Towards a Model-Theory for Esterel

Gerald Lüttgen
Department of Computer Science
The University of Sheffield

Aims of this current research activity:

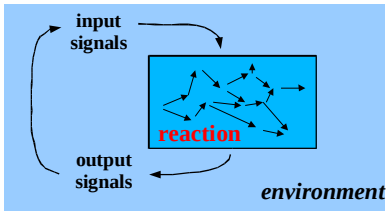
- To give a logic account of declarative aspects of Esterel, a programming language for reactive systems
- To find a suitable logic for reasoning about Esterel
- To clarify the relationship between Esterel and Statecharts

The Programming Language Esterel

- Developed by Gerard Berry since the early 1980's
- Synchronous language aimed at reactive-systems design
  - Textual with a mix of imperative and declarative aspects
- Solid mathematical semantics
  - Pure synchrony hypothesis, causality, reactivity, determinism
  - Three equivalent variants: behavioral, operational, circuit-based
- Commercially supported by the Esterel Studio design tool
  - Graphical front-end SyncCharts, with many more similarities to Harel, Pnueli & Shalev's Statecharts than just "optical" ones...

Reactive Systems

Cycle-based reaction:

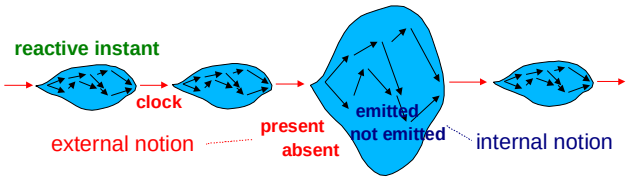


- Read statuses of input signals from the environment
- Determine reaction in the form of emitting further signals
- Output signal statuses to the environment

- External *(environment)* view:
  - Reactions are atomic
- Internal *(system)* view:
  - Reactions are non–atomic

Perfect synchrony hypothesis

Reactions in Esterel



Kernel syntax: *(for programming "instantaneous" reactions)*

P ::= nothing | emit s | present s then P else P | P || P

Logical Coherence Law:

"A signal s is present in an instant if and only if an `emit s' statement is executed in this instant."

The Esterel Puzzle: Emit or not Emit?

module P4:
output o
present o then emit o
end module

?

module P3:
output o
present o else emit o
end module

module P5:
output o1, o2
present o1 then emit o2 || present o2 then emit o1
end module

There should be a causal justification of why signals are or are not emitted, which can be traced back to the statuses of the input signals as given by the system environment.

Esterel's Behavioural Semantics

- **Berry's declarative approach:**
  - Determines the signals that must/cannot be emitted *(status 1 and 0, resp.)* via a monotonic least fixed-point construction
  - Accepts a program as "constructive" exactly if the statuses of *all* signals can be determined
- **New model-theoretic approach:**
  - Read must/cannot functions as predicates over s=1 and s=0
  - Interpret the resulting propositional formulas in the usual way, i.e., models are partial events

    $E \subseteq \{s=0, s=1 \mid s \text{ signal}\}$ $\quad \neg\exists\, s.\ s=0 \in E \sqcap s=1 \in E$
  - Characterise reactions as particular minimal models

Logical Must Analysis

For signal s and program P, consider the following:

s=1 *"s present"*	Must(P, s) *"P must emit s"*
s=0 *"s absent"*	Cannot(P, s) *"P cannot emit s"*

Remark: Assume input event is empty, since "P under I ⊓ P || I"

Must(nothing, s)	:=	*false*
Must(emit a, s)	:=	*true* if a=s, *false* otherwise
Must(present a then P, s)	:=	a=1 ⊓ Must(P, s)
Must(present a else P, s)	:=	a=0 ⊓ Must(P, s)
Must(P || Q, s)	:=	Must(P, s) ⊓ Must(Q, s)

