1 Application Context

This case study models the well-known “Towers of Hanoi” for a varying number of disks. The scenario involves N disks of increasing size, which can be stacked on one of three pegs, with the restriction that a disk must never be on top of a smaller one. Initially, all disks are on the first peg; the goal is to move them all to the second peg, moving only one disk at a time and such that the above restriction is always satisfied.

These benchmarks serve as an example of a system with a very long error path (where the “error” condition describes the normal target configuration). Moving all N disks to the second peg requires $2^N - 1$ moves.

2 Model

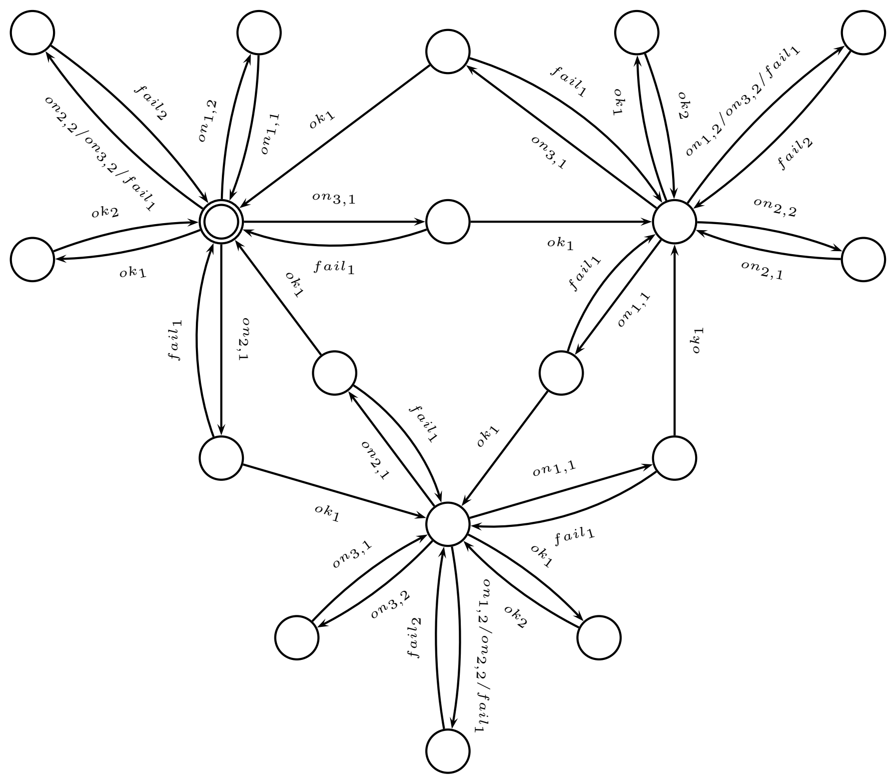

Fig. 1. Automaton modelling one disk in the “Towers of Hanoi” benchmark

The model contains a finite automaton for each disk (see fig. 1 for a typical example; the automata modeling the largest and the smallest disk are a little simpler). A legal move of disk m from peg i to peg j is modeled by a sequence of synchronization events, as follows (where k denotes the third peg, i.e. $k = 6 - i - j$):

- Automaton $disk_m$ sends signal $on_{k,m}$ in order to check if all smaller disks are on peg k;
- if automaton $disk_{m-1}$ is on peg i or j, it sends back a $fail_m$ signal, otherwise it propagates the request via $on_{k,m-1}$, etc;
- the smallest disk, upon receipt of $on_{k,1}$, sends ok_1 if it is on pek k (and $fail_1$ otherwise, like the other automata);
- when receiving a ok_l or $fail_l$ signal when not awaiting one, $disk_l$ forwards it (via ok_{l+1} or $fail_{l+1}$);
- $disk_m$ then executes (when receiving ok_m) or aborts (when receiving $fail_m$) the move.

3 Verification Results

Our heuristics are implemented in UPPAAL/DMC which is our extension of UPPAAL for directed model checking. In [1], we compared the performance of UPPAAL/DMC's greedy search and UPPAAL's randomised depth first search (rDF), which is UPPAAL's most efficient standard search method across many examples.

Our results clearly demonstrate the potential of our heuristics. The heuristic searches consistently find the error paths much faster. Due to the reduced search space size and memory requirements, they can solve all problems. At the same time, they find, by orders of magnitude, *much* shorter error paths in *all* cases.

References

1. Klaus Dräger, Bernd Finkbeiner, and Andreas Podelski. Directed model checking with distance-preserving abstractions. In Antti Valmari, editor, *Model Checking Software. Proceedings of the 13th International SPIN Workshop (SPIN 2006)*, volume 3925 of *Lecture Notes in Computer Science*, pages 19–34. Springer-Verlag, 2006.