Least Fixpoint Approximations for Reachability Analysis*

In-Ho Moon1  James Kukula2  Tom Shiple2  Fabio Somenzi1

1Dept. of ECE, University of Colorado at Boulder  2Advanced Technology Group, Synopsys Inc.

Abstract

The knowledge of the reachable states of a sequential circuit can dramatically speed up optimization and model checking. However, since exact reachability analysis may be intractable, approximate techniques are often preferable. Cho et al. presented the Machine-By-Machine (MBM) and Frame-By-Frame (FBF) methods to perform approximate FSM traversal. FBF produces tighter upper bounds than MBM; however, it usually takes much more time and it may have convergence problems. In this paper, we show that there exists a class of methods—Least Fixpoint Approximations—that compute the same results as FBF (Reached FBF, one of the FBF methods). We show that one member of this class, which we call Least fixpoint MBM (LMBM), is as efficient as MBM, but provably more accurate. Therefore, the trade-off that existed between MBM and RFBF has been eliminated. LMBM can compute RFBF-quality approximations for all the large ISCAS-89 benchmark circuits in a total of less than 9000 seconds.

1 Introduction

The knowledge of the reachable states has been used in the optimization of sequential circuits [9, 6], sequential equivalence and testing [6], and CTL model checking [10, 11]. However, since exact reachability analysis may be intractable, approximate techniques are often preferable [11, 4]. The challenge in computing approximations of the reachable states is to obtain the best accuracy within given time and memory limits.

Cho et al. presented an automatic state space decomposition method [5] and approximate FSM traversal method based on the state space decomposition [4]. Two classes of approximate FSM traversal methods were introduced in these works: Machine-By-Machine (MBM) and Frame-By-Frame (FBF). Two variants of the latter were presented: Reached FBF (RFBF) and To FBF (TFBF).

MBM initially sets the reachable states of all submachines to the tautology and computes the reachable states of each submachine in turn. At each iteration, the transition relation of the submachine being traversed is constrained by the reachable states of the other submachines. The process is repeated until the reachable states sets stop shrinking. MBM is therefore a greatest fixpoint computation with least fixpoint computations (reachability analyses of the submachines) performed at each iteration.

RFBF initially sets the reachable states of each submachine to its initial states. It then interleave the reachability analyses of the submachines by computing one step (one image computation) for each submachine in turn. RFBF also constrains the transition relation of each submachine. When computing the i-th step of reachability analysis for a submachine, the accumulated reachable states of the other submachines from the previous iteration are used. The computation continues until the reachable states sets stop growing. Therefore, unlike MBM, RFBF is a least fixpoint computation. RFBF produces tighter upper bounds than MBM; however it usually takes much more time and memory than MBM.

RFBF owes its name to its use of the reached sets from the previous image computation to constrain the transition relations of the submachines. The other FBF method is called TFBF because it uses the to sets (the images themselves) from the previous image computations to constrain the transition relation. TFBF produces tighter upper bounds than RFBF in many cases; however, it converges more slowly (sometimes much more slowly) and usually requires more memory.

Therefore another method presented in [4], called TMBM, combines TFBF and MBM to avoid the convergence problem. TMBM starts with TFBF for a certain number of iterations, then switches to MBM with the last to sets from TFBF as initial states for MBM. TMBM produces tighter upper bounds than MBM in many cases; however, it usually takes much more time than MBM.

In this paper, we show that there exists a class of methods—Least Fixpoint Approximations—that includes RFBF and such that all methods in the class compute the same results. We show that one member of this class is as efficient as MBM, but provably more accurate. Therefore, the trade-off that existed between MBM and RFBF has been eliminated. In practice, our contribution makes it possible to compute RFBF-quality approximations for all the large ISCAS-89 benchmark circuits in a total of less than 9000 seconds. Since MBM is a greatest fixpoint computation and the new method is a least fixpoint Machine-By-Machine computation, we call this new method LMBM (Least fixpoint MBM).

