Refinement for Symbolic Trajectory Evaluation

Authors

Chalmers

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

Keywords: STE · Refinement · ?

1 Symbolic Trajectory Evaluation

Symbolic trajectory evaluation [3] (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 $(in_1\ \mathbf{is}\,1 \wedge in_2\ \mathbf{is}\,1) \Rightarrow \mathbf{N}(out\ \mathbf{is}\,1)$ 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 [2]. ...

2 Set-theoretic STE

Consider an arbitrary, but fixed, digital circuit M operating in discrete time. A *configuration* of M, denoted by $\mathbb{C}$, 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*, $M_R \subseteq \mathbb{C} \times \mathbb{C}$, where $(c, c') \in M_R$ means that M can move from c to c' in one step[1]. The power set of $\mathbb{C}$, denoted by $\wp(\mathbb{C})$, can be viewed as a the set of *predicates* on configurations, where $\cap$, $\cup$, and $\subseteq$ correspond to conjunction, disjunction and implication, respectively. We denote by $\cap S$ and $\cup S$ the intersection and union of all members of any $S \subseteq \wp(\mathbb{C})$.

M_R induces a *predicate transformer* $M_F \in \wp(\mathbb{C}) \to \wp(\mathbb{C})$ using the relational image operation:

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

It is intuitively obvious that if M is in one of the configurations in $C \in \wp(\mathbb{C})$, 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:

$$M_F(\cup S) = \cup\{M_F(C) \mid C \in S\}$$

for all $S \subseteq \wp(\mathbb{C})$. 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 $\mathbb{C}$ and its signals are, is not important in this section. In practice, however, signals are typically divided into external, such as "inputs" and "outputs", 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 $M \in \wp(\mathbb{B}^3) \to \wp(\mathbb{B}^3)$ as follows:

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

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 $A = (S, s_0, R, \pi_\mathrm{a}, \pi_\mathrm{c})$, 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*, $\pi_\mathrm{a} \in S \to \wp(\mathbb{C})$ and $\pi_\mathrm{c} \in S \to \wp(\mathbb{C})$ label each state s with an *antecedent* $\pi_\mathrm{a}(s)$ and a *consequent* $\pi_\mathrm{c}(s)$. We assume that $(s, s_0) \notin S$ 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 specific, a *trajectory* of M is a non-empty sequences of configurations, $\tau \in \mathbb{C}^+$, 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?

A τ satisfies the antecedents of ρ, denoted by $\tau \models_{\mathrm{a}} \rho$, iff $\tau_n \in \pi_{\mathrm{a}}(\rho_n)$ for all $n \in \mathbb{N} : n < |\tau| = |\rho|$; satisfaction of consequents is defined similarly with π_{c} and denoted by $\tau \models_{\mathrm{c}} \rho$.

That M satisfies A, denoted by $M \models A$, can then be formalized as follows:

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

where $\mathit{Traj}(M)$ and $\mathit{Runs}(A)$ denote the sets of all trajectories of M and runs of A, respectively. This satisfaction can be formulated equivalently as a problem for deterministic finite automaton.

2.1 Refinement

Consider another fixed, but arbitrary, circuit model $N \in \wp(\mathbb{D}) \to \wp(\mathbb{D})$, where $\mathbb{D}$ is a non-empty and finite set of configurations. Exactly what configurations such as $\mathbb{C}$ and $\mathbb{D}$ are, were not important previously. But to reason about refinement, which relates the external behaviour of circuits, we make a distinction between their elements. Let $\sim \subseteq \mathbb{C} \times \mathbb{C}$ be an equivalence relation on $\mathbb{C}$. The equivalence class of a $c \in \mathbb{C}$ under $\sim$, denoted by $[c]$, is defined as $[c] = \{c' \in \mathbb{C} | c' \sim c\}$. With a slight abuse of notation, we overload both $\sim$ and $[\cdot]$ to accept configurations in $\mathbb{D}$. We also extend $[\cdot]$ to sets $C \subseteq \mathbb{C}$ as $[C] = \cup\{[c] \in \wp(\mathbb{C}) \mid c \in C\}$.

Refinement by trajectories Let there be a Galois connection between predicates $\wp(\mathbb{C})$ and $\wp(\mathbb{D})$ ordered by set inclusion. The usual definition of a Galois connection is in terms of an *abstraction* $\alpha \in \wp(\mathbb{C}) \to \wp(\mathbb{D})$ and a *concretisation* $\gamma \in \wp(\mathbb{D}) \to \wp(\mathbb{C})$ function, such that $\alpha(C) \subseteq D \Leftrightarrow C \subseteq \gamma(D)$ for all $C \in \wp(\mathbb{C})$ and $D \in \wp(\mathbb{D})$. For example, a Galois connection between ...

