THE THEORETICAL ASPECTS OF THE OPTIMAL FIXEDPOINT*

by

Zohar Manna and Adi Shamir**

Research sponsored by
Advanced Research Projects Agency
ARPA Order No. 2494

COMPUTER SCIENCE DEPARTMENT
Stanford University
THE THEORETICAL ASPECTS OF THE OPTIMAL FIXEDPOINT

by

Zohar Manna and Adi Shamir

ABSTRACT

In this paper we define a new type of fixedpoint of recursive definitions and investigate some of its properties. This optimal fixedpoint (which always uniquely exists) contains, in some sense, the maximal amount of "interesting" information which can be extracted from the recursive definition, and it may be strictly more defined than the program's least fixedpoint. This fixedpoint can be the basis for assigning a new semantics to recursive programs.

This is a modified and extended version of part I of a paper [4] presented at the symposium on Theory of Computing, Albuquerque, New Mexico (May 1975). Present address: Computer Science Department, University of Warwick, Coventry CV47AL, England

This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract DAHC 15-73-C-0435. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Stanford University, ARPA, or the U.S. Government.

Reproduced in the U.S.A. Available from the National Technical Information Service, Springfield, Virginia 22151.
INTRODUCTION

Recursive definitions are usually considered from two different points of view, namely:

(i) As an algorithm for computing a function by repeated substitutions of the function definition for its name.

(ii) As a functional equation, expressing the required relations between values of the defined function for various arguments. A function that satisfies these relations (a solution of the equation) is called a fixedpoint.

The functional equation represented by a recursive definition may have many fixedpoints, all of which satisfy the relations dictated by the definition. There is no a priori preferred solution and therefore, if the definition has more than one fixedpoint, one of them must be chosen. A number of works describing a least (defined) fixedpoint approach towards the semantics of recursive definitions have been published recently (e.g., Scott [8]). Researchers in the field have chosen the least fixedpoint as the "best solution" for three reasons:

(i) It uniquely exists for a wide class of practically applicable recursive definitions.

(ii) The classical stack implementation technique computes this fixedpoint for any recursive definition.

(iii) There is a powerful method (computational induction) for proving properties of this fixedpoint.

However, as a mathematical model for extracting information from an implicit functional equation, the selection of the least defined solution seems a poor choice; for many recursive definitions, the least fixedpoint does not reveal all the useful information embedded in the definition. In general, the more defined the solution, the more valuable it is. On the other hand, this argument
should be applied with caution, as there are inherently underdefined recursive
definitions. Consider the extreme example $F(x) \leq F(x)$, for which any partial
function is a solution. A randomly chosen total function is by no means
superior to the totally undefined least fixedpoint in this case.

The **optimal fixedpoint**, defined in this paper, tries to remedy this situation.
It is intended to supply the maximally defined solution relevant to the given
recursive definition. Consider, for example, the following recursive definition for
solving the discrete form of the Laplace equation, where $F(x,y)$ maps pairs of
integers in $[-100,100] \times [-100,100]$ into reals:

$$F(x,y) \leq \begin{cases} x^2 + y^2 & \text{if } x<-100 \lor x>100 \lor y<-100 \lor y>100 \\ \frac{1}{4}[F(x-1,y)+F(x+1,y)+F(x,y-1)+F(x,y+1)]. \end{cases}$$

This concise organization of knowledge is defined enough to have a unique
total fixedpoint (which is our optimal fixedpoint), but its least fixedpoint
is totally undefined inside the square $[-100,100] \times [-100,100]$.

While the notion of the optimal fixedpoint is theoretically well-defined, its
computation aspects contain many pitfalls, since the optimal fixedpoints of
certain recursive definitions are non-computable partial functions. We do not
pursue in this paper the practical aspects of the optimal fixedpoint approach;
in Manna and Shamir[4,5], and in more detail in Shamir[8], we suggest several
techniques directed toward the computation of the optimal fixedpoint.

In Part I of this paper, a few structural properties of the set of all fixedpoints
of recursive definitions are proven. The **optimal fixedpoint** is then introduced
'in Part II) as the formalization of our intuitive notion of the "best solution"
of recursive definitions. The existence of a unique optimal fixedpoint for any
recursive definition, as well as some of its properties, are established. In Part III we consider the computability (from the point of view of recursive function theory) of the optimal fixedpoint of recursive definitions.

An informal exposition of the main ideas and philosophies of the optimal fixed-point approach is contained in [5]. A more complete investigation of the various fixedpoints (including the optimal fixedpoint) or recursive definitions appears in [9]. Results which are somewhat related to this work have been obtained by Myhill [6], who investigated ways in which total functions can be defined by systems of formulae.

PART I. SOME STRUCTURAL PROPERTIES OF THE SET OF FIXEDPOINTS

In this part we introduce our terminology and prove those structural properties of the set of fixedpoints of recursive definitions which are needed in Part II.

A. Basic Definitions

Let $D^+$ be a domain of defined values $D$ to which the "undefined element" $\omega$ is added. The identity relation over $D^+$ is denoted by $\equiv$. The set of all mappings of $(D^+)^n$ into $D^+$ is called the set of partial functions of $n$ arguments over $D$, and is denoted by $PF(D,n)$.

The binary relation "less defined or equal," $\subseteq$, over various domains plays a fundamental role in the theory.

