x_3, \ldots, x_n) \approx f(y_1, y_2, y_3, x_4, \ldots, x_n) \text{ from } x_3 \approx y_3$$
$$\vdots$$

lastly,

$$f(y_1, y_2, \ldots, y_{n-1}, x_n) \approx f(y_1, y_2, \ldots, y_{n-1}, y_n) \text{ from } x_2 \approx y_2$$

Transitivity from Lemma 2.1 does the rest. □

2.4 Corollary. *For any function symbol f of arity n, and any terms t_i and s_i, $i = 1, 2, \ldots, n$,*

$$\vdash t_1 \approx s_1 \Rightarrow \cdots \Rightarrow t_n \approx s_n \Rightarrow \Big(f(t_1, \ldots, t_n) \approx f(s_1, \ldots, s_n)\Big)$$

Proof. By 2.3 and the substitution theorem ([10], Cor. 3.7). □

In particular, 2.4 says that the restriction "and distinct variables x_i and y_i" in the statement of 2.3 is not significant in practice, since the t_i, s_i above can be *any* variables. The stated constraint just helped to honour the restriction stated in Lemma 2.2 and hence use the Lemma in the proof of 2.3.

2.5 Theorem. *The* 3.83′ *(p.5).*

Proof. We do induction on terms E.

Basis-1. E is a constant or a variable other than z. Then 3.83′ reads

$$e \approx f \Rightarrow E \approx E$$

and is a(n absolute) theorem indeed, by $\vdash E \approx E$ and tautological implication.

Basis-2. E is the variable z. Then 3.83′ reads

$$e \approx f \Rightarrow e \approx f$$

and is a(n absolute) theorem since it is a tautology (Post, again).

Induction step. E is $g(t_1, \ldots, t_n)$, where g is a function symbol of arity n and t_i, $i = 1, \ldots, n$, are terms.

Add now $e \approx f$ as an assumption so that we can use the Deduction theorem. The induction hypothesis guarantees the claim for all terms that are "simpler" or "smaller" than E. Thus, ***on the adopted assumption*** $e \approx f$ ***we have the following*** n ***non-absolute theorems***:

$$t_i[z := e] \approx t_i[z := f], \text{ for } i = 1, \ldots, n$$

By Corollary 2.4 and MP,

$$g(t_1[z := e], \ldots, t_n[z := e]) \approx g(t_1[z := f], \ldots, t_n[z := f])$$

is a (non absolute) theorem. The above is the same as (assuming we remember how substitution is defined!)

$$g(t_1, \ldots, t_n)[z := e] \approx g(t_1, \ldots, t_n)[z := f]$$

in short

$$E^z_e \approx E^z_f$$

By the Deduction theorem,

$$\vdash e \approx f \Rightarrow E^z_e \approx E^z_f.$$

□

2.6 Theorem. *The* 3.83″ *(p.6).*

Proof. Let z be a variable that does not occur in A as either free or bound.

Then $A[x := z]$ is defined, hence, by **Ax6**

$$\vdash x \approx z \Rightarrow (A \equiv A[x := z]) \tag{1}$$

Applying the theorem on (simultaneous) substitutions ([10], Cor. 3.7) via the simultaneous substitution $[x, z := t, s]$ we obtain (from (1))

$$\vdash t \approx s \Rightarrow (A[x := t] \equiv A[x := z][z := s])$$

i.e.,

$$\vdash t \approx s \Rightarrow (A[x := t] \equiv A[x := s])$$

□

An "in-house" proof of 3.83″ (that in reality just mimicks the proof of Cor. 3.7 of [10] in the special case above) is as follows:

Let z be a variable that does not occur in either t or s, and which moreover does not occur in A as either free or bound.

Then $A[x := z]$ is defined, hence, by **Ax6**

$$\vdash x \approx z \Rightarrow (A \equiv A[x := z]) \tag{1'}$$

Next, let w be a new variable—different from z—which is not free in either t or s and is neither free nor bound in A.

