Technische Universität Braunschweig

Institut für
Betriebssysteme und Rechnerverbund

Prof. Dr. H. Langendörfer · Prof. Dr. M. Zitterbart

Übungen zu Algorithmen und Datenstrukturen I+II

Wiederholungsblatt

1. Aufgabe: Programmverifikation

(a) Gegeben ist der folgende kleine Programmausschnitt mit den boolean-Variablen `p`, `q` und `s`. Verifizieren Sie den Ausschnitt unter Verwendung des WP-Kalküls.

$\{ \neg p \wedge q \}$

```
if (p)
  s = false;
else
  s = !q;
s = !s;
```

$\{ \neg p \wedge s \}$

Hinweis für mathematisches Rechnen mit booleans:

- $(a \Rightarrow b) \equiv (\neg a \vee b)$
- $a \wedge (b \vee c) \equiv (a \wedge b) \vee (a \wedge c)$
- $a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c)$

(b) Gegeben ist der folgende Programmausschnitt mit den `int`-Variablen `a`, `b`, `d`, `i` und `s` sowie der Konstanten `n` (`n` ≥ 0).

$\{ \mathit{TRUE} \}$

```
i = 0;
b = a;
s = a;
while (i < n)
{
   i += 1;
   b += d;
   s += b;
}
```

$\{ (b = a + n * d \wedge 2 * s = (n+1) * (a+b) \}$

Zeigen Sie die totale Korrektheit des Programmausschnitts mit Hilfe des WP-Kalküls. Geben Sie zuerst die Schleifeninvariante Q an.

(c) Gegeben ist eine Menge von n AuDI-Assis und -Hiwis ($0 < n < 6$). Erraten Sie die Mächtigkeit m dieser Menge sowie die Lieblingsbiersorten der einzelnen Elemente und besorgen Sie bis zur Klausur ein 30-Liter-Faß von jeder Sorte.

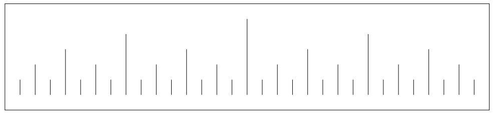
2. Aufgabe: Petri-Netze: Erzeuger-Verbraucher-System

Das Erzeuger-Verbraucher-System besteht aus drei Teilen: Ein Produzent, der irgend eine Art Gegenstände herstellt, die in einem Puffer/Warteschlange gesammelt werden. Aus dem Puffer entnimmt der Konsument jeweils einen Gegenstände und verbraucht diesen. Es gelten hierbei folgende Bedingungen: Es wird nur produziert, wenn Platz im Puffer ist. Der Konsument wartet solange, bis er einen Gegenstand aus dem Puffer entnehmen kann.

(a) Geben Sie ein Petri-Netz für das Erzeuger-Verbraucher-System mit Pufferplatz für einen Gegenstand an.

(b) Wie sieht das Petri-Netz für ein System mit zwei Pufferplätzen aus?

3. Aufgabe: Aufwandsberechnung



Betrachten Sie die Aufgabe, Teilstriche für jeden Zoll auf einem Lineal zu zeichnen. Es existieren Teilstriche im Punkt 1/2", etwas kürzere Teilstriche in Intervallen von 1/4", usw. wie in der Abbildung dargestellt. Dieses Problem kann mit einem einfachen 'divide-and-conquer' Verfahren gelöst werden:

```
void rule ( int l, int r, int h )
{
  if ( h > 0 ) {
    int m = ( l + r ) / 2;
    mark ( m, h );     // Teilstrich an Position m mit Hoehe h
    rule ( l, m, h-1 );
    rule ( m, r, h-1 );
  }
}
```

(a) Wie oft wird die Funktion `rule` in Abhängigkeit des Parameters `h` aufgerufen ? Der initiale Aufruf der Funktion ist: rule(0, 2^h, h).

(b) Was ist die Komplexitätsklasse der Funktion `rule` ?

(c) Geben Sie eine äquivalente nichtrekursive Funktion für `rule` an.

Die Vordiplomklausur „Praktische Informatik" findet am Montag, 28. September 1998, von $09^{00} - 13^{00}$ im AudiMax und PK 15.1 (ehemals P4, befindet sich unter dem Audi-Max) statt.
*Als Hilfsmittel sind ausschließlich Unterlagen der Vorlesung zugelassen, d.h. eigene Mitschriften sowie Kopien der Folien. **Nicht** zugelassen sind weitere Literatur (Bücher etc) sowie jedwede Form von elektronischen Hilfsmitteln.*