Definitions:

(a) For $x,y \in D^+$, $x \subseteq y$ if $x \equiv \omega$ or $x \equiv y$.

(b) For $\bar{x}, \bar{y} \in (D^+)^n$, $\bar{x} \subseteq \bar{y}$ if $x_i \subseteq y_i$ for all $1 \leq i \leq n$.

(c) For $f_1,f_2 \in PF(D,n)$, $f_1 \subseteq f_2$ if $f_1(\bar{x}) \subseteq f_2(\bar{x})$ for every $\bar{x} \in (D^+)^n$.

(d) A function $f \in PF(D,n)$ is monotonic if $\bar{x} \subseteq \bar{y} \Rightarrow f(\bar{x}) \subseteq f(\bar{y})$.

The relation $\subseteq$ is a partial ordering of $PF(D,n)$. We shall henceforth use
the standard terminology concerning partially ordered sets. In particular:

**Definitions:** For any subset $S$ of $PF(D,n)$:

(a) $f \in S$ is the **least element** of $S$ if $f \subseteq g$ for any $g \in S$.

(b) $f \in S$ is a **minimal element** of $S$ if there is no $g \in S$ which satisfies $g \sqsubset f$.

(c) $f \in PF(D,n)$ is an **upper bound** of $S$ if $g \subseteq f$ for all $g \in S$.

(d) $f \in PF(D,n)$ is the **least upper bound (lub)** of $S$ if $f$ is the least element in the set of upper bounds of $S$.

The notions of the **greatest element**, a **maximal element**, a **lower bound** and the **greatest lower bound (glb)** of $S$ are dually defined.

**Definitions:**

(a) $f,g \in PF(D,n)$ are **consistent** if $f(\overline{x}) \neq \omega$ and $g(\overline{x}) \neq \omega \Rightarrow f(\overline{x}) = g(\overline{x})$ for every $\overline{x} \in (D^+)^n$.

(b) A subset $S$ of $PF(D,n)$ is **consistent** if every two functions, $f,g \in S$ are consistent.

From the definition it follows that:

(i) A subset $S$ of $PF(D,n)$ has a lub, denoted by $\text{lub } S$, if and only if $S$ is consistent.

(ii) Every non-empty subset $S$ of $PF(D,n)$ has a glb, which is denoted by $\text{glb } S$.

**Definitions:**

(a) A **functional** is a mapping of $PF(D,n)$ into $PF(D,n)$.

(b) A functional $\tau$ over $PF(D,n)$ is **monotonic** if $f \subseteq g \Rightarrow \tau[f] \subseteq \tau[g]$ for every $f,g \in PF(D,n)$.

(c) A recursive definition is of the form $F(\overline{x}) \triangleq \tau[F](\overline{x})$, where $\tau$ is a functional and $F$ is a function variable.
All the functionals we shall deal with in this paper will be monotonic over \( PF(D,n) \). In practice, there are many types of functionals which are monotonic only over a certain subset \( S \) of \( PF(D,n) \). The theory developed in this paper can be applied to any such restricted functional, provided that \( S \) satisfies the following two conditions:

(i) any consistent subset of \( S \) has a lub in \( S \), and
(ii) any non-empty subset of \( S \) has a glb in \( S \).

For simplicity, we do not consider in this part functions over multiple domains (e.g., \( D_1^+ \times \ldots \times D_n^+ \rightarrow D^+ \)) or systems of functionals (e.g., \( (\tau_1, \ldots, \tau_k) \)). However, all the results can be extended easily to the more general cases.

**B. Fixedpoints, Pre-fixedpoints, and Post-fixedpoints**

**Definition:** A function \( f \in PF(D,n) \) is a fixedpoint, pre-fixedpoint, or post-fixedpoint of \( \tau \) if \( f \equiv \tau[f] \), \( f \subseteq \tau[f] \), or \( \tau[f] \subseteq f \), respectively.

The sets of all fixedpoints, pre-fixedpoints, or post-fixedpoints of \( \tau \) are denoted by \( \text{FXP}(\tau) \), \( \text{PRE}(\tau) \) or \( \text{POST}(\tau) \), respectively.

Clearly \( \text{FXP}(\tau) = \text{PRE}(\tau) \cap \text{POST}(\tau) \). A few useful properties of these sets for a monotonic functional \( \tau \) are:

(i) \( \text{FXP}(\tau) \), \( \text{PRE}(\tau) \), and \( \text{POST}(\tau) \) are closed under the application of \( \tau \).
(ii) If \( S \subseteq \text{PRE}(\tau) \) is consistent, then \( \text{lub} S \in \text{PRE}(\tau) \).
(iii) If \( S \subseteq \text{POST}(\tau) \) is non-empty, then \( \text{glb} S \in \text{POST}(\tau) \).

The most important property of pre- and post-fixedpoints is that they enable us to uniformly approach a fixedpoint of \( \tau \), either by monotonically ascending
or-by monotonically descending to it. The theoretical background of this process is contained in the theorem:

**Theorem 1** (Hitchcock and Park): Let \((S, \preceq)\) be a partially ordered set, with a least element \(\Omega\), and such that any totally ordered subset has a lub. Then for any monotonic mapping \(\tau : S \to S\), the set of fixedpoints of \(\tau\) contains a least element.

