

---

# Diskrete Mathematik 2

FS 2013

# Contents

<b>1</b>	<b>Erste Woche</b>	<b>1</b>
1.1	Semesterablauf . . . . .	1
1.2	Quantifizierung . . . . .	1
1.3	Semantik . . . . .	3
<b>2</b>	<b>Zweite Woche</b>	<b>4</b>
2.1	Freie/Gebundene Variablen . . . . .	4
<b>3</b>	<b>Dritte Woche</b>	<b>6</b>
3.1	Saetze zur Quantifizierung . . . . .	6
<b>4</b>	<b>Vierte Woche</b>	<b>8</b>
4.1	Saetze zur Quantifizierung (Fortsetzung) . . . . .	8
4.2	Anwendung . . . . .	9
<b>5</b>	<b>Fuenfte Woche</b>	<b>10</b>
5.1	Magisches Quadrat . . . . .	10
5.2	Mathematische Induktion . . . . .	10
<b>6</b>	<b>Sechste Woche</b>	<b>12</b>
6.1	Vollstaendige Induktion . . . . .	12
<b>7</b>	<b>Siebte Woche</b>	<b>16</b>
7.1	Zahlentheorie . . . . .	16
<b>8</b>	<b>Achte Woche</b>	<b>18</b>
8.1	Euklidische Division . . . . .	18
8.2	Fundamentaler Schleifen-Satz . . . . .	20
<b>9</b>	<b>Neunte Woche</b>	<b>21</b>
9.1	Euklid auf $\mathbb{Z}$ , $\mathbb{N}$ . . . . .	21

# 1 Erste Woche

## 1.1 Semesterablauf

- Arithmetik in  $\mathbb{Z}$
- Modulares Rechnen
- Gruppen
- RSA
- Quantifizierung
- Induktion
  - Rekursion
  - Invarianten

- 
- Kein Laptop
  - Zwischenpruefung: 30.04.2013 (1 Stunde)
  - 5. Maerz 2013 Unterricht nur bis 18:20
- 

- Buecher:
    - Gries/Schneider  
A logical approach to Discrete Math  
Springer, 1993
    - Jean Gallier  
Discrete Math  
Springer, 2010
    - Struckermann/Waetiger  
Mathematik fuer Informatiker  
Spektrum, 2007
- 

## 1.2 Quantifizierung

$$\mathbb{N} = \begin{cases} \{0, 1, 2, \dots\} (?) \\ \{1, 2, \dots\} (?) \end{cases}$$

$$\sum_{i=1}^n i^2 = 1^2 + 2^2 + \dots + n^2$$

$$\sum_{i=1}^{-1} i^2 = \begin{cases} \text{ungueltig (?) } \\ 1^2 \text{ (?) } \\ 1^2 + 0^2 + (-1)^2 \text{ (?) } \\ 0 \text{ (} \rightarrow ja, \text{ Neutrales Element) } \end{cases}$$

$$\sum_{i=1}^n i^2 + 1 = 1^2 + 2^2 + \dots + n^2 + 1 \text{ (?)}$$

$$\sum_{i=1}^n (i^2 + 1) = (1^2 + 1) + (2^2 + 1) + \dots + (n^2 + 1) \text{ (?)}$$

$$\sum_{\substack{i=1 \\ \text{odd}(i)}}^n i^2 = 1^2 + 3^2 + \dots + n^2 \text{ , falls } \text{odd}(n), \text{ sonst } (n-1)^2$$

$$\prod_{i=1}^n i^2 = 1^2 * 2^2 * \dots * n^2$$

$$\prod_{i=1}^{-1} i^2 = 1 \text{ (neutrales Element)}$$

(Java ==)

$$\forall_{i=0}^{n-1} (b[i] == 0) = (b[0] == 0) \wedge (b[1] == 0) \wedge \dots \wedge (b[n-1] == 0)$$

$$\exists_{i=0}^{n-1} (b[i] == 0) = (b[0] == 0) \vee (b[1] == 0) \vee \dots \vee (b[n-1] == 0)$$

