INFORMATION AND COMPUTATION **138**, 101–123 (1997)
ARTICLE NO. IC972655

On the Jacopini Technique

Jan Kuper

Department of Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
E-mail: jankuper@cs.utwente.nl

The general concern of the Jacopini technique is the question: "Is it consistent to extend a given lambda calculus with certain equations?" The technique was introduced by Jacopini in 1975 in his proof that in the untyped lambda calculus Ω is easy, i.e., Ω can be assumed equal to any other (closed) term without violating the consistency of the lambda calculus. The presentations of the Jacopini technique that are known from the literature are difficult to understand and hard to generalise. In this paper we generalise the Jacopini technique for arbitrary lambda calculi. We introduce the concept of *proof-replaceability* by which the structure of the technique is simplified considerably. We illustrate the simplicity and generality of our formulation of the technique with some examples. We apply the Jacopini technique to the **λμ**-calculus, and we prove a general theorem concerning the consistency of extensions of the **λμ**-calculus of a certain form. Many well known examples (e.g., the easiness of Ω) are immediate consequences of this general theorem.

1. INTRODUCTION

In 1975 Jacopini proved that in the untyped lambda calculus Ω is easy (cf. Jacopini, 1975), i.e., it is consistent to extend the lambda calculus with an equation of the form $\Omega = N$, where N is an arbitrary closed term. To prove this result, Jacopini used a certain proof theoretical technique. We shall call this technique the *Jacopini technique*. In 1979 Baeten and Boerboom proved the easiness of Ω by the same technique though in a different presentation (they also gave a proof by semantic means, cf. Baeten and Boerboom, 1979). Originally, Jacopini formulated his technique for a single equation. In (Mitchell, 1996, Sect. 4.4.4) a generalisation towards an arbitrary finite number of equations can be found. Other techniques for proving that certain terms are easy may be found in the literature, e.g., (Jacopini and Venturini Zilli, 1978, 1985; Intrigila, 1991; Berarducci and Intrigila, 1993).

In general, the Jacopini technique can be used to tackle questions of the form:

> Is it consistent to add one or more equations $P = Q$ (with P, Q closed) to some lambda calculus?

0890-5401/97 $25.00

In (Statman, 1986) a closely related question is considered, namely the question of whether equations of the form $M\vec{x} = N\vec{x}$ are solvable in $\vec{x}$ (in extensions of a model). Statman too uses the Jacopini technique.

Roughly, the idea of the Jacopini technique is as follows. Let λ be some (consistent) lambda calculus, and suppose that λ extended with an equation $P = Q$ is *in*consistent. Then there is a proof of, e.g., $0 = 1$ in this extended calculus. In this proof the equation $P = Q$ is used a number of times, say n times. Now for certain P and Q it is possible—by means of the Jacopini technique—to construct a new proof of $0 = 1$, in which the equation $P = Q$ is only used $n - 1$ times. By repeating this construction, a proof of $0 = 1$ can be found in which the equation $P = Q$ is not used at all. That is to say, $0 = 1$ is provable within λ itself, i.e., λ is inconsistent. This contradicts the assumption that λ is consistent, hence $\lambda + P = Q$ is also consistent. This line of reasoning remains the same if more than one equation is added to the calculus.

In the presentations mentioned above it is difficult to recognize the essential structure of the Jacopini technique. Furthermore, it is hard to see how it can be generalised towards other calculi. The main goal of this paper is to simplify the technique and to formulate it in such a way that it can easily be applied in situations other than the easiness of Ω. In general, our presentation of the technique works in any combinatory reduction system (CRS, cf. Klop, 1980) in which lambda abstraction and β-reduction exist.

In Section 3 we introduce the notion of *proof-replaceability*. The importance of this notion is that it makes the essential structure of the Jacopini technique more explicit and easier to understand. As a consequence, it becomes easier to apply the technique in various situations. We illustrate this with some examples (Section 4). The intention of Sections 3 and 4 is mainly to simplify and to illustrate an existing technique.

In Section 5 we describe an application of this technique to a λ-calculus which is extended with a μ-abstractor. This leads to a new result stating that (under fairly simple conditions) it is consistent to extend this calculus with an equation of a certain form. This result seems to be rather general: the easiness of Ω is an immediate corollary of it, and it leads to a short and elegant proof of the (well-known) fact that Curry's and Turing's fixed point combinator can be consistently identified.

An alternative formulation of the main result of Section 5 uses Curry's fixed point combinator. It is an open problem whether this result can also be formulated using Turing's fixed point combinator.

2. PRELIMINARIES

We analyse the Jacopini technique for a more general situation than just untyped lambda calculus. We only assume that a calculus fulfills the following minimal properties:

— it is a lambda calculus, i.e., lambda abstraction, application, and β-reduction are available in the calculus,

— the reduction relation is compatible, i.e., if $M \to N$, then $C[M] \to C[N]$ for every context $C[\_]$ (cf. Barendregt, 1984, Sect. 3.1.1),

— of course, we also assume consistency, i.e., not all terms are convertible to each other,

— strictly speaking, to prove the main theorem (3.5), the reduction relation of the given calculus need not have the Church–Rosser property. However, in order to carry out certain constructions in applications of the Jacopini technique, we will assume that the Church–Rosser property does hold (see Sections 4 and 5).

Apart from these minimal assumptions there is a lot of freedom: the calculus may be typed or untyped, it may or may not contain constants, there may or may not be other abstractors (e.g., the μ-abstractor, see Section 4), etc.

Standard Property. As is well known, a calculus which fulfills the above requirements can also be formulated as a theory $\boldsymbol{\lambda}$, such that the property

$$M =_\lambda N \Leftrightarrow \boldsymbol{\lambda} \vdash M = N,$$

holds, where $=_\lambda$ is the reflexive, symmetric, and transitive closure of $\to$.

Clearly, the axioms of $\boldsymbol{\lambda}$ are all equations $R = L$ whenever $R \to L$ is a reduction rule of the calculus. The derivation rules of $\boldsymbol{\lambda}$ are also well known.

$\boldsymbol{\lambda} \vdash M = N$	$\Rightarrow \boldsymbol{\lambda} \vdash N = M$	Symmetry Rule
$\boldsymbol{\lambda} \vdash M = L, L = N$	$\Rightarrow \boldsymbol{\lambda} \vdash M = N$	Transitivity Rule
$\boldsymbol{\lambda} \vdash M = N$	$\Rightarrow \boldsymbol{\lambda} \vdash C[M] = C[N]$	Compatibility Rules

In fact, the compatibility rule is a scheme of rules: for every context $C[\ ]$ we get a concrete rule. Clearly, these contexts may be restricted to *elementary* contexts, i.e., to contexts which arise from the term formation rules. For example, for the untyped lambda calculus these elementary contexts are $X[\ ]$, $[\ ]X$, $\lambda x.[\ ]$, where X is an arbitrary term.

3. THE JACOPINI TECHNIQUE

As already stated in the introduction, the Jacopini technique is a technique to examine whether the extension of a given calculus $\boldsymbol{\lambda}$ with a set of equations is consistent (if, of course, $\boldsymbol{\lambda}$ itself is consistent). In this section we describe this technique. We introduce the notion of *proof-replaceability*, by which the technique is simplified considerably.

Notation. We write $\mathbf{P}$ for a sequence of terms $P_1, ..., P_n$. We use $F\mathbf{P}$ as a shorthand notation for $FP_1P_2 \cdots P_n$, with function application associative to the left, as usual. We write $\mathbf{P} = \mathbf{Q}$ for the sequence of equations

$$P_1 = Q_1, ..., P_n = Q_n.$$

We denote an extension of $\boldsymbol{\lambda}$ with the equations $\mathbf{P} = \mathbf{Q}$ as $\boldsymbol{\lambda} + \mathbf{P} = \mathbf{Q}$ and the consistency of this extension by $\mathbf{Con}(\mathbf{P} = \mathbf{Q})$. We assume that all P_i, Q_i in $\mathbf{P}$, $\mathbf{Q}$ are closed.