A formal proof, using a transfinite sequence of approximations \(\tau^{(\lambda)}(\Omega)\) which converges to the least fixedpoint of \(\tau\), appears in Hitchcock and Park[1]. An immediate corollary of Theorem 1 is:

**Theorem 2:** For monotonic functional \(\tau:\)

(a) \(\text{FXP}(\tau)\) contains a least element, denoted by \(\text{lfixp}(\tau)\).

(b) If \(f \in \text{PRE}(\tau)\) then the set \((f' \in \text{FXP}(\tau) \mid f \preceq f')\) contains a least element.

(c) If \(f \in \text{POST}(\tau)\) then the set \((f' \in \text{FXP}(\tau) \mid f' \preceq f)\) contains a greatest element.

**Proof:**

(a) Immediate by Theorem 1, taking \(\text{PF}(D,n)\) as \(S\), \(\preceq\) as \(\preceq\), and the totally undefined function as \(\Omega\).

(b) Define \(S_f \equiv (f' \in \text{PF}(D,n) \mid f \preceq f')\). \(S_f\) is partially ordered by \(\preceq\), and contains \(f\) as its least element. Since any totally ordered subset \(s\) of \(S_f\) is consistent, \(\text{lub} S\) exists. Furthermore, \(\text{lub} S \in S_f\) since \(f \preceq \text{lub} S\).

The given monotonic functional \(\tau\) maps \(\text{PF}(D,n)\) into \(\text{PF}(D,n)\). It is easy to show that \(\tau\) maps \(S_f\) into itself. Therefore, we may...
consider the monotonic functional \( \tau' \) mapping \( S_f \) into \( S_f' \), which is the restriction of \( \tau \) to \( S_f' \). Theorem 1 ensures the existence of a least fixedpoint for \( \tau' \), which is exactly the fixedpoint required.

(c) Using the reverse order, i.e., \( f_1 \preceq f_2 \iff f_2 \subseteq f_1 \), a proof dual to the proof of part (b) can be obtained. Q.E.D.

**Definition:** A fixedpoint \( f \) of \( \tau \) is **FXP-consistent** if for any \( f' \in \text{FXP}(\tau) \), \( f \) and \( f' \) are consistent. The set of all FXP-consistent fixedpoints of \( \tau \) is denoted by \( \text{FXPC}(\tau) \).

From the definition, it follows that for any monotonic functional \( \tau \):

(a) Since \( \text{lfxp}(\tau) \) is FXP-consistent, \( \text{FXPC}(\tau) \) is non-empty.

(b) Since any two FXP-consistent fixedpoints are consistent, \( \text{FXPC}(\tau) \) is consistent, and thus \( \text{lub} \ \text{FXPC}(\tau) \) exists.

**Theorem 3:** For a monotonic functional \( \tau \), \( \text{FXPC}(\tau) \) contains a greatest element.

**Proof:** We know that \( f_1 \equiv \text{lub} \ \text{FXPC}(\tau) \) exists. As a lub of fixedpoints, \( f_1 \in \text{PRE}(\tau) \). Thus, by Theorem 2b, the set \( \{ f' \in \text{FXP}(\tau) | f_1 \subseteq f' \} \) contains a least element, say \( f_2 \). We show now that \( f_2 \in \text{FXPC}(\tau) \), implying that \( f_2 \) is the greatest function in \( \text{FXPC}(\tau) \).

Let \( g \) be any fixedpoint of \( \tau \). We would like to prove that \( f_2 \) and \( g \) are consistent, by showing the existence of a function \( f_3 \) such that \( f_2 \subseteq f_3 \) and \( g \subseteq f_3 \). The set of fixedpoints \( S = \text{FXPC}(\tau) \cup \{ g \} \) is consistent by the definition of \( \text{FXPC}(\tau) \), and therefore by Theorem 2b again there exists some \( f_3 \in \text{FXP}(\tau) \) such that \( \text{lub} \ S \subseteq f_3 \). Thus, \( g \subseteq f_3 \) and \( \text{lub} \ \text{FXPC}(\tau) \subseteq f_3 \). Since \( f_2 \) was defined as the least fixedpoint
such that \( \text{lub } FXPC(\tau) \subseteq f_2 \), we have \( f_2 \subseteq f_3 \)  

Q.E.D.

C. Maximal Fixedpoints

**Definition:** A fixedpoint \( f \) of a functional \( \tau \) is said to be **maximal** if there is no other fixedpoint \( g \) which satisfies \( f \subseteq g \). The set of all maximal fixedpoints of \( \tau \) is denoted by \( \text{MAX}(\tau) \).

Unlike the case of minimal fixedpoints, a monotonic functional may have any number of maximal fixedpoints. \( \text{MAX}(\tau) \) "covers" \( FXP(\tau) \) in the sense that:

**Theorem 4:** For monotonic functional \( \tau^- \), if \( f \in \text{PRE}(\tau^-) \) then \( f \subseteq g \) for some \( g \in \text{MAX}(\tau^-) \).

In other words, if \( f(d) = c \) for some \( f \in \text{PRE}(\tau^-) \), \( d \in (D^+)^n \) and \( c \in D \), then there must exist \( g \in \text{MAX}(\tau^-) \) such that \( g(d) \cup c \).

