Fully-abstract Statecharts Semantics via Intuitionistic Kripke Models

Gerald Lüttgen[1] and Michael Mendler[2]

[1] ICASE, Mail Stop 132C, NASA Langley Research Center, Hampton, Virginia 23681-2199, USA, luettgen@icase.edu
[2] Department of Computer Science, Sheffield University, 211 Portobello Street, Sheffield S1 5DP, U.K., M.Mendler@dcs.shef.ac.uk

Abstract. The semantics of Statecharts macro steps, as introduced by Pnueli and Shalev, lacks compositionality. This paper first analyzes the compositionality problem and traces it back to the invalidity of the Law of the Excluded Middle. It then characterizes the semantics via a particular class of linear, intuitionistic Kripke models, namely stabilization sequences. This yields, for the first time in the literature, a simple fully-abstract semantics which interprets Pnueli and Shalev's concept of failure naturally. The results not only give insights into the semantic subtleties of Statecharts, but also provide a basis for developing algebraic theories for macro steps and for comparing different Statecharts variants.

1 Introduction

Statecharts is a well-known design notation for specifying the behavior of embedded systems [6]. It extends *finite state machines* by concepts of *hierarchy* and *concurrency*. Semantically, a Statechart may respond to an event entering the system by engaging in an enabled transition. This may generate new events which, by *causality*, may in turn trigger additional transitions while disabling others. The *synchrony hypothesis* ensures that one execution step, a so-called *macro step*, is complete as soon as this chain reaction comes to a halt.

Pnueli and Shalev presented two equivalent formalizations of Statecharts' macro-step semantics in a seminal paper [16]. However, their semantics violates the desired property of *compositionality*. Huizing and Gerth [10] showed that combining compositionality, causality, and the synchrony hypothesis cannot be done within a simple, single-leveled semantics. Some researchers then devoted their attention to investigating new variants of Statecharts, obeying just two of the three properties. In *Esterel* [3] and *Argos* [15] causality is treated separately from compositionality and synchrony, while in (synchronous) *Statemate* [8] and *UML Statecharts* [7] the synchrony hypothesis is rejected. Other researchers achieved combining all three properties by storing semantic information via preorders [14, 17] or transition systems [5, 13]. However, no analysis of exactly how much information is needed to achieve compositionality has been made, yet.

This paper first illustrates the compositionality defect of Pnueli and Shalev's semantics by showing that equality of response behavior is not preserved by the concurrency and hierarchy operators of Statecharts (cf. Sec. 2). The reason is that macro steps abstract from causal interactions with a system's environment, thereby imposing a closed-world assumption. Indeed, the studied problem can be further traced back to the invalidity of the *Law of the Excluded Middle*. To overcome the problem, we interpret Statecharts, relative to a given system state, as intuitionistic formulas. These are given meaning as specific *intuitionistic Kripke structures* [18], namely linear increasing sequences of event sets, called *stabilization sequences*, which encode interactions between Statecharts and environments. In this domain, which is also characterized algebraically via semi-lattices, we develop a *fully-abstract* macro-step semantics in two steps. First, we study Statecharts without hierarchy operators. We show that in this fragment, stabilization sequences naturally characterize the largest congruence contained in equality of response behavior (cf. Sec. 3). In the second step, based on a non-standard *distributivity law* and our lattice-theoretic characterization of the intuitionistic semantics, we lift our results to arbitrary Statecharts (cf. Sec. 4). We refer the reader to [12] for the proofs of our results.

2 Statecharts: Notation, Semantics, & Compositionality

Statecharts is a visual language for specifying *reactive systems*, i.e., concurrent systems interacting with their *environment*. They subsume labeled transition systems where labels are pairs of *event* sets. The first component of a pair is referred to as *trigger*, which may include *negated events*, and the second as *action*. Intuitively, a transition is enabled if the environment offers all events in the trigger but not the negated ones. When a transition fires, it produces the events specified in its action. Concurrency is introduced by allowing Statecharts to run in parallel and to communicate by *broadcasting* events. Additionally, *basic states* may be hierarchically refined by injecting other Statecharts.

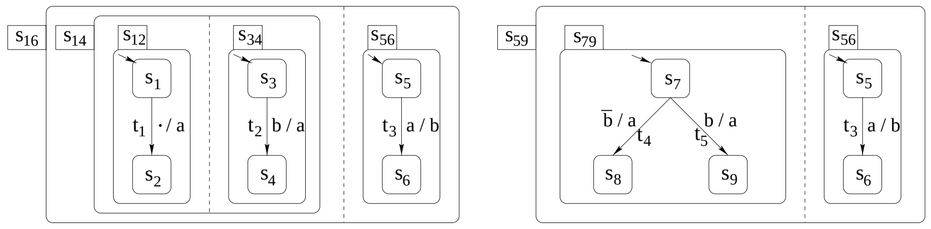

Fig. 1. Two example Statecharts

As an example, the Statechart depicted in Fig. 1 on the left consists of an *and-state* s_{16}, which puts and-state s_{14} and or-state s_{56} in parallel. Similarly, state s_{14} is a parallel composition of or-states s_{12} and s_{34}. Each of these *or-states* describes a sequential state machine and is refined by two *basic states*. In case of s_{12}, basic state s_1 is the initial state which is connected to basic state s_2 via transition t_1. Here, s_1 is the source state of t_1, state s_2 is its target state, "$\cdot$" symbolizes its empty trigger, and a is its action. Hence, t_1 is always enabled in the initial state, regardless of the events offered by the environment. Its firing produces event a and switches the active state of s_{12} to s_2. This initiates a causal chain reaction, since the generation of a in turn triggers t_3 which introduces event b. As a consequence, t_2 is enabled and fires within the same *macro step*.

