

Homework lecture

Mutual Exclusion

Jaap-Henk Hoepman
jhh@cs.ru.nl

June 4, 2018

Question 1: In Lamport's bakery algorithm it is assumed that the lottery numbers $num[i]$ are unbounded. What goes wrong if we need to store these variables in a fixed size register that can only store bounded values?

Answer: In Lamport's bakery algorithm, an entering node reads all $num[j]$ and sets its own value to the maximum *plus 1*. Consider the following sequence of steps: node i enters, sees node j already entered and sets $num[i]$ to $num[j] + 1$. Node j enters the critical section, leaves, and wants to enter again. It now sees node i already entered and sets $num[j]$ to $num[i] + 1$ (i.e. its previous value plus 2). This can be repeated at infinitum until at some point the register will overflow. If we decide to remedy this by computing modulo the size of the register, then we break the property that nodes that enter later will not overtake nodes already contending for the critical section, thus violating the no-starvation requirement.

Question 2: What would go wrong in Lamport's bakery algorithm if we remove the use of the $choosing[i]$ variables?

Answer: In the proof of Lemma 1, it says

...then by assumption that now $num[k] \neq 0$, node k must have entered after node i saw $num[k] = 0$. Node i set its current value of $num[i]$ before that. Hence node k must have seen this value for $num[i]$ when computing a ticket. By the protocol k sets $num[k]$ to a larger value.

This is no longer true however. Consider the following schedule.

- Node k reads $num[i]$ (which is equal to 0).
- Node k starts writing $num[k]$ but it takes very long and does not finish writing yet. So $num[k] = 0$ still.
- Now node i starts and reads $num[k]$. It sees it is equal to zero (even though k is already entering!).

- Node i writes $num[i]$ and enters the critical section
- Node k wakes up and finishes the write to $num[k]$ (which does not take the current value of $num[i]$ into account). In the worst case also k enters the critical section.