**Proof:** Let \( S_f = \{ f' \in FXP(\tau^-) \mid f \subseteq f' \} \). By Theorem 2b, \( S_f \) contains at least one element - the least fixedpoint which is more defined than \( f \).

We now show that \( S_f \) contains an upper bound for any totally ordered subset. Let \( S \) be such a subset. Since it is totally ordered, it is in particular consistent and thus \( \text{lub } S \) exists. Furthermore, as an lub of fixedpoints, \( \text{lub } S \) is a pre-fixedpoint. Using Theorem 2b once more, there is a fixedpoint \( f_1 \) which is more defined than \( \text{lub } S \), i.e., which is an upper bound of \( S \). By the definition of \( S \) and \( S_f, f_1 \in S_f \) and thus \( S \) has an upper bound in \( S_f \).

We have thus shown that \( S_f \) is non-empty and contains an upper bound for any totally ordered subset in it. By Zorn's Lemma, any partially ordered set having these two properties contains a maximal element. This maximal
element \( g \) is clearly a maximal fixedpoint of \( \tau \), and \( f \sqsubseteq g \) by the definition of \( S_f \). Q.E.D.

As a result of Theorem 4, we obtain

**Corollary:** For any monotonic functional \( \tau \), \( \text{MAX}(\tau) \) is non-empty.

**Proof:** Follows by the fact that \( \text{PRE}(\tau) \) is non-empty, since the totally undefined function \( \Omega \) is always in \( \text{PRE}(\tau) \). Q.E.D.

We also have

**Theorem 5:** For a monotonic functional \( \tau \), if \( f \in \text{PRE}(\tau) \) and \( g \in \text{MAX}(\tau) \), then either \( f \sqsubseteq g \) or \( f \) and \( g \) are not consistent.

**Proof:** By contradiction. Suppose \( f \not\sqsubseteq g \), and \( f \) and \( g \) are consistent. Then \( f_1 \equiv \text{lub}\{f,g\} \) exists and \( \in f_1 \in \text{PRE}(\tau) \). Thus by Theorem 2b there is a fixedpoint \( f_2 \) such that \( f_1 \sqsubseteq f_2 \). Therefore, \( g \sqsupseteq f_2 \), which contradicts the maximality of \( g \). Q.E.D.

From Theorem 5 we obtain

**Corollary:** Any two distinct maximal fixedpoints of \( \tau \) are not consistent.

**Proof:** If \( f, g \in \text{MAX}(\tau) \), then in particular \( f \in \text{PRE}(\tau) \) and we can thus apply Theorem 5. The possibility \( f \sqsubseteq g \) is ruled out by the maximality of \( f \), and thus \( f \) and \( g \) are non-consistent. Q.E.D.

**PART II-. THE OPTIMAL FIXEDPOINT**

**A. Definition and Properties**

By its definition, an FXP-consistent fixedpoint is a function which agrees in value with every other fixedpoint of \( \tau \) for any argument. In particular,
if such a fixedpoint has a defined value \( c \) at argument \( d \), then there can be no fixedpoint of \( \tau \) which has a different defined value \( c' \) at \( \overline{d} \). This value \( c \) is then said to be **weakly defined** by \( \tau \) at \( \overline{d} \) (it is not "strongly defined," however, since there may be fixedpoints that are not defined at all at \( \overline{d} \)). A fixedpoint which is not FXP-consistent, on the other hand, represents some random selection of values from the many which are possible. It is in this sense that we may say that a recursive definition really "well defines" only its FXP-consistent solutions.

Among these "genuine" solutions of \( \iota \), the more defined the solution, the more informative it is. Motivated by this quality criterion, we introduce our main definition:

**Definition:** The **optimal fixedpoint** of a monotonic functional \( \tau \) is its greatest FXP-consistent fixedpoint. It is denoted by \( \text{opt}(\tau) \).

Note that Theorem 3 guarantees the existence of the (uniquely defined) optimal fixedpoint of any monotonic functional. Using properties of \( \text{MAX}(\tau) \), we can characterize the optimal fixedpoint from a different point of view.

**Definition:** Since \( \text{MAX}(\tau) \) is non-empty, \( \text{glb} \ \text{MAX}(\tau) \) always exists, and is denoted by \( \text{lmax}(\tau) \).

As a glb of fixedpoints, \( \text{lmax}(\tau) \in \text{POST}(\tau) \), but it is not necessarily a fixedpoint. For example, consider the following functional over \( \text{PF}(N,1) \): 

\[
\tau[F](x) : \text{if } x=0 \text{ then } F(x) \text{ else } 0 \cdot F(x-1).
\]

The fixedpoints of \( \tau \) are the totally undefined function \( \Omega \), and all the functions \( f_i \), \( i=0,1,... \), defined as:

\[
\begin{align*}
\text{l} & \quad \text{N denotes the set of natural numbers.}
\end{align*}
\]
It is clear that $\text{MAX}(\tau) = \{f_0, f_1, \ldots\}$. The glb of this set of functions is:

$$l\text{max}(\tau)(x) = \begin{cases} \omega & \text{if } x = 0 \\ \eta & \text{otherwise} \end{cases}$$

This function is not a fixedpoint of $\tau$, but is a post-fixedpoint of $\tau$. It descends to the fixedpoint $\Omega$ by repeatedly applying $\tau$ to it.

