Fully-abstract Statecharts Semantics via Intuitionistic Kripke Models

Gerald Lüttgen¹ and Michael Mendler²

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 Pnuei 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 Pnuei 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.

Pnuei 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 Argo [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 pre-orders [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 Pnuei 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 via semi-lattices and in which Pnueli and Shalev’s semantics may be explained by considering a distinguished sub-domain, 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.

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 or-state 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, "\( .\)" 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 \( s_{70} \). 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, \( s_{50} \) 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 \( s_{50} \) 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 \( \Pi \) and \( \mathcal{T} \) be countable sets of events and transition names, respectively. For every event \( e \in \Pi \), its negated counterpart is denoted by \( \bar{e} \). We define \( \mathcal{E} = \{ \bar{e} \mid e \in \mathcal{E} \} \). With every \( t \in \mathcal{T} \) we associate a transition \( E/A \) consisting of a trigger \( \text{trg}(t) = \bar{A} \subseteq \text{fin} \Pi \cup \mathcal{T} \) and an action \( \text{act}(t) = \bar{A} \subseteq \text{fin} \Pi \), where \( E \) and \( A \) are required to be finite sets. For simplicity we also write \( x_1 \cdots x_n | a_1 \cdots a_m \) for transition \( \{x_1, \ldots, x_n \}/\{a_1, \ldots, a_m \} \). The syntax of Statecharts terms is the BNF \( C ::= 0 | x + t | C \cdot C | C + C \), where \( t \in \mathcal{T} \) and \( x \) is a variable. Terms not containing variables are called configurations. Intuitively, the configuration \( 0 \) 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 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, but 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 called parallel contexts and choice contexts, respectively. We tacitly assume that transition names are unique in every term, and we let \( \text{trans}(C) \) 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 \( s_{70} \), in their initial states, correspond to \( C_{14} = C_{70} \).

$t_1||t_2\text{ and } C_{70} =_{df} t_4 + t_5$, respectively. The Statecharts depicted in Fig. 1 are then formalized as $C_{16} =_{df} \Phi_{56}[C_{14}]$ and $C_{50} =_{df} \Phi_{56}[C_{70}]$, respectively, where $\Phi_{56}[x] =_{df} x||t_3$. 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 as both require us to extend our syntax for configurations. However, our semantics is potentially able to accommodate these features.

To present the response behavior of a configuration $C$, as defined by Pnueli and Shalev [16], we have to determine which transitions in $\text{trans}(C)$ 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 $T \subseteq \text{trans}(C)$, in signs $t \in \text{consistent}(C,T)$, 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 $t \in \text{triggered}(C,E)$, 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 $t \in \text{enabled}(C,E,T) =_{df} \text{consistent}(C,T) \cap \text{triggered}(C,E \cup \bigcup_{t’ \in T} \text{act}(t’))$. Intuitively, assuming transitions $T$ are known to fire, $\text{enabled}(C,E,T)$ 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.

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

This procedure computes nondeterministically, 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 $A =_{df} E \cup \bigcup_{t \in T} \text{act}(t) \subseteq_{\text{fin}} \Pi$ 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 $C \Downarrow_{\perp} A$. Note that $E$ may be modeled by a parallel context consisting of the single transition $\cdot /E$, i.e., $C \Downarrow_{E} A$ if and only if $(C|| \cdot /E) \Downarrow_{\perp} A$. This macro-step semantics induces a natural equivalence relation $\sim$ over configurations, called step equivalence, satisfying $C \sim D$, whenever $C \Downarrow_{E} A$ if and only if $D \Downarrow_{E} A$, for all $E, A \subseteq_{\text{fin}} \Pi$. 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 $s_2, s_4, s_6, s_8,$
We start 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" $E_0 \subseteq E_1 \subseteq \cdots \subseteq E_n$. Each $E_i \subseteq \Pi \setminus \{\bot\}$ 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 $E_n$.

