

Formal Verification: A New Partial Order Approach

Lubomir Ivanov

Department of Computer Science
Iona College
715 North Avenue
New Rochelle, NY 10801
livanov@iona.edu

Ramakrishna Nunna

Department of Electrical and Computer Engineering
California State University Fresno, MS 94
Fresno, CA 93740
rnunna@csufresno.edu

Abstract

Verification methodologies are trying to catch up with the increasing functionality and complexity of ASICs and Systems on Chips. Traditional simulation based procedures, though vital in certain segments of the overall design, are not efficient for large scale structures. In this paper, we explore the suitability of partial orders for formal verification of hardware. We present a new partial order verification approach based on the inductively defined notion of a series-parallel poset. Series-parallel posets can be used to model the behavior of combinational and finite state machine systems. We also show how to define temporal verification properties and how to check for the satisfaction of these properties within the behavior of the system.

Introduction

With the recent advances in computer technology - both at the hardware and software level - the necessity for formal verification of new designs has become more and more apparent. This resurgence of interest in formal verification has led to the development of hundreds of new methods for proving the correctness of hardware and/or software systems. Some of these methods have gained tremendous popularity and have been the basis for the development of industrial-level verification tools (HOL, SMV, FormalCheck, etc.). A very good overview of the field of formal verification can be found in [KM 93]

One popular partial order approach is based on the notion of a Petri Net introduced by Nielsen, Plotkin, and Winskel in [NPW 81]. This approach has been used extensively to model protocols and other structures. More recently, other partial verification methods have emerged [PG 95], [NG 96], [DP 96].

The main appeal of partial order based verification is in its clarity and intuitiveness. All partial order methods exploit the natural idea of ordering or independence of events occurring in a system. In some cases, partial order methods have been shown to reduce the complexity of verifying properties of asynchronous circuits from exponential to polynomial. Other strengths of the partial order approach include its extensibility and relative ease of automation.

In this paper we present a new partial order verification method based on the inductively defined notion of a series-parallel poset. Using this method, we can describe and verify properties of the complex interactions existing in combinational and sequential logic systems' structure and

behavior at the low-end to complete top-level behavioral descriptions at the high end.

Series Parallel Posets

A partially ordered set (poset) is a set with a reflexive, antisymmetric, and transitive relation defined on the set elements.

A Σ^* -labeled poset $P=(P, \leq, l)$ consists of a poset (P, \leq) , and an assignment of a nonempty word (i.e. a label) $l(v) \in \Sigma^*$ to each vertex v in P . We define two operations on labeled posets:

Given posets P and Q with $P \cap Q = \emptyset$,

Concatenation: $P \bullet Q := (P \cup Q, \leq_{P \bullet Q})$

Shuffle: $P \otimes Q := (P \cup Q, \leq_{P \otimes Q})$,

where: $v \leq_{P \bullet Q} v' \Leftrightarrow v \leq_P v' \vee v \leq_Q v' \vee (v \in P \wedge v' \in Q)$

$v \leq_{P \otimes Q} v' \Leftrightarrow v \leq_P v' \vee v \leq_Q v'$

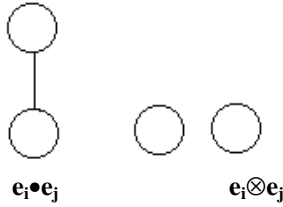
A Series-Parallel Poset is then defined inductively as follows:

- The empty poset, I , is a SPP
- The singleton posets labeled σ , for each $\sigma \in \Sigma$
- If P and Q are series-parallel posets, so are $P \bullet Q$ and $P \otimes Q$

The set of all series parallel posets, denoted $SP(\Sigma^*)$, with concatenation and shuffle, forms a bimonoid, with identity the empty poset I . Bloom and Esik proved that $SP(\Sigma^*)$ is freely generated in the variety of all bimonoids by the set Σ^* .

