SOLVABILITY IN LAMBDA CALCULI

H. Barendregt
University of Utrecht

ABSTRACT. This paper discusses for various versions of the type-free \( \lambda \)-calculus the concept of solvability, introduced in [1]. For the \( \lambda K \)-calculus an equivalent notion of head normal form was introduced by Wadsworth [17]. Arguments are given for the utility of this concept and the proposal that the unsolvable terms should be considered as terms without a meaning.

ACKNOWLEDGEMENT. I would like to thank Professor G. Kreisel for useful conversations about the concept of solvability.

INTRODUCTION.

We assume familiarity with the \( \lambda K \)-and \( \lambda I \)-calculus as presented in [6], [7], and Scott’s models \( D_\omega \), \( P_\omega \) (see [15] [16]).

The discussion starts with a comparison between the \( \lambda I \)- and \( \lambda K \)-calculus. It turns out that in the I-case terms without a normal form (nf) can be equated consistently, but not so in the K-case. The general theorem is that the theory identifying all unsolvable terms is consistent. Moreover it will be shown that this theory has a unique Hilbert-Post complete extension\(^1\).

For the I-case this HP complete extension is the extensional \( \lambda I \)-calculus in which all terms without a nf are equated: for the K-case it is the set of equations true in any of Scott’s lattice theoretic models for the \( \lambda \)-calculus.

The representation of the partial recursive functions gives other arguments for the usefulness of the notion of solvability. For the \( \lambda I \)-calculus partial recursive functions were represented in such a way that if \( f(n) \) is undefined, then, the representing term has no nf. Taking this representation literally for the \( \lambda K \)-calculus causes several difficulties, which can be avoided by interpreting undefinedness as unsolvability. In this way it is possible to give a representation faithful w.r.t. the definitions of the \( \mu \)-recursive functions.

The following versions of the type-free \( \lambda \)-calculus will be considered:

\(^*\) La première version de ce travail a été présentée au Colloque de logique d’Orléans en 1972.

1) Let \( \mathcal{F} \) be some set of equation between \( \lambda \)-terms. \( \mathcal{F} \) is consistent iff the theory \( \lambda \vdash \mathcal{F} \) does not prove every equation. \( \mathcal{F} \) is Hilbert-Post (HP) complete iff (i) \( \mathcal{F} \) is consistent.
   (ii) \( \lambda \vdash \mathcal{F} \vdash M \equiv N \Rightarrow \lambda \vdash \mathcal{F} + M = N \) is inconsistent. HP complete theories are maximal consistent collections of equations.
the \( \lambda I \) and the \( \lambda K \)-calculus, with and without extensionality and their combinatorial versions. For an introduction see \([3], [6]\) or \([8]\).

**NOTATIONS.** \( \lambda I(\eta) \rightarrow \) denotes provability in the \( \lambda I \)-calculus (+ extensionality). Similarly for \( \lambda K(\eta) \rightarrow \Lambda_I, \Lambda_K \)

\( \lambda K \)-terms, \( \Lambda_I^*, \Lambda_K^* \) denote the subsets of terms without free variables.

If \( \lambda(\eta) \lambda_l, \Lambda_l^* \) are used in a certain context, then these symbols can be read throughout as \( \lambda I(\eta), \Lambda_I, \Lambda_I^* \) or \( \lambda K(\eta), \Lambda_K, \Lambda_K^* \).

The same applies to derived notions (e.g. I-vs. K-solvable).

Also the notion of consistency depends on the system in which one works.

\( \Omega \) denotes the term \((\lambda x.xxx)(\lambda x.xxx) \in \Lambda_I^* \).

1. \( \lambda I \)-vs. \( \lambda K \)-solvability.

The \( \lambda I \)-calculus was introduced by Church. The \( \lambda K \)-calculus is the \( \lambda \)-variant of the theory of combinators introduced by Schönfinkel and Curry.

In \([7]\), S.S. Church is cited as giving the following arguments for his preference of the \( \lambda I \)-calculus over the \( \lambda K \)-calculus.

