# The Intuitionism Behind Statecharts Steps

GERALD LÜTTGEN and MICHAEL MENDLER
The University of Sheffield

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. 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 insight into the semantic subtleties of Statecharts, but also provide a basis for an implementation, for developing algebraic theories for macro steps, and for comparing different Statecharts variants.

Categories and Subject Descriptors: D.2.1 [**Software Engineering**]: Requirements/Specifications—*languages*; D.2.2 [**Software Engineering**]: Design Tools and Techniques—*state diagrams*; D.3.1 [**Programming Languages**]: Formal Definitions and Theory—*semantics*; F.3.2 [**Logics and Meanings of Programs**]: Semantics of Programming Languages—*algebraic approaches to semantics*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*model theory*

General Terms: Languages, Theory

Additional Key Words and Phrases: Compositionality, full abstraction, intuitionistic logic, Kripke semantics, Statecharts

## 1. INTRODUCTION

*Statecharts* is a well–known visual design notation for specifying the behavior of *reactive systems* [Harel 1987]. It extends *finite state machines* with concepts of (i) *hierarchy*, so that one may speak of a state as having sub–states, (ii) *concurrency*, thereby allowing the definition of systems having simultaneously active sub–systems, and (iii) *priority*, such that one may express that certain system activities have precedence over others. The success of Statecharts in the software–engineering community is founded on the language's capability for intuitively modeling the com-

Authors' address: Department of Computer Science, The University of Sheffield, Regent Court, 211 Portobello Street, Sheffield S1 4DP, U.K., {*G.Luettgen,M.Mendler*}*@dcs.shef.ac.uk*.
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), Mail Stop 132C, NASA Langley Research Center, Hampton, Virginia 23681–2199, USA. The second author also acknowledges support under EPSRC grant GR/M99637.
An extended abstract of this paper appeared in U. Montanari, J. Rolim, and E. Welzl, eds., *27th Intl. Col. on Automata, Languages and Programming* (ICALP 2000), vol. 1853 of Lecture Notes in Computer Science, pp. 163–174, Geneva, Switzerland, July 2000, Springer–Verlag.

plex control aspects inherent in many software systems. However, the search for a practically and theoretically satisfying semantics for Statecharts is still actively pursued at many research laboratories and has led to the definition of numerous Statecharts variants [von der Beeck 1994].

In a seminal paper, Pnueli and Shalev [1991] presented two equivalent formalizations of Statecharts semantics. According to their semantic model, 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. Unfortunately, Pnueli and Shalev's semantics violates the desired property of *compositionality* which is a prerequisite for modular analyses of Statecharts specifications as well as for compositional code generation. Huizing and Gerth, published in [Huizing 1991], 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 [Berry 2000] and ARGOS [Maraninchi 1992], causality is treated separately from compositionality and synchrony, while in (synchronous) STATEMATE [Harel and Naamad 1996] the synchrony hypothesis is rejected. Other researchers achieved combining all three properties by storing complex semantic information via preorders [Levi 1997; Maggiolo-Schettini et al. 1996; Uselton and Smolka 1994] or transition systems [Damm et al. 1997; Lüttgen et al. 2000]. However, no analysis of exactly how much information is needed to achieve compositionality has been made so far.

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. The reason is that macro steps abstract from causal interactions with a system's environment, thereby imposing a closed–world assumption. As we will show in more detail in the next section, the studied problem can be traced back to the invalidity of the *Law of the Excluded Middle*, which is the classical axiom representing the closed–world assumption. The point is that Pnueli and Shalev's semantics implicitly assumes that every event is either present or absent throughout any given macro step. However, this ignores the possibility that an event might be introduced by a system's environment in the middle of a macro step. A compositional semantics must be faithful to the non–atomicity of a macro–step and must not exclude this "middle possibility."

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* [van Dalen 1986], namely linear increasing sequences of event sets, called *stabilization sequences*, which encode interactions between Statecharts and environments. In this domain, which we characterize 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; note that the hierarchy operator is in fact a choice operator in our setting since we observe single macro steps only. We show that in this fragment, stabilization sequences naturally characterize the largest congruence contained in equality of response behavior. In the second step, based on

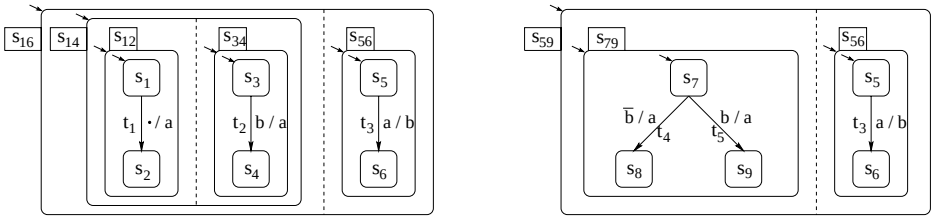


Fig. 1. Two example Statecharts

a non–standard *distributivity* and *expansion law*, as well as on our lattice–theoretic characterization of our intuitionistic semantics, we lift our results to arbitrary Statecharts. It is worth remarking that these results are achieved in a slightly extended Statecharts algebra that allows for general choice operators and also introduces an explicit *failure event*. We show that this extension is conservative over the standard "visual" syntax of Statecharts. As a byproduct, this paper suggests a natural way of admitting disjunctions in transition triggers, thereby solving a logical inadequacy of Pnueli and Shalev's setting. Moreover, our results build a foundation for an efficient implementation of Pnueli and Shalev's semantics that avoids backtracking, for algebraic characterizations of macro–step semantics, and also for comparisons among related Statecharts variants.

The remainder of this paper is organized as follows. The next section presents our notation for Statecharts, recalls the classical Statecharts semantics of Pnueli and Shalev, and analyzes the compositionality problem. Sec. 3 presents a new intuitionistic semantics for Statecharts macro steps, characterizes Pnueli and Shalev's semantics within the novel framework, and provides a full-abstraction result for the Statecharts language without hierarchy operator. The latter result is extended to the full language in Sec. 4. Finally, Secs. 5 and 6 discuss related work and present our conclusions and directions for future work, respectively.

## 2. STATECHARTS: SYNTAX, SEMANTICS, AND 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 *negative events*, and the second as *action*. Intuitively, a transition is enabled if the environment offers all events in the trigger but not the negative 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. This creates composite states of two possible sorts, which are referred to as *and*–states and *or*–states, respectively. Whereas and–states permit parallel decompositions of states, or–states allow for sequential decompositions. Consequently, an and–state is *active* if all of its sub–states are active, and an or–state is active if exactly one of its sub–states is.

As an example, the Statechart in Figure 1 on the left consists of 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 regardless of the events offered by the environment. Its firing produces event $a$ and switches the active state of $s_{12}$ from $s_1$ to $s_2$. This initiates a causal chain reaction, since event $a$ in turn triggers transition $t_3$ in parallel component $s_{56}$ which introduces event $b$. Thus, transition $t_2$ in or–state $s_{34}$ becomes enabled and fires within the same *macro step*.

The Statechart depicted in Figure 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 transitions $t_4$ and $t_5$ from state $s_7$. The trigger of $t_4$ is $\overline{b}$, 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, and–state $s_{59}$ can autonomously engage in $t_4$. The generation of $a$ in turn triggers transition $t_3$ 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 simultaneously present and absent within the same macro step, this behavior is rejected as *globally inconsistent*. Thus, the response of $s_{59}$ to the empty environment is failure, which is operationally different from an empty response.

### 2.1 Statecharts Configurations and Step Semantics

Like Pnueli and Shalev [1991] we present the semantics of Statecharts as a single–step semantics which is given relative to a fixed but arbitrary set of active states. As a consequence, Statecharts' hierarchy operator acts exactly like a choice operator. Formally, let $\Pi$ and $\mathcal{T}$ be countably infinite sets of events and transition names, respectively. For every event $e \in \Pi$, its negative counterpart is denoted by $\overline{e}$. We define $\overline{\overline{e}} =_{\mathrm{df}} e$ and write $\overline{E}$ for $\{\overline{e} \,|\, e \in E\}$. With every $t \in \mathcal{T}$, we associate a transition $E/A$ consisting of a trigger $\mathsf{trg}(t) =_{\mathrm{df}} E \subseteq_{\mathrm{fin}} \Pi \cup \overline{\Pi}$ and an action $\mathsf{act}(t) =_{\mathrm{df}} A \subseteq_{\mathrm{fin}} \Pi$, where $E$ and $A$ are required to be finite sets. For simplicity, we use the abbreviation $e_1 \cdots e_n/a_1 \cdots a_m$ for transition $\{e_1, \ldots, e_n\}/\{a_1, \ldots, a_m\}$, and we denote an empty trigger or action in a transition by symbol '$\cdot$'. We also write $P, \overline{N}/A$ for label $E/A$ when we wish to distinguish the set $P =_{\mathrm{df}} E \cap \Pi$ of *positive* trigger events from the set $N =_{\mathrm{df}} \overline{E \cap \overline{\Pi}}$ of *negative* trigger events. Now, we may describe a Statechart relative to a set of active states as a term in the following BNF

$$C \quad ::= \quad \mathbf{0} \quad | \quad x \quad | \quad t \quad | \quad C \parallel C \quad | \quad C + C\,,$$

where $t \in \mathcal{T}$ and $x$ is a variable. Terms not containing variables are called *configurations*. Intuitively, configuration $\mathbf{0}$ represents a Statechart state with no outgoing transitions (basic state), configuration $t$ represents a Statechart state with outgoing transition $t$, $C \parallel D$ 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). As mentioned earlier, the latter construct $+$ coincides with Statecharts' hierarchy operator, which reduces to choice when considering single macro steps only; thus, we refer to operator $+$ also as choice operator. Moreover, in the visual Statecharts notation, $C + D$ is somewhat more restrictive, in that it requires $D$ to be a choice of transitions. For instance, $(t_1 \parallel t_2) + (t_3 \parallel t_4)$ is prohibited in Statecharts visual

syntax whereas it is a valid configuration in our setting. Semantically, however, our generalization is inessential with respect to the considered semantics of Pnueli and Shalev (cf. Sec. 4.4). The set of all configurations is denoted by $\mathcal{C}$ and ranged over by $C$ and $D$. The set of '+'–free, or *parallel*, configurations is written as $\mathcal{PC}$. 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 form $x \parallel C$ 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 state, correspond to configurations $C_{14} =_{\mathrm{df}} t_1 \parallel t_2$ and $C_{79} =_{\mathrm{df}} t_4 + t_5$, respectively. The Statecharts depicted in Figure 1 are then formalized as $C_{16} =_{\mathrm{df}} \Phi_{56}[C_{14}]$ and $C_{59} =_{\mathrm{df}} \Phi_{56}[C_{79}]$, respectively, using the parallel context $\Phi_{56}[x] =_{\mathrm{df}} x \parallel t_3$. 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 [Pnueli and Shalev 1991] for details. As in Pnueli and Shalev's paper, we do not consider *interlevel transitions*, *state references*, and *priority concepts along state hierarchies* [von der Beeck 1994] to keep our syntax for Statecharts sufficiently simple. Although the syntax would have to be extended, our semantics can accommodate these features, too (cf. Sec. 5.2). Finally, we want to remark that the unique naming of transitions is not an essential assumption but just a convenient means in the operational semantics to define the step response of a Statechart configuration. We will see that the intuitionistic model theory developed in this paper allows us to do away with naming transitions. This is in contrast to the approach of von der Beeck [2000] who develops a Statecharts semantics whose compositionality is solely based on the consideration of transition names; however, that semantics is by no means fully abstract.

To present the *response behavior* of a configuration $C$, as defined by Pnueli and Shalev, 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*, i.e., "orthogonal," and that obey *causality* and *global consistency*. In the following, we formally introduce some of these notions.

—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 $t' \in T$ with $t \neq t'$. Formally,

$$\mathsf{consistent}(C, T) =_{\mathrm{df}} \{t \in \mathsf{trans}(C) \mid \forall t' \in T.\ t \triangle_C t'\}\,,$$

where $t \triangle_C t'$, if $t = t'$ or if $t$ and $t'$ are on different sides of some occurrence of operator $\parallel$ in $C$.

—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 $E$. Formally,

$$\mathsf{triggered}(C, E) =_{\mathrm{df}} \{t \in \mathsf{trans}(C) \mid \mathsf{trg}(t) \cap \Pi \subseteq E \text{ and } \overline{(\mathsf{trg}(t) \cap \overline{\Pi})} \cap E = \emptyset\}\,.$$

—A transition $t$ is *enabled* in $C$ with respect to a finite set $E$ of events and a set $T$ of transitions, if $t \in \mathsf{enabled}(C, E, T)$ where

$$\mathsf{enabled}(C, E, T) =_{\mathrm{df}} \mathsf{consistent}(C, T) \cap \mathsf{triggered}(C, E \cup \bigcup_{t \in T} \mathsf{act}(t)) .$$

Intuitively, assuming the transitions in $T$ 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 $E$. In the following, we write $\mathsf{act}(T)$ for $\bigcup_{t \in T} \mathsf{act}(t)$.

With these preliminaries, we may now present the iterative *step–construction procedure* of Pnueli and Shalev [1991] for causally determining macro steps relative to a configuration $C$ and a finite set $E$ of environment events.

```
procedure step–construction(C, E);
  var T := ∅;
  while T ⊂ enabled(C, E, T) do
    choose t ∈ enabled(C, E, T) \ T;
    T := T ∪ {t}
  od;
  if T = enabled(C, E, T) then return T
  else report failure
end step–construction.
```

This procedure computes *nondeterministically*, relative to a configuration $C$ and a finite environment $E$, sets $T$ of those transitions that can fire together in a macro step. Note that due to failures raised when detecting global inconsistencies, the step construction might involve *backtracking*, which makes the above algorithm inefficient for implementation. To highlight the role of failures further in this paper, it will be useful to introduce a special failure event $\bot \in \Pi$, modeling a global inconsistency, for representing the failure behavior of the step semantics explicitly. For instance, we can then define transition $a/\bot$ which raises a failure exception as soon as event $a$ becomes present. Note that, e.g., the firing of transition $\overline{a}/a$, which can already be expressed in the standard syntax, raises a failure in the absence of event $a$. Using the failure event, this transition $\overline{a}/a$ becomes equivalent to $\overline{a}/\bot$. Hence, the representation of failure behavior is more symmetrical in the sense that it allows us to enforce the *presence* as well as the *absence* of certain events in a macro step. It should be stressed that, as we will show in Sec. 4.4, adding event $\bot$ is a conservative extension that does not change the semantics of the original Statecharts language. It permits, however, a more uniform algebra of configurations. In particular, having $\bot$ available has the important technical advantage that certain semantic constructions on the original Statecharts language become syntactically representable. Moreover, there are also new behaviors expressible that may be useful in applications. We will study both variants of Statecharts semantics, with and without $\bot$, in this paper.

Following Pnueli and Shalev [1991], a set $T$ of transitions is called *constructible*, for a given configuration $C$ and a finite set $E$ of environment events, if it can be obtained as a result of successfully executing procedure *step–construction*. Whenever

we wish to indicate the environment, we say that $T$ is *$E$–constructible*. For each $E$–constructible set $T$, set $A =_{\mathrm{df}} E \cup \mathsf{act}(T) \subseteq_{\mathrm{fin}} \Pi$ is called the *(step) response* of $C$ for $E$, in signs $C \Downarrow_E A$. If event $\bot$ is considered, we require $\bot \notin A$, too. Moreover, if $E = \emptyset$, we simply write $C \Downarrow A$. Note that $E$ may also be modeled by a parallel context consisting of a single transition $\cdot/E$, as $C \Downarrow_E A$ if and only if $(C \parallel \cdot/E) \Downarrow A$ holds. Pnueli and Shalev also provided an equivalent *declarative* definition of their operational step semantics. A set $T$ of transitions is called *$E$–separable* for $C$ if there exists a proper subset $T' \subset T$ such that $\mathsf{enabled}(C, E, T') \cap (T \setminus T') = \emptyset$. Further, $T$ is *$E$–admissible* for $C$ if (i) $T$ is $E$–inseparable for $C$, (ii) $T = \mathsf{enabled}(C, E, T)$, and (iii) $\bot \notin \mathsf{act}(T)$. When configuration $C$ and environment $E$ are understood, we also say that $T$ is *admissible* or *separable*, respectively.

THEOREM 2.1 PNUELI AND SHALEV [1991]. *For all configurations $C \in \mathcal{C}$ and event sets $E \subseteq_{fin} \Pi$, a set $T$ of transitions is $E$–admissible for $C$ iff $T$ is $E$–constructible for $C$.*

While this theorem emphasizes the mathematical elegance of Pnueli and Shalev's semantics, it still does not support efficient implementations. However, because of Thm. 2.1, one may confuse the notions of constructibility and admissibility. In fact, the approach we are going to present in the following sections is derived more conveniently from the declarative characterization.

## 2.2 The Compositionality Problem

The 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 $E, A \subseteq_{\mathrm{fin}} \Pi$. For simplicity, $\sim$ does not account for target states of transitions since these can be encoded in event names. The compositionality defect of the macro–step semantics manifests itself in the fact that $\sim$ is not a congruence for the configuration algebra. Consider our example of Figure 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 $A \subseteq_{\mathrm{fin}} \Pi$, as $C_{59}$ always fails for the empty environment. Hence, the equivalence $C_{14} \sim C_{79}$ is not preserved by context $\Phi_{56}[x]$. The intuitive reason for why $C_{14}$ and $C_{79}$ are identified in the first place is that the response semantics does not account for any proper interaction with the environment. It adopts the classical *closed–world assumption* which states 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, such as event $b$ in $C_{16} \Downarrow \{a, b\}$. In short, a *compositional* macro–step semantics does not validate the *Law of the Excluded Middle* $true = \neg b \vee b$. This law forces one to identify the configurations $C_{14} = \cdot/a \parallel b/a$ and $C_{79} = \overline{b}/a + b/a$, which are essentially equivalent to the single transitions $true/a$ and $\overline{b} \vee b \,/\, a$, respectively, when permitting generalized Boolean triggers. As seen above, this identification is inadequate in the light of compositionality. Since *intuitionistic logic* [van Dalen 1986] differs from classical logic by refuting the Law of the Excluded Middle, it is a good candidate framework for analyzing the step semantics of Statecharts.

It must be stressed that the compositionality defect is an issue of operator $\|$ and not of choice $+$. Configuration $C_{79} = \overline{b}/a + b/a$ has exactly the same behavior as $C'_{79} =_{\mathrm{df}} \overline{b}/a \parallel b/a$ which we could have used instead. The compositionality problem can also be demonstrated by the two parallel configurations $D_1 =_{\mathrm{df}} \cdot/a \parallel b/c$ and $D_2 =_{\mathrm{df}} \overline{b}/a \parallel b/ac$ which have the same step responses but can be distinguished in context $\Phi_{56}[x]$, as $\Phi_{56}[D_1] \Downarrow \{a, b, c\}$ but $\Phi_{56}[D_2] \not\Downarrow A$, for any $A \subseteq_{\mathrm{fin}} \Pi$.

