

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

Serie 5

26. November 2003

Thema : Verteilte Einigung mit Prozeßfehlern (Aufgaben mit Lösungshinweisen)

Ausgabetermin: 26. November 2003

Abgabe: 1. Dezember 2003

Der Zettel behandelt *verteilte Einigung*, speziell die *randomisierte Variante*, und zwar in synchronen Netzen mit unzuverlässiger Kommunikation (Nachrichtenverlust).

Aufgabe 1 (Aufgabe 6.8 (8 Punkte)) Beweisen Sie Aufgabe 6.8 aus [?]: Geben Sie also Code für den *OptFloodSet*-Algorithmus und zeigen Sie die angegebenen Eigenschaften.

Solution: The algorithm is informally given on page 105. It's an optimization of the very basic *FloodSet* for agreement.

The solution given below uses an additional boolean *flag* to indicate when it has received any new value, i.e., some value different from its own. If this happens *for* the first time, it chooses one of the “dissenting” values and propagates it. Ever after, the process remains silent until it takes its decision.

```

Process(i) {
  states_i

  rounds = 0 : Nat;
  decision : V ∪ {unknown}, initially unknown;
  W : 2V, initially containing only the process's initial value;
  send = :V⊥, initially the init value // aux. send register
  update = false : Bool;           // boolean flag, set once when
                                   // a discrepancy has been detected

  msg_i:
    broadcast the value of send;

  trans_i:

    rounds := rounds + 1;
    let Xj be the message from j, for each j from which a message arrives

    if update = false then
      Wnew := W \ ∪j Xj; // potential new values
      if Wnew ≠ ∅
      then

```

```

        send := some  $u \in W_{new}$ ;
        update := true;
    else
        send :=  $\perp$ ;
    if rounds =  $f + 1$  then
        if update = true then decision := 1, where  $W = \{w\}$  // update is never reset
        else decision :=  $v_0$ ;

```

Aufgabe 2 (EIGStop(3 Punkte)) Beweisen Sie Aufgabe 6.13 aus [?]. Beschreiben Sie ein Szenario bei denen dieser Algorithmus fehlerhaft verhält, wenn er nur f statt $f + 1$ Runden läuft, wenn f die Anzahl der Ausfälle ist.