



1. Musterlösung Informatik II (Abgabe: 02.05.2003)

1. Aufgabe: (Mengenvereinigung)

(8 Punkte)

- (a) O.B.d.A. Induktionsbeweis über die Anzahl der Elemente der Menge M (Beweis für N analog)

Um den Induktionsschluß zu verdeutlichen, machen wir zusätzlich zum Induktionsanfang für $\#M=0$ auch die Überprüfung für $\#M=1$.

Induktionsanfang:

$M=\emptyset$: Dann gilt: $\#M=0$

$$\Rightarrow \#(M \cup N) = \#N \leq \#N + 0 = \#N + \#M$$

$M=\{x\}$: Dann gilt: $\#M=1$

1.Fall: $x \in N$

$$\Rightarrow \#(M \cup N) = \#N \leq \#N + 1 = \#N + \#M$$

2.Fall: $x \notin N$

$$\Rightarrow \#(M \cup N) = \{x\} \cup \#N \leq \#N + 1 = \#N + \#M$$

Induktionsvoraussetzung:

Wir setzen voraus, dass für die Menge M gilt:

$$\#(M \cup N) \leq \#N + \#M$$

Induktionsschluß:

Sei $\tilde{M} = M \cup \{x\}$ mit $x \notin M$. Dann gilt: $\#\tilde{M} = \#M + 1$

1.Fall: $x \in N$

$$\Rightarrow \#(\tilde{M} \cup N) = \#(M \cup N) \stackrel{I.V.}{\leq} \#N + \#M \leq \#N + \#\tilde{M}$$

2.Fall: $x \notin N$

$$\Rightarrow \#(\tilde{M} \cup N) = \#(M \cup \{x\} \cup N) = \#(M \cup N) + 1 \stackrel{I.V.}{\leq} \#N + \#M + 1 = \#N + \#\tilde{M}$$

- (b) O.B.d.A. Induktionsbeweis über die Anzahl der Elemente der Menge M (Beweis für N analog!)

Induktionsanfang:

$M=\emptyset$: Dann gilt: $\#M=0$

Insbesondere gilt dann zum einen: $M \cap N = \emptyset \cap N = \emptyset$

und zum anderen: $\#(M \cup N) = \#N = \#N + 0 = \#N + \#M$

Für $M=\emptyset$ ist die Behauptung also erfüllt.

$M=\{x\}$: Dann gilt: $\#M=1$

1.Fall: $M \cap N = \{x\} \cap N = \emptyset$

Dann ist x kein Element der Menge N . Daher gilt hier Gleichheit:

$$\#(M \cup N) = \{x\} \cup \#N = \#N + 1 = \#N + \#M$$

2.Fall: $M \cap N = \{x\} \cap N = \{x\} \neq \emptyset$

Dann ist x Element der Menge N und es gilt hier keine Gleichheit:

$$\#(M \cup N) = \#N < \#N + 1 = \#N + \#M$$

Durch die beiden Fälle werden alle Möglichkeiten abgedeckt, die für das Element x der Menge M in Frage kommen. Dadurch ist die Äquivalenzbehauptung für $M=\{x\}$ bewiesen.



1. Musterlösung Informatik II

(Abgabe: 02.05.2003)

Induktionsvoraussetzung:

Wir setzen voraus, dass für die Menge M gilt:

$$\#(M \cup N) = \#N + \#M \iff M \cap N = \emptyset$$

Induktionsschluß:

Sei $\tilde{M} = M \cup \{x\}$ mit $x \notin M$. Dann gilt: $\#\tilde{M} = \#M + 1$

Zusätzlich setzen wir voraus, dass $M \cap N = \emptyset$ gilt.

1. Fall: $\tilde{M} \cap N = \emptyset$

Dann ist x kein Element der Menge N . Daher gilt hier Gleichheit:

$$\#(\tilde{M} \cup N) = \#(M \cup \{x\} \cup N) = \#(M \cup N) + 1 \stackrel{I.V.}{=} \#N + \#M + 1 = \#N + \#\tilde{M}$$

2. Fall: $\tilde{M} \cap N \neq \emptyset$

Dann gilt insbesondere $\tilde{M} \cap N = \{x\}$, da $M \cap N = \emptyset$.

