What is in a Step: New Perspectives on a Classical Question*

Willem-Paul de Roever[1], Gerald Lüttgen[2], and Michael Mendler[2]

[1] Institute of Computer Science and Applied Mathematics,
Christian-Albrechts-University of Kiel, Germany,
wpr@informatik.uni-kiel.de

[2] Software Technologies and Informatics Theory Research Groups,
Otto-Friedrich-University of Bamberg, Germany,
gerald.luettgen@swt-bamberg.de, michael.mendler@uni-bamberg.de

Point of Departure: Pnueli & Shalev's 1991 paper "What's in a Step: On the semantics of Statecharts"

- Pnueli and Shalev show how, while observing global consistency and causality, the synchronous language Statecharts can be given coinciding operational and declarative (i.e., fixed point) step semantics
- Over the past decade, this semantics has been supplemented with order-theoretic, fully abstract and compositional denotational, axiomatic and game-theoretic semantics and used to emphasize the close connection with Esterel and logic programming
- This reveals the Pnueli-Shalev step semantics as a rather canonical interpretation of the synchrony hypothesis

Short intro to Statecharts

- A hierarchical, concurrent Mealy machine
- Basic states hierarchically refined by injecting other Statecharts
- Composite states of 2 possible sorts: and-states and or-states
- And-states permit parallel and or-states sequential decomposition
- An and-state is active if all its substates are active, an or-state if exactly one of its substates is active
- Set of active states during execution called a configuration

The synchrony hypothesis

- Statecharts belongs to the family of SYNCHRONOUS languages (s.a. Esterel, Signal, Lustre, Argos)
- Semantics based on a cycle-based reaction, in which events output by the system's env. are sampled first and pot. cause the firing of transitions that may produce new events
- Generated events output to the env. when the reaction ends
- SYNCHRONY HYPOTHESIS ensures that:
  this complex non-atomic step bundled into ONE ATOMIC STEP
- Justification: reactions computed quicker than time it takes for new events to arrive from the system's env

What exactly constitutes a step?

- Are generated events sensed only in the next step, or already in the current step, and thus trigger the firing of further transitions?
- First option: Harel's official non-compositional "semantics A" implemented in Statemate
- Second option: A step involves a causal chain of firing transitions:
- A transition fires if its positive triggers (offered by env or generated by a trans. fired previously in the same step) are present and its negative triggers are absent (i.e., not present)

