Homework lecture 4
Self-stabilisation

Jaap-Henk Hoepman
jhh@cs.ru.nl

May 6, 2019

Question 1: What are the worst case number of steps before Dijkstra's algorithm reaches a legitimate state, when $K \geq N$ and assuming a central daemon and when started in an arbitrary bad state?

Answer: Start the system in the following state: $x[i] = N - i$ for all i, $0 < i \leq N$, and let $x[0] = 0 (= x[N])$. In round r, $0 \leq r$, let

- Node 0 take a step ($x[0] = x[N] = r$) to obtain the value $x[0] = r + 1 \bmod K$. This is one step.
- For j equals $N - r$ up to N, node j sets $x[j] = x[j-1]$. This sues $r+1$ steps.

At the end of round r, we have $x[N-r-1] = \ldots = x[N] = x[0] = r+1$. We can repeat this until $r = N-1$; after that the state is legitimate (it becomes so with the last step of $x[N]$ becoming $N \bmod K$ (we may have wrap around!). The total number of steps taken is

$$\sum_{r=0}^{N-1} r + 2 = 2N + \sum_{r=0}^{N-1} r = N + \sum_{r=0}^{N} r == N + 1/2N(N+1) = 1/2N^2 + 3/2N$$

So the number of steps is $O(n^2)$, and we now from the correctness proof that this is also maximum number of steps it takes the algorithm to stabilise.

Of course this method does not *prove* that this particular case is really the worst case scenario. On the other hand, we can extend the self-stabilisation proof explained during the lecture to show that the protocol always stabilises in at most $O(n^2)$ steps, so this is the worst case in terms of order of magnitude.

Question 2: Prove Dijkstra's algorithm correct if $K > N$ assuming a distributed daemon.

Answer: We first prove convergence.

Let $K > N$, and let node 0 be about to take the first step. In that case (just before the step), $x[0] = x[N]$. As there are $N + 1$ nodes, this means the maximum number of different values held by all these nodes is at most N. With $K > N$, there is a possible value, say a, that is not held by any node. Continue taking steps (verify for yourself that this is always possible) until $x[0]$ becomes a. This is the first time the value a occurs in the system. The next time node 0 takes a step happens when $x[N] = a$ too. This only happens if all intermediate nodes have copied the value a, i.e. $x[i] = a$ for all i. This is a legitimate state.

Closure is proven the same way as in the central daemon case.