An important issue in getting tight upper bounds on the reachable states is the quality of the state space decomposition. The partitioning method in [5] produces disjoint submachines. Recently, a new partitioning method using overlapping projections was presented in [8]. This method allows a state variable to belong to multiple submachines, and produces tighter upper bounds than the disjoint decomposition method. Its effects are orthogonal to those of LMBM in that the overlapping projections method improves the quality of decomposition, whereas LMBM improves the approximate FSM traversal for a given decomposition.

2 Preliminaries

2.1 Sequential Machines and their Decompositions

We model sequential circuits as finite state machines. We are concerned with circuits in which inputs, outputs, and states are encoded by strings of binary values. We ignore the outputs because they are irrelevant to reachability analysis. Finally, w.l.o.g., we consider deterministic circuits. (We can use extra inputs to model nondeterminism.) These assumptions lead us to the following definition.

Definition 1 Let \( x = \{x_1, \ldots, x_n\} \), \( y = \{y_1, \ldots, y_n\} \), and \( w = \{w_1, \ldots, w_p\} \) be sets of variables ranging over \( B = \{0, 1\} \). A (finite state) machine is a pair of boolean functions \( (T(x; w; y), I(x)) \), where \( T : B^{2n+p} \rightarrow B \) is 1 if and only if there is a transition from the state encoded by \( x \) to the state encoded by \( y \) under the input encoded by \( w \). 1: \( B^0 \rightarrow B \) is 1 if the state encoded by \( x \) is an initial state. The sets \( x \), \( y \), and \( w \) are called the present state, next state, and input variables, respectively.

---

*This work is supported in part by SRC contract 98-DJ-620.

0-7803-5832-X /99/$10.00 ©1999 IEEE.
The assumption of determinism implies that $T$ can be written as the conjunction of bit relations:

$$T(x, w, y) = \land_{1 \leq i \leq n} T_i^p(x, w, y).$$

The reachability problem consists of finding the set of states $R$ reachable from states in $I$ via a possibly empty sequence of transitions in $T$. Symbolic, BDD-based, algorithms for reachability [2, 7] solve the reachability problem by repeatedly computing the image of a set of states:

$$\text{Img}(T(x, w, y), S(x)) = (\exists x.w.(T(x, w, y) \cdot S(x)))_{x \in x}.$$

**Definition 2** A state space decomposition, $D$, of a finite state machine $(T(x, w, y), I(x))$ is a collection of sets $\{x' \subseteq x\}$ such that $\bigcup_j x' = x$. A finite state machine decomposed according to $D$ is a collection of machines $\{T_j, I_j\}$ such that

$$T_j(x, w, y) = \land_{n \in \mathbb{N}} T_i^p(x, w, y_j)$$

$$I_j(x) = \exists(x \backslash \lambda x'. I(x).$$

That is, $T_j$ is the conjunction of the bit relations for the variables in $x'$, while $I_j$ is the projection of the initial states onto the subspace of submachine $j$.

### 2.2 Closed Sequences of Monotonic Functions

**Definition 3** Let $H$ be a finite lattice, and $f$ be a function $H \rightarrow H$. We say that $f$ is monotonic if for $(h_1, h_2) \in H^2$, $h_1 \leq h_2$ implies $f(h_1) \leq f(h_2)$. A fixpoint of $f$ is an $h \in H$ such that $f(h) = h$.

Tarski [13] showed that a monotonic $f$ has a least fixpoint that is the greatest lower bound of all the fixpoints of $f$, and that can be computed by repeated application of $f$ starting from 0 (the least element of the lattice). The least fixpoint of $\land Z. f$ is denoted by $\rho Z. f$. By duality, the greatest fixpoint $\lor Z. f$ can be computed by repeated application of $f$ starting from 1.