Dann ist x Element der Menge N und es gilt hier keine Gleichheit:

$$\#(\tilde{M} \cup N) = \#(M \cup N) \stackrel{I.V.}{=} \#N + \#M < \#N + \#M + 1 = \#N + \#\tilde{M}$$

Die beiden Fälle beweisen die Äquivalenz. Induktiv folgt also die Behauptung.

2. Aufgabe: (Mengenkreuzprodukt)

(6 Punkte)

O.B.d.A. Induktionsbeweis über die Anzahl der Elemente der Menge M (Beweis für N analog)

Induktionsanfang:

$M = \emptyset$: Dann gilt: $\#M = 0$

$$\implies \#(M \times N) = \#\emptyset = 0 = 0 \cdot \#N$$

$M = \{x\}$: Dann gilt: $\#M = 1$

$$\implies \#(M \times N) = \#\{(x, n) \mid n \in N\} = \#N = 1 \cdot \#N$$

Induktionsvoraussetzung:

Wir setzen voraus, dass für die Menge M gilt:

$$\#(M \times N) = \#M \cdot \#N$$

Induktionsschluß:

Sei $\tilde{M} = M \cup \{x\}$ mit $x \notin M$. Dann gilt: $\#\tilde{M} = \#M + 1$

$$\implies \#(\tilde{M} \times N) = \#(\{(m, n) \mid m \in M, n \in N\} \cup \{(x, n) \mid n \in N\})$$

$$\stackrel{1(b)}{=} \#\{(m, n) \mid m \in M, n \in N\} + \#\{(x, n) \mid n \in N\} \quad \text{da } x \notin M$$

$$\stackrel{Def.}{=} \#(M \times N) + \#N$$

$$\stackrel{I.V.}{=} (\#M \cdot \#N) + (1 \cdot \#N) = (\#M + 1) \cdot \#N$$

3. Aufgabe: (Mengenpotenzierung)

(6 Punkte)

Induktionsbeweis über n

Induktionsanfang:

$n = 1$: Dann gilt:

$$\#(M^1) = \#M = \#M^1$$

Induktionsvoraussetzung:



1. Musterlösung Informatik II

(Abgabe: 02.05.2003)

Wir setzen voraus, dass für die Menge M gilt:

$$\#(M^n) = (\#M)^n$$

Induktionsschluß:

$$\#(M^{n+1}) = \#(M^n \times M)$$

$$\stackrel{A2}{=} \#(M^n) \cdot \#M$$

$$\stackrel{I.V.}{=} (\#M)^n \cdot \#M = (\#M)^{n+1}$$

4. Aufgabe: (endliche Mengen)

(5 Punkte)

- (a) $A \cup B = \{1, 2, 4, \alpha, \beta, \gamma, \emptyset\}$;
 $|A \cup B| = 7$;
- (b) $A \cap B = \emptyset$;
 $|A \cap B| = 0$;
- (c) $A^2 = A \times A$
 $= \{(1,1), (1,2), (1,4), (2,1), (2,2), (2,4), (4,1), (4,2), (4,4)\}$;
 $|A^2| = 9$;
- (d) $A \times B = \{(1,\alpha), (1,\beta), (1,\gamma), (1,\emptyset), (2,\alpha), (2,\beta), (2,\gamma), (2,\emptyset), (4,\alpha), (4,\beta), (4,\gamma), (4,\emptyset)\}$;
 $|A \times B| = 12$;
- (e) $|A^2 \times B^3 \times A|$
 $= |A^2| \cdot |B^3| \cdot |A|$
 $= 9 \cdot 64 \cdot 3$
 $= 1728$;

5. Aufgabe: (Boolesche Ausdrücke)

(10 Punkte)