Thus, $A[x := w]$ is defined, and $A[x := z][x := w]$ is just $A[x := z]$. $\quad (O)$

By (1′) and generalization,

$$\vdash (\forall x)\Big(x \approx z \Rightarrow (A \equiv A[x := z])\Big)$$

Hence, by **Ax2** and MP (and using observation (O) above),

$$\vdash w \approx z \Rightarrow (A[x := w] \equiv A[x := z])$$

By generalization,

$$\vdash (\forall w)\Big(w \approx z \Rightarrow (A[x := w] \equiv A[x := z])\Big)$$

hence, by **Ax2** and MP,

$$\vdash t \approx z \Rightarrow (A[x := w][w := t] \equiv A[x := z][w := t]) \tag{2'}$$

Since $A[x := z]$ contains no w, and $A[x := w][w := t]$ is $A[x := t]$, $(2')$ becomes

$$\vdash t \approx z \Rightarrow (A[x := t] \equiv A[x := z]) \tag{3'}$$

One more "cycle" of what we have been doing, and we are done. So, generalize $(3')$ to get:

$$\vdash (\forall z)\Big(t \approx z \Rightarrow (A[x := t] \equiv A[x := z])\Big)$$

Following this by an invocation of **Ax2** and MP we obtain

$$\vdash t[z := s] \approx s \Rightarrow (A[x := t][z := s] \equiv A[x := z][z := s]) \tag{4'}$$

i.e.,

$$\vdash t \approx s \Rightarrow (A[x := t] \equiv A[x := s])$$

since (explaining the simplifications effected to $(4')$, from left to right)

- t has no free z
- $A[x := t]$ has no free z
- $A[x := z][z := s]$ is $A[x := s]$

3. *Reaching for the $*$'s*

We address here "the other quantifiers" only briefly and anecdotally since all these "others" are, or stem from, nonlogical symbols so that we cannot speak intelligently or completely about them ***in the absence of nonlogical axioms governing their intended behaviour***.

In particular, I am here only interested in what happens to Leibniz 8.12(a,b) when $*$ are one of "$+$" or "$\times$".

First off, both these latter symbols are *nonlogical* symbols. They are used in the text [5] in the context of Arithmetic, or "Peano Arithmetic".

The latter is a *theory* (based on first order logic), which has additional axioms (nonlogical) that tell us how its nonlogical symbols, namely, $+$, $\times$, “S”,[†] “$<$”, and “0” behave.

We sample the behaviour of the “binary” $+$ and $\times$ before we turn to their use in [5] as “quantifiers”.

Axioms for (binary) $+$ For all (object) variables x, y (of type $\mathbb{N}$, if we have other types too),

$$\begin{gathered} x + 0 \approx x \\ x + S(y) \approx S(x + y) \end{gathered}$$

The above is a *recursive* or *inductive* definition. The recursion is using the variable y as “recursion-variable”—x being just a “parameter”—using “value” “0” for the basis, and then telling us: “if you know how to do “$x + y$”, then the way to compute the “value” of $x + S(y)$[†] is to add “1” to what you already computed as the “value” of $x + y$.”

Axioms for (binary) $\times$ For all (object) variables x, y (of type $\mathbb{N}$, if we have other types too),

$$\begin{gathered} x \times 0 \approx 0 \\ x \times S(y) \approx (x \times y) + x \end{gathered}$$

To “serve” these inductive definitions (and other needs) Peano Arithmetic also features the “Induction Axiom”, to the effect that *for every formula A and (object) variable x (of type $\mathbb{N}$, of course), the following is an axiom*

$$A[x := 0] \wedge (\forall n)\Big(A[x := n] \Rightarrow A[x := S(n)]\Big) \Rightarrow (\forall x)A$$

Now, the above induction axiom is instrumental towards allowing us to *introduce new function symbols* in Peano Arithmetic by *recursive definitions* just like the above two.

Please note that a “ definition” in this context (“recursive definition”) is *formal*, i.e., defines a new *formal* symbol via *defining **Axioms***. As such it should not be confused with *informal abbreviations* we usually make, such as allowing the “text” $(\exists x)A$ be a brief way of writing $(\neg(\forall x)(\neg A))$.