Logical Cannot Analysis

For signal s and program P, consider the following:

s=1 *"s present"*	Must(P, s) *"P must emit s"*
s=0 *"s absent"*	Cannot(P, s) *"P cannot emit s"*

Remark: s = 0 ⊓ ☹s = 1 and Cannot(P, s) ⊓ ☹Must(P, s)

Cannot(nothing, s)	:=	*true*
Cannot(emit a, s)	:=	*false* if a=s, *true* otherwise
Cannot(present a then P, s)	:=	a=0 ⊓ Cannot(P, s)
Cannot(present a else P, s)	:=	a=1 ⊓ Cannot(P, s)
Cannot(P || Q, s)	:=	Cannot(P, s) ⊓ Cannot(Q, s)

Esterel Semantics

$$\text{Spec}(P) := \bigwedge_s (\text{Must}(P, s) \Rightarrow s=1) \Rightarrow (\text{Cannot}(P, s) \Rightarrow s \Box 0)$$

where $s \Box 0 := (☺ s=1) \Rightarrow s=0$ *("weak absence")*

Note: An "open systems" view is taken!

- **"s=1 ⇒ Must(P,s)" is not specified**
- **Weak absence "s□0" (as opposed to the stronger "s=0")**
  - **Accounts for the asymmetry between 1 and 0**
    - **Only the presence of a signal can be enforced by a program**
  - **Reflects the intuition that 0 is a default value only**
    - **The environment might overwrite the default 0 by 1**
    - **s=0 ⇒ s=1 is logically inconsistent, whereas s□0 ⇒ s=1 ⇒ s=1**

Why Classical Logic Fails

Example: **P0 := (present s then emit s) || (present s else emit s)**

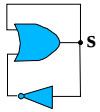
Spec(P0)

= (Must(P0, s) ⇒ s=1) ⇒ (Cannot(P0, s) ⇒ s□0)

= (((s=1□*true*) ⇒ (s=0□*true*)) ⇒ s=1)

⇒ (((s=0□*false*) ⇒ (s=1□*false*)) ⇒ s□0)

⇒ ☺ s=0



Here, "s=1" is logically coherent but not a valid reaction!

- **Classical logic equates ☺s=0 with s=1, ignoring that the status of s might not yet be determined**
- **This lack of expressiveness makes classical logic unsuitable and suggests the use of constructive/intuitionistic logics**

Intuitionistic Semantics

Formulas are interpreted over linear, intuitionistic Kripke structures, i.e., strictly increasing sequences of events $M = (E_1 \Rightarrow E_2 \Rightarrow \ldots \Rightarrow E_n)$.

$E_i \models$ true		always			
$E_i \models$ s■x	iff	⇒j ⇒ i	s■x ⇒ E_j		
$E_i \models$ ☺⇒	iff	⇒j ⇒ i	$E_j \not\models$ ⇒		
$E_i \models$ ⇒ ⇒ ⇒	iff	⇒j ⇒ i	$E_j \models$ ⇒	and	$E_j \models$ ⇒
$E_i \models$ ⇒ ⇒ ⇒	iff	⇒j ⇒ i	$E_j \models$ ⇒	or	$E_j \models$ ⇒
$E_i \models$ ⇒ ⇒ ⇒	iff	⇒j ⇒ i	$E_j \models$ ⇒	⇒	$E_j \models$ ⇒

M⊨ ⇒ iff ⇒i. E_i ⊨⇒ **Monotonicity of truth**

Interpret negation globally instead of locally!

Application to Esterel

- **Characterisation of Esterel's semantics via particular classical models, called reaction models, that are minimal in a certain constructive/intuitionistic sense**
- **Partial event E is a reaction model of P if**
  - E ⊨ Spec(P), i.e., E is a classical model of P; and
  - For all sequences $M = (E_1 \Rightarrow E_2 \Rightarrow \ldots \Rightarrow E)$ that end in E:
    M ⊨Spec(P) implies M = (E).