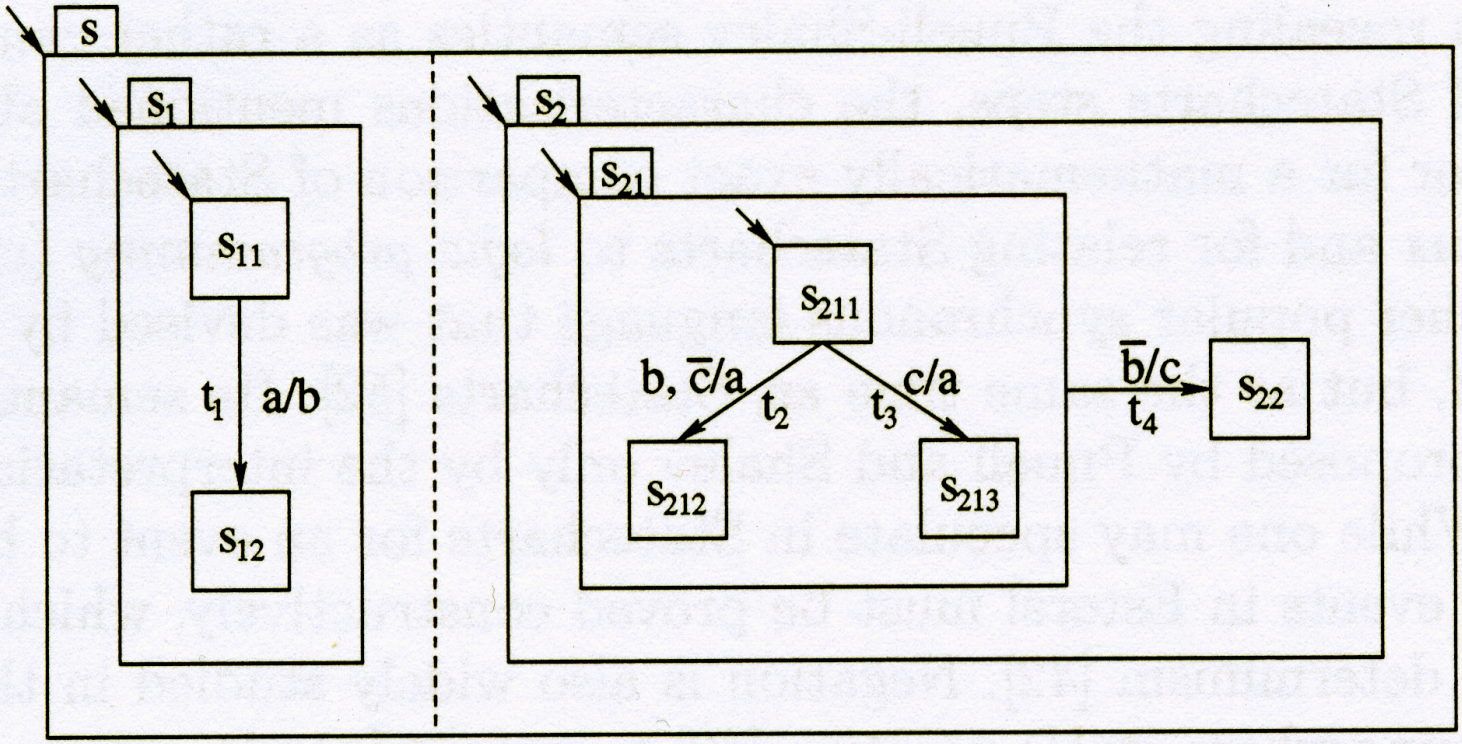

Fig. 1. Example Statechart.

What exactly constitutes a step (cont'd)?

- Thus, when it fires, a transition may, as part of its action, BROADCAST new events, which, by the principle of CAUSALITY, may trigger further transitions
- Only when this chain reaction of firing transitions comes to a halt is a step COMPLETE, and, acc. to the synchrony hypothesis, an atomic entity
- This semantics is NONCOMPOSITIONAL, since bundling a trans. into an atomic step implies forgetting the transition's causal justification
- Also, it is not GLOBALLY CONSISTENT, as it permits the same event to be both present and absent within the same step: an event that occurs negatively in the trigger of one firing transition MAY BE GENERATED BY A TRANS. THAT FIRES LATER IN THE SAME STEP

Pnueli & Shalev's contribution

- In Pnueli and Shalev's words, "a proven sign of healthy and robust understanding of the meaning of a programming or specification language is the possession of both an operational and declarative semantics, which are consistent with one another"
- They showed that adding global consistency is the key to achieving this ambitious goal for Statecharts
- The resulting operational semantics relies on an iterative FIXED-POINT CONSTRUCTION over a non-monotonic enabledness function for transitions
- This construction ensures causality but involves backtracking once a global inconsistency is introduced
- Their declarative semantics for Statecharts identifies the desired fixed point of the enabledness fu thru the notion of SEPARABILITY

Intro to Statecharts (cont'd)

- Statechart steps defined relative to a configation C and a set E of events given to the system by its environment
- Key to a step are transitions t each of which is labeled by two sets of events: a trigger trg(t) and an action act(t)
- Trigger $trg(t) = P, N^{co}$ split into positive events $P \subseteq \prod$ and negative events $N \subseteq \prod^{co}$.
- t is enabled and thus fires if the set $E \subseteq \prod$ is such that all events of P, but NONE of N, are in E, i.e., $P \subseteq E$ and $N \cap E = \emptyset$
- The effect of firing t is the generation of all events in the action act(t) of t, where a transition's action act(t) consists of positive events only

Transition t is *consistent* with set T of transitions, in signs $t \in \mathsf{consistent}(C, T)$, if t is not in the same "parallel component" as any $t' \in T \setminus \{t\}$. 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 (i) $t = t'$ or (ii) t and t' are in different substates of an enclosing and-state. Further, transition t is *triggered* by a 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:

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

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

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

Pnueli-Shalev Semantics

```
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.
```

Operational semantics

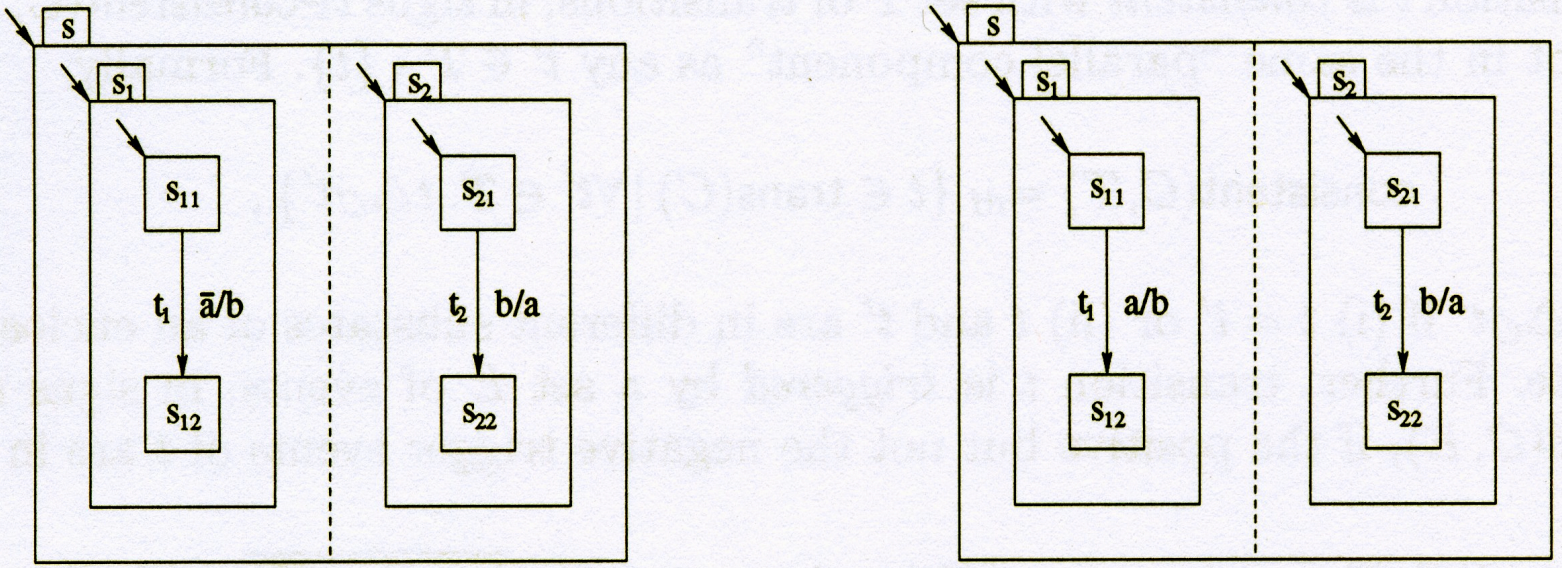

Fig. 2. Further example Statecharts.

Following Pnueli and Shalev's terminology, a set T of transitions is called *constructible* for a given configuration C and a set E of environment events, if it can be obtained as a result of successfully executing procedure *step-construction*. For each constructible set T, set $A =_{\text{df}} E \cup \mathsf{act}(T) \subseteq \Pi$ is called the *(step) response* of C for E.