The Statechart depicted in Fig. 1 on the right is like the one on the left, except that and-state s_{14} is replaced by or-state s_{79}. The latter state encodes a choice regarding the execution of t_4 and t_5 from state s_7. The trigger of t_4 is $\bar{b}$, i.e., t_4 is triggered by the absence of event b. Starting with an environment offering no event, thus assuming b to be absent, s_{59} can autonomously engage in t_4. The generation of a in turn triggers t_3, which fires and produces b. However, t_4 was fired under the assumption that b is absent. Since Statecharts is a synchronous language and no event can be both present and absent within a macro step, this behavior is rejected as *globally inconsistent*. Thus, the response of s_{59} to the empty environment is not an empty response but failure.

Statecharts Configurations and Step Semantics. We formalize the Statecharts language relative to a given set of active states. Let Π and $\mathcal{T}$ be countable sets of events and transition names, respectively. For every event $e \in \Pi$, its negated counterpart is denoted by $\overline{e}$. We define $\overline{\overline{e}} =_{\mathrm{df}} e$ and write $\overline{E}$ for $\{\overline{e} \mid e \in E\}$. With every $t \in \mathcal{T}$, we associate a transition $\mathsf{trg}(t)/\mathsf{act}(t)$ consisting of a trigger $\mathsf{trg}(t) \subseteq_{\mathrm{fin}} \Pi \cup \overline{\Pi}$ and an action $\mathsf{act}(t) \subseteq_{\mathrm{fin}} \Pi$, where $\mathsf{trg}(t)$ and $\mathsf{act}(t)$ are required to be finite sets. For simplicity we also write $e_1 \cdots e_n / a_1 \cdots a_m$ for transition $\{e_1, \ldots, e_n\}/\{a_1, \ldots, a_m\}$. The syntax of Statecharts terms is the BNF $C ::= \mathbf{0} \mid x \mid t \mid C \| C \mid C + C$, where $t \in \mathcal{T}$ and x is a variable. Terms not containing variables are called *configurations*. Intuitively, the configuration $\mathbf{0}$ represents a Statechart state with no outgoing transitions (basic state), $C \| D$ denotes the parallel composition of configurations C and D (and-state), and $C + D$ stands for the choice between executing C or D (or-state). The latter construct $+$ coincides with Statecharts' hierarchy operator which reduces to choice on the macro-step level; thus, we refer to operator $+$ also as choice operator. In the standard visual Statecharts notation, $C + D$ is somewhat more restrictive in that it requires D to be a choice of transitions; e.g., $(t_1 \| t_2) + (t_3 \| t_4)$ is prohibited according to Statecharts' syntax, whereas it is a valid configuration in our setting. Semantically, however, our generalization is inessential wrt. the semantics of Pnueli and Shalev which underlies this work (cf. [12]). The set of all configurations is denoted by C and ranged over by C and D. The set of "+"-free, or *parallel*, configurations is written as PC. We call terms $\Phi[x]$ with a single variable occurrence x *contexts* and write $\Phi[C]$ for the substitution of C for x in $\Phi[x]$. Contexts of the form $x \| C$ and $x + C$ are referred to as *parallel contexts* and *choice contexts*, respectively. We tacitly assume that transition names are unique in every term, and we let $\mathsf{trans}(C)$ stand for the set of transition names occurring in C.

Any Statechart in a given set of active states corresponds to a configuration. For example, Statecharts s_{14} and s_{79}, in their initial states (indicated by small arrows in Fig. 1), correspond to $C_{14} =_{\mathrm{df}} t_1 \| t_2$ and $C_{79} =_{\mathrm{df}} t_4 + t_5$, respectively.

The Statecharts depicted in Fig. 1 are then formalized as $C_{16} =_{\rm df} \Phi_{56}[C_{14}]$ and $C_{59} =_{\rm df} \Phi_{56}[C_{79}]$, respectively, where $\Phi_{56}[x] =_{\rm df} x \| t_3$. Moreover, since transitions are uniquely named in configurations and thus may be associated with their source and target states, one can easily determine the set of active states reached after firing a set of transitions; see [16] for details. In this paper, we do not consider *interlevel transitions* and *state references* which would require an extension of our syntax for configurations. However, our semantics should be able to accommodate these features, too.

To present the *response behavior* of a configuration C, as defined by Pnueli and Shalev [16], we have to determine which transitions in $\mathsf{trans}(C)$ may fire together to form a macro step. A macro step comprises a *maximal* set of transitions that are *triggered* by events offered by the environment or produced by the firing of other transitions, that are mutually *consistent* ("orthogonal"), and that obey *causality* and *global consistency.* A transition t is consistent with $T \subseteq \mathsf{trans}(C)$, in signs $t \in \mathsf{consistent}(C, T)$, if t is not in the same parallel component as any $t' \in T$. A transition t is *triggered* by a finite set E of events, in signs $t \in \mathsf{triggered}(C, E)$, if the positive, but not the negative, trigger events of t are in E. Finally, we say that t is *enabled* in C regarding a finite set E of events and a set T of transitions, if $t \in \mathsf{enabled}(C, E, T) =_{\rm df} \mathsf{consistent}(C, T) \cap \mathsf{triggered}(C, E \cup \bigcup_{t \in T} \mathsf{act}(t))$. Intuitively, assuming transitions T are known to fire, $\mathsf{enabled}(C, E, T)$ determines the set of all transitions of C that are enabled by the actions of T and the environment events in E. We may now present Pnueli and Shalev's *step-construction procedure* for causally determining macro steps:

```
procedure $step\text{-}construction(C,\ E)$; var $T := \emptyset$;
  while $T \subset \mathsf{enabled}(C, E, T)$ do choose $t \in \mathsf{enabled}(C, E, T) \setminus T$; $T := T \cup \{t\}$ od;
  if $T = \mathsf{enabled}(C, E, T)$ then (return $T$) else ($report\ failure$)
```

