EFSM Semantics*

CARA Team
Department of Computer and Information Science
University of Pennsylvania
POC: lee@cis.upenn.edu

August 15, 2002

1 Syntax

Definition 1.1 An EFSM (extended finite state machine) E is a tuple $\langle N, n_0, T, V\rangle$, where

- N is a set of nodes,
- n_0 is the initial node,
- T is a set of transitions, and
- $V = V_g \cup V_l$ is a set of typed variables.

All sets are assumed to be finite. □

Each node $n \in M$ represents a unique place and can have a label to identify it.

A transition $t \in T$ is of the form $\langle n, g, \alpha, n'\rangle$, where n is the source node, g is the guard condition over variables, α is the action that represents a set of assignments over variables, and n' is the target node.

The set V of variables are partitioned into two disjoint subsets, V_g and V_l. The set V_g is called global variables and the set V_l is called local variables. For a given EFSM, its global variables are visible outside the EFSM, and local variables are not visible outside the EFSM but can be used within the EFSM. Each variable x has a value domain, $dom(x)$, and global variables, V_g may have a set of possible initial values.

1.1 Operations on EFSMs

There are various operations on EFSMs to allow the construction of an EFSM from its component EFSMs.

Variable Hiding. The hiding operator makes a set of EFSM variables local.

Definition 1.2 Given an EFSM $E = \langle N, n_0, T, V_g \cup V_l\rangle$, the EFSM E
$V_h = \langle N, n_0', T, V_g' \cup V_l'\rangle$, where $V_g' = V_g - V_h$ and $V_l' = V_l \cup V_h$. □

*This research was supported in part by ...

Parallel Composition. The parallel composition of EFSMs allows the construction of a complex EFSM from simpler EFSMs.

Definition 1.3 Given two EFSMs, E_1 and E_2, the parallel composition $E_1||E_2$ is an EFSM $E = \langle N, n_0, T, V_g \cup V_l \rangle$, where

- $N = E_1.S \times E_2.S$,
- $n_0 = (E_1.n_0, E_2.n_0)$,
- T is given as follows (note that this is pure interleaving): for $\langle n_1, g_1, \alpha_1, n_1' \rangle \in E_1.T$ and $\langle n_2, g_2, \alpha_2, n_2' \rangle \in E_2.T$, T includes $\langle (n_1, n_2), g_1, \alpha_1, (n_1', n_2) \rangle$ and $\langle (n_1, n_2), g_2, \alpha_2, (n_1, n_2') \rangle$. (Note: We could also include $\langle (n_1, n_2), g_1 \wedge g_2, \alpha_1 \cup \alpha_2, (n_1', n_2') \rangle$.)
- $V_g = E_1.V_g \cup E_2.V_g$, and $V_l = (E_1.V_l \cup E_2.V_l)$. (Note that we assume that $E_1.V_l \cap E_2.V_1 =$.)

□

Restriction: Each variable in V_g is either read only or read/write. If a global variable is declared as read/write in one (primivite) EFSM, it cannot be declared as read/write in another EMFS.

Other operators to consider later:

- Parallel operators used in PET
- Sequential composition, abort/kill a subprocess

2 Semantics

Given a set of variables in V, we use Q_v to denote a valuation of the variables in V. The value of a variable x in the valuation Q is denoted as Q(x).

In modeling the execution of an EFSM, a state is represented by a pair $\langle n, Q \rangle$, where n is a node, $Q = Q_g \cup Q_l$ is the valuation of the variables $V = V_g \cup V_l$. The execution of an EFSM starts at a state $\langle n_0, Q_0 \rangle$, where n_0 is the initial node and Q_0 is the valuation that is consistent with the constraints, $Init(V_g)$, for the initial values of V_g. (Note: Should this constraints be part of the EFSM definition?)

A transition $\langle n, g, \alpha, n' \rangle$ can be taken from the current state $\langle n, Q \rangle$ only if the current valuation satifies the guard condition g. The effect of taking the transition $\langle n, g, \alpha, n' \rangle$ from a state $\langle n, Q \rangle$ is a state $\langle n', Q' \rangle$, where Q' is the new valuation resulted from executing the assignment statements specified in the set α.

Definition 2.1 An execution of an EFSM E $= \langle N, n_0, T, V \rangle$ is a finite or infinite sequence of the form

$$s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} s_3 \ldots$$

where $s_i = \langle n_i, Q_i \rangle$ satisfying

1. Initially:

2. Succession Constraint:

(Need to define compositionally, that is, allow enviromental update on V_g.)

□

3 Tools

For concrete syntax and tool support, we should use either CHARON or SPIN.