# Some Existence and Preservation Results for Optimal Fixpoints

Vahan Margaryan
Department of Informatics and Applied Mathematics
Yerevan State University
Yerevan, Armenia
vahan80a@netsys.am
Keywords: optimal fixpoint, recursive operator

**Abstract**

*Optimal fixpoints of recursive operators extract maximum consistent information from recursive definitions. Although the optimal fixpoint always exists for a recursive operator, it can be uncomputable. The paper considers the restriction of the recursive operator to computable inputs and the set of consistent fixpoints induced by this restriction. The properties of the greatest element of this set is studied, particularly, its relation to computable optimal fixpoint. It has been shown in previous work that the greatest consistent fixpoint of a restricted operator may differ from the computable optimal fixpoint. The article illustrates that the greatest consistent fixpoint can also disappear as a result of the restriction.*

## 1 Introduction

Optimal fixpoints were introduced by Z.Manna and A.Shamir in [1]. Optimal fixpoint extracts maximum consistent information from a recursive program. It can be contrasted with the more usual (and fundamental) notion of least fixpoint, which extracts the minimal information. Although the definition of optimal fixpoints seems to imply practical benefits, it also leads to difficulties. Thus, optimal fixpoints, although they always exist for a recursive operator, aren't necessarily computable.

We discuss the existence and properties of greatest consistent fixpoints when the recursuve operators are limited to computable inputs. The restriction is realized by introducing the notion of the optimal fixpoint of an extensional function. In [3] it has been shown that the computable optimal fixpoint of a recursive operator can differ from the optimal fixpoint of the corresponding extensional function. In general, an extensional function may not have an optimal fixpoint. Here we show that an extensional function may fail to have an optimal fixpoint even when the optimal fixpoint of the corresponding operator is computable.

## 2 Definitions and Comments

The notation and comments generally follow [3].

**Definition 1** A total recursive function $f$ is called *extensional* if

$$\forall x \forall y (\varphi_x = \varphi_y \Rightarrow \varphi_{f(x)} = \varphi_{f(y)}) \tag{1}$$

**Theorem 2** (J.Myhill, J.Shepherdson, [4]) *For every recursive operator $\Phi$ there is a corresponding extensional function $f$, such that*

$$\forall x (\varphi_{f(x)} = \Phi[\varphi_x]). \tag{2}$$

*Conversely, every extensional function $f$ uniquely determines a recursive operator, such that the above property holds.*

It follows from the theorem that extensional functions corresponding to the same recursive operators are equivalent as index functions, i.e. for any two such functions $f$ and $g$ we have $\forall x (\varphi_{f(x)} = \varphi_{g(x)})$.

Whereas the theorem establishes equivalence of transformations performed by a recursive operator and its corresponding extensional functions, the latter operate on Gödel numbers and, therefore, represent the restriction of the operator to computable functions.

We use the usual point-wise complete partial order on functions.

**Definition 3** We say that partial function $f$ *precedes* $g$ *(*symbolically: $f \preceq g$*),* if

$$\forall x(g(x)\downarrow \Rightarrow (f(x)\downarrow \;\&\; g(x) = f(x))). \quad (3)$$

The following definitions are taken from [2].

**Definition 4** Partial functions $f$ and $g$ are called *consistent* if

$$\forall x((f(x)\downarrow \;\&\; g(x)\downarrow) \Rightarrow f(x) = g(x)). \quad (4)$$

A set of functions is called *consistent,* if any two functions in the set are consistent.

For a consistent set $A$ of functions we can define a *join*, i.e. a function $f$ such that $f(x) = y \Leftrightarrow \exists g \in A$ and $g(x) = y$.

**Definition 5** Let $f$ be extensional function. If $n$ satisfies the equation $\varphi_{f(n)} = \varphi_n$, we call $\varphi_n$ a *fixpoint* of $f$.

The definition deviates from the usual practice where $n$ itself is called a fixpoint. Since $f$ is extensional, it is more natural to speak about fixpoints as functions, rather than function indices.

The set of fixpoints of a recursive operator $\Phi$ will be denoted $FIXP(\Phi)$. Similarly, $FIXP(f)$ denotes the set of fixpoints of extensional function $f$.

**Definition 6** A fixpoint of $\Phi$ *(*respectively, $f$*)* is called *fxp-consistent* if it is consistent with all the fixpoints of $\Phi$ *(resp.* $f$*)*.

**Definition 7** If the set of fxp-consistent fixpoints of $\Phi$ *(*respectively, $f$*)* has a greatest element, it is called the *optimal fixpoint* of $\Phi$ *(resp.* $f$*)*.

