**Universität des Saarlandes**
**FR 6.2 - Informatik**
**Prof. Dr. W.J. Paul**
**Dipl. Inform. M. Klein**
**Dipl. Inform. T. In der Rieden**

# 3. Musterlösung Informatik II

**(Abgabe: 16.05.2003)**

1. **Aufgabe: (Topologische Sortierung)** **(6 + 10 Punkte)**

   (a) Algorithmus:
   Sei $G = (V, E)$ ein DAG mit endlicher Knotenmenge, $i = \#V$
   1. Finde eine beliebige Senke $s$ und nummeriere $s$ mit $i$.
   2. Entferne $s$ aus $V$ und alle Kanten $(v, s)$ aus $E$.
   3. $i - -$
   4. Wenn $i > 0$ dann goto 1 sonst terminiere.

   (b) **Terminierung:**
   1. Da $G$ ein endlicher DAG ist existiert in jedem Reduktionsschritt mindestens eine Senke (s. Vorlesung).
   2. $G$ bleibt durch die obigen Transformationen in jedem Schritt ein endlicher DAG.
   Beweis:
   endlich ist klar (es verschwinden nur Knoten),
   gerichtet ist klar (es kommen keine ungerichteten Kanten dazu),
   azyklisch ist klar (sei $u \to ... \to u$ ein Zyklus im neuen Graph, dann war es auch ein Zyklus im alten Graph, da alle benutzten Knoten und Kanten dort auch schon vorhanden waren)
   3. Der Algorithmus terminiert nach genau #V Schritten wegen Punkt 3 und 4. Da V endlich ist terminiert der Algorithmus in endlicher Zeit.
   **Korrektheit:** Beweis durch Widerspruch:
   **Ann.:** es existiert $(u, v)$ aus $E$ mit $s(u) > s(v)$.
   $\Longrightarrow$ u ist vor v nummeriert worden (da i immer kleiner wird in obigem Algorithmus)
   $\Longrightarrow$ u war zu diesem Zeitpunkt eine Senke im reduzierten Graph, v war zu diesem Zeitpunkt ebenfalls im reduzierten Graph vorhanden
   $\Longrightarrow$ es kann keine Kante $(u, v)$ im reduzierten Graph zu diesem Zeitpunkt gegeben haben
   $\Longrightarrow$ es kann zu keinem Zeitpunkt eine Kante $(u, v)$ im Graph geben haben, da in den vorhergegangenen Schritten keine Kanten $(x, v)$ für beliebige $x$ gelöscht wurden

2. **Aufgabe: (DAGs)** **(4 Punkte)**

   Def.: Ein Graph $G$ hat die Tiefe $n$ wenn die größte Tiefe eines Knotens aus $G$ $n$ beträgt.

   Konstruiere ein Gegenbeispiel zur Aussage:

   Betrachte den Graphen $G = (V, E)$ mit $V = \mathbb{N}, E = (v, v + 1)$

   Annahme: $G$ hat eine Tiefe. Sei $n$ die Tiefe von $G$.

   $\Longrightarrow$ Dann gibt es ein $v \in V$ mit $D(v) = n$. Dann ist aber $D(v + 1) = n + 1$ und das ist ein Widerspruch zur Annahme.

3. **Aufgabe: (Potenzrechnung)** **(6 Punkte)**

   (a) Induktion über m
   **I.A.** m = 0
   $a^{(n+0)} = a^n = a^n \cdot 1 = a^n \cdot a^0$
   **I.V** Es sei gezeigt: $a^{n+m} = a^n \cdot a^m$

**Universität des Saarlandes**
**FR 6.2 - Informatik**
**Prof. Dr. W.J. Paul**
**Dipl. Inform. M. Klein**
**Dipl. Inform. T. In der Rieden**

# 3. Musterlösung Informatik II

**(Abgabe: 16.05.2003)**

**I.S** $m \to m+1$

$a^{n+(m+1)} = a^{(n+m)+1} = a^{(n+m)} \cdot a \overset{I.V.}{=} (a^n \cdot a^m) \cdot a = a^n \cdot (a^m \cdot a) = a^n \cdot a^{m+1}$