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" $E_0 \subseteq E_1 \subseteq \cdots \subseteq E_n$. Each $E_i \subseteq \Pi \setminus \{\bot\}$ 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 $E_n$.  

and $s_0$ are all equivalent. It is easy to see that configurations $C_{14}$ and $C_{70}$ have the same response behavior. Both $C_{14} \upharpoonright E A$ and $C_{70} \upharpoonright E A$ are equivalent to $A = E \cup \{\alpha\}$, no matter whether event $b$ is present or absent in environment $E$. However, $\Phi_{50}[C_{14}] = C_{10} \neq C_{50} = \Phi_{50}[C_{70}]$ since $C_{10} \downarrow \{a, b\}$ but $C_{50} \downarrow A$, for any $A$. Hence, the equivalence $C_{14} \sim C_{70}$ is not preserved by context $\Phi_{50}[x]$. The intuitive reason for why $C_{14}$ and $C_{70}$ are identical is that the response semantics does not account for any proper interaction with the environment. It adopts the classical closed world assumption, stating that every 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 $C_{10} \downarrow \{a, b\}$. Consider further that $C_{70} = \not\equiv a + b/a$ is also step-equivalent to $C_{12} = \cdot a/a$. Hence, a compositional macro-step semantics does not validate the Law of the Excluded Middle $b \lor \neg b = \text{true}$. Since intuitionistic logic [18] differs from classic logic by refuting the Law of the Exclude Middle, it is a good candidate framework for analyzing Statecharts semantics. It should be stressed, however, 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, one can trivially obtain: $C \simeq D$ if and only if $[C]_0 = [D]_0$, for $[C]_0 = \{\langle A, \Phi[x] \rangle \mid \Phi[C] \downarrow A\}$, where $[\cdot]_0$ 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.
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. More precisely, each world contains the events assumed or 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 classical.

**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 **irrelevant**, i.e. $V(i - 1) \subseteq V(i)$ 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 $C \in PC$. Then, $M$ is a sequence model of $C$, written $M \models C$, according to the following clauses:

(i) always $M \models \emptyset$; (ii) $M \models C|D$ iff $M \models C$ and $M \models D$; (iii) $M \models E/A$ iff $\{E \cap \Pi \cap V(n - 1) = \emptyset$ 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 $C \in PC$ as an intuitionistic formula [18], i.e., when taking events to be atomic propositions and replacing \( \pi \) by negation $\neg a$, concatenation of events and “||” by conjunction “\&”, and “\ implies “\)” by implication “\.” An empty trigger, an empty action, and $\emptyset$ are identified with $\text{true}$. Then, $M \models C$ iff $M$ holds for the intuitionistic Kripke structure $M$. In the sequel we abbreviate $\{M \mid M \models C\}$ by $\text{SM}(C)$.

In our example, $C_{70} = b/a + b/a$ is step-congruent to $C'_{70} = b/a || b/a$ (cf. Sec. 4) which may be identified with formula $\langle \neg b \geq a \rangle \& (b \geq a)$. In classical logic, $C_{70}$ is equivalent to the single transition $C_{12} = \vdash a$ corresponding to formula $\text{true} \geq a$. As mentioned before, this is inadequate as both have different operational behavior, since $C_{70}||a/b$ fails 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 $C'_{70}$ 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, $M = (1, V) \in \text{SM}(C)$ is a response model of $C$ if $K^* = M^*$ implies $K = M$, for all $K \in \text{SM}(C)$.

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 \( C \in PC \) and \( E, A \subseteq_{fin} \Pi \). Then, \( C \downarrow_E A \) iff \( A \) is a response model of \( C \mid \cdot \downarrow E \).

Thm. 1 provides a simple model-theoretic characterization of operational step responses. For example, configuration \( \pi/a \) forces Pnueli and Shalev’s step construction procedure to fail. Indeed, the only sequence model of \( \pi/a \) 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 \( K = (2, V) \in SM(\pi/a) \) with \( V(0) = \alpha \neq \emptyset \) and \( V(1) = \alpha \neq \emptyset \). Since \( \pi/a \) does not have any response model, it can only fail. As another example, consider \( a/b \parallel b/a \) which possesses the sequence models \((2, V)\), where \( V(0) = \alpha \neq \emptyset \) and \( V(1) = \alpha \neq \emptyset \), and \((1, V')\), where \( V'(0) = \alpha \neq \emptyset \). Only the latter is a response model, in accordance with causality. Thus, \((a/b) \parallel b/a \downarrow \emptyset \) is the only response.

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

**Theorem 2.** Let \( C, D \in PC \). Then, \( SM(C) = SM(D) \) iff \( \forall E, A \subseteq_{fin} \Pi. C \mid \cdot \downarrow_E A \iff D \mid \cdot \downarrow_E A \).