For our purposes, the intuitive interpretation of series-parallel posets will be as follows:

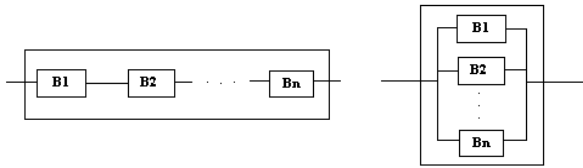
Let each event, e_i , occurring in a system be represented by a singleton poset, e_i . If event e_i follows event e_j , then $e_i < e_j$, and can be represented as $e_i \bullet e_j$. On the other hand, the independence of the events e_i and e_j can be represented by the series-parallel poset $e_i \otimes e_j$. Pictorially one can describe the two situations as follows:



Alternately, if B_1 and B_2 are two series-parallel posets, the series-parallel poset $B_1 \bullet B_2$ represents the occurrence of the events in B_1 followed by the occurrence of the events in B_2 , whereas $B_1 \otimes B_2$ represents the independent occurrence of the events of B_1 and B_2 .

Modeling Behaviors of Non-Iterated Systems

Interpreting series-parallel posets as descriptions of the dependence or independence of sets of events allows us to model the behavior of a system in terms of the sequences of events occurring during its operation. More specifically, we can consider two types of system behavior. In one case the behavior of the system is determined by the independent operation of a number of sub-components. In the other case, the system behavior is described in terms of the computation of a number of components in sequence.



In the case of the system consisting of independent sub-components, the behavior can be described as a series-parallel poset in concatenation form:

$$B = B_1 \bullet B_2 \bullet \dots \bullet B_n$$

where B_i are themselves series-parallel posets.

In the case of the system consisting of sub-components in series, the behavior can be represented in shuffle of series-parallel sub-posets form:

$$B = B_1 \otimes B_2 \otimes \dots \otimes B_n$$

Thus, we define the behavior of a non-iterated system to be a series-parallel poset in shuffle or concatenation form.

The properties we are interested in verifying involve the occurrence of events in a certain order. Therefore, it makes sense to represent these properties as series-parallel posets as well. These posets represent our expectation for the order of occurrence of certain sets of events - a subset of the set of all system events.

The verification questions will be specified as predicates over series-parallel posets which verify that a particular property is satisfied within a system behavior. The specific predicates are:

- **AS**(B, P), which we shall read as “Property P is always satisfied within the behavior B ”, is a binary predicate which takes two series-parallel posets and verifies that the events in P always occur in the specified order within the system behavior, B .
- **SS**(B, P), which we shall read as “Property P is sometimes satisfied within the behavior B ”, is a binary predicate which takes two series-parallel posets and verifies that the events in P can sometimes occur in the specified order within the system behavior, B .
- **Independent_B**(P, Q) which we shall read as “The two sets of events P and Q are independent within the behavior B .” This predicate verifies that no event in P is a predecessor of an event in Q and vice versa.

The formal definitions of the predicates are given below:

- Independent_B**(P, Q) iff

$$L(\text{pred}_B(\text{set}(P))) \cap L(\text{set}(Q)) = \emptyset \wedge$$

$$L(\text{pred}_B(\text{set}(Q))) \cap L(\text{set}(P)) = \emptyset$$
- AS**(B, P) iff :
 - $|P| = 1 \wedge l(P) \in \Sigma_B$
 - $P = P_1 \bullet P_2 \bullet \dots \bullet P_m \wedge$
 $\forall i \in [m] \text{AS}(B, P_i) \wedge$
 $\forall i \in [m-1] [\forall e \in P_{i+1}$
 $(L(\text{pred}_B(\{e\})) \cap L(\text{set}(P_i)) = L(\text{set}(P_i)))]$
 - $P = P_1 \otimes P_2 \otimes \dots \otimes P_m \wedge$
 $\forall i \in [m] \text{AS}(B, P_i) \wedge$
 $\forall i \in [m-1] \text{Independent}_B(P_i, P_{i+1})$
- SS**(B, P) iff :
 - $|P| = 1 \wedge l(P) \in \Sigma_B$
 - $P = P_1 \bullet P_2 \bullet \dots \bullet P_m \wedge$
 $\forall i \in [m] \text{SS}(B, P_i) \wedge$
 $\forall i \in [m-1] [\forall e \in P_{i+1} (L(\text{pred}_B(\{e\})) \cap L(\text{set}(P_i)) \neq \emptyset$
 $\vee \text{Independent}_B(P_i, P_{i+1})]$
 - $P = P_1 \otimes P_2 \otimes \dots \otimes P_m \wedge$