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 $\Phi[x]$. While the compositionality defect is well–known, a fully–abstract semantics with respect to Pnueli and Shalev's macro–step semantics has not yet been presented in the literature. Of course, one can trivially obtain that $C \simeq D$ is equivalent to $[\![C]\!]_0 = [\![D]\!]_0$, where $[\![C]\!]_0 =_{\mathrm{df}} \{\langle A, \Phi[x]\rangle \mid \Phi[C] \Downarrow A\}$. However, $[\![\cdot]\!]_0$ is a syntactic characterization rather than a semantic one, 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 (cf. Sec. 5.1).

## 3. INTUITIONISTIC SEMANTICS VIA STABILIZATION SEQUENCES

We start off by investigating parallel configurations within parallel contexts, for which many semantic insights may already be obtained. First, we propose a novel intuitionistic semantics for this fragment, then show its close relation to Pnueli and Shalev's original semantics, and finally derive the desired full–abstraction result. The next section will generalize this result to our full configuration algebra, i.e., to arbitrary configurations within arbitrary contexts.

Our new semantic interpretation of parallel configurations $C$, based on an "open–world assumption," is given in terms of finite increasing sequences of *worlds* (or *states*) $E_0 \subset E_1 \subset \cdots \subset E_n$, for some natural number $n$. Each $E_i \subseteq_{\mathrm{fin}} \Pi \setminus \{\bot\}$ is the set of events generated or present in the respective world, and the 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 within $C$. The step–construction procedure, which tracks and accumulates all these events, then defines the new world $E_i$. In accordance with this intuition, we call the above sequences *stabilization sequences*. The overall response of $C$ after $n$ interactions with the environment is the event set $E_n$.

The monotonicity requirement of stabilization sequences reflects the fact that our knowledge of the presence and absence of events increases in the process of constructing 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, $e \notin E(i)$, 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. Indeed, we shall see that parallel configurations are most naturally viewed as intuitionistic formulas specifying linear intuitionistic Kripke models.

### 3.1 Intuitionistic Semantics for Parallel Configurations

Formally, a stabilization sequence $M$ is a pair $(n, V)$, where $n \in \mathbb{N} \setminus \{0\}$ is the *length* of the sequence and $V$ is a *state valuation*, i.e., a monotonic mapping from the interval $[0, \ldots, n-1]$ to finite subsets of $\Pi \setminus \{\bot\}$. Stabilization sequences of length $n$ are also referred to as *$n$–sequences*. It will be technically convenient to assume that $M$ is *irredundant*, i.e., $V(i-1) \neq V(i)$, for all $0 < i < n$.

*Definition* 3.1 *Sequence Model.* Let $M = (n, V)$ be an irredundant stabilization sequence and $C \in \mathcal{PC}$. Then, $M$ is said to be a *sequence model* of $C$, if $M \models C$, where the satisfaction relation $\models$ is defined along the structure of $C$ as follows:

(1) Always $M \models \mathbf{0}$,

(2) $M \models C \parallel D$ if $M \models C$ and $M \models D$, and

(3) $M \models P, \overline{N}/A$ if
$(N \cap V(n-1) = \emptyset$ and $P \subseteq V(i))$ implies $A \subseteq V(i)$, for all $i < n$.

This definition is a shaved version of the standard semantics obtained when reading a parallel configuration as an intuitionistic formula [van Dalen 1986], i.e., when taking events to be atomic propositions and replacing $\overline{a}$ by the negation $\neg a$, concatenation of events and '$\parallel$' by conjunction '$\wedge$', and the transition slash '$/$' by implication '$\supset$'. An empty trigger, an empty action, and configuration $\mathbf{0}$ are identified with *true*. Then, $M \models C$ if and only if $C$ holds for the intuitionistic Kripke structure $M$. Hence, for $M = (n, V)$, we are allowed to write $M \models a$, if $\forall 0 \leq i < n./; a \in V(i)$, as well as $M \models \neg b$, if $b \notin V(n-1)$. In the sequel, we abbreviate the set $\{M \mid M \models C\}$ of sequence models of $C$ by $SM(C)$. It will sometimes be useful to consider the sequence models $2SM(C)$ of $C$ of length at most two only, i.e., $2SM(C) =_{\mathrm{df}} \{(n, V) \mid (n, V) \in SM(C)$ and $n \leq 2\}$.

In our introductory example, configuration $C_{79}$ is behaviorally equivalent to $C'_{79} =_{\mathrm{df}} \overline{b}/a \parallel b/a$. The latter configuration may be identified with formula $(\neg b \supset a) \wedge (b \supset a)$ which states "*if $b$ is absent throughout a macro step or $b$ is present throughout a macro step, then $a$ is asserted.*" In classical logic, configuration $C'_{79}$ would be deemed equivalent to the single transition $C_{12} = \cdot/a$ corresponding to formula *true* $\supset a$. As mentioned before, this is inadequate as both do not have the same operational behavior, since $C'_{79} \parallel a/b$ fails whereas $C_{12} \parallel a/b$ has step response $\{a, b\}$ in the empty environment. In our intuitionistic semantics, the difference is faithfully witnessed by the 2–sequence $M = (2, V)$, where $V(0) =_{\mathrm{df}} \emptyset$ and $V(1) =_{\mathrm{df}} \{a, b\}$. Here, $M$ is a sequence model of $C'_{79}$ but not of $C_{12}$.

As another example, consider configurations $\overline{a}/a$ and $\cdot/a$ corresponding to formulas $\neg a \supset a$ and *true* $\supset a$, respectively. In classical logic both are equivalent. Yet, they differ in their operational behavior. The former configuration fails in the empty environment while the latter produces response $\{a\}$. In our intuitionistic semantics, however, both are distinguished: $\neg a \supset a$ specifies "*eventually $a$ must be present*," as $\overline{a}/a$ expects the environment to assert event $a$ in order to avoid failure. This is different from *true* $\supset a$ which specifies "*always $a$.*" Formally, formula $\neg a \supset a$ possesses two (irredundant) sequence models over the event set $\{a\}$: (i) 2–sequence $(2, V_1)$, where $V_1(0) =_{\mathrm{df}} \emptyset$ and $V_1(1) =_{\mathrm{df}} \{a\}$, as well as (ii) 1–sequence $(1, V_2)$, with $V_2(0) =_{\mathrm{df}} \{a\}$. However, according to Def. 3.1, $(2, V_1)$ is not a sequence model of

formula $true \supset a$. Finally, consider formula $(a \supset b) \wedge (b \supset a)$ which corresponds to configuration $a/b \parallel b/a$. This has exactly three (irredundant) sequence models over event set $\{a, b\}$: (i) 2–sequence $(2, W_1)$, where $W_1(0) =_{\mathrm{df}} \emptyset$ and $W_1(1) =_{\mathrm{df}} \{a, b\}$, (ii) 1–sequence $(1, W_2)$ with $W_2(0) =_{\mathrm{df}} \emptyset$, as well as (iii) 1–sequence $(1, W_3)$ with $W_3(0) =_{\mathrm{df}} \{a, b\}$. Hence, the environment has to provide at least one event, $a$ or $b$, for response $\{a, b\}$ to occur, i.e., the transitions $a/b$ and $b/a$ cannot mutually trigger each other, in accordance with the principle of causality [von der Beeck 1994].

Note that the classical semantics of Statecharts configurations is contained in our intuitionistic one by considering 1–sequences only. More precisely, every 1–sequence $M = (1, V)$ may be identified with a Boolean valuation $V' \in \Pi \to \mathbb{B}$ by taking $V'(a) = tt$ if $a \in V(0)$. Then, $M \models C$ if and only if $C$ classically evaluates to $tt$ under valuation $V'$. Moreover, it will be convenient to identify a 1–sequence $(1, V)$ with a subset of events, i.e., the set $V(0) \subseteq_{\mathrm{fin}} \Pi \setminus \{\bot\}$. Vice versa, a subset $A \subseteq_{\mathrm{fin}} \Pi \setminus \{\bot\}$ induces the 1–sequence $(1, V)$, for $V(0) =_{\mathrm{df}} A$. Every $n$–sequence also contains a distinguished classical structure, namely its final state. We refer to the final state of $M = (n, V)$ as $M^*$, i.e., $M^* = (1, V^*)$ where $a \in V^*(0)$ if and only if $a \in V(n-1)$; sometimes, $M^*$ is simply identified with the final state $V(n-1)$. Finally, we also employ the notation $M^i$, for $i < n$, to denote the suffix sequence of $M$ that starts at state $i$, i.e., $M^i =_{\mathrm{df}} (n-i, V^i)$ where $V^i(j) =_{\mathrm{df}} V(i+j)$. It is easy to show that, whenever $M \in SM(C)$, then $M^i \in SM(C)$, too.

PROPOSITION 3.2. *Let $C \in \mathcal{PC}$ and $M$ be an $n$–sequence. Then, $M \models C$ implies $M^i \models C$, for all $i < n$.*

As a consequence, one may always construct a model in $2SM(C)$ when given a model in $SM(C)$.

### 3.2 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 those 1–sequence models of $C$, i.e., subsets $A \subseteq_{\mathrm{fin}} \Pi \setminus \{\bot\}$, that do not occur as the final world of any other sequence model of $C$ except itself. 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. More precisely, if event set $A$ occurs as the final state of an $n$–sequence model $M$, where $n > 1$, then $M$ represents a proper interaction sequence of the considered configuration with its environment that *must* occur in order for $C$ to participate in response $A$. Hence, if there is no non–trivial $n$–sequence with $M^* = A$, then $C$ can produce $A$ as an autonomous response.

*Definition* 3.3 *Response Model.* Let $C \in \mathcal{PC}$. Then, $M = (1, V) \in SM(C)$ is a *response model* of $C$ if $K^* = M^*$ implies $K = M$, for all $K \in SM(C)$. The set of response models of $C$ is denoted by $RM(C)$.

Hence, response models of $C$ may be identified with specific classical models of $C$. Observe, however, that their definition involves essential reference to the intuitionistic semantics of configurations.

THEOREM 3.4 CHARACTERIZATION. *Let $C \in \mathcal{PC}$ and let $E, A \subseteq_{fin} \Pi$. Then, $C \Downarrow_E A$ iff $A$ is a response model of configuration $C \parallel \cdot/E$.*

PROOF. Let us start with a comment concerning our notation for transitions. In this and in the following proofs we will often identify a transition $P, \overline{N}/B$ with the intuitionistic formula $P \wedge \neg N \supset B$. More precisely, formulas $P$ and $B$ stand for the conjunctions of the events in sets $P$ and $B$, respectively, and formula $\neg N$ abbreviates the conjunction of the negations of all events in set $N$. This propositional notation reflects precisely our intuitionistic semantics of Def. 3.1. Since $C \Downarrow_E A$ if and only if $(C \parallel \cdot/E) \Downarrow A$, it suffices to show that $D \Downarrow A$ if and only if $A$ is a response model of $D$, for all $D \in \mathcal{PC}$ and $A \subseteq_{\text{fin}} \Pi$.

"$\Longrightarrow$". Let $D \Downarrow A$, and let $T$ be the set of admissible transitions generating response $A$; in particular, $\bot \notin A$. We show that $A$ is a response model of $D$. Let us first convince ourselves that $A$ is a model of $D$, i.e., $A \models D$. Recall that we identify $A$ with the stabilization sequence $(1, V)$, where $V(0) =_{\text{df}} A$. Let $t = P \wedge \neg N \supset B$ be a transition in $D$. If $A \not\models P \wedge \neg N$, then nothing needs to be shown because $A \models t$ trivially holds. So, suppose that $A \models P \wedge \neg N$, i.e., $P \subseteq A$ and $N \cap A = \emptyset$. Since $A$ is the set of events generated from $T$ and since $t$ is enabled by $A$, we conclude that $t$ must have fired, i.e., $t \in T$. This implies $B \subseteq A$. Thus, $A \models B$, which proves $A \models t$. Since $t$ was arbitrary, $A$ validates all (parallel) transitions of $D$, whence $A \models D$, as desired.

Next we show that $A$ is in fact a response model, i.e., there exists no non–classical irredundant extension of $A$ that is a model of $D$. Suppose $K = (n, V)$ is such an irredundant $n$–sequence model of $D$ with $K^* = A$ and $K \models D$. If $n = 1$, then $K = A$, and we are done. Otherwise, if $n \geq 2$, the sequence $K$ has at least two states; in particular, we must have $V(n-2) \subset A$. Sequence model $K$ has the following useful properties:

(1) $\forall b \in \Pi.\ A \models \neg b$ implies $K \models \neg b$, i.e., $A$, $K$ have the same negated truths.
(2) $\exists a \in \Pi.\ A \models a$ but $K \not\models a$.

Prop. (1) implies that $K$ satisfies the negative triggers of all transitions that have fired to produce $A$, since those are all valid in $A$ and, hence, must be valid in $K$. Now, we use the fact that if $T$ is the set of transitions— or, more precisely, their corresponding formulas— which have fired to produce $A$ and if $\neg R$ are the cumulated negative triggers, then $T \wedge \neg R \models A$ is a valid consequence in intuitionistic logic. This can be shown without difficulties as an auxiliary lemma, using essentially the deductive nature of the step semantics, e.g., by induction on the number of iterations of the step–construction procedure. Thus, (i) $T \wedge \neg R \models A$, (ii) $K \models T$, since it is a model of $D$, and (iii) $K \models \neg R$, whence $K \models A$. But this contradicts Prop. (2).

"$\Longleftarrow$". Suppose $A$ is a response model of $D$. We must prove $D \Downarrow A$. To this end, consider the set $T_A$ of all (parallel) transitions of $D$ that are enabled in $A$. We show that

(1) $T_A$ is an ($\emptyset$–)admissible set of transitions in $D$, and
(2) $\mathsf{act}(T_A) = A$.

Note that it is clear that $\bot \notin A$, as $A$ is a sequence model. Regarding Prop. (2), take any $t \in T_A$, say $t = P \wedge \neg N \supset B$. Since trigger $P \wedge \neg N$ of $t$ is valid in $A$ and since $A$ is a model of $D$, we must have $A \models B$, whence $B \subseteq A$. Thus, $\mathsf{act}(T_A) \subseteq A$. For the other inclusion, suppose there exists some $a \in A$ which does not appear as an action of any transition in $T_A$. We claim, then, that we can extend $A$ to an irredundant 2–sequence model $K$ of $D$ with $K^* = A$. To obtain such a $K$, take $K =_{\mathrm{df}} (2, V)$, where $V(1) =_{\mathrm{df}} A$ and $V(0) =_{\mathrm{df}} A \setminus \{a\}$. Now, we show that $K$ is a model of $D$. Take any transition $t$ of $D$, say $P \wedge \neg N \supset B$. For establishing $K \models t$, we follow the semantic definition of transitions (cf. Def. 3.1). Suppose $i \in \{0,1\}$, $V(1) \cap N = \emptyset$, and $P \subseteq V(i)$. We have to show that $B \subseteq V(i)$. Since $V(1) = A$ and $A \models t$, this follows immediately in case $i = 1$. So, consider $i = 0$. The assumptions $P \subseteq V(0) \subset V(1)$ and $V(1) \cap N = \emptyset$ mean that $t$ is enabled in $A = V(1)$, whence $t \in T_A$ by construction. But then $a \notin B$, since all events in $B$ are actions of $T_A$ and since $a$ does by assumption not appear as an action in $T_A$. Now, $a \notin B$ finally means $B = B \setminus \{a\} \subseteq A \setminus \{a\} = V(0)$. Hence, $B \subseteq V(0)$, as desired. This completes the proof that $K$ is a model of $t$, for arbitrary $t \in \mathsf{trans}(D)$, whence $K \models D$. Consequently, we have extended $A$ to an irredundant sequence model $K$ of $D$ of length 2, which contradicts the assumption that $A$ is a response model. Thus, $A \subseteq \mathsf{act}(T_A)$, and, putting our results together, $A = \mathsf{act}(T_A)$.

Regarding Prop. (1), it is not difficult to prove that $T_A = \mathsf{enabled}(D, \emptyset, T_A)$. Let $t \in T_A$. We claim that $t$ is enabled by the set of actions of $T_A$. Since, by Prop. (2), $A$ is exactly the set of all actions generated by $T_A$ and since $t$ is enabled in $A$, transition $t$ must be enabled by $T_A$. Hence, $T_A \subseteq \mathsf{enabled}(D, \emptyset, T_A)$. Vice versa, let $t$ be a transition of $D$ enabled in $T_A$, whence enabled in $A$. Then, $t \in T_A$ by definition. This proves the first part of admissibility. It remains to be shown that there exists some $t \in T_A \setminus T$ such that $t \in \mathsf{enabled}(D, \emptyset, T)$, for any $T \subset T_A$. Let $T \subset T_A$ be a proper subset of $T_A$. Consider the set $\mathsf{act}(T)$ of actions generated from $T$, which satisfies $\mathsf{act}(T) \subseteq \mathsf{act}(T_A) = A$ by Prop. (2). We distinguish two cases. First, if $\mathsf{act}(T) = A$, then by definition all transitions in $T_A$ are enabled by $\mathsf{act}(T)$. Thus, since $T_A \setminus T$ is non–empty, there exists at least one transition in $T_A$ outside of $T$ that is enabled by $T$. Second, assume $\mathsf{act}(T) \subset A$ is a proper subset. We then define the irredundant stabilization sequence $K =_{\mathrm{df}} (2, V)$ as a model extension of $A$, such that $V(0) =_{\mathrm{df}} \mathsf{act}(T)$ and $V(1) =_{\mathrm{df}} A$. Since $A = K^*$ is a response model by assumption, $K$ cannot be a model of $D$. Thus, there exists some transition $t$, say $P \wedge \neg N \supset B$, in $D$ such that $K \not\models t$. By the semantic definition for transitions (cf. Def. 3.1) this means that there exists an $i \in \{0,1\}$ such that (i) $P \subseteq V(i)$, (ii) $V(1) \cap N = \emptyset$, and (iii) $B \not\subseteq V(i)$. Since $P \subseteq V(i) \subseteq V(1)$ and $V(1) \cap N = \emptyset$, transition $t$ is enabled in $A = V(1)$. Thus, $t \in T_A$. The remaining fact $B \not\subseteq V(i)$ implies $t \notin T$; otherwise, if $t \in T$ then $B \subseteq \mathsf{act}(T) = V(0)$, which contradicts $B \not\subseteq V(i)$, since $V(0) \subseteq V(i)$, for any $i$. Hence, $t \in T_A \setminus T$ and $t \in \mathsf{enabled}(D, \emptyset, T)$, as desired.

