Electronic Notes in Theoretical Computer Science 65 No. 5 (2002)
URL: http://www.elsevier.nl/locate/entcs/volume65.html 15 pages

Towards a Model–Theory for Esterel

Gerald Lüttgen [1]

Department of Computer Science, The University of Sheffield, U.K.

Michael Mendler [2]

*Fakultät für Wirtschaftsinformatik und Angewandte Informatik,
Universität Bamberg, Germany*

Abstract

Esterel is a synchronous language for reactive–systems design and builds the core of the commercial tool Esterel Studio. This paper shows how the constructive semantics of a combinational fragment of Esterel, as presented by Berry, can be derived in a model–theoretic fashion, thus complementing the existing behavioral, operational, and circuit–based approaches to Esterel semantics. Technically, Esterel programs are read as formulas in propositional intuitionistic logic, which are interpreted over simple linear Kripke structures, referred to as Gödel valuations. Esterel reactions are then characterized as specific Gödel valuations, called response models, and it is shown that the approach is compositional in the structure of Esterel programs.

The obtained results are an important step towards explaining the logic behind Esterel semantics. In addition, the intuitionistic setting advocated in this paper nicely links to Pnueli and Shalev's semantics of Harel's Statecharts, another synchronous language for reactive–systems design. This offers interesting insights into the similarities of and the differences between Esterel and Statecharts semantics.

1 Introduction

Esterel is a textual imperative language for specifying the behavior of reactive systems, developed by Berry since the 1980s [1,2]. The language provides primitives for decomposing reactions sequentially and concurrently, where concurrent reactions might involve a complex exchange of signals. The semantics of Esterel is based on the idea of *cycle–based reaction*, where first the statuses of the input signals, as defined by a system's environment, are sampled at the

[1] Email: g.luettgen@dcs.shef.ac.uk.
[2] Email: michael.mendler@wiai.uni-bamberg.de. Research support was provided under EPSRC grant GR/M99637 and by the EU Types Working Group IST–EU–29001.

beginning of each cycle, then the system's reaction in the form of the emission of further signals is determined, and finally the new signal statuses are output to the environment. The semantics of Esterel has significantly evolved over the years, and is designed around the key principles of *synchrony*, *reactivity*, *determinism*, and *causality* [1,2]. The synchrony requirement reflects the mechanism behind cycle–based reaction and is mathematically modeled via the *perfect synchrony hypothesis*. This hypothesis ensures that reactions and the propagations of signals are instantaneous, which is an idealized system behavior that is nevertheless often reflected in practice: reactive systems usually perform much faster than their environments. Determinism demands reactions to be uniquely determined by the system environment's inputs. This is a property very much desired, since nondeterministic systems are often difficult to understand; sometimes encountered system bugs might even not be reproducible. Causality refers to the requirement that the reason for a signal being emitted or not emitted in a system reaction can be traced back to the input signals provided by the environment. While this property is very natural, it is quite hard to enforce in a simple mathematical way. In earlier approaches to Esterel semantics, causality was dealt with in a preprocessing step: only Esterel programs were considered which could be shown to be causal by means of a static analysis [2]. Such static checks, however, compute approximations of causality and sometimes reject programs that are perfectly causal from a semantic point of view. In his recent draft book [1], Berry describes a much improved version of Esterel semantics that is founded on the idea of *constructiveness* and that encodes the principle of causality in a precise, not an approximative way. Berry also established the coincidence of three constructive styles of Esterel semantics, a *behavioral* or fixed–point semantics, an *operational* semantics, and a *circuit* semantics, thereby testifying to the mathematical elegance and robustness of the latest version of Esterel semantics. Today, this constructive semantics builds the core of the commercial design tool Esterel Studio, which is employed by major companies in the avionics and communications industry (`www.esterel-technologies.com`).

In this paper we present a novel model–theoretic account of Esterel semantics, for a fragment of the language concerned with instantaneous reactions. Our approach reads Esterel programs as simple propositional formulas in intuitionistic logic, which correspond to the *must* and *cannot* functions as defined in Berry's behavioral semantics [1]. These functions determine which signals must and, respectively, cannot be emitted relative to some given statuses —present or absent— of the input signals. Our propositional formulas are interpreted in an intuitionistic way over two–world linear Kripke structures [8], to which we refer as *Gödel valuations*. In this setting we obtain our two main results. We first characterize valid Esterel reactions as specific Gödel valuations that respect the principle of causality. In addition, we show that our approach is compositional in the structure of Esterel programs, which is one of the virtues of Berry's behavioral semantics.

The motivations for the suggested model–theoretic approach to Esterel semantics are threefold. To begin with, our results provide a first step towards explaining the logic behind Esterel's constructive semantics. Although Berry considers a semantics based on the three–valued Scott domain in his book, that approach leads to an algebraic rather than a logical semantics. Secondly, our intuitionistic model–theoretic approach links Esterel's semantics to the original variant of Statecharts semantics [5,6], as conceived by Pnueli and Shalev [7]. Like Esterel, Statecharts [3] is a popular language for reactive–systems design, which obeys the perfect synchrony hypothesis and causality. However, Statecharts permits nondeterminism and non–reactivity, and signal statuses might be inferred by speculation. In this light, our results suggest a way for extending Esterel by a concept of nondeterminism. This is of particular importance when interfacing Esterel with design or verification methodologies, many of which are based on abstraction or refinement techniques. Third, our setting might be used for establishing full–abstractness results for Esterel, similar to the ones obtained for Statecharts [6].

2 Esterel and its Behavioral Semantics

We first present the simple but nontrivial combinational fragment of Esterel that will be considered in the remainder and recall its constructive behavioral semantics as defined in [1]. This semantics is essentially a fixed–point semantics, which we will then characterize in terms of *separability*, a notion that is adapted from Statecharts, where it is employed for encoding causality [7].

Syntax and behavioral semantics. The fragment of Esterel we are interested in deals with instantaneous reactions, i.e., single reaction cycles. Its syntax is defined by the following BNF, where s stands for a signal name taken from some finite universe $\mathcal{S}$.

$$\begin{array}{rll} P & ::= 0 & \texttt{nothing} \\ & \mid\ !s & \texttt{emit}\ s \\ & \mid\ s{=}1?(P) & \texttt{present}\ s\ \texttt{then}\ P \\ & \mid\ s{=}0?(P) & \texttt{present}\ s\ \texttt{else}\ P \\ & \mid\ P \,|\, P & P\,||\,P \end{array}$$

In analogy to digital circuits we refer to programs in this fragment as *combinational programs*. Esterel's more general choice statement "$\texttt{present}\ s\ \texttt{then}\ P\ \texttt{else}\ Q$" can be recovered in our syntax by the term $s{=}1?(P) \mid s{=}0?(Q)$. Treating the then– and else–branches separately will prove to be notationally convenient later–on. In this paper we omit the combinational operators for sequential composition and signal definition. A consequence of this omission is that the completion codes needed in the behavioral semantics' definition for the full language [1] become obsolete.

The constructive behavioral semantics uses a fixed–point construction on so–called *partial events*. A partial event is simply a consistent set E of signal statuses of the form $s{=}\mathtt{1}$ and $s{=}\mathtt{0}$; in particular, for any signal s, set E is not allowed to contain both $s{=}\mathtt{1}$ and $s{=}\mathtt{0}$. Status $s{=}\mathtt{1}$ represents the fact that s is positively known to be present, while status $s{=}\mathtt{0}$ means that s is positively known to be absent. Signals not in E have an unknown status. A partial event E is called *complete* if it contains either $s{=}\mathtt{1}$ or $s{=}\mathtt{0}$, for every signal s. One can consider partial events as intuitionistic valuations of signals and complete events as their classical two–valued completion.