$$\forall i \in [m] \text{SS}(\mathbf{B}, P_i) \wedge \\ \forall i \in [m-1] \text{Independent}_{\mathbf{B}}(P_i, P_{i+1})$$

where

$$l: S \rightarrow \Sigma, l(s) = \sigma, s \in S \text{ and } \sigma \in \Sigma, \\ L: P(S) \rightarrow P(\Sigma), L(\{s_i \mid 0 \leq i < n\}) = \{l(s_i) \mid 0 \leq i < n\}$$

are two labeling functions with S - the set of singleton posets, $P(S)$ - the powerset of S , and Σ - the alphabet of system event labels, and $\text{pred}_{\mathbf{B}}(\mathbf{P})$ is a predecessor function which takes the set of vertices of a series-parallel poset \mathbf{P} and returns the set of predecessors of those vertices in a series-parallel poset \mathbf{B} , or \emptyset if the set is empty. The function $\text{set}(\mathbf{P})$ takes a series-parallel poset \mathbf{P} and returns the set of vertices of that poset.

More details as well as an intuitive interpretation and justification for the above definitions can be found in [INB'99].

The above definitions give rise to a verification algorithm for non-iterated systems. The worst case time complexity of the algorithm is $O(mn \log n)$, where $m = |\mathbf{B}|$ and $n = |\mathbf{P}|$. Usually, $m \gg n$ since the system behavior involves all possible events occurring in the system (usually hundreds of thousands), whereas a typical property that we are trying to verify involves only a few of these events.

Complexity Reduction

Let us define a function Pr (for Projection), which takes two arguments - $\text{set}(\mathbf{P})$ and \mathbf{B} - and substitutes the empty poset, \mathbf{I} , for every event in \mathbf{B} which is not in $\text{set}(\mathbf{P})$. The intuition behind this function is that we ignore all events in the system behavior except the ones which are part of the property we are trying to verify.

Theorem

Let $\mathbf{B}' = Pr(\mathbf{B}, \text{set}(\mathbf{P}))$. Then:

- $\text{SS}(\mathbf{B}, \mathbf{P})$ iff $\text{SS}(\mathbf{B}', \mathbf{P})$
- $\text{AS}(\mathbf{B}, \mathbf{P})$ iff $\text{AS}(\mathbf{B}', \mathbf{P})$

Proof:

Notice that the projection function Pr does not modify the ordering in \mathbf{B} of the events of $\text{set}(\mathbf{P})$. Thus, for any event, e , in $\text{set}(\mathbf{B}) \cap \text{set}(\mathbf{P})$, $\text{pred}_{\mathbf{B}}(\{e\}) = \text{pred}_{\mathbf{B}'}(\{e\})$.

Then, two series-parallel posets \mathbf{P} and \mathbf{Q} are independent in \mathbf{B} if and only if they are independent in \mathbf{B}' , i.e.

$$\text{Independent}_{\mathbf{B}}(\mathbf{P}, \mathbf{Q}) \Leftrightarrow \\ L(\text{pred}_{\mathbf{B}}(\text{set}(\mathbf{P}))) \cap L(\text{set}(\mathbf{Q})) = \emptyset \wedge \\ L(\text{pred}_{\mathbf{B}}(\text{set}(\mathbf{Q}))) \cap L(\text{set}(\mathbf{P})) = \emptyset \\ \Leftrightarrow L(\text{pred}_{\mathbf{B}'}(\text{set}(\mathbf{P}))) \cap L(\text{set}(\mathbf{Q})) = \emptyset \wedge \\ L(\text{pred}_{\mathbf{B}'}(\text{set}(\mathbf{Q}))) \cap L(\text{set}(\mathbf{P})) = \emptyset \\ \Leftrightarrow \text{Independent}_{\mathbf{B}'}(\mathbf{P}, \mathbf{Q}).$$

We can then show inductively that $\text{SS}(\mathbf{B}, \mathbf{P})$ iff $\text{SS}(\mathbf{B}', \mathbf{P})$ and $\text{AS}(\mathbf{B}, \mathbf{P})$ iff $\text{AS}(\mathbf{B}', \mathbf{P})$.

