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

1. Begründen Sie, daß der Algorithmus von Habermann sämtliche Bedingungen an eine Sperrsynchronisation erfüllt.

2. In einer älteren Version des Bakery-Algorithmus aus dem Jahre 1974 lautet der Kern des Eintrittsprotokolls

```
zieht[p]:= TRUE;
Nummer[p]:= Maximum + 1;
zieht[p]:= FALSE;
FOR a:= 1 TO Prozesszahl DO BEGIN
  WHILE zieht[a] DO Null;
  WHILE (Nummer [a] <> 0) AND kleiner (a, p) DO Null
END;
```

mit `VAR zieht: ARRAY [1..Prozesszahl] OF BOOLEAN;`, wobei `zieht` mit `0` initialisiert ist.

a) Begründen Sie die Korrektheit des Algorithmus.

b) Erklären Sie den Unterschied zu der späteren Version.

3. Lösen Sie das Problem kritischer Abschnitte mit einem Koordinator-Prozesses wie folgt: wenn ein Prozeß den kritischen Abschnitt betreten will, teilt er das dem Koordinator-Prozeß mit und wartet auf dessen Erlaubnis. (Hinweis: [A], S. 120–124).

a) Entwickeln Sie die notwendigen Protokolle.

b) Begründen Sie die Korrektheit Ihrer Algorithmen.

4. Lösen Sie das Produzenten-/Konsumenten-Problem mit aktivem Warten:
Gegeben sei ein beschränkter Puffer, der nebenläufig von einem Produzenten aufgefüllt und von einem Konsumenten entleert wird. Bei vollem Puffer muß der Produzent warten; wenn der Puffer leer ist, muß entsprechend der Konsument verzögert werden. Der Puffer wird als Warteschlange realisiert: der Konsument erhält immer den ältesten Pufferinhalt zuerst.
Verwenden Sie für die Implementierung die klassische Repräsentation in einem "Ringpuffer": einem Feld mit Zeigern auf den ersten freien und den ersten vollen Pufferplatz.