This completes the proof of Thm. 3.4. □

Thm. 3.4 provides a simple model–theoretic characterization of step responses. For example, recall that configuration $\overline{a}/a$ forces Pnueli and Shalev's step construction procedure to fail. As shown before, the only sequence model of $\overline{a}/a$ of length 1 and using only event $a$ is $(1, V_2)$ with $V_2(0) = \{a\}$. But $(1, V_2)$ is not a response model

since it is the final world of 2–sequence model $(2, V_1)$ with $V_1(0) = \emptyset$ and $V_1(1) = \{a\}$. Since $\neg a \supset a$ does not have any response model, transition $\overline{a}/a$ can only fail in the empty environment. As another example, re–visit configuration $a/b \parallel b/a$, for which just $(1, W_2)$ with $W_2(0) = \emptyset$ is a response model. Thus, $(a/b \parallel b/a) \Downarrow \emptyset$ is the only response in environment $\emptyset$.

### 3.3 Full–abstraction Result for Parallel Configurations

Sequence models are not only elegant for characterizing Pnueli and Shalev's semantics, but also lead to a fully–abstract semantics for parallel configurations within parallel contexts. The following full–abstraction theorem states that the response behavior of a parallel configuration in arbitrary parallel contexts is a function of its sequence models, or indeed of its 1– and 2–sequence models. We will elaborate on this in Sec. 3.4, where we give an explicit construction for this function.

THEOREM 3.5 FULL ABSTRACTION. *For all $C, D \in \mathcal{PC}$, the following statements are equivalent:*

*(1)* $SM(C) = SM(D)$ .

*(2)* $2SM(C) = 2SM(D)$ .

*(3)* $(C \parallel R) \Downarrow_E A$ *iff* $(D \parallel R) \Downarrow_E A$, *for all* $R \in \mathcal{PC}$ *and* $E, A \subseteq_{fin} \Pi$ .

*(4)* $RM(C \parallel R) = RM(D \parallel R)$, *for all* $R \in \mathcal{PC}$ .

Hence, sequence models contain precisely the information needed to capture all possible interactions of a parallel configuration within all potential environments. To prove Thm. 3.5, we first establish an auxiliary lemma to show that the set of sequence models of at most length two contains the same information as the set of sequence models of arbitrary length.

LEMMA 3.6. *Let $C, D \in \mathcal{PC}$, and let $K$ be a stabilization sequence of arbitrary length such that $K \models C$, $K \not\models D$, and $K^* \models D$. Then, there exists a 2–sequence $M$ with $M \models C$, $M \not\models D$, and $M^* = K^*$.*

PROOF. Let parallel configurations $C$ and $D$ and $n$–sequence $K = (n, W)$ be given as stated in the lemma. Clearly, $n \geq 2$, as $K = K^*$ would be inconsistent with the assumptions $K \not\models D$ and $K^* \models D$. Now, let $0 \leq l \leq n-2$ be the largest $l$ such that $K^l \not\models D$ and $K^{l+1} \models D$. Consider the 2–sequence model $M =_{\text{df}} (2, V)$ where $V(0) =_{\text{df}} W(l)$ and $V(1) =_{\text{df}} W(n-1)$, i.e., $M$ consists of the first and the last state of $K^l$. Obviously, $M^* = K^*$. We will show that $M \models C$ but $M \not\models D$. We first prove that for every transition $t \in \mathsf{trans}(C) \cup \mathsf{trans}(D)$, say $P \wedge \neg N \supset A$,

$$K^l \models t \quad \text{if and only if} \quad M \models t. \tag{1}$$

From this our claim follows because (i) parallel configurations $C$ and $D$ are conjunctions of transitions, (ii) $K^l \models C$ by Prop. 3.2 and because $K \models C$, and (iii) $K^l \not\models D$. Since $K^* = W(n-1) = V(1) = M^*$ we immediately have

$$K^* \models t \text{ if and only if } M^* \models t \quad \text{and} \quad M \models \neg N \text{ if and only if } K^l \models \neg N\,.$$

By construction, $W(l) = V(0)$, whence they force the same events, e.g., for $P$:

$$P \subseteq W(l) \text{ if and only if } P \subseteq V(0) \quad \text{and} \quad A \subseteq W(l) \text{ if and only if } A \subseteq V(0)\,.$$

Taking all this together implies Statement (1). □

Proof of Theorem 3.5. We begin with the equivalence of the first two statements. It is obvious that $SM(C) = SM(D)$ implies $2SM(C) = 2SM(D)$, as 1– and 2–sequence models are just special sequence models. For the other direction, assume w.l.o.g. that $SM(C) \not\subseteq SM(D)$. Hence, there must exist a stabilization sequence $K$ such that $K \models C$ and $K \not\models D$. In the case $K^* \not\models D$, we obtain $2SM(C) \not\subseteq 2SM(D)$ since $K^* \models C$ and since $K^*$ is a classical structure. In the case $K^* \models D$, we apply Lemma 3.6 which yields a 2–sequence model $M$ satisfying $M \models C$ and $M \not\models D$. Thus, $2SM(C) \not\subseteq 2SM(D)$, too.

The equivalence of Statements (3) and (4) can be established as follows.

"$\Longrightarrow$". Let $R \in \mathcal{PC}$ and $A \subseteq_{\text{fin}} \Pi$ such that $A \in RM(C \parallel R)$. Since the empty transition $\cdot/\cdot$ is logically equivalent to *true*, this yields $A \in RM((C \parallel R) \parallel \cdot/\cdot)$. By Thm. 3.4, the latter implies $(C \parallel R) \Downarrow A$. According to Statement (3), we also have $(D \parallel R) \Downarrow A$ and may then apply the above steps in reverse order to obtain $A \in RM((D \parallel R))$. Hence, $RM(C \parallel R) \subseteq RM(D \parallel R)$. The other inclusion is proved similarly.

"$\Longleftarrow$". Let $R \in \mathcal{PC}$ and $E, A \subseteq_{\text{fin}} \Pi$ such that $(C \parallel R) \Downarrow_E A$. Due to Thm. 3.4, $A \in RM((C \parallel R) \parallel \cdot/E)$ holds. Then, according to Statement (4), $A$ is also a response model of $(D \parallel R) \parallel \cdot/E$. Hence, one arrives at $(D \parallel R) \Downarrow_E A$ when applying Thm. 3.4 again. The reverse direction of Statement (3) is proved analogously.

It remains to establish the equivalence of Statements (2) and (4).

"$\Longrightarrow$". Suppose $2SM(C) = 2SM(D)$ and $R \in \mathcal{PC}$. Then, $A \in RM(C \parallel R)$ implies $A \models C$ and $A \models R$. Since $A$ is a classical sequence model of $C$, it must be a sequence model of $D$ and, hence, of $D \parallel R$. We claim that $A$ actually is a response model of $D \parallel R$. Suppose it was not. Then, there would exist an irredundant sequence model $K = (n, V)$ of $D \parallel R$ satisfying $n \geq 2$ and $K^* = V(n-1) = A$. Since $K$ is irredundant, it contains the 2–sequence $M = (2, W)$, where $W(0) =_{\text{df}} V(n-2)$ and $W(1) =_{\text{df}} V(n-1)$. By Prop. 3.2, $K \models D \parallel R$ implies $M \models D \parallel R$. Hence, there exists a 2–sequence model $M$ with $M^* = A$ and $M \models D \parallel R$. Since $2SM(C) = 2SM(D)$, this implies $M \models C \parallel R$, contradicting the assumption that $A$ is a response model of $C \parallel R$.

"$\Longleftarrow$". This proof direction needs slightly more work as it involves the construction of a discriminating context. We start off with assumption $2SM(C) \neq 2SM(D)$. W.l.o.g., let $M$ be an irredundant stabilization sequence of length one or two such that $M \models C$ and $M \not\models D$. Moreover, define $A =_{\text{df}} M^*$. We now distinguish two cases.

(1) $\underline{A \not\models D.}$ Consider the context

$$R =_{\text{df}} \parallel \{L(0)/A \mid (n, L) \in 2SM(C) \text{ and } L^* = A\}\,,$$

where we extend the binary operator $\parallel$ for an arbitrary, but finite number of arguments. Observe that $R$ is a parallel composition of *finitely* many transitions

as $A$ is finite. Moreover, $R$ is non–empty since $M(0)/A$ is a transition in $R$. It is immediate that $A$ cannot be a response model of $D \parallel R$ because, by assumption, it is not even a model of $D$. We are done if we can show that $A \in RM(C \parallel R)$. Since every transition of $R$ is of the form $L(0)/A$, we have $A \models R$. Also, $A \models C$ holds because $M \models C$ and $A = M^*$. Hence, $A \models C \parallel R$. Moreover, it is not difficult to show that there cannot exist a 2–sequence $K$ such that $K^* = A$ and $K \models C \parallel R$. If such $K$ would exist, it would have to satisfy $K(0) \subset A$ and $K \models C$. Hence, by construction, transition $K(0)/A$ is a parallel component of $R$. This means $K \not\models R$, since $K \not\models K(0)/A$, which follows from $K(0) \subseteq K(0)$ and $A \not\subseteq K(0)$. But $K \not\models R$ would be a contradiction to $K \models C \parallel R$. Since, by Prop. 3.2, every nontrivial sequence model of $C \parallel R$ contains a 2–sequence model, this shows that there exists no proper weakening $K$ of $A$ that is still a model of $C \parallel R$. Thus, $A$ is a response model of $C \parallel R$.

(2) $A \models D$. In this case, we construct a configuration $R$ such that $A$ is a response model of $D \parallel R$ but not of $C \parallel R$. Consider an arbitrary sequence $K$. We define transitions $t_K^M$ as follows; recall that $M$ is a 2–sequence, whence $M(1) = M^* = A$:

$$t_K^M \;=_{\mathrm{df}} \begin{cases} K(0)/M(0) & \text{if } K(0) \subseteq M(0) \\ K(0)/M(1) & \text{otherwise.} \end{cases}$$

Again the sets $K(0)$, $M(0)$, and $M(1)$ are finite. These transitions have the property that $M \models t_K^M$, for all sequences $K$, and $K \not\models t_K^M$, for all sequences $K$ such that $K(0) \neq M(0)$, $K(0) \neq M(1)$, and $K^* = M(1) = A$. The context configuration $R$ is now formed as

$$R \;=_{\mathrm{df}} \parallel \{t_L^M \mid L \in 2SM(D) \text{ and } L^* = A\}.$$

As before, there is only a finite number of $L$ with $L^* = A$, as $A$ is finite. It follows from the above that $M \models R$ and also $A \models R$. Now we compare the response models of $C \parallel R$ and $D \parallel R$. Obviously, $A \notin RM(C \parallel R)$, since $M$ is irredundant with $M^* = A$, and also $M \models C$ and $M \models R$, whence $M \models C \parallel R$. We claim that $A \in RM(D \parallel R)$. First of all, $A \models D \parallel R$. Now suppose there exists an irredundant stabilization sequence $K$ such that $K^* = A$ and $K \models D \parallel R$. We may assume that $K$ has length 2 according to Prop. 3.2. By construction, $R$ then contains transition $t_K^M$, whence $K \models t_K^M$. However, this is impossible unless $K(0) = M(0)$ or $K(0) = M(1)$. If, however, $K(0) = M(0)$, then $K \not\models D$. This follows from $K^* = M(1) = A$ and the assumption $M \not\models D$, as one can show without difficulties. So, we must have $K(0) = M(1)$. Since $K^* = A = M(1)$ and since $K$ is irredundant, we conclude $K = A$. Thus, there cannot exist a non–trivial weakening of $A$ that is a model of $D \parallel R$. Hence, $A \in RM(D \parallel R)$, as desired.

This completes the proof of Thm. 3.5. □

### 3.4 Characterization of Sequence Models

Thm. 3.5 does not mean that every set of stabilization sequences can be obtained from a parallel configuration. In fact, from the model theory of intuitionistic logic it is known that in order to specify arbitrary linear sequences, nested implications are needed [van Dalen 1986]. Statecharts configurations, however, only use first–order

implications and negations. Therefore, we may expect the semantics of configurations to satisfy additional structural properties. In fact, it turns out that the sets $SM(C)$ are closed under *sub–sequences*, *refinement*, and *sequential composition*. These notions are defined as follows:

—The $m$–sequence $M = (m, V)$ is a *sub–sequence* of the $n$–sequence $N = (n, W)$, written $M \preceq N$, if there exists a mapping $f : [0, \ldots, m-1] \to [0, \ldots, n-1]$ such that $V(i) = W(f(i))$, for all $0 \leq i < m$, and $V(m-1) = W(n-1)$. Note that $f$ must be strictly monotonic since $V$ and $W$ are strictly increasing. In other words, $M \preceq N$ holds if $M$ is obtained from $N$ by dropping some states while preserving the final state.

—The $k$–sequence $K = (k, U)$ is a *refinement* of the $m$–sequence $M = (m, V)$ and the $n$–sequence $N = (n, W)$, written $K \preceq M \sqcap N$, if there exist mappings $f_M : [0, \ldots, k-1] \to [0, \ldots, m-1]$ and $f_N : [0, \ldots, k-1] \to [0, \ldots, n-1]$ such that $U(k-1) = V(m-1) = W(n-1)$ and $U(i) = V(f_M(i)) \cap W(f_N(i))$, for $i < k$. Intuitively, $K \preceq M \sqcap N$ holds if $M$, $N$, and $K$ have the same final state and if every state of $K$ arises from the intersection of a state from $M$ with one from $N$.

—Finally, the *sequential composition* of $M = (m, V)$ and $N = (n, W)$, such that $V(m-1) \subset W(0)$, is the sequence $M \,;\, N = (m+n, U)$ where $U(i) = V(i)$, for $0 \leq i < m$, and $U(i) = W(i-m)$, otherwise. Simply speaking, $M \,;\, N$ is the concatenation of sequence $M$ followed by sequence $N$.

One can easily verify that the set $SM(C)$, for every parallel configuration $C \in \mathcal{PC}$, is closed under sub–sequences, refinement, and sequential composition. In the finite case, the converse is also valid, i.e., every finite set of stabilization sequences which is closed under sub–sequences, refinement, and sequential composition is the set of sequence models of some parallel configuration, relative to some fixed finite set of events. However, instead of working with sets of sequence models, we will present an equivalent characterization that is much more compact and that employs simple finite lattice structures which we refer to as *behaviors*.

*Definition* 3.7 *Behavior.* A *behavior* $\mathcal{C}$ is a pair $\langle F, I \rangle$, where $F \subseteq 2^{\Pi \setminus \{\perp\}}$ and $I$ is a function that maps every $B \in F$ to a set $I(B) \subseteq 2^B$ of subsets of $B$, such that

(1) $I$ is monotonic, i.e., $B_1 \subseteq B_2$ implies $I(B_1) \subseteq I(B_2)$, for any $B_1, B_2 \in F$.

(2) $I(B)$ is closed under intersection, i.e., $B_1, B_2 \in I(B)$ implies $B_1 \cap B_2 \in I(B)$.

(3) $B \in I(B)$, for all $B \in F$.

If $F = \{A\}$, for some $A \subseteq_{\text{fin}} \Pi$, then $\mathcal{C}$ is called *$A$–bounded*, or simply *bounded* if $A$ is understood. Moreover, $\mathcal{C}$ is *directed* if $F \neq \emptyset$ and $\forall B_1, B_2 \in F \;\; \exists B \in F.\; B_1 \subseteq B$ and $B_2 \subseteq B$.

Intuitively speaking, the first component $F$ of a behavior $\mathcal{C} = \langle F, I \rangle$ captures the possible final responses of $\mathcal{C}$. For every such final response $B \in F$, the event sets $I(B) \subseteq 2^B$ represent the states of all stabilization sequences of $C$ that end in $B$. Any strictly increasing sequence that moves only through states $I(B)$ and ends in $B$ is considered a stabilization sequence of the behavior. In case $I(B) = \{B\}$ set $B$ is an autonomous response of $\mathcal{C}$. This interpretation is confirmed below in Lemma 3.9 for those behaviors that are obtained from parallel Statecharts configurations.

It is not difficult to show that the pairs of initial and final states occurring together in the sequence models of $C \in \mathcal{PC}$ induce a behavior. More precisely, the *induced* behavior $Beh(C)$ of $C$ is the pair $\langle F(C), I(C)\rangle$ which is defined as follows:

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

From the property of sub–sequence closure we know that the initial and final states of any sequence model of $C$ form a 2–sequence model of $C$. Thus, we can also define behavior $\langle F(C), I(C)\rangle$ directly from $2SM(C)$:

$$X \in F(C) \quad \text{if and only if} \quad X \in 2SM(C), \text{ and}$$
$$X \in I(C)(Y) \quad \text{if and only if} \quad (X,Y) \in 2SM(C) \text{ or } X = Y \in SM(C),$$

where we identify a 1–sequence $(1,V)$ with the subset $V(0)$ and a 2–sequence $(2,V)$ with the pair $(V(0),V(1))$. From our construction it is clear that $Beh(C)$ is uniquely determined by $SM(C)$ or, in fact, by $2SM(C)$.

LEMMA 3.8. *For $C \in \mathcal{PC}$, $Beh(C)$ is a behavior and, if $\perp$ does not occur in $C$, then $Beh(C)$ is directed.*

PROOF. Observe that, for all stabilization sequences $(n,V)$, we have $\perp \notin V(n-1)$ by definition. Hence, $F(C) \subseteq 2^{\Pi\setminus\{\perp\}}$.

First, we show that $I(C)$ is monotonic. Let $B_1, B_2 \in F(C)$ such that $B_1 \subseteq B_2$, and let $E \in I(C)(B_1)$. If $B_1 = B_2$ nothing needs to be shown, as $E \in I(C)(B_2)$ trivially holds. So, suppose $B_1 \subset B_2$, whence, for some $(n,V) \in SM(C)$, both $V(0) = E$ and $V(n-1) = B_1$ hold. We claim that the sequence $(n+1,W)$ defined by $W(i) =_{\mathrm{df}} V(i)$, for $0 \le i < n$, and $W(n) =_{\mathrm{df}} B_2$ is a model of $C$, which then entails $E \in I(C)(B_2)$. To prove $(n+1,W) \in SM(C)$ we proceed by contradiction. Assume that there exists a transition $t$, say $P \wedge \neg N \supset D$, of $C$ such that $(n+1,W) \not\models t$. This implies that there must exist some $0 \le i \le n$ such that $P \subseteq W(i)$, $N \cap W(n) = \emptyset$, and $D \not\subseteq W(i)$. Since $B_2 \in F(C)$, set $B_2$ is the final state of a sequence model of $C$. Thus, by the properties of intuitionistic truth, the singleton sequence $B_2$ must be a model of $C$, too. This means that the final state $W(n) = B_2$ of $W$ must satisfy $t$, i.e., $D \subseteq W(n)$. Hence, $0 \le i < n$ and $W(i) = V(i)$. Now, $N \cap B_2 = N \cap W(n) = \emptyset$ and $B_1 \subseteq B_2$ implies $N \cap V(n-1) = N \cap B_1 = \emptyset$. From this we conclude $(n,V) \not\models t$ which contradicts assumption $(n,V) \in SM(C)$. Hence, we have $(n+1,W) \in SM(C)$ and, as a consequence, $W(0) = V(0) = E$ and $W(n) = B_2$, i.e., $E \in I(C)(B_2)$. This establishes that $I(C)$ is monotonic.