The behavior of an Esterel program P is usually studied with respect to an event E_I determining the status of all *input signals* $i \in I = \{i_1, \dots, i_n\} \subseteq \mathcal{S}$. In this paper we do away with distinguished input signals, thereby simplifying our presentation. This is possible since the behavior of P under I is equivalent to the behavior of $P \,|\, !i_{j_1} \,|\, \cdots \,|\, !i_{j_m}$, where the indexes $j_1, \dots, j_m \in \{1, \dots, n\}$ are exactly those for which $i_{j_k}{=}\mathtt{1} \in E_I$. The standard Esterel semantics, as well as the model–theoretic semantics developed here, are fully compatible with this point of view. We now reproduce Berry's definitions of the *Must* and *Cannot* functions over partial events [1], which are inductively defined as follows, where $\mathcal{S}_0$ denotes the set $\{s{=}\mathtt{0} \mid s \in \mathcal{S}\}$.

$$
\begin{array}{llll}
\mathit{Must}(0,E) & := \emptyset & \mathit{Must}(s{=}\mathtt{1}?(P),E) & := \begin{cases} \mathit{Must}(P,E) & \text{if } s{=}\mathtt{1} \in E \\ \emptyset & \text{otherwise} \end{cases} \\
\mathit{Must}(!s,E) & := \{s{=}\mathtt{1}\} & \mathit{Must}(s{=}\mathtt{0}?(P),E) & := \begin{cases} \mathit{Must}(P,E) & \text{if } s{=}\mathtt{0} \in E \\ \emptyset & \text{otherwise} \end{cases} \\
 & & \mathit{Must}(P \,|\, Q,E) & := \mathit{Must}(P,E) \cup \mathit{Must}(Q,E) \\
\mathit{Cannot}(0,E) & := \mathcal{S}_0 & \mathit{Cannot}(s{=}\mathtt{1}?(P),E) & := \begin{cases} \mathcal{S}_0 & \text{if } s{=}\mathtt{0} \in E \\ \mathit{Cannot}(P,E) & \text{otherwise} \end{cases} \\
\mathit{Cannot}(!s,E) & := \mathcal{S}_0 \setminus \{s{=}\mathtt{0}\} & \mathit{Cannot}(s{=}\mathtt{0}?(P),E) & := \begin{cases} \mathcal{S}_0 & \text{if } s{=}\mathtt{1} \in E \\ \mathit{Cannot}(P,E) & \text{otherwise} \end{cases} \\
 & & \mathit{Cannot}(P \,|\, Q,E) & := \mathit{Cannot}(P,E) \cap \mathit{Cannot}(Q,E)
\end{array}
$$

Intuitively, $\mathit{Must}(P,E)$ and $\mathit{Cannot}(P,E)$, where P is a combinational Esterel program and E is a partial event, denote the partial events including all signals that P must and cannot emit, respectively, relative to E. As expected, the *Must* and *Cannot* functions satisfy the property $\nexists s.\, s{=}\mathtt{1} \in \mathit{Must}(P,E)$ and $s{=}\mathtt{0} \in \mathit{Cannot}(P,E)$, for any P and E. Moreover, both functions are monotonic in E. With these auxiliary definitions we can now state Esterel's constructive behavioral semantics. Every program P defines a monotonic function $[\![P]\!]$ on partial events: $[\![P]\!](O) := \mathit{Must}(P,O) \cup \mathit{Cannot}(P,O)$. We say that $[\![P]\!]$ is the *response function* of P. If O is the *least fixed–point* of $[\![P]\!]$, then O is called the *response* of P, written $P \Downarrow O$. Moreover, program P is called *constructive*, if O is complete. Observe that the response $P \Downarrow O$ is on *partial* events O. The constructive behavioral semantics of Esterel, however, is only that subrelation of $\Downarrow$ where O is complete, written $P \downarrow O$.

Let us illustrate this semantics by means of an example. Consider the program $P := a{=}\mathtt{1}?(a{=}\mathtt{0}?(!b)) \mid a{=}\mathtt{0}?(!c) \mid b{=}\mathtt{0}?(!d)$. Although in this example none of the signals a, b, c, d has an unguarded emit, it still produces the constructive response $P \downarrow \{a{=}\mathtt{0}, b{=}\mathtt{0}, c{=}\mathtt{1}, d{=}\mathtt{1}\}$. Here and elsewhere we omit from the response all absent signals that do not syntactically occur in the program at hand. The first iteration of the fixed–point construction gives $[\![P]\!](\emptyset) = \{a{=}\mathtt{0}\}$ since P does not contain an emit statement for signal a, i.e., $Cannot(P, \emptyset) = \{a{=}\mathtt{0}\}$. Then, the second iteration decides the two left most signal guards and identifies P with $!c \mid b{=}\mathtt{0}?(!d)$, which produces $[\![P]\!](\{a{=}\mathtt{0}\}) = \{a{=}\mathtt{0}, b{=}\mathtt{0}, c{=}\mathtt{1}\}$. A third iteration yields $[\![P]\!](\{a{=}\mathtt{0}, b{=}\mathtt{0}, c{=}\mathtt{1}\}) = \{a{=}\mathtt{0}, b{=}\mathtt{0}, c{=}\mathtt{1}, d{=}\mathtt{1}\}$, the desired fixed–point.

The example demonstrates two salient features of the constructive semantics, which deserve to be highlighted. Firstly, the fixed–point construction corresponds to the derivation of logical consequences regarding the presence and absence of signals. This deductive closure implements a causality chain of abstract signal propagations. Only those facts that can positively be determined from the specification of the system in finitely many steps are considered in the final response. Secondly, there is an asymmetry in the treatment of positive and negative signal facts. While the presence of signals is always derived from emit statements explicitly contained in the program text, the absence of a signal is inferred indirectly from the absence of emits. This amounts to a form of default assumption that is also known from Statecharts, namely that signals are assumed to be absent whenever it is "safe" to do so [3,7]. Both languages, however, differ in what they consider "safe"; more will be said about this in Sec. 4. In the above example, a is considered absent outright since it positively cannot be emitted by the program. Moreover, b is absent since $a{=}\mathtt{0}$, and thus the emit $!b$ in $a{=}\mathtt{1}?(a{=}\mathtt{0}?(!b))$ is positively not reachable.

Inseparability and admissibility. In analogy to Pnueli and Shalev's declarative semantics for Statecharts [7] we define a notion of inseparability. It provides for an alternative characterization of the minimality condition of the least fixed–point of $[\![P]\!]$, which will be useful for our model–theoretical analysis. A fixed–point O is called *inseparable* for P if $[\![P]\!](O') \cap (O \setminus O') \neq \emptyset$, for all $O' \subsetneq O$. Hence, O is inseparable if it does not contain any proper subset O' that is closed under $[\![P]\!]$. Informally, this requires O to be internally causal with respect to the response function, i.e., every signal status in O has a causal justification in terms of iterated applications of $[\![P]\!]$ (cf. [6]).

Proposition 2.1 *Let O be a fixed–point of $[\![P]\!]$, for some Esterel program P. Then, O is inseparable for P if and only if O is the least fixed–point of $[\![P]\!]$.*

Proof. For direction ($\Rightarrow$), suppose that O is an inseparable fixed–point and that O' is another fixed–point. Assume further $O \not\subseteq O'$, i.e., $O \cap O' \subsetneq O$. Then, because of the inseparability of O, there exists some $s \in O \setminus (O \cap O') = O \setminus O'$ with $s \in [\![P]\!](O \cap O')$. Since $[\![P]\!]$ is monotonic, $[\![P]\!](O \cap O') \subseteq [\![P]\!](O') = O'$. Hence, we derive $(O \setminus O') \cap O' \neq \emptyset$, which is a contradiction.

