Refinement for Symbolic Trajectory Evaluation

Authors

Chalmers

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

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: $n\,\mathbf{is}\,1$ and $n\,\mathbf{is}\,0$, 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 $(\mathit{in_1}\,\mathbf{is}\,1 \wedge \mathit{in_2}\,\mathbf{is}\,1) \Rightarrow \mathbf{N}(\mathit{out}\,\mathbf{is}\,1)$ 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 Lattice-theoretic STE

Circuit Models 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.

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 step[1]. The power set of C, denoted by $\mathcal{P}(C)$, can be viewed as a the set of *predicates* on configurations, where $\cap$, $\cup$, and $\subseteq$ correspond to conjunction, disjunction and implication, respectively. Furthermore, for any $Q \subseteq \mathcal{P}(C)$, we denote by $\cap Q$ and $\cup Q$ the intersection and union of all members of Q.

M_R induces a *predicate transformer* $M_F \in \mathcal{P}(C) \to \mathcal{P}(C)$ 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 $M_F(p)$, or in other words, $(c, c') \in M_R \Leftrightarrow c' \in M_F(\{c\})$. We adopt this functional model of M and drop its subscript.

Since the value of input signals is controlled by the external environment, a circuits model itself does not impose any constraint on them. Signals that do not correspond to inputs are further divided into state-holding signals and outputs. For such non-input signals, the model is a montonic function where imposed constraints are determined by the circuit topology and functionality. For instance, if assume the circuit is a two-input AND gate followed by an inverter, its model have the type $M \in \mathcal{P}(\mathbb{B}^4) \to \mathcal{P}(\mathbb{B}^4)$ and can be defined as:

$$M(p) = \{\langle b_1, b_2, i_1 \wedge i_2, \neg s\rangle \in \mathbb{B}^4 \mid \langle i_1, i_2, s, o\rangle \in \cap p\}$$

Here i_1 and i_2 refer to the two inputs of the AND gate, s to the internal inverter, and o is the ignored output; b_1 and b_2 are unconstrained in the new configuration.

Manipulating subsets of $\mathbb{B}^m$ is however 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 $1 \sqsubseteq X$[2]. For any $m \in \mathbb{N}$, this ordering on $\mathbb{T}$ is lifted component-wise to $\mathbb{T}^m$.

Ternary lattices 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.

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

[1] Mention how this affects circuits with zero-delays?

[2] We use the reverse ordering of what is originally used in STE.

element $\bot$ is therefore introduced, such that $\bot \sqsubseteq t$ and $\bot \neq t$ for all $t \in \mathbb{T}^m$. The extended $\mathbb{T}^m_\bot = \mathbb{T}^m \cup \{\bot\}$ then becomes a complete lattice. We denote its top element $\langle X, \ldots, X \rangle$ by $\top$.

Let there also be Galois connection $\ll \in \subseteq \mathcal{P}(C) \times \hat{P}$, relating concrete and abstract predicates such that for all $Q \subseteq \mathcal{P}(C)$ and $\hat{Q} \subseteq \hat{P}$:

$$Q \ll \hat{Q} \Leftrightarrow \cup Q \ll \sqcap \hat{Q}$$

where $Q \ll \hat{Q} \Leftrightarrow \forall p \in Q : \forall \hat{p} \in \hat{Q} : p \ll \hat{p}$. Intuitively, $p \ll \hat{p}$ states that p can be "approximated" as $\hat{p}$, and we note that $\ll$ is an extension of the partial orders of $\mathcal{P}(C)$ and $\hat{P}$ to an ordering between $\mathcal{P}(C)$ and $\hat{P}$.

It is sometimes convenient to define a Galois connection 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, from which one can derive $\ll$ as follows: $p \ll \hat{p} \Leftrightarrow p \subseteq \gamma(\hat{p})$ or $p \ll \hat{p} \Leftrightarrow \alpha(p) \sqsubseteq \hat{p}$. For example, a Galois connection from $\mathcal{P}(\mathbb{B}^m)$ to $\mathbb{T}^m_\bot$ for any $m \in \mathbb{N}$ can be defined in a natural way by specifying its concretisation function $\Gamma \in \mathbb{T}^m_\bot \rightarrow \mathcal{P}(\mathbb{B}^m)$:

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

Listing each boolean sequence approximated by the given ternary sequence. For example, $\langle 1, X, 0 \rangle$ abstracts both the boolean sequence $\langle 1, 1, 0 \rangle$ and $\langle 1, 0, 0 \rangle$ since they all agree on their first and third element.