$$\sum_{i=1}^n i^2 = (\sum i : \mathbb{N} \mid 1 \leq i \leq n : i^2)$$

$$(\sum i : \mathbb{N}, j : \mathbb{N} \mid 1 \leq i \leq 2 \wedge 1 \leq j \leq 3 : i^j)$$

$$(\circ v_1 : T_1, \dots, v_n : T_n \mid R : P)$$

$$- \circ : T \times T \rightarrow T \text{ (wobei } T \text{ ein Typ ist)}$$

$$\text{Bsp: } + : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

$$+(3, 4) = 7$$

$$\text{ABELSCHES MONOID} \left\{ \begin{array}{l} a \circ b = b \circ a \text{ fuer alle } a, b : T \text{ (Symmetrie)(Kommutativitaet)} \\ \text{MONOID} \left\{ \begin{array}{l} (a \circ b) \circ c = a \circ (b \circ c) \text{ fuer alle } a, b, c : T \text{ (Assoziativitaet)} \\ u \circ a = a = a \circ u \\ \text{es gibt ein } u : T, \text{ so dass fuer alle } a : T \text{ (neutrales Element)} \end{array} \right. \end{array} \right.$$

$\circ$	$T$	$u$
$+\sum$	$\mathbb{Z}$	$0$
$*\prod$	$\mathbb{Z}$	$1$
$\forall$	$\mathbb{B}$	$true$
$\exists$	$\mathbb{B}$	$false$

---

String mit Konkatination: Nicht-abelsches Monoid

$(\text{"a"} + \text{"b"}) + \text{"c"} \text{ equals } \text{"a"} + (\text{"b"} + \text{"c"})$

$\text{"a"} + \text{""} \text{ equals } \text{"a"}$

$\text{"a"} + \text{"b"} \text{ !equals } \text{"b"} + \text{"a"} \text{ (nicht equals)}$

-  $T_1, \dots, T_n$  Datentypen

-  $V_1, \dots, V_n$  Variablen

alle paarweise verschieden

$V_i$  vom Typ:  $T_i$

-  $R$  : boolescher Ausdruck, kann  $V_1 \dots V_n$  enthalten, Bereich (Range)

-  $P$  : beliebiger Ausdruck vom Typ  $T$ , kann  $V_1 \dots V_n$  enthalten, Koerper (Body)

Typ der Quantifizierung :  $T$

---

$(\forall i : \mathbb{N} \mid 0 \leq i \leq n : b[i] = 0)$  und das Ganze ist :  $\mathbb{B}$

$(\circ V_1 : T_1 \mid R : P)$  wobei  $T_1 : \mathbb{N}, P : \mathbb{B}$

$\wedge : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$

$P : T_1 \times T_2 \times \dots \times T_n \rightarrow T$

---

### 1.3 Semantik

Bsp:  $(+i : \mathbb{Z} \mid -1 \leq i \leq 2 : i^2)$

1. Fall (Topf  $\neq \emptyset$ )

Von  $\mathbb{Z}$  alle Zahlen ausfiltern  $(-1, 0, 1, 2)$  (Menge)

$\rightarrow^{1^2} ((-1)^2, 1^2, 0^2, 2^2)(1, 1, 0, 4)$  (Multimenge)

$\rightarrow 2^2 + 1^2 + (-1)^2 + 0^2$

2. Fall (Topf = 0)

$\rightarrow$  Topf leer  $\rightarrow$  Resultat: Neutrales Element (von  $+$ )  $\rightarrow 0$

---

Beispiele:

$$1) (+ i : \mathbb{N} \mid 0 \leq i < 4 : i * 8) = (0 * 8) + (1 * 8) + \dots$$

$$2) (* i : \mathbb{N} \mid 0 \leq i < 3 : i + 1) = (0 + 1) * (1 + 1) * \dots$$

$$3) (\wedge i : \mathbb{N} \mid 0 \leq i < 2 : i * d \neq 6) = ((0 * d) \neq 6) \wedge ((1 * d) \neq 6) \wedge \dots$$

