Approximate Fixed Points in Abstract Interpretation

Chris Hankin and Sebastian Hunt
Department of Computing
Imperial College of Science, Technology and Medicine
180 Queen's Gate, London SW7 2BZ

Abstract

Much of the earlier development of abstract interpretation, and its application to imperative programming languages, has concerned techniques for finding fixed points in large (often infinite) lattices. The standard approach in the abstract interpretation of functional languages has been to work with small, finite lattices and this supposedly circumvents the need for such techniques. However, practical experience has shown that, in the presence of higher order functions, the lattices soon become too large (although still finite) for the fixed-point finding problem to be tractable. This paper develops some approximation techniques which were first proposed by Hunt and shows how these techniques relate to the earlier use of widening and narrowing operations by the Cousots.

1. Introduction

Any account of abstract interpretation of functional languages must address the problem of defining a suitable abstraction of functional values. There are a number of alternatives, ranging from the relational approach espoused in the Cousot's work and instantiated in the minimal function graph approach of [Jon85] to the approach of [BHA86] where functions are abstracted by functions. In the following we will adopt the latter approach. In such a setting it is well-known that the problem of finding fixed points, the central operation in abstract interpretation, is of n-iterated exponential complexity [Mey85]. There was a naive expectation that the development of clever algorithms such as the frontiers algorithm [CPJ85], [MH87], [HH91] would ameliorate this situation but practical experience has shown that this was misplaced optimism.

In [Hun89] and [HH91], we developed a formal approach to allow the evaluation of approximate fixed points, in fact generating upper and lower bounds for the true fixed point. In the classical approach to abstract interpretation pioneered by Patrick and Radhia Cousot it is common to work with lattices that do not satisfy the ascending chain condition and, in this context, it is essential to work with approximate fixed points; they have developed a general theory of widening and narrowing operations to support this work. It is possible to relate our approach to the widening/narrowing approach of the Cousots and this is our programme in this paper; indeed our work constitutes the only published example of higher-order widening/narrowing.

In the next section, we review the main results from [HH91]. Section 3 develops the theory somewhat further and presents a scheme for using the approach in fixed point computation. Section 4 defines widening and narrowing operations and demonstrates the correspondence between the two approaches. We conclude with Section 5.

2. The Abstraction Ordering

We work with a family of finite lattices L:

$$\begin{array}{l} \mathbf{2} \in L \text{ where } \mathbf{2} \equiv (\{0,1\},\ 0 \leq 1) \\ D_\perp \in L \text{ if } D \in L \\ D^\top \in L \text{ if } D \in L \\ D_1 \times D_2 \in L \text{ if } D_i \in L,\ i = 1,2 \\ [D \to D'] \in L \text{ if } D, D' \in L \end{array}$$

where $[D \to D']$ is the lattice of monotonic functions from D to D'. Such a family of finite lattices has proved to be useful in a wide range of analyses including strictness analysis, parallel sharing analysis, and binding time analysis.

We define an abstraction ordering on L:

$$\begin{array}{l} \mathbf{2} \leq D \text{ for all } D \in L \\ D_\perp \leq D'_\perp \text{ if } D \leq D' \\ D^\top \leq D'^\top \text{ if } D \leq D' \\ D_1 \times D_2 \leq D'_1 \times D'_2 \text{ if } D_1 \leq D'_1 \text{ and } D_2 \leq D'_2 \\ [A \to B] \leq [A' \to B'] \text{ if } A \leq A' \text{ and } B \leq B'^{1} \end{array}$$

Notice that $\leq$ is a partial order.

[1]Note that $\leq$ does not capture the usual notion of approximation which has $\to$ contravariant in its first argument. We will see that $\leq$ means that there is a Galois connection between the two lattices - this amounts to the standard domain theoretic practice of using embedding/projection pairs to avoid the contravariance of $\to$.

We introduce the categories **FL**, $\mathbf{FL}^{ec}$ and $\mathbf{FL}^{ep}$.

