

Refinement for Symbolic Trajectory Evaluation

Authors

Chalmers

Abstract. Model refinement such that it preserves symbolic trajectory evaluations.

Keywords: STE · Refinement · ?

1 Introduction to STE

1.1 Original STE

Symbolic trajectory evaluation [5] (STE) is a high-performance model checking technique based on *symbolic simulation* extended with a temporal *next-time* operator to describe circuit behaviour over time. In its simplest form, STE tests the validity of an *assertion* of the form $A \Rightarrow C$, where both the *antecedent* A and *consequent* C are formulas in the following logic:

$$f ::= p \mid f \wedge f \mid P \rightarrow f \mid \mathbf{N} f$$

Here, p is a simple predicate over “values” in a circuit and P is a Boolean propositional formula, and the operators \wedge , \rightarrow and \mathbf{N} are conjunction, domain restriction and the next-time operator, respectively.

If the circuit contains Boolean signals, p is typically drawn from the following two predicates: $n \text{ is } 1$ and $n \text{ is } 0$, where n ranges over the signals (or nodes) in a circuit. For example, suppose we have a unit-delayed, two-input AND-gate, then it is reasonable to assume that the assertion $(in_1 \text{ is } 1 \wedge in_2 \text{ is } 1) \Rightarrow \mathbf{N}(out \text{ is } 1)$ is true. Indeed, STE efficiently validates such statements for us.

While the truth semantics of an assertion in STE is defined as the satisfaction of its “defining” trajectory (bounded sequence of states) relative to a model structure of the circuit, what the STE algorithm computes is exactly the solution of a data-flow equation [1] in the classic format [4]. . . .

1.2 Set-theoretic STE

Consider an arbitrary, but fixed, digital circuit M operating in discrete time. A *configuration* of M , denoted by C , is non-empty and finite set that represents a snapshot of M at a discrete point in time. If the circuit M has m boolean signals, then its set of configurations is typically represented as a sequence \mathbb{B}^m , where $\mathbb{B} = \{0, 1\}$ is the set of boolean values.

Circuit Model A simple conceptual model of M is a *transition relation*, $M_R \subseteq C \times C$, where $(c, c') \in M_R$ means that M can move from c to c' in one step¹. The power set of C , denoted by $\mathcal{P}(C)$, can be viewed as the set of *predicates* on configurations, where \cap , \cup , and \subseteq correspond to conjunction, disjunction and implication, respectively. We denote by $\cap Q$ and $\cup Q$ the intersection and union of all members of any $Q \subseteq \mathcal{P}(C)$.

M_R induces a *predicate transformer* $M_F \in \mathcal{P}(C) \rightarrow \mathcal{P}(C)$ using the relational image operation:

$$M_F(p) = \{c' \in C \mid \exists c \in p : (c, c') \in M_R\}$$

It is intuitively obvious that if M is in one of the configurations in $p \in \mathcal{P}(C)$, then in one time step it must be in one of the configurations in $M_F(p)$. We also see that M_F distributes over arbitrary unions:

$$M_F(\cup Q) = \cup \{M_F(q) \mid q \in Q\}$$

for all $Q \subseteq \mathcal{P}(C)$. In general, any M_F that satisfies this distributive property also defines a M_R through the equivalence $(c, c') \in M_R \Leftrightarrow c' \in M_F(\{c\})$, that is to say, there is no loss of information going from M_R to M_F or vice versa. We adopt this functional model of M and drop its subscript.

Exactly what C and its signals are, is not important in this section. In practice, however, signals are typically divided into external, i.e. “input” and “output”, and internal parts. While an input signal is generally controlled by the external environment, and thus unconstrained by M itself, non-input signals are determined by the circuit topology and functionality. For example, supposed M is the earlier example of a unit-delayed two-input AND gate, we could then define its model $M \in \mathcal{P}(\mathbb{B}^3) \rightarrow \mathcal{P}(\mathbb{B}^3)$ as follows:

$$M(p) = \{\langle b_1, b_2, i_1 \wedge i_2 \rangle \in \mathbb{B}^3 \mid \langle i_1, i_2, o \rangle \in p\}$$

Here i_1 and i_2 refer to the two inputs of the AND gate, o the ignored output, and b_1 and b_2 are unconstrained inputs for the new configurations.