1) What Church wanted to do with the \( \lambda \)-calculus could be done with the \( \lambda I \)-version, e.g. the representation of the partial recursive functions.

2) If one - like Church - considers as significant only terms having a normal form\( (2) \), then significant \( \lambda K \)-terms may have non-significant parts (e.g. \( KI \Omega \)). However, this is not so for \( \lambda I \)-terms.

3) The \( \lambda K \)-calculus might lead to inconsistencies.

The reason behind argument 1) can be explained by the existence of several approximations for \( K \) definable in the \( \lambda I \)-calculus.

In the first place there exists a \( K_I \in \Lambda_I^* \), such that \( K_I x n = x \) for all numerals \( n \) (for Church numerals take \( K_I = \lambda xy.yI Ix \)). Secondly, more recently it is shown, that for a finite set of normal forms there exists a local \( K \):

1.1. **Theorem.** (Barendregt, Klop) Let \( \mathfrak{N} \subseteq \Lambda_I^* \) be a finite set of terms having a normal form. Then there exists a \( K_{\mathfrak{N}} \in \Lambda_I^* \) such that

\[ \lambda I \rightarrow K_{\mathfrak{N}} MN = M \] for all \( M \) and all \( N \in \mathfrak{N} \cup \{ K_{\mathfrak{N}} \} \).

**Proof.** See \([12]\), 9.

Since Church believed in his thesis, the fact that all partial recursive functions can be represented in the \( \lambda I \)-calculus was satisfactory for him: all computational processes can be represented in the \( \lambda I \)-calculus. However this representation is not the most efficient one. For the representation of programming languages the \( \lambda I \)-calculus has the disadvantage that a call by name mechanism

\( (2) \) On several occasions, Scott strongly disagreed with this position e.g. \([14]\), p. 159, 165.

Below it will be shown that the views of Church and Scott are nevertheless compatible.
cannot be implemented: a $\lambda$-computation of $0.2^{100}$ has to first evaluate $2^{100}$ and then map the result on 0. In the $\lambda K$-calculus $0.2^{100}$ can reduce immediately to 0. This feature of the $\lambda K$-calculus is due to the fact mentioned in argument 2, which therefore is rather an advantage of this theory.

As to argument 3, the $\lambda K$-calculus itself is consistent, as follows from the extension of the Church-Rosser theorem to the $\lambda K$-calculus, see e.g., [8], [3]. However it would seem that Church is right that the $\lambda K$-calculus may lead to inconsistencies. For if one interprets the «non-significance» of terms having no nf in such a way that they are to be identified, then the $\lambda K$-calculus becomes inconsistent. (This is not so for the $\lambda I$-calculus, see 1.16).

1.2. Theorem. It is inconsistent to identify in the $K$-calculus all terms without a nf.
Proof. Let $M_0 \equiv \lambda x . x \Omega$ where $\Omega$ is a term without nf. Then

$$\lambda K \vdash M_0 = M_1 \leftarrow 0 = M_0 K = M_1 K = 1$$

However it will become clear that in the $\lambda K$-calculus not all terms without a nf should be considered as meaningless.

The inconsistency shown in 1.2 is due to the different «solving» behaviour of $M_0$, $M_1$.

1.3. Definition. A closed term $M$ is solvable iff $\lambda \leftarrow M N_1 \ldots N_n = I$ for some $n$ and some $N_1 \ldots N_n \in \Lambda$. An arbitrary term $M$ is solvable iff its closure $\lambda x . M$ is solvable.

To see the particular role if I ($= \lambda x . x$) in this definition, note that $M$ (closed) is solvable iff $\forall P \exists N M N = P$.

The definition of solvability can be given for the various versions of the $\lambda$-calculus. Most of these concepts are equivalent.

1.4. Theorem. (i) $M$ is solvable in one of the versions of the $\lambda$-calculus iff $M$ is solvable in this version extended with extensionality.
(ii) Solvability is invariant under the $\lambda \neq CL$ translations.