This procedure *nondeterministically* computes, relative to configuration C and finite environment E, those sets T of transitions that can fire together in a macro step. Due to failures raised when detecting global inconsistencies, the construction might involve *backtracking.* The role of failures may be highlighted further by a conservative extension of Pnueli and Shalev's setting that includes an explicit failure event $\perp \in \Pi$. It will be instructive to study the semantics with and without $\perp$ in this paper. Now, for each set T returned by the above procedure, we say that $A =_{\rm df} E \cup \bigcup_{t \in T} \mathsf{act}(t) \subseteq_{\rm fin} \Pi$ is a *(step) response*, in signs $C \Downarrow_E A$. When $\perp$ is considered, we also require that $\perp \notin A$. If $E = \emptyset$, we simply write $C \Downarrow A$. Note that E may be modeled by a parallel context consisting of the single transition $\cdot/E$, i.e., $C \Downarrow_E A$ iff $(C \| \cdot/E) \Downarrow A$. This macro-step semantics induces a natural equivalence relation $\sim$ over configurations, called *step equivalence*, satisfying $C \sim D$, whenever $C \Downarrow_E A$ iff $D \Downarrow_E A$, for all $E, A \subseteq_{\rm fin} \Pi$. For simplicity, $\sim$ does not account for target states of transitions since these can be encoded as event names.

The Compositionality Problem. The compositionality defect of the macro-step semantics manifests itself in the fact that $\sim$ is not a congruence for the configuration algebra. Consider Fig. 1 and assume that states s_2, s_4, s_6, s_8, and s_9 are all equivalent. It is easy to see that configurations C_{14} and C_{79} have the same response behavior. Both $C_{14} \Downarrow_E A$ and $C_{79} \Downarrow_E A$ are equivalent to $A = E \cup \{a\}$, no matter whether event b is present or absent in environment E. However, $\Phi_{56}[C_{14}] = C_{16} \not\sim C_{59} = \Phi_{56}[C_{79}]$ since $C_{16} \Downarrow \{a, b\}$ but $C_{59} \not\Downarrow A$, for any A. Hence, the equivalence $C_{14} \sim C_{79}$ is not preserved by context $\Phi_{56}[x]$. Intuitively, C_{14} and C_{79} are identified because the response semantics does not account for any interaction with the environment. It adopts the classic *closed-world assumption*, stating that every environment event is either present from the very beginning of a given macro step or will never arise. This eliminates the possibility that events may be generated due to interactions with the environment, in this case event b in $C_{16} \Downarrow \{a, b\}$. In short, a *compositional* macro-step semantics does not validate the *Law of the Excluded Middle* $b \vee \neg b = \mathit{true}$. Since *intuitionistic logic* [18] differs from classic logic by refuting the Law of the Excluded Middle, it is a good candidate framework for analyzing Statecharts semantics. It should be stressed that the compositionality defect is mainly an issue of operator $\|$ and not of $+$, as we will see below.

Our goal is to characterize the largest congruence $\simeq$, called *step congruence*, contained in step equivalence, where $C \simeq D$, if $\Phi[C] \sim \Phi[D]$ for all contexts $\Phi[x]$. Of course, $C \simeq D$ iff $[\![C]\!]_0 = [\![D]\!]_0$, for $[\![C]\!]_0 =_{\mathrm{df}} \{\langle A, \Phi[x]\rangle \mid \Phi[C] \Downarrow A\}$. However, $[\![\cdot]\!]_0$ is a syntactical characterization rather than a semantical characterization which we will develop below. Note that we intend to achieve compositionality in the (declarative) sense of a fully-abstract semantics and not in the (constructive) sense of a denotational semantics.

3 Macro-step Semantics via Stabilization Sequences

We start off by investigating parallel configurations within parallel contexts. We propose a novel semantics for this fragment, show its relation to Pnueli and Shalev's original semantics, and derive a full-abstraction result. Section 4 generalizes this result to arbitrary configurations within arbitrary contexts.

Our new interpretation of parallel configurations C, based on an "open-world assumption," is given in terms of *finite increasing sequences of "worlds"* $E_0 \subset E_1 \subset \cdots \subset E_n$. Each $E_i \subseteq_{\mathsf{fin}} \Pi \setminus \{\bot\}$ is the set of events generated or present in the respective world. The required absence of $\bot$ ensures that each world is consistent. A sequence represents the interactions between C and a potential environment during a macro step. Intuitively, the initial world E_0 contains all events e which are generated by those transitions of C that can fire autonomously. When transitioning from world E_{i-1} to E_i, some events in $E_i \setminus E_{i-1}$ are provided by the environment, as reaction to the events validated by C when reaching E_{i-1}. The new events destabilize world E_{i-1} and may enable a chain reaction of transitions in C. The step-construction procedure, which tracks and accumulates all these events, then defines the new world E_i. Accordingly, we call the above sequences *stabilization sequences*. The overall response of C after n interactions with the environment is the set E_n.

The monotonicity requirement of stabilization sequences reflects the fact that our knowledge of the presence and absence of events increases within the construction of a macro step. Each world contains the events assumed or positively known to be present. Only if an event is not included in the final world, it is known to be absent for sure; the fact that an event e is not present in a world does not preclude e from becoming available later in the considered stabilization sequence. This semantic gap between "not present" and "absent" makes the underlying logic *intuitionistic* as opposed to classic.