The argument is based on the following:

$\text{SS}(\mathbf{B}, \mathbf{P})$ iff

- $(|\mathbf{P}|=1 \wedge l(\mathbf{P}) \in \Sigma_{\mathbf{B}}) \vee$
- $(\mathbf{P} = \mathbf{P}_1 \bullet \mathbf{P}_2 \bullet \dots \bullet \mathbf{P}_m \wedge \forall i \in [m] \text{SS}(\mathbf{B}, P_i) \wedge \forall i \in [m-1][\forall e \in \mathbf{P}_{i+1}(L(\text{pred}_{\mathbf{B}}(\{e\}))) \cap L(\text{set}(\mathbf{P}_i)) \neq \emptyset \vee \text{Independent}_{\mathbf{B}}(P_i, P_{i+1})]) \vee$
- $(\mathbf{P} = \mathbf{P}_1 \otimes \mathbf{P}_2 \otimes \dots \otimes \mathbf{P}_m \wedge \forall i \in [m] [\text{SS}(\mathbf{B}, P_i)] \wedge \forall i \in [m-1] \text{Independent}_{\mathbf{B}}(P_i, P_{i+1}))$

\Leftrightarrow

- $(|\mathbf{P}| = 1 \wedge l(\mathbf{P}) \in \Sigma_{\mathbf{B}}) \vee$
- $(\mathbf{P} = \mathbf{P}_1 \bullet \mathbf{P}_2 \bullet \dots \bullet \mathbf{P}_m \wedge \forall i \in [m] \text{SS}(\mathbf{B}', P_i) \wedge \forall i \in [m-1][\forall e \in \mathbf{P}_{i+1}(L(\text{pred}_{\mathbf{B}'}(\{e\}))) \cap L(\text{set}(\mathbf{P}_i)) \neq \emptyset \vee \text{Independent}_{\mathbf{B}'}(P_i, P_{i+1})]) \vee$
- $(\mathbf{P} = \mathbf{P}_1 \otimes \mathbf{P}_2 \otimes \dots \otimes \mathbf{P}_m \wedge \forall i \in [m] \text{SS}(\mathbf{B}', P_i) \wedge \forall i \in [m-1] \text{Independent}_{\mathbf{B}'}(P_i, P_{i+1}))$

$\Leftrightarrow \text{SS}(\mathbf{B}', \mathbf{P})$.

A similar argument shows that $\text{AS}(\mathbf{B}, \mathbf{P}) \Leftrightarrow \text{AS}(\mathbf{B}', \mathbf{P})$.

This simple observation lead to a substantial decrease in the complexity of the verification algorithms. If $|\mathbf{B}| = m$ and $|\mathbf{P}| = n$, then after taking the projection of \mathbf{B} with respect to $\text{set}(\mathbf{P})$, the size of the new behavior poset is bound by the size of the property poset, i.e. $|\mathbf{B}'| = n$. This is true because each event can appear only once in a behavior poset, and there are $(n-1)$ operations concatenation or shuffle in the expression. Thus the complexity of the verification algorithm is reduced from $O(mn \log n)$ to $O(n^2 \log n)$. Considering that $m \gg n$, one can see that the reduction is significant.

Modeling Behavior of Iterated Systems

Similar reasoning about events occurring in a system and their temporal relation leads to analogous algorithms for the case of systems involving iteration.