Assertions and satisfaction A *trajectory assertion* for M is quintuple $A = (S, s_0, R, \pi_a, \pi_c)$, where S is a finite set of *states*, $s_0 \in S$ is an *initial state*, $R \subseteq S \times S$ is a *transition relation*, $\pi_a \in S \rightarrow \mathcal{P}(C)$ and $\pi_c \in S \rightarrow \mathcal{P}(C)$ label each state s with an *antecedent* $\pi_a(s)$ and a *consequent* $\pi_c(s)$. We assume that $(s, s_0) \notin R$ for all $s \in S$ without any loss of generality.

The circuit model M intuitively *satisfies* an assertion A if, for every *trajectory* τ through M and every *run* ρ through A , τ satisfying the antecedents of ρ entails that τ also satisfies the consequents of ρ . To be more specific, a *trajectory* of M is a non-empty sequences of configurations, $\tau \in C^+$, such that $\tau_n \in M(\{\tau_{n-1}\})$ for all $n \in \mathbb{N} : 0 < n < |\tau|$. And a *run* of A is a non-empty sequence of states, $\rho \in S^+$, such that $\rho_0 = s_0$ and $(\rho_{n-1}, \rho_n) \in R$ for all $n \in \mathbb{N} : 0 < n < |\rho|$.

¹ Mention how this affects circuits with zero-delays?

We can then say that a τ satisfies the antecedents of ρ , denoted by $\tau \models_a \rho$, iff $\tau_n \in \pi_a(\rho_n)$ for all $n \in \mathbb{N} : n < |\tau| = |\rho|$; satisfaction of consequents is defined similarly with π_c instead and denoted by $\tau \models_c \rho$. Finally, that M satisfies A , denoted by $M \models A$, can be formalized as²:

$$\forall \tau \in \text{Traj}(M) : \forall \rho \in \text{Runs}(A) : |\tau| = |\rho| \Rightarrow (\tau \models_a \rho \Rightarrow \tau \models_c \rho)$$

where $\text{Traj}(M)$ and $\text{Runs}(A)$ denote the sets of all trajectories of M and runs of A , respectively.

1.3 Lattice-theoretic STE

Manipulating subsets of \mathbb{B}^m is impractical for even moderately large m , which leads us to one of the key insights of STE. Namely, instead of manipulating subsets of \mathbb{B}^m directly, one can use sequences of ternary values $\mathbb{T} = \mathbb{B} \cup \{X\}$ to approximate them, whose sizes are only linear in m . Here the 1 and 0 from \mathbb{B} denotes specific, defined values whereas X denotes an “unknown” value that could be either 1 or 0. This intuition induces a partial order \sqsubseteq on \mathbb{T} , where $0 \sqsubseteq X$ and $1 \sqsubseteq X$ ³. For any $m \in \mathbb{N}$, this ordering on \mathbb{T} is lifted component-wise to \mathbb{T}^m .

Note that \mathbb{T}^m does not quite form a complete lattice because it lacks a bottom: both $0 \sqsubseteq X$ and $1 \sqsubseteq X$ but 0 and 1 are equally defined. A special bottom element \perp is therefore introduced, such that $\perp \sqsubseteq t$ and $\perp \neq t$ for all $t \in \mathbb{T}^m$. The extended $\mathbb{T}_\perp^m = \mathbb{T}^m \cup \{\perp\}$ then becomes a complete lattice. We denote the top element $\langle X, \dots, X \rangle$ of \mathbb{T}_\perp^m by \top .

Ternary lattices Generalising from any specific domain, let (\hat{P}, \sqsubseteq) be a finite, complete lattice of *abstract predicates* in which the meet \sqcap and join \sqcup of any subset $Q \subseteq \hat{P}$ exists. Similar to the previous set operations for power sets, \sqcap , \sqcup and \sqsubseteq correspond to conjunction, disjunction and implication for abstract predicates, respectively. Furthermore, for any $Q \subseteq \hat{P}$, we denote by $\sqcap Q$ and $\sqcup Q$ the meet and join of all members of Q .