FL	finite lattices, monotone maps
$\mathbf{FL}^{ec}$	finite lattices, embedding-closure pairs
$\mathbf{FL}^{ep}$	finite lattices, embedding-projection pairs[2]

We write the left and right components of an embedding-closure (embedding-projection) pair ϕ as ϕ^e and ϕ^c (ϕ^e and ϕ^p). Given an embedding-closure pair ϕ, ϕ^e and ϕ^c determine each other uniquely. Similarly for embedding-projection pairs. Thus $\mathbf{FL}^{ec}$ and $\mathbf{FL}^{ep}$ may both be viewed as sub-categories of **FL**.

We introduce the functors $\_{}_\perp$, $\_{}^\top$, $\_\times\_$ and $\_\rightarrow\_$ on $\mathbf{FL}^{ec}$ and $\mathbf{FL}^{ep}$. These functors have the same (formal) definitions for both categories. The definitions of these functors are the expected ones, in particular:

$A \rightarrow B = [A \rightarrow B]$

for $\phi : A_1 \rightarrow A_2$ and $\psi : B_1 \rightarrow B_2$,

$\phi \rightarrow \psi : [A_1 \rightarrow B_1] \rightarrow [A_2 \rightarrow B_2]$ is defined by

$(\phi \rightarrow \psi)^e f = \psi^e \circ f \circ \phi^c$

$(\phi \rightarrow \psi)^c g = \psi^c \circ g \circ \phi^e$

It is straightforward to verify that these are indeed functors on the appropriate category ($\mathbf{FL}^{ep}$ and $\mathbf{FL}^{ec}$). We next define two pairs of maps between pairs of lattices related by the abstraction ordering. The safe maps give overestimates of values and the live maps give underestimates.

Definition 2.1

For each $A \le B \in L$, we define an $\mathbf{FL}^{ec}$-morphism $\mathrm{Safe}_{A,B} : A \rightarrow B$. We write $\mathrm{UpS}_{A,B}$ for $\mathrm{Safe}_{A,B}{}^e$ and $\mathrm{DownS}_{B,A}$ for $\mathrm{Safe}_{A,B}{}^c$:

$$\begin{aligned}
\mathrm{UpS}_{2,B}\, a &= \perp_B \text{ if } a = 0 \\
&\quad \top_B \text{ if } a = 1 \\
\mathrm{DownS}_{B,2}\, b &= 0 \text{ if } b = \perp_B \\
&\quad 1 \text{ otherwise} \\
\mathrm{SafeA}_\perp{,}B_\perp &= (\mathrm{Safe}_{A,B})_\perp \\
\mathrm{Safe}_{A^\top,B^\top} &= (\mathrm{Safe}_{A,B})^\top \\
\mathrm{Safe}_{A1 \times B1, A2 \times B2} &= \mathrm{Safe}_{A1,A2} \times \mathrm{Safe}_{B1,B2}
\end{aligned}$$

[2] An embedding-closure pair is a pair of continuous functions ($e{:}A \rightarrow B, c{:}B \rightarrow A$) such that:

$e \circ c \ge id$ and $c \circ e = id$

and an embedding-projection pair is a pair of continuous functions ($e{:}A \rightarrow B, p{:}B \rightarrow A$) such that:

$e \circ p \le id$ and $p \circ e = id$.

$$\mathrm{Safe}_{[A1\to B1],[A2\to B2]} = \mathrm{Safe}_{A1,A2}\to\mathrm{Safe}_{B1,B2}$$

[]

We must verify that:

i) $\mathrm{UpS}_{2,B} \circ \mathrm{DownS}_{B,2} \geq \mathrm{id}_B$

ii) $\mathrm{DownS}_{B,2} \circ \mathrm{UpS}_{2,B} = \mathrm{id}_2$

These verifications are routine.