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 it preserves $\bot$, $\hat{M}(\bot) = \bot$; is monotonic, $\hat{p} \sqsubseteq \hat{q} \Rightarrow \hat{M}(\hat{p}) \sqsubseteq \hat{M}(\hat{q})$ for all $\hat{p}, \hat{q} \in \hat{P}$; and $\ll$ is a *simulation relation* from $\mathcal{P}(C)$ to $\hat{P}$, $p \ll \hat{p} \Rightarrow M(p) \ll \hat{M}(\hat{p})$ for all $p \in \mathcal{P}(C)$ and $\hat{p} \in \hat{P}$.

Example of how $\hat{M}$ and M relate in practice.

Assertions and satisfaction A *trajectory assertion* for $\hat{M}$ is a quintuple $\hat{A} = (S, s_0, R, \pi_a, \pi_c)$, 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*, $\pi_a \in S \rightarrow \hat{P}$ and $\pi_c \in S \rightarrow \hat{P}$ label each state s with an *antecedent* $\pi_a(s)$ and a *consequent* $\pi_c(s)$. Furthermore, we assume that $(s, s_0) \notin S$ for all $s \in S$ without any loss of generality.

Define $F \in S \rightarrow (\hat{P} \rightarrow \hat{P})$ and $\mathcal{F} \in (S \rightarrow \hat{P}) \rightarrow (S \rightarrow \hat{P})$ as follows:

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

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

for all $\Phi \in S \rightarrow \hat{P}$ and $s \in S$. F preserves $\bot$ and both F and $\mathcal{F}$ are monotonic, where two $\Phi, \Phi' \in S \rightarrow \hat{P}$ are ordered as $\Phi \sqsubseteq \Phi' \Leftrightarrow \forall s \in S : \Phi(s) \sqsubseteq \Phi'(s)$. Let $\Phi_* \in S \rightarrow \hat{P}$ be the least fixpoint of the equation $\Phi = \mathcal{F}(\Phi)$ [3]. Since both S and $\hat{P}$ are finite, Φ_* is given by $\lim \Phi_n(s)$ where Φ_n is defined as follows:

$$\Phi_n = \textbf{if } (n = 0) \textbf{ then } (\lambda s \in S : \bot) \textbf{ else } \mathcal{F}(\Phi_{n-1}) \tag{3}$$

We say that the abstract circuit $\hat{M}$ *satisfies* a lattice-based, abstract trajectory assertion $\hat{A}$, denoted by $\hat{M} \models \hat{A}$, iff:

$$\forall s \in S : \Phi_*(s) \sqcap \pi_\alpha(s) \sqsubseteq \pi_c(s) \tag{4}$$

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

2 System refinement (WIP)

Consider another fixed, but arbitrary, circuit N such that configurations of M and N have the same externally visible elements but can differ internally. Let $\hat{N} \in \hat{Q} \to \hat{Q}$ be an abstract predicate transformer of N, we then we say that $\hat{M}$ *refines* $\hat{N}$ if every *externally visible behaviour* allowed by $\hat{M}$ is also allowed by $\hat{N}$, regardless of any initial configurations.

M always sets it $\hat{I}$ to $\top$ and doesn't really work for zero-delay circuits.

The value domain for inputs and outputs must be the same so they are comparable. Not just that they have the "same" inputs and output elements.

Example?

Drivers Let the *externally visible* parts of an abstract predicate $\hat{P}$ be the subsets given by two projections, i and o, identifying the "inputs" and "outputs" of $\hat{P}$, respectively. Further, let $[\![\cdot]\!] \in \hat{P} \to \hat{O}$ be a mapping that takes each $\hat{p} \in \hat{P}$ to its visible outputs $[\![\hat{p}]\!] \in \hat{O}$; $[\![\cdot]\!]$ is extended to sequences component-wise. With a slight abuse of notation, we overload both projections and the mapping to also accept predicates from $\hat{Q}$ and note that $i(\hat{P}) = i(\hat{Q}) = \hat{I}$ and $o(\hat{P}) = o(\hat{Q}) = \hat{O}$ since inputs and outputs in M and N have the same size and value domain.