However, we show now that the function $l\text{max}(\tau)$ is closely related to $\text{opt}(\tau)$:

**Theorem 6:** For a monotonic functional $\tau$, $\text{opt}(\tau)$ is the greatest element of the set $\{f' \in \text{FXP}(\tau) \mid f' \subseteq l\text{max}(\tau)\}$.

**Proof:** Let us denote by $f_1$ the greatest element in the set. By Theorem 2c, the function $f_1$ must exist since $l\text{max}(\tau) \in \text{POST}(\tau)$. We now have to show that $\text{opt}(\tau) \subseteq f_1$ and $f_1 \subseteq \text{opt}(\tau)$.

To show $\text{opt}(\tau) \subseteq f_1$, we note that by definition, $\text{opt}(\tau)$ is consistent with any maximal fixedpoint $f$ of $\tau$. By Theorem 5, it follows that $\text{opt}(\tau) \subseteq f$. Thus, $\text{opt}(\tau)$ is a lower bound of $\text{MAX}(\tau)$, and therefore $\text{opt}(\tau) \subseteq l\text{max}(\tau) \equiv \text{glb MAX}(\tau)$. Since $f_1$ is the greatest element of $(f' \in \text{FXP}(\tau) \mid f' \subseteq l\text{max}(\tau))$ we obtain $\text{opt}(\tau) \subseteq f_1$.

We now show that $f_1 \subseteq \text{opt}(\tau)$. By the definition of $\text{opt}(\tau)$, it suffices to show that $f_1 \in \text{FXPC}(\tau)$. Let $f$ be any fixedpoint of $\tau$. Theorem 4 implies that there exists some $f_2 \in \text{MAX}(\tau)$ such that $f \subseteq f_2$. By the
definition of \( f_1 \), it follows that \( f_1 \subseteq f_2 \). Thus, \( f_2 \) is an upper bound of \( f \) and \( f_1 \), which implies that they are consistent. Since this holds for any \( f \in FXP(\tau) \), \( f_1 \in FXPC(\tau) \). Q.E.D.

The original definition of \( \text{opt}(T) \) and Theorem 6 suggest that \( \text{opt}(T) \) can be "reached" both from below (by ascending from \( \text{lfixp}(\tau) \) as high as possible in \( FXPC(\tau) \)), or from above (by descending from \( \text{MAX}(\tau) \)). This situation is illustrated by the schematic diagram of Figure 1. In our graphical representation, the set \( \{ f' \in FXP(\tau) \mid f \subseteq f' \} \) is shown as an upper cone (Figure 2A), and the set \( \{ f' \in FXP(\tau) \mid f' \subseteq f \} \) is shown as a lower cone (Figure 2B).

The following properties of \( \text{opt}(\tau) \), for a monotonic functional \( \tau \), are immediate consequences of its definition and Theorem 6:

(a) If \( \text{lfixp}(\tau) \) is a total function, then \( \text{opt}(\tau) = \text{lfixp}(\tau) \).

(b) \( \text{opt}(\tau) \in \text{MAX}(\tau) \) if and only if \( \tau \) has a unique maximal fixedpoint.

It is clear that a necessary condition for \( \text{opt}(T)(d) = c \) for some \( d \in (D^+)^n \) and \( c \in D \) is:

(i) \( f(d) \equiv w \) or \( f(d) \equiv c \) for all \( f \in FXP(\tau) \), and

(ii) \( f(d) \equiv c \) for at least one \( f \in FXP(\tau) \).

However, this condition is not sufficient, as demonstrated in the previous example:

\[ \tau[F](x): \text{if } x=0 \text{ then } F(x) \text{ else } 0 \cdot F(x-1). \]

All the fixedpoints of \( \tau \) are either undefined or defined as 0 at \( x \equiv 1 \) and there are fixedpoints which are defined at \( x \equiv 1 \), while \( \text{opt}(T)(1) \equiv w \).

\subsection*{3. Examples}

In this section we illustrate the theory presented in this part with two
Fig. 1. The fixedpoints of a recursive program
functionals. These functionals are monotonic only over the subset $\text{MON}(N,1)$ of all monotonic functions in $\text{PF}(N,1)$. Since $\text{MON}(N,1)$ satisfies the two conditions mentioned at the end of section I-A, we may restrict the discussion to the domain $\text{MON}(N,1)$ rather than $\text{PF}(N,1)$.

Example 1: Consider first the monotonic functional $\tau_1$ over $\text{MON}(N,1)$:

$$\tau_1[F](x) \begin{cases} 1 & \text{if } x = 0 \\ F(F(x-1)) & \text{else} \end{cases}.$$ 

The least fixedpoint of this functional is

$$\lfxp(\tau_1) = \begin{cases} 1 & \text{if } x = 0 \\ \omega & \text{otherwise}. \end{cases}$$ 

We would like to show that $\text{opt}(\tau_1) \equiv \lfxp(\tau_1)$. For this purpose, it suffices to find two fixedpoints $f_1, f_2 \in \text{FXP}(\tau_1)$ whose values disagree for any positive $x$. Two such functions are, for example:

$$f_1(x) = \begin{cases} 1 & \text{if } x \in N \\ \omega & \text{if } x = \omega \end{cases}$$

and

$$f_2(x) = \begin{cases} x+1 & \text{if } x \in N \\ \omega & \text{if } x = \omega \end{cases}.$$ 

Thus both $\text{opt}(\tau_1)$ and $\text{imax}(\tau_1)$ cannot be defined for any positive integer $x$; since $f(w) \equiv \omega$ for any $f \in \text{FXP}(\tau_1)$, we finally obtain that $\text{opt}(\tau_1) \equiv \text{imax}(\tau_1) \equiv \lfxp(\tau_1)$.