Second, we verify that $I(C)(B)$ is intersection closed, for all $B \in F(C)$. Let $E_1, E_2 \in I(C)(B)$ and sequences $(n_1,V_1) \in SM(C)$ and $(n_2,V_2) \in SM(C)$ such that $V_1(n_1-1) = V_2(n_2-1) = B$, $E_1 = V_1(0)$, and $E_2 = V_2(0)$. Consider the 2–sequence $(2,U)$, where $U(0) =_{\mathrm{df}} E_1 \cap E_2$ and $U(1) =_{\mathrm{df}} B$. We claim that $(2,U) \in SM(C)$. Suppose, by way of contradiction, that $t$ is a transition of $C$, say $P \wedge \neg N \supset D$, for which $(2,U) \not\models t$. Since $B = U(1)$ and $B \in F(C)$, i.e., $B$ is a singleton model of $C$, we know that $U(1) \models t$. Hence, any violation of $t$ by $(2,U)$ can only occur if $P \subseteq U(0)$, $N \cap U(1) = N \cap B = \emptyset$, and $D \not\subseteq U(0)$. Since $U(0) = E_1 \cap E_2$ it follows that $P \subseteq E_1$ and $P \subseteq E_2$. Furthermore, $D \not\subseteq U(0)$ implies $D \not\subseteq E_i$, for $i = 1$ or $i = 2$. In either case, the fact that $N \cap B = \emptyset$, as $B$ is the final state of $(n_i,V_i)$ for both $i \in \{1,2\}$, implies $(n_i,V_i) \not\models t$ which

contradicts our assumption. Thus, $(2, U) \in SM(C)$, as desired. By construction we have $U(0) = E_1 \cap E_2$ and $U(1) = B$, whence $E_1 \cap E_2 \in I(C)(B)$. This completes the proof that $I(C)(B)$ is intersection closed.

Third, observe that $B \in I(C)(B)$ holds, too, since $B \in SM(C)$, for all $B \in F(C)$.

Finally, we show that $Beh(C)$ is directed if failure event $\perp$ does not occur in $C$. Let $E_1, E_2 \in F(C)$, i.e., $V_1(n_1 - 1) = E_1$ and $V_2(n_2 - 1) = E_2$, for some sequence models $(n_1, V_1), (n_2, V_2) \in SM(C)$. Now, consider the 1–sequence $(1, V)$, where $V(0) =_{\mathrm{df}} E_1 \cup E_2 \cup \mathsf{act}(\mathsf{triggered}(C, E_1 \cup E_2))$, *i.e.*, $E_1 \subseteq V(0)$ and $E_2 \subseteq V(0)$. Note that $V(0) \subseteq \Pi \setminus \{\perp\}$ and that $\perp$ is by assumption not included in any action. Hence, $(1, V)$ is a stabilization sequence. Moreover, $(1, V)$ is clearly a model of each transition of $C$ and, thus, of $C$. This implies $(1, V) \in SM(C)$ and, further, $V(0) \in F(C)$. It can also be seen that $\mathsf{act}(\mathsf{trans}(C)) \subseteq \Pi \setminus \{\perp\}$ is a classical model of $C$, whence $F \neq \emptyset$. Thus, $Beh(C)$ is directed, which finishes the proof. □

The relationship between $Beh(C)$ and $SM(C)$ can be further clarified as follows.

LEMMA 3.9. *Let $C \in \mathcal{PC}$ be a parallel configuration.*

(1) *For every stabilization sequence $(n, V)$, we have $(n, V) \in SM(C)$ iff $V(n-1) \in F(C)$ and $V(i) \in I(C)(V(n-1))$, for all $0 \leq i < n$.*

(2) *$B \in RM(C)$ iff $B \in F(C)$ and $I(C)(B) = \{B\}$.*

According to Part (1), a stabilization sequence $M$ is an element of $SM(C)$ if and only if it is a sequence of states from $I(C)(B)$ such that $B \in F(C)$ and $B$ is the final state of $M$. This implies that not only is $Beh(C)$ uniquely determined by $SM(C)$ but also, vice versa, $SM(C)$ is uniquely determined by $Beh(C)$.

PROOF OF LEMMA 3.9. Part (2) follows directly from the definition of $Beh(C)$ and $RM(C)$. In addition, direction "$\Longrightarrow$" of Part (1) is trivial as it follows from the definition of $Beh(C)$ and from Prop. 3.2. To obtain the reverse direction of Part (1), we assume that $V(n-1) \in F(C)$ and $V(i) \in I(C)(V(n-1))$, for all $0 \leq i < n$. Now, suppose $(n, V) \notin SM(C)$, i.e., there exists a transition $t$, say $P \wedge \neg N \supset A$, of $C$ satisfying $(n, V) \not\models t$. Let $0 \leq i < n$ be some index with $P \subseteq V(i)$, $N \cap V(n-1) = \emptyset$, and $A \not\subseteq V(i)$. Note that such an $i$ must exist since $(n, V)$ refutes $t$. From the assumption $V(i) \in I(C)(V(n-1))$ we infer the existence of a stabilization sequence $(m, W) \in SM(C)$ with $W(0) = V(i)$ and $W(m-1) = V(n-1)$. But this implies $P \subseteq W(0)$, $N \cap W(m-1) = \emptyset$, and $A \not\subseteq W(0)$, which means $(m, W) \not\models t$ in contradiction to $(m, W) \in SM(C)$. Hence, $(n, V) \in SM(C)$, as desired. □

As a consequence of Lemma 3.9, we obtain that $Beh(C)$ contains the same semantic information as $SM(C)$.

THEOREM 3.10 CHARACTERIZATION. *Let $C, D \in \mathcal{PC}$. Then, $Beh(C) = Beh(D)$ iff $SM(C) = SM(D)$.*

PROOF. Direction "$\Longleftarrow$" follows immediately from the fact that the behavior of a configuration is derived from its sequence models. The other direction "$\Longrightarrow$" is an implication of Lemma 3.9(1). □

In conjunction with Thm. 3.5, we conclude that equivalence in arbitrary parallel contexts can equally well be decided by behaviors: $Beh(C) = Beh(D)$ if and only if $(C \parallel R) \Downarrow_E A$ is equivalent to $(D \parallel R) \Downarrow_E A$, for all $R \in \mathcal{PC}$ and $E, A \subseteq_{\text{fin}} \Pi$. The advantage of $Beh(C)$ over $SM(C)$ is that the former provides an *irredundant* representation of parallel configurations. Moreover, as we will show next, every "finite" behavior can be represented by a parallel configuration. We call a behavior $\mathcal{C} = \langle F, I\rangle$ *$A$–finite*, for $A \subseteq_{\text{fin}} \Pi$, if $\mathcal{C}$ is uniquely determined by the subsets of $A$, i.e., $B \in F$ if and only if $B \cap A \in F$, and $X \in I(B)$ if and only if $X \cap A \in I(B \cap A)$. If $\mathcal{C}$ is $A$–finite, then the *$A$–restriction* $\mathcal{C}|_A =_{\text{df}} \langle F|_A, I|_A\rangle$, such that $F|_A =_{\text{df}} F \cap 2^A$ and $I|_A(B) = I(B)$, is finite and contains complete information about $\mathcal{C}$. For representation purposes it is convenient to confuse an $A$–finite behavior $\mathcal{C}$ with its finite restriction $\mathcal{C}|_A$. In a similar vein, we identify an $A$–bounded behavior $\mathcal{D} = \langle\{A\}, I\rangle$ with the $A$–finite behavior generated by it, i.e., the uniquely defined behavior $\mathcal{C}$ such that $\mathcal{C}|_A = \mathcal{D}$. We frequently use these implicit restrictions and extensions in our examples without further mention. The exactness of behaviors as models of configurations is now an implication of the following theorem.

THEOREM 3.11 COMPLETENESS. *$\mathcal{C}$ is an $A$–finite (directed) behavior iff there exists a configuration $C \in \mathcal{PC}$ over events $A$ (not using failure event $\perp$) such that $\mathcal{C} = Beh(C)$.*

PROOF. Direction "$\Longleftarrow$" of Thm. 3.11 is essentially the statement of Lemma 3.8. $A$–finiteness is a trivial consequence of the fact that the semantics of a configuration only depends on the events mentioned in it. We may thus focus on direction "$\Longrightarrow$".

Let $\mathcal{C} = \langle F, I\rangle$ be an $A$–finite behavior. We are going to construct a configuration $C$ over events $A$ such that $Beh(C) = \mathcal{C}$. Since $Beh(C) = \langle F(C), I(C)\rangle$ is also $A$–finite, we can prove $Beh(C) = \mathcal{C}$ simply by establishing that their $A$–restrictions are identical. Thus, we only need to consider subsets of $A$, i.e., prove $F \cap 2^A = F(C) \cap 2^A$ and then $I(Y) = I(C)(Y)$ under the additional assumption that $Y \subseteq A$. Moreover, depending on whether $\mathcal{C}$ is directed or not, we can make further assumptions about $A$. First, if $\mathcal{C}$ is non-directed, then we assume that $\perp \in A$. This is permitted since $A$–finiteness is not affected by adding $\perp$ to $A$. Alternatively, if $\mathcal{C}$ is directed, then we may assume $A \in F$. Since $F \neq \emptyset$, the set $F \cap 2^A$ must be non–empty, too, and by directedness must contain a greatest element $A^\top \in F \cap 2^A$. Then, $\mathcal{C}$ is also $A^\top$–finite. Thus, if $A \notin F$, we may use $A^\top$ instead of $A$.

Our construction of configuration $C$ uses the following uniform construction of transitions. We associate with every $E \subseteq B \subseteq A$, such that $B \in F$, an event set $(E, B)^\top \subseteq \Pi$ defined by

$$(E, B)^\top =_{\text{df}} \bigcap \{E' \in I(B) \mid E \subseteq E' \subseteq B\}\,.$$

Note that this intersection is always non–empty since $E \subseteq B \subseteq B$ and $B \in I(B)$, by Prop. (3) of behaviors (cf. Def. 3.7). Intuitively, $(E, B)^\top$ is the "best upper approximation" of stabilization sequence $(2, V)$, where $V(0) =_{\text{df}} E$ and $V(1) =_{\text{df}} B$, in $\mathcal{C}$. By construction and by Prop. (2) of behaviors,

$$E \subseteq (E, B)^\top \subseteq B \quad \text{ as well as } \quad (E, B)^\top \in I(B)\,.$$

Note that the left–hand inclusion $E \subseteq (E,B)^\top$ becomes an equality $(E,B)^\top = E$ precisely if $E \in I(B)$. Now, we define a configuration $C \in \mathcal{PC}$ from $\mathcal{C}$ as follows:

$$\begin{aligned} C \;=_{\mathrm{df}}\; & \parallel \{(E \cup (\overline{A \setminus B}))/(E,B)^\top \mid E \subseteq B \subseteq A \text{ and } B \in F\} \\ & \parallel \{(B \cup (\overline{A \setminus B}))/(A \setminus B) \mid B \subseteq A \text{ and } B \notin F\}\,. \end{aligned}$$

This is a finite configuration since all sets involved are finite and subsets of $A$. Observe that if $A$ does not contain $\perp$, then configuration $C$ does not use $\perp$ either. Hence, if $\mathcal{C}$ is directed, then $C$ is $\perp$–free, since by our assumptions $A \in F$ holds which implies $\perp \notin A$. On the other hand, if $\mathcal{C}$ is non–directed, our assumption $\perp \in A$ has the effect that configuration $C$ actually uses event $\perp$ in its transitions.

The claim now is that $Beh(C) = \mathcal{C}$, i.e., $\langle F(C), I(C)\rangle = \langle F, I\rangle$. As discussed above, by $A$–finiteness, we can restrict ourselves to subsets of $A$. Moreover, whenever stabilization sequences occur, it suffices by Lemma 3.6 to consider those of at most length two. For convenience, a sequence $(1,V)$ is identified with the redundant sequence $(2,W)$, where $W(0) =_{\mathrm{df}} W(1) =_{\mathrm{df}} V(0)$. For stabilization sequences $(2,V)$, we also write $(V(0),V(1))$.

*(1)* We first show $F(C) \cap 2^A = F \cap 2^A$ and start with $F \cap 2^A \subseteq F(C) \cap 2^A$. Suppose $Y \subseteq A$ is such that $Y \notin F(C)$, i.e., there exists no $X \subseteq Y$ satisfying $(X,Y) \in \mathit{2SM}(C)$. In particular, $(Y,Y) \notin \mathit{2SM}(C)$. Hence, there is a transition $t$ in $C$ which is falsified by $(Y,Y)$. If $t = (B \cup (\overline{A \setminus B}))/(A \setminus B)$, for some $B \subseteq A$ and $B \notin F$, we must have $Y = B$, whence $Y \notin F$. In case $t = (E \cup (\overline{A \setminus B}))/(E,B)^\top$, for some $E \subseteq B \subseteq A$ and $B \in F$, we obtain $E \subseteq Y$ and $Y \cap (A \setminus B) = \emptyset$ and $(E,B)^\top \not\subseteq Y$. The second property $Y \cap (A \setminus B) = \emptyset$ is equivalent to $Y \subseteq B$. Thus, together with the first property, we have $E \subseteq Y \subseteq B$. Now, suppose $Y \in F$. By Prop. (1) of behaviors (monotonicity), $I(Y) \subseteq I(B)$. This implies $(E,B)^\top \subseteq (E,Y)^\top \subseteq Y$ which would contradict the third property $(E,B)^\top \not\subseteq Y$. Hence, $Y \notin F$, as desired. This proves the inclusion $F \cap 2^A \subseteq F(C) \cap 2^A$.

For the inclusion $F(C) \cap 2^A \subseteq F \cap 2^A$, let $Y \in F(C)$ and $Y \subseteq A$. In particular, we must have $Y \subset A$ since $\perp \in A$ by our assumption and $\perp \notin Y$ since none of the elements of $F(C)$ contains the failure event. In addition, $Y \in F(C)$ means $(Y,Y) \in \mathit{2SM}(C)$ by definition. Now, if $Y \notin F$ would hold, then $C$ would contain transition $t = (Y \cup (\overline{A \setminus Y}))/(A \setminus Y)$. However, since $Y \subset A$, we have $(Y,Y) \not\models t$, contradicting $(Y,Y) \in \mathit{2SM}(C)$. Hence, $Y \in F$ and $F(C) \cap 2^A \subseteq F \cap 2^A$.

*(2)* We show $I(C)(Y) = I(Y)$, for all $Y \in F \cap 2^A = F(C) \cap 2^A$. Fix any $Y \in F \cap 2^A$. We first prove the inclusion $I(Y) \subseteq I(C)(Y)$. To this end, let $X \in I(Y)$ be given. We claim that $(X,Y) \in \mathit{2SM}(C)$ which implies $X \in I(C)(Y)$. In order to show that $(X,Y)$ is a 2–sequence model of $C$ it will be convenient to use indices to refer to the states $X$ and $Y$ of this sequence and to use the notation $(V(0),V(1)) =_{\mathrm{df}} (X,Y)$. Now, consider any transition $t = (E \cup (\overline{A \setminus B}))/(E,B)^\top$, where $E \subseteq B \subseteq A$ and $B \in F$. We check that $(V(0),V(1)) \models t$ following the definition of our semantics. If $V(1) \cap (A \setminus B) \neq \emptyset$ or, for no $i \in \{1,2\}$, $E \subseteq V(i)$, then we are done immediately. So assume $V(1) \cap (A \setminus B) = \emptyset$ which is the same as $V(1) \subseteq B$, and choose any $i \in \{0,1\}$ such that $E \subseteq V(i)$. Hence, we have $E \subseteq V(i) \subseteq V(1) \subseteq B$. By Prop. (1) of behaviors, $Y = V(1) \subseteq B$ implies $I(Y) \subseteq I(B)$. Furthermore, we have $V(i) \in I(Y)$. In case $i = 0$, this follows from Prop. (3) of behaviors; in case $i = 1$, this is the assumption $X \in I(Y)$. But $I(Y) \subseteq I(B)$ and $V(i) \in I(Y)$ implies $V(i) \in I(B)$. Hence, $V(i)$ is one of

the $E'$ in the intersection $(E,B)^\top = \bigcap\{E' \in I(B) \mid E \subseteq E' \subseteq B\}$, from which we conclude $(E,B)^\top \subseteq V(i)$. This establishes $(V(0),V(1)) \models t$. Now consider any of the other transitions $t = (B \cup (\overline{A \setminus B}))/(A \setminus B)$, for $B \subseteq A$ with $B \notin F$. To show $(V(0),V(1)) \models t$, again, we just need to consider the case $V(1) \cap (A \setminus B) = \emptyset$ or, equivalently, $V(1) \subseteq B$, and any $i \in \{0,1\}$ such that $B \subseteq V(i)$. Then, we have $B \subseteq V(i) \subseteq V(1) \subseteq B$. This yields $Y = V(1) = B$ which contradicts the assumptions $Y \in F$ and $B \notin F$ by the construction of $t$. Hence, the proof of $(V(0),V(1)) \models t$ is complete and, moreover, $X \in I(C)(Y)$. Thus, $I(Y) \subseteq I(C)(Y)$.

For the other inclusion, $I(C)(Y) \subseteq I(Y)$, let $X \subseteq A$ be given such that $X \notin I(Y)$. We establish $(X,Y) \not\models (X \cup (\overline{A \setminus Y}))/(X,Y)^\top$ which is a transition of $C$, as $Y \in F$ by assumption. But this follows from the fact that $X \subset (X,Y)^\top$, because $X \notin I(Y)$, and $(A \setminus Y) \cap Y = \emptyset$. Thus, $(X,Y) \notin \mathit{2SM}(C)$, whence $X \notin I(C)(Y)$. This completes the proof of Thm. 3.11. □

Summarizing, behaviors $\mathit{Beh}(C)$, for parallel configurations $C$, yield a very simple model representation of $\mathit{SM}(C)$. For any given $B \in F(C)$, the set $I(C)(B)$ is a finite $(\cap,\subseteq)$ semi–lattice with maximal element $B$. For every $B' \supseteq B$, the semi–lattice $I(C)(B)$ is a full sub–lattice of $I(C)(B')$. As a simple example, consider the configuration $C =_{\mathrm{df}} bc/a \parallel ac/b \parallel \overline{a}/a \parallel \overline{b}/b \parallel \overline{c}/c$ over events $A = \{a,b,c\}$. Its behavior $\mathit{Beh}(C)$ is $A$–finite and, when restricted to the relevant events $A$, may be depicted as in Figure 2. Since $F(C) = \{A\}$ is a singleton set we only have one $(\cap,\subseteq)$ semi–lattice $I(C)(A)$. Moreover, $\mathit{SM}(C)$ is precisely the set of sequences whose worldwise intersection with $A$ are paths in the diagram ending in top element $A$.