$$4) (\vee i : \mathbb{N} \mid 0 \leq i < 21 : b[i] = 0) = (b[0] == 0) \vee (b[1] == 0) \vee \dots$$

$$5) (\sum k : \mathbb{N} \mid k^2 = 4 : k^2) = 2$$

$$6) (\sum k : \mathbb{Z} \mid k^2 = 4 : k^2) = 2 + (-2) = 0$$

---

## 2 Zweite Woche

### 2.1 Freie/Gebundene Variablen

$$(\circ v_1 : T_1, \dots, v_n : T_n \mid R : P)$$

---

$$\text{E1:} \quad (\sum i : \mathbb{Z} \mid 0 \leq i < n : i^2)$$

- Wert haengt von  $n$  ab, nicht von  $i$

$$n = 3 : \quad 0 \quad 1 \quad 2$$

$$0^2 + 1^2 + 2^2 = 5$$

$$n = 0 : \text{kein } i$$

$$0 \text{ (neutral +)}$$

---

$$\text{E2:} \quad (\sum j : \mathbb{Z} \mid 0 \leq j < n : j^2)$$

$$n = 3 \rightarrow 5$$

$$n = 0 \rightarrow 0$$

---

$$E3: (\sum i \textcircled{1} : \mathbb{Z} \mid 0 \leq i \textcircled{2} < n : i^2 \textcircled{3}) + 1 \textcircled{4}$$

$(\leftarrow \rightarrow)$  : Gueltigkeitsbereich von  $i$  (scope)

$i$  tritt hier 4 mal auf (occurs)

Auftreten (occurrences)  $\textcircled{1}, \textcircled{2}, \textcircled{3}$  gebunden

Auftreten  $\textcircled{4}$  frei

$\textcircled{2}$  und  $\textcircled{3}$  gebunden an  $\textcircled{1}$

$\textcircled{2}$  und  $\textcircled{3}$  angewandte Auftreten (applied)

$\textcircled{1}$  bindende, deklarierende Auftreten (binding)

Eine Variable heisst frei in einem Ausdruck  $E$  (expresion), falls sie in  $E$  frei vorkommt.

$FV(E)$  = Menge der freie Variablen von  $E$

$FV(E_3) = \{ 'n', 'i' \}$  (Die Variablennamen und nicht die Werte der Variablen)

$$x, y : \mathbb{Z}$$

$$x = 3, y = 5$$

$$\{x, y\} = \{3, 5\}$$

$$x = y = 3$$

$$\{x, y\} = \{3\}$$

$$x + y * 2$$

$$y, 2 : * \text{ Operator}$$

dann das Resultat mit  $x$  und  $+$  Operator

$$E4: (\sum i : \mathbb{Z} \mid 0 \leq i < n : i^2) * (\sum i : \mathbb{Z} \mid 0 \leq i < n : i^3)$$

$$FV(E_4) = \{ 'n' \}$$

$$E5: (\prod n \mid k \leq n \leq l : (\sum i : \mathbb{Z} \mid 0 \leq i < n : i^2) * (\sum i : \mathbb{Z} \mid 0 \leq i < n : i^3))$$

$$FV(E_5) = \{ 'k', 'l' \}$$

$$E6: (\sum i : \mathbb{Z} \mid 0 \leq i \leq (\sum i : \mathbb{Z} \mid 2 \leq i < 3 : i^2) : i^2)$$

$$FV(E_6) = \emptyset$$

Ein Ausdruck  $E$  ohne freie Variablen ( $FV(E) = \emptyset$  oder  $\{ \}$ ) heisst geschlossen

