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 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 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. 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* $M_F \in \mathcal{P}(C) \to \mathcal{P}(C)$ 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 $M_F(p)$. 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 $M \in \mathcal{P}(\mathbb{B}^3) \to \mathcal{P}(\mathbb{B}^3)$ 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*, $\pi_a \in S \to \mathcal{P}(C)$ and $\pi_c \in S \to \mathcal{P}(C)$ label each state s with an *antecedent* $\pi_a(s)$ and a *consequent* $\pi_c(s)$. We assume that $(s, s_0) \notin S$ 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 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?

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 and denoted by $\tau \models_c \rho$. That M satisfies A, denoted by $M \models A$, can then formalized[2]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.

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 $\mathbb{T} = \mathbb{B} \cup \{\mathrm{X}\}$ 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 $0 \sqsubseteq \mathrm{X}$ and $1 \sqsubseteq \mathrm{X}$[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 $0 \sqsubseteq \mathrm{X}$ and $1 \sqsubseteq \mathrm{X}$ 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 $\mathbb{T}^m_\perp = \mathbb{T}^m \cup \{\perp\}$ then becomes a complete lattice. We denote the top element $\langle \mathrm{X}, \ldots, \mathrm{X} \rangle$ of $\mathbb{T}^m_\perp$ 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* $\alpha \in \mathcal{P}(C) \to \hat{P}$ and a *concretisation* $\gamma \in \hat{P} \to \mathcal{P}(C)$ 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 $\mathbb{T}^m_\perp$ for any $m \in \mathbb{N}$ can be defined in a natural way by its concretisation function $\gamma \in \mathbb{T}^m_\perp \to \mathcal{P}(\mathbb{B}^m)$:

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

[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 $\alpha \in \mathcal{P}(\mathbb{B}^m) \to \mathbb{T}_\bot^m$ instead finds the most precise abstract predicate for a set of concrete predicates:

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

Abstract circuit model An *abstract predicate transformer* $\hat{M} \in \hat{P} \to \hat{P}$ is an *abstract interpretation* [2] of $M \in \mathcal{P}(C) \to \mathcal{P}(C)$ iff: $\hat{M}$ preserves $\bot$, i.e. $\hat{M}(\bot) = \bot$; $\hat{M}$ 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 α, or γ, form a *simulation relation* between $\mathcal{P}(C)$ and $\hat{P}$, 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 $\hat{p} \in \hat{P}$.

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

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

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

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

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 $\hat{\pi}_a \in S \to \hat{P}$ and $\hat{\pi}_c \in S \to \hat{P}$ label each state s with an abstract predicate for its antecedent and consequent, respectively.

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

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

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

$\hat{F}$ preserves $\bot$, and both $\hat{F}$ and $\hat{\mathcal{F}}$ are monotonic; two $\hat{\Phi}, \hat{\Phi}' \in S \to \hat{P}$ are ordered as $\hat{\Phi} \sqsubseteq \hat{\Phi}' \Leftrightarrow \forall s \in S : \hat{\Phi}(s) \sqsubseteq \hat{\Phi}'(s)$. Let $\hat{\Phi}_* \in S \to \hat{P}$ be the least fixpoint of the equation $\hat{\Phi} = \hat{\mathcal{F}}(\hat{\Phi})$ [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:

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

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

2 Refinement

2.1 Set-Theoretic refinement

Consider another fixed, but arbitrary, circuit model $N \in \mathcal{P}(D) \to \mathcal{P}(D)$, where D is a non-empty and finite set of configurations. In order to relate D to the previous C, let there be a Galois connection $\ll \subseteq \mathcal{P}(C) \times \mathcal{P}(D)$ given by means of a binary relation between $\mathcal{P}(P)$ and $\mathcal{P}(Q)$ ordered by set inclusion $\subseteq$. The relation $\ll$ acts as an extension of the orderings inside $\mathcal{P}(C)$ and $\mathcal{P}(D)$ to one between them, such that configurations in any $P \subseteq \mathcal{P}(C)$ and $Q \subseteq \mathcal{P}(D)$ are related iff their union and intersection are related:

$$P \ll Q \Leftrightarrow \forall p \in P : \forall q \in Q : p \ll q \Leftrightarrow \cup P \ll \cap Q$$