As it is beyond our scope to pursue this discussion formally (for reasons well beyond the fact that I did not even spell out all the Peano axioms) we will continue our investigation *informally*, heavily “cheating” (!) on the way by—among other logical indiscretions—including concepts from *sets* in the theory.[‡]

[†] *The successor* unary function, meant to convey “+1”, i.e., $S(x)$, “has as standard meaning” $x + 1$.

[†] That is, $x + (y + 1)$.

[‡] Formal Peano Arithmetic is developed outside set theory and thus does not benefit from set-theoretic techniques and tools.

Given the above resolution, we will also stop using $\approx$ (and use "$=$" instead) to equate arithmetical objects, and will use $1, 2, 3, 4$, etc. (instead of $S(0), S(S(0)), S(S(S(0))), S(S(S(S(0))))$, etc.) and "$x + 1$" rather than "$S(x)$" in what follows.

We have stated that $+$ and $\times$ are "built-in" functions in Peano Arithmetic. We can have many (infinitely many) "user-defined" functions.

For example, we can define a new function h by the axioms

$$\begin{aligned} h(0) &= 1 \\ h(n+1) &= (n+1) \times h(n) \end{aligned}$$

$h(n)$ is our familiar "factorial function", usually denoted by "$n!$". This technique of axiomatically introducing new functions by recursive definitions is omnipresent in Arithmetic.

Here are the two *examples* of main interest in this section.

Suppose we are give a function f (of arity 2, to avoid needless "generality"). We first define a function called "*sum*" in terms of f by the axioms (recursive schema):

$$\begin{aligned} sum(0, y) &= 0 \\ sum(x+1, y) &= sum(x, y) + f(x, y) \end{aligned}$$

Thus, we have the following endless sequence of Peano Arithmetic theorems (stated without formal proof,[†] but whose validity should be *intuitively obvious*!)

$$sum(0, y) = 0$$

$$\begin{aligned} sum(1, y) &= f(0, y) \\ sum(2, y) &= f(0, y) + f(1, y) \\ sum(3, y) &= f(0, y) + f(1, y) + f(2, y) \end{aligned}$$

and so on.

This function, *sum*, is what we normally write as

$$\sum_{0 \le i < x} f(i, y)$$

or, in [5]-notation, using "+" to now mean the "quantifier" ($\sum$, really) rather than the "built-in" binary "+",

$$\big(+ i \,|\, 0 \le i < x : f(i, y) \big)$$

The "range" $0 \le i < x$ does not dictate any particular order of summation. However, commutativity and associativity of the (binary) "+" make the order of summation of the $f(i, y)$ irrelevant, hence we have chosen the convenient ascending (with respect to i) order.

[†] Which is not difficult. The first one is the first axiom for *sum* anyway.

Thus, an "application" of 8.12 here would be

$$\begin{aligned} t = s \vdash \big(+ i \,|\, (0 \le i < x)[x := t] : f(i,y)\big) \\ = \big(+ i \,|\, (0 \le i < x)[x := s] : f(i,y)\big) \end{aligned} \tag{a}$$

and

$$\begin{aligned} 0 \le i < x \Rightarrow t = s \vdash \big(+ i \,|\, 0 \le i < x : f(i,y)[y := t]\big) \\ = \big(+ i \,|\, 0 \le i < x : f(i,y)[y := s]\big) \end{aligned} \tag{b}$$

Now, *if neither* t *nor* s *have a free* i, then we can add all the terms $(f(0,y), f(1,y), \ldots)$ and do the substitutions into x or y *afterwards*, that is, *in this restricted case*, (a) and (b) become (a') and (b') below.

$$t = s \vdash sum(t,y) = sum(s,y) \tag{a'}$$

and

$$0 \le i < x \Rightarrow t = s \vdash sum(x,t) = sum(x,s) \tag{b'}$$

(a') is valid by Theorem 2.5 ($3.83'$).