Pnueli & Shalev's declarative semantics

- Given a config C and set of env events E, a set of trans. T is separable for C and E if $\exists\, T' \neq T$ s.t. $T' \subset T$ and enabled$(C,E,T') \cap (T \setminus T') = \varnothing$
- T is admissable for C and E if T is inseparable (not sep.) for C and E and $T = \text{enabled}(C,E,T)$, i.e., the declarative sem. is a fixed-point sem.
- Since enabled (C,E, .) may involve transitions with a negative trigger, it is in general non-monotonic, and a unique least fixed point may not exist.
- The notion of separability chooses distinguished fixed points that reflect causality
- A separable set of transitions points to a break in the causality chain when firing these transitions
- Thm 1 (Pnueli & Shalev). For all configs C and event sets E, a set T of trans. is admissable for C and E iff T is constructable for C and E

3.1 Configuration Syntax

This paper focuses on the semantics of single Statecharts steps, since the semantics across steps is clear and well understood. It will therefore be convenient to reduce the Statecharts notation to the bare essentials and identify a Statecharts configuration with its set of leaving transitions, to which we — by abuse of terminology — also refer as *configuration*. We formalise configurations using the following, simple syntax, where $I \subseteq \Pi \cup \overline{\Pi}$ and $A \subseteq \Pi$:

$$C \quad ::= \quad 0 \mid I/A \mid C \| C \; .$$

Intuitively, 0 stands for the configuration with the empty behaviour. Configuration I/A encodes a transition t with $\mathsf{trg}(t) = I$ and $\mathsf{act}(t) = A$. When triggered, transition t fires and generates the events in A. Transitions I/A with empty trigger, i.e., $I = \emptyset$, are simply written as A below. If we wish to emphasise that trigger I consists of the positive events $P \subseteq \Pi$ and the negative events $\overline{N} \subseteq \overline{\Pi}$, i.e., $I = P \cup \overline{N}$, then we denote transition I/A by $P, \overline{N}/A$. Finally, configuration $C_1 \| C_2$ describes the parallel composition of configurations C_1 and C_2. Observe that 0 coincides semantically with a transition with empty action; nevertheless, it seems natural to include 0. Using this syntax, we may encode the initial configuration C_1 of our example Statechart of Fig. 1 as

$$a/b \parallel b, \overline{c}, \overline{e_3}, \overline{e_4}/a, e_2 \parallel c, \overline{e_2}, \overline{e_4}/a, e_3 \parallel \overline{b}, \overline{e_2}, \overline{e_3}/c, e_4 \; .$$

- For simplicity, in this expo we focus on Statecharts w.r.t. the empty environment only
- This is no restriction, since considering a set E of events from env for a config C is equivalent to considering C//E relative to the empty set of events

New Perspective: Order-Theoretic Perspective

- Statecharts are viewed as process terms in process algebra, whose sem. is given by a compositional transl. into labelled trans. systs
- A transition represents a config. step decorated by an ACTION LABEL, specifying the synchr. causal interaction with the env.
- (Causality) labels are ordered (globally) consistent sets to encode causal info
- A causality label (or basic action) is a pair $(l,<)$ where
  - $l \subseteq \prod \cup \prod^{co}$ is a consistent set of pos. or neg. evnts, i.e., $l \cap l^{co} = \emptyset$
  - $A<B$ is an irreflexive and transitive causality ordering on subsets $A,B \subseteq l$, with $B=\emptyset$ or $B=\{b\}$ for $b \in \prod$, where
  - irreflexivity means that $A<\{b\}$ implies $b \notin A$ and,
  - transitivity that if $A<\{b\}$ and $b \in C < D$ then $((C \setminus \{b\}) \cup A) < D$

- Causality labels represent globally consistent and causally closed interactions that are composed from Statechart transitions
- Every transition $t \in trans(C)$ leaving config C induces a causality label , where
  - $l_t =_{def} trg(t) \cup act(t)$
  - $<_t =_{def} \{trg(t) <_t \{e'\} : e' \in act(t)\}$
  - $trg(t) \cap act(t) = \emptyset$ and for no $e \in \prod$ both $e, e^{co} \in trg(t) \cup act(t)$
- Then l_t is consistent, irreflexive and transitive

