1 Methodology

Each search is independent, and has its own control stack and Tarjan stack. A search is started at an arbitrary node startNode that has not yet been considered by any other search. Each search proceeds much as in the standard Tarjans Algorithm, as long as it does not encounter a node that is part of another current search. A difficulty occurs if suspending a search would create a cycle of searches, each locked on the next. Clearly we need to take some action to ensure progress. We transfer the relevant nodes of those searches into a single search, and continue, thereby removing the blocking-cycle.

1.1 Suspending and Resuming Searches

Each node n includes a field blocked: List[Search], storing the searches that have encountered this node and are blocked on it. When the node is completed, those searches can be resumed. Note that testing whether n is complete and updating blocked has to be done atomically. In addition, each suspended search has a field waitingFor, storing the node it is waiting on. We record which searches are blocked on which others in a map suspended from Search to Search, encapsulated in a Suspended object. The Suspended object has an operation suspend(s: Search, n: Node) to record that s is blocked on n.

1.2 Scheduling

The implementation uses a number of worker threads which execute searches. We use a Scheduler object to provide searches for workers, thereby implementing a form of task-based parallelism. The scheduler keeps track of searches that have been unblocked as a result of the blocking node becoming complete. A dormant worker can resume one of these. (Note that when a search is unblocked, the update to the Scheduler is done after the updates to the search itself, so that it is not resumed in an inconsistent state.) The algorithm can proceed in one of two different modes: rooted, where the search starts at a particular node, but the state space is not known in advance; and unrooted, where the state space is known in advance, and new searches can start at arbitrary nodes.[**?**]