Model-theoretic Semantics for Parallel Configurations. Formally, a stabilization sequence M is a pair (n, V), where $n \in \mathbb{N} \setminus \{0\}$ is its *length* and V is a *state valuation*, i.e., a monotonic mapping from the interval $[0, \ldots, n-1]$ to finite subsets of $\Pi \setminus \{\bot\}$. The *final world* $V(n-1)$ of M is denoted by M^*. We shall assume that M is *irredundant*, i.e. $V(i-1) \subset V(i)$ for all $0 < i < n$, and identify sequences $(1, V)$ of length 1 with subsets $V(0) \subseteq_{\text{fin}} \Pi \setminus \{\bot\}$.

Definition 1. *Let $M = (n, V)$ be a stabilization sequence and $C \in \mathsf{PC}$. Then, M is a* sequence model *of C, written $M \models C$, according to the following clauses: (i) always $M \models \mathbf{0}$; (ii) $M \models C \| D$ iff $M \models C$ and $M \models D$; (iii) $M \models E/A$ iff $\{\overline{E \cap \overline{\Pi}} \cap V(n-1) = \emptyset$ and $E \cap \Pi \subseteq V(i)\}$ implies $A \subseteq V(i)$, for all $0 \leq i < n$.*

Def. 1 is a shaved version of the standard semantics obtained when reading $C \in \mathsf{PC}$ as an intuitionistic formula [18], i.e., when taking events to be atomic propositions and replacing $\overline{a}$ by negation $\neg a$, concatenation of events and "$\|$" by conjunction "$\wedge$", and "$/$" by implication "$\supset$". An empty trigger, an empty action, and $\mathbf{0}$ are identified with *true*. Then, $M \models C$ iff C holds for the intuitionistic Kripke structure M. In the sequel we write $SM(C)$ for $\{M \mid M \models C\}$.

In our example $C_{79} = \overline{b}/a + b/a$ is step-congruent to $C'_{79} = \overline{b}/a \,\|\, b/a$ (cf. Sec. 4) which may be identified with formula $(\neg b \supset a) \wedge (b \supset a)$. In classic logic, C'_{79} is equivalent to the single transition $C_{12} = \cdot/a$ corresponding to formula *true* $\supset a$. As mentioned before, this is inadequate as both have different operational behavior, since $C'_{79} \| \, a/b$ fails in the empty environment whereas $C_{12} \| \, a/b$ has step response $\{a, b\}$. In our intuitionistic semantics, the difference is faithfully witnessed by the stabilization sequence $M = (2, V)$, where $V(0) = \emptyset$ and $V(1) = \{a, b\}$. Here, M is a sequence model of C'_{79} but not of C_{12}.

Characterization of Pnueli and Shalev's Semantics. We now show that the step responses of a parallel configuration C, according to Pnueli and Shalev's semantics, can be characterized as particular sequence models of C, to which we refer as *response models*. The response models of C are the sequence models of C of length 1, i.e. subsets of $\Pi \setminus \{\bot\}$ that do not occur as the final world of any other sequence model of C except itself.

Definition 2. *Let $C \in PC$. Then, $M = (1, V) \in SM(C)$ is a* response model *of C if $K^* = M^*$ implies $K = M$, for all $K \in SM(C)$.*

Intuitively, the validity of this characterization is founded in Pnueli and Shalev's closed-world assumption which requires a response to emerge from within the considered configuration and not by interactions with the environment.

Theorem 1. *Let $C \in \mathsf{PC}$ and $E, A \subseteq_{fin} \Pi$. Then, $C \Downarrow_E A$ iff A is a response model of $C \,\|\, \cdot\, / E$.*

Thm. 1 provides a simple model-theoretic characterization of operational step responses; e.g., configuration $\overline{a}/a$ forces Pnueli and Shalev's step-construction procedure to fail. Indeed, the only sequence model of $\overline{a}/a$ of length 1 (and using only event a) is $A = \{a\}$. But A is not a response model since it is the final world of $K = (2, V) \in \mathit{SM}(\overline{a}/a)$ with $V(0) =_{\mathrm{df}} \emptyset$ and $V(1) =_{\mathrm{df}} A$. Since $\overline{a}/a$ does not have any response model, it can only fail. As another example, consider $a/b \,\|\, b/a$ which possesses the sequence models $(2, V)$, where $V(0) =_{\mathrm{df}} \emptyset$ and $V(1) =_{\mathrm{df}} \{a, b\}$, and $(1, V')$, where $V'(0) =_{\mathrm{df}} \emptyset$. Only the latter is a response model, in accordance with causality. Thus, $(a/b \,\|\, b/a) \Downarrow \emptyset$ is the only response.

Full Abstraction. Sequence models also lead to a fully-abstract semantics for parallel configurations within parallel contexts.

Theorem 2. *Let $C, D \in \mathsf{PC}$. Then, $\mathit{SM}(C) = \mathit{SM}(D)$ iff $\forall R \in \mathsf{PC}\, \forall E, A \subseteq_{fin} \Pi.$ $C \| R \Downarrow_E A$ iff $D \| R \Downarrow_E A$.*

Hence, sequence models contain precisely the information needed to capture all possible interactions of a parallel configuration with all potential environments.

Characterization of Sequence Models. Of course, Thm. 2 does not mean that every set of stabilization sequences can be obtained from a (parallel) configuration. In fact, in intuitionistic logic it is known that in order to specify arbitrary linear sequences, nested implications are needed [18]. Configurations, however, only use first-order implications and negations. Their sequence models may be characterized by simple lattice structures which we refer to as *behaviors*.