Since $\lfxp(\tau_1)$ and $\text{opt}(\tau_1)$ are the least and greatest elements of $\text{FXPC}(\tau_1)$, $\lfxp(\tau_1)$ is clearly the only element of $\text{FXPC}(\tau_1)$.

The functions $f_1$ and $f_2$ above are maximal, since they cannot be extended at $x = \omega$. It is quite an instructive exercise to characterize all the maximal fixedpoints of $\tau_1$. For example, it can be easily shown that any maximal fixedpoint other than $f_2$ is a total, ultimately periodic function.
over-N.

Example 2: Let us consider now the functional \( \tau_2 \), defined over the same domain:

\[
\tau_2[F](x) : \text{if } x=0 \text{ then } 1 \text{ else } 2F(F(x-1)).
\]

One can easily show that \( \text{lexp}(\tau_2) \equiv \text{lexp}(\tau_1) \). The fixedpoint \( \text{opt}(\tau_2) \) cannot be obtained by the technique used in the previous example, since no appropriate fixedpoints \( f_1 \) and \( f_2 \) can be found. As a matter of fact, this functional has exactly three fixedpoints:

\[
\begin{align*}
f_1(x) &\equiv \begin{cases} 
1 & \text{if } x = 0 \\
0 & \text{otherwise}
\end{cases} \\
f_2(x) &\equiv \begin{cases} 
1 & \text{if } x = 0 \\
2 & \text{if } x = 1 \\
1 & \text{if } x = 2 \\
1 & \text{otherwise}
\end{cases} \\
f_3(x) &\equiv \begin{cases} 
2 & \text{if } x = 0 \\
1 & \text{if } x = 3 \\
4 & \text{if } x = 3i+1 \\
0 & \text{if } x = 3i+2 \text{ or } 3i+3, \text{ for } i = 0,1,2,...
\end{cases}
\end{align*}
\]

These fixedpoints are related by \( f_1 \preceq f_2 \preceq f_3 \), and therefore

\[
\begin{align*}
\text{lexp}(\tau_2) &\equiv f_1 \\
\text{opt}(\tau_2) &\equiv \text{lexp}(\tau_2) \equiv f_3 \\
\text{MAX}(\tau_2) &\equiv \{f_3\} \\
\text{FXPC}(\tau_2) &\equiv \text{FXP}(\tau_2) \equiv \{f_1, f_2, f_3\}.
\end{align*}
\]

PART III. THE COMPUTABILITY OF OPTIMAL FIXEDPOINTS.

In this part we state several results concerning the computability of optimal fixedpoints over the natural numbers. In our constructions we shall use systems of functionals \( \Phi = (\tau_1, \ldots, \tau_k) \), where each \( \tau_i \) is a monotonic functional mapping any \( k \)-tuple \( (f_1, \ldots, f_k) \) of partial functions into a partial
function \( \tau_k[f_1, \ldots, f_k] \). Thus, \( \tau \) maps any k-tuple \((f_1, \ldots, f_k)\) of partial functions into the k-tuple \((\tau_1[f_1, \ldots, f_k], \ldots, \tau_k[f_1, \ldots, f_k])\); it represents a system of recursive definitions of the form

\[
\begin{align*}
\tau_1(x) &<\tau_1[F_1(x)], \\
\tau_k(x) &<\tau_k[F_1(x), \ldots, F_k(x)].
\end{align*}
\]

A fixedpoint of \( \tau \) is now defined as a k-tuple \((f_1, \ldots, f_k)\) mapped by \( \tau \) to itself. We shall be interested in the computability of the function \( f_1 \) appearing as the first element in such a tuple (this function is usually called the main function; the others are called the auxiliary functions).

All the definitions and results contained in parts I and II of the paper can be extended easily to this general case.

We first show that the collection of optimal fixedpoints of recursive definitions over the natural numbers contains as main functions all the partial computable functions:

**Theorem 7:** Any partial recursive function \( \sigma_i \) over the natural numbers is the optimal fixedpoint of some effectively constructable system of recursive definitions.

**Proof:** Any partial recursive function can be computed by a counter machine with two counters (cf. Hopcroft and Ullman [2], page 98). Such a machine can be simulated by a system of recursive definitions in the following way.