For direction ($\Leftarrow$), suppose that O is the least fixed point and that $O' \subsetneq O$ is a proper subset. Assume further that O is obtained by the approximation chain $[\![P]\!]^0(\emptyset) \subsetneq [\![P]\!]^1(\emptyset) \subsetneq [\![P]\!]^2(\emptyset) \subsetneq \cdots \subsetneq [\![P]\!]^n(\emptyset) = O$, where $[\![P]\!]^0(\emptyset) = \emptyset$ and $[\![P]\!]^{i+1}(\emptyset) = [\![P]\!]([\![P]\!]^i(\emptyset))$. Let k be the largest index with $[\![P]\!]^k(\emptyset) \subseteq O'$. Then, $0 \leq k < n$ and $[\![P]\!]^{k+1}(\emptyset) \cap (O \setminus O') \neq \emptyset$. By monotonicity, $[\![P]\!]^k(\emptyset) \subseteq O'$ implies $[\![P]\!]^{k+1}(\emptyset) \subseteq [\![P]\!](O')$. Thus, there exists some $s \in O \setminus O'$ such that $s \in [\![P]\!](O')$. But this implies that O is inseparable, as desired. $\Box$

Following Pnueli and Shalev's terminology we call a partial event O *admissible* for P, if O is an inseparable fixed–point of $[\![P]\!]$. Hence, by Prop. 2.1, admissibility for Esterel coincides with the least fixed–point property. The notion of admissibility can also be used for non–monotonic response functions such as those involved in Statecharts, where least fixed–points do not always exist [7].

3 A Model–Theoretic Semantics for Esterel

In this section we give a model–theoretic characterization of the behavioral semantics of combinational Esterel programs. First, such programs are read as formulas in propositional logic, essentially by translating the *Must* and *Cannot* functions into predicates. These formulas are then interpreted in the style of intuitionistic logics, over simple linear Kripke structures to which we refer as *Gödel valuations*.

Intuitionistic logic translation. We associate with each combinational program P and each signal s two predicates $\mathtt{Must}(P, s)$ and $\mathtt{Cannot}(P, s)$, whose intuitionistic model–theoretic semantics precisely captures the *Must* and *Cannot* functions. The atomic propositions employed in these predicates, besides *true* and *false*, are the signal statuses $s{=}\mathtt{1}$ and $s{=}\mathtt{0}$, with the obvious interpretations. We start off with the $\mathtt{Must}(P, s)$ predicate, for a signal s, which is defined along the structure of P. Intuitively, $\mathtt{Must}(P, s)$ should hold exactly if P must emit signal s, i.e., s is driven $\mathtt{1}$ in P and hence the statement $s{=}\mathtt{1}$ becomes true.

$$\mathtt{Must}(0, s) := \mathit{false} \qquad \mathtt{Must}(a{=}\mathtt{1}?(P), s) := a{=}\mathtt{1} \wedge \mathtt{Must}(P, s)$$

$$\mathtt{Must}(!a, s) := \begin{cases} \mathit{true} & \text{if } a = s \\ \mathit{false} & \text{otherwise} \end{cases} \qquad \begin{aligned} \mathtt{Must}(a{=}\mathtt{0}?(P), s) &:= a{=}\mathtt{0} \wedge \mathtt{Must}(P, s) \\ \mathtt{Must}(P \,|\, Q, s) &:= \mathtt{Must}(P, s) \vee \mathtt{Must}(Q, s) \end{aligned}$$

Obviously, $\mathtt{Must}(P, s)$ does not say anything about when s is driven $\mathtt{0}$, i.e., when $s{=}\mathtt{0}$ should be true. Because of the asymmetry between $\mathtt{1}$ and $\mathtt{0}$ in Esterel, this needs some care. In contrast to $\mathtt{1}$, the signal value $\mathtt{0}$ is a weak kind of value, in the sense that s is held at $\mathtt{0}$ only in so far as neither P nor its environment emits s. In other words, $\mathtt{0}$ is a *default value* only. For this reason we cannot use the validity of $s{=}\mathtt{0}$ directly in order to express that s is kept at $\mathtt{0}$. For if our logical specification of P would allow us to infer $s{=}\mathtt{0}$ in some situation, then value $\mathtt{0}$ could no longer be overridden by some emit, since

$s{=}0 \wedge s{=}1$ is logically inconsistent. However, we can define a weaker "default pull–down" of s by the formula $s{\approx}0 := \neg s{=}1 \supset s{=}0$, where $\supset$ stands for logical implication. It states that if $\neg s{=}1$ is true, i.e., if we are positively sure that s will never be emitted, then $s{=}0$ is true. Otherwise, nothing is known about the status of s. Note that, while $s{=}0 \wedge s{=}1$ is inconsistent, $s{\approx}0 \wedge s{=}1$ is equivalent to $s{=}1$, as desired. Thus, a weak 0 still permits s to be emitted.

We now turn our attention to the $\mathtt{Cannot}(P, s)$ predicate whose definition requires us to decide in which situations one may specify a default pull–down of s. If we simply specified $s{\approx}0$ for any signal s, then $s{=}0$ is equivalent to $\neg s{=}1$, i.e., a signal is absent if and only if it will never become present, which would essentially give us a Statecharts–like semantics. Such an approach, however, rules out the possibility that a signal value is truly undefined, i.e., neither $s{=}1$ nor $s{=}0$ is valid. Yet, the eminent truth–value gap is an essential feature of Esterel which reflects its circuit semantics [1]; there one needs to account for subtle electrical phenomena, such as meta–stability and signal oscillations, which can occur in synchronous circuits with asynchronous feedback. In Esterel semantics, one may only conclude that signal s is 0 when P cannot emit s, which is stronger than saying that P just happens not to emit s. The predicate $\mathtt{Cannot}(P, s)$ is the formalization of this stronger statement; it is defined along the structure of P.

$$\mathtt{Cannot}(0, s) := \mathit{true}$$

$$\mathtt{Cannot}(!a, s) := \begin{cases} \mathit{true} & \text{if } a \neq s \\ \mathit{false} & \text{otherwise} \end{cases}$$

$$\mathtt{Cannot}(a{=}1?(P), s) := a{=}0 \;\vee\; \mathtt{Cannot}(P, s)$$

$$\mathtt{Cannot}(a{=}0?(P), s) := a{=}1 \;\vee\; \mathtt{Cannot}(P, s)$$

$$\mathtt{Cannot}(P \,|\, Q, s) := \mathtt{Cannot}(P, s) \wedge \mathtt{Cannot}(Q, s)$$

Then, the translation $\mathtt{Spec}(P)$ of a combinational Esterel program P into propositional logic simply is

$$\mathtt{Spec}(P) := \bigwedge_{s \in S} (\mathtt{Must}(P, s) \supset s{=}1) \;\wedge\; (\mathtt{Cannot}(P, s) \supset s{\approx}0)\,.$$

Before formally defining our model–theoretic semantics we consider a simple example: $P^* \;:=\; s_1{=}0?(!s_2) \mid !s_2$. According to the above definitions we derive the following propositional formula for $\mathtt{Spec}(P^*)$, considering only those signals that actually occur in P^*:

$$\begin{aligned}\mathtt{Spec}(P^*) = {} & (((s_1{=}0 \;\wedge\; \mathit{false}) \;\vee\; \mathit{false}) \supset s_1{=}1) \wedge (((s_1{=}1 \;\vee\; \mathit{true}) \;\wedge\; \mathit{true}) \supset s_1{\approx}0) \\ & \wedge\; (((s_1{=}0 \;\wedge\; \mathit{true}) \;\vee\; \mathit{true}) \supset s_2{=}1) \wedge (((s_1{=}1 \;\vee\; \mathit{false}) \;\wedge\; \mathit{false}) \supset s_2{\approx}0)\end{aligned}$$

