Correctness

Theorem:

For any state (C, I, V), $nextStep(C, I, V)$ is a macro step.

Proof.

The proof goes in two steps:

1. We prove enabledness, consistency, and maximality by applying some standard results from fixed point theory, in particular Tarski's-Kleene fixpoint theorem;
2. Then we consider priority and use some monotonicity argument.

□

RWTH AACHEN UNIVERSITY

Step execution

What happens in performing a step?

For a single statechart, executing a step results in performing the actions of all the edges in the step, and changing "control" to the target nodes of these edges.

Interference

Actions in statechart SC_j may influence the sets of events of other statecharts, e.g., SC_i with $i \neq j$ if action *send* $i.e$ is performed by SC_j in a step.

Thus:

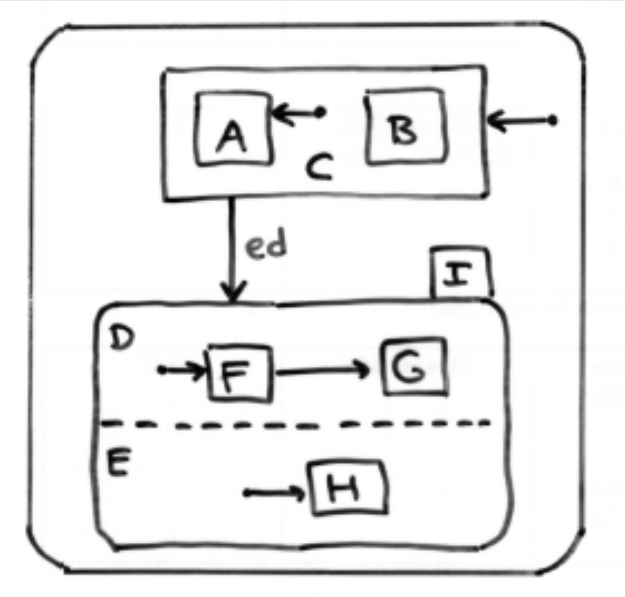
Execution of steps is considered on the system $(SC_1, \ldots, SC_n)$.

Default completion

Definition (Default completion)

The default completion C' of some set C of nodes is the canonical superset of C such that C' is a configuration. If C' contains an OR-node x and $children(x) \cap C = \varnothing$ implies $default(x) \in C'$.

Example:



1. Default completion of $C = \{\text{root}, I\}$ is $C' = C \cup \{D, E, F, H\}$
2. Default completion of $C = \{\text{root}, C\}$ is $C' = C \cup \{A\}$.

Step execution by a single statechart

- Let C_j be the current configuration of statechart SC_j
- Let $T_j \subseteq Edges_j$ be a step for SC_j
- The next state (C'_j, I'_j, V'_j) of statechart SC_j is given by:
  1. C'_j is the default completion of

$$\bigcup_{X \xrightarrow{e[g]/A} Y \in T_j} Y \cup \{x \in C_j \mid \forall X \to Y \in T_j . \neg(x \trianglelefteq scope(X \to Y))\}$$

  2. $I'_j = \bigcup_{k=1}^{n} \{e \mid \exists X \xrightarrow{e[g]/A} Y \in T_k . send\ j.e \in A\}$
  3. $$V'_j(v) = \begin{cases} V_j(v) & \text{if } \forall X \xrightarrow{e[g]/A} Y \in T_j . v := \ldots \notin A \\ val(\text{expr}) & \text{if } \exists X \xrightarrow{e[g]/A} Y \in T_j . v := \text{expr} \in A \end{cases}$$

RWTH AACHEN UNIVERSITY

Mealy machines [Mealy, 1953]

Definition (Mealy machine)

A Mealy machine $\mathcal{A} = (Q, q_0, \Sigma, \Gamma, \delta, \omega)$ with:

- Q is a finite set of states with initial state $q_0 \in Q$
- Σ is the input alphabet
- Γ is the output alphabet
- $\delta : Q \times \Sigma \to Q$ is the deterministic (input) transition function, and
- $\omega : Q \times \Sigma \to \Gamma$ is the output function

Intuition

A Mealy machine (or: finite-state transducer) is a finite-state automaton that produces output on a transition, based on current input and state.

Moore machines

In a Moore machine $\omega : Q \to \Gamma$, output is purely state-based.

From statecharts to a Mealy machine (1)

States

A state q is a tuple of the (local) states of SC_1 through SC_n.

Input and output events

Any input is a set of events, and any output is a set of events.

Next-state function δ

Defines the effect of executing a step.

Output function ω

Defines all events sent to some SC outside the system $(SC_1, \ldots, SC_n)$.

From statecharts to a Mealy machine (2)

States

A state q is a tuple of the (local) states of SC_1 through SC_k.

Formally:

- $Q = \prod_{k=1}^{n}(Conf_k \times 2^{E_k} \times Val_k)$ is the set of states
  - where $Conf_k$ is the set of configurations of SC_k,
  - E_k is the set of the events of SC_k,
  - and Val_k is the set of variable valuations of SC_k

- $q_0 = \prod_{k=1}^{n}(C_{0,k}, \varnothing, Val_{0,k})$ is the initial state
  - where $C_{0,k}$ is the default completion of the set {root}
  - the initial set of events is empty
  - $Val_{0,k}$ is the initial variable valuation of SC_k

RWTH AACHEN UNIVERSITY

From statecharts to a Mealy machine (3)

Input and output events

Any input is a set of events, and any output is a set of events.

Formally,

- Input alphabet: $\Sigma = 2^E - \{ \varnothing \}$
  - where $E = \bigcup_{k=1}^{n} E_k$ is the set of events in all statecharts

- Output alphabet: $\Gamma = 2^{E'}$
  - with $E' = \underbrace{\left\{ send\ j.e \in \bigcup_{k=1}^{n} SC_k \mid j \notin \{1, \ldots, n\} \right\}}_{\text{all outputs that cannot be consumed}}$

From statecharts to a Mealy machine (4)

Next-state function δ

Defines the effect of executing a step.

Formally,

- $(s'_1, \ldots, s'_n) \in \delta((s_1, \ldots, s_n), E)$ where
  - $s''_i = (C'_i, I''_i, V'_i)$ is the next state after executing $T_i = nextStep(C_i, I_i, V_i)$
  - and $s'_i = (C'_i, I''_i \cup (E \cap E_i), V'_i)$

RWTH AACHEN UNIVERSITY

From statecharts to a Mealy machine (5)

Output function ω

Defines all events sent to some SC outside the system $(SC_1, \ldots, SC_n)$.

Formally,

- $\omega((s_1, \ldots, s_n), E) =$
$$\left\{ send\ j.e \mid j \notin \{1, \ldots, n\} \wedge \exists i.\, \exists X \xrightarrow{e[g]/send\ j.e} Y \in nextStep(C_i, I_i, V_i) \right\}$$