The input value is stored in variable \( x_0 \), and with each counter \( c_i \) \((i=1,2)\) is associated a variable \( x_1 \). The main recursive definition which initializes the counters is

\[
F_1(x) < F_2(x,0,0).
\]

The function variables \( F_2, \ldots, F_k \) correspond to the states \( q_2, \ldots, q_k \) of
the counter machine. The i-th recursive definition is either of the form

\[ F_i(x_0, x_1, x_2) \leq \text{if } x_0 = 0 \text{ then } x_1 \text{ else } F_m(x''_0, x'_1, x''_2), \]

or of the form (for \( j = 1, 2 \))

\[ F_i(x_0, x_1, x_2) \leq \text{if } x_j = 0 \text{ then } F_n(x'_0, x'_1, x'_2) \text{ else } F_m(x''_0, x'_1, x''_2), \]

where the indexes \( n, m \) are chosen according to the state to which the counter machine transits when it is in state \( q_i \), and counter \( c_j \) has the respective value (zero or non-zero). Each transformed variable \( x' \) or \( x'' \) stands for either \( x+1 \) or \( x-1 \), according to the operation done on the counter or the input value upon transition.

The evaluation of the least fixedpoint of this system of recursive definitions is done by repeatedly replacing a term \( F_i(x_0, x_1, x_2) \) by the appropriate term \( F_n(x'_0, x'_1, x'_2) \) or \( F_m(x''_0, x'_1, x''_2) \), thus simulating the state transitions of the counter machine. The process stops if and when a term \( F_i(x_0, x_1, x_2) \) is replaced by the term \( x_1 \) (according to a definition of the first type), and the current value of \( x_1 \) is taken as the result of computation.

Due to the simple nature of these recursive definitions, their optimal fixedpoint coincides with their least fixedpoint (the main function in which is \( \phi_i \)). To show this, define for any natural number \( c \) the following k-tuple of functions \((f^c_1, \ldots, f^c_k)\):

\[ f^c_1(x) \equiv \begin{cases} \text{c} & \text{if evaluation of } F_1(x) \text{ is non-terminating} \\ \text{y} & \text{if evaluation of } F_1(x) \text{ terminates with value } y, \end{cases} \]

and similarly, for \( i \geq 2 \):

\[ f^c_i(x_0, x_1, x_2) \equiv \begin{cases} \text{c} & \text{if evaluation of } F_i(x_0, x_1, x_2) \text{ is non-terminating} \\ \text{y} & \text{if evaluation of } F_i(x_0, x_1, x_2) \text{ terminates with value } y. \end{cases} \]

For any \( c \), the k-tuple \((f^c_1, \ldots, f^c_k)\) so defined is a fixedpoint of the system. It is a maximal fixedpoint by its totality. The optimal fixedpoint
(f_1, \ldots, f_k) is less defined than (f_1^c, \ldots, f_k^c) for all c, and thus f_1(x) cannot be defined if the evaluation of F_1(x) is non-terminating.

Q.E.D.

Theorem 7 shows that any function which can be defined as the main function in the least fixedpoint of an effective recursive definition (i.e., any partial recursive function) can also be defined as the main function in the optimal fixedpoint of a (perhaps different) effective recursive definition. The converse, however, is not true. To show this, it suffices to consider the following simple functional over the natural numbers:

\[ \tau[F](x) : \text{if } F(x) \equiv 1 \text{ then } h(x) \text{ else } 0, \]

where h(x) is the halting function, defined as:

\[ h(x) \equiv \begin{cases} 1 & \text{if } \varphi_x(x) \text{ is defined} \\ 0 & \text{if } \varphi_x(x) \text{ is undefined} \end{cases} \]

The function h(x) is computable, as are all the other base functions which appear in the definition. In order to find the optimal fixedpoint of \( \tau \), we analyze the possible values of F(x) for any x (there is absolutely no relation between values of F for different arguments x). The value of F(x) can always be \( \omega \) or 0, as a direct substitution shows. The value 1 is possible only if h(x) \( \equiv 1 \). Any maximal fixedpoint of \( \tau \) is a composition of values 0 and 1 (only if legal) for the various arguments x. The optimal fixedpoint is then defined as 0 whenever only 0 is a possible value, while it is \( \omega \) whenever both 0 and 1 are possible values. Thus

\[ \text{opt}(\tau)(x) = \begin{cases} \omega & \text{if } \varphi_x(x) \text{ is defined} \\ 0 & \text{if } \varphi_x(x) \text{ is undefined} \end{cases} \]

and this "inverted halting function" is non-computable.

In order to see how non-computable an optimal fixedpoint may be, we prove:
Theorem 8: Let \( f(x_1, \ldots, x_n) \) be a total predicate over the natural numbers, which is the main function in the optimal fixedpoint of some system of recursive definitions \((\tau_1, \ldots, \tau_k)\). Then there is a system of recursive definitions \((\tau_1', \tau_2', \tau_3', \ldots, \tau_k')\) such that:

\[
\text{opt}(\tau_1')(x_2, \ldots, x_n) \equiv (\exists x_1 \in \mathbb{N})[f(x_1, x_2, \ldots, x_n)]
\]

Proof: The two additional recursive definitions \(\tau_1\) and \(\tau_2\) are given by:

\[
\begin{align*}
F_1(x_2, \ldots, x_n) &\leq F_2(0, x_2, \ldots, x_n) \\
F_2(x_1, x_2, \ldots, x_n) &\leq \text{if } F_3(x_1, x_2, \ldots, x_n) > 0 \text{ then } 1 \text{ else } 2 \cdot F_2(x_1 + 1, x_2, \ldots, x_n)
\end{align*}
\]

The first definition simply initializes the search conducted by the second definition for a value of \(x_1\) for which \(F_3(x_1, x_2, \ldots, x_n)\) is non-zero (true). Such a sequential search is legal, because we assume that in the optimal fixedpoint \(F(x_1, x_2, \ldots, x_n)\) represents a total function. If this search is successful, \(F_2(0, x_2, \ldots, x_n)\) (which is the value returned by the main definition \(\tau_1\)) is 2 to the power of the first such \(x_1\) found, and this value is clearly non-zero.