Definition 3. *An A-*behavior *$\mathcal{C}$, for $A \subseteq_{fin} \Pi$, is a pair $\langle F, I \rangle$, where $F \subseteq 2^{A \setminus \{\perp\}}$ and I is a monotonic function that maps every $B \in F$ to a set $I(B) \subseteq 2^B$ such that $B \in I(B)$ and $I(B)$ is closed under intersection, i.e., $B_1, B_2 \in I(B)$ implies $B_1 \cap B_2 \in I(B)$, for all $B \in F$. Furthermore, $\mathcal{C}$ is called* bounded*, if $A \in F$.*

It is not difficult to show that the pairs of initial and final states occurring together in the sequence models of $C \in \mathsf{PC}$ induce a behavior. More precisely, if A is the set of events mentioned in C, then the *induced* A-behavior $\mathit{Beh}(C)$ of C is the pair $\langle F(C), I(C) \rangle$, where

$$\begin{aligned} F(C) &=_{\mathrm{df}} \{E \subseteq A \mid \exists (n, V) \in \mathit{SM}(C).\ V(n-1) = E\} \\ I(C)(B) &=_{\mathrm{df}} \{E \subseteq B \mid \exists (n, V) \in \mathit{SM}(C).\ V(0) = E \text{ and } V(n-1) = B\}\,. \end{aligned}$$

Note that the response models B of C are precisely those $B \in F(C)$ for which $I(C)(B) = \{B\}$. As desired, we obtain the following theorem.

Theorem 3. *$\forall C, D \in \mathsf{PC}$. $\mathit{Beh}(C) = \mathit{Beh}(D)$ iff $\mathit{SM}(C) = \mathit{SM}(D)$.*

In conjunction with Thm. 2 it is clear that equivalence in arbitrary parallel contexts can as well be decided by behaviors: $\mathit{Beh}(C) = \mathit{Beh}(D)$ iff $\forall R \in \mathsf{PC}$ $\forall E, A \subseteq_{\mathrm{fin}} \Pi.\ C \| R \Downarrow_E A$ iff $D \| R \Downarrow_E A$. In contrast to $\mathit{SM}(C)$, however, $\mathit{Beh}(C)$ provides an *irredundant* representation of parallel configurations:

Theorem 4. *$\mathcal{C}$ is a (bounded) A-behavior iff there exists a configuration $C \in \mathsf{PC}$ over events A (not using $\perp$) such that $\mathcal{C} = Beh(C)$.*

Summarizing, behaviors $Beh(C)$, where $C \in \mathsf{PC}$, yield a model representation of $SM(C)$. For each B in $F(C)$, the set $I(C)(B)$ is a $(\cap, \subseteq)$ semi-lattice with maximal element B. As a very simple example, consider $C =_{\mathrm{df}} bc/a \,\|\, ac/b \,\|\, \overline{a}/a \,\|\, \overline{b}/b \,\|\, \overline{c}/c$ over events $A = \{a, b, c\}$. Its corresponding bounded A-behavior $Beh(C)$ is depicted in Fig. 2. Since $F(C) = \{A\}$, we only have the $(\cap, \subseteq)$ semi-lattice $I(C)(A)$. Generally speaking, $SM(C)$ is the set of sequences whose world-wise intersection with A are paths in the lattice diagrams ending in maximal elements. Moreover, the maximal elements are the classic solutions of C which may become actual responses in suitable parallel contexts.

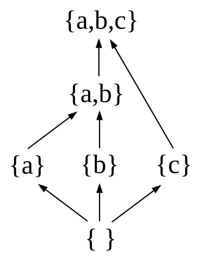

Fig. 2. Bounded $\{a, b, c\}$-behavior

4 Generalizing the Full-abstraction Result

In this section we reduce the problem of full abstraction for arbitrary configurations in arbitrary contexts to that for parallel configurations in parallel contexts.

Reduction to Parallel Contexts. For extending the full-abstraction result to arbitrary contexts, one must address a compositionality problem for $+$ which already manifests itself in Pnueli and Shalev's semantics. Consider configurations $C =_{\mathrm{df}} \overline{a}/b$ and $D =_{\mathrm{df}} \overline{a}/b \,\|\, a/a$ which have the same responses in all parallel contexts. However, in the choice context $\Phi[x] = (\cdot/e + x) \| \cdot/a$ we have $\Phi[D] \Downarrow \{a\}$ but $\Phi[C] \not\Downarrow \{a\}$ (as $\Phi[C] \Downarrow \{a, e\}$ only). This context is able to detect that D is enabled by the environment $\cdot/a$ while C is not. Hence, one has to take into account whether there exists a transition in C that is triggered for a set A of events. To store the desired information we use the *triggering indicator* $\rho(C, A) \in \mathbb{B} =_{\mathrm{df}} \{\mathit{ff}, \mathit{tt}\}$ defined by $\rho(C, A) =_{\mathrm{df}} \mathit{tt}$, if $\mathsf{triggered}(C, A) \neq \emptyset$, and $\rho(C, A) =_{\mathrm{df}} \mathit{ff}$, otherwise.

Lemma 1. *Let $C, D \in \mathsf{C}$. Then $C \simeq D$ iff $\forall P \in \mathsf{PC}, A \subseteq_{\mathit{fin}} \Pi, b \in \mathbb{B}$. $(C \| P \Downarrow A$ and $\rho(C, A) = b)$ iff $(D \| P \Downarrow A$ and $\rho(D, A) = b)$.*

Thus, to ensure compositionality for arbitrary contexts we only need to record $[\![C]\!]_1^b =_{\mathrm{df}} \{\langle A, P\rangle \,|\, C \| P \Downarrow A,\ \rho(C, A) = b,\ P \in \mathsf{PC}\}$, for $b \in \mathbb{B}$, instead of $[\![C]\!]_0$. We may view $[\![C]\!]_1^{tt}$ as the collection of *active* and $[\![C]\!]_1^{\mathit{ff}}$ as the collection of *passive* responses for C in parallel contexts, according to whether a transition of C takes part in response A. By Lemma 1, $C \simeq D$ iff $[\![C]\!]_1^{tt} = [\![D]\!]_1^{tt}$ and $[\![C]\!]_1^{\mathit{ff}} = [\![D]\!]_1^{\mathit{ff}}$.