where $p \ll q$ reads as "p approximates q". The usual functions for abstraction $\alpha \in \mathcal{P}(C) \to \mathcal{P}(D)$ and concretisation $\gamma \in \mathcal{P}(D) \to \mathcal{P}(C)$ can be derived from $\ll$ as follows: $\alpha(c) = \cap\{d \in \mathcal{P}(D) \mid c \ll d\}$ and $\gamma(d) = \cup\{c \in \mathcal{P}(C) \mid c \ll d\}$. Conversely, the relation $\ll$ can be derived from α or γ as: $c \ll d \Leftrightarrow \alpha(c) \subseteq d$ or $c \ll d \Leftrightarrow c \subseteq \gamma(d)$. Note properties of α and γ?

Exactly what configurations such as C and D are, were not important previously. To reason about refinement, which relates the visible behaviour of circuits, we make a distinction between their external and internal elements. Let the visible elements of a configuration in C be identified by two predicates, $i \in C \to \mathcal{P}(C)$ and $\mathrm{o} \in C \to \mathcal{P}(C)$, where $i(c)$ denotes the configurations with *inputs* equal to a $c \in C$, and $\mathrm{o}(c)$ denotes configurations with equal *outputs*. Further, let $i(\cdot)$ and $\mathrm{o}(\cdot)$ induce equivalence relations $\sim_i$ and $\sim_\mathrm{o}$ on configurations in C, respectively, such that $c \sim_i c' \Leftrightarrow i(c) = i(c')$ and $c \sim_\mathrm{o} c' \Leftrightarrow \mathrm{o}(c) = \mathrm{o}(c')$. With a slight abuse of notation, we overload $i(\cdot)$ and $\mathrm{o}(\cdot)$ to accept configurations in D as well.

We now formalize an intuition of whether M *refines* N, denoted by $M \leq N$:

$$\forall \tau \in \mathit{Traj}(M) : \exists \upsilon \in \mathit{Traj}(N) : |\tau| = |\upsilon| \wedge \tau \ll_i \upsilon \wedge \tau \ll_\mathrm{o} \upsilon$$

where $\langle \tau_0, \ldots \rangle \ll_i \langle \upsilon_0, \ldots \rangle$ iff $i(\tau_n) \ll i(\upsilon_n)$ for all $n \in \mathbb{N} : n < |\tau| = |\upsilon|$; $\tau \ll_\mathrm{o} \upsilon$ is similarly defined by $\mathrm{o}(\cdot)$. Intuitively, $M \leq N$ states that, for every sequence of configurations τ permitted by M, there must exist a sequence υ for N which approximates the input-output behaviour of τ. That is, N must "cover" the input-output behaviour of M.

[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].

While refinement talks about input behaviours for M and N, we should recall that models generally cannot control their inputs, and leave such signals unconstrained for every transition. A trajectory $\tau \in \mathit{Traj}(M)$ is thus *driven* by an implicit choice of inputs. Let $\mathit{Traj}(M)(\delta)$ be the equivalence class of trajectories under a common *driver* $\delta \in i[C]^+$, defined as $\mathit{Traj}(M)(\delta) = \{\tau \in \mathit{Traj}(M) \mid \forall n \in \mathbb{N} : n < |\delta| \Rightarrow i(\tau_n) = \delta_n\}$.

$$\forall \delta \in i[C]^+ : \forall \tau \in \mathit{Traj}(M)(\delta) : \exists \upsilon \in \mathit{Traj}(N)(\delta) : \tau \ll_{\mathrm{o}} \upsilon$$

A Appendices

A.1 Theorem ??: $M \le N \Leftrightarrow M \le_{\mathbf{set}} N$

We first prove a few lemmas.

Lemma 1. $\tau \in \mathit{Traj}(M) \Leftrightarrow \tau \in \mathit{Traj}(M)(i(\tau))$

$$\mathit{Traj}(M)(i(\tau)) = \{\tau \in C^+ \mid \tau \in \mathit{Traj}(M) : i(\tau) = i(\tau)\} = \mathit{Traj}(M)$$

Lemma 2. $\bigcup_{\delta \in i[C]^+} \{\tau_0 \mid \tau \in \mathit{Traj}(M)(\delta)\} = C$

Input equality forms an equivalence class for trajectories, and the union of each class cover $\mathit{Traj}(M)$. Further, as the initial element of each trajectory is unconstrained, their union must contain every possible configuration.