In the spirit of model–theoretic semantics, one would first consider the models of $\mathtt{Spec}(P^*)$ according to classical propositional logic. In this case one would obtain the classical models $\{s_1{=}0, s_2{=}1\}$ and $\{s_1{=}1, s_2{=}1\}$. However, only the former describes a valid response in Esterel. The latter model's conclusion $s_1{=}1$ is not causally justified; it seems to come from nowhere. Note that the classical model $\{s_1{=}1, s_2{=}1\}$ is also *minimal* since no proper subset is a classical model. Hence, the classical logical semantics of our specification $\mathtt{Spec}(P^*)$ is not expressive enough for explaining the Esterel semantics of P^*. In the remainder we show that intuitionistic logic, with its richer model structure, is suited to identify those classical models of $\texttt{Spec}(P^*)$ that indeed correspond to valid Esterel responses.

Intuitionistic semantics and Gödel valuations. The structures we consider for evaluating our propositional formulas intuitionistically are linear Kripke structures, of length two, over partial events. We refer to these structures as *Gödel valuations*, since Gödel was the first to study this class of structures as possible truth values for intuitionistic logic. More precisely, a Gödel valuation is a pair (E_1, E_2) of partial events such that $E_1 \subseteq E_2$. Intuitively, (E_1, E_2) validates $s{=}\texttt{1}$ if and only if $s{=}\texttt{1} \in E_1$, and it validates $s{=}\texttt{0}$ if and only if $s{=}\texttt{0} \in E_1$. The second component E_2 is used for interpreting negation: (E_1, E_2) validates $\neg s{=}\texttt{1}$ if and only if $s{=}\texttt{1} \notin E_2$, and (E_1, E_2) validates $\neg s{=}\texttt{0}$ if and only if $s{=}\texttt{0} \notin E_2$. Then, (E_1, E_2) is a model of $\texttt{Spec}(P)$, written $(E_1, E_2) \models \texttt{Spec}(P)$, if (E_1, E_2) validates formula $\texttt{Spec}(P)$ in the intuitionistic sense [8]. Formally, for a monotonically increasing sequence of partial events $K = (E_1, E_2, \ldots, E_n)$ and index $1 \leq i \leq n$, we define the intuitionistic validity of some formula ϕ in K at index i along the structure of ϕ as follows:

$K, i \models \mathit{true}$	always	$K, i \models \neg\phi$	iff $K, n \not\models \phi$
$K, i \models \mathit{false}$	never	$K, i \models \phi \wedge \psi$	iff $K, i \models \phi$ and $K, i \models \psi$
$K, i \models s{=}\texttt{1}$	iff $s{=}\texttt{1} \in E_i$	$K, i \models \phi \vee \psi$	iff $K, i \models \phi$ or $K, i \models \psi$
$K, i \models s{=}\texttt{0}$	iff $s{=}\texttt{0} \in E_i$	$K, i \models \phi \supset \psi$	iff $\forall j \geq i.\ K, j \models \phi$ implies $K, j \models \psi$.

Then, $K \models \phi$ if $K, 1 \models \phi$. This definition implies that all Gödel valuations satisfy $\neg(s{=}\texttt{1} \wedge s{=}\texttt{0})$, for any signal s. An important special case is when both components are identical, i.e., $E_1 = E_2$. Then (E_1, E_2) also satisfies the classical axioms of the Excluded Middle, $s{=}\texttt{1} \vee \neg s{=}\texttt{1}$ and $s{=}\texttt{0} \vee \neg s{=}\texttt{0}$. Therefore, we call such valuations *classical*. Another special case of a Gödel valuation occurs if the second component E_2 is a complete event, i.e., if for all signals s, either $s{=}\texttt{1} \in E_2$ or $s{=}\texttt{0} \in E_2$. Then, we have $(E_1, E_2) \models \neg\neg(s{=}\texttt{1} \vee s{=}\texttt{0})$ which means that s is eventually driven to either $\texttt{1}$ or $\texttt{0}$. When $(E_1, E_2) \models \phi$, we call (E_1, E_2) a Gödel model of ϕ. Having formally defined the semantics we may simplify the propositional formula $\texttt{Spec}(P^*)$ of our example program P^*. Using $\equiv$ to denote semantical equivalence we find $\texttt{Spec}(P^*) \ \equiv\ s_1{\approx}\texttt{0} \wedge s_2{=}\texttt{1}$. It is easy to check that the Gödel valuations $(\{s_1{=}\texttt{0}, s_2{=}\texttt{1}\}, \{s_1{=}\texttt{0}, s_2{=}\texttt{1}\})$, $(\{s_2{=}\texttt{1}\}, \{s_1{=}\texttt{1}, s_2{=}\texttt{1}\})$, and $(\{s_1{=}\texttt{1}, s_2{=}\texttt{1}\}, \{s_1{=}\texttt{1}, s_2{=}\texttt{1}\})$ are precisely the two–world models (sequences of length 2) of $\texttt{Spec}(P^*)$. Both $\{s_1{=}\texttt{0}, s_2{=}\texttt{1}\}$ and $\{s_1{=}\texttt{1}, s_2{=}\texttt{1}\}$ are classical models (sequences of length 1) of $\texttt{Spec}(P^*)$, but only one of them, $\{s_1{=}\texttt{0}, s_2{=}\texttt{1}\}$, is an actual response of P.

We conclude this section by considering some of the illuminating examples given in Berry's book [1]. For each example, we state its corresponding simplified propositional formula as well as the formula's Gödel models, relative to the domain of signal names occurring in the example program.

- $P_0 := s{=}\mathtt{1}?(!s)\,|\,s{=}\mathtt{0}?(!s)$:

$$\begin{aligned}\mathtt{Spec}(P_0) &\equiv ((s{=}\mathtt{1} \vee s{=}\mathtt{0}) \supset s{=}\mathtt{1}) \wedge ((s{=}\mathtt{0} \wedge s{=}\mathtt{1}) \supset s{\approx}\mathtt{0}) \\ &\equiv ((s{=}\mathtt{1} \vee s{=}\mathtt{0}) \supset s{=}\mathtt{1}) \wedge (\mathit{false} \supset s{\approx}\mathtt{0}) \\ &\equiv s{=}\mathtt{0} \supset s{=}\mathtt{1} \;\equiv\; \neg s{=}\mathtt{0} \\ \mathit{Models} &: (\emptyset,\emptyset),\ (\emptyset,\{s{=}\mathtt{1}\}),\ (\{s{=}\mathtt{1}\},\{s{=}\mathtt{1}\})\end{aligned}$$

- $P_3 \;:=\; s{=}\mathtt{0}?(!s)$:

$$\begin{aligned}\mathtt{Spec}(P_3) &\equiv (s{=}\mathtt{0} \supset s{=}\mathtt{1}) \wedge (s{=}\mathtt{1} \supset s{\approx}\mathtt{0}) \;\equiv\; \neg s{=}\mathtt{0} \wedge \mathit{true} \;\equiv\; \neg s{=}\mathtt{0} \\ \mathit{Models} &: (\emptyset,\emptyset),\ (\emptyset,\{s{=}\mathtt{1}\}),\ (\{s{=}\mathtt{1}\},\{s{=}\mathtt{1}\})\end{aligned}$$