We start with an informal introduction of the technique. In order to prove $\mathbf{Con}(\mathbf{P} = \mathbf{Q})$, the Jacopini technique proceeds by contraposition, i.e., it assumes

$$\boldsymbol{\lambda} + \mathbf{P} = \mathbf{Q} \vdash M = N$$

for some critical M, N. For example, in the untyped lambda calculus, $M \equiv \mathbf{K}$ and $N \equiv \mathbf{S}$.

The technique then tries to eliminate the applications of the equations $P_i = Q_i$ in this proof. If this elimination succeeds, the result is

$$\boldsymbol{\lambda} \vdash M = N,$$

i.e., $\boldsymbol{\lambda}$ is inconsistent. The main result of this section is the formulation of two sufficient properties such that the elimination indeed succeeds (see Lemma 3.4).

Suppose that in some proof of $M = N$ in the extended calculus $\boldsymbol{\lambda} + \mathbf{P} = \mathbf{Q}$ an equation from $\mathbf{P} = \mathbf{Q}$ is used m times. Then this proof has the informal structure

$$M = \cdots = X_1 = Y_1 = \cdots = X_j = Y_j = \cdots = X_m = Y_m = \cdots = N, \qquad (1)$$

where the displayed equalities $X_j = Y_j$ correspond to the applications of an equation $P_i = Q_i$ from $\mathbf{P} = \mathbf{Q}$. That is to say, for each $j = 1, ..., m$ there are a context $C_j[\ ]$ and an i, with $i = 1, ..., n$, such that

$$X_j \equiv C_j[P_i] \qquad \text{and} \qquad Y_j \equiv C_j[Q_i],$$

or

$$X_j \equiv C_j[Q_i] \qquad \text{and} \qquad Y_j \equiv C_j[P_i],$$

depending on whether the equation $P_i = Q_i$ is applied from left to right or from right to left. Since all P_i and Q_i are closed, we can abstract away from the direction in which $P_i = Q_i$ is used, and also from *which* equation from $\mathbf{P} = \mathbf{Q}$ is used. This is achieved by letting F_j be one of the following two terms:

$$\lambda x_1 \cdots x_n y_1 \cdots y_n . C_j[x_i],$$

$$\lambda x_1 \cdots x_n y_1 \cdots y_n . C_j[y_i].$$

Hence, we may reformulate (1) as

$$M = \cdots = F_1 \mathbf{PQ} = F_1 \mathbf{QP} = \cdots = F_m \mathbf{PQ} = F_m \mathbf{QP} = \cdots = N. \qquad (2)$$

All other equalities in (2), i.e., all equalities not of the form $F_j\mathbf{PQ} = F_j\mathbf{QP}$, are proved in $\boldsymbol{\lambda}$ itself, i.e., without using an equation from $\mathbf{P} = \mathbf{Q}$ (for a formal presentation of this part of the technique, see Lemma 3.1).

The next part of the Jacopini technique consists of the elimination of the applications of the equations $\mathbf{P} = \mathbf{Q}$ from the proof of $M = N$. Replace in (2) $\mathbf{P}$ by $\mathbf{Q}$ in all equations of the form $F_j\mathbf{PQ} = F_j\mathbf{QP}$. This yields

$$M \cdots F_1\mathbf{QQ} = F_1\mathbf{QQ} \cdots F_m\mathbf{QQ} = F_m\mathbf{QQ} \cdots N. \tag{3}$$

In general, this is not a proof of $M = N$ any more. However, if the following conditions are satisfied,

— for all i, with $i = 1, \ldots, n$, P_i is *operationally less defined* than Q_i (Definition 3.2),

— $\mathbf{P}$ is *proof-replaceable* by $\mathbf{Q}$ (Definition 3.3),

then from (3) we can construct a proof of $M = N$ in which the equations $\mathbf{P} = \mathbf{Q}$ are used $m - 1$ times:

$$M = \cdots = G_1\mathbf{PQ} = G_1\mathbf{QP} = \cdots = G_{m-1}\mathbf{PQ} = G_{m-1}\mathbf{QP} = \cdots = N.$$

Repeating this process, all applications of the equations $\mathbf{P} = \mathbf{Q}$ can be eliminated from the proof of $M = N$.

Now we come to the formalisation of this line of reasoning.

LEMMA 3.1 (Jacopini, 1975). *Let* $\mathbf{P} = \mathbf{Q}$ *be a sequence of closed equations* $P_i = Q_i$, *with* $i = 1, \ldots, n$. *Then*

$$\boldsymbol{\lambda} + \mathbf{P} = \mathbf{Q} \vdash M = N$$

if and only if there exist $F_1, \ldots, F_m$, $m \geqslant 0$, *such that*

$$\begin{aligned} \boldsymbol{\lambda} &\vdash & M &= F_1\mathbf{PQ}, \\ \boldsymbol{\lambda} &\vdash & F_j\mathbf{QP} &= F_{j+1}\mathbf{PQ} \qquad (j = 1, \ldots, m-1), \\ \boldsymbol{\lambda} &\vdash & F_m\mathbf{QP} &= N. \end{aligned}$$

Remark. In case that $m = 0$, the right hand side of the "if and only if" is to be read as $\boldsymbol{\lambda} \vdash M = N$.

Proof. "$\Leftarrow$": Immediate, since $\boldsymbol{\lambda} + \mathbf{P} = \mathbf{Q} \vdash F_j\mathbf{PQ} = F_j\mathbf{QP}$, for all j. "$\Rightarrow$": By induction on the length of the proof of $\boldsymbol{\lambda} + \mathbf{P} = \mathbf{Q} \vdash M = N$.

Basic case. $M = N$ is an axiom. If $M = N$ is one of the axioms of $\boldsymbol{\lambda}$, we may take $m = 0$, since $\boldsymbol{\lambda} \vdash M = N$. If $M = N$ is one of the equations from $\mathbf{P} = \mathbf{Q}$, say $P_i = Q_i$, then take $m = 1$ and $F_1 \equiv \lambda\mathbf{xy}.x_i$ (here too, $\mathbf{x}$, $\mathbf{y}$ are sequences of variables of the appropriate length).

Induction case. There are three main cases to distinguish, depending on the last rule applied in the proof of $\boldsymbol{\lambda} + \mathbf{P} = \mathbf{Q} \vdash M = N$.

— If the last rule is the symmetry rule, then we may apply the induction hypothesis on $\boldsymbol{\lambda} + \mathbf{P} = \mathbf{Q} \vdash N = M$. Hence, there are $F_1, ..., F_m$ such that in $\boldsymbol{\lambda}$ we have

$$\begin{aligned} N &= F_1 \mathbf{PQ} \\ F_j \mathbf{QP} &= F_{j+1} \mathbf{PQ} \qquad (j = 1, ..., m-1), \\ F_m \mathbf{QP} &= M. \end{aligned}$$

Define $F'_k \equiv \lambda \mathbf{xy}.F_j \mathbf{yx}$, where $j = m + 1 - k$. It is immediately clear that these F_k's do the job.

— If the last rule is the transitivity rule, then there is a term L, such that $\boldsymbol{\lambda} + \mathbf{P} = \mathbf{Q} \vdash M = L,\ L = N$. The result follows immediately from the induction hypothesis.