A *driver* of $\hat{M}$ and $\hat{N}$ is a nonempty sequence of inputs, $\delta \in \hat{I}^+$, and induces a trajectory τ in $\hat{M}$ (resp. $\hat{N}$) where $\tau[0] = \top$ and $\forall i \in \mathbb{N} : 0 < i < |\delta + 1| \Rightarrow \tau[j] = \hat{M}(\delta[j-1] \sqcap \tau[j-1])$; the trajectory induced by a driver δ in $\hat{M}$ is denoted by $\mathit{Traj}(\hat{M})(\delta)$. Intuitively, if $\hat{M}$ produces the same, or at least more specified, outputs than $\hat{N}$ for all possible drivers, then every visible behaviour of $\hat{M}$ is covered by $\hat{N}$. We thus say that $\hat{M}$ refines $\hat{N}$, denoted by $\hat{M} \leq \hat{N}$, iff:

Footnote that "Traj" comes from trajectories in prev. papers? We have to use $=$ for ours since $\sqsubseteq$ allows one to pick bad states for unconstrained wires.

$$\forall \delta \in \hat{I}^+ : [\![\mathit{Traj}(\hat{M})(\delta)]\!] \sqsubseteq [\![\mathit{Traj}(\hat{N})(\delta)]\!]$$

Example!

Simulation Let $\preceq \in \hat{P} \times \hat{Q}$ be a simulation relation such that $\hat{p} \preceq \hat{q}$ implies (1) $[\![\hat{p}]\!] \sqsubseteq [\![\hat{q}]\!]$ and (2) $\hat{M}(\hat{\imath} \sqcap \hat{p}) \preceq \hat{N}(\hat{\imath} \sqcap \hat{q})$ for all inputs $\hat{\imath} \in \hat{I}$. We extend this relation to $\hat{M}$ and $\hat{N}$ such that $\hat{M} \preceq \hat{N}$ iff their top elements are related, $(\top \in \hat{P}) \preceq (\top \in \hat{Q})$. We then simplify refinement thus: $\hat{M} \leq \hat{N} \Leftrightarrow \hat{M} \preceq \hat{N}$.

Simplify? What?! Reference proof in appendix. Also, is the meaning of $(\top \in \hat{P}) \preceq (\top \in \hat{Q})$ clear?

Example!

Trajectory A trajectory assertion $\hat{A} = (S, s_0, R, \pi_a, \pi_c)$ for $\hat{N}$ where antecedents only mention inputs, $\pi_a \in S \to \hat{I}$, and consequents only mention outputs, $\pi_c \in S \to \hat{O}$, is referred to as an *external trajectory assertion*. Intuitively, a satisfied external trajectory assertion is property of N that must hold regardless of its internal state.

Let $G \in S \to (\hat{Q} \to \hat{Q})$ and $\mathcal{G} \in (S \to \hat{Q}) \to (S \to \hat{Q})$ be the duals of F and $\mathcal{F}$ in $\hat{N}$, respectively. Further, let Ψ_* be the least fix point of $\Psi = \mathcal{G}(\Psi)$ and the dual of Φ_* in $\hat{N}$. Fix for red below.

For all below, assume $\hat{M} \preceq \hat{N}$.

(1): $\hat{p} \preceq \hat{q} \Rightarrow \forall s \in S : F(s)(\hat{p}) \preceq G(s)(\hat{q})$. $F(s)(\hat{p}) = \hat{M}(\pi_a(s) \sqcap \hat{p})$ and $G(s)(\hat{q}) = \hat{N}(\pi_a(s) \sqcap \hat{q})$ where $\pi_a(s) \in \hat{I}$. By prop. 2 of $\hat{p} \preceq \hat{q}$, we thus know that $F(s)(\hat{p}) \preceq G(s)(\hat{q})$.

(2): $\Phi_*(s) \preceq \Psi_*(s)$ for all $s \in S$. By induction on n, like below. Base case and case when $s = s_0$ are both trivial. For the inductive step, assume that $\Phi_n(s) \preceq \Psi_n(s)$ for all $s \in S$. For any $s \neq s_0$, we have that $\Phi_{n+1} = \sqcup\{F(s')(\Phi_n(s')) \mid (s', s) \in R\}$. By (1) we know that ...

If $\hat{M} \preceq \hat{N}$, we claim that $[\![\Phi_*(s)]\!] \sqsubseteq [\![\Psi_*(s)]\!]$ for all $s \in S$. Since $\Phi_*(s) = \lim \Phi_n(s)$ and $\Psi_*(s) = \lim \Psi_n(s)$, it suffices to prove that $[\![\Phi_n(s)]\!] \sqsubseteq [\![\Psi_n(s)]\!]$ for all $s \in S$ and $n \in \mathbb{N}$. We do so by induction on n. The base case is trivially true ($[\![\bot]\!] \sqsubseteq [\![\bot]\!]$). For the inductive step, assume $[\![\Phi_n(s)]\!] \sqsubseteq [\![\Psi_n(s)]\!]$ for all $s \in S$. That $[\![\Phi_{n+1}(s)]\!] \sqsubseteq [\![\Psi_{n+1}(s)]\!]$ for s_0 is also trivially correct ($[\![\top]\!] \sqsubseteq [\![\top]\!]$). For any $s \neq s_0 \in S$, we have $[\![\Phi_{n+1}(s)]\!] = [\![\sqcup\{F(s')(\Phi_n(s')) \mid (s', s) \in R\}]\!] \sqsubseteq [\![\sqcup\{F(s')(\Psi_n(s')) \mid (s', s) \in R\}]\!]$. I would like to claim that this is $\sqsubseteq [\![\sqcup\{G(s')(\Psi_n(s')) \mid (s', s) \in R\}]\!] = [\![\Psi_{n+1}]\!]$. But I'm missing a relation between F and G, which should come from $\preceq$.