- $P_4 \;:=\; s{=}\mathtt{1}?(!s)$:

$$\begin{aligned}\mathtt{Spec}(P_4) &\equiv (s{=}\mathtt{1} \supset s{=}\mathtt{1}) \wedge (s{=}\mathtt{0} \supset s{\approx}\mathtt{0}) \;\equiv\; \mathit{true} \wedge \mathit{true} \;\equiv\; \mathit{true} \\ \mathit{Models} &: (\emptyset,\emptyset),\ (\emptyset,\{s{=}\mathtt{1}\}),\ (\emptyset,\{s{=}\mathtt{0}\}),\ (\{s{=}\mathtt{1}\},\{s{=}\mathtt{1}\}),\ (\{s{=}\mathtt{0}\},\{s{=}\mathtt{0}\})\end{aligned}$$

- $P_6 \;:=\; s_1{=}\mathtt{1}?(!s_2)\,|\,s_2{=}\mathtt{1}?(!s_1)$:

$$\begin{aligned}\mathtt{Spec}(P_6) &\equiv (s_1{=}\mathtt{1} \supset s_2{=}\mathtt{1}) \wedge (s_1{=}\mathtt{0} \supset s_2{\approx}\mathtt{0}) \wedge \\ &\qquad (s_2{=}\mathtt{1} \supset s_1{=}\mathtt{1}) \wedge (s_2{=}\mathtt{0} \supset s_1{\approx}\mathtt{0}) \\ \mathit{Models} &: (\emptyset,\emptyset),\ (\emptyset,\{s_1{=}\mathtt{1},s_2{=}\mathtt{1}\}),\ (\{s_1{=}\mathtt{1},s_2{=}\mathtt{1}\},\{s_1{=}\mathtt{1},s_2{=}\mathtt{1}\}), \\ &\qquad (\emptyset,\{s_1{=}\mathtt{0},s_2{=}\mathtt{0}\}),\ (\{s_1{=}\mathtt{0},s_2{=}\mathtt{0}\},\{s_1{=}\mathtt{0},s_2{=}\mathtt{0}\})\end{aligned}$$

We now formally state that the `Must` and `Cannot` predicates correctly encode the *Must* and *Cannot* functions, as suggested in the previous section.

Proposition 3.1 *Let (E',E) be a Gödel valuation, P a combinational program, and $s \in \mathcal{S}$.*

(i) $(E',E) \models \mathtt{Must}(P,s)$ *if and only if* $s{=}\mathtt{1} \in \mathit{Must}(P,E')$.

(ii) $(E',E) \models \mathtt{Cannot}(P,s)$ *if and only if* $s{=}\mathtt{0} \in \mathit{Cannot}(P,E')$.

The proofs of both statements of this proposition are not difficult and proceed by induction on the structure of P.

Model–theoretic characterization. As demonstrated earlier, not every (minimal) classical model of the propositional formula $\mathtt{Spec}(P)$ corresponds to a valid response of P according to Esterel's behavioral semantics. This is due to the fact that $\mathtt{Spec}(P)$ implicitly contains negations in the propositions $s{\approx}\mathtt{0}$. The right notion is that of a *response model*, which turns out to characterize exactly the desired Esterel responses.

Definition 3.2 Let P be a combinational Esterel program and E be a partial event. Then, E is a *response model* of $\mathtt{Spec}(P)$ if (1) $(E,E) \models \mathtt{Spec}(P)$, i.e., E is a classical model of $\mathtt{Spec}(P)$, and if (2) $E' = E$, for all Gödel valuations (E',E) with $(E',E) \models \mathtt{Spec}(P)$.

Note that this definition heavily borrows from our intuitionistic interpretation of $\mathtt{Spec}(P)$ and is adapted from an earlier paper by the authors on the semantics of Statecharts [6]. It guarantees that the considered models are not only classical models but also respect the principle of causality. In order to see this, consider a Gödel evaluation (E', E) such that $(E', E) \models \mathtt{Spec}(P)$. Intuitively, if $E' \neq E$, then the proper inclusion $E' \subsetneq E$ corresponds to a non–causal reaction in the construction of E, implying that some of the additional signal statuses in $E \setminus E'$ have been introduced due to some external effect and are not solely causally dependent on the ones in E'. On the other hand, if there is no Gödel valuation ending in E other than (E, E) itself, then all signal statuses in E must be causally justified. We should point out again that without negations in $\mathtt{Spec}(P)$ the notions of response model and minimal classical model coincide. The importance of stating signal absence as $s{\approx}\mathtt{0}$ will be further highlighted in Sec. 4.

For example, the Gödel valuation $(\{s_2{=}\mathtt{1}\}, \{s_1{=}\mathtt{1}, s_2{=}\mathtt{1}\})$ is an intuitionistic model of $\mathtt{Spec}(P^*)$, for our program $P^* = s_1{=}0?(!s_2) \mid !s_2$, which is a witness to the fact that $\{s_1{=}\mathtt{1}, s_2{=}\mathtt{1}\}$, although a minimal classical model, is *not* a response model. Indeed, Esterel's declarative semantics rejects the emission of s_1 since it is not causally justified. The assertion of signal s_1 cannot be inferred from the partial event $\{s_2{=}\mathtt{1}\}$. On the other hand, $\{s_1{=}\mathtt{0}, s_2{=}\mathtt{1}\}$ is a response model for $\mathtt{Spec}(P^*)$, and it is as well a valid response in Esterel. Similarly, one can check that only the empty set is a response model of P_0, P_3, P_4, and P_6. Since the response $\emptyset$ is not complete, these programs are rejected by the Esterel compiler.

Theorem 3.3 (Characterization) *Let P be a combinational program and O be a partial event. Then, $P \Downarrow O$ iff O is a response model of* $\mathtt{Spec}(P)$.

Thus, O is a constructive Esterel response for P, i.e., $P \downarrow O$, if and only if O is complete and a response model of $\mathtt{Spec}(P)$.

Proof. By Prop. 2.1, it is sufficient to prove that O is a response model of $\mathtt{Spec}(P)$ if and only if O is admissible for P. We start off with the direction "*response model $\Rightarrow$ admissible*". Given a response model O of $\mathtt{Spec}(P)$ we prove that O is admissible by showing the following:

(i) $s{=}\mathtt{1} \in O$ implies $s{=}\mathtt{1} \in \mathit{Must}(p, O)$ (iii) $[\![P]\!](O) \subseteq O$

(ii) $s{=}\mathtt{0} \in O$ implies $s{=}\mathtt{0} \in \mathit{Cannot}(p, O)$ (iv) O is inseparable for P

From Statements (i) and (ii) we get $O \subseteq [\![P]\!](O)$, which together with Statement (iii) shows that O is a fixed–point of $[\![P]\!]$. Note that Statements (i)–(iv) are equivalent to O being admissible for P, which in turn is equivalent, by Prop. 2.1, to O being the least fixed–point of $[\![P]\!]$.

(i) Let $s{=}\mathtt{1} \in O$ and $O' := O \setminus \{s{=}\mathtt{1}\}$. Since O is a response model we know $(O, O) \models \mathtt{Spec}(P)$ and $(O', O) \not\models \mathtt{Spec}(P)$. It is not difficult to show that the assumption $(O, O) \models \mathtt{Spec}(P)$, and thus $(O, O) \models \bigwedge_a \mathtt{Cannot}(P, a) \supset a{\approx}\mathtt{0}$, implies $(O', O) \models \bigwedge_a \mathtt{Cannot}(P, a) \supset a{\approx}\mathtt{0}$ as well. This is due to the fact that the difference between O' and O is a positive signal $s{=}\mathtt{1}$ and that this difference does not change validity of any $a{\approx}\mathtt{0}$ predicate, and that $\mathtt{Cannot}(P, a)$ can only become false, so the implication $\mathtt{Cannot}(P, a) \supset a{\approx}\mathtt{0}$ can only become "more true.” Hence, we must have $(O', O) \not\models \mathtt{Must}(P, a) \supset a{=}\mathtt{1}$ for some signal a, and this can only be $a = s$. This means $(O', O) \models \mathtt{Must}(P, s)$. From Prop. 3.1(i) we conclude $s{=}\mathtt{1} \in \mathit{Must}(P, O') \subseteq \mathit{Must}(P, O)$.