Proof. See [9], 5 or [4] ■

Hence there are essentially only two concepts of solvability:

Definition 1.5. Let $M \in \Lambda$.
(i) $M$ is $I$-solvable iff $\exists N \in \Lambda_I \lambda \leftarrow M N = I$
(ii) $M$ is $K$-solvable iff $\exists N \in \Lambda_K \lambda K \leftarrow M N = I$.

An arbitrary $M \in \Lambda$ is $I$- or $K$-solvable iff its closure is.

In both cases $M$ may be a $\lambda K$-term. Since the $\lambda K$-calculus is conservative over the $\lambda I$-calculus (see [2] 1.3), it does not matter if for $M \in \Lambda_I$ $I$-solvability is defined by provability in $\lambda I$ or $\lambda K$.

In both cases a syntactic characterization can be given.
1.6. **Definition** (Wadsworth) (i) A $\lambda$-term $M$ is in head normal form (hnf) iff $M$ is of the form $\lambda x_1 \ldots x_n. y N_1 \ldots N_n$, for some $n, k \geq 0$.

(ii) A $\lambda K$-term $M$ has a hnf iff $\lambda K \equiv M \equiv M'$ and $M'$ is in hnf.

1.7. **Theorem.** (i) (Barendregt) For the $\lambda I$-calculus: $M$ is solvable $\iff$ $M$ has a normal form.

(ii) (Wadsworth) For the $\lambda K$-calculus: $M$ is solvable $\iff$ $M$ has a head normal form.

**Proof.** See [2] 2.1 or [12] 7 and [18] 5.3.

Head normal forms are a proper extension of the normal forms. Hence both for the $I$- and $K$-case terms with a normal form are solvable.

Passing from the $\lambda I$- to the $\lambda K$-calculus there is a choice for the generalization of the class of $\lambda I$-terms (i.e., $M$ has a nf) $\equiv M | M$ is solvable.

We claim that solvability is the more natural and fruitful concept.

The unsolvable (i.e., not solvable) terms not only have no nf, but even hereditarily so.

1.8. **Theorem.** Let $M \in \Lambda^*$. Then $M$ is solvable $\iff \exists N M N$ has a nf.

**Proof.** $\Rightarrow$ by definition. $\Leftarrow$ since normal forms are solvable. $

This property of hereditary undefinedness motivates the following:

1.9. **Proposal.** The unsolvable terms should be considered as being meaningless (undefined).

Adopting proposal 1.9 reconciles the positions of Church and Scott concerning the significance of terms. Church had in mind the $\lambda I$-calculus, hence his standpoint is by 1.7 (i) equivalent to 1.9. Scott on the other hand was speaking about the $\lambda K$-calculus, where the solvable terms form a proper extension of the terms having a normal form. That Scott will surely accept 1.9 follows from 1.18 below.

The definition of $\exists M$ is in hfn, viz. $\exists M$ has no redex subterms, is syntactical.

It was pointed out by Wadsworth that the concept of normalizability is essential syntactical for by his 2.13 it follows that there is no set $\exists \mathcal{G} \subset \mathcal{D}_o$ such that

$M$ has a nf $\iff [M]_{\mathcal{D}_o} \in \mathcal{G}$

On the other hand, by its very definition, the property of solvability is semantic, i.e., makes sense in models, although not necessarily first order. (In $\mathcal{D}_o$ and $\mathcal{P}_{\mathcal{G}}$, solvability is a first order property as follows from 1.18). This is another argument for the preferability of taking the unsolvable terms rather than the terms without a nf, as the class of terms without a meaning.

Another argument for 1.9 is that the unsolvable terms are from a computational point of view not very informative, they are generic (in the sense of algebraic geometry: if a generic point has a certain property, the whole space has this property).
1.10. **Theorem** (genericity lemma). Let \( \Omega \) be unsolvable. If \( F\Omega = I \), then \( FM = I \) for all \( M \).

**Proof.** See \cite{4} or \cite{1} 3.2.3. ■

1.11. **Corollary.** Let \( \Omega \) be unsolvable. If \( F\Omega \) is solvable, then \( FM \) is solvable for all \( M \).