Ex. a/b // b,c^{co}/d

- Thus, $t_1 =_{def} a/b$ and $t_2 =_{def} b,c^{co}/d$ correspond to labels $l_1 = \{a,b\}$, $\{a\} <_1 \{b\}$, and $l_2 = \{b,c^{co},d\}$ with $\{b,c^{co}\} <_2 \{d\}$
- Their joint execution would be label $l_3 = \{a,b,c^{co},d\}$ with causalities $\{a\} <_3 \{b\}$, $\{b,c^{co}\} <_3 \{d\}$ and $\{a,c^{co}\} <_3 \{d\}$
- Here, the last pair arises from the combined reaction of t_1 triggering t_2; its presence is enforced by transitivity of $<_3$
- Note that this ex. composes causality labels in parallel
- In general, the parallel composition of causality labels $\sigma_1 = (l_1, <_1)$ and $\sigma_2 = (l_2, <_2)$ is the set $\sigma_1 \mathbf{X} \sigma_2$ of all maximal, irreflexive and transitive suborderings of the transitive closure $(<_1 \cup <_2)^+$

Next we define the operation of parallel composition between causality labels $\sigma_1 = (\ell_1, \prec_1)$ and $\sigma_2 = (\ell_2, \prec_2)$ to form the full causal and concurrent closure of all interactions coded in two orderings. Due to nondeterminism, the composition $\sigma_1 \times \sigma_2$ does not yield a single causality label but rather a set of them. They are obtained as the maximal irreflexive and transitive sub-orderings of the transitive closure $(\prec_1 \cup \prec_2)^+$. Here, the transitive closure of $\prec_1 \cup \prec_2$ is the smallest relation $\prec$ with $\prec_1 \cup \prec_2 \subseteq \prec$ such that, if $A \prec \{b\}$ and $b \in B \prec C$, then $(B \setminus \{b\}) \cup A \prec C$. Now, $(\ell, \prec) \in \sigma_1 \times \sigma_2$ if (i) $\ell = \ell_1 \cup \ell_2$, (ii) $(\ell, \prec)$ is a causality label, and (iii) $\prec$ is maximal in $(\prec_1 \cup \prec_2)^+$.

Theorem 2 (Correctness & Completeness). *If C is a configuration and $A \subseteq \Pi$, then A is a Pnueli-Shalev step response of C if and only if there exists a causality label σ with $C \mapsto \sigma$ such that $\emptyset$ enables σ and $A = \mathsf{act}(\sigma)$.*

Compositional, Fully Abstract and Denotational Semantics

- The Pnueli & Shalev semantics lacks compositionality because an interaction with the environment is only allowed at the beginning of a step but NOT during a step
- Compositionality can only be achieved by exhausting the communication potential of a step
- This is done by regarding interaction steps, basically, sequences of monotonically increasing fixed-points of the enabledness function, extending the communication potential until this potential is exhausted

Interaction steps

- Read a configuration C of a Statechart as a specification of a set of interaction steps between a Statechart and all its possible environments
- This set is nonempty since one may always construct an environment that disables those transitions in C that would cause global inconsistency and, thus, failure in the sense of Pnueli and Shalev
- An interaction step is a monotonically increasing sequence $M = (M_0, M_1, \ldots, M_n)$ of reactions $M_i \subseteq \prod$, where $M_{i-1} \subsetneq M_i$ for all i, and each reaction contains events representing both the environmental input and the Statecharts response.
- By the requirement for monotonicity, such a sequence extends the communication potential between the Statechart and its environment, until this potential is exhausted

Interaction steps (cont'd)

- An interaction step is best understood as a separation of a Pnueli-Shalev step response M_n in its n properly contained causally closed sub-fixed-points
- Each M_i extends M_{i-1} by new environmental stimuli plus the Statecharts response to these
- Here, responses are computed according to Pnueli and Shalev, except that events not contained in M_n are assumed to be absent in M_i
- Thus, global consistency is interpreted as a logical specification over the full interaction step M, and NOT only relative to a single reaction M_i

Interaction steps (cont'd)

- Thus, each interaction step separates a Pnueli-Shalev step response into causally-closed sets of events
- Each passage from M_{i-1} to M_i represents a non-causal "step" triggered by th environment
- This creates a separation between M_{i-1} and M_i in the spirit of P-S: as all events generated by the transitions enabled under M_{i-1} are contained in M_{i-1}, their intersection with $M_i \setminus M_{i-1}$ is empty