Let there be a Galois connection relating “concrete” predicates $\mathcal{P}(C)$ and abstract predicates \hat{P} . The usual definition of a Galois connection is in terms of an *abstraction* $\alpha \in \mathcal{P}(C) \rightarrow \hat{P}$ and a *concretisation* $\gamma \in \hat{P} \rightarrow \mathcal{P}(C)$ function, such that $\alpha(p) \sqsubseteq \hat{p} \Leftrightarrow p \subseteq \gamma(\hat{p})$ for all $p \in \mathcal{P}(C)$ and $\hat{p} \in \hat{P}$. For example, a Galois connection from $\mathcal{P}(\mathbb{B}^m)$ to \mathbb{T}_\perp^m for any $m \in \mathbb{N}$ can be defined in a natural way by its concretisation function $\gamma \in \mathbb{T}_\perp^m \rightarrow \mathcal{P}(\mathbb{B}^m)$:

$$\begin{aligned} \gamma(\langle t_0, \dots, t_{m-1} \rangle) &= \{ \langle b_0, \dots, b_{m-1} \rangle \in \mathbb{B}^m \mid \forall i < m : t_i \neq X \Rightarrow b_i = t_i \} \\ \gamma(\perp) &= \emptyset \end{aligned}$$

² This is equivalent to a DFA formulation [1].

³ We use the reverse ordering of what is originally used in STE.

Listing each concrete predicate approximated by a given abstract predicate. Its abstraction function $\alpha \in \mathcal{P}(\mathbb{B}^m) \rightarrow \mathbb{T}_\perp^m$ instead finds the most precise abstract predicate for a set of concrete predicates:

$$\begin{aligned}\alpha(p) &= \sqcup \{ \langle t_0, \dots, t_{m-1} \rangle \in \mathbb{T}_\perp^m \mid \langle b_0, \dots, b_{m-1} \rangle \in p, \forall i < m : b_i = t_i \} \\ \alpha(\emptyset) &= \perp\end{aligned}$$

Abstract circuit model An *abstract predicate transformer* $\hat{M} \in \hat{P} \rightarrow \hat{P}$ is an *abstract interpretation* [2] of $M \in \mathcal{P}(C) \rightarrow \mathcal{P}(C)$ iff: \hat{M} preserves \perp , i.e. $\hat{M}(\perp) = \perp$; \hat{M} is monotonic, i.e. $\hat{p} \sqsubseteq \hat{q} \Rightarrow \hat{M}(\hat{p}) \sqsubseteq \hat{M}(\hat{q})$ for all $\hat{p}, \hat{q} \in \hat{P}$; **and its Galois connection between $\mathcal{P}(C)$ and \hat{P} is a *simulation relation***, i.e. $\alpha(M(p)) \sqsubseteq \hat{M}(\alpha(p))$ for all $p \in \mathcal{P}(C)$, or $M(\gamma(\hat{p})) \subseteq \gamma(\hat{M}(\hat{p}))$ for all $\hat{p} \in \hat{P}$.

Note that \hat{M} does not distribute over arbitrary join because information is potentially discarded by the ternary logic during a join. As an example, let the following \hat{M} abstract the earlier unit-delayed AND gate:

$$\begin{aligned}\hat{M}(\langle 1, 1, p_2 \rangle) &= \langle X, X, 1 \rangle & \hat{M}(\langle 0, 0, p_2 \rangle) &= \langle X, X, 0 \rangle \\ \hat{M}(\langle 0, X, p_2 \rangle) &= \langle X, X, 0 \rangle & \hat{M}(\langle X, 0, p_2 \rangle) &= \langle X, X, X \rangle \\ \hat{M}(\langle p_0, p_1, p_2 \rangle) &= \langle X, X, X \rangle\end{aligned}$$

where the last, most general matching is overlapped by the more concrete ones. If we apply \hat{M} to the join of $\langle 0, 1, X \rangle$ and $\langle 1, 0, X \rangle$, or if we apply \hat{M} to them individually and then join, we get two different results:

$$\begin{aligned}\hat{M}(\langle 0, 1, X \rangle \sqcup \langle 1, 0, X \rangle) &= \hat{M}(\langle X, X, X \rangle) = \langle X, X, X \rangle \\ \hat{M}(\langle 0, 1, X \rangle) \sqcup \hat{M}(\langle 1, 0, X \rangle) &= \langle X, X, 0 \rangle \sqcup \langle X, X, 0 \rangle = \langle X, X, 0 \rangle\end{aligned}$$

The inequality $\sqcup \{ \hat{M}(\hat{q}) \mid \hat{q} \in \hat{Q} \} \sqsubseteq \hat{M}(\sqcup \hat{Q})$ for all $\hat{Q} \sqsubseteq \hat{P}$ does however hold, since it is implied by the monotonicity of \hat{M} .