Theorem:

- **E is a reaction of P regarding Esterel's behavioural semantics if and only if E is a reaction model of P.**

Example (revisited)

- **Recall:** P0 := (present s then emit s) || (present s else emit s)
  - ❒ ☺s=0
- **Both {s=0} and {s=1} are not proper reactions:**
  - {s=0} is not a classical model of Spec(P0)
  - (∅ ⊂ {s=1}) ⊨ Spec(P0)

  Hence, {s=0} and {s=1} are not reaction models.
- **Remark:** The approach of intuitionistic reaction models is also applicable to Harel, Pnueli & Shalev's Statecharts (for its parallel fragment w/o explicit nondeterministic-choice operator [ICALP 2000]):
  - Read transitions $a_1,\ldots,a_l,☺b_1,\ldots,☺b_m$ / $c_1,\ldots,c_n$ as implications $(a_1 ❒ \ldots ❒ a_l ❒ ☺b_1 ❒ \ldots ❒ ☺b_m) ❒ (c_1 ❒ \ldots ❒ c_n)$
  - Read parallel composition as conjunction

Refining Esterel to Statecharts

- The parallel fragment of Statecharts coincides with a special Esterel theory of reactions, where s≈0 for all signals s
  - s≈0 enables speculation on the absence of signal s
- **Consequences:**
  - s=0 ■ ☺s=1, i.e., one may write s and ☺s for s=1 and s=0
  - Parallel composition || reduces to conjunction ❒
- **Example:** P := (present a then (present b else emit c)) || emit a
  - In logics: ((a=1 ❒ ☺b=1) ❒ c=1) ❒ a=1
  - In Statecharts: a,☺b/c || ·/a

Embedding of Esterel in Statecharts

- "Spec" offers an embedding of Esterel into Statecharts:
  - For example, Spec(P) ❒

    a=1 ❒ (☺b=1 ❒ b=0) ❒ (b=0 ❒ c=1) ❒ ((b=1 ❒ ☺c=1) ❒ c=0)
  - Corresponding Statecharts program:

    ·/a=1 || ☺b=1➢b=0 || b=0/c=1 || b=1,☺c=1/c=0
- Our framework allows one to "mix" Esterel and Statecharts:
  - Introduce s≈0 for those signals on whose absence one wishes to speculate *(à la Statecharts)*
  - The absence of the other signals must be justified constructively *(à la Esterel)*

Conclusions and Future Work

- **Conclusions:**
  - Presented a model-theory for Esterel reactions
  - Adopted an intuitionistic (not classical) interpretation that
    - Yields a compositional semantics for Esterel
    - Allows for a comparison of Esterel and Statecharts
- **Future work:** (Collaborator: Prof. Mendler, Univ. of Bamberg, D)
  - Enrich our kernel syntax by operators for signal hiding, sequential composition; expand our semantic theory
  - Is our micro-sequence semantics fully-abstract?
    (The answer is "yes" for Statecharts [ACM TOCL 2002])
  - An axiomatic comparison of Esterel and Statecharts
    (Axiomatisation for Statecharts recently developed [CONCUR 2002])
  - Implementation of the micro-sequence semantics

Other Active Research Areas

- **Symbolic model checking for asynchronous systems:**
  - **Using multi-valued decision diagrams and exploiting ideas of event locality and node saturation**
  - ***Collaborator:* Prof. Ciardo (College of William & Mary, USA)**

- **Basic research in semantics:**
  - **Studying concepts of preemption for priority and real-time [Handbook of Process Algebra, 2001; Annals of Software Engineering, 1998; TCS, 1998] [EPSRC project]**
  - **Mixing operational and declarative styles of specification by integrating process algebra & temporal logics [FSTTCS 2000] [NSF project]**
  - **Exploring faster-than relations for asynchronous systems [Information & Computation, accepted]**
  - ***Collaborators:* Prof. Cleaveland (SUNY at Stony Brook, USA), Prof. Vogler (Univ. of Augsburg, D)**