Definition 2.2

For each $A \leq B \in L$, we define an $\mathbf{FL}^{ep}$-morphism $\mathrm{Live}_{A,B} : A \to B$. We write $\mathrm{UpL}_{A,B}$ for $\mathrm{Live}_{A,B}{}^{e}$ and $\mathrm{DownL}_{B,A}$ for $\mathrm{Live}_{A,B}{}^{p}$. The definitions of the Live maps are (formally) identical to those of the Safe maps except for $\mathrm{DownL}_{B,2}$ which is:

$$\mathrm{DownL}_{B,2}\, b \;=\; \begin{cases} 1 \text{ if } b = \top_B \\ 0 \text{ otherwise} \end{cases}$$

[]

It is tempting to say that the Live maps are dual to the Safe maps, i.e.:

$$\mathrm{Live}_{A,B} \;=\; \mathrm{Safe}_{A^{OP},B^{OP}}$$

We cannot actually say this since the family L is not closed under the operation of forming opposites. However, we can establish an order ($\leq$) isomorphism $O : L \to L$ such that for each $A \in L$ we have:

$$O(A) \cong A^{OP}$$

Thus:

$$\begin{aligned}
O(2) &= 2\\
O(A_\perp) &= (O(A))^\top\\
O(A^\top) &= (O(A))_\perp\\
O(A\times B) &= O(A)\times O(B)\\
O([A\to B]) &= [O(A)\to O(B)]
\end{aligned}$$

Note that $O(O(A)) = A$. For each $A \in L$, the isomorphism of A^{OP} and $O(A)$ is established via the (contravariant) map $R_A: A \to O(A)$:

$$\begin{aligned}
R_2\, 0 &= 1 & R_2\, 1 &= 0\\
R_{A\perp}\, \perp &= \top & R_{A\perp}\,(\mathrm{lift}\ a) &= \mathrm{colift}(R_A\, a)\\
R_{A^\top}\, \top &= \perp & R_{A^\top}\,(\mathrm{colift}\ a) &= \mathrm{lift}(R_A\, a)\\
R_{A\times B}\,(a,b) &= (R_A\, a, R_B\, b)\\
R_{[A\to B]}\, f &= R_B \circ f \circ R_{O(A)}
\end{aligned}$$

Note that:

$$R_{O(A)} \circ R_A \quad = \quad id_A$$

A key property of the R maps is the following:

Lemma 2.3

For all A,B $\in$ L:

$$R_B(f\,a) \quad = \quad (R_{[A \to B]}\, f)\,(R_A\, a) \qquad []$$

Fact 2.4

For all $A \leq B \in L$:

i) $UpL_{O(A),O(B)} \quad = \quad R_{[A \to B]}(UpS_{A,B})$

ii) $DownL_{O(B),O(A)} \quad = \quad R_{[B \to A]}(DownS_{B,A})$ []

The following properties of the Safe and Live maps are standard for embedding-closure pairs and embedding-projection pairs [GHK*80]:

- UpS and UpL are injective
- DownS and DownL are onto and strict
- UpS is $\top$-preserving and UpL is strict.

In addition we have the following:

Lemma 2.5

For all $A \leq B \in L$, $UpS_{A,B}$ is strict.

Proof

induction on the height of the proof that $A \in L$ []

Corollary 2.6

For all $A \leq B \in L$, $UpL_{A,B}$ is $\top$-preserving.

Proof

Let $A \leq B \in L$. Since $O : L \to L$ is an order isomorphism, we can write A as O(A') and B as O(B'). Then

$UpL_{O(A'),O(B')}\ \top_{O(A')}$

$=$	$(R(UpS_{A',B'}))\ \top_{O(A')}$	by Fact 2.4
$=$	$(R(UpS_{A',B'}))\ (R\ \bot_{A'})$	since R an isomorphism of A'^{OP} onto O(A')
$=$	$R(\ UpS_{A',B'}\ \bot_{A'})$	by Lemma 2.3
$=$	$R(\ \bot_{B'}\ )$	by the Lemma 2.5
$=$	$\top_{O(B')}$	since R an isomorphism of B'^{OP} onto O(B') []

In what follows we will assume "dual" results such as this corollary to be clear and will not spell out the details of their proof.

