Constructive Semantics for Instantaneous Reactions

Joaquín Aguado and Michael Mendler

Faculty of Information Systems and Applied Computer Sciences, University of Bamberg, Germany
What is this talk about?

Problem: Synchronous Programming

- requires fixed points of non-monotonic functions
- incoherence of semantics, compositionality, ..., problems.

This talk:

- takes a new game-theoretic approach
- offering a uniform understanding of several constructive solutions, in particular reactions specified in
  - Esterel/SynchCharts (Berry)
  - Statecharts (Pnueli & Shalev)

that are long known but whose fundamental nature is only recently becoming clearer.
Hierarchical State Machines

\[ \text{REACT} := \\
  t_1 \supset b \land t_2 \supset b \land t_3 \supset c \land t_4 \supset b \land t_5 \supset a \land \\
  (s_{11} \land a \land \neg t_2) \supset t_1 \land \\
  (s_{11} \land \neg a \land \neg t_1) \supset t_2 \land \\
  (s_{31} \land \neg a \land b) \supset t_3 \land \\
  (s_2 \land c) \supset t_4 \land \\
  (s_{21} \land b \land \neg c \land \neg t_4) \supset t_5 \]

- flat conjunction of transitions
- obtained from visual syntax, structurally and incrementally
- negations code non-determinism, priorities and hierarchy
In which sense does REACT describe an atomic macro step i.e. combined non-deterministic response?

\[
\begin{align*}
REACT & := \\
&t_1 \supset b \land t_2 \supset b \land t_3 \supset c \land t_4 \supset b \land t_5 \supset a \land \\
&(s_{11} \land a \land \lnot t_2) \supset t_1 \land \\
&(s_{11} \land \lnot a \land \lnot t_1) \supset t_2 \land \\
&(s_{31} \land \lnot a \land b) \supset t_3 \land \\
&(s_2 \land c) \supset t_4 \land \\
&(s_{21} \land b \land \lnot c \land \lnot t_4) \supset t_5
\end{align*}
\]
In which sense does REACT describe an atomic macro step i.e. combined non-deterministic response?

Cyclic dependencies through negations! How are we to schedule these?
What is in a Step? - A Profusion of Options

- ban negations [Modecharts 1994]
- only accept conflict-free, deterministic programs [Argos, Normal Logic Programming]
- give up global consistency [Huizing&_al. 1988]
- add consistency as implicit trigger [Maggiolo-Schettini&_al. 1996, Lüttgen&_al. 1999]
- speculate on absence, if necessary backtrack [Pnueli&Shalev 1991]
- only schedule causally independent transitions, give up synchrony hypothesis [Statemate]
- negation as “positive“ absence [Berry 2000]
- negation with implicit delay [Saraswat‘s TCCP 1994]
The Problem, formalised ...
Synchronous Reactive Component

$C = (S, T, pos, neg, act)$

$S$ atomic signals \hspace{1cm} $pos, neg, act : T \rightarrow 2^S$

$T$ atomic transitions \hspace{1cm} positive, negative triggers, actions

The response of a reactive component with initial environment $E \subseteq S$ is determined by two functions:

$produced(T) = act(T) \cup E$

$enabled(S) = \{ t \mid pos(t) \subseteq S \land neg(t) \subseteq \overline{S} \}$

$PE_C(S) = produced(enabled(S))$ "response function"
Causal Response = Least Fixed Point?

Problem

The response function

$$PE_C(S) = \text{act}(\{t \mid \text{pos}(t) \subseteq S \land \text{neg}(t) \subseteq \overline{S}\}) \cup E$$

is not monotonic!

- no unique least fixed points!
- compositionality and full-abstraction problems!
- different computation methods!
  → different notions of steps, instants, reactions ...
Playing the Game, Constructively

\[ s \in \text{PE}(S) \iff \exists t \in \mathcal{T}. \ s \in \text{act}(t) \land \]

\[ \text{pos}(t) \subseteq S \land \text{neg}(t) \subseteq \overline{S} \]
Playing the Game, Constructively

We approximate $S / PE(S)$

- from inside via $P / PE^+(P,O)$ (= „player“)
- from outside via $O / PE^-(P,O)$ (= „opponent“)
Each \((P,O)\) with \(P \cap O = \emptyset\) is a ternary valuation called a front-line.