— If the last rule is one of the compatibility rules, then there are terms M', N', and a context $C[\ ]$, such that $M \equiv C[M']$, $N \equiv C[N']$, and $\boldsymbol{\lambda} + \mathbf{P} = \mathbf{Q} \vdash M' = N'$. By the induction hypothesis there are $F_1, ..., F_m$ such that in $\boldsymbol{\lambda}$ we have

$$\begin{aligned} M' &= F_1 \mathbf{PQ}, \\ F_j \mathbf{QP} &= F_{j+1} \mathbf{PQ} \qquad (j = 1, ..., m-1), \\ F_m \mathbf{QP} &= N'. \end{aligned}$$

Define $F'_j \equiv \lambda \mathbf{xy}.C[F_j \mathbf{xy}]$. The result follows immediately. ∎

The next definition comes from (Plotkin, 1977, Berry *et al.* 1985).

DEFINITION 3.2 (Operationally less defined). A term P is *operationally less defined* then Q, if for each term F, whenever FP has a normal form, then FQ has the same normal form.

DEFINITION 3.3 (Proof-Replaceability). Let $\mathbf{P}$, $\mathbf{Q}$ be sequences of closed terms, and let both sequences be of equal length.

We say that $\mathbf{P}$ is *proof-replaceable by* $\mathbf{Q}$, if for all F, F' for which

$$\boldsymbol{\lambda} \vdash F\mathbf{P} = F'\mathbf{P},$$

there exists a G such that

$$\begin{aligned} \boldsymbol{\lambda} &\vdash G\mathbf{PQ} = F\mathbf{Q}, \\ \boldsymbol{\lambda} &\vdash G\mathbf{QP} = F'\mathbf{Q}. \end{aligned}$$

LEMMA 3.4. *Let* $\mathbf{P}$, $\mathbf{Q}$ *be two sequences of closed terms of length n, and let* M, N *be in normal form. If* P_i *is operationally less defined than* Q_i (*for all* $i=1, \ldots, n$), *and* $\mathbf{P}$ *is proof-replaceable by* $\mathbf{Q}$, *then*

$$\boldsymbol{\lambda} + \mathbf{P} = \mathbf{Q} \vdash M = N \Rightarrow \boldsymbol{\lambda} \vdash M = N.$$

Proof. By assumption

$$\boldsymbol{\lambda} + \mathbf{P} = \mathbf{Q} \vdash M = N.$$

Hence, by Lemma 3.1, there are $F_1, \ldots, F_m$, $m \geqslant 0$, such that in $\boldsymbol{\lambda}$ we have

$$\left.\begin{aligned} M &= F_1\mathbf{PQ} \\ F_j\mathbf{QP} &= F_{j+1}\mathbf{PQ} \qquad (j=1, \ldots, m-1) \\ F_m\mathbf{QP} &= N \end{aligned}\right\} (*)$$

By induction on m we show that from this it follows that $\boldsymbol{\lambda} \vdash M = N$.

Basic cases. If $m=0$, then immediately $\boldsymbol{\lambda} \vdash M = N$ (see the remark after Lemma 3.1).

If $m=1$, then there is an F_1 such that in $\boldsymbol{\lambda}$ we have

$$\begin{aligned} M &= F_1\mathbf{PQ}, \\ F_1\mathbf{QP} &= N. \end{aligned}$$

Since for each i, P_i is operationally less defined than Q_i, and M, N are in normal form, it follows by Definition 3.2 that in $\boldsymbol{\lambda}$,

$$\begin{aligned} M &= F_1\mathbf{QQ}, \\ F_1\mathbf{QQ} &= N. \end{aligned}$$

Hence,

$$\boldsymbol{\lambda} \vdash M = N.$$

Induction case. Define for $j=1, \ldots, m$ ($\mathbf{x}$ is a sequence of fresh variables):

$$\begin{aligned} F'_j &\equiv \lambda\mathbf{x}.F_j\mathbf{Qx}, \\ F''_j &\equiv \lambda\mathbf{x}.F_j\mathbf{xQ}. \end{aligned}$$

Then for $j=1, \ldots, m-1$ we have (in $\boldsymbol{\lambda}$)

$$\begin{aligned} F'_j\mathbf{P} &= F_j\mathbf{QP} \\ &= F_{j+1}\mathbf{PQ} \\ &= F''_{j+1}\mathbf{P}. \end{aligned}$$

Since $\mathbf{P}$ is proof-replaceable by $\mathbf{Q}$, there are $G_1, \ldots, G_{m-1}$ such that

$$G_j\mathbf{PQ} = F'_j\mathbf{Q} = F_j\mathbf{QQ},$$
$$G_j\mathbf{QP} = F''_{j+1}\mathbf{Q} = F_{j+1}\mathbf{QQ}.$$

By Definition 3.2 again it follows that

$$F_1\mathbf{QQ} = M,$$
$$F_m\mathbf{QQ} = N.$$

Hence,

$$M = F_1\mathbf{QQ} = G_1\mathbf{PQ}$$
$$G_j\mathbf{QP} = F_{j+1}\mathbf{QQ} = G_{j+1}\mathbf{PQ} \qquad (j=1, \ldots, m-2)$$
$$G_{m-1}\mathbf{QP} = F_m\mathbf{QQ} = N.$$

By the induction hypothesis it follows that $\boldsymbol{\lambda} \vdash M=N$. ▮

The main theorem of this section is a simple consequence of Lemma 3.4.

THEOREM 3.5. *Let* $\mathbf{P}, \mathbf{Q}$ *be sequences of closed terms of length n. If* P_i *is operationally less defined than* Q_i *(for all* $i=1, \ldots, n$*), and* $\mathbf{P}$ *is proof-replaceable by* $\mathbf{Q}$*, then* $\mathbf{Con}(\mathbf{P}=\mathbf{Q})$.

Proof. By contraposition. Suppose $\neg\mathbf{Con}(\mathbf{P}=\mathbf{Q})$, i.e., for all M, N,

$$\boldsymbol{\lambda} + \mathbf{P}=\mathbf{Q} \vdash M=N.$$

By Lemma 3.4 it follows that $\boldsymbol{\lambda}$ is inconsistent. ▮

4. EXAMPLES

In this section we give some examples of the use of the Jacopini technique. All examples deal with the question whether it is consistent to extend a given lambda calculus with one or more equations $P_i = Q_i$, where P_i, Q_i are closed. The main reason for these examples is not to present new results, but to illustrate the simplicity and the generality of our approach. In particular, the examples show how to construct a G for arbitrary F, F'.

In Section 5 we do present a new and general result following from an application of the Jacopini technique to the $\boldsymbol{\lambda\mu}$-calculus. It turns out that the first two examples given below are special cases of this result.

We start with a general and informal description of the constructions needed to construct the above mentioned G. This description may be considered as a blueprint for the concrete examples.

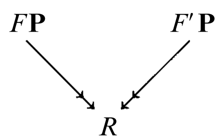
So suppose λ is some lambda calculus. We assume that the Church–Rosser property holds for λ. Now assume that a set of equations $\mathbf{P}=\mathbf{Q}$ is added to λ. We have to show that

$$\lambda + \mathbf{P} = \mathbf{Q}$$

is consistent. By the results of the previous section it is sufficient to prove the following two properties:

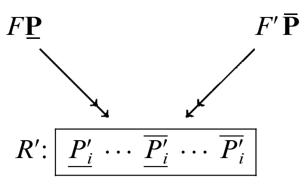
(a) P_i is *operationally less defined* than Q_i (for all i). That is to say, if FP_i has a normal form N, then $FQ_i = N$. By the Church–Rosser property we have that $FP_i \twoheadrightarrow N$. The obvious technique to prove that $FQ_i = N$ is to mark P_i in FP_i (say by underlining or by labelling) and to keep track of marked terms during the reduction. Then one needs some reduction rules to govern the behaviour of marked terms during the reduction. Which rules one has to choose depends on the specific terms P_i and Q_i. Sometimes one needs rules to extend markings to larger terms and sometimes one needs rules to shrink markings to smaller terms. This will become more clear in the examples.