Lemma 2.7

For all $A \leq B \leq C \in L$:

i) $\quad \mathrm{Safe}_{A,C} \quad = \quad \mathrm{Safe}_{B,C} \circ \mathrm{Safe}_{A,B}$

ii) $\quad \mathrm{Safe}_{A,A} \quad = \quad \mathrm{id}_A$

Proof

(i) proof by induction over the type of A:

$A \equiv 2$:

$$\begin{aligned}
\mathrm{UpS}_{B,C}(\mathrm{UpS}_{2,B}\ 0) &= \mathrm{UpS}_{B,C}\ \bot_B, \text{ by definition}\\
&= \bot_C, \text{ by Lemma 2.5}\\
\mathrm{UpS}_{B,C}(\mathrm{UpS}_{2,B}\ 1) &= \mathrm{UpS}_{B,C}\ \top_B, \text{ by definition}\\
&= \top_D \text{ since UpS is top-preserving}
\end{aligned}$$

$\mathrm{DownS}_{B,2}(\mathrm{DownS}_{C,B}\ x) = 0 \Rightarrow \mathrm{DownS}_{C,B}\ x = \bot_B$

$\Rightarrow \quad x = \bot_C$[3]

in this case $\mathrm{DownS}_{C,2}\ x = 0$ as well.

$\mathrm{DownS}_{B,2}(\mathrm{DownS}_{C,B}\ x) = 1 \Rightarrow \mathrm{DownS}_{C,B}\ x \neq \bot_B$, by definition

$\Rightarrow \quad x \neq \bot_C$, since DownS is strict.

and thus $\mathrm{DownS}_{C,2}\ x = 1$.

The result follows by extensionality.

The inductive cases are all the same and follow from the functorial properties of the constructors that we use; we illustrate two cases - the unary functor $\_{}_\bot$ and the binary functor $\_\times\_$:

$A \equiv A'_\bot$:

Then $B \equiv B'_\bot$ and $C \equiv C'_\bot$ and then:

$$\begin{aligned}
\mathrm{Safe}A'_\bot,C'_\bot &= (\mathrm{Safe}_{A',C'})_\bot\\
&= (\mathrm{Safe}_{B',C'} \circ \mathrm{Safe}_{A',B'})_\bot \text{ by IH}\\
&= (\mathrm{Safe}_{B',C'})_\bot \circ (\mathrm{Safe}_{A',B'})_\bot \text{ since } \_{}_\bot \text{ is a functor}\\
&= \mathrm{Safe}_{B,C} \circ \mathrm{Safe}_{A,B}
\end{aligned}$$

$A \equiv A1 \times A2$:

Then $B \equiv B1 \times B2$ and $C \equiv C1 \times C2$ and:

[3]Both implications hold because DownS is bottom-reflecting which follows by a simple argument using the properties of embedding-projection pairs.

$$\begin{aligned}
\text{Safe}_{A,C} &= \text{Safe}_{A1,C1} \times \text{Safe}_{A2,C2} \\
&= (\text{Safe}_{B1,C1} \circ \text{Safe}_{A1,B1}) \times (\text{Safe}_{B2,C2} \circ \text{Safe}_{A2,B2}) \quad \text{by IH} \\
&= (\text{Safe}_{B1,C1} \times \text{Safe}_{B2,C2}) \circ (\text{Safe}_{A1,B1} \times \text{Safe}_{A2,B2}) \\
&= \text{Safe}_{B,C} \circ \text{Safe}_{A,B}
\end{aligned}$$

(ii) follows from $\text{Safe}_{2,2} = \text{id}_2$ since functors (_×_, _→_, etc...) preserve identities.

[]

Thus we may view Safe as being a functor from the (poset) category L to $\mathbf{FL}^{ec}$. By the "dual" of the Lemma, Live is a functor from L to $\mathbf{FL}^{ep}$.