Theorem 1. *Given two abstract predicate transformers $\hat{M}$ and $\hat{N}$ such that $\hat{M} \preceq \hat{N}$, and a external trajectory assertion $\hat{A}$ for $\hat{N}$, then $\hat{N} \models \hat{A} \Rightarrow \hat{M} \models \hat{A}$.*

A Proofs (WIP)

A.1 $\hat{M} \preceq \hat{N} \Leftrightarrow \hat{M} \leq \hat{N}$

We prove the two directions of $\hat{M} \preceq \hat{N} \Leftrightarrow \hat{M} \leq \hat{N}$ separately.

$\hat{M} \preceq \hat{N} \Rightarrow \hat{M} \leq \hat{N}$: For any $\delta \in \hat{I}^+$, let $\tau \in \hat{P}^+$ and $v \in \hat{Q}^+$ be the induced trajectories $\mathit{Traj}(\hat{M})(\delta)$ and $\mathit{Traj}(\hat{N})(\delta)$, respectively. We first prove that $\tau[k] \preceq v[k]$ for each $\delta \in \hat{I}^+$ and $k \in \mathbb{N} : k < |\tau| = |v| = |\delta + 1|$ by induction on k. The base case, $\tau[0] = (\top \in \hat{P}) \preceq (\top \in \hat{Q}) = v[0]$, follows immediately from the definition of $\hat{M} \preceq \hat{N}$. For the inductive step, assume that $\tau[k] \preceq v[k]$. That also $\tau[k+1] = \hat{M}(\delta[k] \sqcap \tau[k]) \preceq \hat{N}(\delta[k] \sqcap v[k]) = v[k+1]$ follows from the second property of $\tau[k] \preceq v[k]$ since $\delta[k] \in \hat{I}$. Finally, for any $\delta \in \hat{I}^+$, that $[\![\tau[k]]\!] \sqsubseteq [\![v[k]]\!]$ for each k follows from the first property of $\tau[k] \preceq v[k]$.

$\hat{M} \preceq \hat{N} \Leftarrow \hat{M} \leq \hat{N}$: Define $\preceq \in \hat{P} \times \hat{Q}$ as follows:

$$\bigcup_{\delta \in I^+} \{(\mathit{Traj}(\hat{M})(\delta)[k], \mathit{Traj}(\hat{N})(\delta)[k]) \mid k \in \mathbb{N}, k < |\delta + 1|\}$$

Here $\mathit{Traj}(\hat{M})(\delta)[k]$ is the k-th predicate of the trajectory induced by δ. By definition of $\preceq$, and that $\hat{M} \leq \hat{N}$, we thus have that any such pair of k-th predicates must have ordered outputs. That is, for any $\hat{p} \preceq \hat{q}$, we have that $[\![\hat{p}]\!] \sqsubseteq [\![\hat{q}]\!]$. Given that $\hat{p} = \mathit{Traj}(\hat{M})(\delta)[k]$ and $\mathit{Traj}(\hat{N})(\delta)[k]$ for some common k and $\delta \in \hat{I}^+$, we must also have that $\delta \frown \hat{\imath}$ (δ followed by $\hat{\imath}$) is in $\hat{I}^+$ and thus $\hat{M}(\hat{\imath} \sqcap \hat{p}) = \mathit{Traj}(\hat{M})(\delta \frown \hat{\imath})[k+1] \preceq \mathit{Traj}(\hat{N})(\delta \frown \hat{\imath})[k+1] = \hat{N}(\hat{\imath} \sqcap \hat{q})$ for all $\hat{\imath} \in \hat{I}$. Finally, that $\hat{M} \preceq \hat{N}$ follows from how $[\![\top \in \hat{P}]\!] \sqsubseteq [\![\top \in \hat{Q}]\!]$ is obviously true and that $\langle \hat{\imath} \rangle \in \hat{I}^+$ for all $\hat{\imath} \in \hat{I}$.

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)