Reduction to Parallel Configurations. For eliminating the choice operator from configurations we employ a *distributivity law*. However, the naive distributivity law $C \simeq D$ for $C =_{\mathrm{df}} (t_1 + t_2) \| t_3$ and $D =_{\mathrm{df}} (t_1 \| t_3') + (t_2 \| t_3'')$, where transitions t_3' and t_3'' are identical to t_3 except for their name, does in general not hold. Consider $t_i =_{\mathrm{df}} a_i \overline{b_i}/c_i$, for $1 \leq i \leq 3$, and assume that all events are mutually distinct. Then, in a context in which t_2 is enabled but not t_1, transition t_3 in C is forced to interact with t_2, while in D transition t_3' may run by itself in the summand $t_1 \| t_3'$. E.g., if $E = \{a_2, a_3\}$ then $D \Downarrow_E \{c_3, a_2, a_3\}$, but the only A with $c_3 \in A$ and $C \Downarrow_E A$ is $A = \{c_2, c_3, a_2, a_3\}$.

The naive distributivity law can be patched by adding configurations $D_1(t_3)$ and $D_2(t_3)$ such that $C \simeq t_1 \| D_1(t_3) + t_2 \| D_2(t_3)$. Here, $D_i(t_3)$ must weaken t_3 such that it disables t_3, whenever t_i is not enabled but t_{3-i} is. A simple way to achieve this is to define $D_1(t_3) =_{\mathrm{df}} D_1 \| t_3'$ and $D_2(t_3) =_{\mathrm{df}} D_2 \| t_3''$, where $D_i =_{\mathrm{df}} \overline{a}_i a_{3-i} \overline{b}_{3-i} / \bot \; \| \; b_i a_{3-i} \overline{b}_{3-i} / \bot$, for $i \in \{1, 2\}$. As desired, the "watchdog" configuration D_i satisfies for all parallel contexts P: $D_i \| P \Downarrow A$ iff (i) $P \Downarrow A$ and (ii) A triggers t_i or does not trigger t_{3-i}. It should be clear how this can be generalized, i.e., how one constructs for any $C, D \in \mathsf{C}$ a configuration $\mathsf{watch}(C, D)$ such that $P \| \mathsf{watch}(C, D) \Downarrow A$ iff (i) $P \Downarrow A$ and (ii) $\mathsf{triggered}(C, A) \neq \emptyset$ or $\mathsf{triggered}(D, A) = \emptyset$.

Lemma 2. *Let $C_1, C_2, D \in \mathsf{C}$. Then, $(C_1 + C_2) \| D \simeq (\mathsf{watch}(C_1, C_2) \| C_1 \| D) + (\mathsf{watch}(C_2, C_1) \| C_2 \| D)$.*

The fact that we have available an explicit failure event $\bot$ makes this distributivity law particularly simple. The use of $\bot$, however, is inessential as it can be eliminated [12]. Now, by repeatedly applying distributivity we may push occurrences of operator $+$ to the outside of configurations.

Lemma 3. *Let $C \in \mathsf{C}$. Then, there exists a finite index set $\mathsf{ind}(C)$ and $C_i \in \mathsf{PC}$, for $i \in \mathsf{ind}(C)$, such that $C \simeq \sum_{i \in \mathsf{ind}(C)} C_i$.*

Hence, $[\![C]\!]_1 = [\![\sum_{i \in \mathsf{ind}(C)} C_i]\!]_1$. Moreover, since an active response of a sum must be an active response of *one* of its summands and since a passive response of a sum always is a passive response of *all* of its summands, $[\![\sum_{i \in \mathsf{ind}(C)} C_i]\!]_1^{tt} = \bigcup_{i \in \mathsf{ind}(C)} [\![C_i]\!]_1^{tt}$ and $[\![\sum_{i \in \mathsf{ind}(C)} C_i]\!]_1^{ff} = \bigcap_{i \in \mathsf{ind}(C)} [\![C_i]\!]_1^{ff}$ hold. Thus, we obtain:

Lemma 4. *Let $C, D \in \mathsf{C}$. Then, $C \simeq D$ iff $\bigcup_{i \in \mathsf{ind}(C)} [\![C_i]\!]_1^{tt} = \bigcup_{j \in \mathsf{ind}(D)} [\![D_j]\!]_1^{tt}$ and $\bigcap_{i \in \mathsf{ind}(C)} [\![C_i]\!]_1^{ff} = \bigcap_{j \in \mathsf{ind}(D)} [\![D_j]\!]_1^{ff}$.*

Full-abstraction Result. Now, we are able to use our analysis of Sec. 3 to phrase Lemma 4 in terms of behaviors. All we need to do is to replace the parallel configuration $P \in \mathsf{PC}$ in every pair $\langle A, P \rangle \in [\![C_i]\!]_1$, for $i \in \mathsf{ind}(C)$, by its behavior $\mathit{Beh}(P)$. It turns out that the pairs obtained in this way can be uniquely determined from the behavior $\mathit{Beh}(C_i)$ of C_i, for any $i \in \mathsf{ind}(C)$.

Definition 4. *Let $A \subseteq_{\mathit{fin}} \Pi$. An A-behavior $\langle F, I \rangle$ is called an A-context for $C \in \mathsf{PC}$ if (i) $F = \{A\}$, (ii) $A \in F(C)$, and (iii) $I(A) \cap I(C)(A) = \{A\}$.*

Note that A-contexts for C are bounded behaviors, i.e., they can be represented without $\bot$. An A-context $\mathcal{P}$ of C represents a set of sequences that all end in the final world A, in which also some sequence model of C must end and which only have world A in common with the sequence models of C ending in A. These properties imply $C \| P \Downarrow A$, for every P with $\mathit{Beh}(P) = \mathcal{P}$. Hence, A-contexts $\mathcal{P}$ are "relativized complements" of C wrt. the final response A.

