2. Übung zur *Nichtsequentiellen Programmierung* (Ch. Maurer, WS 97/98)

1. Ersetzen Sie die Implementierung von `FUNCTION TAS` im Programm `INC11` durch die viel einfachere `BEGIN Wait (T) TAS:= a; a:= TRUE Signal (T) END;` und testen Sie `INC11` damit.

a) Was geht schief?

b) Warum?

c) Und warum ist die Implementierung aus der Vorlesung korrekt?

2. Konstruieren Sie eine Schloßvariable unter Rückgriff auf die in der Vorlesung angegebene Anweisung `DEC`.
Begründen Sie die Korrektheit Ihrer Protokolle!
Was ist die Repräsentation von `locked`?

3. Untersuchen Sie die Modifikation von Programm `INC4` zur Sperrsynchronisation, wenn die Implementierung von `lock1` (entsprechend `lock2` per `eins` $\leftrightarrow$ `zwei`) wie folgt geändert wird:

```
REPEAT einsEintrittswillig:= TRUE;
       IF zweiEintrittswillig
         THEN einsEintrittswillig:= FALSE
UNTIL einsEintrittswillig
```

4. Beweisen Sie, daß der Algorithmus von Dekker für zwei Prozesse

a) deren gegenseitigen Ausschluß garantiert,

b) behinderungsfrei ist,

c) fair ist

d) und nicht zu Verklemmungen führen kann.