**Proof.** Let \( F\Omega \) be solvable. Then \( F\Omega N = I \).

Hence by 1.10 (applied to \( F' = \lambda x. FxN \)) \( FMN = I \), i.e. \( FM \) is solvable, for all \( M \). ■

In \cite{14} 1.8 Scott introduced the following notion:

1.12. **Definition.** \( M \subseteq N \iff \forall F \text{ closed} \ [FM \text{ defined} \Rightarrow FN \text{ defined}] \).

Interpreting «defined» as «solvable», as is proposed in 1.9, we derive the following axiom of Scott \cite{14} 1.10.

1.13. **Corollary.** If \( \Omega \) is undefined, then \( \Omega \subseteq M \) for all \( M \).

**Proof.** By 1.11. ■

The hereditary undefinedness of the unsolvable terms makes it plausible that they can be identified.

1.14. **Definition.** \( \mathcal{P} = \{ \Omega = \Omega' \mid \Omega, \Omega' \text{ are unsolvable} \} \). If it is necessary to distinguish between the I- and K-case we use the notations \( \mathcal{P}_I \) and \( \mathcal{P}_K \).

1.15. **Theorem.** The theory extended with \( \mathcal{P} \) (+extensionality) is consistent (both for the I- and K-case).

**Proof.** The first proof in \cite{1} went via the genericity lemma.

See 2.4 for a more elegant version of this proof. A model theoretic proof for the K-case follows from 1.19. ■

Relativizing 1.15 to the I-case establishes:

1.16. **Corollary.** In the \( \lambda \)-calculus (+ extensionality) it is consistent to equate all terms without a nf.

1.17. **Definition.** A model \( \mathcal{M} \) of the \( \lambda \)-calculus is **sensible** iff all unsolvable terms are equal in \( \mathcal{M} \).

The term model of \( \mathcal{P} \), which exists by 1.15, is a sensible model. Also Scotts models are sensible.

1.18. **Theorem.** (Hyland; Wadsworth). Let \( D_\omega \) and \( P_\omega \) be Scott's well-known models for the \( \lambda \)-calculus, with least elements \( \downarrow \) and \( \emptyset \) respectively.

Then \( M \) is unsolvable \( \Leftrightarrow D_\omega \models M \models P_\omega \models M = \emptyset \).

**Proof.** See \cite{10} and \cite{18}. ■

1.19. **Corollary.** \( D_\omega \) and \( P_\omega \) are sensible models.

2. **The Hilbert Post completion of \( \mathcal{P} \).**

Let \( \text{Th}(D_\omega) \) be the set of equations true in Scott's \( D_\omega \) models for the \( \lambda \)-calculus. Hyland and Wadsworth gave a syntactic characterization of \( \text{Th}(D_\omega) \)
independent of the initial lattice D. Moreover they showed that \( \text{Th}(D_\omega) \) is in the \( \lambda K \)-calculus the unique HP-complete extension \( \mathcal{X} \) of \( \mathcal{X} \). The existence of \( \mathcal{X}^* \) can be proved without going into the details of \( \Omega_\omega \).

2.1. Definition

(i) Let \( M, N \in \Lambda^+ \). \( M \) and \( N \) are solvably equivalent, notation \( M \approx N \), iff for all \( F \) \( \{ F \text{ is solvable} \} \implies \{ F N \text{ is solvable} \} \).

(ii) Let \( M, N \in \Lambda \) be arbitrary. \( M \approx N \) iff \( \lambda x M \approx \lambda x N \), where \( x \) is a string of variables such that \( \text{FV}(M) \cup \text{FV}(N) \subseteq \{ x \} \).

Note that the definition 2.1 (ii) is independent of the choice of \( x \).

2.2. Definition. \( \mathcal{X}^* = \{ M = N \mid M \approx N \} \)

2.3. Theorem.

(i) \( \lambda x \mathcal{X}^* \vdash M = N \implies M \in \mathcal{X}^* \)

(ii) \( \mathcal{X}^* \) is consistent.

Proof.

(i) Induction on the length of proof of \( \lambda x \mathcal{X}^* \vdash M = N \).