Consider again example C from above, whose sequence models $SM(C)$ are described by the behavior of Fig. 2. To get the A-contexts of C, where $A = \{a, b, c\}$, we must take the "complement" of $I(C)(A)$, i.e., all $B \subset A$ that are missing in the lattice of Fig. 2. As shown in Fig. 3, C has two A-contexts $\mathcal{P}_1$ and $\mathcal{P}_2$ covering this complement; configurations that denote them are $P_1 =_{\mathrm{df}} \cdot/ac \,\|\, \overline{b}/b$ and $P_2 =_{\mathrm{df}} \cdot/bc \,\|\, \overline{a}/a$, respectively. These provide complete information since every A-context must be contained in $\mathcal{P}_1$ or $\mathcal{P}_2$. For all $C \in \mathsf{PC}$ and $b \in \mathbb{B}$ we are finally led to define $[\![C]\!]_2^b =_{\mathrm{df}} \{\langle A, \mathcal{P}\rangle \mid A \subseteq_{\mathrm{fin}} \Pi, \rho(C, A) = b, \mathcal{P}$ is A-context of $C\}$ and obtain as a corollary to Lemma 4 and Thm. 2:

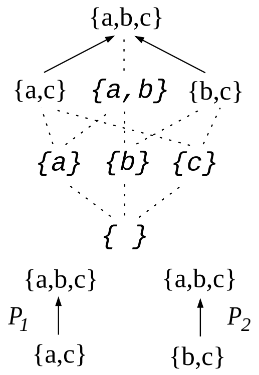

Fig. 3. $\{a, b, c\}$-contexts

Theorem 5. *Let $C, D \in \mathsf{C}$. Then, $C \simeq D$ iff $\bigcup_{i \in ind(C)} [\![C_i]\!]_2^{tt} = \bigcup_{j \in ind(D)} [\![D_j]\!]_2^{tt}$ and $\bigcap_{i \in ind(C)} [\![C_i]\!]_2^{ff} = \bigcap_{j \in ind(D)} [\![D_j]\!]_2^{ff}$.*

With Thm. 5 we have finally achieved our goal, as $[\![C]\!]_2$ is satisfactorily semantical and finite. In combination with Lemma 2 it directly lends itself to be applied for a model-based implementation of Pnueli and Shalev's semantics, which does not require backtracking for handling failure. Finally, it should be stressed that the above theorem also holds if we restrict ourselves to "+"-configurations of the form $C + t$, as in Statecharts, instead of permitting configurations $C + D$, for arbitrary $C, D \in \mathsf{C}$ (cf. Sec. 2). We now return to the example of Figs. 2 and 3. Let id_B be the behavior $\langle \{B\}, B \mapsto \{B\}\rangle$, for $B \subseteq \Pi$. We have $[\![C]\!]_2^{tt} = \{\langle \{a, b, c\}, \mathcal{P}_i\rangle \mid i = 1, 2\}$ and $[\![C]\!]_2^{ff} = \emptyset$. The same semantics can be generated as $D_1 + D_2$, where $D_1 = bc/a \,\|\, \overline{b}/b \,\|\, \overline{a}/a$ and $D_2 = ac/b \,\|\, \overline{b}/b \,\|\, \overline{c}/c$, since $[\![D_i]\!]_2^{tt} = \{\langle \{a, b, c\}, \mathcal{P}_i\rangle\}$, $[\![D_1]\!]_2^{ff} = \{\langle \{a, b\}, \mathrm{id}_{\{a,b\}}\rangle\}$, $[\![D_2]\!]_2^{ff} = \{\langle \{b, c\}, \mathrm{id}_{\{b,c\}}\rangle\}$. Hence, $[\![D_1]\!]_2^{tt} \cup [\![D_2]\!]_2^{tt} = [\![C]\!]_2^{tt}$ and $[\![D_1]\!]_2^{ff} \cap [\![D_2]\!]_2^{ff} = \emptyset = [\![C]\!]_2^{ff}$. By Thm. 5, $C \simeq D_1 + D_2$. A similar reasoning reveals $C_{79} \simeq C'_{79}$ (cf. Sec. 3).

5 Discussion and Related Work

Our investigation focused on Pnueli and Shalev's presentation of Statecharts and its macro-step semantics. The elegance of their operational semantics manifests itself in the existence of an equivalent *declarative fixed point semantics* [16]. However, as illustrated in [16], this equivalence is violated when allowing disjunctions in transition triggers. For example, the configurations $(\overline{a} \vee b)/a$ and $\overline{a}/a \,\|\, b/a$ do not have the same response behavior. This subtlety can now be explained in our framework. In Pnueli and Shalev's setting, $\overline{a} \vee b$ is classically interpreted as "throughout a macro step, not a or b." In contrast, this paper reads the configuration as "throughout a macro step not a or throughout b."

Our framework can also be employed for analyzing various other variants of Statecharts semantics, such as the one of Maggiolo-Schettini et al. [14] which in turn is inspired by the process-algebraic semantics presented in [17]. In [14] the step-construction procedure cannot fail since a transition is only considered to be enabled, if it is enabled in the sense of Pnueli and Shalev *and* if it does not produce any event that violates global consistency. As an example, consider the configuration $C =_{\mathrm{df}} t_1 \| t_2$, where $t_1 =_{\mathrm{df}} a/b$ and $t_2 =_{\mathrm{df}} \overline{b}/a$. According to [14], when C is evaluated for the empty environment, response $\{a\}$ is obtained; in Pnueli and Shalev's semantics, however, the step construction fails. The difference can be explained in terms of stabilization sequences. While Pnueli and Shalev take t_1 to stand for the specification $a \supset b$ and t_2 for $\neg b \supset a$, Maggiolo-Schettini et al. apply the interpretation $a \supset (b \vee \neg b)$ for t_1 and $\neg b \supset (a \vee \neg a)$ for t_2. Indeed, as one verifies, $\{a\}$ then is a response model of $t_1 \| t_2$. Note again that $a \vee \neg a$ is different from *true* in intuitionistic logic. Generalizing this example, the transition semantics of [14] can be captured in terms of response models by reading a transition E/A as formula $E \supset (A \vee \neg A)$, if our setting would be extended to allowing disjunctions as part of actions.