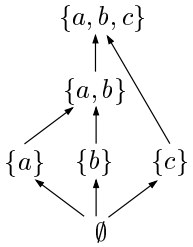

Fig. 2. $\{a,b,c\}$–bounded behavior

## 4. FULLY–ABSTRACT SEMANTICS

We have seen in the previous section that the behavior of a parallel configuration $P$ in all parallel contexts is captured by its set of sequence models $\mathit{SM}(P)$ or, equivalently, its behavior $\mathit{Beh}(P)$. This yields a denotational semantics in which parallel composition is intersection, i.e., $\mathit{SM}(P_1 \parallel P_2) = \mathit{SM}(P_1) \cap \mathit{SM}(P_2)$. Similarly, $\mathit{Beh}(P_1 \parallel P_2) = \mathit{Beh}(P_1) \cap \mathit{Beh}(P_2)$, where the intersection is taken pointwise. This section shows how this semantics can easily be extended to work with arbitrary contexts, thereby completely characterizing the semantics of $\mathcal{PC}$. However, the question, which still needs to be answered, is how to capture the semantics of the choice operator $+$. In view of the fact that $\parallel$ is logical *conjunction* $\wedge$ in the intuitionistic logic of stabilization sequences, it would be natural to expect that $+$ corresponds to logical *disjunction* $\vee$ over sequence models. However, it will turn out that the choice operator $+$ is not a disjunction on sequence models but on behaviors, i.e., on *sets* of sequence models.

As a counter example, showing that logical disjunction on sequence models does not suffice, consider transitions $a/b$ and $b/a$. Assume that the semantics of configuration $a/b + b/a$ would be completely described by formula $(a \supset b) \vee (b \supset a)$, when interpreted over stabilization sequences, i.e., $\mathit{SM}((a \supset b) \vee (b \supset a)) = \mathit{SM}(a \supset b) \cup \mathit{SM}(b \supset a)$. Now, as one can show, we have $K \models a \supset b$ or $K \models b \supset a$, for *every*

stabilization sequence $K$. Thus, $SM((a \supset b) \vee (b \supset a))$ contains all stabilization sequences, whence the formula $(a \supset b) \vee (b \supset a)$ is a logical tautology. In terms of sequence models alone, $a/b + b/a$ would be equivalent to the empty configuration $\mathbf{0}$. But obviously both configurations have different response behavior, as, e.g., $(a/b + b/a) \Downarrow_{\{a\}} \{a, b\}$ but only $\mathbf{0} \Downarrow_{\{a\}} \{a\}$. Also, the obvious idea of replacing linear stabilization sequences by arbitrary intuitionistic Kripke models does not work. We will see later that $a/b + b/a$ is step congruent to $a/b \parallel b/a$. Since the formulas $(a \supset b) \vee (b \supset a)$ and $(a \supset b) \wedge (b \supset a)$ are not intuitionistically equivalent, we cannot read $+$ as disjunction on arbitrary Kripke models. It does not appear sensible to try and find an intermediate class of intuitionistic Kripke models such that the behavior of choice configurations $P_1 + P_2$ can be characterized by the disjunctive formula $P_1 \vee P_2$. Such a semantics would have to use a modified interpretation of transition implication to account for different enabling properties. The next section shows that we need to distinguish transition $a/a$, which is triggered by $a$, from transition $b/b$, which is triggered by $b$. The naive logical interpretation would identify both transitions with *true*.

Instead of trying to read operator $+$ as logical disjunction, we will use semantic–preserving transformations to eliminate $+$ in favor of parallel composition, whose semantics we already know. There are two methods for achieving this. The naive method, which is discussed in a technical report [Lüttgen and Mendler 2000] is to encode $+$ in terms of $\parallel$ using additional distinguished events to achieve mutual exclusion between the transitions on different sides of the choice operator. The other method is to use an expansion law to distribute operator $+$ over operator $\parallel$ and to transform a configuration $C \in \mathcal{C}$ into a standard form $\sum_i C_i$, where all $C_i \in \mathcal{PC}$ are parallel configurations. The semantics of $C$ is then uniquely determined from the ones of all $C_i$. Here, we only consider the second method, since it is more algebraic than the first one and also does not depend on the use of distinguished events.

### 4.1 Reduction to Parallel Contexts

For extending the full–abstraction result to arbitrary contexts, one must address the following compositionality problem for $+$ which already manifests itself in Pnueli and Shalev's semantics. Consider configurations $C =_{\mathrm{df}} \overline{a}/b$ and $D =_{\mathrm{df}} \overline{a}/b \parallel a/a$ which have the same responses in all parallel contexts, i.e., $Beh(C) = Beh(D)$. However, in the choice context $\Phi[x] = (\cdot/e + x) \parallel \cdot/a$, we obtain $\Phi[D] \Downarrow \{a\}$ but $\Phi[C] \not\Downarrow \{a\}$. This context is able to detect that $D$ is enabled by environment $\cdot/a$ while $C$ is not. Hence, to be fully compositional 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 *enabling indicator* $\rho(C, A) \in \mathbb{B} =_{\mathrm{df}} \{ff, tt\}$ defined by $\rho(C, A) =_{\mathrm{df}} tt$, if $\mathsf{triggered}(C, A) \neq \emptyset$, and $\rho(C, A) =_{\mathrm{df}} ff$, otherwise. When $A \models C$, let us call $A$ *active*, if $\rho(C, A) = tt$, and *passive*, otherwise. This distinction is all we need to reduce step congruence to parallel contexts. Indeed, two configurations are step–congruent if and only if they have the same active and passive step responses in all *parallel* contexts.

PROPOSITION 4.1. *Let $C, D \in \mathcal{C}$. Then, $C \simeq D$ iff $\forall P \in \mathcal{PC}$, $E, A \subseteq_{fin} \Pi, b \in \mathbb{B}$. $((C \,\|\, P) \Downarrow_E A$ and $\rho(C, A) = b)$ iff $((D \,\|\, P) \Downarrow_E A$ and $\rho(D, A) = b)$.*

This proposition is a corollary to the more general Thm. 4.13 presented in Sec. 4.4. Prop. 4.1 now suggests the following refinement of the naive fully–abstract semantics $[\![\cdot]\!]_0$. For every $C \in \mathcal{C}$, we define

$$[\![C]\!]_1^b =_{\mathrm{df}} \{\langle A, P\rangle \mid (C \parallel P) \Downarrow A,\, \rho(C, A) = b,\, P \in \mathcal{PC}\}\,,$$

where $b \in \mathbb{B}$. We may view $[\![C]\!]_1^{tt}$ as the collection of *active* and $[\![C]\!]_1^{ff}$ as the collection of *passive* responses for $C$ in *parallel contexts*. From Prop. 4.1, then, we obtain the following result.

PROPOSITION 4.2. *$\forall C, D \in \mathcal{C}$. $C \simeq D$ iff $[\![C]\!]_1^{tt} = [\![D]\!]_1^{tt}$ and $[\![C]\!]_1^{ff} = [\![D]\!]_1^{ff}$.*

## 4.2 Reduction to Parallel Configurations

The next step is to eliminate the choice operator from the configurations themselves and to show that the response behavior of every configuration can be determined from that of its parallel components. As mentioned earlier, this will be achieved by transforming configurations into a standard form in which the choice operator is the outermost operator.

To begin with the development of a standard form, observe that the naive distributivity law $(t_1 + t_2) \parallel t_3 \simeq (t_1 \parallel t_3) + (t_2 \parallel t_3)$, with the two occurrences of $t_3$ on the right–hand side suitably renamed, does not in general hold. As a counter example, consider transitions $t_i =_{\mathrm{df}} a_i\overline{b_i}/c_i$, for $1 \le i \le 3$, and assume that all events are mutually distinct. Then, in a context in which transition $t_2$ is enabled but not transition $t_1$, transition $t_3$ in $C =_{\mathrm{df}} (t_1 + t_2) \parallel t_3$ is forced to interact with $t_2$, while in $D =_{\mathrm{df}} (t_1 \parallel t_3) + (t_2 \parallel t_3)$ it may run by itself in the summand $t_1 \parallel t_3$. For example, if $E = \{a_2, a_3\}$ then $D \Downarrow_E \{a_2, a_3, c_3\}$, but the only $A$ with $c_3 \in A$ and $C \Downarrow_E A$ is $A = \{a_2, a_3, c_2, c_3\}$. The same applies if the context enables $t_1$ but not $t_2$. The distributivity law, however, can be patched as

$$(t_1 + t_2) \parallel t_3 \;\simeq\; t_1 \parallel D_1(t_3) + t_2 \parallel D_2(t_3)\,,$$

where configurations $D_i(t_3)$, for $i \in \{1, 2\}$, are suitable weakenings of $t_3$ that disable transition $t_3$, whenever $t_i$ is disabled but $t_{3-i}$ is enabled. There are two ways for defining such configurations.

The most elegant solution is to exploit the failure event $\perp$. In the example, we could define $D_i(t_3) =_{\mathrm{df}} D_i \parallel t_3$, for $i \in \{1, 2\}$, where

$$D_i =_{\mathrm{df}} \overline{a}_i a_{3-i}\overline{b}_{3-i}/\perp \parallel b_i a_{3-i}\overline{b}_{3-i}/\perp\,.$$

The "watchdog" configuration $D_i$ is enabled exactly if $t_i$ is not enabled and $t_{3-i}$ is, in which case it produces a failure. Formally, for all parallel contexts $P$, configuration $D_i$ has the property $(D_i \parallel P) \Downarrow A$ if and only if (i) $P \Downarrow A$ and (ii) $A$ triggers $t_i$ or does not trigger $t_{3-i}$. Thus, $D_i$ does not change any of the responses of $P$, it only prohibits some of them. We will see below how this can be generalized, namely how one may construct, for any given configurations $C_1$ and $C_2$, a watchdog configuration $\mathsf{watch}(C_1, C_2)$ such that $(D \parallel \mathsf{watch}(C_1, C_2)) \Downarrow A$ if and only if $D \Downarrow A$ and $\mathsf{triggered}(C_1, A) \neq \emptyset$ or $\mathsf{triggered}(C_2, A) = \emptyset$.

To formally construct watchdogs in a finitary fashion, we need to refer to the events that occur in a configuration. For every configuration $C$, let $\Pi(C)$ denote the set of all events that syntactically occur in $C$. Then, we define $\mathsf{watch}(C_1, C_2) \in \mathcal{PC}$

to be the parallel configuration

$$\| \{A, \overline{E \setminus A}/\bot \mid A \subseteq E = \Pi(C_1) \cup \Pi(C_2),\ \rho(C_1, A) = \mathit{ff},\ \text{and}\ \rho(C_2, A) = \mathit{tt}\}\,.$$

The crucial semantic property of watchdogs is stated in the following proposition.

PROPOSITION 4.3. *Let $C_1, C_2, D \in \mathcal{C}$ and $A \subseteq \Pi$. Then, $(D \parallel \mathsf{watch}(C_1, C_2)) \Downarrow A$ iff $D \Downarrow A$ and $\mathsf{triggered}(C_1, A) \neq \emptyset$ or $\mathsf{triggered}(C_2, A) = \emptyset$.*

PROOF. In the following, let $E =_{\mathrm{df}} \Pi(C_1) \cup \Pi(C_2)$ and $A \subseteq E \cup \Pi(D)$. We begin with direction "$\Longrightarrow$". Since all transitions of $\mathsf{watch}(C_1, C_2)$ have event $\bot$ as their only action event, it follows from $(D \parallel \mathsf{watch}(C_1, C_2)) \Downarrow A$ that none of the watchdog transitions can be enabled. This implies that response $A$ must come from configuration $D$ alone, i.e., $D \Downarrow A$. In particular, transition $A', \overline{E \setminus A'}/\bot$, where $A' = E \cap A$, cannot be included in $\underline{\mathsf{watch}}(C_1, C_2)$; otherwise, it would be enabled in the response $A$ since $A' \subseteq A$ and $\overline{E \setminus A'} \cap A = \emptyset$. But this implies $\rho(C_1, A') \neq \mathit{ff}$ or $\rho(C_2, A') \neq \mathit{tt}$. Hence by the definition of $A'$, also $\rho(C_1, A) \neq \mathit{ff}$ or $\rho(C_2, A) \neq \mathit{tt}$ or, equivalently, $\mathsf{triggered}(C_1, A) \neq \emptyset$ or $\mathsf{triggered}(C_2, A) = \emptyset$.

For proving direction "$\Longleftarrow$", let us assume (1) $D \Downarrow A$ and (2) $\mathsf{triggered}(C_1, A) \neq \emptyset$ or $\mathsf{triggered}(C_2, A) = \emptyset$. This implies that $A$ does not enable any transitions of $\underline{\mathsf{watch}}(C_1, C_2)$. Suppose otherwise, i.e., there exists an $A' \subseteq E$ such that (a) $A' \subseteq A$ and $\overline{E \setminus A'} \cap A = \emptyset$ and (b) $\rho(C_1, A') = \mathit{ff}$ and $\rho(C_2, A') = \mathit{tt}$. Property (a) means $A \cap E = A'$ and further Property (b) implies $\rho(C_1, A) = \mathit{ff}$ and $\rho(C_2, A') = \mathit{tt}$, which contradicts Assumption (2). Hence, $A$ does not enable any transitions of $\mathsf{watch}(C_1, C_2)$. But then Assumption (1) implies $(D \parallel \mathsf{watch}(C_1, C_2)) \Downarrow A$ by the definition of step responses. □

The watchdogs admit the following simple expansion law whose proof, which can be found in [Lüttgen and Mendler 2000], is a direct application of Prop. 4.1.

LEMMA 4.4 EXPANSION. *Let $P, Q, R \in \mathcal{C}$. Then, the following holds:*
$(P + Q) \parallel R \simeq (\mathit{watch}(P, Q) \parallel P \parallel R) + (\mathit{watch}(Q, P) \parallel Q \parallel R)$.

Repeated application of Lemma 4.4 can be used to push systematically all occurrences of the choice operator $+$ to the outside of the configuration $C$ under consideration, until $+$ becomes the outermost operator. We can think of this transformation of $C$ as a static analysis which reveals the top–level choice structure of $C$. The general expansion algorithm, which is omitted here, associates with every $C \in \mathcal{C}$ a set $\mathsf{ind}(C)$ of indices and, for every $i \in \mathsf{ind}(C)$, a parallel configuration $C_i \in \mathcal{PC}$. The configurations $C_i$ essentially correspond to the maximal consistent subsets of $\mathsf{trans}(C)$, patched up with appropriate watchdog configurations.

LEMMA 4.5 STANDARD FORM. *Let $C \in \mathcal{C}$. Then, there exists a finite index set $\mathit{ind}(C)$ and $C_i \in \mathcal{PC}$, for $i \in \mathit{ind}(C)$, such that $C \simeq \sum_{i \in \mathit{ind}(C)} C_i$.*

Hence, $[\![C]\!]_1^b = [\![\sum_{i \in \mathsf{ind}(C)} C_i]\!]_1^b$ by Prop. 4.2, for $b \in \mathbb{B}$. 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* its summands, we have

$$[\![ \sum_{i \in \mathsf{ind}(C)} C_i ]\!]_1^{\mathit{tt}} = \bigcup_{i \in \mathsf{ind}(C)} [\![C_i]\!]_1^{\mathit{tt}} \quad \text{and} \quad [\![ \sum_{i \in \mathsf{ind}(C)} C_i ]\!]_1^{\mathit{ff}} = \bigcap_{i \in \mathsf{ind}(C)} [\![C_i]\!]_1^{\mathit{ff}}\,.$$

Thus, we obtain the following proposition which states the desired reduction of the full–abstraction problem to parallel configurations within parallel contexts.

PROPOSITION 4.6. *Let $C, D \in \mathcal{C}$. Then, $C \simeq D$ iff*

$$\bigcup_{i \in \mathit{ind}(C)} [\![C_i]\!]_1^{tt} = \bigcup_{j \in \mathit{ind}(D)} [\![D_j]\!]_1^{tt} \quad \text{and} \quad \bigcap_{i \in \mathit{ind}(C)} [\![C_i]\!]_1^{ff} = \bigcap_{j \in \mathit{ind}(D)} [\![D_j]\!]_1^{ff}.$$

The proof of this proposition, and also the one of Thm. 4.9 below, requires the following distributivity property stated in terms of admissible sets of transitions and proved in [Lüttgen and Mendler 2000].

LEMMA 4.7 DISTRIBUTIVITY. *Let $S, C, D \in \mathcal{C}$, let $E \subseteq_{fin} \Pi$, and let $T \subseteq_{fin} \mathcal{T}$. Then, $T$ is $E$–admissible for $(C + D) \parallel S$ iff one of the following conditions holds:*

*(1) $T \cap \mathsf{trans}(C) \neq \emptyset$, and $T$ is $E$–admissible for $C \parallel S$.*

*(2) $T \cap \mathsf{trans}(D) \neq \emptyset$, and $T$ is $E$–admissible for $D \parallel S$.*

*(3) $T \subseteq \mathsf{trans}(S)$, and $T$ is $E$–admissible for both $C \parallel S$ and $D \parallel S$.*

*In Case (1) we also have $T \subseteq \mathsf{trans}(C \parallel S)$ and in Case (2) $T \subseteq \mathsf{trans}(D \parallel S)$.*

Using this lemma we now prove Prop. 4.6.

PROOF OF PROPOSITION 4.6. We present the proof for two indices only, i.e., we assume $C \simeq C_1 + C_2$ and $D \simeq D_1 + D_2$, where $C_i, D_j \in \mathcal{PC}$ are parallel configurations. The general case is handled in the same way, noting that $+$ is associative. Observe that the statement of Prop. 4.6 reduces to the congruence condition $[\![C]\!]_1^b = [\![D]\!]_1^b$ expressed in Prop. 4.2, in case both configurations have only one index. In what follows, $\rho(C, A) \in \mathbb{B}$ denotes again the enabling indicator, so that $\rho(C, A) = \mathit{ff}$, if $\mathsf{triggered}(C, A) = \emptyset$, and $\rho(C, A) = \mathit{tt}$, otherwise.

"$\Longleftarrow$". We assume

$$[\![C_1]\!]_1^{tt} \cup [\![C_2]\!]_1^{tt} = [\![D_1]\!]_1^{tt} \cup [\![D_2]\!]_1^{tt} \quad \text{and} \tag{2}$$

$$[\![C_1]\!]_1^{ff} \cap [\![C_2]\!]_1^{ff} = [\![D_1]\!]_1^{ff} \cap [\![D_2]\!]_1^{ff}. \tag{3}$$

We must show, by Prop. 4.1, that for every $P \in \mathcal{PC}$ and $E, A \subseteq_{\mathrm{fin}} \Pi$,

$$\begin{array}{l}((C_1 + C_2) \parallel P) \Downarrow_E A \ \text{ implies} \\ \quad ((D_1 + D_2) \parallel P) \Downarrow_E A \text{ and } \rho(D_1 + D_2, A) = \rho(C_1 + C_2, A)\end{array} \tag{4}$$

and vice versa, with the roles of $C_i$ and $D_i$ interchanged. We may assume that $E = \emptyset$ since any $E$ is already quantified implicitly by $P$. Moreover, it suffices to prove the implication in Statement (4) because of symmetry. Suppose that $((C_1 + C_2) \parallel P) \Downarrow A$. By Lemma 4.7 we have to consider the following two cases:

(1) There exists some index $i \in \{1, 2\}$ such that $(C_i \parallel P) \Downarrow A$ and $\rho(C_i, A) = \mathit{tt}$.
(2) For both indices $i \in \{1, 2\}$, it is true that $\rho(C_i, A) = \mathit{ff}$ and $(C_i \parallel P) \Downarrow A$.