(ii) Note that I \( \notin \Omega \) and use (i).

2.4. Theorem.

(i) \( \mathcal{X} \subset \mathcal{X}^* \); (ii) \( \mathcal{X} \) is consistent.

Proof. (i) Suppose \( \Omega = \Omega' \in \mathcal{X} \). Then \( \Omega, \Omega' \) are unsolvable. We may assume \( \Omega, \Omega' \) are closed. Claim \( \Omega \approx \Omega' \). Indeed if \( F \Omega \) were solvable, then by 1.11 \( F \Omega' \) is solvable and conversely.

(ii) By (i), since \( \mathcal{X} \) is consistent.

2.5. Theorem (Böhm). Let \( M, N \) have different \( \beta \eta \) nf's. Then \( M = N \) is inconsistent.

Proof. For the K-case this is proved in [5]. For the I-case the proof is the same using the proof of 1.1, see [4].

2.6. Lemma. (Jacopini) Let \( \omega_3 = \lambda x.xxx \) and \( \Omega_3 = \omega_3 \omega_3 \omega_3 \). Then \( I = \Omega_3 \) is inconsistent.

Proof. \( I = \Omega_3 \vdash I = \Omega_3 = \Omega_3 \omega_3 = I \omega_3 = \omega_3 \).

Now 2.5 applies to I, \( \omega_3 \).

2.7. Lemma. \( \mathcal{X} + M = N \) is consistent \( \Rightarrow M = N \in \mathcal{X}^* \)

Proof. Suppose \( M = N \in \mathcal{X} \), i.e. \( M \not\in \mathcal{X} \). Let, say, \( F \) \( \text{solvable} \) and \( FN \) unsolvable. Then \( \exists \bar{N} \) \( \lambda F \bar{N} = I \) and \( \lambda \mathcal{X} \vdash \bar{N} \mathcal{X} \). Therefore \( \lambda \mathcal{X} + M = N \vdash FN \mathcal{X} = \Omega_3 \), since \( FN \mathcal{X} = \Omega_3 \).
hence by 2.6 \( \mathcal{K}^+ M = N \) is inconsistent. ■

2.6. Theorem. \( \mathcal{K}^* \) is the unique HP-complete extension of \( \mathcal{K} \).

Proof. By 2.3 (ii), 2.4 (i) \( \mathcal{K}^* \) is a consistent extension of \( \mathcal{K} \).

Let \( \mathcal{T} \supseteq \mathcal{K} \) be consistent. Then by 2.7 \( \mathcal{T} \subseteq \mathcal{K}^* \). Hence \( \mathcal{K}^* \) contains all consistent extensions of \( \mathcal{K} \) and is therefore the unique maximal consistent, i.e. HP-complete, theory extending \( \mathcal{K} \). ■

Relativizing definition 2.2 to the \( K \)- and \( I \)-cases one obtains the HP-complete theories \( \mathcal{K}_I \), \( \mathcal{K}_I^* \).

Both sets have an interesting characterization, 2.9 and 2.12.

2.9. Theorem. (Hyland; Wadsworth) \( M = N \mid \models M = N \).

Proof. See \cite{10}, \cite{18} or \cite{4}. ■

2.10. Corollary. \( D^{\omega} = \{ [M]^{D^{\omega}} \mid M \text{ closed term} \} \) is algebraically simple.

Proof. If \( D^{\omega} \) would have a proper homomorphic image \( D' \), then \( Th(D') \) would be a consistent extension of \( \mathcal{K}^* \). ■

2.11. Lemma. (Curry) \( M \in \Lambda \) has a \( \beta \)-nf \( \iff \) \( M \) has a \( \beta \eta \)-nf.

Proof. See \cite{2}, 1.4. ■

2.12. Theorem. The set of equations provable in \( \lambda \eta (M = N \mid M, N \ \lambda \eta \text{-terms without nf} \) (i.e. in \( \lambda \eta^* \mathcal{K}_I \)) is \( \mathcal{K}_I^* \) and hence HP-complete.