As for (b'), we assume that we have obtained a proof of the premise

$$0 \le i < x \Rightarrow t = s$$

from assumptions that have no free i (compare with the restriction in Theorem 1.1). First off, if $x = 0$, then without consulting the hypothesis we have that $sum(x,t) = sum(x,s)$ is a theorem (from the first axiom for sum, $3.83'$, and Lemma 2.1). Let then $x > 0$.

By generalization we obtain $(\forall i)(0 \le i < x \Rightarrow t = s)$, hence the following segment of equational proof yields $t = s$:

$$\begin{aligned} & (\forall i)(0 \le i < x \Rightarrow t = s) \\ = & \Big\langle \text{WLUS} \Big\rangle \\ & (\forall i)\big((0 > i \vee x \ge i) \vee t = s\big) \\ = & \Big\langle t = s \text{ has no free } i \Big\rangle \\ & t = s \vee (\forall i)(0 > i \vee x \ge i) \\ = & \Big\langle (\forall i)(0 > i \vee x \ge i) \text{ is provably } \equiv \textit{false} \text{ since } x > 0 \Big\rangle \\ & t = s \end{aligned}$$

Having obtained a proof of $t = s$, $sum(x,t) = sum(x,s)$ follows by $3.83'$.

The verification of (b) gets somewhat trickier if t or s *do* have free occurrences of i, in which case the substitutions cannot be done "afterwards", and "*sum*" is therefore useless.

The assumption that $0 \le i < x \Rightarrow t = s$ was proved from premises with no free occurrences of i still holds.

To preserve our sanity, let t specifically be $g(i, y, z)$ and s specifically be $h(i, y, z)$, where g and h are functions of arity 3 that have already being introduced in Arithmetic.

We use Sum_g and Sum_h below to handle (b):

$$\begin{aligned} Sum_g(0, y, z) &= 0 \\ Sum_g(x + 1, y, z) &= Sum_g(x, y, z) + f(x, g(x, y, z)) \end{aligned}$$

and

$$\begin{aligned} Sum_h(0, y, z) &= 0 \\ Sum_h(x + 1, y, z) &= Sum_h(x, y, z) + f(x, h(x, y, z)) \end{aligned}$$

Informally,

$$Sum_g(x, y, z) = \big(+ i \,|\, 0 \le i < x : f(i, g(i, y, z))\big)$$

and

$$Sum_h(x, y, z) = \big(+ i \,|\, 0 \le i < x : f(i, h(i, y, z))\big)$$

Thus, (b) becomes

$$0 \le i < x \Rightarrow g(i, y, z) = h(i, y, z) \vdash Sum_g(x, y, z) = Sum_h(x, y, z) \qquad (b'')$$

and we prove it by induction on x. This induction must be carried out within Peano Arithmetic, but we agreed to be sloppy, so we sketch it informally.

The induction hypothesis allows

$$Sum_g(x, y, z) = Sum_h(x, y, z)$$

and we then try to prove

$$Sum_g(x + 1, y, z) = Sum_h(x + 1, y, z)$$

on the assumption that

$$0 \le i < x + 1 \Rightarrow g(i, y, z) = h(i, y, z) \qquad (1)$$

that is, we try to prove

$$Sum_g(x, y, z) + f(x, g(x, y, z)) = Sum_h(x, y, z) + f(x, h(x, y, z)) \qquad (2)$$

Now, since $0 \le x < x + 1$ is provable, we have a proof of $g(x, y, z) = h(x, y, z)$ from (1) and substitution (Cor. 3.7, [10][†]) followed by MP, hence $f(x, g(x, y, z)) = f(x, h(x, y, z))$ is provable by $3.83'$. The induction hypothesis and one more application of $3.83'$ yield (2).

[†]Which is applicable because (1) was proved from premises with no i.

For $x = 0$ (basis) the conclusion is valid (since $Sum_g(0, y, z) = 0$ and $Sum_h(0, y, z) = 0$ are axioms).