- (a) $(\sim (1 \vee (0 \wedge X_1)))$
 ist ein vollständig geklammerter Boolescher Ausdruck, da
 $1, 0, X_1 \in B_1$,
 $(0 \wedge X_1) \in B_2$,
 $(1 \vee (0 \wedge X_1)) \in B_3$ und
 $(\sim (1 \vee (0 \wedge X_1))) \in B_4$
- (b) $((0 \vee (\sim 1)) \wedge (X_2 \vee (\sim (\sim X_{52}))))$
 ist ein vollständig geklammerter Boolescher Ausdruck, da
 $0, 1, X_2, X_{52} \in B_1$,
 $(\sim 1), (\sim X_{52}) \in B_2$,
 $(0 \vee (\sim 1)), (\sim (\sim X_{52})) \in B_3$,
 $(X_2 \vee (\sim (\sim X_{52}))) \in B_4$ und
 $((0 \vee (\sim 1)) \wedge (X_2 \vee (\sim (\sim X_{52})))) \in B_5$
- (c) $(\sim (\sim (X_1 \wedge X_2)))$
 ist kein vollständig geklammerter Boolescher Ausdruck, da die Anzahl der schließenden Klammern die der öffnenden um eins übertrifft.



1. Musterlösung Informatik II (Abgabe: 02.05.2003)

(d) $((\sim (\sim (X_1 \wedge X_2))) \wedge 1)$

ist ein vollständig geklammerter Boolescher Ausdruck, da

$X_1, X_2, 1 \in B_1$,

$(X_1 \wedge X_2) \in B_2$,

$(\sim (X_1 \wedge X_2)) \in B_3$,

$(\sim (\sim (X_1 \wedge X_2))) \in B_4$ und

$((\sim (\sim (X_1 \wedge X_2))) \wedge 1) \in B_5$

(e) $((\sim (\sim (X_1 \wedge X_2))) \vee ((X_{52} \wedge (\sim X_{51}))))$

ist kein vollständig geklammerter Boolescher Ausdruck, da

$(X_{52} \wedge (\sim X_{51})) \in B_3$, aber

$((X_{52} \wedge (\sim X_{51}))) \notin BA$, da keine Regel der Form

$a \in B_i \Rightarrow (a) \in B_{i+1}$ existiert.

6. Aufgabe: (Polnische Notation)

(8 Punkte)

(a) $\cdot(+ (2, 4), -(2, 1)) = \cdot(6, 1) = 6$

(b) $+(-(2, \cdot(4, 2)), 1) = +(-(2, 8), 1) = +(-6, 1) = -5$

(c) $2/2/2/2 = /(/(/(2, 2), 2), 2)$ oder $/(2, /(2, /(2, 2)))$ je nachdem ob man die Division als rechts- oder linksassoziativ betrachtet.

(d) $2 - - - - 2 = -(2, -(0, -(0, -(0, 2))))$

7. * Aufgabe: (Polnische Notation 2)

(10 Punkte *)

Es soll an dieser Stelle ein Stack-basierter Algorithmus gezeigt werden, der die Auswertung von Ausdrücken in polnischer Notation realisiert.

Voraussetzungen: Da das resultierende Programm einfach bleiben soll, sind vorher einige Annahmen zu machen:

- Die Eingabe ist in korrekter polnischer Notation, d.h. es findet im Programm keine Überprüfung der Syntax statt.
- Der Stack kann ausschließlich reelle Zahlen und Operatoren aufnehmen.
- Die Ausgabe ist eine reelle Zahl.

Stacks: Ein Stack besitzt die sog. LIFO (Last In, First Out) Eigenschaft, d.h. dasjenige Element, das dem Stack als letztes hinzugefügt wurde, steht zuoberst auf dem Stack. Folgende Zugriffsoperationen sind zulässig:

- **push(x)**: Legt das Element x auf den Stack.
- **pop()**: Entfernt das oberste Element vom Stack.
- **top()**: Liefert das oberste Element vom Stack, ohne es zu entfernen.

Beschreibung des Algorithmus: Der Stack ist zunächst leer. Der Algorithmus läuft in folgenden Schritten ab:

- Solange die Eingabe nicht leer ist, lese das nächste Zeichen ein (von links nach rechts). Nun gibt es drei Möglichkeiten:
 - Wenn das aktuelle Zeichen eine Klammer oder ein Komma ist, gehe zu 1.
 - Ist das aktuelle Zeichen ein Operator, so stelle es auf den Stack und gehe zu 1.



1. Musterlösung Informatik II

(Abgabe: 02.05.2003)

iii. Handelt es sich um eine Zahl, so prüfe, ob das oberste Element des Stacks auch eine Zahl ist.