Assertions and satisfaction A trajectory assertion for an abstract model \hat{M} is a quintuple $\hat{A} = (S, s_0, R, \hat{\pi}_a, \hat{\pi}_c)$, where S , s_0 , and R are as in section 1.2 and $\hat{\pi}_a \in S \rightarrow \hat{P}$ and $\hat{\pi}_c \in S \rightarrow \hat{P}$ label each state s with an abstract predicate for its antecedent and consequent, respectively.

Here follows the definition in [1]. For all functions $\hat{\Phi} \in S \rightarrow \hat{P}$ and states $s \in S$, define $\hat{F} \in S \rightarrow (\hat{P} \rightarrow \hat{P})$ and $\hat{J} \in (S \rightarrow \hat{P}) \rightarrow (S \rightarrow \hat{P})$ as follows:

$$\hat{F}(s)(\hat{p}) = \hat{M}(\pi_a(s) \sqcap \hat{p}) \tag{1}$$

$$\hat{J}(\hat{\Phi})(s) = \text{if } (s = s_0) \text{ then } \top \text{ else } \sqcup \{ \hat{F}(s')(\hat{\Phi}(s')) \mid (s', s) \in R \} \tag{2}$$

\hat{F} preserves \perp , and both \hat{F} and \hat{J} are monotonic; two $\hat{\Phi}, \hat{\Phi}' \in S \rightarrow \hat{P}$ are ordered as $\hat{\Phi} \sqsubseteq \hat{\Phi}' \Leftrightarrow \forall s \in S : \hat{\Phi}(s) \sqsubseteq \hat{\Phi}'(s)$. Let $\hat{\Phi}_* \in S \rightarrow \hat{P}$ be the least fixpoint of the equation $\hat{\Phi} = \hat{J}(\hat{\Phi})$ [3]. Since both S and \hat{P} are finite, $\hat{\Phi}_*$ is given by $\lim \hat{\Phi}_n(s)$, where $\hat{\Phi}_n$ is defined as follows:

$$\hat{\Phi}_n = \text{if } (n = 0) \text{ then } (\lambda s \in S : \perp) \text{ else } \hat{\mathcal{F}}(\hat{\Phi}_{n-1}) \quad (3)$$

\hat{M} satisfies a trajectory assertion \hat{A}^4 , denoted by $\hat{M} \models_{\text{lat}} \hat{A}$, iff $\forall s \in S : \hat{\Phi}_*(s) \sqcap \pi_\alpha(s) \sqsubseteq \pi_c(s)$.

2 Refinement

2.1 Set-Theoretic refinement

Consider another fixed, but arbitrary, circuit model $N \in \mathcal{P}(D) \rightarrow \mathcal{P}(D)$, where D is a non-empty and finite set of configurations which intersects the earlier set C . Exactly what configurations such as C and D are, were not important previously. To reason about refinement, however, we need to make a distinction between their external and internal elements. The rationale is that refinement relates the visible behaviour of circuits. **Unnecessary to assume that internal states can be related.**

Let the visible components visible elements of configurations in D be identified by two projection mappings, $o \in \mathcal{P}(D) \rightarrow \mathcal{P}(D)$ and $i \in \mathcal{P}(D) \rightarrow \mathcal{P}(D)$, identifying its “inputs” and “outputs”, respectively. The set of all possible inputs and outputs in N are given by the images $i[\mathcal{P}(D)] = \{i(p) \mid p \in \mathcal{P}(D)\}$ and $o[\mathcal{P}(D)]$. Note that, since models generally cannot control their input signals, transitions in N from a state $d \in \mathcal{P}(D)$ are driven by its intersection with a chosen input $i \in i[\mathcal{P}(D)]$, that is, $N(d \cap i)$ yields the next state of d for input i . We assume the inputs and outputs of M are contained by N , i.e. $i[\mathcal{P}(C)] \subseteq i[\mathcal{P}(D)]$ and $o[\mathcal{P}(C)] \subseteq o[\mathcal{P}(D)]$.