Interpreting configurations , logically

- Transitions $P, N^{co}/A$ of a config are interpreted on interaction steps $M = (M_0,\ldots,M_n)$ as follows: For each M_i, either
- (1) all events in A are also in M_i (the transition is enabled and thus fires), or
- (2) one or more events in A are not in M_i and $P \not\subseteq M_i$ (not all positive trigger events are present, disabling the transition), or
- (3) one or more events in A are not in M_i, and some event $e \in N$ is in M_j for some $i \leq j \leq n$ (global consistency is enforced over the whole interaction step M, disabling the transition)

Correspondence with intuitionistic propositional logic

- This interpretation correponds exactly to that of intuitionistic logic, reading negative events e^{co} as $\neg e$, transition slashes / as logical implication, and the composition of events in triggers and actions, and parallel composition // of configurations, as conjunction
- Interaction steps M are then linear Kripke structures
- This leads to the following def of logical satisfaction $\models$:
  An interaction step $M = (M_0,\ldots,M_n)$ satisfies configuration C, $M \models C$, if $M,i \models C$ for all $0 \le i \le n$, where
  - $M,i \models 0$ always (i.e.,configuration 0 is identified with true)
  - $M,i \models P,N^{co}/A$ if $P \subseteq M_i$ and $N \cap M_n = \emptyset$ implies $A \subseteq M_i$
  - $M,i \models C_1 // C_2$ if $M,i \models C_1$ and $M,i \models C_2$
- Now, $M \models C$ iff C is valid in the linear Kripke structure M

Main Result

- Note that for interaction steps of lenght 1, the notions of interaction model and classical model coincide, and we simply write M_1 for (M_1)
- Step responses of a config C in the sense of Pnueli and Shalev are now exactly those interaction models of lenght 1, called response models, that are not suffixes of interaction models $N=(N_0,\ldots,N_m,M)$ of C with lenght $m\geq 0$.
  For, if such a singleton interaction model was suffix of a longer interaction model, the reaction would be separable and hence not causal. Thus we have
- **Theorem 3 (Correctness and Completeness). If C is a configuration and $M \subseteq \Pi$, then M is a Pnueli-Shalev step response of C iff M is a response model of C**

– Firstly, consider the configuration $\overline{a}/b$ which exhibits the Pnueli-Shalev step response $\{b\}$ for the empty environment. Indeed, $\{b\}$ is a response model, i.e., a model and not a suffix of a longer interaction model. The only possibility would be the interaction step $(\emptyset, \{b\})$, but this is not an interaction model since $(\emptyset, \{b\}), 0 \not\models \overline{a}/b$: by definition, we have to consider $\emptyset \subseteq \emptyset$ and $\{a\} \cap \{b\} = \emptyset$ implies $\{b\} \subseteq \emptyset$, and this implication is false because $b \notin \emptyset$.

– Secondly, configuration $C_2 =_{\text{df}} \overline{a}/b \,\|\, b/a$ has no response model. Although $\{a, b\}$ is a classical model of C_2, it may be left-extended to the interaction model $(\emptyset, \{a, b\})$. Note in particular that $(\emptyset, \{a, b\}), 0 \models \overline{a}/b$: by definition, we have to consider $\emptyset \subseteq \emptyset$ and $\{a\} \cap \{a, b\} = \emptyset$ implies $\{b\} \subseteq \emptyset$, and this implication trivially holds. In other words, event a is absent at position 0 of the interaction step $(\emptyset, \{a, b\})$ since it is added later in the step, namely at position 1, and thus is *not* absent.

– Thirdly, consider configuration $C_3 =_{\mathrm{df}} a/b \,\|\, b/a$ with its Pnueli-Shalev step response $\emptyset$. It is easy to see that $\emptyset$ is trivially a response model. In contrast, the set $\{a, b\}$ — while being a classical model of C_3 — is not a response model since the suffix extension $(\emptyset, \{a, b\})$ is an interaction model of C_3.

– Fourthly, configuration $\overline{a}/b \,\|\, \overline{b}/a$ offers two response models, namely $\{a\}$ and $\{b\}$, which are exactly the configuration's Pnueli-Shalev step responses. As in the example regarding configuration C_2 above, neither response model can be left-extended to an interaction model of length greater than one.