- \(s \in P\) s present (true)
- \(s \in O\) s absent (false)
- \(s \notin P \cup O\) s undefined
Playing the Game, Constructively

\[ s \in \text{PE}^+(P, O) \iff \exists t \in \mathcal{T}. s \in \text{act}(t) \land \text{pos}(t) \subseteq P \land \text{neg}(t) \subseteq O \]

\[ s \in \text{PE}^-(P, O) \iff \forall t \in \mathcal{T}. s \in \text{act}(t) \Rightarrow \text{pos}(t) \cap O \neq \emptyset \lor \text{neg}(t) \cap P \neq \emptyset \]
Response Functions as Games...
2-Player Symmetric Maze Games

A maze is a transition system \( M = (\mathbb{R}, \rightarrow) \) with
rooms (states) \( \mathbb{R} = \mathbb{R}_l \cup \mathbb{R}_\tau \) \( \mathbb{R}_l \) visible \( \mathbb{R}_\tau \) secret
corridors (transition relation) \( \rightarrow \subseteq \mathbb{R} \times \{ \nu, \tau \} \times \mathbb{R} \)
\( x \xrightarrow{\nu} y \) visible corridor \( x \xrightarrow{\tau} y \) secret corridor

Two players \( \mathcal{P} = \{ A, B \} \) move a token through \( M \) taking turns.

- **visible corridor** = visible move = turn changes
- **secret corridor** = invisible move = player continues
- **winning condition** (example): last player loses

We assume \( M \) is a fixed, finitely branching maze.
"A-Mazing" Programs

coding logical dependencies

\[
\text{REACT} = (c \land \neg a) \supset b \land (b \land c \land \neg d) \supset d \land \\
(b \land \neg a) \supset c \land \neg a \supset e \land \neg e \supset a
\]
Defensible Front-Lines

\[ \Pi^\alpha(P,O) \subseteq \Pi \] denotes the set of plays

- starting in a P-room with A or an O-room with B and
- player A follows strategy \( \alpha \).

A winning condition is a (time-invariant) subset \( \text{Win} \subseteq \Pi \) of plays.

A front-line \((P,O)\) is defensible according to a winning condition \( \text{Win} \) if player A has a strategy \( \alpha \) such that:

- \( \Pi^\alpha(P,O) \) is consistent (positions of A and B disjoint)
- \( \Pi^\alpha(P,O) \subseteq \text{Win} \)
Some winning conditions...

1 Trivial Winning
2 Coherent Winning
3 Lazy Winning
4 Eager Winning
1 Trivial Winning
1 Trivial Winning Condition

Win = \prod

Observation:
If \((P,O)\) defensible then it is defensible by \(\alpha = \emptyset\).
1 Trivial Winning Condition

**Win** = \( \Pi \)

\((P_1, O_1)\) is defensible

\( P_1 = \{ a, c, e, \ldots \} \)

\( O_1 = \{ b, d, \ldots \} \)

**Observation:**

If \((P, O)\) defensible then it is defensible by \( \alpha = \emptyset \).
1 Trivial Winning Condition

\[ \text{Win} = \Pi \]

\( (P_1, O_1) \) is defensible

\( P_1 = \{ a, c, e, ... \} \)

\( O_1 = \{ b, d, ... \} \)

Proposition:

The maximal defensible \((P, O)\) are binary and coincide with the classical models of REACT.

\[
\text{REACT} = (c \land \neg a) \supset b \land (b \land c \land \neg d) \supset d \land (b \land \neg a) \supset c \land \neg a \supset e \land \neg e \supset a
\]
2 Coherent Winning
2 Liveness

Definition

(P, O) is coherent if it is defensible by a live strategy $\alpha$, i.e. A always makes a move when he gets the turn.
(P₁, O₁) is not coherent!

P₁ = \{ a, c, e, ... \}

O₁ = \{ b, d, ... \}

Definition

(P, O) is coherent if it is defensible by a live strategy \( \alpha \), i.e. A always makes a move when he gets the turn.

A must make a move!
(P₂, O₂) is maximal coherent under α

P₂ = { b, c, e, ... }  
O₂ = { a, ...}

**Definition**

(P, O) is coherent if it is defensible by a live strategy α, i.e. A always makes a move when he gets the turn.
Proposition:
The maximal coherent \((P,O)\) coincide with the supported models \((LP)\)

\[
\text{REACT} = (c \land \neg a) \supset b \land (b \land c \land \neg d) \supset d \land (b \land \neg a) \supset c \land \neg a \supset e \land \neg e \supset a
\]
3 Lazy Winning
3 Reactiveness

Definition

(P, O) is lazy if it is defensible by a reactive strategy $\alpha$, i.e. A always eventually hands over to B in a visible room.
3 Reactiveness

(P₂, O₂) is not lazy!

P₂ = \{ b, c, e, ... \}
O₂ = \{ a, ... \}

In the cyclic play b → c → x → b

B only plays from secret rooms!

Definition

(P, O) is lazy if it is defensible by a reactive strategy α, i.e. A always eventually lets B play from a visible room.
3 Reactiveness

(P₃, O₃) is maximal lazy under α!

