Addenda to part I.

In §3.1 we referred to a result of Jacopini [1971] which has not yet been published. Since we have not seen his argument, we give here an outline of a possibly different proof based on the results of Ch.3.

Theorem (Jacopini)
There is an extensional combinatory model which is not an ω-model.
Proof (outline)
Let $\Omega = SII(SII)$.
Define $\mathcal{E} = \{\Omega KZ = \Omega SZ \mid Z$ is [redacted] a closed CL-term$\}$.
Then by 3.2.16 $CL + \mathcal{E} + ext$ is consistent since $\mathcal{E} \subset \mathcal{H}_{CL}$.
Further
$CL + \mathcal{E} + \omega\text{-rule} \vdash \Omega K = \Omega S$, but
$CL + \mathcal{E} + ext \nvdash \Omega K = \Omega S$.
Hence the term model of $CL + \mathcal{E} + ext$ is an extensional model which is not an ω-model, that is, its interior is not extensional. ⊠

Corollary.
There is no set of equations (in the language of CL), in fact no set of universal formulas, that is propositional combinations of equations, whose models are exactly the extensional models of CL.

For, any submodel, in particular the interior, of a model satisfying a set $\mathcal{E}$ of universal sentences also satisfies $\mathcal{E}$.

This corollary justifies the remark on page x of the Introduction to Part I where it was stated that the concept of extensionality of models of CL cannot be expressed by equations. (Only the set of equations which are valid in all extensional models is characterized by the equational rule ext; or more interestingly by the finite number of combinatory axioms, given in Curry and Feys [1958],Ch.6 C, which axiomatize the rule ext.

The proof above of Jacopini's theorem is also of interest for our work in the ω-rule in Chapter 2. If it should turn out that CL itself is ω-complete, that is that the ω-rule is simply valid for CL + ext, the result stated in §2.2 would be completely superseded. However, the method of §2.2 can be adapted to prove the consistency of the ω-rule, for CL + & , a system which is certainly not ω-complete. Thus, whether or not CL itself is ω-complete, at least for some extensions of CL + ext we certainly cannot establish the consistency of the ω-rule simply by proving ω-completeness.

We give here a short account of CL_I introduced by Rosser [1935], which is the combinatory counterpart of the λI-calculus.

Definition

We define a theory CL_I formulated in the following language:

Alphabet:	a,b,c,...	variables
	I,J	constants
	(,)	improper symbols
	⩾, =	reduction, equality

Terms: The terms are defined inductively by

1) Any variable or constant is a term
2) If M,N are terms, then (MN) is a term.

Formulas: If M,N are terms, then $M = N$ and $M \geqslant N$ are formulas.

Definition

CL_I is defined by the following axioms and rules:

I 1. $IM \geqslant M$

2. $JMNLP \geqslant MN(MPL)$

II, III } Same axioms and rules as for CL (cf. appendix I, A2 of part I)

In the above M,N,L and P are arbitrary terms.

In CL_I it is possible to define an abstraction operator λ^* simulating the λ-operator of the λI-calculus.

Lemma

In CL_I we can define closed terms B,C and S such that

$CL_I \vdash Babx = a(bx)$

$CL_I \vdash Cabx = axb$

$CL_I \vdash Sabx = ax(bx)$

Proof.

Define

T = JII

C = JT(JT)(JT)

B = C(JIC)(JI)

W = C(C(BC(C(BJT)T))T)

S = B(B(BW)C)(BB)

Then B, C and S have the required properties. ⊠

<u>Definition</u>

For every term M such that $x \in FV(M)$ we define a term $\lambda^* xM$.

$$\lambda^* xx = I$$

$$\lambda^* x(M_1M_2) = \begin{cases} BM_1(\lambda x^* M_2) & \text{if } x \notin FV(M_1) \text{ and } x \in FV(M_2) \\ C(\lambda^* xM_1)M_2 & \text{if } x \in FV(M_1) \text{ and } x \notin FV(M_2) \\ S(\lambda^* xM_1)(\lambda^* xM_2) & \text{if } x \in FV(M_1) \text{ and } x \in FV(M_2) \end{cases}$$

<u>Theorem</u>

For every term M with $x \in FV(M)$ we have

1) $FV(\lambda^* xM) = FV(M) - \{x\}$

2) $(\lambda^* xM)x \geqslant M$

Hence similar to 1.4 in part

<u>Corrections to Part I.</u>

<u>Introduction</u>. On p. xxi it was stated that for a CL-term M

M has no normal form $\Longleftrightarrow$ ¤$_{R'}$(M) = * .

This should be (see 2.30)

1) M has no normal form $\Rightarrow$ ¤$_{R'}$(M) = *

2) M is in normal form $\Rightarrow$ ¤$_{R'}$(M) ≠ *

<u>Appendix II</u>. Some statements have to be corrected by explicit reference to free and bound variables:

<u>Lemma 5</u>. If $FV(N) \cap BV(MM') = \emptyset$ and $FV(N') \cap BV(N') = \emptyset$ and if $\lambda' \vdash M \geqslant_1 M'$, $\lambda' \vdash N \geqslant_1 N'$, then

$\lambda' \vdash [x/N]M \geqslant_1 [x/N']M'$.

To prove lemma 5 one notes:

1) If $\lambda' \vdash M \geqslant_1 M'$, then $FV(M) = FV(M')$

2) If $y \notin FV(N_1)$ and $x \neq y$, then $[x/N_1]([y/N_2]M) \equiv [y/[x/N_1]M_2]([x/N_1]M)$

3) $[x/N_1]([x/N_2]M) \equiv [x/[x/N_1]N_2]M$

Lemma 7 follows from the new version of lemma 5 and the observations:

1) If $\lambda' \vdash M \geqslant_1 N$ and $N \equiv_\alpha N'$, then $\lambda' \vdash M \geqslant_1 N'$

2) For every M there exists an M' such that

$M \equiv_\alpha M'$ and $FV(M) \cap BV(M') = \emptyset$.

Addenda

Gödel, Kurt

[1958] Über eine bisher noch nicht benutzte Erweiterung des finiten Standpunktes, Dialectica, vol.12, pp.280-287.

Morris, James H.

[1968] Lambda-calculus models of programming languages, Ph.D.dissertation, MIT.

Strong, H.Raymond

[1968] Algebraically generalized recursive function theory, IBM J.Research and Development, vol.12, pp.465-475.