Full abstraction. The interaction models of a configuration C encode all possible interactions of C with all its environments and nothing more. Firstly, any differences between the interaction models of C are differences in the interactions of C with its environments and thus can be observed. Secondly, any observable difference in the interaction of C with its environments should imply a difference in the interaction models, and this holds by the very construction of interaction models. Therefore, the above interaction step semantics provides the desired compositional and fully abstract semantics for Pnueli-Shalev steps:

Theorem 4 (Compositionality & Full Abstraction). *Let C_1, C_2 be configurations. Then, C_1 and C_2 have the same interaction models if and only if, for all configurations C_3, the parallel configurations $C_1 \| C_3$ and $C_2 \| C_3$ have the same Pnueli-Shalev step responses.*

3.4 Algebraic Perspective

We now turn to characterising the Pnueli-Shalev step semantics, or more precisely the largest congruence contained in equality on step responses, in terms of axioms. These are derived from general axioms of propositional intuitionistic formulas over linear Kripke models. Thus, the algebraic characterisation presented here is closely related to the above denotational characterisation.

Table 1. Axiom system for the Pnueli-Shalev step semantics

(A1)	$C_1 \parallel C_2 = C_2 \parallel C_1$		
(A2)	$(C_1 \parallel C_2) \parallel C_3 = C_1 \parallel (C_2 \parallel C_3)$		
(A3)	$C \parallel C = C$		
(A4)	$C \parallel 0 = C$		
(B1)	$P, I/P = 0$		
(B2)	$I/A \parallel I/B = I/(A \cup B)$		
(B3)	$I/A = I/A \parallel I, J/A$		
(B4)	$I/A \parallel A, J/B = I/A \parallel A, J/B \parallel I, J/B$		
(B5)	$P, \overline{N}/A = 0$		*if* $P \cap N \neq \emptyset$
(C1)	$P, \overline{N}/A = P, \overline{N}/A, B$		*if* $N \cap A \neq \emptyset$
(C2)	$P, \overline{N}/A = P, e, \overline{N}/A \parallel P, \overline{N}, \overline{e}/A$		*if* $N \cap A \neq \emptyset$
(C3)	$I, \overline{N}/B \parallel P, \overline{N}/A = \{I, \overline{N}, \overline{e}/B : e \in P\} \parallel P, \overline{N}/A,$		*if* $N \cap A \neq \emptyset$ *and* $P \neq \emptyset$

Theorem 5 (Correctness & Completeness). *$C_1 = C_2$ can be derived from the axioms of Table 1 via standard equational reasoning if and only if, for all interaction steps M, $M \models C_1$ iff $M \models C_2$.*

Theorem 6 (Correctness & Completeness). *Let C be a configuration and M_C be the maze associated with C. Then, $A \subseteq \Pi$ is a Pnueli-Shalev step response of C if and only if there exists a lazy front line $(R_A, S \setminus R_A)$ in M_C such that $A = R_A \cap \Pi$.*

The proof of this theorem can be found in [1]. Note how the game model accommodates both the failure and nondeterminism of step responses. Depending on M_C, it may happen that there is no strategy to avoid a (visible) room m being visited by both players infinitely often. This corresponds to Pnueli and Shalev's step-construction procedure returning a failure. Also, a room m may occur in two different lazy front lines, which yields nondeterministic behaviour.

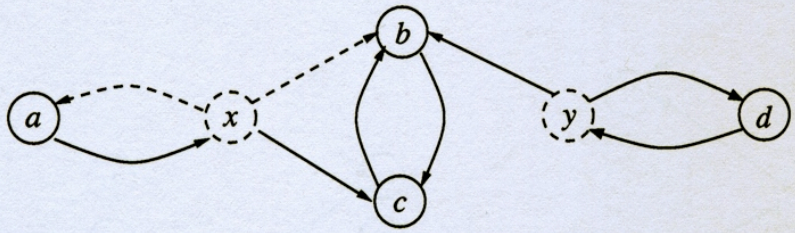

Fig. 4. The maze M_C for component $C = \overline{c}/b \,\|\, \overline{b}/c \,\|\, c, \overline{a}, \overline{b}/a \,\|\, b, d/d$ with maximal lazy front lines $(\{b, x, y\}, \{a, c, d\})$ and $(\{c, y\}, \{b, d\})$.

