Controllers for Generalized Planning

Sergio Jiménez and Anders Jonsson
Interaction Technologies, Universitat Pompeu Fabra
138, 08018 Barcelona, Spain
s.jimenez,anders.jonsson}@upf.edu

Figure 1: Example of a binary tree with seven nodes.

Figure 2: Hierarchical FSC C[1] that traverses a binary tree.

In this paper we introduce a novel formalism for representing and computing compact and generalized planning solutions that we call hierarchical FSCs. Our formalism extends standard FSCs for planning in three ways. First, a hierarchical FSC can involve multiple individual FSCs. Second, each FSC can call other FSCs. Third, each FSC has a parameter list, and when an FSC is called, it is necessary to specify the arguments assigned to the parameters. As a special case, our formalism makes it possible to implement recursion by allowing an FSC to call itself with different arguments.

To illustrate this idea, Figure 2 shows an example hierarchical FSC C[1] that implements DFS traversal of a binary tree using recursion. Here, n is the lone parameter of the controller and represents the current node of the binary tree. Condition leaf(n) tests whether n is a leaf node, while a byphora - indicates that the transition fires no matter what. Action visit(n) visits node n, while copy_l(n,m) and copy_r(n,m) assign the left and right child of node n to m, respectively. Action call(m) is a recursive call to the FSC itself, assigning argument n to the only parameter of the controller and resuming execution from its initial node Q0.

Intuitively, by repeatedly assigning the right child of n to n itself (using the action copy_r(n,m)) and following the cycle of controller states Q0, Q1, Q2, Q0, ..., the FSC C[1] has the effect of visiting all nodes on the rightmost branch of