The po werset of the set of states of a sequential machine is a lattice, and the functional $\land Z. T \lor \text{Img}(T, Z)$ that operates on that po werset is monotonic. Therefore, Tarski’s theorem guarantees that reachability analysis can be accomplished by repeated image computations. In approximate reachability we deal with the application of more than one functional. Hence, we need to generalize the notion of “repeated application of $f$.”

**Definition 4** Let $H$ be a finite lattice and let $F = \{f_1, \ldots, f_k\}$ be a finite set of monotonic functions $H \rightarrow H$. For $s$ a finite sequence over $F$, let $f_s$ be the function $H \rightarrow H$ obtained by composing all the functions in $s$ in the order specified by the sequence. We say that $s$ is closed if for $1 \leq i \leq k$ we have $f_i(f_s(0)) = f_i(0)$.

**Lemma 1** Let $s_1$ be a finite sequence over $F$ and let $s_2$ be a suffix of $s_1$. Then $f_{s_2}(0) \leq f_{s_1}(0)$.

**Theorem 1** Let $s_1, s_2$ be two closed sequences over $F$. Then $f_{s_1}(0) = f_{s_2}(0)$.

**Proof.** Let $(s_1, s_2)$ designate the concatenation of sequences $s_1$ and $s_2$. Obviously $s_2$ is a suffix of $(s_1, s_2)$. Hence, by Lemma 1 and by closure of $s_1$:

$$f_{s_2}(0) \leq f_{(s_1, s_2)}(0) = f_{s_1}(0).$$

By symmetric argument:

$$f_{s_1}(0) \leq f_{(s_2, s_1)}(0) = f_{s_2}(0).$$

Therefore, $f_{s_1}(0) = f_{s_2}(0)$. □

### 3 Least Fixpoint Approximations

Let $\mathbb{N}$ be the set of the non-negative integers. Suppose we are given a decomposed machine $\{(I_0, h_0), \ldots, \{I_{m-1}, h_{m-1}\}\}$. Let us consider the family of methods defined by the following recursion ($i \geq 0$):

$$L_j^{-1} = I_j$$

$$L_j^i = \left\{ \begin{array}{ll} L_j^{-1} \lor \text{Img}(T_j, \land_k L_k^{-1}) & \text{if } j = \sigma(i) \\ L_j^{-1} & \text{otherwise}\end{array} \right.$$  

where $\sigma : \mathbb{N} \rightarrow \{0, \ldots, m-1\}$ is the selection function, returning the index of the submachine selected at the $i$-th iteration. We require that $\sigma$ be a fair selection function: Each submachine whose input submachines change in response to an evaluation is eventually selected. Fairness of the selection functions and the finiteness of the state space guarantee that convergence is eventually reached. That is, there exists a finite $c \in \mathbb{N}$ such that, for $i \geq c$, $L_j^i = L_j^{i+c}$, for $j = 0, \ldots, m-1$. We denote $L_j^c$ at convergence by $L_j^\omega$.

We call the family of methods thus defined Least Fixpoint Approximations. One can see that RFBF is the procedure obtained by choosing

$$\sigma(i) = i \mod m.$$

Of particular interest to us is a method that we call Least fixpoint MBM (LMBM). LMBM is obtained with the following recursive definition ($i \geq 0$):

$$\sigma(0) = 0$$

$$\sigma(i+1) = \left\{ \begin{array}{ll} \sigma(i) & \text{if } L_j^{\sigma(i)} \neq L_j^{\sigma(i)+1} \\ \sigma(i)+1 \mod m & \text{otherwise}\end{array} \right.$$  

In words, RFBF uses a round-robin selection policy, whereas LMBM takes each submachine to convergence before switching to another submachine.

**Theorem 2** For the same decomposition, any least fixpoint approximation—including RFBF and LMBM—produces at convergence the same set $\{L_j^\omega\}$.