(ii) Here we are looking at a negative signal $s{=}\mathtt{0} \in O$, which we remove in $O' := O \setminus \{s{=}\mathtt{0}\}$. Since O is a response model, $(O', O) \not\models \mathtt{Spec}(P)$. The only possibility for this to be the case is if $(O', O) \models \mathtt{Cannot}(P, s)$ and $(O', O) \not\models s{\approx}\mathtt{0}$. This is due to the fact that none of the implications $\mathtt{Must}(P, a) \supset a{=}\mathtt{1}$, for any signal a, and none of the implications $\mathtt{Cannot}(P, a) \supset a{\approx}\mathtt{0}$, for any signal $a \neq s$, can become false in reducing O to O' by removing the negative signal status $s{=}\mathtt{0}$ from O. But $(O', O) \models \mathtt{Cannot}(P, s)$ implies by Prop. 3.1(ii) $s{=}\mathtt{0} \in \mathit{Cannot}(P, O') \subseteq \mathit{Cannot}(P, O)$.

(iii) Let $s{=}\mathtt{1} \in [\![P]\!](O)$, i.e., $s{=}\mathtt{1} \in \mathit{Must}(P, O)$. Apply Prop. 3.1(i) with $E' = E = O$ to derive $(O, O) \models \mathtt{Must}(P, s)$. Since $(O, O) \models \bigwedge_a \mathtt{Must}(P, a) \supset a{=}\mathtt{1}$, this implies $(O, O) \models s{=}\mathtt{1}$, whence $s{=}\mathtt{1} \in O$. Further, let $s{=}\mathtt{0} \in [\![P]\!](O)$, i.e., $s{=}\mathtt{0} \in \mathit{Cannot}(P, O)$. From Prop. 3.1(ii) we get $(O, O) \models \mathtt{Cannot}(P, s)$. Since by assumption $(O, O) \models \bigwedge_a \mathtt{Cannot}(P, a) \supset a{\approx}\mathtt{0}$, this implies $(O, O) \models s{\approx}\mathtt{0}$. Hence, $s = b \in O$, for some $b \in \{0, 1\}$. Now consider $O' := O \setminus \{s = b\}$. Then, $(O', O) \not\models \mathtt{Spec}(P)$ as O is a response model. But this must be because $(O', O) \not\models \mathtt{Cannot}(P, s) \supset s{\approx}\mathtt{0}$ since we must have $(O', O) \models \mathtt{Must}(P, s) \supset s{=}\mathtt{1}$. For otherwise, by Prop. 3.1(i), $s{=}\mathtt{1} \in \mathit{Must}(P, O)$ which contradicts $s{=}\mathtt{0} \in \mathit{Cannot}(P, O)$. Now, $(O', O) \not\models \mathtt{Cannot}(P, s) \supset s{\approx}\mathtt{0}$ implies $(O', O) \not\models s{\approx}\mathtt{0}$, which can only be if $b = 0$. Thus, $s{=}\mathtt{0} \in O$ as desired.

(iv) To show that O is inseparable, let $O' \subsetneq O$ be given. Because O is a response model, $(O', O) \not\models \mathtt{Spec}(P)$. Suppose then, $(O', O) \not\models \bigwedge_a \mathtt{Must}(P, a) \supset a{=}\mathtt{1}$. Since $(O, O) \models \bigwedge_a \mathtt{Must}(P, a) \supset a{=}\mathtt{1}$, Prop. 3.1(i) implies there exists some $s{=}\mathtt{1} \notin O'$ such that $s{=}\mathtt{1} \in \mathit{Must}(P, O') \subseteq [\![P]\!](O')$. Furthermore, by monotonicity of *Must*, we have $s{=}\mathtt{1} \in \mathit{Must}(P, O)$. By another application of Prop. 3.1(i) then, we infer $s{=}\mathtt{1} \in O$. This shows that $s{=}\mathtt{1} \in [\![P]\!](O') \cap (O \setminus O')$. It remains to consider the case $(O', O) \not\models \mathtt{Cannot}(P, s) \supset s{\approx}\mathtt{0}$, for some s. Since $(O, O) \models \mathtt{Cannot}(P, s) \supset s{\approx}\mathtt{0}$, this can only be because $(O', O) \models \mathtt{Cannot}(P, s)$ and $s{=}\mathtt{0} \in O \setminus O'$; this follows from the intuitionistic semantics. The former implies $s{=}\mathtt{0} \in \mathit{Cannot}(P, O') \subseteq [\![P]\!](O')$ by Prop. 3.1(ii). So, in the second case, too, we find that $[\![P]\!](O') \cap (O \setminus O') \neq \emptyset$.

We now prove direction “*admissible* $\Rightarrow$ *response model*”. Let O be admissible for P, i.e. $O = [\![P]\!](O)$ and $[\![P]\!](O') \cap (O' \setminus O) \neq \emptyset$, for all $O' \subsetneq O$. We claim that O is a response model of $\mathtt{Spec}(P)$, i.e., $(O, O) \models \mathtt{Spec}(P)$ and $(O', O) \not\models \mathtt{Spec}(P)$, for all $O' \subsetneq O$.

First, let us check that $(O, O) \models \mathtt{Spec}(P)$. It is easy to show that $(O, O) \models \mathtt{Cannot}(P, s) \supset s{\approx}\mathtt{0}$, for all signals s. For if $(O, O) \models \mathtt{Cannot}(P, s)$, then $s{=}\mathtt{0} \in \mathit{Cannot}(P, O)$ by Prop. 3.1(ii). Thus, $s{=}\mathtt{0} \in [\![P]\!](O) = O$, whence $(O, O) \models s{\approx}\mathtt{0}$. Similarly, $(O, O) \models \mathtt{Must}(P, s) \supset s{\approx}\mathtt{0}$, for all signals s: by Prop. 3.1(ii), the premise $(O, O) \models \mathtt{Must}(P, s)$ implies $s{=}\mathtt{1} \in \mathit{Must}(P, O) \subseteq [\![P]\!](O)$. Since $O = [\![P]\!](O)$, we have $s{=}\mathtt{1} \in O$ and thus $(O, O) \models s{=}\mathtt{1}$.

