Christian-Albrechts-Universität zu Kiel
Institut für Informatik und Praktische Mathematik

Prof. Dr. W.-P. de Roever
Martin Steffen, Immo Grabe

Verteilte Algorithmen

Wintersemester 2003/04 | 17. November 2003

Serie 4

Thema : Consensus mit Fehlern/Random Attack (Aufgaben mit Lösungshinweisen)

Ausgabetermin: 17. November 2003

Abgabe: 24. November 2003

Aufgabe 1 (6 Punkte) Beweisen Sie Lemmata 5.2 und 5.3 aus [?] (Aufgabe 5.5 + 5.6).

Lösung:

Proof of Lemma 5.2/Aufgabe 1(a): For a contradiction assume that there exists two different processes i and j and a round k such that[1]

$$\mathit{level}(i,k) - \mathit{level}(j,k) \geq 2$$

Let's set $m = \mathit{level}(i,k)$ and $n = \mathit{level}(j,k)$. The definition of the level function immediately gives that $k > 0$ and furthermore $(j,0) \leq (i,k)$, i.e., intuitively, there must have been some "communication" from j to i. So with i, j, and k given, we can write the assumption we wish to lead to a contradiction more shortly as

$$m > 1 + n \ . \tag{1}$$

Now consider for process i the knowledge it has about the state of j, i.e. the value of l_j (or more precisely $l_j^{i,k}$), as given in the definition:

$$l_j^{i,k} \triangleq \max\{\mathit{level}(j,k') \mid (j,k') \leq_\gamma (i,k)\} \ .$$

Intuitively, this value ($= n'$) denotes "the latest news" of process i about j. The following inequation is easy to show (cf. Lemma **??**):

$$n' \leq n \ . \tag{2}$$

[1] Wlog, we can pick i to have the larger level, of course.

According to the definition of the level of a process as the *minimum*, we further get

$$m \leq 1 + n' \ , \tag{3}$$

and thus we get with transitivity $m \leq 1+n$, contradicting our assumption (**??**) from above. □

Lemma 1 For all processes i and j (assumed to be different) and rounds k, such that the condition of case (3) of the level definition in [**?**] are satisfied, we are given:

$$l_j^{i,k} \leq \mathit{level}(j,k) \tag{4}$$

Proof: First note that $(j,k') \leq (i,k)$ implies $k' < k$ when $j \neq i$, and furthermore that $(i,k_1) \leq (i,k_2)$, when $k_1 \leq k_2$.

Again we can argue by contradiction. Assume, contrary to Equation (**??**), that

$$l_j^{i,k} = \max\{\mathit{level}(j,k') \mid (j,k') \leq_\gamma (i,k)\} > \mathit{level}(j,k) \ .$$

As stated, $k' < k$ for all the k's quantified over in the maximum, each single $\mathit{level}(j,k') \leq \mathit{level}(j,k)$. in the max, we know that *level*(The definition of $\mathit{level}(j,k)$, however, expands to $1 + \min\{l_i \mid i \neq j\}$ □

Lemma 2 (Monoticity) $k_1 \leq k_2$ implies $\mathit{level}(i,k_1) \leq \mathit{level}(i,k_2)$.

Proof: Basically by transitivity of $\leq_\gamma$.

In a bit more detail, let's first argue, that (in the situation of case 3 of the definition of *level*), $l_j^{k_1} \leq l_j^{k_2}$, whenever $k_1 \leq k_2$. The case when $k_1 = k_2$ is immediate. If otherwise $k_1 < k_2$, then clearly

$$\{\mathit{level}_\gamma(j,k') \mid (j,k') \leq (i,k_1)\} \subseteq \{\mathit{level}_\gamma(j,k') \mid (j,k') \leq (i,k_2)\}$$

by transitivity of the $\leq_\gamma$-relation. Hence

$$l_j^{i,k_1} \leq l_j^{i,k_2} \ ,$$

and this directly gives

$$\mathit{level}_\gamma(i,k_1) \leq \mathit{level}_\gamma(i,k_2) \ ,$$

as required.

The remaining cases 1) and 2) of the *level*-definition are simpler. □

Proof of Lemma 5.3/Aufgabe 1(b): In this lemma we are given perfect communication, i.e., $(i, j, k) \in \gamma$ for all i, j, and k.

In addition to the statement of the lemma, we prove by simultaneous induction on k, the following two assertions

$$\begin{aligned} l_j^{i,k} &= k-1 \qquad \text{for } k > 0 && (5) \\ \mathit{level}(i,k) &= k\ . && (6) \end{aligned}$$

For $k = 0$, the property holds directly by definition of *level*. For $k > 0$, first note that case 2) of the *level*-definition does not apply, hence we need to deal with case 3), only. By the assumption of perfect communication, the definition of $l_j^{i,k}$ simplifies the following way

$$\begin{aligned} l_j^{i,k} = \max\{\mathit{level}(j,k') \mid (j,k') \leq_\gamma (i,k)\} &= && \text{by perfect communication, and } i \neq j \\ \max\{\mathit{level}(j,k') \mid k' < k\} &= && \text{by monontonicity} \\ \mathit{level}(j,k-1) &= && \text{by induction} \\ k-1\ . & && \end{aligned}$$

Thus, the minimum operator for $\mathit{level}(i,k)$ is superfluous, and we directly get:

$$\mathit{level}(i,k) = k-1\ .$$

□

Aufgabe 2 (*RandomAttack*(4 Punkte)) Bearbeiten Sie Übung 5.7 (= Teile aus Theorem 5.4) aus [**?**].