Our intuitionistic approach is also related to recent work in *synchronous languages*, especially for Berry's *Esterel* [3]. In Esterel, causality was traditionally treated separately from compositionality and synchrony as part of type-checking specifications. If the (conservative) type checker found causality to be violated, it rejected the specification under consideration. Otherwise, the specification's semantics could be determined in a very simple fashion; one may — in contrast to Statecharts semantics — abstract from the construction details of macro steps while preserving compositionality, as shown by Broy in [4]. Version 5 of Esterel [2] replaced the treatment of causality by defining a semantics via a particular Boolean logic that is *constructive*, as is intuitionistic logic.

Denotational semantics and full abstraction were also studied by Huizing et al. [10, 11] for an early and later-on rejected Statecharts semantics [9]. That semantics does not consider global consistency, which makes their result largely incomparable to ours. Finally, it should be mentioned that the lack of compositionality of Statecharts semantics inspired the development of new languages, such as Alur et al.'s *communicating hierarchical state machines* [1].

6 Conclusions

To the best of our knowledge, this is the first paper to present a fully-abstract Statecharts semantics for Pnueli and Shalev's original macro-step semantics [16]. The latter semantics is non-compositional as it employs classic logic for interpreting macro steps. In contrast, our semantics borrows ideas from intuitionistic logic. It encodes macro steps via stabilization sequences which we characterized using semi-lattice structures, called behaviors. Behaviors capture the interactions between Statecharts and their environments and consistently combine the notions of causality, global consistency, and synchrony. Moreover, our approach suggests a model-based implementation of Pnueli and Shalev's semantics, thereby eliminating the need to implement failure via backtracking.

Acknowledgments. This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the authors were in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, Virginia 23681-2199, USA. We would also like to thank the members of the Sheffield Verification and Testing Group and our anonymous referees for their valuable comments and suggestions.

References

1. R. Alur, S. Kannan, and M. Yannakakis. Communicating hierarchical state machines. In *ICALP '99*, vol. 1644 of *LNCS*, pp. 169–178, 1999.
2. G. Berry. The constructive semantics of pure Esterel, 1999. Draft Version 3. Available at http://www-sop.inria.fr/meije/Personnel/Gerard.Berry.html.
3. G. Berry and G. Gonthier. The Esterel synchronous programming language: Design, semantics, implementation. *SCP*, 19(2):87–152, 1992.
4. M. Broy. Abstract semantics of synchronous languages: The example Esterel. Tech. rep. TUM-I9706, Munich Univ. of Technology, 1997.
5. W. Damm, B. Josko, H. Hungar, and A. Pnueli. A compositional real-time semantics of STATEMATE designs. In *Compositionality: The Significant Difference*, vol. 1536 of *LNCS*, pp. 186–238, 1997.
6. D. Harel. Statecharts: A visual formalism for complex systems. *SCP*, 8:231–274, 1987.
7. D. Harel and E. Gery. Executable object modeling with Statecharts. *IEEE Computer*, pp. 31–42, July 1997.
8. D. Harel and A. Naamad. The STATEMATE semantics of Statecharts. *ACM Trans. on Software Engineering*, 5(4):293–333, 1996.
9. D. Harel, A. Pnueli, J. Pruzan-Schmidt, and R. Sherman. On the formal semantics of Statecharts. In *LICS '87*, pp. 54–64. IEEE Computer Society Press, 1987.
10. C. Huizing. *Semantics of Reactive Systems: Comparison and Full Abstraction.* PhD thesis, Eindhoven Univ. of Technology, 1991.
11. C. Huizing, R. Gerth, and W.-P. de Roever. Modeling Statecharts behavior in a fully abstract way. In *CAAP '88*, vol. 299 of *LNCS*, pp. 271–294, 1988.
12. G. Lüttgen and M. Mendler. What is in a step: A fully-abstract semantics for Statecharts macro steps via intuitionistic Kripke models. Tech. rep. CS-00-04, Univ. of Sheffield, 2000.
13. G. Lüttgen, M. von der Beeck, and R. Cleaveland. Statecharts via process algebra. In *CONCUR '99*, vol. 1664 of *LNCS*, pp. 399–414, 1999.
14. A. Maggiolo-Schettini, A. Peron, and S. Tini. Equivalences of Statecharts. In *CONCUR '96*, vol. 1119 of *LNCS*, pp. 687–702, 1996.
15. F. Maraninchi. Operational and compositional semantics of synchronous automaton compositions. In *CONCUR '92*, vol. 630 of *LNCS*, pp. 550–564, 1992.
16. A. Pnueli and M. Shalev. What is in a step: On the semantics of Statecharts. In *TACS '91*, vol. 526 of *LNCS*, pp. 244–264, 1991.
17. A.C. Uselton and S.A. Smolka. A compositional semantics for Statecharts using labeled transition systems. In *CONCUR '94*, vol. 836 of *LNCS*, pp. 2–17, 1994.
18. D. van Dalen. Intuitionistic logic. In *Handbook of Philosophical Logic*, vol. III, chap. 4, pp. 225–339. Reidel, 1986.