

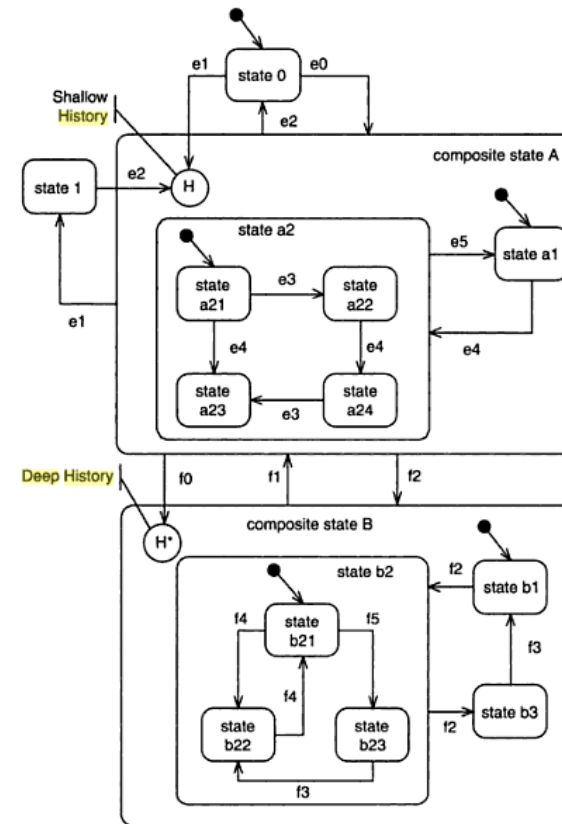
As we have seen, composite states are decomposed into nested states. The default nested state for a composite state is indicated with an initial **pseudostate**. When the composite state is exited and later reentered,

<sup>9</sup> In the general case, this is very difficult to automatically check for, because the guards may invoke operations that have nested operations. Therefore, modeling tools don't usually check for this kind of error.

the same nested state is again reentered. But what if the semantics you want are not to reenter the same nested state but to instead reenter the *last active* nested state?

The statechart answer is **history**. The **history pseudostate** is a letter “H” in a circle. Figure 3-7 shows the different aspects of the **history pseudostate**. To be activated, the transition must terminate on it, as with the transition from state 0, labeled e1, and the transition labeled e2 from state 1. Both of these transitions terminate on the **history pseudostate**. The transition labeled e0 from state 0 does not, so the normal default (to nested state a1) is activated whenever that transition is taken. However, if the active state is state a2 when transition e1 is taken, followed by transition labeled e2, then, the object reenters state a2. Of course, when the object is in state a2, it must also be in one of its nested states. Because of the kind of **history** used for composite state A, the default nested state for state a2 will always be reentered even if the last active nested state was state a24.

There is another kind of **history** as well—**deep history**. **Deep history** differs in that while shallow **history** only applies to the immediate level of nested and no deeper, **deep history** applies through all levels of nesting no matter how **deep**. **Deep history** is indicated with a circumscribed  $H^*$ , and is shown in composite state B. When the current state is state b23, and the object receives an event f1 (taking it to the default state of composite state A) followed by an event f0, the object reenters the state b23. As before, transitions that do not terminate on the **history pseudostate** don't activate the **history** semantics.



**Figure 3-7: *History***

selection (since only one branch can be taken), the multiple transitions exiting a fork indicate *concurrency*, at least in the logical sense, because *all* branches are taken, essentially at the same time. Figure 3-8 shows a fork in use. The default nested states are the InitTool and the InitArm