Next, let $O' \subsetneq O$ be given. Because of the property of admissibility,

$[\![P]\!](O')\cap(O\setminus O')$ is nonempty. Suppose there is some $s{=}1 \in [\![P]\!](O')\cap(O\setminus O')$. Then, $s{=}1 \in \mathit{Must}(P,O')$ and $s{=}1 \in (O \setminus O')$, whence, by Prop. 3.1(i), $(O',O) \models \mathtt{Must}(P,s)$. Since $s{=}1 \notin O'$, we have $(O',O) \not\models s{=}1$ and thus $(O',O) \not\models \mathtt{Spec}(P)$. On the other hand, suppose there is some $s{=}0 \in [\![P]\!](O')\cap (O \setminus O')$, i.e., $s{=}0 \in \mathit{Cannot}(P,O')$, and $s{=}0 \in O \setminus O'$. The former implies $(O',O) \models \mathtt{Cannot}(P,s)$ by Prop. 3.1(ii). The latter implies $(O',O) \not\models s{\approx}0$. Hence, $(O',O) \not\models \mathtt{Spec}(P)$, and O is a response model of $\mathtt{Spec}(P)$. □

A note on compositionality. We showed above how to derive a propositional formula $\mathtt{Spec}(P)$ for a given combinational program P. This was done with the help of the predicates $\mathtt{Must}$ and $\mathtt{Cannot}$, both of which are defined via structural induction on P, which lead to the logical specification $\mathtt{Spec}(P,s) := \mathtt{Must}(P,s) \supset s{=}1 \wedge \mathtt{Cannot}(P,s) \supset s{\approx}0$, for every signal s. The formula $\mathtt{Spec}(P,s)$ itself, however, is *not* declared directly along the structure of P, yet. Here, we show that $\mathtt{Spec}(P,s)$ can indeed be defined structurally for the constructive responses, under the additional assumption that every signal stabilizes eventually, i.e., $\neg\neg(s{=}0 \vee s{=}1)$, for all signals s.

Theorem 3.4 *Let P,Q be combinational programs and s be a signal. Then,*

$$\begin{aligned}
\mathtt{Spec}(P\,|\,Q,s) &\equiv (\mathtt{M}_1 \supset s{=}1) \wedge (\mathtt{C}_1 \supset s{\approx}0)\\
\mathtt{Spec}(a{=}1?(P),s) &\equiv (\mathtt{M}_2 \supset s{=}1) \wedge (\mathtt{C}_2 \supset s{\approx}0)\\
\mathtt{Spec}(a{=}0?(Q),s) &\equiv (\mathtt{M}_3 \supset s{=}1) \wedge (\mathtt{C}_3 \supset s{\approx}0)
\end{aligned}$$

are valid semantic equivalences, where

$$\begin{aligned}
&\mathtt{M}_1 := (\mathtt{Spec}(P,s) \supset s{=}1) \vee (\mathtt{Spec}(Q,s) \supset s{=}1)\\
&\mathtt{C}_1 := (\mathtt{Spec}(P,s) \supset s{\approx}0) \wedge (\mathtt{Spec}(Q,s) \supset s{\approx}0)\\
&\mathtt{M}_2 := a{=}1 \wedge (\mathtt{Spec}(P,s) \supset s{=}1)\\
&\mathtt{C}_2 := a{=}0 \vee ((\mathtt{Spec}(P,s) \vee \neg\mathtt{Spec}(P,s)) \supset s{\approx}0)\\
&\mathtt{M}_3 := a{=}0 \wedge (\mathtt{Spec}(Q,s) \supset s{=}1)\\
&\mathtt{C}_3 := a{=}1 \vee ((\mathtt{Spec}(Q,s) \vee \neg\mathtt{Spec}(Q,s)) \supset s{\approx}0).
\end{aligned}$$

Proof (Sketch) The observation underlying the inductive characterization is that $\mathtt{Must}(P,s)$ can be recovered from $\mathtt{Spec}(P,s)$ as $\mathtt{Spec}(P,s) \supset s{=}1$, and that $\mathtt{Cannot}(P,s)$ can be recovered as $(\mathtt{Spec}(P,s) \vee \neg\mathtt{Spec}(P,s)) \supset s{\approx}0$. Using case analysis, one verifies the equivalences

$$\begin{aligned}
&\mathtt{Must}(P,s) \wedge \neg\neg s{=}1 \equiv (\mathtt{Spec}(P,s) \supset s{=}1) \wedge \neg\neg s{=}1\\
&\mathtt{Must}(P,s) \wedge \neg\neg s{=}0 \equiv (\mathtt{Spec}(P,s) \supset s{=}1) \wedge \neg\neg s{=}0
\end{aligned}$$

from which $\mathtt{Must}(P,s) \equiv \mathtt{Spec}(P,s) \supset s{=}1$ follows. To show $\mathtt{Cannot}(P,s) \equiv (\mathtt{Spec}(P,s) \vee \neg\mathtt{Spec}(P,s)) \supset s{\approx}0$ we use a fourfold case analysis:

$$\begin{aligned}
&\mathtt{Cannot}(P,s) \wedge \neg\neg s{=}1 \wedge \neg\mathtt{Must}(P,s) \equiv\\
&\qquad ((\mathtt{Spec}(P,s) \vee \neg\mathtt{Spec}(P,s)) \supset s{\approx}0) \wedge \neg\neg s{=}1 \wedge \neg\mathtt{Must}(P,s)\\
&\mathtt{Cannot}(P,s) \wedge \neg\neg s{=}1 \wedge \neg\neg\mathtt{Must}(P,s) \equiv\\
&\qquad ((\mathtt{Spec}(P,s) \vee \neg\mathtt{Spec}(P,s)) \supset s{\approx}0) \wedge \neg\neg s{=}1 \wedge \neg\neg\mathtt{Must}(P,s)\\
&\mathtt{Cannot}(P,s) \wedge \neg\neg s{=}0 \wedge \neg\mathtt{Must}(P,s) \equiv\\
&\qquad ((\mathtt{Spec}(P,s) \vee \neg\mathtt{Spec}(P,s)) \supset s{\approx}0) \wedge \neg\neg s{=}0 \wedge \neg\mathtt{Must}(P,s)
\end{aligned}$$

$$\texttt{Cannot}(P,s) \wedge \neg\neg s{=}\texttt{0} \wedge \neg\neg\texttt{Must}(P,s) \equiv ((\texttt{Spec}(P,s) \vee \neg\texttt{Spec}(P,s)) \supset s{\approx}\texttt{0}) \wedge \neg\neg s{=}\texttt{0} \wedge \neg\neg\texttt{Must}(P,s)$$

The details of these proofs are not too difficult but tedious. For the last of these equivalences one also needs the fact that $\neg\neg\texttt{Must}(P,s) \equiv \neg\texttt{Cannot}(P,s)$. □

4 Discussion and Related Work

This section discusses our model–theoretic approach to Esterel semantics in the light of related work, with a focus on the semantic relation between Esterel and Statecharts. The intuitionistic semantics presented in this paper has previously been used by the authors to characterize Pnueli and Shalev's step semantics for the parallel, combinational fragment of Statecharts [7], which is *not* equipped with an explicit nondeterministic–choice construct. More precisely, it is shown in [6] that, if every Statecharts transition $a_1, \ldots, a_l, \overline{b}_1, \ldots, \overline{b}_m/c_1, \ldots, c_n$ is read as an implication $(a_1 \wedge \cdots \wedge a_l \wedge \neg b_1 \wedge \cdots \wedge \neg b_m) \supset (c_1 \wedge \cdots \wedge c_n)$ and parallel composition as conjunction, then the Gödel models of the resulting Statecharts formula provide a compositional and fully–abstract semantics for Pnueli and Shalev's macro steps. This semantic interpretation is generalized to the full Statecharts language in [5].

In the present paper we use the same model–theoretic principles to characterize the reactive semantics of combinational Esterel programs in terms of propositional logic formulas. From the point of view of our model theory, Esterel can now be seen as a refinement of Statecharts and Statecharts as a specialization of Esterel. To be precise, the parallel fragment of Statecharts coincides with the special Esterel theory for combinational programs in which, for all signals s, the axiom $s{\approx}\texttt{0}$ is assumed. Indeed, if we add the axiom $s{\approx}\texttt{0}$ to our logic, then the implications $\texttt{Cannot}(P,s) \supset s{\approx}\texttt{0}$ in $\texttt{Spec}(P)$ all collapse to *true* and $s{=}\texttt{0}$ becomes equivalent to $\neg s{=}\texttt{1}$. We may then simply identify $a{=}\texttt{1}$ with the name a and consider a as a propositional atom. For example, the program $a{=}\texttt{1}?(b{=}\texttt{0}?(!c))$ would thus translate, up to logical equivalence, into the formula $(a{=}\texttt{1} \wedge \neg b{=}\texttt{1}) \supset c{=}\texttt{1}$, which has the same semantics as the Statecharts transition $a, \overline{b}/c$. Similarly, one can show that, under the axioms $s{\approx}\texttt{0}$, parallel composition reduces to conjunction, i.e., $\texttt{Spec}(P_1 \,|\, P_2) \equiv \texttt{Spec}(P_1) \wedge \texttt{Spec}(P_2)$ so that Esterel "collapses" to Statecharts.