A *driver* $\delta \in i[\mathcal{P}(D)]^+$ for a circuit N is a non-empty sequences of inputs, which induces a trajectory $\tau \in D^+$ for N , denoted by N_δ , where $\tau_0 = D$ and $\tau_{n+1} = N(\tau_n \cap \delta_n)$ for all $n \in \mathbb{N} : n < |\delta|$. With a slight abuse of notation, we extend $o(\cdot)$ to sequences component-wise and overload it to accept configurations in C . We then say the circuit M *refines* the circuit N , denoted by $M \leq N$, iff $\forall \delta \in i[\mathcal{P}(C)]^+ : o(M_\delta) \subseteq o(N_\delta)$.

Recall that a trajectory assertion for N is a quintuple $A = (S, s_0, R, \pi_a, \pi_c)$. If π_a only mention inputs and π_c outputs, i.e. $\pi_a[S] \subseteq i[\mathcal{P}(D)]$ and $\pi_c[S] \subseteq o[\mathcal{P}(D)]$, then we refer to A as a *visible trajectory assertion* which we denote by A_{vis} . Define $\alpha(A) = (S, s_0, R, \alpha(\pi_a), \alpha(\pi_c))$, where $\alpha(\pi_a) = \lambda s \in S : \alpha(\pi_a(s))$ and $\alpha(\pi_c) = \lambda s \in S : \alpha(\pi_c(s))$. Note that, if A is an assertion for N , then $\alpha(A)$ is an assertion for M .

Theorem 1. *If M and N are circuit models such that $M \leq N$, and A_{vis} is a visible trajectory assertion for N , then:*

$$N \models A_{\text{vis}} \Rightarrow M \models \alpha(A_{\text{vis}})$$

⁴ That \hat{M} satisfies \hat{A} implies that a concretisation of \hat{A} can also be satisfied by the original, set-based model M [1].

We say that $\ll \subseteq \mathcal{P}(C) \times \mathcal{P}(D)$ is a *visible simulation relation* between configurations in M and N iff $c \ll d$ implies that $\text{o}(c) \subseteq \text{o}(d)$ and for all inputs $i \in i[\mathcal{P}(C)]$, their next states are related $M(i \cap c) \ll N(i \cap d)$. We then say that M *refines* N by *set-theoretic simulation*, denoted by $M \leq_{\text{set}} N$, iff their most general states are related $C \ll D$.

Theorem 2. *For circuit models M and N :*

$$M \leq N \Leftrightarrow M \leq_{\text{set}} N$$

2.2 Lattice-Theoretic refinement

Let $\hat{N} \in \hat{Q} \rightarrow \hat{Q}$ be an abstract interpretation of N , where \hat{Q} is an abstract predicate for which there is a Galois connection to $\mathcal{P}(D)$.

Let the output of an abstract predicate \hat{Q} be identified by the idempotent mapping $\hat{\text{o}} \in \hat{Q} \rightarrow \hat{Q}$, such that $\hat{\text{o}}(\cdot)$ is monotonic, i.e. $\hat{p} \sqsubseteq \hat{q} \Rightarrow \hat{\text{o}}(\hat{p}) \sqsubseteq \hat{\text{o}}(\hat{q})$; and the greatest bound for an output, i.e. $\hat{\text{o}}(\hat{p}) \sqsubseteq \hat{\text{o}}(\hat{q}) \Leftrightarrow \hat{p} \sqsubseteq \hat{\text{o}}(\hat{q})$. In a similar vein, let the inputs of an abstract predicate \hat{Q} be identified by the mapping $\hat{\text{i}} \in \hat{Q} \rightarrow \hat{Q}$.

In the case of an unit-delayed AND gate and its predicates $\langle p_0, p_1, p_2 \rangle \in \mathbb{T}_\perp^3$, for example, $\hat{\text{i}}(\cdot)$ and $\hat{\text{o}}(\cdot)$ can be defined naturally as projections which maps each non-input and non-output element to X , respectively:

$$\begin{aligned} \hat{\text{i}}(\langle \hat{p}_0, \hat{p}_1, \hat{p}_2 \rangle) &= \langle \hat{p}_0, \hat{p}_1, X \rangle & \hat{\text{o}}(\langle \hat{p}_0, \hat{p}_1, \hat{p}_2 \rangle) &= \langle X, X, p_2 \rangle \\ \hat{\text{i}}(\perp) &= \perp & \hat{\text{o}}(\perp) &= \perp \end{aligned}$$

It is easy to see that $\hat{\text{o}}(\cdot)$ is monotonic, and that it produces the greatest bound for any \mathbb{T}^3 with an output $\sqsubseteq \hat{p}_2$.