Proof. The theory \( \lambda \eta^* \mathcal{K}_I \) is consistent by 1.16. Hence by 2.8 relativized to the \( I \)-case \( \lambda \eta^* \mathcal{K}_I \models M = N \Rightarrow M = N \in \mathcal{K}_I^* \).

Now suppose \( M \approx I N \). Claim \( \lambda \eta^* \mathcal{K}_I \models M = N \). Case 1. \( M, N \) are \( I \)-unsolvable.

Then \( \mathcal{K}_I \models M = N \).

Case 2. Exactly one of \( M, N \) is \( I \)-solvable. Then not \( M \approx I N \).

Case 3. \( M, N \) are \( I \)-solvable. By 1.7 (i) and 2.11 \( M, N \) have \( \beta \eta \)-nf's.

Subcase 3.1 \( M, N \) have identical \( \beta \eta \)-nf's.

Then \( \lambda \eta \models M = N \).

Subcase 3.2 \( M, N \) have different \( \beta \eta \)-nf's.

Then by 2.5 not \( M \approx I N \), since \( \mathcal{K}_I^* \) is consistent. ■

It should be noted that \( \mathcal{K}_I^* \) and \( \mathcal{K}_K^* \) are overlapping sets of equations, and hence by their HP-completeness incompatible, i.e. \( \mathcal{K}_I^* \cup \mathcal{K}_K^* \) is inconsistent.

2.13. Theorem. (Wadsworth) Let \( H = \lambda x y x(j y) \) and \( J = F P H \), i.e. the fixed point of \( H \). Then \( D^{\omega} \models I = J \).

Proof. See \cite{18}, 3.2. ■
2.14. Remark. \( \mathcal{H}_1 \) and \( \mathcal{H}_K \) are overlapping.

Proof. \( I \in \mathcal{H}_K \) by 2.13 and 2.9, but \( I \not\in \mathcal{H}_I \), since \( I \) has no nf.

\[ \Omega = \lambda x.x \Omega \in \mathcal{H}_I \] since both terms have no nf, hence are I-unsolvable,
but
\[ \Omega = \lambda x.x \Omega \not\in \mathcal{H}_K \] since \( \lambda x.x \) is K-solvable and \( \Omega \) not. ■

3. The representation of the partial recursive functions.

In § 1 it was noted that the notions of solvability and having a normal form are the same for the \( \lambda I \)-calculus, but not so for the \( \lambda K \)-calculus. Definitions involving the concept of normal form which worked well for the I-case, have some disadvantages for the K-case.

3.1. Definition (Church) A partial function \( f : \omega \rightarrow \omega \) is \( \lambda I \)-definable iff for some \( F \in \Lambda_1 \)

\[ \lambda I \vdash F n = m \quad \text{if} \quad f(n) = m \]
\[ F \not\vdash \text{has no nf} \quad \text{if} \quad f(n) \text{ is undefined.} \]

In this situation \( F \) is said to represent \( f \).

Here \( 0, 1, \ldots \) is some sequence \( \in \Lambda_1 \) representing the integers.

3.2. Theorem (Kleene). Let \( f : \omega \rightarrow \omega \) be a partial function. Then \( f \) is \( \lambda I \)-definable \( \Leftrightarrow \) \( f \) is partial recursive.


In [7] 13A definition 3.1 was copied literally for the \( \lambda K \)-calculus and the corresponding version of 3.2 was proved.

This had the following disadvantages:

1. Because of the call by name character of the \( \lambda K \)-calculus, the representation of a composition is not necessarily the composition of the representations. E.g. Let \( f(n) = 0 \) and \( g(n) \) be undefined for all \( n \). Then
\[ F = \lambda x.0 \quad \text{and} \quad G = \lambda x.\Omega \quad \text{\( \lambda K \)-represent} \ f \text{ and} \ g. \]
\[ f \cdot g \] is the totally undefined function \( = g \). However
\[ F \cdot G = \lambda x. F(Gx) = \lambda x. 0 , \] which does not represent \( g \).