Game-Theoretic Perspective

the results. It has been observed in [1] that Pnueli and Shalev's interpretation of steps coincides exactly with the so-called *stable models* introduced by Gelfond and Lifschitz [21]. Consider configuration C as a propositional logic program. Given a set of events $E \subseteq \Pi$, let C_E be the program in which (i) all transitions with negative triggers in E are removed, i.e., we drop from C all $P, \overline{N}/A$ with $N \cap E \neq \emptyset$; and (ii) all remaining transitions are relieved from any negative events, i.e., every $P, \overline{N}/A$ with $N \cap E = \emptyset$ is simplified to P/A. The pruned program C_E has no negations, and thus it has a unique minimal classical model M. A classical model of C_E is a set $M \subseteq \Pi$ making all transitions/clauses of C_E true, i.e., for all P/A from C_E for which $P \subseteq M$ we have $A \subseteq M$. A set $M \subseteq \Pi$ is called a *stable model* of C if M is the minimal classical model of C_M. It has been shown in [21, 49] that stable models yield a more general semantics which consistently interprets a wider class of NLP programs than SLDNF.

Theorem 8 (Correctness & Completeness). *$M \subseteq \Pi$ is a stable model of configuration C if and only if M is a Pnueli-Shalev step response of C.*

Relation to Logic Programming

It is interesting to note that, while Pnueli and Shalev's notion of synchronous steps has not had much impact on synchronous programming tools, stable models have gained practical importance for NLP as the semantical underpinning of *answer set programming* [48]. From a wider perspective, therefore, it is fair to say that Pnueli-Shalev steps have indeed been implemented successfully in software engineering, albeit in a different domain. In addition, the theoretical results obtained around the Pnueli-Shalev semantics have ramifications in NLP. For instance, Thm. 4 of Sec. 3.3 implies that the standard intuitionistic semantics of logic provides a compositional and fully-abstract semantics for ground NLP programs under the stable interpretation.

Pnueli-Shalev semantics has been implemented in answer-set programming!

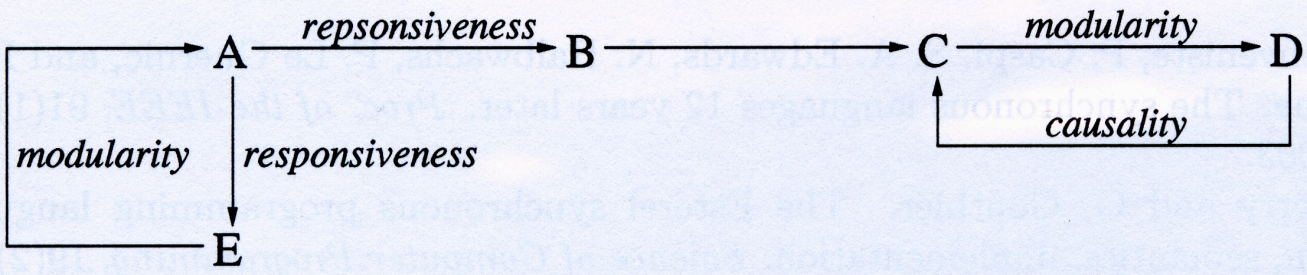

Fig. 5. Overview of the relationships among semantics A–E [32].

Amir's view summer 1986:

- Semantics A violates Synchr Hyp
- Sem. B introduces microsteps--too subtle
- Sem. C--Pnueli-Shalev--doesn't explain beh. in. terms of macrosteps
- Sem D--that of Argos--has problems with causality
- Sem. E, used in current impl. of Statemate,generates events in next step, but before reaction has died out, no new input from env. allowed.....???

NO MODULAR + RESPONSIVE + CAUSAL SEMANTICS CAN EXIST (GERTH&HUIZING,1988)