Estratto da

A. Zanardo (a cura di), *Atti degli incontri di logica matematica* Volume 4, Siena 27-30 maggio 1987.

Disponibile in rete su http://www.ailalogica.it

A SHORT COURSE IN INTUITIONISTIC METAMATHEMATICS

A.S. TROELSTRA
Department of Mathematics - Universiteit van Amsterdam

The text below is a shortened version of the two lectures presented. Full details may be found in the book "Constructivism in Mathematics" by A.S. Troelstra and D. van Dalen, chapter 9.

It will be shown how to prove that certain "derived rules" well-known to hold for many intuiionistic formal systems (such as the disjunction property and Church's rule) hold for an interesting, not too complicated and in many ways typical example, namely the operator-part of Feferman's theory of operators and classes. We call the theory **APP**, and although it is proof-theoretically of the same strength as intuitionistic and classical first-order arithmetic, we have to work harder to obtain the derived rules.

Preliminaries.

HA is intuitionistic first-order arithmetic, similar to classical first-order arithmetic **PA**, and formalized with symbols for all primitive recursive functions. **EL** is an extension of **HA** with function variables α, β, γ, ...; the domain of functions is supposed to be closed under "recursive in", i.e. we have the axiom schema of quantifier-free choice

QF-AC $\quad \forall n \exists m\, A(n,m) \rightarrow \exists \alpha \forall n\, A(n,\alpha n)$ (A quantifier-free).

Elementary recursion theory can be formalized in **HA** (for partial recursive functions) or **EL** (for partial recursive functionals).

Elementary inductive definitions relative to **HA**, **EL** or **APP** can be replaced by explicit definitions. We can profit from this fact since it is often easier to handle the inductive definitions with the corresponding induction principle for the defined classes directly, instead of working with the corresponding explicit definitions.

Description of **APP**.

APP is a first-order theory based on E^+-logic, that is intuitionistic predicate logic with a predicate **E** for "exists" (so **E**t means "t exists or "t denotes"); the deviations from ordinary predicate logic (in the form of a natural deduction system, say) consist in modified ∀-elimination and ∃-introduction rules:

$$\forall E^E\ \frac{\forall x\, A(x) \qquad \mathbf{E}t}{A(t)} \qquad \exists I^E\ \frac{A(t) \qquad \mathbf{E}t}{\exists x\, A(x)}$$

and an axiom "variables exist"

$$\mathbf{E}x.$$

For equality we have

$$\mathbf{E}t \leftrightarrow t = t,$$

and we define "equal and equally defined" by

$$t \simeq s := (\mathbf{E}t \vee \mathbf{E}s) \rightarrow t = s.$$

The language $\mathcal{L}$(**APP**) contains:

Variables x,y,z,u,v,w for operations,

Individual constants **k**, **s** (combinators), **p**, $\mathbf{p}_0$, $\mathbf{p}_1$ (pairing and unpairing operators), 0 (zero),S (successor), P (predecessor), and **d** (definition by numerical cases).

There is also a binary partial operator Ap for application: if t, s are terms, so is Ap(t,s); we write ts for Ap(t,s) and $t_1 \ldots t_n$ abbreviates $(\ldots((t_1t_2)t_3)\ldots)t_n$.

Predicate constants: = for equality, N for the natural numbers, and **E** for existence.

Prime formulas are of the forms t = s, **E**t, Nt (also written $t \in N$); formulas are built from prime formulas by →, ∧, ∨, ∀, ∃. We use some Abbreviations. $\bot := (0 = S0)$, and we may regard ∨ as defined by

$$A \vee B := \exists x{\in}N((x = 0 \rightarrow A) \wedge (x \neq 0 \rightarrow B)).$$

Furthermore the numerals are introduced as usual, and we sometimes write (s,t) for **p**st, $(t)_i$ for $\mathbf{p}_i t$.

n, m are often used for variables ranging over N, i.e. ∀n, ∃n abbreviate $\forall n(n{\in}N \rightarrow \ldots)$, $\exists n(n{\in}N \wedge \ldots)$.