(b) $\mathbf{P}$ is *proof-replaceable* by $\mathbf{Q}$, i.e., if $F\mathbf{P} = F'\mathbf{P}$, then there is a G such that $G\mathbf{PQ} = F\mathbf{Q}$ and $G\mathbf{QP} = F'\mathbf{Q}$. In order to prove proof-replaceability, we have to construct G. By the Church–Rosser property, $F\mathbf{P}$ and $F'\mathbf{P}$ have a common reduct R:



We mark all P_i in $F\mathbf{P}$ and in $F'\mathbf{P}$ (say by underlining and overlining, respectively). As before, how to handle the marked terms during the reduction depends on $\mathbf{P}$ and $\mathbf{Q}$.

Suppose that marked reductions lead to the following result (R' is the same term as R, except for the markings; in fact, R' is the combination of *two* marked variants of R, since underlinings only occur in the left reduction, whereas overlinings only occur in the right reduction):



We assume that the reduction rules for marked terms are such that inside R' every marked term corresponds (in one way or another) to one of the terms in the sequence $\mathbf{P}$. In the diagram only marked terms P'_i, corresponding to P_i, are displayed (not all terms P'_i need to be identical).

Furthermore, we assume that inside R' marked terms do not contain other marked terms as proper subterms, i.e., we assume that all marked terms P'_i inside R' are underlined, overlined, *or* both.

If these two conditions are fulfilled, it is possible to define (all x_i, y_i are fresh variables):

$$G \equiv \lambda x_i \cdots x_n\, y_i \cdots y_n . \boxed{y_i \cdots Q_i \cdots x_i}\, .$$

In this definition of G it is understood that the box represents the same term as the box in R', except that *under*lined terms P'_i are replaced by y_i, whereas *over*lined terms P'_i are replaced by x_i. All terms P'_i that are *both* underlined and overlined are replaced by Q_i.

To complete the proof, we have to show that (again, the boxes represent the same term with the indicated substitutions)

$$F\mathbf{Q} = \boxed{Q_i \;\cdots\; Q_i \;\cdots\; P_i} \qquad (= G\mathbf{PQ}),$$
$$F'\mathbf{Q} = \boxed{P_i \;\cdots\; Q_i \;\cdots\; Q_i} \qquad (= G\mathbf{QP}).$$

The obvious way to do this is to replace (in the marked reductions above) every marked term corresponding to P_i by Q_i ($i = 1, ..., n$), and to proceed by induction on the length of the marked reductions.

In the first example below we prove the easiness of Ω (see Jacopini, 1975, Baeten and Boerboom, 1979). The second example shows that adding $\mathbf{YB} = \mathbf{I}$ to the untyped lambda calculus with the η-rule is consistent ($\mathbf{B}$ is the combinator for function composition). This question was mentioned to me by Professor Jacopini (Jacopini, 1994). The third example is described in (Intrigila, 1991) and defines a term P such that $P(\Omega\Omega)$ is easy but $P\Omega$ is not. The first two examples extend the lambda calculus with a single equation; the third example adds two equations.

Our treatment of the examples will be somewhat informal but sufficiently precise so that the reader will be able to fill in the details.

In Section 5 we give an application of the Jacopini technique to the $\lambda\mu$-calculus, which leads to a general result about the consistency of extensions of this calculus with equations of a certain form. It turns out that the first two examples below are special cases of this general result.

EXAMPLE 1. *Ω is easy.*

We have to prove that $\lambda + \Omega = Q$ is consistent for every closed term Q. Here, λ need not be restricted to the untyped lambda calculus. For example, it can also be PCF, if we take $\mathbf{YI}$ for Ω.

Proof of Property (a). $F\Omega \twoheadrightarrow N$, where N is a normal form. Since Ω is unsolvable, it follows that $FQ \twoheadrightarrow N$ for every term Q (by the Genericity Lemma, cf. Barendregt, 1984, Sect. 14.3.24).

Proof of Property (b). Assume that $F\Omega = F'\Omega$, and let R be the common reduct:

$$F\Omega \twoheadrightarrow R \twoheadleftarrow F'\Omega.$$

In this case the marking rules can be very simple: there are no rules concerning extension or shrinking of markings, and marked terms may be reduced internally (i.e., if $A \to B$, then $\underline{A} \to \underline{B}$ and $\overline{A} \to \overline{B}$). Now it is easy to see that

$$\begin{array}{ccc} F\underline{\Omega} & & F'\overline{\Omega} \\ & \searrow\!\!\!\searrow \quad \swarrow\!\!\!\swarrow & \\ \multicolumn{3}{c}{R' : \boxed{\underline{\Omega} \cdots \underline{\overline{\Omega}} \cdots \overline{\Omega}}} \end{array}$$

By induction on the length of the reductions it now follows that

$$F\mathbf{Q} \twoheadrightarrow \boxed{Q \cdots Q \cdots \Omega},$$

$$F'\mathbf{Q} \twoheadrightarrow \boxed{\Omega \cdots Q \cdots \Omega}.$$

Now define G as follows (see above):

$$G \equiv \lambda xy.\boxed{y \cdots Q \cdots x}.$$

Then clearly,

$$G\Omega Q = FQ,$$

$$GQ\Omega = F'Q,$$

which was to be proved. ∎

In Section 5 we show that in $\boldsymbol{\lambda\mu}$ the easiness of Ω is just a special case of Theorem 5.3.

Example 2. $\mathbf{YB} = \mathbf{I}$.

Let $\boldsymbol{\lambda}$ be the untyped lambda calculus with the η-rule. We show that

$$\boldsymbol{\lambda} + \mathbf{YB} = \mathbf{I}$$

is consistent, where $\mathbf{B} \equiv \lambda xyz.x(yz)$ is the combinator for function composition, and $\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ is Curry's fixed point combinator.

Proof. First notice that

$$\mathbf{YB} = (\lambda xyz.xx(yz))(\lambda xyz.xx(yz)).$$

We denote the right hand side of this equation by P and prove $\mathbf{Con}(P = \mathbf{I})$. The only possible reduction inside P is

$$P \to \lambda yz.P(yz).$$

Second, notice that by $\beta\eta$-reduction we have

$$\lambda yz.\mathbf{I}(yz) = \mathbf{I}. \tag{4}$$

Proof of Property (a). Suppose $FP \twoheadrightarrow N$, where N is a normal form. Mark P, say by underlining, and add the rule

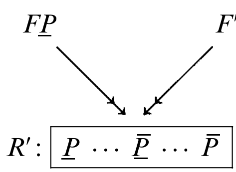
$$\underline{P} \rightarrow \lambda yz.\underline{P}(yz).$$

Clearly, we still have $F\underline{P} \twoheadrightarrow N$. All underlined terms occurring in this reduction are of the form $\underline{P}$. Furthermore, N is a normal form, hence N does not contain underlined subterms. By induction on the length of the reduction it is now easy to show that $F\mathbf{I} = N$ (use Eq. (4) above).

Proof of Property (b). Suppose $FP = F'P$. By the Church–Rosser property there is a common reduct, say R:

$$FP \twoheadrightarrow R \twoheadleftarrow F'P.$$

As before, it is easy to see that we have



Again, G can be defined as

$$G \equiv \lambda xy.\boxed{y \cdots \mathbf{I} \cdots x}\ .$$

We leave it to the reader to verify that

$$\begin{aligned} GP\mathbf{I} &= F\mathbf{I}, \\ G\mathbf{I}P &= F'\mathbf{I}. \quad \blacksquare \end{aligned}$$

Example 2 also is a special case of the general theorem to be proved in Section 5 (see Theorem 5.4).