Another interesting way to look at the relationship between Esterel and Statecharts is to observe that the translation $\texttt{Spec}(P)$ essentially offers a faithful embedding of Esterel into Statecharts. Consider the program $P = a{=}\texttt{1}?(b{=}\texttt{0}?(!a) \,|\, !b)$. Its translation yields, modulo some trivial simplifications:

$$\texttt{Spec}(P) \equiv ((a{=}\texttt{1} \wedge b{=}\texttt{0}) \supset a{=}\texttt{1}) \;\wedge\; ((b{=}\texttt{1} \wedge \neg a{=}\texttt{1}) \supset a{=}\texttt{0}) \;\wedge\; (a{=}\texttt{1} \supset b{=}\texttt{1}) \;\wedge\; ((a{=}\texttt{0} \wedge \neg b{=}\texttt{1}) \supset b{=}\texttt{0})$$

which corresponds to the Statecharts program

$$a{=}\texttt{1}, b{=}\texttt{0}/a{=}\texttt{1} \;\mid\; b{=}\texttt{1}, \overline{a{=}\texttt{1}}/a{=}\texttt{0} \;\mid\; a{=}\texttt{1}/b{=}\texttt{1} \;\mid\; a{=}\texttt{0}, \overline{b{=}\texttt{1}}/b{=}\texttt{0}\,.$$

Our results now imply that the execution of this program in Statecharts, under arbitrary external inputs, yields exactly the same responses as if P was executed under the fixed–point semantics of Esterel. Note that, in this execution of $\texttt{Spec}(P)$ under the operational semantics of Statecharts, any additional assumption of the form $s{\approx}\texttt{0}$, which translates into the Statecharts transition $\overline{s{=}\texttt{1}}/s{=}\texttt{0}$, effectively allows us to speculate on the absence of s at any time in the construction of a Statecharts response. As pointed out above, it is the omission of these assumptions that makes Esterel a refinement of Statecharts. Moreover, our framework offers the possibility to mix Esterel and Statecharts consistently: we can introduce $s{\approx}\texttt{0}$ selectively for those signals that we wish to subject to a "speculative" Statecharts regime, while for all other signals we keep the strict rule of Esterel that forces the absence of a signal to be justified in a constructive, non–speculative way. In this context it is worth noting that the nondeterminism in Statecharts' parallel fragment is solely due to negations; without negative triggers, parallel Statecharts programs would be deterministic like Esterel programs. Note also that the smooth integration of Esterel and Statecharts outlined above seems to depend crucially on the use of weak signal absence, $s{\approx}\texttt{0}$, and hence the use of negation. It is because of negations that the intuitionistic viewpoint comes into play.

A quite different way of giving a logical account of Esterel is to encode or axiomatize Esterel's semantics directly in a suitable predicate logic. For instance, in [4] the semantics is formalized in the constructive higher–order logic of the *Calculus of Constructions*, and its implementation in Coq was used to verify the correctness of Berry's circuit translation [1] for a large fragment of Esterel. To achieve these results, the approach taken in [4] uses a deep embedding in the Calculus of Constructions. Our translation corresponds to a shallow embedding, and it is an embedding in propositional rather than in higher–order logic. Our approach is also distinct from Berry's logical semantics of "constructive value propagation" (Chap. 10.3 in [1]). This semantics for Esterel circuits is presented in terms of a predicate $I, R \vdash e \hookrightarrow b$ with the interpretation "*for input I and (register) state R, the propositional expression e (built from wires and constant values) constructively evaluates to the Boolean value b.*" The predicate $I, R \vdash e \hookrightarrow b$ is an inductive relation defined by a set of derivation rules similar to a logic calculus. The relationship of this calculus with our logic translation still needs to be investigated.

Let us finally mention a couple of other open problems that we hope to address in future work. Firstly, while we have shown compositionality of our model–theoretic semantics for Esterel, the full–abstractness question is still open. We conjecture that two Esterel programs P and Q have the same partial responses in all contexts if and only if $\texttt{Spec}(P)$ and $\texttt{Spec}(Q)$ have the same Gödel models. Secondly, note that we have verified the compositionality of $\texttt{Spec}(P)$ in the structure of P only relative to a fixed signal, i.e., we have shown how to construct the models of $\texttt{Spec}(P\,|\,Q, s)$ from those of $\texttt{Spec}(P, s)$ and $\texttt{Spec}(Q, s)$, for any fixed signal s. One might also try to obtain the models of $\texttt{Spec}(P \mid Q)$ from those of $\texttt{Spec}(P)$ and $\texttt{Spec}(Q)$. Thirdly, our model–theoretic semantics needs to be extended to cover other combinational operators of Esterel, in particular local signal declarations.

5 Conclusions and Future Work

This paper presented a novel, model–theoretic account of the semantics of a combinational fragment of Esterel, which complements the declarative, operational, and circuit–based approaches in [1]. Our technical setting is based on propositional intuitionistic logic, where formulas are interpreted over Gödel valuations. The obtained characterization of Esterel semantics via Gödel models suggests that the simple approach of explaining signal statuses in a three–valued Scott domain, which leads to a Kleene–style algebraic semantics as detailed in [1], may not be sufficiently expressive: it is too coarse since it only provides an algebra of signal values but not of truth values. In this light, our results promise to be a significant step forward in finding a native logic for Esterel, thereby explaining what kind of constructive logic Esterel is based on.

Regarding future work we plan to extend our results to a richer Esterel fragment. Moreover, our approach is expected to yield an axiomatization of Esterel reactions on the basis of a lattice–theoretic characterization of those Gödel valuations that arise with respect to Esterel semantics.

References

[1] G. Berry. The constructive semantics of pure Esterel. CMA, Ecole des Mines, INRIA, 1999. Draft version 3.0.

[2] G. Berry and G. Gonthier. The Esterel synchronous programming language: Design, semantics, implementation. *SCP*, 19(2):87–152, 1992.

[3] D. Harel. Statecharts: A visual formalism for complex systems. *SCP*, 8:231–274, 1987.

[4] D. Kaplan-Terasse. Vers la certification du compilateur v5 d'Esterel dans Coq. Techn. Rep. 4092, INRIA, 2000.

[5] G. Lüttgen and M. Mendler. Statecharts: From visual syntax to model-theoretic semantics. In *Workshop on Integrating Diagrammatic and Formal Specification Techniques*, pp. 615–621. Austrian Computer Society, 2001.

[6] G. Lüttgen and M. Mendler. The intuitionism behind Statecharts steps. *ACM Trans. on Computational Logic*, 3(1):1–41, 2002.

[7] A. Pnueli and M. Shalev. What is in a step: On the semantics of Statecharts. In *TACS '91*, vol. 526 of *LNCS*, pp. 244–264. Springer-Verlag, 1991.

[8] D. van Dalen. Intuitionistic logic. In *Handbook of Philosophical Logic*, vol. III, ch. 4, pp. 225–339. Reidel, 1986.