**Proof.** It is sufficient to observe that RFBF, LMBM, and all other least fixpoint approximations apply the same functionals in different orders. Hence, by Theorem 1, at convergence all least fixpoint approximations compute the same result. □

### 4 Comparison to BFS and MBM

To compare Least Fixpoint Approximations to BFS we put RFBF in this form:

$$R_j^{-1} = I_j$$

$$R_j^i = R_j^{i-1} \lor \text{Img}(T_j, \land_k R_k^{i-1}).$$

BFS, on the other hand, can be described as follows:

$$B^{-1} = I$$

$$B^i = B^{i-1} \lor \text{Img}(T, B^{i-1}).$$

**Theorem 3** Least Fixpoint Approximation computes an upper bound of $R$, the set of reachable states.
Proof. Thanks to Theorem 2, it is sufficient to prove that RFBF converges to an upper bound of $R$, which in turn is computed by BFS. Though this is not a new result [4, Theorem 4.5], we present a proof here for completeness. We prove by induction on $i$ that $R_j^i \geq B_i$. This clearly holds for $i = -1$. We then have for $i \geq 0$:

\[
R_j^i = R_j^{i-1} \vee \text{Img}(T_j \wedge \bigwedge_k R_k^{i-1}) \\
\geq R_j^{i-1} \vee \text{Img}(T \wedge \bigwedge_k R_k^{i-1}) \\
\geq B_i \vee \text{Img}(T, B_i^{i-1}) = B_i.
\]

Therefore, $\bigwedge_j R_j^{\infty} \geq R$, where $R_j^{\infty}$ is $R_j^i$ at convergence. \qed

To compare Least Fixpoint Approximations to MBM we put LMBM in this form:

\[
L_j^{-1} = 1_j \\
L_j^i = \begin{cases} 
\mu Z L_j^{i-1} \vee \text{Img}(T_j \wedge \bigwedge_k L_k^{i-1}, Z) & \text{if } j = \sigma(i) \\
L_j^{i-1} & \text{otherwise,}
\end{cases}
\]

where $\sigma$ is any fair selection function. MBM, on the other hand, is described by the following recursion, in which $\rho$ is any fair selection function.

\[
G_j^{-1} = 1 \\
G_j^i = \begin{cases} 
\mu Z I_j \vee \text{Img}(T_j \wedge \bigwedge_k G_k^{i-1}, Z) & \text{if } j = \rho(i) \\
G_j^{i-1} & \text{otherwise.}
\end{cases}
\]

Theorem 4 For the same decomposition, the result of Least Fixpoint Approximation is a lower bound on the result of MBM.

Proof. We prove by induction on $i$ that $L_j^i \leq G_j^i$. The base follows from $L_j^{-1} \leq 1 = G_j^{-1}$. For $i \geq 0$, we observe that under the inductive hypothesis:

\[
\alpha = \mu Z L_j^{i-1} \vee \text{Img}(T_j \wedge \bigwedge_k L_k^{i-1}, Z) \\
\leq \mu Z I_j \vee \text{Img}(T_j \wedge \bigwedge_k G_k^{i-1}, Z) = \beta.
\]

We distinguish four cases.

1. $j \neq \sigma(i), j \neq \rho(i)$: $L_j^i = L_j^{i-1} \leq G_j^{i-1} = G_j^i$.

2. $j = \sigma(i), j = \rho(i)$: $L_j^i = \alpha \leq \beta = G_j^i$.

3. $j = \sigma(i), j \neq \rho(i)$: $L_j^i = \alpha \leq \beta \leq G_j^{i-1} = G_j^i$.

4. $j \neq \sigma(i), j = \rho(i)$: $L_j^i = L_j^{i-1} \leq \alpha \leq \beta = G_j^i$.