**Theorem 8** (Z. Manna, and A. Shamir, [1]) *Every recursive operator has an optimal fixpoint.*

Thus, optimal fixpoint, like the least fixpoint, is uniquely determined for a recursive operator. However, it can be uncomputable.

We define $def$ to be the following recursive operator:

$$def[f](x) = sgn(f(x) + 1) \quad (5)$$

$def(f)$ is thus a function that is 1 whenever $f(x)$ is defined and undefined otherwise.

# 3 Optimal fixpoints of extensional functions

By analogy with optimal fixpoints of recursive operators, we aim to study the existence and properties of optimal fixpoints of extensional functions. In [3], the following general existence result has been obtained.

**Proposition 9** *There exists an extensional function that does not have an optimal fixpoint.*

The proof of the proposition constructs a recursive operator that has an uncomputable optimal fixpoint. The corresponding extensional function doesn't have an optimal fixpoint. Based on the proof, it is easy to construct an operator $\Phi$ that has an uncomputable optimal fixpoint, whereas the corresponding $f$ has an optimal fixpoint.

Additionally, it can be shown that when $f$ does not have an optimal fixpoint, the set of its fxp-consistent fixpoints doesn't have maximal elements.

We further consider operators with computable optimal fixpoints. Examples show that, unsurprisingly, computable optimal fixpoints often coincide with the optimal fixpoints of the corresponding extensional functions. The following proposition from [3], however, shows that they can differ.

**Proposition 10** *There exists a recursive operator such that its optimal fixpoint is computable, the corresponding extensional function also has an optimal fixpoint, but they differ.*

The remaining question in this context is whether an operator with a computable optimal fixpoint can give rise to extensional function that does not have an optimal fixpoint.

The following propositions indicate that it's not so easy to make the optimal fixpoint disappear.

**Proposition 11** *If the join of fxp-consistent fixpoints of an extensional function is a computable function, then the optimal fixpoint exists.*

**Corollary 12** *If the extensional function has a finite number of computable fixpoints, the optimal fixpoint exists.*

Therefore, to illustrate the possibility of disappearance of a computable optimal fixpoint we need to have an extensional function with an infinite number of fxp-consistent computable fixpoints. The join of these has to be uncomputable, and yet the optimal fixpoint of the corresponding recursive operator has to be computable (because that is the case we are interested in).

The construction of such an example that we have obtained depends on a few basic tricks. In what follows, we construct a new recursive operator from given operators so that the corresponding sets of fixpoints are combined in some desirable way.

**Union of recursive operators.** Assume we have recursive operators - $\Psi$ and $\Phi$. Let $\Psi$ act on $f$ and $\Phi$ act on $g$, yielding again a pair of functions $\Psi[f]$ and $\Phi[g]$. If we obtain $f$ and $g$ as a result, then this pair is a fixpoint of the pair of operators. It follows from the definition that the set of fixpoints of the pair is precisely the cartesian product of sets of fixpoints for individual operators.

We want to construct a single recursive operator that intuitively corresponds to this construction. We define the disjoint union of functions $f$ and $g$ as follows:

$$f \sqcup g = \begin{array}{ll} \text{f(k)}, & \text{if } x = 2 * k; \\ \text{g(k)}, & \text{if } x = 2 * k + 1. \end{array} \tag{6}$$

The union operator for two operators $\Psi$ and $\Phi$ is then defined as follows:

$$(\Psi \sqcup \Phi)[f \sqcup g] = \Psi[f] \sqcup \Phi[g] \tag{7}$$

That this equation defines a recursive operator is intuitively clear and can be verified by considering the corresponding enumeration operators (the Gödel number of the enumeration operator corresponding to $\Psi \sqcup \Phi$ uniformly depends on the Gödel numbers of the enumeration operators corresponding to $\Psi$ and $\Phi$).

If $f_0$ is a fixpoint of $\Psi$ and $g_0$ is a fixpoint of $\Phi$, then $f_0 \sqcup g_0$ is a fixpoint of $\Psi \sqcup \Phi$. Conversely, such pairs exhaust the set of fixpoints of $\Psi \sqcup \Phi$.

**Parameterized recursive operator.** We define recursive operator $\Psi[f, g]$ from two functional arguments as follows:

$$\Psi[f, g] = \Omega[f \sqcup g] \tag{8}$$

where $\Omega$ is some recursive operator.

The first argument of the operator will serve as a parameter. We are interested in the fixpoints of this operator, i.e. for given $f$ we seek a $g$ such that $\Psi[f, g] = g$.