Furthermore, let the binary relation $\ll \subseteq \wp(\mathbb{C}) \times \wp(\mathbb{D})$, where $C \ll D$ reads "C can be approximated by D", be derived from the above α or γ, such that:

$$C \ll D \Leftrightarrow \alpha([C]) \subseteq [D] \qquad C \ll D \Leftrightarrow [C] \subseteq \gamma([D])$$

Here $\subseteq$ on the α-derivation side is the inclusion order of $\wp(\mathbb{D})$, and on the γ-derivation side $\subseteq$ is the inclusion order of $\wp(\mathbb{C})$. Intuitively, $\ll$ acts as an extension of the orderings inside $\wp(\mathbb{C})$ and $\wp(\mathbb{D})$ to one between equivalence classes of them. We require that $\alpha([c]) \neq \emptyset$ for all $c \in \mathbb{C}$. We extend $\ll$ to sequences component wise, such that $\tau \ll \upsilon$ iff $\{\tau_n\} \ll \{\upsilon_n\}$ for all $\tau \in \mathbb{C}^+$, $\upsilon \in \mathbb{D}^+$, and $n \in \mathbb{N} : n < |\tau| = |\upsilon|$.

We can now formalize that M *refines* N, denoted by $M \leq N$, as follows:

$$\forall \tau \in \mathit{Traj}(M) : \exists \upsilon \in \mathit{Traj}(N) : |\tau| = |\upsilon| \wedge \tau \ll \upsilon$$

In other words, for every sequence of configurations τ permitted by M, there must exist a sequence υ for N which approximates the behaviour of τ according to their equivalence relations. Example.

Refinement & assertions Recall that a trajectory assertion for N is a quintuple $A = (S, s_0, R, \pi_\mathrm{a}, \pi_\mathrm{c})$, where $\pi_\mathrm{a} \in S \to \wp(\mathbb{D})$ and $\pi_\mathrm{c} \in S \to \wp(\mathbb{D})$ label each $s \in S$ with its antecedents and consequents, respectively. If π_a and π_c accept equivalence classes in $\mathbb{D}/\sim$, i.e. $d \in \pi_\mathrm{a}(s) \Leftrightarrow [d] \subseteq \pi_\mathrm{a}(s)$ for all $s \in S$ and similarly for π_c, then we refer to A as an *name trajectory assertion* and suffix it as A_n. Furthermore, we define $\gamma(A) = (S, s_0, R, \gamma(\pi_\mathrm{a}), \gamma(\pi_\mathrm{c}))$, where $\gamma(\pi_\mathrm{a}) = \lambda s \in S : \gamma(\pi_\mathrm{a}(s))$ and $\gamma(\pi_\mathrm{c}) = \lambda s \in S : \gamma(\pi_\mathrm{c}(s))$.

We are now ready to state that, if M refines N, then a concretisation of every name trajectory assertion satisfied in N can also be satisfied in M:

Theorem 1. $M \leq N \Rightarrow (N \models A_n \Rightarrow M \models \gamma(A_n))$

Text.

Refinement by simulation Refinement can be equivalently formulated as $\ll$ being a simulation relation. More specifically, we say that M refines N by *set-theoretic name refinement*, denoted by $M \leq_\mathrm{set} N$, iff (1) $\ll$ is a name, i.e. $C \ll D \Rightarrow \forall c \in C : \exists d \in D : \{c\} \ll \{d\}$; and (2) $\ll$ is a *simulation relation* from $\wp(\mathbb{C})$ to $\wp(\mathbb{D})$, i.e. $C \ll D \Rightarrow M(C) \ll N(D)$. That $\ll$ is a simulation relation can also be stated directly in terms of the usual abstraction function: $\alpha([M(C)]) \subseteq N(\alpha([C]))$ for all $C \in \wp(\mathbb{C})$, or the concretisation function: $M(\gamma([D]) \subseteq \gamma(N([D]))$ for all $D \in \wp(\mathbb{D})$.

Theorem 2. $M \leq N \Leftrightarrow M \leq_{set} N$

A Appendices

A.1 Theorem 1

We first prove a few lemmas.

Lemma 1. $[d] \cap \pi_a(\rho) \neq \emptyset \Rightarrow [d] \subseteq \pi_a(\rho)$

Since they intersect, there must exist $d' \in [d]$ such that $d' \in \pi_\mathrm{a}(\rho)$. By property name of $\pi_\mathrm{a}(\rho)$, it must be that $[d'] = [d] \subseteq \pi_\mathrm{a}(\rho)$. □

Lemma 2. $d \in \pi_c(\rho) \wedge \{c\} \ll \{d\} \Rightarrow c \in \gamma(\pi_c(\rho))$

By property name of π_c, $d \in \pi_\mathrm{c}(\rho) \Rightarrow [d] \subseteq \pi_\mathrm{c}(\rho)$ which, by the monotonicity of γ, implies $\gamma([d]) \subseteq \gamma(\pi_\mathrm{c}(\rho))$. By definition of $\{c\} \ll \{d\}$, we know $[c] \subseteq \gamma([d])$, thus $[c] \subseteq \gamma(\pi_\mathrm{c}(\rho))$. □

Lemma 3. $c \in \gamma(\pi_a(\rho)) \wedge \{c\} \ll \{d\} \Rightarrow d \in \pi_a(\rho)$

By property name of π_a and the definition of $\gamma(\pi_\mathrm{a})$, $c \in \gamma(\pi_\mathrm{a}(\rho))$ states that $[c] \subseteq \gamma(\pi_\mathrm{a}(\rho))$. And thus $\alpha([c]) \subseteq \alpha(\gamma(\pi_\mathrm{a}(\rho)) \subseteq \pi_\mathrm{a}(\rho)$ by the monotonicity of α. By definition $\{c\} \ll \{d\}$, we also have that $\alpha([c]) \subseteq [d]$. Since $\alpha([c]) \neq \emptyset$, it must be that $[d] \cap \pi_\mathrm{a}(\rho) \neq \emptyset$, and thus $[d] \subseteq \pi_\mathrm{a}(\rho)$ by lemma 1. Then, by property

name of π_a, we must have that $d \in \pi_a(\rho)$. □

For the theorem, we are given $\tau \in \mathit{Traj}(M)$ and $\rho \in \mathit{Runs}(\gamma(A))$, such that $|\tau| = |\rho|$ and $\tau_n \in \gamma(\pi_a(\rho_n))$ for all $n \in \mathbb{N} : n < |\tau|$. We must then show that $\tau_n \in \gamma(\pi_c(\rho_n))$. By the refinement assumption, there must exist a $\upsilon \in \mathit{Traj}(N)$ such that $|\tau| = |\upsilon| = |\rho|$ and $\{\tau_n\} \ll \{\upsilon_n\}$. By lemma 3, we know $\upsilon_n \in \pi_a(\rho_n)$ and thus $\upsilon_n \in \pi_c(\rho_n)$. Lemma 2 then states that $\tau_n \in \gamma(\pi_c(\rho_n))$. □

A.2 Theorem 2

Lemma 4. $C \ll D \wedge D \subseteq D' \Rightarrow C \ll D'$

That C is approximated by D' follows immediately: $\alpha(C) \subseteq D \subseteq D'$. The first property of $\ll$ follows from the definition of subset, and the second by the monotonicity of N: $\alpha(M(C)) \subseteq N(D) \subseteq N(D')$. □

We prove each direction of the theorem separately.

($\Rightarrow$) : If $C \ll D$, then by definition $\alpha([C]) \subseteq [D]$. As α distributes over arbitrary union, it follows that $\alpha([c]) \subseteq [D]$ for all $c \in C$. Furthermore, applying the assumption $M \leq N$ to one-length trajectories starting in c, there must also exist $d \in \mathbb{D}$ such that $\{c\} \ll \{d\}$, or $\alpha([c]) \subseteq [d]$ by definition of $\ll$. Since $\alpha([c]) \neq \emptyset$, it must be that $[d] \cap [D] \neq \emptyset$. By lemma 1 then, we know $[d] \subseteq [D]$ and thus $d \in D$. This shows that the first property of $\ll$ is implied. For the second property, that $\ll$ is a simulation relation, consider all two-length trajectories from C. For any $c' \in M(C)$, there exists a $d' \in N(D)$ such that $\{c'\} \ll \{d'\}$, which implies that $\{c'\} \ll N(D)$ by lemma 4. Combining all such orderings, we have the desired $M(C) \ll N(D)$. □

($\Leftarrow$) : We show this claim by induction on the length of τ. For the base case, when $|\tau| = 1$, we are given $\tau = \langle \tau_1 \rangle$ where τ_1 is unconstrained, i.e. we only know that $\tau_1 \in \mathbb{C}$. But a Galois connection always relates the most general states of $\mathbb{C}$ and $\mathbb{D}$, so $\alpha(\mathbb{C}) \subseteq \mathbb{D}$. As $[\mathbb{C}] = \mathbb{C}$ and $[\mathbb{D}] = \mathbb{D}$, we also know that $\alpha([\mathbb{C}]) \subseteq [\mathbb{D}]$, or $\mathbb{C} \ll \mathbb{D}$. Using the first property of $\ll$ then tells us that there exists $d \in \mathbb{D}$ such that $\{\tau_1\} \ll \{d\}$. For the inductive step, when $|\tau| = n + 1$, we are given $\tau = \langle \ldots, \tau_n \rangle$ and assume there exists $\upsilon = \langle \ldots, \upsilon_n \rangle$ such that $\tau \ll \upsilon$. By the simulation property of $\ll$, we have that $M(\tau_n) \ll N(\upsilon_n)$. Applying the first property of $\ll$ then states that there exists $d \in N(\upsilon_n)$ such that $\{\tau_{n+1}\} \ll \{d\}$ for any $\tau_{n+1} \in M(\tau_n)$. The concatenation of υ and d, denoted $\upsilon^\frown\langle d \rangle$, thus forms a valid trajectory in $\mathit{Traj}(N)$ such that $|\tau| = |\upsilon^\frown\langle d \rangle|$ and $\tau \ll \upsilon^\frown\langle d \rangle$. □

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. Muchnick, S., et al.: Advanced compiler design implementation. Morgan kaufmann (1997)

3. 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)