Axioms and rules of **APP**.

$$\begin{cases} t = s \rightarrow \mathbf{E}t \wedge \mathbf{E}s, \quad Nt \rightarrow \mathbf{E}t; \\ \mathbf{E}(ts) \rightarrow \mathbf{E}t \wedge \mathbf{E}s. \\ \mathbf{E}t \leftrightarrow \exists x(x = t), \end{cases}$$

$$\begin{cases} Nt \wedge t = s \rightarrow Ns; \\ \mathbf{E}(rs) \wedge t = s \rightarrow rt = rs; \\ \mathbf{E}(tr) \wedge t = s \rightarrow tr = sr. \end{cases}$$

$$\begin{cases} \mathbf{E}t \rightarrow \mathbf{k}st \simeq s; \\ \mathbf{E}t \wedge \mathbf{E}t' \rightarrow \mathbf{E}(\mathbf{s}tt'), \ \mathbf{s}tt't'' \simeq tt''(t't''); \\ \mathbf{E}t' \rightarrow \mathbf{p}_0(\mathbf{p}tt') \simeq t; \ \mathbf{E}t' \rightarrow \mathbf{p}_1(\mathbf{p}t't) \simeq t; \end{cases}$$

$$\begin{cases} 0 \in N, \quad t \in N \rightarrow St \in N, \quad St \neq 0; \\ P0 = 0, \quad t \in N \rightarrow Pt \in N, \ P(St) \simeq t; \\ \mathbf{E}t_1 \wedge \mathbf{E}t_2 \wedge t{\in}N \wedge t'{\in}N \wedge t{\neq}t' \rightarrow \mathbf{d}t_1t_2tt = t_1 \wedge \mathbf{d}t_1t_2tt' = t_2\,; \\ (0) \wedge \forall x{\in}N(Ax) \rightarrow A(Sx)) \rightarrow \forall x{\in}N\, A(x) \ \text{(induction)}. \end{cases}$$

The standard model of **APP** is PRO, the (codes of) partial recursive functions; the application operator Ap is partial recursive function application.

We can now adopt many techniques familiar from combinatory logic. As in combinatory logic one shows that for each term t there is a term, written as λx.t, such that

$$\mathbf{APP} \vdash \mathbf{E}(\lambda x.t) \wedge (\mathbf{E}s \rightarrow (\lambda x.t)s \simeq t[x/s]).$$

There is a fixed-point operator fix satisfying

$$\mathbf{APP} \vdash \mathbf{E}(\mathrm{fix}(x)) \wedge \mathrm{fix}(x)(y) \simeq x(\mathrm{fix}(x))(y),$$

and from this we construct a recursor **r** such that

$$\mathbf{E}t' \rightarrow \mathbf{r}tt'0 \simeq t,$$

$$Nt' \rightarrow \mathbf{r}tt'(St'') \simeq t'(\mathbf{r}tt't'')t''$$

and a minimum-operator. Thus all total recursive functions are available in **APP**, and **HA** can be embedded into **APP** as a subsystem; **EL** can also be embedded into **APP**. In fact **APP** is a conservative extension of **HA**, as may be seen by interpreting **APP** in PRO.

Term models.

CNFS is the model of closed terms in normal form with application defined by strict reduction.

A strict reduction sequence is a sequence where always the leftmost minimal redex is converted; and a redex is minimal if it does not have a proper subterm which is a redex.

Normal form, reduction sequence etc. are defined in the usual way from the conversions

$\mathbf{k}tt'$ conv t, $\mathbf{s}tt't''$ conv $tt''(t't'')$, $\mathbf{p}_i(t_0,t_1)$ conv t_i $(i\in\{0,1\})$,
$\mathbf{d}tt'\bar{n}\bar{n}$ conv t, $\mathbf{d}tt'\bar{n}\bar{m}$ conv t' if $\bar{n}\neq\bar{m}$ ($\bar{n}$, $\bar{m}$ are numerals)
$P(St)$ conv t, $P0$ conv 0.