EXAMPLE 3. An example of Intrigila.

Both examples above add a single equation to the lambda calculus. The following example is given by Intrigila (cf. Intrigila, 1991) and is interesting because it adds *two* equations at the same time to the untyped lambda calculus.

Consider the following term P:

$$\begin{aligned} V &\equiv \lambda xy.xx\Omega(xxy), \\ P &\equiv VV. \end{aligned}$$

The main subject of Intrigila's paper is to show that $P(\Omega\Omega)$ is easy and to contrast this with the fact that $P\Omega$ is *not* easy (take $P\Omega = \mathbf{K}$. This implies $\mathbf{K} = \mathbf{KK}$). In order to prove the easiness of $P(\Omega\Omega)$, Intrigila needs three different extensions of $\beta\eta$-reduction, all of which must have the Church–Rosser property.

We use the Jacopini technique to show the easiness of $P(\Omega\Omega)$. In fact, we prove the stronger statement

$$\boldsymbol{\lambda} + (P\Omega = \mathbf{I}) + (P(\Omega\Omega) = M) \text{ is consistent,}$$

where $\boldsymbol{\lambda}$ is the untyped lambda calculus and M is an arbitrary closed term.

By virtue of Theorem 3.5 it is sufficient to prove that

(i) $P\Omega$, $P(\Omega\Omega)$ are operationally less defined than $\mathbf{I}$, M, respectively,

(ii) $\langle P\Omega, P(\Omega\Omega)\rangle$ is proof replaceable by $\langle \mathbf{I}, M\rangle$.

The proof of (i) immediately follows from the fact that both $P\Omega$ and $P(\Omega\Omega)$ are unsolvable (both have no head normal form), i.e., they are operationally less defined than any term.

It remains to show (ii). Thus, suppose

$$F(P\Omega)(P(\Omega\Omega)) = F'(P\Omega)(P(\Omega\Omega)).$$

We have to construct a term G such that

$$G(P\Omega)(P(\Omega\Omega))\,\mathbf{I}M = F\mathbf{I}M,$$

$$G\mathbf{I}M(P\Omega)(P(\Omega\Omega)) = F'\mathbf{I}M.$$

As before, there is a common reduct R:

$$F(P\Omega)(P(\Omega\Omega)) \twoheadrightarrow R \twoheadleftarrow F'(P\Omega)(P(\Omega\Omega)).$$

First we bring these reductions into some desired form.

Notice that a typical reduction of a term of the form PX (with X closed) is

$$\begin{aligned} PX &\to (\lambda y.P\Omega(Py))X \\ &\to P\Omega(PX). \end{aligned} \qquad (*)$$

Now reduce all β-redexes inside R which begin with λy (we assume that λy only occurs in P and not in F, F', Ω). This leads to a new common reduct R'. By the Standardization Theorem it follows that there are standard reductions

$$F(P\Omega)(P(\Omega\Omega)) \twoheadrightarrow R' \twoheadleftarrow F'(P\Omega)(P(\Omega\Omega)),$$

i.e., any contraction of P in these reductions is immediately followed by the second step as in $(*)$.

Next, we mark the terms $P\Omega$ and $P(\Omega\Omega)$ inside these standard reductions. As before, we use underlining for the left reduction and overlining for the right reduction. So the marked left and right reductions start with (respectively)

$$F(\underline{P\Omega})(\underline{P(\Omega\Omega)}), \qquad F'(\overline{P\Omega})(\overline{P(\Omega\Omega)}).$$

Reductions inside marked terms are allowed. We add the rules

$$\begin{aligned} \underline{P\Omega(PX)} &\to \underline{P\Omega}(\underline{PX}), \\ \overline{P\Omega(PX)} &\to \overline{P\Omega}(\overline{PX}), \end{aligned}$$

i.e., markings can be split up. These splitting rules are applied immediately after each reduction pattern of the form $(*)$.

Now it is easy to see that the marked reductions can be such that all marked terms inside R'' (the marked variant of R') are of the form $P\Omega$ or $P(\Omega\Omega)$, and they are underlined, overlined, *or* both. Suppose R'' is of the form

$$\boxed{\cdots\ \underline{P\Omega}\ \cdots\ \underline{\overline{P\Omega}}\ \cdots\ \overline{P\Omega}\ \cdots\ \underline{P(\Omega\Omega)}\ \cdots\ \underline{\overline{P(\Omega\Omega)}}\ \cdots\ \overline{P(\Omega\Omega)}\ \cdots}\ ;$$

then we can define G as

$$G \equiv \lambda uvxy.\ \boxed{\cdots\ x\ \cdots\ \mathbf{I}\ \cdots\ u\ \cdots\ y\ \cdots\ M\ \cdots\ v\ \cdots}\ .$$

In order to show that G has the required properties, replace in the reductions all terms of the form $\underline{P\Omega}$, $\overline{P\Omega}$ by $\mathbf{I}$, and all terms of the form $\underline{P(\Omega\Omega)}$, $\overline{P(\Omega\Omega)}$ by M. In case of a reduction sequence as in $(*)$, the replacement of the intermediate terms is skipped. For example, in such a case, we replace

$$\begin{aligned} \underline{P(\Omega\Omega)} &\to \underline{(\lambda y.\, P\Omega(Py))(\Omega\Omega)} \\ &\to \underline{P\Omega(P(\Omega\Omega))} \\ &\to \underline{P\Omega}(\underline{P(\Omega\Omega)}) \end{aligned}$$

by

$$M = \mathbf{I}M.$$

The proof is completed by induction on the length of the reductions. ∎

5. AN APPLICATION TO λμ

In this section we consider calculi which contain μ-abstraction in addition to λ-abstraction; i.e., there are terms of the form $\mu x.M$. The corresponding reduction rule is

$$\mu x.M \to M[x := \mu x.M]$$

(where $A[x := B]$ denotes substitution of B for the free occurrences of x in A; we will assume that clashes of variables do not arise).

Unlike the situation in the previous sections, we now restrict ourselves to untyped or simply typed calculi, possibly including the η-rule, and possibly extended with constants and δ-rules for natural numbers and booleans. Without going into details, the reason for this is that $\boldsymbol{\lambda\mu}$ must be weakly non-ambiguous (cf. Klop, 1980; Van Raamsdonk, 1992), and $\boldsymbol{\lambda\mu}$ may not contain reduction rules which produce new μ-terms.

An example of a calculus for which the results of this section hold is a PCF-like calculus, in which fixed point combinators are replaced by μ-terms. Clearly, from a computational point of view this difference is inessential, since $\mu x.M$ may be considered as shorthand notation for $\mathbf{Y}(\lambda x.M)$. However, we feel that μ has certain advantages, one of them being the possibility to define the notion of *μ-hierarchicalness* (see Definition 5.1). This notion is very convenient to formulate an application of the Jacopini technique to $\boldsymbol{\lambda\mu}$.

The main result of this section is Theorem 5.3 which asserts that (under certain conditions) extending $\boldsymbol{\lambda\mu}$ with an equation of the form $\mu x.M = N$ is consistent. This result is general in the sense that many examples are special cases of it (e.g., Examples 1 and 2 of Section 4). An equivalent formulation of this result uses Curry's fixed point combinator. It is an open problem whether the result also holds for Turing's fixed point combinator.