In Case (1), by definition, $\langle A, P \rangle \in [\![C_i]\!]_1^{tt}$. From Equation (2) it follows that there exists some index $j \in \{1, 2\}$ such that $\langle A, P \rangle \in [\![D_j]\!]_1^{tt}$. But this statement yields $((D_1 + D_2) \parallel P) \Downarrow A$ and $\rho(D_j, A) = \mathit{tt}$ when reading Lemma 4.7(1/2)

backwards. Hence, $\rho(D_1 + D_2, A) = tt = \rho(C_1 + C_2, A)$ which proves Statement (4) in Case (1). Regarding Case (2), $\langle A, P\rangle \in [\![C_i]\!]_1^{ff}$ holds, whence by Equation (3), $\langle A, P\rangle \in [\![D_j]\!]_1^{ff}$, for both $j \in \{1, 2\}$. This in consequence means that $(D_1 \parallel P) \Downarrow A$ and $(D_2 \parallel P) \Downarrow A$, as well as $\rho(D_1, A) = ff = \rho(D_2, A)$. So, by employing Lemma 4.7(3) backwards, we obtain $((D_1 + D_2) \parallel P) \Downarrow A$ and $\rho(D_1 + D_2, A) = ff = \rho(C_1 + C_2, A)$, as desired.

"$\Longrightarrow$". For this direction, let us assume $C \simeq D$, i.e., $C_1 + C_2 \simeq D_1 + D_2$. Let $\langle A, P\rangle \in [\![C_1]\!]_1^{tt}$, i.e., $(C_1 \parallel P) \Downarrow A$ and $\rho(C_1, A) = tt$. By Lemma 4.7(1), then, $((C_1 + C_2) \parallel P) \Downarrow A$, from which we may infer $((D_1 + D_2) \parallel P) \Downarrow A$ and $\rho(D_1 + D_2, A) = \rho(C_1 + C_2, A) = tt$ by Prop. 4.1. This means, by Lemma 4.7(1/2), that $\rho(D_i, A) = tt$ and $(D_i \parallel P) \Downarrow A$, for some $i \in \{1, 2\}$. Thus, we have $\langle A, P\rangle \in [\![D_1]\!]_1^{tt} \cup [\![D_2]\!]_1^{tt}$ which implies $[\![C_1]\!]_1^{tt} \subseteq [\![D_1]\!]_1^{tt} \cup [\![D_2]\!]_1^{tt}$. A similar argument shows that $[\![C_2]\!]_1^{tt} \subseteq [\![D_1]\!]_1^{tt} \cup [\![D_2]\!]_1^{tt}$ which yields $[\![C_1]\!]_1^{tt} \cup [\![C_2]\!]_1^{tt} \subseteq [\![D_1]\!]_1^{tt} \cup [\![D_2]\!]_1^{tt}$. The other direction follows by symmetry.

Finally, assume $\langle A, P\rangle \in [\![C_1]\!]_1^{ff} \cap [\![C_2]\!]_1^{ff}$, i.e., $\rho(C_i, A) = ff$ and $(C_i \parallel P) \Downarrow A$. Lemma 4.7(3) implies $((C_1 + C_2) \parallel P) \Downarrow A$. Now we apply Prop. 4.1 again, which establishes $((D_1 + D_2) \parallel P) \Downarrow A$. Moreover, $\rho(D_1 + D_2, A) = \rho(C_1 + C_2, A) = ff$, whence both $\rho(D_1, A) = ff = \rho(D_2, A)$. A final reference to Lemma 4.7(3) implies $(D_1 \parallel P) \Downarrow A$ and $(D_2 \parallel P) \Downarrow A$. This verifies the inclusion $[\![C_1]\!]_1^{ff} \cap [\![C_2]\!]_1^{ff} \subseteq [\![D_1]\!]_1^{ff} \cap [\![D_2]\!]_1^{ff}$. The other direction, again, is by symmetry.

This finishes the proof. □

Prop. 4.6 yields a second refinement of our fully–abstract semantics that now only depends on the response behavior of parallel configurations in parallel contexts. However, it still refers to the syntax. Next, the main work will be done, presenting a semantic analysis of the dynamic interaction between parallel configurations.

### 4.3 Full–abstraction Theorem

Once a configuration $C \in \mathcal{C}$ is transformed into a sum $\sum_{i \in \mathsf{ind}(C)} C_i$ of parallel configurations, its semantics may be uniquely determined by the behavior of the $C_i \in \mathcal{PC}$. By Prop. 4.1 it is enough to know the responses of each $C_i$ in all parallel contexts, together with the information of whether these responses are active or passive. Moreover, by Thms. 3.5 and 3.10, the responses of $C_i$ in all parallel contexts are characterized by their behaviors $Beh(C_i)$. Therefore, the responses of $C$ in all contexts must be determined by the sets $Beh(C_i)$ and predicates $\rho(C_i, A)$, for all $i \in \mathsf{ind}(C)$ and $A \subseteq_{\text{fin}} \Pi \setminus \{\bot\}$. Unfortunately, the obvious but somewhat naive idea of simply collecting all the sets $Beh(C_i)$, together with their triggering behavior $\rho(C_i) =_{\text{df}} \lambda A.\, \rho(C_i, A)$, and then considering the identity of sets as equivalence does not work. The semantics defined in this direct way, namely $[\![C]\!] =_{\text{df}} \{\langle\, Beh(C_i)\,,\, \rho(C_i)\,\rangle \mid i \in \mathsf{ind}(C)\}$, would not allow us to derive, e.g., the congruence $a/b + b/a \simeq a/b \parallel b/a$. Indeed, it is not the case that $[\![a/b + b/a]\!] = \{\langle Beh(a/b), \rho(a/b)\rangle\,,\, \langle Beh(b/a), \rho(b/a)\rangle\}$ is the same set as $[\![a/b \parallel b/a]\!] = \{\langle Beh(a/b \parallel b/a), \rho(a/b \parallel b/a)\rangle\}$ since, e.g., $Beh(a/b)$ is different from $Beh(a/b \parallel b/a)$. However, it is true that $Beh(a/b)$ and $Beh(b/a)$ together "cover" the same behavior as $Beh(a/b \parallel b/a)$. To achieve a simple formalization of this covering property, it is useful to consider the "complements" of $Beh(a/b)$, $Beh(b/a)$, and $Beh(a/b \parallel b/a)$, to which we refer as *contexts*.

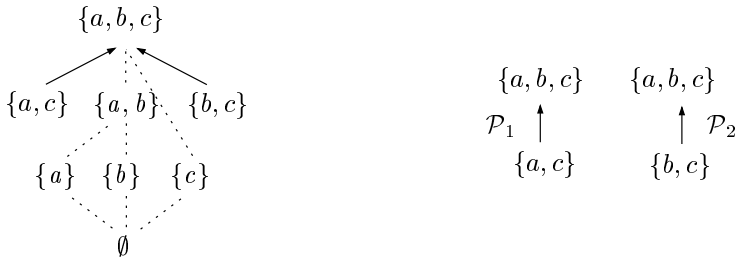


Fig. 3. Complement behavior for Figure 2 (left) and its covering $\{a, b, c\}$–contexts (right).

*Definition* 4.8 *Context.* Let $A \subseteq_{\text{fin}} \Pi$. An $A$–bounded behavior $\mathcal{P} = \langle F, I\rangle$ is called an $A$–*context* for $C \in \mathcal{PC}$ if (i) $A \in F(C)$, and (ii) $I(A) \cap I(C)(A) = \{A\}$ holds, where $\langle F(C), I(C)\rangle = \mathit{Beh}(C)$.

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 (cf. Prop. (i)), but which only have the final world $A$ in common with the sequence models of $C$ (cf. Prop. (ii)). These properties imply that, for every configuration $P$ with $\mathit{Beh}(P) = \mathcal{P}$, we have $(C \parallel P) \Downarrow A$. Note that, since every $A$–context $\langle F, I\rangle$ is $A$–bounded, $I$ is essentially just a $\cap$–closed subset of $2^A$ with top element $A$. In other words, an $A$–context $\langle F, I\rangle$ may be identified with the complete $(\cap, \subseteq)$ sub–semi–lattice $I(A)$ of $2^A$. We will henceforth use the simpler presentation $\langle A, I(A)\rangle$ rather than $\langle\{A\}, I\rangle$. In fact, we might even write $I(A)$ since the top element is uniquely determined, but it is often useful to indicate the top element explicitly.

In the following we will only be interested in the maximal $A$–contexts of a configuration $C \in \mathcal{PC}$, where maximality is with respect to the natural component–wise subset ordering on $A$–bounded behaviors. More precisely, given two $A$–bounded behaviors $\mathcal{P} = \langle A, I(A)\rangle$ and $\mathcal{P}' = \langle A, I'(A)\rangle$, we say that $\mathcal{P}$ is a *sub–behavior* of $\mathcal{P}'$, written $\mathcal{P} \subseteq \mathcal{P}'$, if $I(A) \subseteq I'(A)$. Then, an $A$-context $\mathcal{P}$ of $C$ is called *maximal* if $\mathcal{P} \subseteq \mathcal{P}'$ implies $\mathcal{P} = \mathcal{P}'$, for all $A$–contexts $\mathcal{P}'$ of $C$. Because of the finiteness of $A$–bounded behaviors, every $A$–context of $C$ must be contained in a maximal one. Intuitively, the property of maximality of $\mathcal{P}$ implies, whenever $P$ is a configuration such that $\mathit{Beh}(P) = \mathcal{P}$, that $(C \parallel P) \Downarrow A$ and also that each nontrivial transition in $P$ is necessarily involved in the generation of response $A$.

Consider again the example $C =_{\text{df}} bc/a \parallel ac/b \parallel \overline{a}/a \parallel \overline{b}/b \parallel \overline{c}/c$ from above, whose $A$–bounded behavior $\mathit{Beh}(C) = \langle\{A\}, I(C)\rangle$, where $A = \{a, b, c\}$, is described by the diagram of Figure 2. To get the $A$–contexts of $C$, we must consider the "complement" in $I(C)(A)$, i.e., all $B \subset A$ that are missing in the lattice of Figure 2. This is illustrated by the left diagram in Figure 3, where lattice $\mathit{Beh}(C)$ is indicated by dashed lines and the complement by solid arrows. As one can see, this complement is not a behavior, e.g., it is not $\cap$–closed, but it can be covered by the two $A$–contexts

$$\mathcal{P}_1 =_{\text{df}} \langle\{a,b,c\}, \{\{a,c\}, \{a,b,c\}\}\rangle, \qquad \mathcal{P}_2 =_{\text{df}} \langle\{a,b,c\}, \{\{b,c\}, \{a,b,c\}\}\rangle, \quad (5)$$

which are drawn separately in Figure 3 on the right. In fact, $\mathcal{P}_1$ and $\mathcal{P}_2$ are the two maximal $A$–contexts of $\mathit{Beh}(C)$. Since they are behaviors, the $A$–contexts can

be represented by parallel configurations, such as $P_1 =_{\mathrm{df}} \cdot/ac \parallel \overline{b}/b$ and $P_2 =_{\mathrm{df}} \cdot/bc \parallel \overline{a}/a$, respectively. These maximal $A$–contexts subsume all environments in which $C$ takes part in response $A$. Indeed, one can check that $(C \parallel P_1) \Downarrow A$ and $(C \parallel P_2) \Downarrow A$. For every $C \in \mathcal{PC}$ and $b \in \mathbb{B}$, we finally define

$$[\![C]\!]_2^b =_{\mathrm{df}} \{\langle A, I(A)\rangle \mid A \subseteq_{\mathrm{fin}} \Pi,\ \rho(C,A) = b,\ \langle A, I(A)\rangle \text{ is an } A\text{–context for } C\}\,.$$

The elements $\langle A, L\rangle \in [\![C]\!]_2^b$ are $(\cap, \subseteq)$ sub–semi–lattices $L$ of $2^A$ that represent all the bounded context behaviors, i.e., environments, generating the joint response $A$. The superscript $b \in \mathbb{B}$ determines whether $C$ is actively participating ($b = tt$) or only passively admitting ($b = ff$) the macro step resulting in $A$. In the latter case, the response must entirely come from the environment. This is reflected in the fact that all passive contexts $\langle A, L\rangle \in [\![C]\!]_2^{ff}$ are of the form $\langle A, L\rangle = \langle A, \{A\}\rangle$, which we will abbreviate as $id_A$ for convenience. The passive $A$–context $id_A$ means that the environment $P$ must be equivalent to transition $\cdot/A$ in order for $(C \parallel P) \Downarrow A$ to hold. Another structural property, which we may exploit, is that an $A$–context $\mathcal{P}$ is contained in $[\![C]\!]_2^{tt}$ if and only if there exists a maximal $A$–context $\mathcal{P}_{\max} \in [\![C]\!]_2^{tt}$ with $\mathcal{P} \subseteq \mathcal{P}_{\max}$. Consequently, we only need to list the maximal elements of $[\![C]\!]_2^{tt}$ relative to any given response $A$. We now obtain our main theorem as a corollary to Prop. 4.6 and Thm. 3.5.

THEOREM 4.9 FULL ABSTRACTION. *Let $C, D \in \mathcal{C}$. Then, $C \simeq D$ iff*

$$\bigcup_{i \in ind(C)} [\![C_i]\!]_2^{tt} = \bigcup_{j \in ind(D)} [\![D_j]\!]_2^{tt} \quad \text{and} \quad \bigcap_{i \in ind(C)} [\![C_i]\!]_2^{ff} = \bigcap_{j \in ind(D)} [\![D_j]\!]_2^{ff}\,.$$

PROOF. We begin with two observations about $A$–contexts, for $A \subseteq_{\mathrm{fin}} \Pi$.

*(1)* For every $P \in \mathcal{PC}$, consider the pair $Beh(P,A) =_{\mathrm{df}} \langle A, L\rangle$ with $L =_{\mathrm{df}} \{V(0) \mid (n,V) \in SM(P) \text{ and } V(n-1) = A\}$. For every $C \in \mathcal{PC}$, pair $Beh(P,A)$ possesses the property

$$(C \parallel P) \Downarrow A \text{ if and only if } Beh(P,A) \text{ is an } A\text{–context of } C\,. \tag{6}$$

This follows essentially from Thm. 3.4 and Def. 4.8 of $A$–contexts, as will be shown in the following. Note that if $A$ is not a classical model of $P$ then $Beh(P,A)$ is not even a behavior.

"$\Longleftarrow$". This direction follows from the equation $I(P)(A) \cap I(C)(A) = \{A\}$, which is valid for contexts, and Defs. 4.8(ii) and 3.3 as well as Thm. 3.4.

"$\Longrightarrow$". Suppose $(C \parallel P) \Downarrow A$. We first show that $Beh(P,A) = \langle A, L\rangle$ with $L = \{V(0) \mid (n,V) \in SM(P) \text{ and } V(n-1) = A\}$ is a behavior. Since $A \in RM(C \parallel P)$, event set $A$ is a classical model of $P$, i.e., $A \in SM(P)$. Thus, $A \in L$. Moreover, $L = I(P)(A)$ is closed under intersection, as $Beh(P)$ is a behavior. In addition, $Beh(P,A)$ is $A$–bounded by construction.

Next, we prove that $Beh(P,A)$ is an $A$–context of $C$. Since $A$ is a response model of $C \parallel P$, it is in particular a classical model of $C$. Hence, $A \in F(C)$. It remains to prove that $L \cap I(C)(A) = \{A\}$. The inclusion "$\supseteq$" is trivial as $A \in L \cap I(C)(A)$. For the other inclusion "$\subseteq$" assume $B \in L \cap I(C)(A)$, i.e., $B \in L = I(P)(A)$ and $B \in I(C)(A)$. This implies $B \subseteq A$. Suppose, $B \neq A$. Because of the choice of $B$ and the definitions of $I(P)(A)$ and $I(C)(A)$, there

must exist sequence models $(n_P, V_P) \in SM(P)$ and $(n_C, V_C) \in SM(C)$ such that $V_P(0) = B = V_C(0)$ and $V_P(n_P - 1) = A = V_C(n_C - 1)$. Now consider the 2–sequence $(B, A)$. One can show that $(n_P, V_P) \models P$ implies $(B, A) \models P$ and similarly that $(n_C, V_C) \models C$ implies $(B, A) \models C$. Therefore, $(B, A) \models C \parallel P$ which contradicts $A \in RM(C \parallel P)$ by Def. 3.1. Hence, $B = A$ and $L \cap I(C)(A) = \{A\}$. This completes the proof that $Beh(P, A)$ is an $A$–context of $C$.

