Pass 3: Section IV Conformance Testing – Status Messages and Reset:

This section describes a method that uses the functionality of status signals or reset signals to test the conformance of a test machine with the specification machine. To describe reliable testing techniques the author has taken several assumptions such as the Specification machine A is strongly connected and is reduced, implementation machine B does not change during the experiment and has the same input alphabet and has no more states than A. Also let $Z_i \ldots Z_n$ be a family of separating sets that are defined in earlier sections.

The author considers that if the machine has a special input status that gives an output status message. The status message provides information about the current state of the machine. In practice such signals do exist for example in hardware testing by observing the contents of the registers, which store the states of a sequential circuit. The status message is said to be reliable if it outputs the current state without affecting it. Then a checking sequence can be easily obtained by simply constructing a covering path of the transition diagram of the specification machine $A$, and applying the status message at each state visited. Since each state is checked with its status message, we verify whether $B$ is similar to $A$. Furthermore, transition is tested because its output is observed explicitly, and its start and end state are verified by their status messages; thus such a covering path provides a checking sequence. The author also notes that if the status message is not reliable then during the covering path we request a status message twice for each state the first it is encountered. For example, consider the specification machine $A$ in the Fig. Starting at state $s_1$. We have a covering path from input sequence $x = abab$. Let $s$ denote the status message. If it is reliable, then we obtain the checking sequence $sasbsbs$. If it is unreliable, then we have the sequence $ssasbssasbssasbs$.

Another signal that can be used for testing is the reset signal. The reset signal is defined as a signal that when given as an input at any state then the output state would be the initial state of the machine. For this method when we first as a breadth first search is tree is made of the transition diagram. Then to verify that B is similar to A we check the states according to breadth first search order and then check edges leading to nodes. For every state $s$, we have a part of the checking sequence that does the following for every member of $Z_i$; first, it resets the machine to $s$ by input $r$, then it applies the input sequence (say $pi_i$) corresponding to the path of the tree from the root $s_1$ to $s_i$ and then applies a separating sequence in $Z_i$. Now we do this for every state in the machine $B$, and when it passes for all the states then we know that there exists similar states in machine $B$ and $A$. Finally, we check non-tree transitions. For every transition, say from state $s_i$ to state $s_j$ on input $a$, we do the following for every member of $Z_i$ reset the machine, apply the input sequence $pi_i$ taking it to the start node $s_i$ of the transition along tree edges, apply the input $a$ of the transition, and then apply a separating sequence in $Z_i$. If the implementation machine $B$ passes this test for all members of $Z_i$ then we know that the transition on input $a$ of the state of $B$ that is similar to $s_i$ gives the correct output and goes to the state that is similar to state $s_j$. If $B$ passes the test for all the transitions, then we can conclude that it is isomorphic to $A$. For the machine in the Fig, a family of separating sets is $Z_1 = \{a, b\}$, $Z_2 = \{a\}$, and $Z_3 = \{a, b\}$. A spanning tree is shown in the Fig. with tree edges. Sequences ra and rb verify state s1. Similarly it can be proven for states s1 and s2. The author concludes that this method has a total of $pn$ transitions and by analysis the algorithm takes $O(pn^3)$ time.