Lemma 3. $\bigcup_{\delta \in i[C]^+} \{\bigcup_{\varepsilon \in \mathrm{o}[C]^+} \{\tau_0 \mid \tau \in \mathit{Traj}(M)(\delta) : \mathrm{o}(\tau) = \varepsilon\}\} = C$

Similar reasoning as above, but partitioned twice.

Lemma 4. $M \le N \Leftrightarrow M \le_{in} N$

We prove each direction separately:

($\Rightarrow$) : Given $\delta \in i[C]^+$ and $\tau \in \mathit{Traj}(M)(\delta)$, where $\delta = [\tau]_i$. By lemma 1, $\tau \in \mathit{Traj}(M)([\tau]_i)$ implies $\tau \in \mathit{Traj}(M)$. And by the assumption, there must exist $\upsilon \in \mathit{Traj}(N)$ such that $|\tau| = |\upsilon|$, $i(\tau) = i(\upsilon)$, and $\mathrm{o}(\tau) = \mathrm{o}(\upsilon)$. Using lemma 1 in reverse, $\upsilon \in \mathit{Traj}(N)$ implies that $\upsilon \in \mathit{Traj}(N)([\upsilon]_i) = \mathit{Traj}(N)(\delta)$.

($\Leftarrow$) : Given $\tau \in \mathit{Traj}(M)$, lemma 1 states that $\tau \in \mathit{Traj}(M)([\tau]_i)$. By the assumption, there must exist $\upsilon \in \mathit{Traj}(N)(i(\tau))$ such that $\mathrm{o}(\tau) = \mathrm{o}(\upsilon)$. By defintion of $\mathit{Traj}(N)(i(\tau))$, it easy to see that $i(\tau) = i(\upsilon)$ and $|\tau| = |\upsilon|$. Using lemma 1 in reverse, $\upsilon \in \mathit{Traj}(N)(i(\tau)) = \mathit{Traj}(N)(i(\upsilon))$ implies that $\upsilon \in \mathit{Traj}(N)$.

Lemma 5. $c_1 \ll d_1 \wedge c_2 \ll d_2 \Rightarrow (c_1 \cup c_2) \ll (d_1 \cup d_2)$

Text.

Lemma 6. $M \le_{in} N \Leftrightarrow M \le_{set} N$

We prove each direction of the theorem separately:

($\Rightarrow$) : We define the relation $\ll \subseteq \mathcal{P}(C) \times \mathcal{P}(D)$ as follows:

$$\mathrm{cl}_{\cup}\{(\{\tau_{|\delta|-1}\}, \{\upsilon_{|\delta|-1}\}) \mid \forall \delta \in i[C]^{+} : \forall \tau \in \mathit{Traj}(M)(\delta) : \forall \upsilon \in \mathit{Traj}(N)(\delta) : \mathrm{o}(\tau) = \mathrm{o}(\upsilon)\}$$

where $\tau_{|\delta|-1}$ and $\upsilon_{|\delta|-1}$ denote the last element of both sequences.

We show that $\ll$ is a simulation relation on the visible elements in $\mathcal{P}(C)$ and $\mathcal{P}(D)$. Firstly, for any $c \ll d$, both c and d are unions of a finite number of $\tau \in \mathit{Traj}(M)(\delta)_{|\delta|-1}$ and $\upsilon \in \mathit{Traj}(N)(\delta)_{|\delta|-1}$ pairs with some common δ. By definition of $\ll$, we know that $[\tau]_{\mathrm{o}} = [\upsilon]_{\mathrm{o}}$ for each such τ and υ, and thus $\mathrm{o}[c] \subseteq \mathrm{o}[d]$. Secondly, the concatenation of any such δ and each $i \in i[C]$ forms another driver $\delta \frown \langle i \rangle \in i[C]^{+}$. ...

Finally, to show that $M \leq_{\mathrm{set}} N$ we must show that $C \ll D$. By lemma 2 and the forall-quantification over drivers, we know that every member of C is considered. Further, as $\mathrm{o}[C] = \mathrm{o}[D]$ and every output of C is considered, every member of D must also be considered.

($\Leftarrow$) : Text.

Corollary 1. *$M \leq N \Leftrightarrow M \leq_{set} N$*

Follows immediately from lemma 4 and 6.

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)