The proof in [7] of the K-version of 3.2 avoids this difficulty by writing the partial recursive functions in Kleene's normal form
\[ f(x) = U(\mu z T(e,x,z)). \]

Then a representation of \( f \) can be found from its index \( e \) and the representations of \( U \) and \( T \). However this representation is not intensional, i.e. does not preserve the definition of \( f \) as a \( \mu \)-recursive function.

2. Having a normal form is not invariant under the standard translations between the various versions of the \( \lambda \)-calculus. E.g. \( S(\lambda I(SI)) \) \( (\lambda K SI) \) is in combinatory
logic a normal form, but not its \( \lambda \)-translation, which is convertible to \( \lambda a. \Omega \).

Therefore the representation had to be proved separately for the different systems. See \[7\], 13 A4. Disadvantage 2 applies as well to the I-case.

Definition 3.1 can be modified in the spirit of proposal 1.9.

3.3. Definition. A partial function \( f : \omega^k \rightarrow \omega \) is strongly \( \lambda \)-definable iff for some term \( F \)
\[
\begin{align*}
\lambda \mapsto F & \equiv m \quad \text{if } f(n) = m \\
F \notin \text{unsolvable} & \quad \text{if } f(n) \text{ is undefined}.
\end{align*}
\]

3.4. Theorem. The strongly \( \lambda \)-definable functions are exactly the partial recursive functions.

Proof. If \( f \) is strongly \( \lambda \)-definable its graph is r.e. (since the \( \lambda \)-calculus is recursively axiomatized) hence \( f \) is partial recursive. Conversely we show as a prime example how the representations of two functions can be used to represent their composition. Let \( f, g \) be functions of one argument and be represented by \( F, G \). Let \( h(x) = f(g(x)) \). Let \( \sigma \) be a sequence of terms such that for all \( n \),
\[
\lambda \mapsto \sigma \equiv 1. \text{ For Church's numerals we can take } \sigma \equiv \Pi, \text{ for the numerals in } [3] \text{ we can take } \sigma \equiv \Pi_{II}.
\]

Now \( h \) can be represented by \( H = \lambda x. (Gx \sigma) (F(Gx)) \) : if \( h(n) = m \), then \( g(n) \) is defined and the "jamming" factor \( G \sigma \) vanishes, i.e. becomes 1. Therefore if \( n = f(g(x)) \). Let \( \sigma \) be a sequence of terms such that for all \( n \), or \( f(g(n)) \) is undefined. In the first case \( G \sigma \) is unsolvable, hence also \( G \sigma (F(Gn)) \), i.e. \( H \sigma \) is unsolvable. Also in the second case \( H \sigma \) is unsolvable.

Similarly one can give a definition-preserving representation of functions defined by primitive recursion or minimalization.

It has been stressed by Kreisel [13], p. 177, 178 that in connection with the so called "superthesis", Church's thesis expresses less then we know. When we say that all mechanically computable numbertheoretic functions are \( \lambda \)-definable or recursive, we merely speak of the results of computations, of their graphs. But we have in mind that \( \lambda \)-terms correspond to our procedures for defining these functions. As far as the \( \mu \)-recursive and the \( \lambda \)-definable functions are concerned, strong definability proves the equivalence not only in the sense of Church's thesis, but also of the super thesis: definitions are preserved.

Apart from the interest of representing the partial recursive intensionality, the following use has been made of strong definability.

3.5. Theorem (Wadsworth). Let \( \varphi_x \) be the partial recursive function with index \( x \). There is a strong representation of the partial recursive functions, \( \varphi_x \) being represented by \( F_x \), such that
\[
\varphi_x \equiv \varphi_y \iff D_x \implies F_x = F_y.
\]

Proof. See [19] or [4].
3.6. Corollary. (Wadsworth) \( \mathcal{N} = \text{Th}(D_\infty) \) is \( \Pi^0_2 \) complete.

Proof. By its very definition \( \mathcal{N} \) is \( \Pi^0_2 \). By 3.5 it is complete, since

\[ \varphi_x \equiv \varphi_y \text{ is. } \]

REFERENCES


[9] Hindley, J.R., Combinatory reductions and lambda reductions, mimeographed, 1975:


