1. INTRODUCTION

The problem of discovering invariant assertions of programs is explored in light of the fixpoint approximations method. Therefore symbolic execution permits optimal invariant assertions to be discovered provided that one can pass to the limit, that is consider infinite paths in the symbolic execution...
We denote by $L^n$ the set of all vectors $X=(X_1,\ldots,X_n)$ the components of which belong to $L$. $(L^n,\leq)$ is a complete lattice with the usual "componentwise" definitions: $(X\leq Y)\iff (\forall i\leq n, X_i\leq Y_i)$, $(\forall i\leq n, Y_i\leq Y_{i+1})$, etc.

A function of several variables $f:L^n\to L$ is said to be isotone (continuous) in the variables jointly if and only if it is isotone (continuous) in the variables separately.

Hereafter we will consider a system of continuous equations with $n$ variables of the form:

\[
\begin{align*}
X_1 &= f_1(X_1,\ldots,X_n) \\
&\vdots \\
X_n &= f_n(X_1,\ldots,X_n)
\end{align*}
\]

This system can be abbreviated by a fixed-point equation $X=f(X)$ where $X$ is the vector $(X_1,\ldots,X_n)$ and $f$ a continuous function of type $L^n\to L^n$.

An element $X$ of $L^n$ is a fixedpoint of $f:L^n\to L^n$ if and only if $f(X)=X$. The least fixedpoint $\text{lpf}(F)$ of $F$ is such that:

\[
\{f(\text{lpf}(F))=\text{lpf}(F)\} \quad \text{and} \quad \{\forall X\in L^n, (X)f=X\Rightarrow \text{lpf}(F)\subseteq X\}\}
\]

THEOREM (Tarski1955) Any monotone map $F$ of a complete lattice $L^n$ into itself has a least fixedpoint defined by $\text{lpf}(F) = \bigcap\{X\subseteq L^n : F(X)\subseteq X\}$.

In practice this theorem is not constructive since in general the set $\{X\subseteq L^n : F(X)\subseteq X\}$ of post-fixedpoints of $F$ is infinite and cannot be easily characterized. Yet, when $F$ is continuous Kleene[1952] and Tarski[1955] suggest that the least fixedpoint of $F$ can be obtained as the limit of a sequence of successive approximations $X^0, X^1, X^2, \ldots$, that is $\text{lpf}(F) = \lim\limits_{X^k} F(X)$ where $F^k$ denotes the $k$-fold composition of $F$ with itself. This is nothing else than Jacobi's method of successive approximations:

\[
\begin{align*}
X_1 &= f_1(X_1,\ldots,X_n) \\
&\vdots \\
X_k &= f_k(X_1,\ldots,X_n) \\
&\vdots
\end{align*}
\]

We now generalize this result by showing that any chaotic iteration method converges to the least fixedpoint of $F$. Otherwise stated this signifies that one can arbitrarily determine at each step which are the components of the system of equations which will evolve and in what order (as long as no component is forgotten indefinitely).

Let $J$ be a non-empty subset of $\{1,\ldots,n\}$. We denote by $J^c$ the map $L^n\to L^n$ defined by $J^c(X_1,\ldots,X_n) = (Y_1,\ldots,Y_n)$ where $\forall i\in J^c$ we have $Y_i=\text{lpf} X_i$ else $X_i$. As before.

DEFINITION 2.1 A chaotic iteration corresponding to the operator $F$ and starting with a given vector $X^0$ such that $X^0\leq F(X^0)\leq \text{lpf}(F)$ is a sequence $X^k, k=0,1,\ldots$ of vectors of $L^n$ defined recursively by $X^k=J^c(X^{k-1})$ where $J^k, k=0,1,\ldots$ is a sequence of subsets of $\{1,\ldots,n\}$ such that $\forall m>0 : \{\forall i\leq n, \forall k\leq 0 : i\notin J^k\}$.

THEOREM 2.2 The limit $X^\infty$ of a chaotic iteration $X^0, X^1, X^2, \ldots$ is equal to $\text{lpf}(F)$.

\[
\text{Lemma 2.2.1:} \ \forall k\geq 0, X^k \leq X^{k+1} \leq F(X^k) \subseteq \text{lpf}(F).
\]

\[
\text{Proof:} \ \text{Let us first remark that whenever } X^k \leq F(X^k) \leq \text{lpf}(F) \text{ we have } \forall i\leq n, X^k \leq F_i(X^k) \subseteq \text{lpf}(F).\ \text{Indeed } \forall i\leq n, X^k \leq F_i(X^k) \text{ therefore if } i\notin J^k \text{ then } X^k \subseteq F_i(X^k) \subseteq F_i(X^k) \text{ otherwise } X^k \subseteq F_i(X^k) \subseteq F_i(X^k).
\]

\[
\text{Since by hypothesis } X^k \leq F(X^k) \subseteq \text{lpf}(F) \text{ this implies } X^k \subseteq F(X^k) \leq X^{k+1} \leq F(X^{k+1}) \subseteq \text{lpf}(F). \text{ For the induction step let us assume that } X^{k-1} \leq X^k \leq F(X^{k-1}) \leq \text{lpf}(F) \text{ for some } k>0. \text{ If } i\notin J^k \text{ then } X^k \subseteq F_i(X^k) \subseteq F_i(X^k) \text{ by isotonicity of } F_i \text{ and } F_i(X^{k-1}) \subseteq F_i(X^{k-1}) \subseteq F_i(X^{k-1}) \subseteq \text{lpf}(F). \text{ In both cases } \forall i\leq n, X^k \subseteq F_i(X^k) \subseteq \text{lpf}(F) \text{ therefore } X^k \leq F(X^k) \subseteq \text{lpf}(F) \text{ proving that } X^k \leq X^{k+1} \subseteq \text{lpf}(F). \text{ End of Proof.}
\]

\[
\text{Lemma 2.2.2:} \ \{\exists q\leq m : \forall k\geq 0, F(X^k) \leq X^{k+q}\}.
\]

\[
\text{Proof:} \ \text{The proof is by reductio ad absurdum. Let us suppose that } \{\forall q\leq m, \exists k\geq 0 : \text{not} (F(X^k) \leq X^{k+q})\} \iff \{\forall q\leq m, \exists k\geq 0 : (X^{k+q} \leq F(X^k)) \text{ or } (X^{k+q} \leq F(X^k) \text{ not comparable with } X^{k+q})\}.
\]

\[
\text{case 1 : Suppose that } \forall q\leq m, \exists k\geq 0 \text{ such that } F(X^k) \text{ not comparable with } X^{k+q}. \text{ This must be true for } q=0 \text{ which contradicts lemma 2.2.1.}
\]

\[
\text{case 2 : Suppose now that } \forall q\leq m, \exists k\geq 0 \text{ such that } X^{k+q} \leq F(X^k), \text{ that is to say by definition of the strict inequality } \leq \text{ we have } X^{k+q} \leq F(X^k) \text{ and } X^{k+q} \leq F(X^k). \text{ This implies that for some component } i\leq n \text{ we have } X^{k+q}_i \leq F_i(X^k), \text{ while for the other components the inequality is not necessarily strict. By definition of}
\]
chaotic iterations \( \forall m \geq 0 : (W_i, W_k, \forall i \in \mathbb{N}, m : i \in k^{k+1}) \), therefore \( X_{i+1}^{k+1} = F_i(X_i^{k+1}) \). But lemma 2.2.1 implies by transitivity that \( X_{i+1}^{k+1} = X_i^{k+1} \), thus by isotony \( F_i(X_{i+1}^{k+1}) \in F_i(X_i^{k+1}) \) which implies \( F_i(X_{i+1}^{k+1}) \in X_i^{k+1} \).

Choosing \( q^{k+1} \) we have \( \exists k \) such that \( F_i(X_{i+1}^{k+1}) \in X_i^{k+1} \) and also by hypothesis such that \( X_{i+1}^{k+1} \in F_i(X_i^{k+1}) \), which is impossible. This contradiction proves the truth of lemma 2.2.2. \textit{End of Proof.}

\textbf{Proof of theorem 2.3:} The proof that \( (k : (k \geq L^n)) \)

\( (k \geq m \Rightarrow \text{Inf}(F) \subseteq X_i^{k+1}) \) is by reductio ad absurdum. Indeed, suppose that \( (W_k : (k \geq L^n)) \)

or \( (k \geq m \text{ and } \text{Inf}(F) \subseteq X_i^{k+1}) \). Let \( \alpha \) be the least ordinal of cardinality strictly greater than \( L^n \). \( W_k \alpha \) we must have \( (k \geq m \text{ and } \text{Inf}(F) \subseteq X_i^{k+1}) \), that is \( \text{Inf}(F) \subseteq X_i^{k+1} \) when choosing \( k \) to be \( \alpha \). Let us define \( \psi \in \alpha + L^n \) by \( \psi(k) = X_i^{k+1} \)

\( W_k \), \( k \in \alpha \) such that \( k \neq k \) (with eventually \( k \neq k \)) exclusive or \( (k \neq k) \) we have \( (L^n \subseteq X_i^{k+1}) \) or \( (L^n \neq X_i^{k+1}) \) hence lemma 2.3.1 imply that \( X_i^{k+1} \neq X_i^{k+1} \)

proving that \( \psi \) is a one to one correspondence of \( \alpha \)

into \( L^n \). Therefore \( \alpha \) is of cardinality less or equal
to that of \( L^n \) which is the desired contradiction.

\textit{End of Proof.}

3. DEDUCTIVE SEMANTICS OF PROGRAMS

The deductive semantics of a program defines
Program entry point $j$:
This general term $P^i$ of the chaotic iterations is proved to be correct by mathematical induction:

**Basis:** $P^1$ and $P^2$ are of the form specified by $P^i$ for $i=1$ and $i=2$.

**Induction step:** Assuming $P^i$ to be

5. **SYSTEMS OF IMPLICATIONS, APPROXIMATE INVARIANTS AND PROOFS OF PARTIAL CORRECTNESS**

Most program verification methods use inequalities of the form $P \Rightarrow P(P)$ whereas we used equalities.
satisfying \( P \models F(P) \). This is not surprising since this condition is based on Tarski's theorem which does not provide an algorithmic construction of the

\[
\begin{align*}
\{ & \{ \exists \Delta \{ Q_j \land \cdots \land (X_k \in E_{i,j}, \ldots, E_{k,j}, \ldots, E_{m,j}) \land \cdots \land (X_m \in E_{i,j}) \} \\
& \implies F_{i,j} \} : i \in I \} \}
\end{align*}
\]
Step 2:

\[ P_1^2 = \{ (\text{true}, \bar{R}, \bar{y}) \} \]

\[ P_1^3 = \{ (\text{true}, \bar{R}, \bar{y}), (\bar{R} \bar{y}), (\bar{R}, \bar{R} \bar{y}, \bar{y}) \} \]

\[ P_1^4 = \{ (\bar{R} \bar{R} \bar{y}), \bar{R}, \bar{y}, \bar{y} \} \]

\[ P_1^5 = \{ (\bar{R} \bar{R} \bar{y}), \bar{R}, \bar{y}, \bar{y} \} \]

So that at iteration 2 we have built the following symbolic execution tree (Hantler & King [1976]):

We have represented the symbolic context \( P_1 \) associated with program point \( i \) by the set of paths associated with each of the nodes labeled \( i \) in the above execution tree. Equivalently we could have represented the symbolic context associated with program point \( i \) by the maximal subtree (of the above symbolic tree) the leaves of which are labeled \( i \).

Then the union \( U \) of symbolic contexts performed at junction of program paths would be the merging of symbolic execution trees.

It is clear that the computation of the next terms of the sequence of chaotic iterations would cause the symbolic execution tree to grow. We can make the tree to grow in whatever direction we want, the result will be the same (2.2). Without particular hypothesis on \( R \) and \( y \) this process would converge to the optimal invariants in infinitely many steps (2.3). Therefore we must be able either to reason about the limit without knowing it (6.4) or to directly pass to the limit (7).

6.5. VERIFICATION OF PROPERTIES OF OPTIMAL SYMBOLIC CONTEXTS

Coming back to the notations of §2.3 in order to prove a property \( P(X^m) \) of the solution to the system of equations \( X = f(X) \) we can prove by induction that all terms \( X^m \) of a chaotic iteration sequence have this property:

\[ (P(X^1) \land P(X^m)) \land (\forall k \in \mathbb{N}, P(X^m) \Rightarrow P(X^{(k+1)m})) \Rightarrow P(\lim X^m) \]

(Yet \( P \) must be an admissible predicate chosen in order to remain true when passing to the limit. Rigorously we should apply the second principle of transfinite induction).

Example: Let us prove the trivial fact that \( \{ \text{Vae}(1, (R=x)/y_i), \text{Rax}(y) \} \) at point 3 of program 6.3.

Base: For the single path of \( P_1^1 \) we have \( \{ \text{Vae}(1,1, \text{Rax}(y)) \} \).

Induction step: We assume that at step \( k \) the symbolic context \( P_k \) is equal to \( \{ p_i, x_i, y_i \} : \text{iC} \) with the induction hypothesis \( \{ \text{Vae}(1, (R=x_i)/y_i), \text{Rax}(y) \} \). The equations 6.3 allow the computation of \( P_{k+1} \):

\[ P_{k+1}^1 = \{ (p_1, x_1, y_1) : \text{iC} \} \]

\[ P_{k+1}^2 = \{ (\text{true}, \bar{R}, \bar{y}), (p_1, x_1, y_1) : \text{iC} \} \]

\[ P_{k+1}^3 = \{ (\bar{R} \bar{R} \bar{y}), \bar{R}, \bar{y}, (p_1, x_1, y_1) : \text{iC} \} \]

We must show that the hypothesis holds for all paths of \( P_{k+1}^3 \). This is trivial for the path \( (\bar{R} \bar{R} \bar{y}), \bar{R}, \bar{y} \). Otherwise we must show that \( \{ \text{Vae}(1, (R=x_i)/y_i+1), \text{Rax}(y) \} \). According to the induction hypothesis, this condition is true \( \text{Vae}(1, (R=x_i)/y_i) \). Finally for \((\bar{R} \bar{R} \bar{y}), \bar{R}, \bar{y} \), the path condition \( x_1, y_1 \) implies \( \text{Rax}(y) \).

This approach for reasoning about the limit of chaotic iterations is implicitly used in the technique of "cut-trees" of Hantler & King [1976]. Indeed the induction step can be understood as consisting in reasoning on the cut tree for \( (\bar{R} \bar{R} \bar{y}) \):

\[ \{ p_1, x_1, y_1 : \text{iC} \} \]

\[ \{ p_1, x_1, y_1 : \text{iC} \} \]

\[ \{ (\bar{R} \bar{R} \bar{y}), \bar{R}, \bar{y}, \} \]

\[ \{ (\bar{R} \bar{R} \bar{y}), x_1, y_1 : \text{iC} \} \]

\[ \{ \text{true}, \bar{R}, \bar{y} \} \]

\[ \{ \text{true}, \bar{R}, \bar{y} \} \]

End of Example.
7. SYNTHESIS OF OPTIMAL INVARIANT ASSERTIONS: THE USE OF DIFFERENCE EQUATIONS

7.1. DISCOVERY OF OPTIMAL SYMBOLIC CONTEXTS

The equations of example 6.3 can be written as:

\[ P_1 = P_0 \cup (P_1 \land x \neq y) (\neg x \lor x \neq y) \]

they are of the form:

\[ P_1 = f_1(P_0) \cup f_2(P_1) \]

The resolution by successive approximations was:

\[ P^0_1 = \emptyset \]

\[ P^1_1 = f_1(P_0) \cup f_2(\emptyset) = f_1(P_0) \]

\[ P^2_1 = f_1(P_0) \cup f_2 f_1(P_0) \]

\[ P^3_1 = f_1(P_0) \cup f_2 f_3 f_1(P_0) = f_1(P_0) \cup f_2 f_3 f_1(P_0) \]

since \( f_2 \) is distributive over \( \cup \). (This comes from the fact that \( f_2 \) is the composition of elementary functions as defined in 6.2. Setting apart the difference in notations they are similar to the predicate transformers of the deductive semantics (3.2). Since the set of predicates form a complete boolean lattice the infinite distributive laws hold for OR and AND.

The general term of the approximation sequence is

\[ P^k_1 = \bigcup_{i=0}^{k-1} f_1 f_2 f_3 f_1(P_0) \]

\[ f_2 \circ f_1(P_0) = f_2 \left( \left\{ \langle p_{ij}, x_{ij}, y_{ij} \rangle : j \in D_i \right\} \right) \]

\[ = \left\{ p_{ij} \land x_{ij} \neq y_{ij} \cdot x_{ij} \neq y_{ij} : j \in D_i \right\} \]

First of all, we can determine the domain of \( j \) which indexes the possible paths. Since \( D_0 = \{1\} \) and \( D_{i+1} = D_i \) we have \( D_1 = \{1\} \) (because there is a single path within the loop). Therefore we can simply ignore the path index and solve the difference equations (in the order of dependence):

\[
\begin{align*}
Y_0 &= \overline{Y} \\
Y_{i+1} &= Y_i \\
x_0 &= \overline{x} \\
x_{i+1} &= x_i \cdot \overline{y_i} \\
p_0 &= \text{true}, \quad p_{i+1} = p_i \quad \text{and} \quad (x_i \neq y_i)
\end{align*}
\]

These recurrence relations can be solved directly (e.g. Cohen & Katcoff[1976]) yielding \( Y_1 = x_0 \cdot \overline{x_0} \cdot i_{y_0} \) and since \( p_0 \text{=true} \) and \( p_{i+1} = p_i \) and \( x_0 \cdot y_0 \cdot i_{y_0} \) we get

\[ \bigcup_{i=1}^{\infty} (x_0 \neq y_0) = \{ w : (1,1), (x_0 \neq y_0) \} \]

Finally the optimal loop invariant of program 6.3 is:

\[ P^\infty_1 = \bigcup_{i=0}^{\infty} \left\{ \langle x_0 \neq y_0, x_0 \neq i_{y_0}, y_0 \rangle \right\} \]

The "difference equations method" was introduced by Elspas, Green, Levitt & Waldinger[1972], Elspas[1974]. It is further used in Greif & Waldinger[1975], Katz & Manna[1976] (algorithmic approach), Cheatham & Townsley[1976]. It has also been used in determining symbolic expression of program complexity (Wegbreit[1975]). However, the technique was un-
We can express $P_1$ by the following equivalent forms:

$$P_1 = \bigcup_{i \in \mathbb{N}} (f_4 \circ f_3 \circ (f_6 \circ f_5 \circ f_3)^i \circ f_1)(P_0)$$

(i and j correspond to individual counters for the loops)

$$P_1 = \bigcup_{k \in \mathbb{N}} (f_1(P_0) \cup f_4 \circ f_3 \circ (f_6 \circ f_5 \circ f_3)^k \circ f_1)(P_0)$$

(k is a common counter for the two loops).

Although these two forms of $P_1$ are equivalent one of them will be more suitable for generating the difference equations and the choice depends on the semantics of the considered program.

follows:

$$P_1 = P_0$$

$$\cup$$

$$(P_1 \text{ and } P_1(y)\neq 0 \text{ and even}(P_1(y))) \{y + P_1(y) \div 2, x + (P_1(x))^2, z + P_1(z)\}$$

$$\cup$$

$$(P_1 \text{ and } P_1(y)\neq 0 \text{ and odd}(P_1(y))) \{y + P_1(y) \div 2, x + (P_1(x))^2, z + P_1(z)\} \cdot P_1(x)$$

Since the two alternatives differ only with respect to $z$ we can factorize as follows:

$$P_1 = P_0 \cup (P_1 \text{ and } y \neq 0) \{x + x^2, y + y \div 2, z \cdot z \cdot (\text{even}(y) + 1 \text{ or } \text{odd}(y) \rightarrow x)\}$$

$$= P_0 \cup f(P_1)$$

This formulation uses the conditional expressions of Sintzloff[1975]:

- $(true \lor v \lor Q \lor v') = v \lor (Q \lor v')$
- $false \lor v \lor Q \lor v' = (Q \lor v')$
- $Q \lor Q' = Q \lor Q'$
- $Q \lor Q' = Q' \lor Q$
- $\square(Q \lor v) = (\square Q \lor v)$
- $\square(Q \lor v) = (\square Q \lor v)$
- $\square(Q \lor v) = (\square Q \lor v)$
- $f(v_1, \ldots, v_1, \ldots, v_n) = f(v_1, \ldots, v_1, \ldots, v_n)$
- $f(v_1, \ldots, (Q \lor w), \ldots, v_n) = (Q \lor f(v_1, \ldots, w, \ldots, v_n))$
- etc.
Finally the optimal output invariant of the program is:

\[ P_7 = P_1 \text{ and } (y = 0) \]

\[ = \bigcup_{i=0}^{\infty} <q_i \text{ and } (y_i = 0), x_i, y_i, z_i> \]

\[ = \bigcup_{b=0, x=a, y=b, z=1} \]

\[ <(\exists i > 0 : (b+2^i = 0) \text{ and } (b+2^{i+1} = 0)), \]

\[ x = a^{2^i}, \]

\[ \mathcal{P}^k(x^k)_1 = x^k \]

\[ \text{else} \]

\[ x^k_1 \cup F_1(x^k) \subseteq \mathcal{P}^k(x^k)_1 \]

\[ fi \]

and \( J^k, k = 0, 1, \ldots \) is a sequence of subsets of \( \{1, \ldots, n\} \) such that \( \exists m_0 : \{\forall i \in [1, n], \forall k \geq 0, \exists k \in [0, m_1 : i \in J^{k+1}\}}\).
approximation sequence at §3.4:

\[ p_2^g = \text{false} \]

\[ p_1^2 = (\exists \alpha \exists \beta) \land (x = \alpha \land y = \beta) \]

The idea of solving these equations by means of chaotic successive approximations is so natural that it was first understood as a program execution on symbolic values. We think that even a semi-automatic resolu-


Special Issue

Proceedings of the Symposium on
Artificial Intelligence and Programming Languages

Association for Computing Machinery
The papers in this volume were presented at the ACM Symposium on Artificial Intelligence and Programming Languages, sponsored jointly by SIGART and SIGPLAN, and held at the University of Rochester, August 15-17, 1977.

The Program Committee wishes to thank all those who submitted papers for consideration.