Before we look at case (a) (p.12) we generalize (b) by allowing the range $0 \le i < x$ to be a general formula $R[i, x]$, where "$[i, x]$" indicates dependence on the free variables i and x *without precluding* dependence on other free variables. This time we use this terse notation also for the terms t and s, i.e., $t[i]$ and $s[i]$—rather than specifying $g(i, y, z)$ and $h(i, y, z)$—to indicate dependence on i, sweeping under the rug the unimportant (possible) dependence on other variables.

We want to argue that still

$$\begin{aligned} 0 \le i < x \Rightarrow t = s \vdash \big(+i \,|\, R[i, x] : f(i, y)[y := t[i]]\big) \\ = \big(+i \,|\, R[i, x] : f(i, y)[y := s[i]]\big) \end{aligned} \tag{b_3}$$

To this end we introduce

- Range-*set* (depends on x and any other free variables of R; but *not* on i):

$$S[x] \stackrel{\text{def}}{=} \{i : R[i, x]\}$$

- The *characteristic function* of R:

$$c_R[i, x] = \begin{cases} 1 & \text{if } R[i, x] \\ 0 & \text{if } \neg R[i, x] \end{cases}$$

- The maximum element function, M:

$$M[x] \stackrel{\text{def}}{=} 1 + \max S[x]$$

The introduction of characteristic functions is possible in formal Peano Arithmetic. The definition yields the two theorems (of Arithmetic) that $c_R[i, x] = 1 \equiv R[i, x]$ and $c_R[i, x] = 0 \equiv \neg R[i, x]$. Informally, "$c_R[i, x] = 1$ iff $R[i, x]$ is 'true'" and "$c_R[i, x] = 0$ iff $R[i, x]$ is 'false'".

The M-function is undefined for those values of x, and any other free variables, that $S[x]$ is infinite. We allow it to have the value 1 when $S[x] = \emptyset$.

Thus, informally, using a brand new variable w,

$$\begin{aligned} &\big(+i \,|\, R[i, x] : f(i, y)[y := t[i]]\big) \\ &\qquad = \Big(+i \,|\, 0 \le i < M[x] : \big(c_R[i, x] \times f(i, y)\big)[y := t[i]]\Big) \\ &\qquad = \Big(+i \,|\, 0 \le i < w : \big(c_R[i, x] \times f(i, y)\big)[y := t[i]]\Big)\big[w := M[x]\big] \end{aligned}$$

and

$$\begin{aligned} &\big(+i \,|\, R[i, x] : f(i, y)[y := s[i]]\big) \\ &\qquad = \Big(+i \,|\, 0 \le i < M[x] : \big(c_R[i, x] \times f(i, y)\big)[y := s[i]]\Big) \\ &\qquad = \Big(+i \,|\, 0 \le i < w : \big(c_R[i, x] \times f(i, y)\big)[y := s[i]]\Big)\big[w := M[x]\big] \end{aligned}$$

and (b_3) reduces to the already verified case (b), using $3.83'$, *for all cases where* $M[x]$ *is defined.*

We now turn to case 8.12(a) (p.12):

We continue to assume that we have a proof of $t[i] = s[i]$ from assumptions, Γ, that have no free i.

We want to know if we can prove

$$\Big(+ i \,|\, R\big[i, t[i]\big] : f(i,y) \Big) = \Big(+ i \,|\, R\big[i, s[i]\big] : f(i,y) \Big) \tag{a_3}$$

from the same assumptions, Γ.