DEFINITION 5.1 (μ-hierarchical). A $\boldsymbol{\lambda\mu}$-term N is *μ-hierarchical*, if all subterms inside N of the form $\mu x.M$, are closed. ∎

EXAMPLES. Clearly, $\Omega \equiv \mu x.x$ is μ-hierarchical. Also, Turing's fixed point combinator $\mathbf{Y}_T$ $(\equiv \mu y.\lambda f.f(yf))$ is μ-hierarchical. On the other hand, Curry's fixed point combinator $\mathbf{Y}_C$ $(\equiv \lambda f.\mu x.fx)$ is *not* μ-hierarchical.

Defining the corresponding concept for a calculus with fixed point combinators instead of μ (e.g., PCF) would involve the requirement that every $\mathbf{Y}$ be on a function position and that in every subterm inside N of the form $\mathbf{Y}F$, F be closed. For the untyped lambda calculus a fixed point combinator itself may only arise after a sequence of β-reductions, which makes the corresponding concept for the untyped lambda calculus undecidable.

We have the following lemma.

LEMMA 5.2. *Let N be closed. If $\mu x.M$ is μ-hierarchical, and $\boldsymbol{\lambda\mu} \vdash M[x := N] = N$, then* (a) *$\mu x.M$ is operationally less defined than N, and* (b) *$\mu x.M$ is proof-replaceable by N.*

Proof. The proof is tedious and therefore postponed to Appendix A. Part (a) is proved by Lemma A.11, part (b) by Lemma A.12. ∎

The main result of this section is the following.

THEOREM 5.3. *Let N be closed. If $\mu x.M$ is μ-hierarchical, and* $\boldsymbol{\lambda\mu} \vdash M[x := N] = N$, *then* $\mathbf{Con}(\mu x.M = N)$.

Proof. By Lemma 5.2 and Theorem 3.5. ∎

The easiness of Ω (see Example 1, Section 4) is an immediate consequence of Theorem 5.3 (remember that in $\boldsymbol{\lambda\mu}$ $\Omega \equiv \mu x.x$).

An alternative formulation of Theorem 5.3 is the following.

THEOREM 5.4. *Let N be closed. If F is μ-hierarchical, and* $\boldsymbol{\lambda\mu} \vdash FN = N$, *then* $\mathbf{Con}(\mathbf{Y}_C F = N)$.

Proof. Easy. Remember that $\mathbf{Y}_C \equiv \lambda f.\mu x.fx$. ∎

It is an immediate consequence of Theorem 5.4 that in $\boldsymbol{\lambda\mu}$ with η we have $\mathbf{Con}(\mathbf{Y}_C \mathbf{B} = \mathbf{I})$ (see Example 2, Section 4).

In this particular case we also have $\mathbf{Con}(\mathbf{Y}_T \mathbf{B} = \mathbf{I})$, where $\mathbf{Y}_T \equiv \mu y.\lambda f.f(yf)$ is Turing's fixed point combinator (left to the reader). However, in general it is an open problem whether Theorem 5.4 also holds for $\mathbf{Y}_T$.

An elegant application of Theorem 5.4 is the following.

COROLLARY 5.5. $\mathbf{Con}(\mathbf{Y}_C = \mathbf{Y}_T)$.

Proof. Define $G \equiv \lambda yf.f(yf)$. Then $G\mathbf{Y}_C = \mathbf{Y}_C$ and $\mathbf{Y}_C G = \mathbf{Y}_T$ (compare Barendregt, 1984, Sect. 6.5.3–6.5.5, see also Klop, 1980).

Hence $\mathbf{Con}(\mathbf{Y}_C G = \mathbf{Y}_C)$, i.e., $\mathbf{Con}(\mathbf{Y}_C = \mathbf{Y}_T)$. ∎

APPENDIX: PROOF OF LEMMA 5.2

In this appendix we give the detailed proof of Lemma 5.2. The proof is given for a simply typed calculus with λ- and μ-abstraction, and with constants for natural numbers and booleans, for the successor function, predecessor function, test for zero, and for conditionals. We denote this calculus with $\boldsymbol{\lambda\mu}$. Clearly, $\boldsymbol{\lambda\mu}$ is computationally equivalent to PCF.

As can be seen from the proofs of Examples 1 and 2 (see Section 4), we have to mark subterms in order to keep track of these subterms during a reduction. We choose for a labelling technique to add three labels 1, 2, and 12 to $\boldsymbol{\lambda\mu}$. Then in proving proof-replaceability we can distinguish between terms that come from the left, terms that come from the right, and terms that come from the left and right. For the proof of the other condition (operationally less defined) one label would be sufficient. For the intuition of this, we refer the reader to Examples 1 and 2 in Section 4.

More formally, we have the labelled terms

$$\mu_1 x.A, \qquad \mu_2 x.A, \qquad \mu_{12} x.A,$$

called μ_1-terms, μ_2-terms, μ_{12}-terms, respectively. Substitution is as in $\boldsymbol{\lambda\mu}$; just leave the labels unchanged. In addition to the rules of $\boldsymbol{\lambda\mu}$ there is the rule

$$\mu_a x.A \to A[x := \mu_a x.A] \qquad (a \in \{1, 2, 12\}).$$

Reduction and conversion in this language, called $\boldsymbol{\lambda\mu}_\ell$, are denoted by $\to_\ell$, $\twoheadrightarrow_\ell$, $=_\ell$.

Notation. If A is a subterm of B, we will write $A \subseteq B$.

By A^{-1} we denote a term A from which all labels 1 are removed. Notice, however, that this also removes the 1 from label 12, so, e.g., $(\mu_{12} x.A)^{-1} \equiv \mu_2 x.\ A$. Likewise for A^{-2}. If *all* labels are removed, we write A^{-12}.

We will write $A \cong B$ if A and B are identical except for the labels, i.e., if $A^{-12} \equiv B^{-12}$.

The notation $A|_N^a$ ($a \in \{1, 2, 12\}$) denotes the term obtained from A by replacing all μ_a-terms by N (where N is label free).

Notice that in the notation A^{-a} label 12 is *not* considered an independent label, but the *union* of the labels 1 and 2. On the other hand, in the notation $A|_N^a$ the labels 1, 2, 12 are considered independent labels.

LEMMA A.1. *If $A \twoheadrightarrow B$, $A' \cong A$, then there is a B' such that $B' \cong B$ and $A' \twoheadrightarrow_\ell B'$.*

Proof. Easy. ∎

DEFINITION A.2. A $\boldsymbol{\lambda\mu}_\ell$-term A is called *ℓ-disjoint* if B is label free for all $\mu_a x.B \subseteq A$ ($a \in \{1, 2, 12\}$).

So in an ℓ-disjoint term nested labellings do not occur.

LEMMA A.3. *Suppose A is ℓ-disjoint and all labeled terms $\mu_a x.A' \subseteq A$ are μ-hierarchical. If $A \twoheadrightarrow_\ell B$, then*

- *B is ℓ-disjoint,*
- *for every $\mu_a x.B' \subseteq B$ there is a $\mu_a x.A' \subseteq A$ such that $A' \twoheadrightarrow B'$,*
- *each $\mu_a x.B' \subseteq B$ is μ-hierarchical.*

Proof. By induction on the length of the reduction $A \twoheadrightarrow_\ell B$.

Basic case. The length of the reduction is 0. Immediate.

Induction case. Suppose $A \twoheadrightarrow_\ell C \to_\ell B$. By the induction hypothesis, we have that

- C is ℓ-disjoint,
- for all $\mu_a x.\ C' \subseteq C$ there is a $\mu_a x.A' \subseteq A$ such that $A' \twoheadrightarrow C'$,
- each $\mu_a x.C' \subseteq C$ is μ-hierarchical.