An iterated system is a system which employs feedback in its own operation or the operation of one or more of its components. Examples of iterated systems at the hardware level include the following:

- At the logic gate level, any sequential circuit (e.g. a flip-flop), which involves feedback
- At the system level, any feedback control system
- Various protocols for serial and parallel data communication, involving the repeated sending and receiving of data

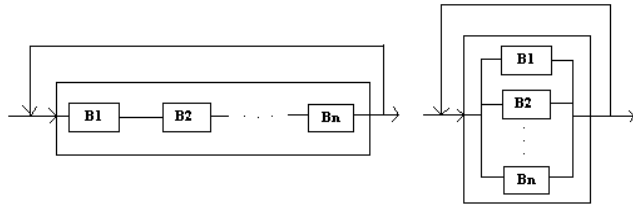
To model the behavior of these systems, though, the notion of series-parallel posets is insufficient. We need to enrich out collection of operations with two more - + and Kleene

star, *. Together with shuffle and concatenation, they define a new structure - the star shuffle semiring $S = (S, +, \bullet, \otimes, *, \emptyset, I)$ of series-parallel posets, defined as follows:

- S - the set of finite subsets of $SP(\Sigma^*)$, closed under the semiring operations
- If $K \in S$ and $L \in S$, $K+L = \{P \mid P \in K \vee P \in L\} \in S$
- If $K \in S$ and $L \in S$, $K \bullet L = \{P \bullet Q \mid P \in K \wedge Q \in L\} \in S$
- If $K \in S$ and $L \in S$, $K \otimes L = \{P \otimes Q \mid P \in K \wedge Q \in L\} \in S$
- If $K \in S$, then $K^* = I + K + K^2 + \dots = \bigcup_{i=0, \dots, \infty} K^i \in S$
- \emptyset is the empty set of posets
- I is the empty poset

Now, the behavior $B \in S$ and the verification property $P \in S$. As in the case of non-iterated systems, we can define standard forms for behaviors and properties, and consider the verification question in each case, defined as predicates over behaviors and properties.

First, consider a somewhat more restricted model of an iterated system - a globally iterated/locally non-iterated system (GILNIS). These systems are comprised of a number of components which, by themselves, do not involve iteration. The global system output, though, is fed back as an input for another iteration. Pictorially, the two cases are presented below:



The respective behavior forms become:

- $B = (B_1 \bullet B_2 \bullet \dots \bullet B_n)^*$
- $B = (B_1 \otimes B_2 \otimes \dots \otimes B_n)^*$

where $B^* = I + B + B^2 + B^3 + \dots = \bigcup_{i=0, \dots, \infty} B^i$, and each B_i is a series-parallel poset.

There are 3 corresponding forms for the property posets:

- $P = P_1 \bullet P_2 \bullet \dots \bullet P_m$
- $P = P_1 \otimes P_2 \otimes \dots \otimes P_m$
- $P = P_1^*$

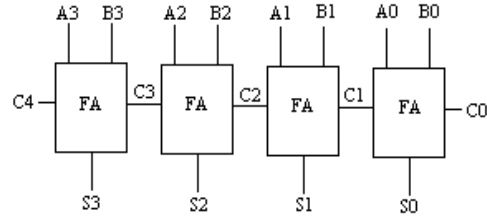
Next, we can consider general iterated systems, which involve global as well as local iteration, i.e. the system consists of components which are iterated systems themselves. In this case, the behavior and the properties can be in one of the following forms:

- $B = B_1 \bullet B_2 \bullet \dots \bullet B_n$
- $B = B_1 \otimes B_2 \otimes \dots \otimes B_n$
- $B = B_1 + B_2 + \dots + B_n$
- $B = B_1^*$
- $P = P_1 \bullet P_2 \bullet \dots \bullet P_m$
- $P = P_1 \otimes P_2 \otimes \dots \otimes P_m$
- $P = P_1^*$

In each case, as in the case of non-iterated systems, appropriate verification predicates are defined. These are the basis for algorithms that verify the properties of iterated systems. Similar reduction, methodologies have been developed for the case of globally-iterated/locally-non-iterated as well as general iterated systems.

An illustrative example:

The proposed methodology can be used to model and reason about complex structures. However, in this section, we will demonstrate the modeling capabilities using a simple 4-bit binary adder example.