Parameterized recursive operator is monotone and continuous relative to each of its arguments.

**Dependent union.** An extension of the union construction is *dependent* union:

$$(\Psi \triangleright \Phi)[f \sqcup g] = \Psi[f] \sqcup \Phi[f, g]$$

Here $\Phi$ is a parameterized recursive operator.

Again, dependent union of a recursive operator and parameterized recursive operator is a recursive operator. The fixpoints of a dependent union operator are of the form $f_i \sqcup h_i$, where $f_i$ is a fixpoint of $\Psi$, since $\Psi$ doesn't depend on $g$. $\Phi$, on the other hand, depends both $f$ and $g$. Therefore, the properties of $h_i$ can be made to depend on certain properties of fixpoint $f_i$.

Dependent union can be used to adjust the set of the fixpoints of a union of operators. We are interested in the following construction. Given recursive operators $\Psi$ and $\Phi$, we want to eliminate from the set of fixpoints of $\Psi \sqcup \Phi$ the fixpoints $f \sqcup g$ such that $f(r_0)$ is undefined and $g(x)$ is defined for some $x$. To do that we construct $\Theta_\Phi$ as follows:

$$\Theta_\Phi[f, g] = \Phi[g] * def[f](r_0)$$

Thus $\Theta_\Phi = \Phi$ whenever $f(r_0)$ is defined and $\Theta_\Phi = \lambda h \lambda x \bot$ otherwise.

We then consider operator $\Psi \triangleright \Theta_\Psi$ and its fixpoints. It follows from the construction that those will be of the form $f \sqcup g$, where $f$ is a fixpoint of $\Psi$ and $g$ is a fixpoint of $\Phi$, if $f(r_0)$ is defined, and totally undefined function otherwise.

We can now use this to construct the example.

**Proposition 13** *There exists a recursive operator such that its optimal fixpoint is computable but the corresponding extensional function doesn't possess an optimal fixpoint.*

**Proof.**

In [3] a recursive operator $\Psi$ is constructed that has a set of 4 fixpoints. For a certain number $r_0$ and uncomputable function $U(x)$, those are:

- $f_1(x) = 1$ if $x = r_0$, $U(x)$ elsewhere;
- $f_2(x) = 0$ if $x = r_0$, $U(x)$ elsewhere;
- $f_3(x) = 1$ if $x = r_0$, undefined elsewhere;
- $f_4(x) = \perp$;

Of these, $f_1$ and $f_2$ are uncomputable, while $f_3$ and $f_4$ are computable.

Another recursive operator constructed there is $\Phi$. This has an infinite set of fixpoints, all of which are consistent. They can be categorized into two groups:

- $H$ - uncomputable fixpoints;
- $G$ - computable fixpoints. The set $G$ doesn't have a greatest element. The optimal fixpoint of $\Phi$ is uncomputable (i.e. belongs to $H$).

Using the construction of $\Theta$ discussed above on top of $\Phi$, we consider operator $\Psi \triangleright \Theta_\Phi$. $f_4$ isn't defined at $r_0$, so it will only pair with an undefined fixpoint. The resultant set of fixpoints, along with the computability designations, is as follows:

- $f_1 \sqcup H$ - not computable
- $f_1 \sqcup G$ - not computable
- $f_2 \sqcup H$ - not computable
- $f_2 \sqcup G$ - not computable
- $f_3 \sqcup H$ - not computable
- $f_3 \sqcup G$ - computable
- $\lambda x \perp$

Since $f_1$, $f_2$, $f_3$ are an inconsistent set (they take values 0 and 1 at $r_0$), the optimal fixpoint is $\lambda x \perp$. It is thus computable.

If we restrict attention to the computable fixpoints, $f_3 \sqcup G$ and $\lambda x \perp$ will remain. This set is consistent and doesn't have a greatest element (because $G$ doesn't). Therefore, the optimal fixpoint doesn't exist.

■

# References


[1] Z. Manna, A. Shamir. The theoretical aspects of the optimal fixpoint semantics and correctness of programs. *SIAM Journal of Computing* 5(3):414-426.

[2] Z. Manna, A. Shamir. The optimal approach to recursive programs. *Communications of the ACM* 11:824-831

[3] V. Margaryan. On certain properties of computable optimal fixpoints. *Proceedings of CSIT-2003*,

[4] J. Myhill, J. C. Shepherdson. Effective operations on partial recursive functions, *Z. math. Logik und Grundl. Math.*, 1955, 1, 310-317.