The proof is completed by a straightforward check of all possible reduction rules by which $C \to_{\ell} B$. ▮

In the proof of Lemma A.8 we use induction on the *μ-height* of a certain term. This term is label free, so in the next definition there needs to be no clause for labelled μ-terms.

DEFINITION A.4. The *μ-height* of a term A, denoted by $h_\mu A$, is defined inductively as follows:

$$\begin{aligned} h_\mu x &= 0 \\ h_\mu c &= 0 \\ h_\mu(AB) &= \max(h_\mu A, h_\mu B) \\ h_\mu(\lambda x.A) &= h_\mu A \\ h_\mu(\mu x.A) &= 1 + h_\mu A. \end{aligned}$$

LEMMA A.5. *If $h_\mu A \geqslant h_\mu B$, then for any context $C[\ ]$*

$$h_\mu(C[A]) \geqslant h_\mu(C[B]).$$

Proof. By induction on the structure of the context $C[\ ]$. ▮

LEMMA A.6. *If A is μ-hierarchical, then for all B*

$$h_\mu(A[x := B]) \leqslant \max(h_\mu A, h_\mu B).$$

Proof. By induction on the structure of A. By assumption, A is μ-hierarchical. Hence, if $A \equiv \mu y.A'$, then $(\mu y.A')[x := B] \equiv \mu y.A'$. All other cases are straightforward. ▮

LEMMA A.7. *Suppose A is μ-hierarchical. If $A \twoheadrightarrow B$, then $h_\mu A \geqslant h_\mu B$.*

Proof. By induction on the length of the reduction $A \twoheadrightarrow B$.

Basic case. Immediate.

Induction case. Let $A \twoheadrightarrow A' \to B$. By the induction hypothesis it follows that $h_\mu A \geqslant h_\mu A'$. We will show that $h_\mu A' \geqslant h_\mu B$. Suppose that P is the contracted redex in A', i.e., $A' \equiv C[P] \to C[Q] \equiv B$. We have three possibilities for $P \to Q$.

— $P \to Q$ by a δ-rule. Then $h_\mu P = h_\mu Q$, hence $h_\mu P \geqslant h_\mu Q$.

— $P \to Q$ by the β-rule. Then

$$\begin{aligned} h_\mu P &= h_\mu((\lambda x.X)\,Y) \\ &= \max(h_\mu X, h_\mu Y) \\ &\geqslant h_\mu(X[x := Y]) \\ &= h_\mu Q. \end{aligned}$$

— $P \to Q$ by the μ-rule. Then

$$\begin{aligned} h_\mu P &= h_\mu(\mu x.X) \\ &= \max(h_\mu X, h_\mu(\mu x.X)) \\ &\geqslant h_\mu(X[x := \mu x.X]) \\ &= h_\mu Q. \end{aligned}$$

In all cases, $h_\mu A' \geqslant h_\mu B$ follows by Lemma A.5. ∎

LEMMA A.8 (Pushout Lemma). *Suppose A is μ-hierarchical and label free. Suppose further that B is ℓ-disjoint. If*

— $A \twoheadrightarrow B^{-12}$,

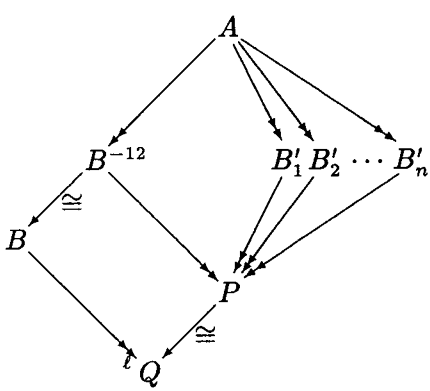
— $A \twoheadrightarrow B'$, *whenever* $\mu_a x.B' \subseteq B$,

then there is a label free term C such that $B \twoheadrightarrow_\ell C$.

Proof. By induction on $h_\mu B^{-12}$.

Basic case. $h_\mu B^{-12} = 0$. Then there are no μ-subterms in B^{-12}, i.e., B is already label free.

Induction case. Suppose B has labelled subterms $\mu_a x.B'_i$, $i = 1, ..., n$, $a \in \{1, 2, 12\}$. If $n = 0$, then B is label free, and we are done. So let $n > 0$. By assumption: $A \twoheadrightarrow B'_i$, and $A \twoheadrightarrow B^{-12}$. By the Church–Rosser property and by Lemma A.1 there are terms P, Q such that the following diagram commutes:



Clearly, $Q^{-12} \equiv P$, so $A \twoheadrightarrow Q^{-12}$. Since A is μ-hierarchical, it follows that B^{-12} is μ-hierarchical too. Thus each $\mu_a x.\ B' \subseteq B$ is μ-hierarchical. Hence, by Lemma A.3, Q is ℓ-disjoint, and for all $\mu_a x.Q' \subseteq Q$ there is a $\mu_a x.B' \subseteq B$ such that $B' \twoheadrightarrow Q'$. By assumption $A \twoheadrightarrow B'$, hence $A \twoheadrightarrow Q'$. Summarizing, Q has all properties that the present lemma assumes for B.

Next, notice that

$$\begin{aligned} h_\mu Q^{-12} &= h_\mu P \\ &\leqslant h_\mu B'_i \\ &< h_\mu B^{-12}. \end{aligned}$$

Hence, we may apply the induction hypothesis on Q; i.e., there is a label free term C such that $Q \twoheadrightarrow_\ell C$, and so $B \twoheadrightarrow_\ell C$. ∎

LEMMA A.9. *Let F, F', A be label free terms, and suppose*

$$\boldsymbol{\lambda\mu} \vdash F(\mu x.A) = F'(\mu x.A).$$

If $\mu x.A$ is μ-hierarchical, then there is an ℓ-disjoint term B such that

$$\begin{aligned} F(\mu_1 x.A) &\twoheadrightarrow_\ell B^{-2}, \\ F'(\mu_2 x.A) &\twoheadrightarrow_\ell B^{-1}. \end{aligned}$$

Proof. By the Church–Rosser property there is a term C such that

$$F(\mu x.A) \twoheadrightarrow C \twoheadleftarrow F'(\mu x.A).$$

By Lemma A.1 there are C_1, C_2, such that

$$\begin{aligned} F(\mu_1 x.A) &\twoheadrightarrow_\ell C_1, \\ F'(\mu_2 x.A) &\twoheadrightarrow_\ell C_2. \end{aligned}$$

Clearly, $C_1 \cong C \cong C_2$, and C_1, C_2 contain only label 1, 2, respectively. Now there is a term C_{12} such that C_{12} is the "join" of C_1 and C_2, i.e., $C_{12}^{-2} \equiv C_1$ and $C_{12}^{-1} \equiv C_2$. Let $\mu_a x.X \subseteq C_{12}$. First notice that, since C_1, C_2 are ℓ-disjoint (by Lemma A.3), X is ℓ-disjoint too. We have the following cases.

— $a = 1$. Then X does not contain label 1, so $\mu_1 x.X^{-2} \subseteq C_1$. Hence, by Lemma A.3, $A \twoheadrightarrow X^{-2}$ ($\equiv X^{-12}$). Furthermore, if $\mu_2 x.X' \subseteq X$, then X' is label free, and so $\mu_2 x.X' \subseteq C_2$. By Lemma A.3, $A \twoheadrightarrow X'$. Hence, by the pushout lemma (Lemma A.8), there is a label free term Y such that $X \twoheadrightarrow_\ell Y$.

— $a = 2$. Likewise.

— $a = 12$. Then X is label free.

Summarizing, in all cases there is a label free term Y such that $X \twoheadrightarrow_\ell Y$ whenever $\mu_a x.\ X \subseteq C_{12}$. Hence, there is an ℓ-disjoint term B such that $C_{12} \twoheadrightarrow_\ell B$.