---


$$(\sum i : \mathbb{Z} \mid 1 \leq i < 2 : (\sum j : \mathbb{Z} \mid 1 \leq j < 3 : i + j))$$

i zuerst:

$$i : \quad \quad \quad \begin{matrix} 1 \\ (\sum j : \mathbb{Z} \mid 1 \leq j < 3 : 1 + j) \end{matrix} + \begin{matrix} 2 \\ (\sum j : \mathbb{Z} \mid 1 \leq j < 3 : 2 + j) \end{matrix}$$

$$j : \quad \begin{matrix} 1 & 2 & 3 \\ ((1 + 1) + (1 + 2) + (1 + 3)) \end{matrix} + \begin{matrix} 1 & 2 & 3 \\ ((2 + 1) + (2 + 2) + (2 + 3)) \end{matrix}$$

j zuerst:

$$j : \quad \quad \quad \begin{matrix} 1 & 2 & 3 \\ (\sum i : \mathbb{Z} \mid 1 \leq i < 2 : ((i + 1) + (i + 2) + (i + 3))) \end{matrix}$$

$$i : \quad \quad \quad \begin{matrix} 1 \\ ((1 + 1) + (1 + 2) + (1 + 3)) \end{matrix} + \begin{matrix} 2 \\ ((2 + 1) + (2 + 2) + (2 + 3)) \end{matrix}$$


---

### 3 Dritte Woche

#### 3.1 Sätze zur Quantifizierung

Satz (Dummy renaming)

$$(\circ v \mid R : P) = (\circ w \mid R[v \leftarrow w] : P[v \leftarrow w])$$

Voraussetzung:  $w \notin FV(R) \cup FV(P)$

Dabei:  $E[v \leftarrow F]$  bezeichnet exakt denselben Ausdruck wie  $E$ , aber alle freien Auftreten von  $v$  ersetzt durch  $(F)$ .

wobei  $E, F$  : Ausdruck,  $v$  : Variable

---

$$\text{Bsp: } (i + 5)[i \leftarrow j + 3] = (j + 3) + 5$$

wobei  $(i + 5) : E, [i : v, j + 3 : F]$

$$(i * 5)[i \leftarrow j + 3] = (j + 3) * 5$$

$$(\sum i \mid \text{true} : i^2)[i \leftarrow j + 3] = (\sum i \mid \text{true} : i^2)$$

$$(\sum i \mid \text{true} : i^2) = (\sum j \mid \text{true} : j^2)$$



$$= (\sum j \mid true[i \leftarrow j] : i^2[i \leftarrow j])$$

$$= (\sum j \mid true : j^2)$$

$$42[i \leftarrow j + 3] = 42 \text{ "Man kann die Bedeutung des Universums nicht aendern."}$$

Es ist ein Unterschied, ob die Ersetzung innerhalb oder ausserhalb einer Quantifizierung angegeben wird.

$$(\sum i \mid true : i^2)[i \leftarrow j + 3]$$

Hier sollen alle freien Auftreten von Variable  $i$  in  $(\sum i \mid true : i^2)$  durch  $j + 3$  ersetzt werden.  
Aber alle Auftreten von  $i$  sind in diesem Ausdruck gebunden, also ist nichts zu ersetzen.

Dummy renaming sagt aus, dass wir die gebundenen Auftreten einer Variablen innerhalb einer Quantifizierung konsistent umbenennen duerfen, solange wir dabei keine freien Variablen einfangen.

$$\begin{aligned} & (\sum i \mid 1 \leq i \leq n : i^2) \\ &= (\sum j \mid 1 \leq i \leq n[i \leftarrow j] : i^2[i \leftarrow j]) \\ &= (\sum j \mid 1 \leq j \leq n : j^2) \end{aligned}$$

Hier sind die Ersetzungen innerhalb der Quantifizierung.  
Und beachten Sie: im Teilausdruck  $1 \leq i \leq n$  ist die Variable  $i$  frei,  
daher liefert  $1 \leq i \leq n[i \leftarrow j]$  den Ausdruck  $1 \leq j \leq n$

Im Gesamtausdruck  $(\sum i \mid true : i^2)$  sind alle Auftreten von  $i$  hingegen gebunden.  
Aber in diesem Ausdruck wollen wir auch nicht ersetzen, sondern eben in den beiden Teilausdruecken.