*(2)* Suppose $\langle A, L\rangle$ is a $(\cap, \subseteq)$ sub–semi–lattice of $2^A$. Then, by Thm. 3.11, there must exist a parallel configuration $P \in \mathcal{PC}$ in the events $A$ and not using $\perp$, such that $Beh(P)$, when restricted to the events $A$, is identical to $\langle A, L\rangle$. These also satisfy, for every $C \in \mathcal{PC}$, the property

$$(C \parallel P) \Downarrow A \text{ if and only if } \langle A, L\rangle \text{ is an } A\text{–context of } C\,. \tag{7}$$

Thm. 4.9 is now a consequence of Prop. 4.6 and the following facts. For all $A \subseteq_{\text{fin}} \Pi$, $D \in \mathcal{PC}$, and $b \in \mathbb{B}$:

$$\forall L\ \exists P.\ \langle A, L\rangle \in [\![D]\!]_2^b \text{ if and only if } \langle A, P\rangle \in [\![D]\!]_1^b\,, \text{ and} \tag{8}$$

$$\forall P\ \exists L.\ \langle A, P\rangle \in [\![D]\!]_1^b \text{ if and only if } \langle A, L\rangle \in [\![D]\!]_2^b\,. \tag{9}$$

For establishing Statements (8) and (9), we use Statements (7) and (6), respectively, together with the construction of behavior $Beh(P, A)$ and Thm. 3.5. In both cases, we also exploit that the enabling indicator $\rho$ only depends on $A$, but not on the above $P$ or $L$. Thm. 4.9 is derived from Statements (8) and (9) and from Prop. 4.6 in the obvious fashion. □

Let us consider some examples. For the configuration $C =_{\text{df}} bc/a \parallel ac/b \parallel \overline{a}/a \parallel \overline{b}/b \parallel \overline{c}/c$ in Figure 3, $[\![C]\!]_2^{tt} = \{\mathcal{P}_1\,, \mathcal{P}_2\}$ and $[\![C]\!]_2^{ff} = \emptyset$, where $\mathcal{P}_1$ and $\mathcal{P}_2$ are given in Equation (5). Note, that here and in the following, we only list maximal $\{a, b, c\}$–contexts. This structure can also be generated from $D_1 + D_2$, where $D_1 =_{\text{df}} ac/b \parallel \overline{b}/b \parallel \overline{c}/c$ and $D_2 =_{\text{df}} bc/a \parallel \overline{b}/b \parallel \overline{a}/a$. One obtains

$$[\![D_1]\!]_2^{tt} = \{\,\mathcal{P}_1\,\}\,, \quad [\![D_2]\!]_2^{tt} = \{\,\mathcal{P}_2\,\}\,, \quad [\![D_1]\!]_2^{ff} = \{id_{\{b,c\}}\}\,, \quad [\![D_2]\!]_2^{ff} = \{id_{\{a,b\}}\}\,.$$

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. 4.9, then, $C \simeq D_1 + D_2$. The Statecharts axiom hidden in this example, which reflects a causality principle, is

$$a/b \parallel b/a \;\simeq\; a/b + b/a\,, \tag{10}$$

for any events $a, b \in \Pi$. Intuitively, this congruence states that, if $a$ and $b$ mutually depend on each other (left–hand side), then *either* $a$ causes $b$ *or* $b$ causes $a$ (right–hand side). We might call this the "*tie–break axiom*" or "*causality axiom.*" More specifically, we obtain the following semantics:

$$\begin{aligned}
[\![a/b \parallel b/a]\!]_2^{tt} &= \{\,\langle\{a,b\}, \{\{a\}\,, \{a,b\}\}\rangle\,,\ \langle\{a,b\}, \{\{b\}\,, \{a,b\}\}\rangle\,\}\\
[\![a/b \parallel b/a]\!]_2^{ff} &= \{\,id_\emptyset\,\}\\
[\![a/b]\!]_2^{tt} &= \{\,\langle\{a,b\}, \{\{a\}\,, \{a,b\}\}\rangle\,\}\\
[\![a/b]\!]_2^{ff} &= \{\,id_\emptyset\,,\ id_{\{b\}}\,\}\\
[\![b/a]\!]_2^{tt} &= \{\,\langle\{a,b\}, \{\{b\}\,, \{a,b\}\}\rangle\,\}\\
[\![b/a]\!]_2^{ff} &= \{\,id_\emptyset\,,\ id_{\{a\}}\,\}\,.
\end{aligned}$$

From this $[\![a/b \parallel b/a]\!]_2^{tt} = [\![a/b]\!]_2^{tt} \cup [\![b/a]\!]_2^{tt}$ and $[\![a/b \parallel b/a]\!]_2^{ff} = [\![a/b]\!]_2^{ff} \cap [\![b/a]\!]_2^{ff}$ is derived. Hence, Thm. 4.9 then establishes the congruence in Equation (10).

To finish off our list of examples, we return to Sec. 3 and re–visit the compositionality problem in the light of our semantics. First of all, we verify that $C_{79} = \overline{b}/a + b/a \simeq \overline{b}/a \parallel b/a = C'_{79}$, as stated in Sec. 3. The semantics of parallel configuration $C'_{79} \in \mathcal{PC}$ is

$$[\![C'_{79}]\!]_2^{tt} = \{ \langle \{a\}, \{\emptyset, \{a\}\} \rangle , \langle \{a,b\}, \{\{b\}, \{a,b\}\} \rangle \} \quad \text{and} \quad [\![C'_{79}]\!]_2^{ff} = \emptyset .$$

The active and passive contexts of $b/a$ were given above; it remains to analyze $\overline{b}/a$:

$$[\![\overline{b}/a]\!]_2^{tt} = \{ \langle \{a\}, \{\emptyset, \{a\}\} \rangle \} \quad \text{and} \quad [\![\overline{b}/a]\!]_2^{ff} = \{ id_{\{b\}}, id_{\{a,b\}} \} .$$

When combining the above pieces, we obtain $[\![C'_{79}]\!]_2^{tt} = [\![\overline{b}/a]\!]_2^{tt} \cup [\![b/a]\!]_2^{tt}$ and $[\![C'_{79}]\!]_2^{ff} = \emptyset = [\![\overline{b}/a]\!]_2^{ff} \cap [\![b/a]\!]_2^{ff}$, whence $C'_{79} \simeq \overline{b}/a + b/a = C_{79}$. In addition, our semantics shows why configurations $C_{79}$ and $C'_{79}$ are distinguished from configuration $C_{14} = \cdot/a \parallel b/a$. Configuration $C_{14}$ has the active $\{a,b\}$–context $\langle \{a,b\}, \{\emptyset, \{b\}, \{a,b\}\} \rangle \in [\![C_{14}]\!]_2^{tt}$ which is neither contained in $[\![C'_{79}]\!]_2^{tt}$ nor in $[\![C_{79}]\!]_2^{tt} = [\![\overline{b}/a]\!]_2^{tt} \cup [\![b/a]\!]_2^{tt}$. This $\{a,b\}$–context $\langle \{a,b\}, \{\emptyset, \{b\}, \{a,b\}\} \rangle$ corresponds to context $\Phi_{56}[x] = x \parallel a/b$ used in Sec. 3 to differentiate $C_{79}$ from $C_{14}$. It shows that $\Phi[C_{14}] \Downarrow \{a,b\}$ but $\Phi[C_{79}] \not\Downarrow \{a,b\}$.

With Thm. 4.9 we have finally achieved our goal. Summarizing, the fully–abstract semantics developed in this paper consists of the mapping $[\![\cdot]\!]_3$ given by

$$[\![C]\!]_3 =_{\text{df}} \langle \bigcup_{i \in \mathsf{ind}(C)} [\![C_i]\!]_2^{tt} , \bigcap_{i \in \mathsf{ind}(C)} [\![C_i]\!]_2^{ff} \rangle .$$

Thm. 4.9 implies that $C \simeq D$ if and only if $[\![C]\!]_3 = [\![D]\!]_3$. This means that $[\![\cdot]\!]_3$ is compositional in the algebraic sense, i.e., if $[\![C]\!]_3 = [\![D]\!]_3$ then $[\![\Phi[C]]\!]_3 = [\![\Phi[D]]\!]_3$, for all contexts $\Phi[x]$. In contrast to $[\![C]\!]_1$, and indeed to the starting point $[\![C]\!]_0$, this fully–abstract interpretation $[\![C]\!]_3$ is both satisfactorily *semantic* and *finite*. It is also natural in that it realizes the obvious logical interpretation of (parallel) configurations as sequences of micro steps. Hence, the Statecharts semantics of Pnueli and Shalev is quite natural and elegant. Moreover, we believe that $[\![C]\!]_3$, in combination with Lemma 4.4, 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.

However, our semantics $[\![\cdot]\!]_3$ is not denotational, which would require that $[\![\Phi[C]]\!]_3$ is obtained directly from $[\![C]\!]_3$, when reading the syntactic operators of $\Phi[x]$ as suitable constructions in the semantic domain. As presented, the definition of $[\![C]\!]_3$ depends on the transformation of $C$ into a sum form $\sum_{i \in \mathsf{ind}(C)} C_i$, which is a purely syntactic process. For a denotational semantics, this "normalization" would have to be performed directly in the semantic domain.

### 4.4 Conservativity

This section establishes that our extension of the standard Statecharts syntax by arbitrary choices $C + D$, where $D \in \mathcal{C}$, and by the failure event $\perp$ is conservative, i.e., the full–abstraction result regarding our configuration algebra is also true for the original, more restricted Statecharts language.

Formally, let $\mathsf{C}_f$ be some distinguished subset of $\mathcal{C}$, and let $\mathsf{PC}_f$ be the parallel configurations in $\mathsf{C}_f$, i.e., $\mathsf{PC}_f =_{\mathrm{df}} \mathsf{C}_f \cap \mathcal{PC}$. In the fragments $\mathsf{C}_f$ and $\mathsf{PC}_f$, we consider two congruences $\simeq_f$ and $\simeq_f^+$, respectively, which are defined as follows:

$C \simeq_f D$ if $\forall \Phi[x] \in \mathsf{C}_f,\ E, A \subseteq_{\mathrm{fin}} \Pi.\ \Phi[C] \Downarrow_E A$ if and only if $\Phi[D] \Downarrow_E A$.
$C \simeq_f^+ D$ if $\forall P \in \mathsf{PC}_f,\ E, A \subseteq_{\mathrm{fin}} \Pi,\ b \in \mathbb{B}$.
$((C \parallel P) \Downarrow_E A$ and $\rho(C, A) = b)$ if and only if $((D \parallel P) \Downarrow_E A$ and $\rho(D, A) = b)$.

In the special case $\mathsf{C}_f = \mathcal{C}$, we simply write $\simeq$ and $\simeq^+$ instead of $\simeq_f$ and $\simeq_f^+$, respectively. The key step towards our conservativity result is to show that, when fragment $\mathsf{C}_f$ encompasses a minimum amount of discriminating contexts, the equivalence of $C \simeq^+ D$ and $C \simeq_f^+ D$ entails the equivalence of $C \simeq_f D$ and $C \simeq_f^+ D$.

LEMMA 4.10. *Let $\mathsf{C}_f$ be a fragment of $\mathcal{C}$ satisfying the following two conditions: (i) $\mathsf{C}_f$ is closed under the operations $[\cdot] + t$ and $[\cdot] \parallel t$, for all transitions $t$ in $\mathsf{C}_f$, is closed under sub–configurations, and contains at least the transitions $\cdot/A$, for all $A \subseteq_{fin} \Pi \setminus \{\bot\}$; and (ii) $C \simeq^+ D$ iff $C \simeq_f^+ D$. Then, $C \simeq_f D$ iff $C \simeq_f^+ D$.*

The proof of this lemma can be found in [Lüttgen and Mendler 2000]. A direct consequence of it, for the fragment $\mathsf{C}_f =_{\mathrm{df}} \mathcal{C}$, is Prop. 4.1 which essentially states that $C \simeq D$ is equivalent to $C \simeq^+ D$. As another consequence, consider the *standard* fragment $\mathsf{C}_s \subseteq \mathcal{C}$ of Statecharts, which consists of all configurations that (1) use the hierarchy operator only in the special form $[\cdot] + t$, for arbitrary $t \in \mathcal{T}$, and (2) do not contain the failure event $\bot$ or its negation in any transition trigger or action.

Given an arbitrary parallel configuration $P \in \mathcal{PC}$, we define its *standardization* to be the configuration $P_s \in \mathsf{PC}_s$, obtained from $P$ by dropping all transitions containing $\bot$ in their triggers or actions, as well as dropping all occurrences of $\overline{\bot}$ from the triggers of the remaining transitions. Note that $P_s$ may be the empty configuration even though $P$ is not. Obviously, by removing from $P$ transitions with $\bot$ in their actions, we lose information about the failure behavior of $P$. In fact, $P_s$ does not produce any failure due to the presence of events. For example, parallel configuration $P$ might contain transition $a/\bot$. Then, $P$ produces a failure whenever the environment offers event $a$, but $P_s$ does not since $a/\bot$ is dropped. To recover this information, we define, for every $P \in \mathcal{PC}$, a set $\mathit{fail}(P) \subseteq 2^{\Pi \setminus \{\bot\}}$ of those environments that would trigger a transition having $\bot$ in its actions and, hence, would produce a failure. More precisely, let $P_\bot \in \mathcal{PC}$ be the parallel composition of all transitions of $P$ that have $\bot$ in their action. Then, $\mathit{fail}(P) =_{\mathrm{df}} \{A \subseteq_{\mathrm{fin}} \Pi \setminus \{\bot\} \mid \rho(P_\bot, A) = tt\}$. Taking into account the sets $\mathit{fail}(P)$, we can show that this standardization $P \mapsto P_s$ does not change the communication behavior of parallel configurations.

LEMMA 4.11. *Let $C \in \mathcal{C}$, $A \subseteq_{fin} \Pi$, and $P \in \mathcal{PC}$. Then, $(C \parallel P) \Downarrow A$ iff $(C \parallel P_s) \Downarrow A$ and $A \notin \mathit{fail}(P)$.*

PROOF. Let $P \in \mathcal{PC}$ and $A \subseteq_{\mathrm{fin}} \Pi \setminus \{\bot\}$ be arbitrary. We first prove that

$$A \notin \mathit{fail}(P) \text{ implies } \mathsf{triggered}(P, A) = \mathsf{triggered}(P_s, A)\,. \tag{11}$$

Inclusion $\mathsf{triggered}(P_s, A) \subseteq \mathsf{triggered}(P, A)$ is trivial since the transitions of $P_s$ are a subset of those of $P$, possibly having an extra trigger event $\overline{\bot}$ in $P$, which does not affect their enabling as $\bot \notin A$. For the inclusion $\mathsf{triggered}(P, A) \subseteq \mathsf{triggered}(P_s, A)$,

we assume $A \notin \mathit{fail}(P)$. Let $t \in \mathsf{triggered}(P, A)$, i.e., $t$ is a transition of $P$ enabled by $A$. Since $A \notin \mathit{fail}(P)$, transition $t$ does not have event $\bot$ in its action. Similarly, it cannot have $\bot$ in its trigger; otherwise, it would not be enabled, given $\bot \notin A$. This means that $t$ must be contained in $P_s$, with any $\overline{\bot}$ in its trigger removed. In any case, transition $t$ is still enabled. Hence, $\mathsf{triggered}(P, A) \subseteq \mathsf{triggered}(P_s, A)$. Statement (11) implies the statement of the lemma. While direction "$\Longleftarrow$" is trivial, observe for direction "$\Longrightarrow$" that $P$ is a parallel context and that $(D \parallel R) \Downarrow A$ implies $\bot \notin A$ and $A \notin \mathit{fail}(R)$, for any configurations $D \in \mathcal{C}$ and $R \in \mathcal{PC}$. □

As a consequence, we now obtain the desired result for the standard fragment.

LEMMA 4.12. *Let $C, D \in \mathcal{C}$. Then, $C \simeq^{+} D$ iff $C \simeq_s^{+} D$.*

PROOF. Direction "$\Longrightarrow$" is trivial since the standard parallel contexts are just a special class of parallel contexts. For the other direction "$\Longleftarrow$", suppose $C \simeq_s^{+} D$. Let $P \in \mathcal{PC}$, $A \subseteq_{\mathrm{fin}} \Pi$, and $b \in \mathbb{B}$ be such that $(C \parallel P) \Downarrow A$ and $\rho(C, A) = b$. By direction "$\Longrightarrow$" of Lemma 4.11, $(C \parallel P_s) \Downarrow A$ and $A \notin \mathit{fail}(P)$. Since $P_s \in \mathsf{PC}_s$ and $C \simeq_s^{+} D$ we infer $(D \parallel P_s) \Downarrow A$ and $\rho(D, A) = b$. Another application of Lemma 4.11, this time direction "$\Longleftarrow$" for configuration $D$, yields $(D \parallel P) \Downarrow A$. Hence, we have shown that, for all $P \in \mathcal{PC}$, $A \subseteq_{\mathrm{fin}} \Pi$, and $b \in \mathbb{B}$,

$$((C \parallel P) \Downarrow A \text{ and } \rho(C, A) = b) \quad \text{implies} \quad ((D \parallel P) \Downarrow A \text{ and } \rho(D, A) = b)\,.$$

Since our argument is symmetric in $C$ and $D$, the other direction holds, too. □

We are now ready to summarize the conservativity properties.

THEOREM 4.13 CONSERVATIVITY. *For arbitrary $C, D \in \mathcal{C}$, the following statements are equivalent:*

$$\textit{(1)}\ C \simeq D\,, \qquad \textit{(2)}\ C \simeq^{+} D\,, \qquad \textit{(3)}\ C \simeq_s D\,, \quad \textit{and} \quad \textit{(4)}\ C \simeq_s^{+} D\,.$$

PROOF. The equivalence "(1) $\Longleftrightarrow$ (2)" follows from Lemma 4.10 for the fragment $\mathsf{C}_f =_{\mathrm{df}} \mathcal{C}$, whereas equivalence "(2) $\Longleftrightarrow$ (4)" is the statement of Lemma 4.12. Finally, equivalence "(3) $\Longleftrightarrow$ (4)" arises from specializing Lemma 4.10 to fragment $\mathsf{C}_s$, using result "(2) $\Longleftrightarrow$ (4)" and the fact that $\mathsf{C}_s$ satisfies Assumption (i) required in Lemma 4.10. □

The equivalence of $C \simeq D$ and $C \simeq_s D$ is a crucial result since it shows that there are no additional semantic distinctions introduced by our use of a more general configuration syntax. Hence, whenever we restrict ourselves to the standard fragment we obtain exactly the same compositional semantics as if we had used the restricted language in the first place. This proves our claim that our semantics is fully abstract for Statecharts and the step semantics of Pnueli and Shalev.

## 5. RELATED WORK AND POSSIBLE EXTENSIONS

This section discusses related work on Statecharts semantics and illustrates how our framework might be extended to deal with some advanced features incorporated in various popular Statecharts dialects.

### 5.1 Related Work

Our investigation focused on Pnueli and Shalev's original presentation of Statecharts and its macro–step semantics. Like Pnueli and Shalev [1991] we only consider single macro steps since it is here where the main challenge regarding a fully–abstract semantics of Statecharts lies. The elegance of their operational semantics manifests itself in the existence of an equivalent *declarative fixed point semantics*. However, as illustrated in [Pnueli and Shalev 1991], this equivalence breaks down when allowing disjunctions in transition triggers. Pnueli and Shalev observed that the configurations $(\overline{a} \vee b)/a \parallel a/b$ and $\overline{a}/a \parallel b/a \parallel a/b$ do not have the same response behavior in the declarative semantics, although they are identified in the operational semantics. This mismatch between declarative and operational semantics can now be explained in our intuitionistic framework. In Pnueli and Shalev's setting, $\overline{a} \vee b$ is classically interpreted as "*throughout the macro step, not a or b.*" In contrast, this paper's approach reads the configuration as "*throughout the macro step not a, or throughout the macro step b.*" Our stronger intuitionistic interpretation restores the coincidence of declarative and operational semantics. This assumes, of course, that the latter is adjusted accordingly, which is not difficult, however. The step procedure must only ensure that, whenever transition $(\overline{a} \vee b)/a$ is fired due to absence of $a$, event $a$ is prohibited to occur in any subsequent micro step. Our approach also suggests other extensions to larger fragments of intuitionistic logic, such as "higher–order" transitions, e.g., $(a \supset b) \supset c$, which may be explored in the future.

Our framework can also be employed for analyzing various other asynchronous Statecharts variants with global consistency. One example is the work of Maggiolo-Schettini et al. [1996], which is inspired by the process–algebraic semantics presented by Levi [1997] and by Uselton and Smolka [1994]. In their setting, and also in [Lüttgen et al. 1999; 2000], 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. The resulting semantics is specified using a notion of *compatibility* [Maggiolo-Schettini et al. 1996] which introduces a look–ahead concept for avoiding failures during the construction of macro steps.

