Universität Stuttgart
Fakultät 5, Institut FMI
Einf.i.d.Informatik II

Sommersemester 2003
Lewandowski, Claus
16. Mai 2003

Aufgabenblatt 3 **Abgabe: 23. Mai, 20.00 Uhr**

1. **Terminierung und Korrektheit III** (mittel) (schriftl., 2+5+1+1+1 Punkte)
Betrachten Sie das folgende Sortierverfahren:

```
type Feldtyp is array (1..Max) of Elementtyp;

procedure SORT2 (F: in out Feldtyp) is
   I, J : Integer := 1;
   T: Elementtyp;
begin
   while I < Max loop
      J := I+1;
      while J <= Max loop
         if F(I) > F(J) then
            T := F(I); F(I) := F(J); F(J) := T;
         end if;
         J := J+1;
      end loop;
      I := I+1;
   end loop;
end SORT2;
```

a) Führen Sie den Algorithmus an dem Beispiel `F := (4,7,2,5,8,3,6);` Schritt für Schritt durch (geben Sie die Vertauschungen und das Feld nach jeder Vertauschung an) und beschreiben Sie kurz(!) die Idee dieses Sortieralgorithmus.

b) Zeigen Sie die Korrektheit mit Hilfe der Hoareschen Regeln.

c) Ersetzen Sie die `if`-Abfrage durch `if F(I) >= F(J) then ....` Welchen Einfluss hat dies auf die Korrektheit des Algorithmus? Begründen Sie Ihre Antwort mit den Hoareschen Regeln.

d) Zeigen Sie, dass der Algorithmus immer terminiert.

e) Schätzen Sie möglichst genau ab, wieviel Vergleiche (`if F(I) > F(J) then ...`) durchgeführt werden.

Begründen Sie Ihre Antworten!

2. **Konstruktion von S/T-Netzen** (leicht-mittel) (votieren, 5+1+1 Punkte)
Das Problem der „dining philosophers“ (nach E. Dijkstra): Fünf Philosophen sitzen an einem runden Tisch, auf dem in der Mitte eine Schüssel Spaghetti steht. Zwischen je zwei benachbarten Philosophen liegt genau eine Gabel (also insgesamt 5 Gabeln). Jeder Philosoph durchläuft die Zustände denkend → hungrig → essend → denkend. Zu Beginn denken alle Philosophen; wird einer von ihnen hungrig, muss er die beiden Gabeln rechts und links von sich nehmen (falls sie nicht bereits von einem Nachbar-Philosophen benutzt werden) um essen zu können und nach dem Essen legt er wieder beide Gabeln auf den Tisch, um so evtl. seinen Nachbarn das Essen zu ermöglichen.

   a) Modellieren Sie dieses Philosophenproblem als S/T-Netz.
   b) Gibt es Aktionsfolgen, bei denen einer der Philosophen nie die Möglichkeit bekommt in den essenden Zustand überzugehen?
   c) Gibt es Aktionsfolgen, nach denen keine Aktion mehr schalten kann?

   Begründen Sie Ihre Antworten für Teil b) und c), geben Sie ggf. Beispiele an.

3. **Analyse von S/T-Netzen** (mittel) (schriftlich, 3 Punkte)
Ist die Kapazität für alle Stellen eines S/T-Netzes gleich 1, so vereinfachen sich die Fragen nach Erreichbarkeit einer Markierung und ob das Netz verklemmungsfrei ist (vgl. Folie 22 in Abschnitt 2.2). Skizzieren Sie umgangssprachlich zwei Algorithmen, die diese Probleme für den Fall lösen, dass jede Stelle nur 1 Marke haben kann, und begründen Sie deren Funktionalität.

Alles weitere unter

http://www.informatik.uni-stuttgart.de/fmi/fk/lehre/ss03/info_II/default.htm

Fragen zur Vorlesung und den Übungen, sowie Anregungen und Kritik können auf dem Schwarzen Brett

http://fachschaft.informatik.uni-stuttgart.de/forum/

diskutiert werden.