The main result concerning these maps and their interaction with the least fixed point operator, **fix**, in [HH91] is the following:

Fact 2.8

For all lattices D, D' ∈ *L* such that D' ≤ D:

(i) $\quad \mathbf{fix}_{D'} = \text{DownS}\ \mathbf{fix}_D$

$\quad\quad \mathbf{fix}_{D'} = \text{DownL}\ \mathbf{fix}_D$

(ii) $\quad \text{UpS}\ \mathbf{fix}_{D'} \geq \mathbf{fix}_D$

$\quad\quad \text{UpL}\ \mathbf{fix}_{D'} \leq \mathbf{fix}_D$ []

2.8(ii) gives a formal basis for the method of finding upper and lower bounds for a true fixed point by iterating in a smaller lattice using safe and live approximations, respectively. In this paper we develop this technique and relate it to Cousot's widening and narrowing operations.

3.Further Properties of the Abstraction Ordering

We start by noting that x is a *pre-fixed point* of f if:

$$x \leq f(x)$$

and a *post-fixed point* of f if:

$$f(x) \leq x$$

We can now state and prove one of the main results of the paper.

Proposition 3.1

Let A' ≤ A ∈ *L*, and let:

$$D \equiv [A \to A]$$
$$D' \equiv [A' \to A']$$

Then for all $f \in D$ and for all $x \in A'$:

$$(\mathrm{DownS}_{D,D'}\, f)\, x = x \quad \Rightarrow \quad f\, (\mathrm{UpS}_{A',A}\, x) \le \mathrm{UpS}_{A',A}\, x$$

Proof

$\mathrm{UpS}_{A',A}\, x$	$=$	$\mathrm{UpS}_{A',A}((\mathrm{DownS}_{D,D'}\, f)x)$	by assumption
	$=$	$(\mathrm{UpS}_{A',A} \circ \mathrm{DownS}_{A,A'} \circ f \circ \mathrm{UpS}_{A',A})x$	by definition
	$\ge$	$f\, (\mathrm{UpS}_{A',A}\, x)$	

[]

Corollary 3.2

For any $[A \to A]$, $[A' \to A']$, $[A'' \to A''] \in L$, with $A'' \le A' \le A$ then for all $g \in [A \to A]$ and any fixed point x of $(\mathrm{DownS}_{[A\to A],[A''\to A'']}\, g\,)$, we have:

$$(\mathrm{DownS}_{[A\to A],[A'\to A']}\, g)\, (\mathrm{UpS}_{A'',A'}\, x) \le \mathrm{UpS}_{A'',A'}\, x$$

Proof

Use Proposition 3.1 with $f = \mathrm{DownS}_{[A\to A],[A'\to A']}\, g$ and use Lemma 2.7 []

This result says that when any fixed point of the approximation of g in a smaller lattice is embedded into any larger lattice it becomes a post-fixed point of the approximation of g in that larger lattice. The dual result for the live maps is that any fixed point becomes a pre-fixed point in the larger lattice.

Proposition 3.3

For any D, D', D'' $\in L$, such that:

$$D \equiv [A \to A]$$
$$D' \equiv [A' \to A']$$
$$D'' \equiv [A'' \to A'']$$

with $A'' \le A' \le A$ and for any $f \in D$:

(i) $\mathrm{UpS}_{A'',A'}(\mathbf{fix}_{A''}(\mathrm{DownS}_{D,D''}\, f)) \ge \mathbf{fix}_{A'}(\mathrm{DownS}_{D,D'}\, f)$

(ii) $\mathrm{UpL}_{A'',A'}(\mathbf{fix}_{A''}(\mathrm{DownL}_{D,D''}\, f)) \le \mathbf{fix}_{A'}(\mathrm{DownL}_{D,D'}\, f)$

Proof

These follow immediately from Fact 2.8 (ii) and Lemma 2.7(i).

[]

Finally, we restate some obvious properties of pre- and post-fixed points.

Fact 3.4
For all $f \in [A \to A]$ and $x \in A$:

(i) if x is a pre-fixed point of f and x is less than $\mathbf{fix}_A f$ then $\{f^n(x) \mid n \geq 0\}$ is an ascending chain and for all n:

$$f^n(x) \leq \mathbf{fix}_A f$$

Moreover, since we are working with finite lattices, the chain will eventually stabilise and the limit will be $\mathbf{fix}_A f$.

(ii) if x is a post-fixed point of f and x is greater than $\mathbf{fix}_A f$ then $\{f^n(x) \mid n \geq 0\}$ is a descending chain and for all n:

$$f^n(x) \geq \mathbf{fix}_A f$$

However, notice that the limit of such a descending chain may not be $\mathbf{fix}_A f$ but some other fixed point. (Thus the need for the generality of Corollary 3.2) []

We can now present the scheme for finding approximate fixed points (to any desired accuracy):

Step 1: Choose some small lattice in which the problem of finding fixed points is tractable and iterate from bottom to find the least fixed point of both the safe and live abstractions of the function.

Step 2: The previous step gives upper and lower bounds for the true fixed point. If these agree on the interesting arguments, or if a safe answer is sufficient, use the upper bound; otherwise

Step 3: Apply UpS to the safe approximation and UpL to the live approximation to move to a larger, intermediate lattice and iterate down from the resultant post-fixed point and up from the resultant pre-fixed point. Repeat Step 2.

One word of caution: if the post-(pre-)fixed point of the safe (live) image of the function in some intermediate lattice happens to be a fixed point then no further improvement of the upper(lower) bound is possible. On the other hand, when the pre-fixed point is a fixed point, it must be the least fixed point.

4. Widening and Narrowing Operations

We start by recalling some definitions and results from [Cou81].

Definition 4.1

For any complete lattice L, an operation $\nabla \in \mathbb{N} \to ((L \times L) \to L)$ is a *widening operation* iff it satisfies the following conditions:

(i) $\forall j > 0, \forall x,y \in L, x \vee y \leq x \nabla(j) y$

(ii) For all ascending chains $x_0 \leq x_1 \leq \ldots \leq x_n \leq \ldots$ in L, the chain $y_0 = x_0$, $y_1 = y_0 \nabla(1) x_1, \ldots y_n = y_{n-1} \nabla(n) x_n$ is eventually stable; i.e. there exists a $k \geq 0$ such for all $i \geq k$, $y_i = y_k$.

[]

A widening operator may be used to generate an "accelerated" fixed point iteration (which in general will overshoot the least fixed point) as shown by the following proposition.

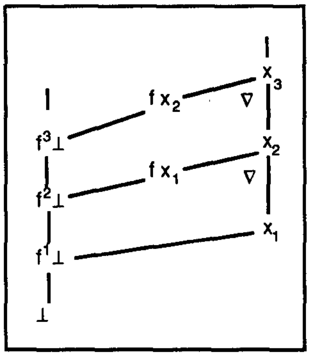
Proposition 4.2

Let f be a monotone operator on L and ∇ a widening operator. The limit u of the sequence:

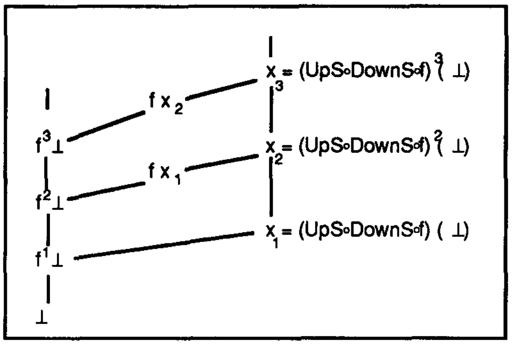
$$x_0 = \perp$$
$$x_{n+1} = x_n, \text{ if } f(x_n) \leq x_n$$
$$x_{n+1} = x_n \nabla(n+1) f(x_n), \text{ otherwise}$$

can be computed in a finite number of steps. Moreover **fix**(f) $\leq$ u and f(u) $\leq$ []

The iteration process described by the proposition and its relationship to the Ascending Kleene Chain is illustrated in the following figure:



Let $f : B \to B$ with $A \leq B \in L$ and consider the sequence $(\text{DownS } f)^n \perp$. For the purposes of comparison with widening, we embed this sequence into B using UpS. It is easily shown that the resulting sequence UpS((DownS f)^n $\perp$) is just (UpS $_o$ DownS $_o$ f)^n $\perp$, giving the modified diagram:



and consequently we have:

Lemma 4.3

For any lattices D, D'$\in$ *L*, such that D' $\le$ D:

$$\lambda j.\lambda(x,y).x \vee \mathrm{UpS}_{D'D}(\mathrm{DownS}_{D,D'}\, y))$$

is a widening operation.

Proof

Observe that $\mathrm{UpS}_{D'D}(\mathrm{DownS}_{D,D'}\, y)) \ge y$ by the definition of embedding-closure pairs and thus:

$$x \vee \mathrm{UpS}_{D'D}(\mathrm{DownS}_{D,D'}\, y)) \ge x \vee y$$

Any ascending chain must be eventually stable since the lattices are all finite.

[]

In our earlier discussion, we presented three steps for computing approximate fixed points. We have now shown the equivalence of step 1 of that process and the Cousot's notion of widening. However, it may be preferable to use the approach of the last section for efficiency reasons since the explicit use of the widening operator requires us to work in a larger lattice.

Still considering the safe maps, we now turn to the process of refining the approximation and start by defining the concept of narrowing.

Definition 4.4

For any complete lattice L, an operation $\Delta \in \mathbb{N} \to ((L \times L) \to L)$ is a *narrowing operation* iff it satisfies the following conditions:

(i) $\forall j > 0, (\forall x,y \in L: y \leq x),\ y \leq x\ \Delta(j)\ y \leq x$

(ii) For all descending chains $x_0 \geq x_1 \geq \ldots \geq x_n \geq \ldots$ in L, the chain $y_0 = x_0$, $y_1 = y_0\ \Delta(1)\ x_1, \ldots\ y_n = y_{n-1}\ \Delta(n)\ x_n$ is eventually stable; i.e. there exists a $k \geq 0$ such for all $i \geq k$, $y_i = y_k$.

[]

Proposition 4.5

Let f be a monotone operator on L and Δ a narrowing operator. Let $u \in L$ be such that **fix** $f \leq u$ and $f(u) \leq u$. The decreasing chain:

$$x_0 = u$$
$$x_{n+1} = x_n\ \Delta(n+1)\ f(x_n)$$

is eventually stable. Moreover $\forall k \geq 0$, **fix**$(f) \leq x_k$.

[]

Step 3 of the procedure outlined in the last section proposed the use of a decreasing iteration which we might reasonably expect to correspond to a narrowing. We now present a very general process, which corresponds to Step 3, in which each iterate may be from a different intermediate lattice. We consider a sequence of lattices $A_1, \ldots A_n$ such that $A_i \leq A_{i+1}$ and $A_n \equiv A$, then we have $D_i \equiv [A_i \rightarrow A_i]$. We construct the sequence:

$$z_0, z_1, \ldots$$

where

$$z_0 = \text{The fixed point found in Step 1}$$
$$z_{i+1} = (\text{DownS}_{D,Di+1}\ f)\ (\text{UpS}_{Ai,Ai+1}\ z_i)$$

The embedding of $\{z_n\}$ into A via the maps $\text{UpS}_{Ai,A}$ results in the decreasing sequence associated with the narrowing operation defined in the following Lemma.

Lemma 4.6

For any sequence of lattices $A_0, A_1, \ldots, A_n = A \in \mathit{L}$, such that $A_i \leq A_{i+1}$ (the A_i need not be distinct), $D_i \equiv [A_i \rightarrow A_i]$, $D \equiv D_n$, $f \in D$:

$$\Delta \equiv \lambda j.\lambda(x,y).x \wedge \text{UpS}_{Aj,A}(\text{DownS}_{A,Aj}\ y)$$

is a narrowing operation.

Proof

$$x\ \Delta(j)(u)\ y \quad = \quad x \wedge \text{UpS}_{Aj,A}(\text{DownS}_{A,Aj}\ y)$$

$$\leq \quad x \qquad \text{by definition of } \wedge$$

Since $\text{UpS}_{Aj,A}(\text{DownS}_{A,Aj}\ y) \geq y$ (by the defining property of embedding-closure pairs), we also have for $y \leq x$ that:

$$x\ \Delta(j)(u)\ y \quad \geq \quad y$$

Eventual stability of the sequence follows from finiteness of the lattices.