As an example, consider the configuration $C =_{\mathrm{df}} t_1 \parallel t_2$, where $t_1 =_{\mathrm{df}} a/b$ and $t_2 =_{\mathrm{df}} \overline{b}/a$. According to Maggiolo-Schettini et al. [1996], when $C$ is evaluated in the empty environment, the response $\{a\}$ is obtained: First, transition $t_2$ fires due to the absence of event $b$, thereby producing event $a$. The presence of $a$ now satisfies the trigger of $t_1$. Its execution would introduce event $b$, whence transition $t_1$ is incompatible with $t_2$ which has fired due to the absence of event $b$. Therefore, transition $t_1$ is disabled in [Maggiolo-Schettini et al. 1996]. In Pnueli and Shalev's original semantics, however, $t_1$ is enabled with the consequence that the step construction is forced to fail. The difference between the two semantics 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 $t_2$. Thus, e.g., $t_1$ is read as "*if a becomes present then either b is asserted or b never becomes present.*" The second case "*b never becomes present*" accommodates the possibility

that $t_1$, even though its trigger $a$ is satisfied, is not taken due to an incompatibility with another transition in the environment that requires the global absence of $b$. A similar remark applies to transition $t_2$. Indeed, one can show that configuration

$$C_{enc} =_{\mathrm{df}} t_1 \parallel t_2 \equiv (a \supset (b \vee \neg b)) \wedge (\neg b \supset (a \vee \neg a))$$

possesses $\{a\}$ as a response model, in the sense of Def. 3.3, which is in accordance with the operational semantics of Maggiolo-Schettini et al. [1996]. Note that this encoding, again, crucially depends on the fact that $a \vee \neg a$ differs from *true* in intuitionistic logic. Generalizing this example, we conjecture that the transition semantics of [Lüttgen et al. 1999; 2000; Maggiolo-Schettini et al. 1996] can be captured in terms of response models by reading a transition $E/A$ as formula $E \supset (A \vee \neg A)$. Of course, our language of configurations needs to be extended to allow disjunctions as part of transition actions. We further want to remark that it is possible to translate between the two considered semantics [Maggiolo-Schettini et al. 1996; Pnueli and Shalev 1991] using our framework. For instance, the sequence model semantics of $C_{enc}$ may be captured by configuration $a/b + \overline{b}/a$. This configuration has the same operational behavior in Pnueli and Shalev's step semantics as $C$ has in [Maggiolo-Schettini et al. 1996]. Moreover, we expect that our semantics may also be useful to derive full–abstractness results for the semantics in [Maggiolo-Schettini et al. 1996] and other Statecharts semantics with global consistency. Lifting our results to sequences of macro steps should not present major difficulties when employing the standard framework of transition systems.

Other investigations into the compositionality problem of Statecharts were conducted by Uselton and Smolka [1994] who model Statecharts' macro steps by labeled transition systems in a process–algebraic style. They achieve compositionality by using partial orders on events, which encode causality information, as transition labels. As was pointed out by Levi [1997], the partial orders on events used by Uselton and Smolka are not sufficient to capture Pnueli and Shalev's semantics faithfully. Levi's semantics remedies the problem by employing partial orders on *sets* of events. Although this semantics complies with the one of Pnueli and Shalev, no full–abstraction result is presented. It should be noted that our semantics, too, uses a lattice–theoretic structure on sets of events. The elements $\langle A, L\rangle$ of $[\![C]\!]_2^{tt}$, which represent the active responses of $C$, are $(\cap, \subseteq)$ sub–lattices of $2^A$ that correspond to the transition labels in Levi's work. The main difference between our approach and the ones in [Levi 1997; Uselton and Smolka 1994] is that our lattices do not contain any negative events, whence they may be considered more semantic in nature. The precise relationship between our semantics and that of Levi [1997] still needs to be explored.

Our intuitionistic approach is also related to recent work in *synchronous languages*, especially ESTEREL [Berry 2000]. In ESTEREL, causality is traditionally treated separately from compositionality and synchrony, as part of type–checking specifications. If the (conservative) type checker finds causality to be violated, it rejects the specification under consideration. Otherwise, the specification's semantics can be determined in a very simple fashion, since one may— in contrast to Statecharts semantics— abstract from the construction details of macro steps while preserving compositionality. This was shown by Broy [1997], using a domain–theoretic account of abstracting from a sequence of micro steps to a macro step,

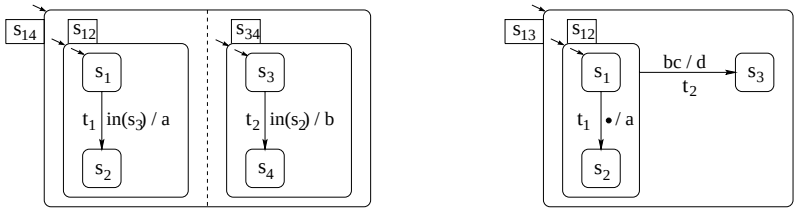


Fig. 4. Example Statecharts for illustrating state references (left) and priority concepts (right).

which was based on *streams*. The more recent Version 5 of ESTEREL, however, replaces the restrictive treatment of causality by defining a semantics via a particular Boolean logic that is *constructive* [Berry 1999], as is intuitionistic logics. The constructive semantics of ESTEREL is especially interesting since it relates to the traditional semantics for digital circuits [Berry and Sentovich 2000].

Denotational semantics and full abstraction were also studied by Huizing [1991], together with Gerth, and Huizing et al. [1988] for an early and lateron rejected Statecharts semantics [Harel et al. 1987]. In particular, that semantics does not consider global consistency, which makes their result largely incomparable to ours. Also, the abstractness result is proved with respect to a richer set of syntactic operators than we consider here. Finally, it should be mentioned that the lack of compositionality of Statecharts semantics inspired the development of new visual languages, such as *Communicating Hierarchical State Machines* [Alur et al. 1999], ARGOS [Maraninchi 1992], and RSML [Leveson et al. 1994].

## 5.2 Extensions

In the classical Statecharts language examined in this paper, three features are left out which are often adopted in other variants, such as in STATEMATE [Harel and Naamad 1996] and in UML Statecharts [Booch et al. 1998].

**State references** permit the triggering of transitions to depend on whether or not a certain parallel component is in a certain state, identified by its unique name. To cope with state references in our setting, it seems natural to introduce distinguished events $\mathsf{in}(s)$, for each state name $s$, to the event set $\Pi$ [Lüttgen et al. 2000]. Intuitively, an event $\mathsf{in}(s)$ should indicate that state $s$ is active. As an example, consider the Statechart depicted on the left in Figure 4. Since state $s_3$ is initially active, whence trigger $\mathsf{in}(s_3)$ of transition $t_1$ is satisfied, $t_1$ fires, which in turn activates state $s_2$. Thus, the trigger $\mathsf{in}(s_2)$ of $t_2$ evaluates to true, and $t_2$ fires, too. Although the idea of encoding state references via distinguished events $\mathsf{in}(s)$ seems to be intuitive, it is incompatible with the semantics of macro steps. This is because macro steps are essentially sequences of micro steps, and firing a transition means exiting the transition's source state and entering its target state. In contrast, once event $\mathsf{in}(s)$ is emitted, it is valid until the end of the macro step under construction, although state $s$ might be exited again in between.

To reflect the correct semantics of state references, one needs to introduce not the events $\mathsf{in}(s)$, but two kinds of distinguished events, $\mathsf{enter}(s)$ and $\mathsf{exit}(s)$, signaling whether state $s$ has been entered or exited, respectively [Pnueli and Shalev 1991]. Then, $\mathsf{in}(s)$ can be understood as an abbreviation of the conjunction of

$\mathsf{enter}(s)$ and $\overline{\mathsf{exit}(s)}$, as state $s$ is currently active if it was entered but has not been exited, yet. Accordingly, one may identify the extended configuration $C_{14}^{\text{ext}}$ with the (basic) configuration $C_{14} =_{\text{df}} (C_{12} \parallel C_{34}) \parallel t_{\text{in}}$, where $t_{\text{in}} =_{\text{df}} \cdot / S$ and $S =_{\text{df}} \{\mathsf{enter}(s_{14}), \mathsf{enter}(s_{12}), \mathsf{enter}(s_{34}), \mathsf{enter}(s_1), \underline{\mathsf{enter}(s_3)}\}$. Moreover, $C_{12} =_{\text{df}} C_1 =_{\text{df}} t_1$ and $C_{34} =_{\underline{\text{df}}} \underline{C_3} =_{\text{df}} t_2$ with $t_1 =_{\text{df}} \mathsf{enter}(s_3), \underline{\overline{\mathsf{exit}(s_3)}} \,/\, a, \mathsf{exit}(s_1), \mathsf{enter}(s_2)$ and $t_2 =_{\text{df}} \mathsf{enter}(s_2), \underline{\overline{\mathsf{exit}(s_2)}} \,/\, b, \mathsf{exit}(s_3), \mathsf{enter}(s_4)$. One can then apply our semantic framework to derive $C_{14} \Downarrow (\{a, b\} \cup S \cup \{\mathsf{exit}(s_1), \mathsf{enter}(s_2), \mathsf{exit}(s_3), \mathsf{enter}(s_4)\})$, as desired.

**Prioritized or–states** are a variant of the traditional types of or–states which usually give priority to transitions at higher levels in the state hierarchy, as in STATEMATE [Harel and Naamad 1996]. In UML Statecharts [Booch et al. 1998], this interpretation is reversed, i.e., transitions at lower levels in the state hierarchy have priority over ones at higher levels. In the following we adopt the former interpretation; the latter one can be dealt with in a dual fashion.

Consider the example Statechart depicted on the right in Figure 4, in its initial states. When both events $b$ and $c$ are offered by the environment, then transition $t_2$ must be fired. Although transition $t_1$ is always enabled, it is not considered since it is located on a lower hierarchy level than $t_2$. To reflect this priority semantics in our setting, one needs to add the logical conjunct $\neg(b \wedge c) \equiv \neg b \vee \neg c$ to the trigger of $t_1$, yielding the transition $(\overline{b} \vee \overline{c}) \,/\, a$. Note that this requires us to permit disjunctions in triggers, which is, however, admissible in our setting as demonstrated above.

**Interlevel transitions** are transitions which cross borderlines of states. Unfortunately, their inclusion in Statecharts prohibits the development of a compositional semantics, since the concept of interlevel transitions resembles "goto–programming" [Simons 2000]. Most importantly, interlevel transitions jeopardize a strictly structural definition of Statecharts configurations, which is a prerequisite for deriving a compositional semantics. Hence, for modeling interlevel transitions, the syntax of Statecharts must be changed so that interlevel transitions are represented by several intralevel transitions which are connected via dedicated *ports*. This can be done either explicitly as in *Communicating Hierarchical State Machines* [Alur et al. 1999], or via a synchronization scheme along the hierarchy of or–states, as in ARGOS [Maraninchi 1992].

## 6. CONCLUSIONS AND FUTURE WORK

To the best of our knowledge, this is the first paper to present a fully–abstract Statecharts semantics for the original macro–step semantics of Pnueli and Shalev [1991]. 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 in a model–theoretic fashion. Thus, our approach suggests a model–based implementation of Pnueli and Shalev's semantics, thereby eliminating the need to implement failure via backtracking. It further permits the introduction of more general trigger conditions, including disjunctions.

Regarding future work, several further theoretical investigations should be conducted. Firstly, we plan to derive a fully–abstract *denotational* semantics for Statecharts on the basis of our results. To this end, we need to find a semantic mapping that does not depend on a syntactic normalization. Secondly, the macro–step semantics for single configurations should be lifted to the full Statecharts semantics which involves sequences of macro steps. This should not prove to be difficult, however, although the consideration of advanced Statecharts features, such as *timeout events* at transitions [von der Beeck 1994], might introduce some complications. We also intend to employ our framework for developing algebraic characterizations of our step congruence and for uniformly comparing various variants of Statecharts' macro–step semantics studied in the literature [Levi 1997; Lüttgen et al. 2000; Maggiolo-Schettini et al. 1996]. Practical applications of our work include semantic–based program transformations, abstract analyses, and compositional code generation.

## ACKNOWLEDGMENTS

We would like to thank the members of the Sheffield Verification and Testing Group for several fruitful discussions on Statecharts semantics. Moreover, we are grateful to Rance Cleaveland and Scott Smolka, as well as to the anonymous referees for their valuable comments and suggestions.

## REFERENCES

Alur, R., Kannan, S., and Yannakakis, M. 1999. Communicating hierarchical state machines. In *26th Intl. Coll. on Automata, Languages and Programming (ICALP '99)*, P. van Emde Boas, J. Wiedermann, and M. Nielsen, Eds. Lect. Notes in Comp. Sci., vol. 1644. Springer-Verlag, Prague, Czech Republic, 169–178.

Berry, G. 1999. The constructive semantics of pure esterel. Draft Version 3.0. Available at http://www-sop.inria.fr/meije/Personnel/Gerard.Berry.html.

Berry, G. 2000. The foundations of esterel. In *Proof, Language and Interaction: Essays in Honour of Robin Milner*, G. Plotkin, C. Stirling, and M. Tofte, Eds. Found. of Comp. MIT Press, Cambridge, MA, USA.

Berry, G. and Sentovich, E. 2000. An implemenatation of constructive synchronous programs in polis. *Form. Meth. in Sys. Des. 17,* 2 (Oct.), 135–161.

Booch, G., Rumbaugh, J., and Jacobson, I. 1998. *The Unified Modeling Language User Guide.* Object Techn. Series. Addison Wesley Longman, Reading, MA, USA.

Broy, M. 1997. Abstract semantics of synchronous languages: The example esterel. Tech. Rep. TUM-I9706, Munich Univ. of Tech., Germany. Mar.

Damm, W., Josko, B., Hungar, H., and Pnueli, A. 1997. A compositional real-time semantics of statemate designs. In *Compositionality: The Significant Difference*, W. de Roever, H. Langmaack, and A. Pnueli, Eds. Lect. Notes in Comp. Sci., vol. 1536. Springer-Verlag, Bad Malente, Germany, 186–238.

Harel, D. 1987. Statecharts: A visual formalism for complex systems. *Sci. Comput. Program. 8*, 231–274.

Harel, D. and Naamad, A. 1996. The statemate semantics of Statecharts. *ACM Trans. Softw. Eng. 5,* 4 (Oct.), 293–333.

Harel, D., Pnueli, A., Pruzan-Schmidt, J., and Sherman, R. 1987. On the formal semantics of Statecharts. In *2nd Symp. on Logic in Computer Science (LICS '87)*. IEEE Comp. Soc. Press, Ithaca, NY, USA, 56–64.

Huizing, C. 1991. Semantics of reactive systems: Comparison and full abstraction. Ph.D. thesis, Eindhoven Univ. of Tech., The Netherlands.

Huizing, C., Gerth, R., and de Roever, W. 1988. Modeling Statecharts behavior in a fully abstract way. In *13th Coll. on Trees and Algebra in Programming (CAAP '88)*, M. Dauchet and M. Nivat, Eds. Lect. Notes in Comp. Sci., vol. 299. Springer-Verlag, Nancy, France, 271–294.

Leveson, N., Heimdahl, M., Hildreth, H., and Reese, J. 1994. Requirements specification for process-control systems. *ACM Trans. Softw. Eng. 20,* 9 (Sept.), 684–707.

Levi, F. 1997. Verification of temporal and real-time properties of Statecharts. Ph.D. thesis, Univ. of Pisa-Genova-Udine, Italy.

Lüttgen, G. and Mendler, M. 2000. The intuitionism behind Statecharts steps. Tech. Rep. 2000-28, NASA Contr. Rep. NASA/CR-2000-210302, Inst. for Comp. Appl. in Sci. and Eng., NASA Langley Research Center, VA, USA. July.

Lüttgen, G., von der Beeck, M., and Cleaveland, R. 1999. Statecharts via process algebra. In *10th Intl. Conf. on Concurrency Theory (CONCUR '99)*, J. Baeten and S. Mauw, Eds. Lect. Notes in Comp. Sci., vol. 1664. Springer-Verlag, Eindhoven, The Netherlands, 399–414.

Lüttgen, G., von der Beeck, M., and Cleaveland, R. 2000. A compositional approach to Statecharts semantics. In *8th Intl. Symp. on the Foundations of Software Engineering (FSE 2000)*. ACM Press, San Diego, CA, USA.

Maggiolo-Schettini, A., Peron, A., and Tini, S. 1996. Equivalences of Statecharts. In *7th Intl. Conf. on Concurrency Theory (CONCUR '96)*, U. Montanari and V. Sassone, Eds. Lect. Notes in Comp. Sci., vol. 1119. Springer-Verlag, Pisa, Italy, 687–702.

Maraninchi, F. 1992. Operational and compositional semantics of synchronous automaton compositions. In *3rd Intl. Conf. on Concurrency Theory (CONCUR '92)*, R. Cleaveland, Ed. Lect. Notes in Comp. Sci., vol. 630. Springer-Verlag, Stony Brook, NY, USA, 550–564.

Pnueli, A. and Shalev, M. 1991. What is in a step: On the semantics of Statecharts. In *1st Intl. Conf. on Theoretical Aspects of Computer Software (TACS '91)*, T. Ito and A. Meyer, Eds. Lect. Notes in Comp. Sci., vol. 526. Springer-Verlag, Sendai, Japan, 244–264.

Simons, A. 2000. On the compositional properties of UML Statechart diagrams. In *3rd Conf. on Rigorous Object-Oriented Methods*, A. Clark, A. Evans, and K. Lano, Eds. Electr. Workshops in Comp. British Computer Society, York, U.K., 4.1–4.19.

Uselton, A. and Smolka, S. 1994. A compositional semantics for Statecharts using labeled transition systems. In *5th Intl. Conf. on Concurrency Theory (CONCUR '94)*, B. Jonsson and J. Parrow, Eds. Lect. Notes in Comp. Sci., vol. 836. Springer-Verlag, Uppsala, Sweden, 2–17.

van Dalen, D. 1986. Intuitionistic logic. In *Handbook of Philosophical Logic*. Vol. III. Reidel, Dordrecht, The Netherlands, Chapter 4, 225–339.

von der Beeck, M. 1994. A comparison of Statecharts variants. In *3rd Intl. School and Symp. on Formal Techniques in Real-time and Fault-tolerant Systems (FTRTFT '94)*, H. Langmaack, W. de Roever, and J. Vytopil, Eds. Lect. Notes in Comp. Sci., vol. 863. Springer-Verlag, Lübeck, Germany, 128–148.

von der Beeck, M. 2000. A concise compositional Statecharts semantics definition. In *Intl. Conf. on Formal Desciption Techniques and Protocol Specification, Testing and Verification (FORTE XIII/PSTV XX 2000)*, T. Bolognesi and D. Latella, Eds. Kluwer Academic Publishers, Pisa, Italy.

Received June 2000; revised January 2001; accepted January 2001