Ein Auftreten einer Variablen kann in einem Teilausdruck frei sein, aber im Gesamtausdruck gebunden.  
Ob ein Auftreten frei oder gebunden ist, hängt immer vom betrachteten (Teil-)Ausdruck ab.

$$\text{Bsp: } (\sum i \mid 1 \leq i \leq n : i^2)$$

$$\text{wobei } i : v, (1 \leq i \leq n) : R, i^2 : P$$

$$= (\sum j \mid (1 \leq i \leq n)[i \leftarrow j] : i^2[i \leftarrow j])$$

$$\text{wobei } j : w$$

$$= (\sum j \mid 1 \leq j \leq n : j^2)$$

Aber: Vorsicht:

$$\begin{aligned} & (\sum i : \mid 1 \leq i \leq n : i^2) \\ & n = 0, \quad 0(\text{neutral}+) \\ & n = 1, \quad 1 \end{aligned}$$

haengt von  $n$  ab

$\neq$

$(\sum n : | 1 \leq n \leq n : n^2)$   
 $\infty$  undefiniert

haengt nicht von  $n$  ab

---

## 4 Vierte Woche

### 4.1 Saetze zur Quantifizierung (Fortsetzung)

$$(\sum i \mid 0 \leq i < n : i^2)[n \leftarrow n^2] = (\sum i \mid 0 \leq i < n^2 : i^2)$$

$(\sum i \mid 0 \leq i < n : i^2)[n \leftarrow i + 1] \neq (\sum i \mid 0 \leq i < i + 1 : i^2)$  (geht nicht)  
 freies Auftreten von  $i$  wird gefangen  $\rightarrow$  name clash

$$\begin{aligned} & (\sum i \mid 0 \leq i < n : i^2)[n \leftarrow i + 1] \\ &= (\sum j \mid 0 \leq j < n : j^2)[n \leftarrow i + 1] \\ &= (\sum j \mid 0 \leq j < i + 1 : j^2) \end{aligned}$$


---

#### Empty range

$(\circ v \mid false : P) = u_\circ$  (Neutrales Element)

---

#### One point

Voraussetzung:  $v \notin FV(E)$

$$(\circ v \mid v = E : P) = P[v \leftarrow E]$$

$$\text{Bsp. } (\sum i \mid i = j + 3 : i^2) = i^2[i \leftarrow j + 3] = (j + 3)^2$$

$$(\sum i \mid i = j + i + 3 : i^2) \neq i^2[i \leftarrow j + i + 3] = (j + i + 3)^2 \text{ (geht nicht)}$$


---

#### Split-off term

$$(\circ i \mid 0 \leq i < n + 1 : P) = (\circ i \mid 0 \leq i < n : P) \circ P[i \leftarrow n]$$

$$\begin{aligned} \text{Bsp. } (\sum i \mid 0 \leq i < n + 1 : i^2) &= (\sum i \mid 0 \leq i < n : i^2) + n^2 \\ 0^2 + 1^2 + \dots + (n - 1)^2 + n^2 &= (0^2 + 1^2 + \dots + (n - 1)^2) + n^2 \end{aligned}$$

$n = 0 :$   
 $(\circ i \mid 0 \leq i < 1 : P) = (\circ i \mid 0 \leq i < 0 : P) \circ P[i \leftarrow 0]$   
 $i = 0 :$   
 $P[i \leftarrow 0] \text{ (One point)} = u_{\circ}(\text{empty range}) \circ P[i \leftarrow 0]$

---

## 4.2 Anwendung

Praedikat

$i + 1 > j : Bool$  macht Aussage ueber Werte von freien Variablen

Feld  $b[0...n-1]$  mit ganzen Zahlen;  $n \geq 0$

" $b$  enthaelt eine  $-1$ ."  $\rightarrow$  bedeutet mindestens

$$(\exists i : \mathbb{N} \mid 0 \leq i < n : b[i] = -1)$$

" $b$  enthaelt genau eine  $-1$ ."

$$(\exists i : \mathbb{N} \mid 0 \leq i < n : (b[i] = -1) \wedge (\forall j : \mathbb{N} \mid (0 \leq j < n) \wedge (j \neq i) : b[j] \neq -1))$$

=

$$1 = (\sum i : \mathbb{N} \mid (0 \leq i < n) \wedge (b[i] = -1 : 1))$$

&&

" $b$  enthaelt keine  $-1$ ."