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 \( C \), for \( A \subseteq_{fin} \Pi \), is a pair \( (F, I) \), where \( F \subseteq 2^A \{=\} \) 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, \( C \) 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 \( C \in PC \) induce a behavior. More precisely, if \( A \) is the set of events mentioned in \( C \), then the induced \( A \)-behavior \( Beh(C) \) of \( C \) is the pair \( (F(C), I(C)) \), where

\[
F(C) =_\alpha \{ E \subseteq A \mid \exists (n, V) \in SM(C), V(n - 1) = E \}
\]

\[
I(C)(B) =_\alpha \{ E \subseteq B \mid \exists (n, V) \in SM(C), V(0) = E \land V(n - 1) = B \}.
\]

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.** \( \forall C, D \in PC. Beh(C) = Beh(D) \iff SM(C) = SM(D) \).

In conjunction with Thm. 2 it is clear that equivalence in arbitrary parallel contexts can as well be decided by behaviors: \( Beh(C) = Beh(D) \iff \forall E, A \subseteq_{fin} \Pi. C \mid \cdot \downarrow_E A \iff D \mid \cdot \downarrow_E A \). In contrast to \( SM(C) \), however, \( Beh(C) \) provides an *irredundant* representation of parallel configurations.
Theorem 4. C is a (bounded) A-behavior iff there exists a configuration $C \in PC$ over events $A$ (not using $\perp$) such that $C = \text{Beh}(C)$.

Summarizing, behaviors $\text{Beh}(C)$, where $C \in PC$, 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 $C = \text{af} \; bc/a \parallel ac/b \parallel \overline{a} \parallel \overline{b} \parallel \overline{c}$ over events $A = \{a, b, c\}$. Its corresponding bounded A-behavior $\text{Beh}(C)$ 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 classical solutions of $C$ which may become actual responses in suitable parallel contexts.

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 $C = \text{af} \; \pi/b$ and $D = \text{af} \; \sigma/b \parallel a/a$ which have the same responses in all parallel contexts. However, in the choice context $\Phi[x] = (\cdot/e + x) \cdot/a$ we obtain $\Phi[D] \downarrow \{a\}$ but $\Phi[C] \downarrow \{a\}$ (as $\Phi[C] \downarrow \{a, c\}$ 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 $\rho(C, A) \in \mathbb{B} = \text{af} \; \{\emptyset, tt\}$ defined by $\rho(C, A) = \text{af} \; tt$, if $\text{triggered}(C, A) \neq \emptyset$, and $\rho(C, A) = \text{af} \; \emptyset$, otherwise.

Lemma 1. Let $C, D \in C$. Then $C \simeq D$ iff $\forall P \in PC, A \subseteq_{\text{fin}} P, b \in \mathbb{B}$. $(\parallel C \parallel \downarrow A$ and $\rho(P, A) = b) \iff (\parallel D \parallel \downarrow A$ and $\rho(D, A) = b)$.

Thus, to ensure compositionality for arbitrary contexts we only need to record $[C]_\rho = \text{af} \; \{(A, P) | C \parallel \downarrow A, \rho(C, A) = b, P \in PC\}$, for $b \in \mathbb{B}$, instead of $[C]_\omega$. We may view $[C]_\rho$ as the collection of active and $[C]_\rho^p$ 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 $[C]_\rho = [D]_\rho$ and $[C]_\rho^p = [D]_\rho^p$.

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 $C = \text{af} \; (t_1 + t_2)$ and $D = \text{af} \; (t_1 \parallel t'_2) + (t_2 \parallel t'_2)$, where transitions $t'_2$ and $t'_2$ are identical to $t_3$ except for their name, does in general not hold. Consider $t_i = \text{af} \; a_i b_i/c_i$, 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, \( D_1(t_3) \) must weaken \( t_3 \) such that it disables \( t_3 \), whenever \( t_i \) is not enabled but \( t_{i-1} \) is. To achieve this, we define \( D_1(t_3) =_{af} D_1(t'_3) \) and \( D_2(t_3) =_{af} D_2(t'_3) \), where \( D_i =_{af} t_i|a_{i-1}|b_{i-1}/\| \) \( = t_{i-1}|a_i|b_i/\), 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 \( t_{i-1} \). It should be clear how this can be generalized, i.e., how one constructs, for any \( C, D \in C \), a configuration \( \text{watch}(C, D) \) such that \( P||\text{watch}(C, D) \downarrow A \) iff (i) \( P \downarrow A \) and (ii) \( \text{triggered}(C, A) \neq \emptyset \) or \( \text{triggered}(D, A) = \emptyset \).