We define

$$S_t \stackrel{\text{def}}{=} \{i : R\big[i, t[i]\big]\} \text{ and } S_s \stackrel{\text{def}}{=} \{i : R\big[i, s[i]\big]\}$$

By $3.83''$,

$$\Gamma \vdash R\big[i, t[i]\big] \equiv R\big[i, s[i]\big]$$

hence (generalization and assumption on Γ)

$$\Gamma \vdash (\forall i)\Big(R\big[i, t[i]\big] \equiv R\big[i, s[i]\big]\Big)$$

thus

$$S_t = S_s \tag{1}$$

Note that the S_t and S_s may depend on free variables (but not on i). As before, we set $M_t = 1 + \max S_t$ and $M_s = 1 + \max S_s$. Thus, by (1), $M_t = M_s$ (or they are both undefined). Therefore, the question (a_3) is now whether Γ can prove

$$\big(+ i \,|\, 0 \le i < M_t : f(i,y) \big) = \big(+ i \,|\, 0 \le i < M_s : f(i,y) \big)$$

which it certainly can ((a') of p.12)—if M_t (and hence M_s) is defined—using $3.83'$ (recall that M_t and M_s have no free i).

We can redo all this (but will not!) changing "+" throughout into "×". The formal function for

$$\prod_{0 \le i < x} f(i,y)$$

or

$$\big(\times i \,|\, 0 \le i < x : f(i,y) \big)$$

we could call *pro*. It would satisfy the two axioms

$$\begin{aligned} pro(0,y) &= 1 \\ pro(x+1,y) &= pro(x,y) \times f(x,y) \end{aligned}$$

Indeed, one can use a more general algebraic system, that of a "commutative monoid"—that is, a nonempty set D along with a binary operation, "∗", that is associative, commutative, and has an identity—and with some effort successfully revisit 8.12(a, b) in cases where the range is finite. Finiteness allows us to give to the members of the range integer subscripts and effect recursive definitions *on the subscript* variable (to define things such as "$a_1 * a_2 * \cdots * a_n$"). Commutativity and associativity remove the ambiguity arising from order and grouping of "summation". Finally, the presence of identity makes for a graceful handling of empty ranges—*cf.* $pro(0, y)$ and $sum(0, y)$.

3.1 Remark. Proving theorems—the practice—is an art, aided by our recognition and utilization of "patterns".

Because of this, it is helpful that [5] draws our attention to the strong similarity between the *logical quantifier* rules 8.12(a, b) [for $* \in \{\exists, \forall\}$] and the "monoid" quantifier rules 8.12(a, b) [for $* \in \{+, \times, \text{other}\}$].

Some caution is necessary, however: Logical quantifier rules 8.12 are valid in *all* mathematics, they are logical rules and come equipped with a (formal proof) certificate. The other "quantifiers" have behaviour dictated by nonlogical axioms, and that behaviour must be formally certified before use. We have *not* provided such a "certificate" (nor did [5]).

All I have done above was to informally argue that 8.12(a, b) hold for Peano Arithmetic, but, do recall, I argued with the help of *sets* (with extreme discomfort this can be avoided).

Bibliography

[1] Jon Barwise, editor. *Handbook of Mathematical Logic.* North-Holland, Amsterdam, 1978.

[2] Jon Barwise. *An introduction to first-order logic*, chapter A.1, pages 5–46. In [1], 1978.

[3] N. Bourbaki. *Éléments de Mathématique; Théorie des Ensembles.* Hermann, Paris, 1966.

[4] Herbert B. Enderton. *A mathematical introduction to logic.* Academic Press, New York, 1972.

[5] David Gries and Fred B. Schneider. *A Logical Approach to Discrete Math.* Springer-Verlag, New York, 1994.

[6] Yu. I. Manin. *A Course in Mathematical Logic.* Springer-Verlag, New York, 1977.

[7] Elliott Mendelson. *Introduction to mathematical logic, 3rd Edition.* Wadsworth & Brooks, Monterey, California, 1987.

[8] Joseph R. Shoenfield. *Mathematical Logic.* Addison-Wesley, Reading, Massachusetts, 1967.

[9] Raymond M. Smullyan. *Gödel's Incompleteness Theorems.* Oxford University Press, Oxford, 1992.

[10] G. Tourlakis. A basic formal equational predicate logic. Technical Report CS-1998-09, York University, Dept. of Comp. Sci., 1998.

[11] G. Tourlakis. On the soundness and completeness of equational predicate logics. Technical Report CS-1998-08, York University, Dept. of Comp. Sci., 1998.