P₃ = { e, ... }
O₃ = { a, b, c, d, ... }

Definition
(P, O) is lazy if it is defensible by a reactive strategy α, i.e. A always eventually lets B play from a visible room.
3 Reactiveness

Proposition:
The maximal lazy (P,O) coincide with the stable models (Prolog) or Statecharts step responses (Pnueli&Shalev) of REACT.

Proposition:
The binary lazy (P,O) coincide with the minimal intuitionistic models of REACT in Gödel’s three-valued logic ($0 \leq \frac{1}{2} \leq 1$).

REACT = \( (c \land \neg a) \supset b \land (b \land c \land \neg d) \supset d \land (b \land \neg a) \supset c \land \neg a \supset e \land \neg e \supset a \)
3 Eager Winning
4 Termination

Definition

(P, O) is \textcolor{blue}{eager} if it is defensible by a \textcolor{blue}{terminating strategy} \( \alpha \), i.e. all plays are finite making B get stuck in a \textcolor{blue}{visible} room.
4 Termination

\((P_3, O_3)\) is not eager!

\(P_3 = \{ e, \ldots \}\)

\(O_3 = \{ a, b, c, d, \ldots \}\)

*every play from e has an infinite cycle*

**Definition**

\((P, O)\) is **eager** if it is defensible by a terminating strategy \(\alpha\), i.e. all plays are finite making B get stuck in a **visible** room.
... this maze has no eager front-line since the opponent B can always force infinite plays.

\[
\text{REACT} = (c \land \neg a) \supset b \land (b \land c \land \neg d) \supset d \land (b \land \neg a) \supset c \land \neg a \supset e \land \neg e \supset a
\]
4 Termination

... we need some “dungeons“:

\[
\text{REACT} = (c \land \neg a) \supset b \land (b \land c \land \neg d) \supset d \land \\
(b \land \neg a) \supset c \land \neg a \supset e \land \neg e \supset a \land \text{true} \supset a
\]
4 Termination

\((P_4, O_4)\) is maximal eager for \(\alpha\)!

\(P_4 = \{ a, \ldots \}\)

\(O_4 = \{ b, c, d, e, \ldots \}\)

\[
\begin{align*}
\text{REACT} & = (c \land \neg a) \sqsupset b \land (b \land c \land \neg d) \sqsupset d \land \\
& \quad (b \land \neg a) \sqsupset c \land \neg a \sqsupset e \land \neg e \sqsupset a \land \text{true} \sqsupset a
\end{align*}
\]
4 Termination

Proposition:
The maximal eager \((P,O)\) coincide with negation-as-failure (Prolog) or Esterel step responses \((Berry)\) of REACT.

Proposition:
The binary eager \((P,O)\) are the constructive Esterel step responses \((Berry)\) of REACT.

\[
\text{REACT} = (c \land \neg a) \supset b \land (b \land c \land \neg d) \supset d \land (b \land \neg a) \supset c \land \neg a \supset e \land \neg e \supset a \land \text{true} \supset a
\]
Wrapping up ...
Different degrees of constructiveness:

- **trivial winning** = classical Boolean valuations
- **coherent winning** = supported models
  - **lazy winning** = Pnueli & Shalev
  - **eager winning** = Esterel = SLDNF
    - = stable models = Gödel
M induces monotone response functions on front-lines

\[ \llbracket M \rrbracket_{efl} \leq \llbracket M \rrbracket_{lf} \leq \llbracket M \rrbracket_{cfl} \leq \llbracket M \rrbracket_{dfl} : FL \rightarrow FL \]

implementing the game-theoretic winning conditions:

- Post-fixed points (pfps) of \( \llbracket M \rrbracket_{dfl} = \text{classic front-lines} \)
- Pfps of \( \llbracket M \rrbracket_{cfl} = \text{coherent front-lines} \)
- Pfps of \( \llbracket M \rrbracket_{lf} = \text{lazy front-lines} \)
- Pfps of \( \llbracket M \rrbracket_{efl} = \text{eager front-lines} \)

**Wrapping up**
Summary

- **Synchronous programming** (also: Normal LP) involves **non-monotonic** step functions
- Non-monotonic step function can be refined into **2-player games**
- Different **winning conditions** can model different degrees of constructive truth values (defensible front-lines)
- Truth-values are characterised as **maximal post-fixed points of monotone functionals** on cpo of front-lines
Open Questions

• What is the precise algebraic structure generated by each of \([M]_{\text{efl}} \preceq [M]_{\text{lf}} \preceq [M]_{\text{clf}}\)?
  (compositionality, full abstraction, alternative logical characterisation, ...)

• How do we compute maximal pfps in each case, efficiently?

• Can we understand constructive degrees as modal operators?

• What other winning conditions lead to natural/known semantics?