[]

To summarise: the upwards iteration in the smaller lattice using a safe approximation of the function corresponds to widening and the refinement of the upper bound by iterating downwards in intermediate lattices corresponds to narrowing.

The situation with live maps is somewhat less straightforward. The live approximations approach the true fixed point from below; this is true both of the initial approximation and the successive refinements. This runs counter to the development of [Cou81] and later work. In [Cou78] alternative definitions of widening and narrowing operators are introduced but these do not correspond very closely to our application. This merits further investigation.

As a closing remark notice that the exact correspondence proved in Lemma 4.3 gives the basis for an alternative proof of the post-fixed point property proved in Proposition 3.1 since any widening operation has this property [Cou78].

5. Conclusions

We have developed the work on approximate fixed points first reported in [Hun89] and shown how it connects with the widening/narrowing approach used in traditional abstract interpretation. We have presented a scheme which computes arbitrarily precise upper and lower approximations of the true least fixed point of a function.

An alternative approach is based on the observation that often only a small part of the function graph is actually required. If a suitable superset of the subgraph (which avoids the plateaux problems described in [CPJ85]) can be identified then an accurate fixed point in the superset can be used. Since the needed elements of the graph may be many orders of magnitude smaller than the cardinality of the graph, this accurate fixed point can be computed very efficiently. These ideas, which are related to Jones and Mycroft's minimal function graphs [Jon85] are currently being developed.

Acknowledgements

We are indebted to our colleagues on the Semantique project for their willingness to discuss this work and for their constructive criticism; specifically, the categorical approach used in Section 2 was suggested by John Hughes and has led to a considerbale simplification of our work. The first author was partially funded by ESPRIT BRA 3124 - Semantique and both authors were partially funded by ESPRIT BRA 3074 - SemaGraph.

References

[BHA86] G. L. Burn, C. L. Hankin and S. Abramsky, *Strictness Analysis for Higher-order functions*, Science of Computer Programming 7 (1986), pp 249-278, North-Holland.

[Cou78] P. Cousot, *Méthodes Itératives de Construction et d'Approximation de Points Fixes d'Opérateurs Monotones sur un Treilli, Analyse Sémantique des Programmes*, Thèse d'Etat, Université de Grenoble, 1978.

[Cou81] P. Cousot, *Semantic foundations of program analysis*, in Muchnick S. S. and Jones N. D. (eds) *Program Flow Analysis*, pp 303-342, Prentice-Hall, 1981.

[CPJ85] C. Clack and S. L. Peyton Jones, *Strictness Analysis - a practical approach*, in J. -P. Jouannaud (ed), *Functional Programming Languages and Computer Architecture, LNCS* 201, pp 35-49, Springer Verlag.

[GHK*80] G. K. Gierz, K. H. Hoffmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott, *A Compendium of Continuous Lattices*, Springer Verlag.

[Hun89] S. Hunt, *Frontiers and open sets in abstract interpretation*, in D. MacQueen (ed), *Functional Programming Languages and Computer Architecture,* pp 1-11, ACM Press.

[HH91] S. Hunt and C. L. Hankin, *Fixed Points and Frontiers: a new perspective*, Journal of Functional Programming **1**(1), pp 91-120, Cambridge University Press.

[Jon85] Jones N. D. and Mycroft A. *Dataflow Analysis of Applicative Programs using Minimal Function Graphs*, privately circulated manuscript, October 1985.

[Mey85] A. R. Meyer, *Complexity of Program Flow Analysis for Strictness: Application of a Fundamental Theorem of Denotational Semantics*, private communication.

[MH87] C. C. Martin and C. L. Hankin, *Finding Fixed Points in Finite Lattices*, in G. Kahn (ed), *Functional Programming Languages and Computer Architecture, LNCS* 274, pp 426-445, Springer Verlag.