Let $\succ_1$ denote one-step reduction, $\succcurlyeq$ reduction in general. t and t' are r-equal if there is a sequence $t=t_0, t_1, t_2, \dots, t_n=t'$ such that for all $i<n$ $t_i \preccurlyeq t_{i+1}$ or $t_{i+1} \succcurlyeq t_i$. The Church-Rosser theorem holds for this reduction relation. We now define application tt' in the term model as the unique t'' (if existing) such that $tt' \succcurlyeq t''$ by a strict reduction sequence, t'' closed normal; the rest of the interpretation is obvious.

A variant is the term model $CNFS^{\alpha}$, $\alpha\in N\to N$. Here we consider terms with a single fixed free variable x^* (with intended interpretation α), and extend our reduction relation by adding an infinite set of conversions

$x^*(S^n0)$ conv $S^{\alpha n}0$ (or $x^*(\bar{n})$ conv $\overline{\alpha n}$).

g-realizability.

Our next tool is an abstract modified version of Kleene's realizability. To each formula A of **APP** we assign a new formula $x\,\mathbf{g}\,A$ ("x realizes A"), $x\notin FV(A)$; $FV(x\,\mathbf{g}\,A)=\{x\}\cup FV(A)$, such that

(i) $x\,\mathbf{g}\,P := \mathbf{E}x\wedge P$ for P prime;
(ii) $x\,\mathbf{g}\,(A\wedge B) := (\mathbf{p}_0x\,\mathbf{g}\,A)\wedge(\mathbf{p}_1x\,\mathbf{g}\,B)$;
(iii) $x\,\mathbf{g}\,(A\to B) := \mathbf{E}x\wedge\forall y(y\,\mathbf{g}\,A\to xy\,\mathbf{g}\,B)\wedge(A\to B)$;
(iv) $x\,\mathbf{g}\,(\forall y\,A) := \forall y(xy\,\mathbf{g}\,A)$;
(v) $x\,\mathbf{g}\,(\exists y\,A) := \mathbf{E}(\mathbf{p}_0x)\wedge(\mathbf{p}_1x\,\mathbf{g}\,A)[y/\mathbf{p}_0x]$.

PROPOSITION

(i) **APP** $\vdash t\,\mathbf{g}\,A\to A\wedge\mathbf{E}t$;
(ii) For $\exists$-free A there is a term τ_A such that
APP $\vdash \exists x(x\,\mathbf{g}\,A)\leftrightarrow\tau_A\,\mathbf{g}\,A\leftrightarrow A$;

(iii) **APP** $\vdash A\Rightarrow$ **APP** $\vdash t\,\mathbf{g}\,A$ for some t, with $FV(t)\subset FV(A)$ (soundness).

COROLLARY. **APP** $\vdash\exists xBx\Rightarrow$ **APP** $\vdash Bt\wedge\mathbf{E}t$ with $FV(t)\subset FV(B)\setminus\{x\}$. □

COROLLARY. Let A be $\exists$-free. Then

APP $\vdash\forall x(Ax\to\exists y\,B(x,y))\Rightarrow$ **APP** $\vdash\forall x(Ax\to\mathbf{E}(tx)\wedge B(x,tx))$

for a term t with $FV(t)\subset FV(A)\cup FV(B)\setminus\{x\}$. □

Note that we can take $x\in N$ or $x\in N^N$ (i.e. $\forall y\in N(xy\in N)$) for Ax.

COROLLARY. Let $\exists x\in N\,C(x)$, $A\vee B$ be closed.

(i) **APP** $\vdash\exists x\in N\,C(x)\Rightarrow$ **APP** $\vdash C\bar{n}$ for some numeral $\bar{n}$.
(ii) **APP** $\vdash A\vee B\Rightarrow$ **APP** $\vdash A$ or **APP** $\vdash B$.

Proof. Suppose $\vdash\exists x\in N\,C(x)$, then by the preceding corollary $\vdash C(t)\wedge t\in N$ for some closed t. This must be true in the term model CNFS, so t strictly reduces to a numeral $\bar{n}$; but then **APP** $\vdash t=\bar{n}$, so $\vdash C(\bar{n})$. □

For further refinements we need to formalize some of our metamathematics. For if $\vdash\forall n\exists m\,C(n,m)$, we obtain for each numeral $\bar{n}$ $\vdash\exists m\,C(\bar{n},m)$, and by recursively searching through possible proofs we can find a numeral $\bar{m}$ such that $\vdash C(\bar{n},\bar{m})$, and so there is a recursive function f such that for all $\bar{n}$ $\vdash C(\bar{n},\overline{fn})$. But we need to do extra work if we want to show that this function can be taken to be provably recursive in **APP**.