Lösung: The *RandomAttack*-algorithm, a solution to the stochastic version of the coordinated attack problem, is given on page 90 and following. We first spell the requested properties. The first states, that the algorithm correctly implements (in a distributed way) the inductive level definition

Lemma 3 (Levels)

$$\mathit{level}_\gamma(i,k) = \mathit{level}[i]_i^k\ ,$$

for all good communication patterns γ, for all $0 \leq k \leq r$, and for all processes i after k-rounds.

Proof: We start with the code and consider more the value $\mathit{level}_i^k[j]$, when $j \neq i$, i.e., the opinion of process i concerning the level for an arbitrary different process j in round k. In the (interesting) case where this value is not equal $\perp$, it must have been set by process i after having received it via the corresponding tupel L which are exchange each round (if not lost). Anyway, since the communication is lossy, the information might not *directly* be received from j, nor might it *immediately* be propagated from j to i, but it must *originate* in j. More concretely, there is a path of the following form[2]

$$j \xrightarrow{k_0} h_1 \xrightarrow{k_1} h_2 \dots h_n \xrightarrow{k_n=k} i \ ,$$

with all process identifiers different (i.e., there's indeed a round of communication invoved), and furthermore

$$\mathit{level}^j_{k_0-1}[j] = \mathit{level}^i_k[j] \ .$$

In other words, the left-hand side of the equation point to the round where the current value of the right-hand side originated.[3]

The definition of $\leq_\gamma$ (using the base case for communication and transitivity) yields:

$$(j, k_0 - 1) \leq_\gamma (i, k) \ .$$

Let us use S as abbreviation for the set

$$\{(j', k')) \mid (j', k') \leq_\gamma (i, k)\} \ .$$

Since $\mathit{level}^i_k[j] = \mathit{level}^j_{k_0-1}[j]$, *induction* yields $\mathit{level}^j_{k_0-1}[j] = \mathit{level}_\gamma(j, k_0 - 1) \in S$, and therefore

$$\mathit{level}^i_k[j] \leq l_j = \max(S) \tag{7}$$

by definition of l_j as the maximum

On the other hand, since there is (at least one) a path from j' to i, the knowledge of process i about j satisfies

$$\mathit{level}^i_k[j] \geq l_j = \max(S) \tag{8}$$

Now finally, the code for the array in position i is defined the same way as in the inductive definition of the function, namely by adding one to the minimum to the "knowledge of the others." Hence

$$\mathit{level}_\gamma(i, k) = \mathit{level}^i_k[i] \ ,$$

as required. □

Lemma 4 (Initialization) After k rounds, if $\mathit{level}[i]^i \geq 1$, then key^i is defined and $\mathit{val}^i[j]$ is defined for all j. Moreover, these values are equal to the actual *key* chosen by process 1 and the actual initial values, respectively.

[2] We write $i_1 \xrightarrow{k} i_2$ more suggestively for $(i_1, i_2, k) \in \gamma$.

[3] The channels are lossy, but they do not alter messages.

Proof: In the code, each process i starts with level 0 for itself, i.e., initially

$$level^i[i] = 0 \ .$$

□

Aufgabe 3 (3 Punkte) Schließlich und endlich: Bearbeiten Sie Übung 5.8 aus immer noch [**?**].

Lösung:

1. Falls ein Prozess mit 0 startet, so ist 0 die einzig mögliche Entscheidung
2. Für jeden Gegner B, für den alles initialen Werte auf 1 sind gilt

   $$PR^B[\text{all process decide 1}] \geq l\epsilon \ ,$$

   wobei l das minimale Level aller Prozesse zur Zeit r in B ist.

Proof: For the first point, where (at least) one process starts with 0 as initial value. Now, a decision 1 being taken implies that $key \neq \perp$ for all processes. This means that each proceass knows each others initial values since they are passed around together with the *key*. Furthermore,, for a decision for 1, the values for all processes (as stored in the corresponding V-arrays) must be 1, which is a contradiction.

In the second part, let l be the mimumum of all levels in round r, which we assume to be fixed for the argument. Anyway, "stochastic" item which influences the decision, one l is fixed, is the key which is chosen randomly from the interval $[1, r]$, each value with a probability of $1/r$.

Now, in the algorithm, a uniform decision for 1 is taken, when

$$key \leq l = \min\{l_i\}$$

This happens in l *of* r cases, i.e.,

$$PR^B[\text{all processes decide 1}] \geq \frac{l}{r} = l\epsilon \ .$$

One could additionally make an argument for $l = 0$. □

The value of *key* is randomly chosen by the *process*. The adversary choses the initial value for the consensus. There are two arrays for each process, one for the levels and the knowledge of the process about the other processes' levels. The other array *val* is the knowledge about the processes (in-)decision. Those two vectors (together with the *key*) are send around each round (or lost). Not that the key is generated only *once* (by process 1 in the first round, and then propagated. The