If no such value \(x_1\) can be found, we claim that the only two possible values of fixedpoints for \(F_2(0, x_2, \ldots, x_n)\) are 0 and 0. The fact that these are possible values is shown by direct evaluation. Suppose now that there is some other possible defined value \(c\). This value should satisfy \(c = 2^{x_1} \cdot F_2(x_1 + 1, \ldots, x_n)\) for any natural number \(x_1\). If \(c > 0\), this cannot hold if \(x_1\) is sufficiently large, no matter what the value of \(F_2(x_1 + 1, \ldots, x_n)\) is. Thus by the definition of the optimal fixedpoint, \(\text{opt}(\tau_1')(x_2, \ldots, x_n) \equiv 0\) in this case.

Q.E.D.

We can now prove:

Theorem 9: Any (total) predicate \(f(x_1, \ldots, x_n)\) in the arithmetic hierarchy of

\[1\] We assume that the truth value \text{false}/\text{true} of the predicate is determined by a zero/non-zero value of \(f\).
predicates over natural numbers can be defined as the main function in the optimal fixedpoint of some system of recursive definitions.

Proof: Any such predicate \( f \) can be expressed by (see, for example, Rogers [7])

\[
f(x_{i+1}, \ldots, x_k) := (\exists x_1)(\exists x_{i-1}) \ldots (\exists x_1)[\varphi_j(x_1, \ldots, x_i, x_{i+1}, \ldots, x_k)],
\]
or by

\[
f(x_{i+1}, \ldots, x_k) : (\sim \exists x_1)(\exists x_{i-1}) \ldots (\exists x_1)[\varphi_j(x_1, \ldots, x_i, x_{i+1}, \ldots, x_k)],
\]

where

\[
\varphi_j(x_1, \ldots, x_k) \text{ is a recursive predicate.}
\]

These two forms can be constructed in the following way. First a system which defines the recursive function \( \varphi_j(x_1, \ldots, x_k) \) is constructed (by its totality, one need not use the method described in Theorem 7 - any system of recursive definitions which yields \( \varphi_j \) as least fixedpoint also yield it as optimal fixedpoint). Then the pair of recursive definitions described in Theorem 8 is added for each existential quantifier, from right to left.

The only change one should make in each pair in order to handle the negation sign is to change the predicate \( F_3(x_1, \ldots, x_n) \geq 0 \) into \( F_3(x_1, \ldots, x_n) = 0 \); thus we search for values which do not satisfy the previous existential condition. Finally, if a form of the second type above should be constructed, the following main recursive definition is added:

\[
F_0(x) := \text{if } F_1(x) > 0 \text{ then } 0 \text{ else } 1,
\]

and the resultant predicate \( F_1(x) \) is thus inverted in \( F_0(x) \).

The proof that the procedure described above constructs a system of recursive definitions yielding the predicate \( f(x) \) as the main function in the optimal fixed-point is a straightforward generalization (by induction) of Theorem 8. Q.E.D.
Once we have constructed recursive definitions for all the predicates in the arithmetic hierarchy, we can also construct recursive definitions for all the partial functions whose graph $\mathcal{A}$ is a predicate of the arithmetic hierarchy.

**Theorem 10:** If $f(x)$ is a partial function with graph $g(x,y)$ in the arithmetic hierarchy, then there exists a system of recursive definitions such that the main function in its optimal fixedpoint is $f(x)$.

**Proof:** By Theorem 9, there exists a system of recursive definitions $(\tau_1, \ldots, \tau_n)$ for which the main function in the optimal fixedpoint is the (total) function $g(x,y)$. The following two recursive definitions $\tau_1$ and $\tau_2$ are added to the system ($\tau_1$ serves as the main definition):

\[
\begin{align*}
\tau_1(x) &\leftarrow F_2(x,0) \\
\tau_2(x,y) &\leftarrow \text{if } F_3(x,y) > 0 \text{ then } y \text{ else } F_2(x,y+1).
\end{align*}
\]

The proof that $F_1(x)$ really yields the desired partial function is a mixture of elements from the proofs of Theorems 7 and 8. The recursive definition $\tau_2$ conducts a search (initialized by 0) for a value $y$ which satisfies $F_3(x,y) > 0$ (i.e., for which $g(x,y)$ is true). If a value $y$ is found, it is taken as the result of computation. Otherwise, due to the simple form of $\tau_2$, any constant value $c$ can serve as a value for a fixedpoint, and thus the main function in the optimal fixedpoint is undefined. Q.E.D.

---

The graph $g(x,y)$ of a partial function $f(x)$ is a predicate defined by:

\[
g(x,y) \equiv \begin{cases} 
\text{true} & \text{if } f(x) = y, y \neq w \\
\text{false} & \text{if } f(x) \neq y, y \neq w \\
w & \text{if } y = w.
\end{cases}
\]

In particular, if $f(x)$ is undefined then $g(x,y)$ is false for all $y \neq w$. 21
References