Let $\lll \subseteq \hat{P} \times \hat{Q}$ be a Galois connection given by means of a binary relation between \hat{P} and \hat{Q} , such that elements of subsets $\hat{P}^* \subseteq \hat{P}$ and $\hat{Q}^* \subseteq \hat{Q}$ are related iff their join and meet are related:

$$\forall \hat{p} \in \hat{P}^* : \forall \hat{q} \in \hat{Q}^* : \hat{p} \lll \hat{q} \Leftrightarrow \sqcup \hat{P}^* \lll \sqcap \hat{Q}^*$$

where $\hat{p} \lll \hat{q}$ reads as “ \hat{p} refines \hat{q} ”. Intuitively, this relation is an extension of the partial orderings inside \hat{P} and \hat{Q} to an ordering between \hat{P} and \hat{Q} . The usual abstraction $\alpha \in \hat{P} \rightarrow \hat{Q}$ and concretisation $\gamma \in \hat{Q} \rightarrow \hat{P}$ functions can be derived from \lll as follows:

$$\alpha(p) = \sqcap \{ \hat{p} \in \hat{P} \mid \hat{q} \lll \hat{p} \} \quad \gamma(\hat{p}) = \sqcup \{ \hat{q} \in \hat{Q} \mid \hat{q} \lll \hat{p} \}$$

Conversely, the relation \lll can be derived from α and γ as: $p \ll \hat{p} \Leftrightarrow \alpha(p) \sqsubseteq \hat{p}$ and $p \ll \hat{p} \Leftrightarrow p \subseteq \gamma(\hat{p})$. We note that γ is monotone, preserves top and distributes over arbitrary meet, i.e. $\gamma(\sqcap \hat{Q}) = \sqcap \{ \gamma(\hat{q}) \in \hat{P} \mid \hat{q} \in \hat{Q} \}$. Similarly, α is monotone, preserves bottom and distributes over arbitrary join.

We say that $\lll \subseteq \hat{P} \times \hat{Q}$ is a *visible simulation relation* iff $\hat{p} \lll \hat{q}$ implies $\hat{o}(\hat{p}) \sqsubseteq \gamma(\hat{o}(\hat{q}))$ and $\hat{M}(\hat{i} \sqcap \hat{p}) \lll \hat{N}(\alpha(\hat{i}) \sqcap \hat{q})$ for all inputs $\hat{i} \in \hat{i}(\hat{P})$.

We say the abstract predicate transformer \hat{M} refines \hat{N} , denote $\hat{M} \leq_{\text{lat}} \hat{N}$, iff $(\top \in \hat{P}) \lll (\top \in \hat{Q})$ and \lll is a visible simulation relation between \hat{P} and \hat{Q} .

Theorem 3. *If \hat{M} and \hat{N} are abstract predicate transformers such that $\hat{M} \leq_{\text{lat}} \hat{N}$, and \hat{A}_{vis} is a visible trajectory assertion for \hat{N} , then:*

$$\hat{N} \models_{\text{lat}} \hat{A}_{\text{vis}} \Rightarrow \hat{M} \models_{\text{lat}} \alpha(\hat{A}_{\text{vis}})$$

Theorem 4. *If \hat{M} and \hat{N} are abstract predicate transformers of the circuit models M and N , then:*

$$\hat{M} \leq_{\text{lat}} \hat{N} \Rightarrow M \leq_{\text{set}} N$$

A Appendices

A.1 Theorem 1

Text.

References

1. Chou, C.T.: The mathematical foundation of symbolic trajectory evaluation. In: International Conference on Computer Aided Verification. pp. 196–207. Springer (1999)
2. Cousot, P.: Abstract interpretation. ACM Computing Surveys (CSUR) **28**(2), 324–328 (1996)
3. Davey, B.A., Priestley, H.A.: Introduction to lattices and order. Cambridge university press (2002)
4. Muchnick, S., et al.: Advanced compiler design implementation. Morgan kaufmann (1997)
5. Seger, C.J.H., Bryant, R.E.: Formal verification by symbolic evaluation of partially-ordered trajectories. Formal Methods in System Design **6**(2), 147–189 (1995)