Following is an example in which LMBM produces a tighter bound than MBM. Suppose we have a machine with a total of two registers $A$ and $B$. At each cycle, the values in the registers are swapped. We choose a partition such that Register $A$ is in Submachine $T_1$ and Register $B$ is in Submachine $T_2$. If the initial values of $A$ and $B$ are equal, the machine just stays in one state forever. LMBM will figure this out, whereas MBM will mark every state as reachable. In the first iteration—before computing the reachable states of $T_1$—LMBM constrains the transition relation of $T_1$ with respect to the reachable states of $T_2$, which are the initial state. Hence, the reachable states of $T_1$ are still the initial state. The same happens in computing the reachable states of $T_2$. Since the reachable states of the two submachines do not change, LMBM stops and the number of the reachable states is only one. By contrast, MBM constrains the transition relation of each submachine with respect to the reachable states of the other submachine, which are initially all states. Hence, the reachable states of each submachine remain all states.

5 Efficient Implementation of LMBM

Several optimizations can be applied to the basic LMBM procedure described in Section 3, without affecting the computed upper bounds.

- Use $\bigwedge_k L_k^i$ to modify the transition relation of a submachine once for each traversal.
- Use constrain with respect to fanins only, instead of conjunction with all submachines.
- Use event-driven selection function. (Stronger condition than fairness.)
- Use frontier-driven selection function.

Some of these optimizations, for instance the use of the frontier states, only work if the same submachine is repeatedly executed. Hence, they are not applicable to RFBF. The pseudocode of Figure 1 shows the result of applying the optimizations.

```
Procedure LMBM({(T_0, I_0), \ldots, (T_m, I_m)})
  
  Active = \{0, \ldots, m-1\}
  
  while (Active \neq 0) {
    j = Select(Active)
    \( T_j = T_j \wedge \bigwedge_k L_k \)
    P = L_j
    L_j = \mu Z L_j \vee \text{Img}(T_j, L_j)
    if (P \neq L_j)
      for all (i \in FanOut(j))
        L_i = L_i \wedge \text{Img}(T_i, L_j)
    }
  return \{L_i\}
```

Figure 1: LMBM procedure.

6 Experimental Results

We have implemented the LMBM procedure described in Figure 1 in VIS 1.3 [1] and compared it to the existing approximate traversal algorithms in VIS 1.3. We have run all experiments on a 400MHz Pentium II machine with 1GB of RAM. Most of the examples we used are large benchmark circuits. We used the partitions of in [4] for the examples from s5378 to s38584 in Table 1, while we used the partitions from VIS 1.3 for the remaining examples. Table 1 compares LMBM to MBM, RFBF, and TFBF in terms of the number of reachable states and Table 2 compares time and peak memory usage. As alluded to in Section 1, RFBF and TFBF are likely to exhaust memory as the size of designs becomes large. Even when both methods completed, those methods took roughly 10X-40X more time than both MBM and LMBM. However, LMBM completed all examples, and even though it produced the same reachable states as RFBF, LMBM was roughly as fast as MBM (even faster than MBM in 6 cases out of 10) and it used even less peak memory than MBM in 5 cases.

We have also implemented TLMBM, which is a combination of TFBF and LMBM, and compared it to TMBM in Table 3. In this experiment, we set the TFBF iterations to 10. We have found 4 cases in which TLMBM produced tighter upper bounds than TMBM, and for the remaining cases, both methods produced the same results. The cases shown in Table 1, but not in Table 3, converge before the transition to either MBM or LMBM, except avq in which both
method from that class, which is called LMBM. LMBM produces the same upper bounds as RFBF; it is as fast as MBM and more accurate. We proved that Least Fixpoint Approximations converge to a valid upper bound and that they all produce the same upper bound as RFBF. Experimental results show that LMBM is a very robust way to get accurate approximations to the reachable state set for large circuits. When combined with TFBF to form TLMBM, LMBM provides a more accurate alternative to TMBM.

Given that the results of Least Fixpoint Approximations are independent of the ordering of the evaluations of the submachines, it makes sense to further study efficient orders of exploration of the state space.

References