(b) Induktion über n

**I.A.** n = 0

$(a \cdot b)^0 = 1 = 1 \cdot 1 = a^0 \cdot b^0$

**I.V** Es sei gezeigt: $(a \cdot b)^n = a^n \cdot b^n$

**I.S** $n \to n+1$

$(a \cdot b)^{n+1} = (a \cdot b)^n \cdot (a \cdot b) \overset{I.V.}{=} a^n \cdot b^n \cdot (a \cdot b) = (a^n \cdot a) \cdot (b^n \cdot b) = a^{n+1} \cdot b^{n+1}$

4. **Aufgabe: (Reihenformel)** **(4 Punkte)**

Induktion über n

I.A. $n = 1$

$$\sum_{i=0}^{n-1}(B-1) \cdot B^i = \sum_{i=0}^{0}(B-1) \cdot B^i = (B-1) \cdot B^0 = B - 1 = B^1 - 1$$

I.V. Es sei gezeigt:

$$\sum_{i=0}^{n-1}(B-1) \cdot B^i = B^n - 1$$

I.S. $n \to n+1$

$$\sum_{i=0}^{n}(B-1) \cdot B^i$$

$$= \sum_{i=0}^{n-1}(B-1) \cdot B^i + (B-1) \cdot B^n$$

$$\overset{IV}{=} B^n - 1 + (B-1) \cdot B^n$$

$$= B^n - 1 + B^{n+1} - B^n$$

$$= -1 + B^{n+1} = B^{n+1} - 1$$

5. **Aufgabe: (Darstellungszerlegung)** **(4 Punkte)**

Beweis durch Umformen

$$< a[n-1:h] >_B \cdot B^h + < a[h-1:0] >_B$$

$$= (\sum_{i=h}^{n-1} a_i \cdot B^{i-h}) \cdot B^h + \sum_{i=0}^{h-1} a_i \cdot B^i$$

$$= \sum_{i=h}^{n-1}(a_i \cdot B^{i-h} \cdot B^h) + \sum_{i=0}^{h-1} a_i \cdot B^i$$

$$= \sum_{i=h}^{n-1}(a_i \cdot B^{i-h+h}) + \sum_{i=0}^{h-1} a_i \cdot B^i$$

**Universität des Saarlandes**
**FR 6.2 - Informatik**
**Prof. Dr. W.J. Paul**
**Dipl. Inform. M. Klein**
**Dipl. Inform. T. In der Rieden**

# 3. Musterlösung Informatik II

**(Abgabe: 16.05.2003)**

$$= \sum_{i=h}^{n-1} a_i \cdot B^i + \sum_{i=0}^{h-1} a_i \cdot B^i$$

$$= \sum_{i=0}^{n-1} a_i \cdot B^i$$

$$=< a[n-1:0] >_B$$

6. **Aufgabe: (Additionsalgorithmus)** **(6 Punkte)**

Beweis: Induktion über n

**I.A.** n=1

$\langle c_0 s_0 \rangle_B = a_0 + b_0 + c_{-}1$

**I.V.**

$\langle a[n-2:0] \rangle_B + \langle b[n-2:0] \rangle_B + c_{-1} = \langle c_{n-2} s[n-2:0] \rangle_B$

**I.S.** n-1 $\rightarrow$ n

$\langle a[n-1:0] \rangle_B + \langle b[n-1:0] \rangle_B + c_{-1}$

$= a_{n-1}B^{n-1} + \langle a[n-2:0] \rangle_B + b_{n-1}B^{n-1} + \langle b[n-2:0] \rangle_B + c_{-1}$

$\stackrel{I.V.}{=} a_{n-1}B^{n-1} + b_{n-1}B^{n-1} + \langle c_{n-2} s[n-2:0] \rangle_B$

$= a_{n-1}B^{n-1} + b_{n-1}B^{n-1} + c_{n-2}B^{n-1} + \langle s[n-2:0] \rangle_B$

$= (a_{n-1} + b_{n-1} + c_{n-2})B^{n-1} + \langle s[n-2:0] \rangle_B$

$= \langle c_{n-1} s_{n-1} \rangle B^{n-1} + \langle s[n-2:0] \rangle_B$

$= \langle c_{n-1} s[n-1:0] \rangle_B$

7. *** Aufgabe: (Surjektivität der Zahlendarstellung)** **(10 Punkte *)**

Folgende Funktion ist gegeben:

$$\langle\ \rangle : \{0,1\}^n \mapsto \{0, \ldots, 2^n - 1\},\ \langle a[n-1:0] \rangle := \sum_{i=0}^{n-1} a_i \cdot 2^i$$

Folgender (einfacher) Algorithmus berechnet diese Funktion rekursiv:

```
FUNCTION foo(a[],i)
{
  IF i==0 THEN
    RETURN a[0];
  ELSE
    RETURN a[i]*2^i + foo(a[],i-1);
}
```

Ein Aufruf `foo(n)` berechnet also die Funktion $\langle (a_{n-1}, \ldots, a_0) \rangle$.

Wenn man nun die Umkehrabbildung $\langle\ \rangle^{-1}$ berechnen will, so muss man obigen Algorithmus lediglich „aufrollen“:

**Universität des Saarlandes**
**FR 6.2 - Informatik**
**Prof. Dr. W.J. Paul**
**Dipl. Inform. M. Klein**
**Dipl. Inform. T. In der Rieden**

# 3. Musterlösung Informatik II

**(Abgabe: 16.05.2003)**

$$\langle\ \rangle^{-1} : \{0, \ldots, 2^n - 1\} \mapsto \{0,1\}^n,\ \langle a\rangle^{-1} := (a_{n-1}, \ldots, a_0)$$

Dabei berechnet man alle $a_i$ $(0 \leq i < n)$ wie folgt:

$$a_i = \left( a - \sum_{k=i+1}^{n-1} a_k \cdot 2^k \right) / 2^i$$

Die Division am Ende der Formel ist natürlich eine ganzzahlige Division.

Der Nachteil an dieser Darstellung ist, dass man, wenn man z.B. lediglich eines der niedrigeren Bits berechnen will, zuerst alle höherwertigen Bits ausrechnen muss.

Folgender Algorithmus ist daher besser geeignet (hier lediglich die Berechnung der einzelnen $a_i$ angegeben, der Rest bleibt wie zuvor):

$$a_i = (a/2^i) \bmod 2$$

Die Divison ist natürlich wieder ganzzahlig.

Hiermit kann man alle Bits getrennt berechnen, ohne zuvor alle höherwertigen kennen zu müssen.

Im Gegensatz zu dem zuerst präsentierten Algorithmus ist hier die Korrektheit alles andere als offensichtlich. Daher muss diese noch bewiesen werden.

Der Beweis ist erbracht, wenn man zeigen kann, dass für alle $a \in \{0, \ldots, 2^n - 1\}$ gilt:

$$\langle \langle a \rangle^{-1} \rangle = a$$

Präziser gesagt, müssen wir zeigen:

$$\sum_{i=0}^{n-1} ((a/2^i) \bmod 2) \cdot 2^i = a$$

Beweis durch Induktion über $n$.

- *Induktionsanfang:* $n = 1$, d.h. $a = \langle a_0 \rangle = a_0$

$$\begin{aligned}
\sum_{i=0}^{n-1} ((a/2^i) \bmod 2) \cdot 2^i &= \sum_{i=0}^{0} ((a_0/2^i) \bmod 2) \cdot 2^i \\
&= ((a_0/2^0) \bmod 2) \cdot 2^0 \\
&= (a_0 \bmod 2) \cdot 1 \\
&= a_0 \bmod 2 \\
&\overset{(*)}{=} a_0 \\
&= a
\end{aligned}$$

  Die Gleichung (*) gilt, weil $a_0 \in \{0,1\}$.

**Universität des Saarlandes**
**FR 6.2 - Informatik**
**Prof. Dr. W.J. Paul**
**Dipl. Inform. M. Klein**
**Dipl. Inform. T. In der Rieden**

# 3. Musterlösung Informatik II

**(Abgabe: 16.05.2003)**

- *Induktionsvoraussetzung:* Stellen $a$ als $\langle a[n-1:0]\rangle$ dar. Dann gilt:

$$\sum_{i=0}^{n-1}((\langle a[n-1:0]\rangle/2^i) \bmod 2)\cdot 2^i = \langle a[n-1:0]\rangle$$

- *Induktionsschritt:* $n \to n+1$

$$\begin{aligned}
& \sum_{i=0}^{n}((\langle a[n:0]\rangle/2^i) \bmod 2)\cdot 2^i \\
= \quad & (\underbrace{(\langle a[n:0]\rangle/2^n)}_{=a_n\ (*1)} \bmod 2)\cdot 2^n + \sum_{i=0}^{n-1}((\langle a[n:0]\rangle/2^i) \bmod 2)\cdot 2^i \\
= \quad & (a_n \bmod 2)\cdot 2^n + \sum_{i=0}^{n-1}((\langle a[n:0]\rangle/2^i) \bmod 2)\cdot 2^i \\
\overset{(*2)}{=} \quad & a_n\cdot 2^n + \sum_{i=0}^{n-1}((\langle a[n:0]\rangle/2^i) \bmod 2)\cdot 2^i \\
= \quad & a_n\cdot 2^n + \sum_{i=0}^{n-1}(((a_n\cdot 2^n + \langle a[n-1:0]\rangle)/2^i) \bmod 2)\cdot 2^i \\
= \quad & a_n\cdot 2^n + \sum_{i=0}^{n-1}((a_n\cdot 2^{n-i} + \langle a[n-1:0]\rangle/2^i) \bmod 2)\cdot 2^i \\
\overset{(*3)}{=} \quad & a_n\cdot 2^n + \sum_{i=0}^{n-1}((\underbrace{(a_n\cdot 2^{n-i}) \bmod 2}_{=0\ (*4)} + (\langle a[n-1:0]\rangle/2^i) \bmod 2) \bmod 2)\cdot 2^i \\
= \quad & a_n\cdot 2^n + \sum_{i=0}^{n-1}((((\langle a[n-1:0]\rangle/2^i) \bmod 2) \bmod 2)\cdot 2^i \\
\overset{(*5)}{=} \quad & a_n\cdot 2^n + \sum_{i=0}^{n-1}((\langle a[n-1:0]\rangle/2^i) \bmod 2)\cdot 2^i \\
\overset{\text{IV}}{=} \quad & a_n\cdot 2^n + \langle a[n-1:0]\rangle \\
\overset{\text{Def}}{=} \quad & \langle a[n:0]\rangle \\
= \quad & a
\end{aligned}$$

Anmerkungen:

(a) $(*1)$ Gilt nach Definition (Erinnerung: Division ganzzahlig)

(b) $(*2)$ $a_n \in \{0,1\}$

(c) $(*3)$ Rechenregel:

$$(a+b) \bmod n = (a \bmod n + b \bmod n) \bmod n$$

Ließe man das letzte modulo $n$ weg, so hätte man lediglich Kongruenz modulo $n$. Wir sind aber an Gleichheit interessiert.

**Universität des Saarlandes**
**FR 6.2 - Informatik**
**Prof. Dr. W.J. Paul**
**Dipl. Inform. M. Klein**
**Dipl. Inform. T. In der Rieden**


# 3. Musterlösung Informatik II

**(Abgabe: 16.05.2003)**

(d) ($*4$) Gilt, weil $2^{n-i} \geq 2$ (und natürlich immer Vielfaches von 2) für alle $i \in \{0, \ldots, n-1\}$. Daher ist der Rest bei Divison durch 2 immer 0.

(e) ($*5$) Rechenregel:

$$(a \bmod n) \bmod n = a \bmod n$$