A. Falls ja: Entferne das oberste Zeichen vom Stack und speichere es zwischen. Entferne das neue oberste Element vom Stack (muss ein Operator sein, wenn die Syntax der Eingabe korrekt ist). Wende diesen Operator auf die beiden Zahlen an und definiere das Ergebnis als das neue aktuelle Zeichen. Gehe zu Schritt (c).

B. Falls nein: Schreibe das aktuelle Zeichen auf den Stack und gehe zu 1.

(b) Entferne das oberste Element vom Stack (sollte bei korrekter Eingabesyntax das einzige und eine Zahl sein) und gib es aus. Das ist das Ergebnis der Auswertung. Der Stack ist nun wieder leer.

Pseudocode:

```

WHILE ( _non_empty(input) ) DO
{
  c = _get_char(input);
  IF ( c=='(' || c=='(' || c=='(' || c==',' ) THEN CONTINUE;
  ELSE
  {
    IF ( _is_operator(c) ) THEN
    {
      PUSH(c);
      CONTINUE;
    }
    ELSE
    {
      WHILE ( _is_number(TOP()) ) DO
      {
        x = POP();
        op = POP();
        c = x op c;
      }
      PUSH(c);
      CONTINUE;
    }
  }
}
RETURN(POP() );

```

Anwendungsbeispiel: Wir wenden nun den soeben erstellten Algorithmus auf das Beispiel von Aufgabe 6(b) an.

Zunächst die Bedeutung der Spaltenüberschriften:

- *Eingabe* stellt den Eingabestring *nach* der Ausführung des aktuellen Arbeitsschritts dar.
- *Position* gibt an, in welchem Fall sich der Algorithmus gerade befindet (korrespondiert mit obiger Beschreibung).



1. Musterlösung Informatik II

(Abgabe: 02.05.2003)

- *Stack* zeigt den Inhalt des Stacks *nach* Ausführung des aktuellen Schritts an. Das am weitesten links stehende Element bezeichnet das oberste auf dem Stack.
- Alle anderen Variablen entsprechen den im Pseudocode aufgeführten.
- $\dots$ bedeutet, dass die Variable entweder keinen Wert enthält oder dass ihr Wert in diesem Arbeitsschritt nicht von Bedeutung ist.

<i>Eingabe</i>	<i>Position</i>	<i>c</i>	<i>x</i>	<i>op</i>	<i>Stack</i>
$+(- (2, \cdot (4, 2)), 1)$	$\dots$	$\dots$	$\dots$	$\dots$	leer
$- (2, \cdot (4, 2)), 1)$	1.(b)	+	$\dots$	$\dots$	+
$- (2, \cdot (4, 2)), 1)$	1.(a)	(	$\dots$	$\dots$	+
$(2, \cdot (4, 2)), 1)$	1.(b)	-	$\dots$	$\dots$	- +
$2, \cdot (4, 2)), 1)$	1.(a)	(	$\dots$	$\dots$	- +
$, \cdot (4, 2)), 1)$	1.(c)i.	2	$\dots$	$\dots$	2 - +
$\cdot (4, 2)), 1)$	1.(a)	,	$\dots$	$\dots$	2 - +
$(4, 2)), 1)$	1.(b)	.	$\dots$	$\dots$	. 2 - +
$4, 2)), 1)$	1.(a)	(	$\dots$	$\dots$	. 2 - +
$, 2)), 1)$	1.(c)i.	4	$\dots$	$\dots$	4 . 2 - +
$2)), 1)$	1.(a)	,	$\dots$	$\dots$	4 . 2 - +
$)), 1)$	1.(c)ii.	2	4	.	2 - +
	1.(c)ii.	8	2	-	-6 +
$), 1)$	1.(a)	)	$\dots$	$\dots$	-6 +
$, 1)$	1.(a)	)	$\dots$	$\dots$	-6 +
$1)$	1.(a)	,	$\dots$	$\dots$	-6 +
$)$	1.(c)ii.	1	-6	+	-5
leer	1.(a)	)	$\dots$	$\dots$	-5
leer	2.	$\dots$	$\dots$	$\dots$	leer

Der Algorithmus terminiert und das Ergebnis -5 wurde ausgegeben.