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

The fact that we have available an explicit failure event \( \perp \) makes this distributivity law particularly simple. The use of \( \perp \), however, is inessential as it can be eliminated [2]. Now, by repeatedly applying distributivity we may push occurrences of operator + to the outside of configurations.

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

Hence, \( [C]_1 = [\sum_{i \in \text{ind}(C)} C_i]_1 \). 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, \([\sum_{i \in \text{ind}(C)} C_i]_1^u = \bigcup_{i \in \text{ind}(C)} [C_i]_1^u \) and \([\sum_{i \in \text{ind}(C)} C_i]_1^f = \bigcap_{i \in \text{ind}(C)} [C_i]_1^f \) hold. Thus, we obtain:

**Lemma 4.** Let \( C, D \in C \). Then, \( C \simeq D \) iff \( \bigcup_{i \in \text{ind}(C)} [C_i]_1^u = \bigcup_{j \in \text{ind}(D)} [D_j]_1^u \) and \( \bigcap_{i \in \text{ind}(C)} [C_i]_1^f = \bigcap_{j \in \text{ind}(D)} [D_j]_1^f \).

**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 \( P \in PC \) in every pair \( \langle A, P \rangle \in [C]_1 \), for \( i \in \text{ind}(C) \), by its behavior \( \text{Beh}(P) \). It turns out that the pairs obtained in this way can be uniquely determined from the behavior \( \text{Beh}(C_i) \) of \( C_i \), for any \( i \in \text{ind}(C) \).

**Definition 4.** Let \( A \subseteq \mathbb{R}^n \). An \( A \)-behavior \( \langle F, I \rangle \) is called an \( A \)-context for \( C \in PC \) 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. 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 the final world \( A \) in common with the sequence models of \( C \) ending in \( A \). These properties imply \( C||P \downarrow A \), for every configuration \( P \) with \( \text{Beh}(P) = \mathcal{P} \). 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 ⊂ A that are missing in the lattice of Fig. 2. As shown in Fig. 4, C has two A-contexts P₁ and P₂ covering this complement; configurations that denote them are P₁ =_{df} ·(/ac) \parallel \bar{b}/b and P₂ =_{df} ·(/bc) \parallel \pi/a, respectively. These provide complete information since every A-context must be contained in P₁ or P₂. For all C ∈ P and b ∈ B we are finally led to define [C]_2^B =_{df} \{A, P\} \mid A \subseteq \text{fin} \Pi, \rho(C, A) = b, P is A-context of C\} and obtain as a corollary to Lemma 4 and Thm. 2.

**Theorem 5.** Let C, D ∈ C. Then, C ≃ D \iff \bigcup_{j \in \text{ind}(C)} [C]_2^j = \bigcup_{j \in \text{ind}(D)} [D]_2^j and \bigcap_{j \in \text{ind}(C)} [C]_2^j = \bigcap_{j \in \text{ind}(D)} [D]_2^j.

With Thm. 5 we have finally achieved our goal, as [C]_2^j 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 ∈ C (cf. Sec. 2). We refer the reader to [12] for details.

Let us return to the example of Fig. 2: [C]_2^B = \{\{(a, b, c), P₁\}, \{(a, b, c), P₂\}\} and [D]_2^B = \{\{a, c\}\}. This structure can be generated from the sum D₁ + D₂, where D₁ = ac/a \parallel \bar{b}/b \parallel \pi/a and D₂ = ac/b \parallel \bar{b}/b \parallel \pi/c, since [D₁]_2^B = \{\{(a, b, c), P₁\}\}, [D₂]_2^B = \{\{(a, b), \{a, b\}\}\}, [D₁]_2^B \cap [D₂]_2^B = \{\{(a, b), \{a, b\}\}\}. Hence, [D₁]_2^B \cup [D₂]_2^B = [C]_2^B and [D₁]_2^B \cap [D₂]_2^B = \{\{a, c\}\}. By Thm. 5, C ≃ D₁ + D₂. A similar reasoning reveals C₁ ≃ C₁₁ (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 (π ∨ b)/a and (π/π/ 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, π ∨ 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 $C = df t_1 || t_2$, where $t_1 =_{df} a \rightarrow b$ and $t_2 =_{df} \overline{b} \rightarrow a$. According to [14], when $C$ is evaluated for the empty environment, then 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 in intuitionistic logic $a \vee \neg a$ is different from true. 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 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 was found to be non-compositional as it employs classical 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 anonymous referees for their valuable comments and suggestions.

References