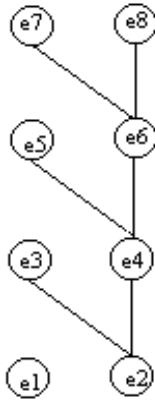
The events we want to represent are the following:

- e_1 - Full Adder 0 produces the sum bit S0
- e_2 - Full Adder 0 produces the carry out bit C1
- e_3 - Full Adder 1 produces the sum bit S1
- e_4 - Full Adder 1 produces the carry out bit C2
- e_5 - Full Adder 2 produces the sum bit S2
- e_6 - Full Adder 2 produces the carry out bit C3
- e_7 - Full Adder 3 produces the sum bit S3
- e_8 - Full Adder 3 produces the carry out bit C4

The relationship between the events are dictated by the structure of the circuit:

- e_3 and e_4 cannot occur before e_2 has occurred
- e_5 and e_6 cannot occur before e_4 has occurred
- e_7 and e_8 cannot occur before e_6 has occurred

This leads to the following behavior:



$$B = (e_1 \otimes (e_2 (e_3 \otimes (e_4 (e_5 \otimes (e_6 (e_7 \otimes e_8)))))))$$

We can now try to verify some properties:

- a) "Are the sum bits S1 and S3 independent of each other?"

Independent_B(e₃, e₇)?

$$L(\text{pred}(\{e_3\})) = \{e_2\} \cap \{e_7\} = \emptyset$$

$$L(\text{pred}(\{e_7\})) = \{e_2, e_4, e_6\} \cap \{e_3\} = \emptyset$$

\Rightarrow **Independent_B(e₃, e₇)** is TRUE

- b) "Is the sum bit S2 independent of the carry bit C2?"

Independent_B(e₂, e₅)?

$$L(\text{pred}(\{e_5\})) = \{e_2, e_4\} \cap \{e_2\} = \{e_2\} \neq \emptyset$$

\Rightarrow **Independent_B(e₂, e₅)** is FALSE

- c) "Does C1 always precede S1?"

AS_NI(B, e₂•e₃)?

Verify **AS_NI(B, e₂)** : TRUE

Verify **AS_NI(B, e₃)** : TRUE

Verify the interconnection e₂•e₃:

$$L(\text{pred}(\{e_3\})) = \{e_2\} \cap \{e_2\} = \{e_2\} \Rightarrow \text{verified}$$

\Rightarrow **AS_NI(B, e₂•e₃)** is TRUE

Conclusion

In this paper we presented an outline of a new formal verification approach for hardware systems with or without iteration as well as for communication protocols. The approach is general enough not to be restricted to hardware systems only. We hope to extend the work to verification of properties of software as well as embedded systems. We also plan to implement the developed algorithms in a software package that will allow user-friendly, automatic verification of system properties from a given set of specifications.

References:

- [KM 93] K. McMillan, "Symbolic Model Checking", Kluwer Academic Publishing, 1993
- [NPW 81] M. Nielsen, G. Plotkin, G. Winskel, "Petri Nets, Event Structures and Domains, part 1", Theoretical Computer Science, 13:85-108, 1981

[PG 95] P. Godefroid, "Partial Order Methods for the Verification of Concurrent Systems: an Approach to the State Explosion Problem", Doctoral Dissertation, University of Liege, 1994-95

[NG 96] R. Nalumasu, G. Gopalakrishnan, "A New Partial Order Reduction Algorithm for Concurrent System Verification", IFIP, 1996

[DP 96] D. Peled, "Combining Partial Order Reductions with On-the-Fly Model Checking", Journal of Formal Methods in Systems Design, 8 (1):39-64, 1996

[BE'96] S.Bloom, Z.Esik, "Free Shuffle Algebras in Language Varieties", Theoretical Computer Science 163 (1996), 55-89, Elsevier

[INB'99] L.Ivanov, R.Nunna, S.Bloom, "Modeling and Analysis of Non-Iterated Systems: An Approach based upon Series-Parallel Posets", Proceedings of ISCAS'99, Orlando, FL, May 30th - June 2nd 1999