We now concentrate on the so-called continuity rule, since ultimately "closure under Church's rule" is a special case of closure under the continuity rule.

Consider terms t with $FV(t)\subset\{x^*\}$. We can give an elementary inductive definition of $VAL^{\alpha}(\ulcorner t\urcorner,x)$, "x is the value of t under the assignment of α to x^*", and where $\ulcorner t\urcorner$ is the gödelnumber of t. The definition uses the clauses:

$VAL^{\alpha}(\ulcorner c\urcorner,c)$ for constants c;
$VAL^{\alpha}(\ulcorner x^*\urcorner,\alpha)$;
$VAL^{\alpha}(\ulcorner t_1\urcorner,x)\wedge VAL^{\alpha}(\ulcorner t_2\urcorner,y)\to VAL^{\alpha}(\ulcorner t_1t_2\urcorner,xy)$.

The proof of the following proposition is not difficult if we use the inductive definition.

PROPOSITION. Let $FV(t[v,x^*])\subset\{v,x^*\}$, then in **APP**

(i) $VAL^{\alpha}(\ulcorner t \urcorner, x) \wedge VAL^{\alpha}(\ulcorner t \urcorner, y) \rightarrow x = y$;
(ii) $\mathbf{E}(t[n,\alpha]) \rightarrow VAL^{\alpha}(\ulcorner t[\bar{n},x^*] \urcorner, t[n,\alpha])$;
(iii) $\forall n \in N\, \forall \alpha \in N^N\, \forall y (VAL^{\alpha}(\ulcorner t[\bar{n},x^*] \urcorner, y) \rightarrow t[n,\alpha] = y)$. □

Let $SRED^{\alpha}(x,y)$ express "t with $\ulcorner t \urcorner = x$ strictly reduces to t' with $\ulcorner t' \urcorner = y$ ". We note that $SRED^{\alpha}$ is r.e. in α and in particular

$$\vdash SRED^{\alpha}(x, \ulcorner \bar{n} \urcorner) \leftrightarrow \exists u (T^{\alpha}(\varphi(x),u) \wedge Uu = n),$$

where T is a suitable version of Kleene's T-predicate, with a single function argument α, φ primitive recursive in x. U is the result-extracting function; we may assume that a computation with code number u uses at most $\bar{\alpha}u$, i.e. values of α for arguments less than u.

THEOREM (Continuity rule). Let $FV(A) \subset \{x,\alpha\}$, then in **APP**

$$\vdash \forall \alpha \in N^N \exists x \in N\, A(\alpha,x) \Rightarrow \vdash \forall \alpha \in N^N (\exists u T^{\alpha}(\bar{m},u) \wedge A(\alpha,Uu))$$

for some numeral $\bar{m}$.

Proof. Let $\vdash \forall \alpha \in N^N \exists x \in N\, A(\alpha,x)$, then ($\mathbf{g}$-realizability), for a suitable term t*, $\vdash \forall \alpha \in N^N (t^*\alpha \in N \wedge A(\alpha,t^*\alpha))$. Hence in particular $\vdash x^* \in N^N \rightarrow t^*x^* \in N$; if we interpret this in the term model $CNFS^{\alpha}$ we obtain

$$\vdash \forall \alpha \in N^N \exists n \in N\, SRED^{\alpha}(\ulcorner t^*x^* \urcorner, \ulcorner \bar{n} \urcorner),$$

and hence

$$\vdash \exists u (T^{\alpha}(m,u) \wedge SRED^{\alpha}(\ulcorner t^*x^* \urcorner, \ulcorner \overline{Uu} \urcorner).$$

But we also have $VAL^{\alpha}(\ulcorner t^*x^* \urcorner, t^*\alpha)$, and with the propositions above

$$\vdash \exists u (T^{\alpha}(m,u) \wedge t^*\alpha = Uu).$$

The conclusion of the rule follows. □