Finally, notice that

$$\begin{aligned} F(\mu_1 x.A) &\twoheadrightarrow_\ell C_1 \\ &\equiv C_{12}^{-2} \\ &\twoheadrightarrow_\ell B^{-2}. \end{aligned}$$

Likewise,

$$F'(\mu_2 x.\, A) \twoheadrightarrow_\ell B^{-1}. \quad \blacksquare$$

LEMMA A.10. *Let F, M, N be label free, and let N be closed. If $\mu x.M$ is μ-hierarchical, and $\boldsymbol{\lambda\mu} \vdash M[x := N] = N$, then*

$$F(\mu_a x.M) \twoheadrightarrow_\ell A \Rightarrow \boldsymbol{\lambda\mu} \vdash FN = A\,|^a_N.$$

Proof. By induction on the length of the reduction $F(\mu_a x.M) \twoheadrightarrow_\ell A$.

Basic case. The length of the reduction is 0. Immediate.

Induction case. The length of the reduction is $n+1$. Then there is an A' such that $A' \to_\ell A$, and $F(\mu_a x.M) \twoheadrightarrow_\ell A'$ in n steps. By Lemma A.3, A' is ℓ-disjoint, hence the following case distinction suffices.

— The reduction $A' \to_\ell A$ is inside a term of the form $\mu_a x.M'$. Then clearly $A'\,|^a_N \equiv A\,|^a_N$, and the result follows by the induction hypothesis.

— The reduction $A' \to_\ell A$ is an application of the rule

$$\mu_a x.M' \to_\ell M'[x := \mu_a x.M'].$$

By Lemma A.3 we have that

$$\boldsymbol{\lambda\mu} \vdash M = M'.$$

By assumption

$$\boldsymbol{\lambda\mu} \vdash N = M[x := N],$$

hence

$$\boldsymbol{\lambda\mu} \vdash N = M'[x := N],$$

hence

$$\boldsymbol{\lambda\mu} \vdash A'\,|^a_N = A\,|^a_N.$$

The result follows by the induction hypothesis.

— In all other cases, $A' \to_\ell A$ immediately implies

$$\boldsymbol{\lambda\mu} \vdash A' |^a_N = A |^a_N. \quad ▮$$

LEMMA A.11 (= Lemma 5.2(a)). *Let N be closed and let A be in normal form. If $\mu x.M$ is μ-hierarchical, and $\boldsymbol{\lambda\mu} \vdash M[x := N] = N$, then*

$$\boldsymbol{\lambda\mu} \vdash F(\mu x.M) = A \;\Rightarrow\; \boldsymbol{\lambda\mu} \vdash FN = A.$$

Proof. By the Church–Rosser property we have that $F(\mu x.M) \twoheadrightarrow A$. Since A is a normal form, it does not contain μ-terms. Hence, A does not contain labelled terms. By Lemma A.1 it now follows that $F(\mu_a x.M) \twoheadrightarrow_\ell A$ for any label a. Hence, by Lemma A.10 $\boldsymbol{\lambda\mu} \vdash FN = A$. ▮

LEMMA A.12 (= Lemma 5.2(b)). *Let N be closed. Then*

$$\left.\begin{array}{l} \mu x.M \text{ is } \mu\text{-hierarchical} \\ \boldsymbol{\lambda\mu} \vdash M[x := N] = N \end{array}\right\} \Rightarrow \mu x.M \text{ is proof-replaceable by } N.$$

Proof. Let F, F' be closed terms such that

$$\boldsymbol{\lambda\mu} \vdash F(\mu x.M) = F'(\mu x.M).$$

We have to prove that there is a G such that

$$\boldsymbol{\lambda\mu} \vdash G(\mu x.M)\, N = FN,$$

$$\boldsymbol{\lambda\mu} \vdash GN(\mu x.M) = F'N.$$

By Lemma A.9, there is an ℓ-disjoint term A such that

$$F(\mu_1 x.M) \twoheadrightarrow_\ell A^{-2},$$

$$F'(\mu_2 x.M) \twoheadrightarrow_\ell A^{-1}.$$

By Lemma A.3, we have that $M \twoheadrightarrow M'$, whenever $\mu_a x.M' \subseteq A$. Hence,

$$\boldsymbol{\lambda\mu} \vdash \mu x.M' = \mu x.M.$$

Let x, y be new variables, and define

$$G \equiv \lambda xy.A |^1_y |^2_x |^{12}_N.$$

Then in $\boldsymbol{\lambda\mu}$ we have

$$\begin{aligned} G(\mu x.M)N &= A|^1_N|^2_{\mu x.M}|^{12}_N \\ &= A|^1_N|^{12}_N|^2_{\mu x.M} \\ &= (A|^1_N|^{12}_N)^{-2} && \text{since } (\mu_2 x.M')^{-2} = \mu x.M, \\ &= A^{-2}|^1_N && \text{immediate,} \\ &= FN && \text{by Lemma A.10.} \end{aligned}$$

Likewise,

$$GN(\mu x.M) = F'N.$$

Hence, $\mu x.M$ is proof-replaceable by N. ∎

ACKNOWLEDGMENTS

It is a pleasure to remember the fruitful discussions with Henk Barendregt and Maarten Fokkinga. I thank Adriana Compagnoni for useful comments. Example 2 in Section 4 was mentioned to me by Professor Jacopini.

Received August 1, 1995; final manuscript received January 30, 1997

REFERENCES

Baeten, J., and Boerboom, B. (1979), Ω can be anything it should not be, *Indag. Math.* **41**, 111–120.

Barendregt, H. P. (1984), "The Lambda Calculus—Its Syntax and Semantics," rev. ed., North-Holland, Amsterdam.

Berarducci, A., and Intrigila, B. (1993), Some new results on easy lambda terms, *Theoret. Comput. Sci.* **121**, 71–88.

Berry, G., Curien, P.-L., and Lévy, J.-J. (1985), Full abstraction for sequential languages: state of the art, *in* "Algebraic Methods in Semantics" (M. Nivat and J. C. Reynolds, Eds.), pp. 89–132, Cambridge University Press, Cambridge, UK.

Intrigila, B. (1991), A problem on easy terms in lambda calculus, *Fund. Inform.* **15**, 99–106.

Jacopini, G. (1975), A condition for identifying two elements of whatever model of combinatory logic, *in* "Lambda Calculus and Computer Science" (C. Böhm, Ed.), pp. 213–219, Springer-Verlag, Berlin.

Jacopini, G. (1994), personal communication, Enschede, The Netherlands.

Jacopini, G., and Venturini Zilli, M. (1978), "Equating for Recurrent Terms of Lamda Calculus and Combinatory Logic," Quaderni IAC, Serie III, Vol. 85, Istituto per le Applicazioni del Calcolo "Mauro Picone," Rome.

Jacopini, G., and Venturini Zilli, M. (1985), Easy terms in the lambda calculus, *Fund Inform.* **80**.

Klop, J. W. (1980), "Combinatory Reduction Systems," Ph.D. Thesis, Mathematical Centre, Amsterdam.

Mitchell, J. C. (1996), "Foundations for Programming Languages," MIT Press, Cambridge, MA.

Plotkin, G. D. (1977), LCF considered as a programming language, *Theoret. Comput. Sci.* **5**, 223–255.

van Raamsdonk, F. (1992), "A Simple Proof of Confluence for Weakly Orthogonal Combinatory Reduction Systems," Report CS-R9234, Centre for Mathematics and Computer Science, Amsterdam.

Statman, R. (1986), Solving functional equations at higher types: Some examples and some theorems, *Notre Dame J. Formal Logic* **27**, 66–74.