$$(\forall i : \mathbb{N} \mid 0 \leq i < n : b[i] \neq -1)$$

=

$$\neg(\exists i : \mathbb{N} \mid 0 \leq i < n : b[i] = -1) \rightarrow (\neg \text{ ("} b \text{ enthaelt mindestens eine } -1 \text{.")})$$

$$\neg(\exists v \mid R : P) = (\forall v \mid R : \neg P)$$

$$\neg(\forall v \mid R : P) = (\exists v \mid R : \neg P)$$

de Morgan

$$\begin{aligned}\neg(\exists v \mid R : P) &= \neg(P_0 \vee P_1 \vee \dots \vee P_{n-1} \vee P_n) \\ &= ((\neg P_0) \wedge (\neg P_1) \wedge \dots (\neg P_n)) \\ &= (\forall v \mid R : \neg P)\end{aligned}$$

---

## 5 Fuenfte Woche

### 5.1 Magisches Quadrat

Uebungsblatt 2, Aufgabe 3

$$k, i : 1 \leq k \leq n, 1 \leq i \leq n$$

$$\begin{aligned}1) (\exists M : \mathbb{N} \mid \text{true} : (\forall i \mid 1 \leq i \leq n : (\sum k \mid 1 \leq k \leq n : Q[i, k]) = M \\ \wedge (\sum k \mid 1 \leq k \leq n : Q[k, i]) = M \\ \wedge (\sum k \mid 1 \leq k \leq n : Q[k, k]) = M \\ \wedge (\sum k \mid 1 \leq k \leq n : Q[k, (n+1) - k]) = M \\ \wedge (\forall m : \mathbb{N} \mid 1 \leq m \leq n^2 : \\ (\exists i, j \mid 1 \leq i < n \wedge 1 \leq j < n : m = Q[i, j]))))\end{aligned}$$

$$2) M = \frac{\sum_{i=1}^{n^2} i}{n}$$

$$n * M = (\sum i \mid 1 \leq i \leq n^2 : i)$$

$$M = \frac{(\sum i \mid 1 \leq i \leq n^2 : i)}{n} = \frac{n^2 * (n^2 + 1)}{2 * n} = \frac{n * (n^2 + 1)}{2}$$

---

### 5.2 Mathematische Induktion

$(\mathbb{B} : \text{Boolean})$

Sei  $P : \mathbb{N} \rightarrow \mathbb{B}$

zu zeigen:

$$(\forall n : \mathbb{N} \mid \text{true} : P(n))$$

Beispiel

$P(n) : n^3 + 5 * n$  ist ein Vielfaches von 6

$z$  ist Vielfaches von 6 heisst:

$$(\exists i : \mathbb{Z} \mid \text{true} : i * 6 = z)$$

$$0^3 + 5 * 0 = 0 (\text{Zeuge}) * 6$$

$$1^3 + 5 * 1 = 1 * 6$$

$$2^3 + 5 * 2 = 3 * 6 \text{ (Muss bei allen } true \text{ zurueck geben!!)}$$

Idee: Induktionsprinzip

Man zeigt:

- 1)  $P(0)$
- 2)  $P(n) \Rightarrow P(n+1)$  fuer alle  $n : \mathbb{N}$

$P(0)$  gilt: Wegen 1)

$$(P(0) \wedge (P(0) \Rightarrow P(1)) \Rightarrow P(1)$$

wegen 2) mit  $n = 0$

$$(P(1) \wedge (P(1) \Rightarrow P(2)) \Rightarrow P(2)$$

wegen 2) mit  $n = 1$

Damit gilt  $P(n)$  fuer alle  $n : \mathbb{N}$

Unser Beispiel

- 1) Induktionsanfang (Base case)

zu zeigen:  $P(0)$

$$0^3 + 5 * 0 = 0(\text{Zeuge}) * 6$$

- 2) Induktionsschritt (inductive step)

zu zeigen:  $P(n) \Rightarrow P(n+1)$  fuer alle  $n : \mathbb{N}$

Sei  $n$  eine beliebige natuerliche Zahl.

Annahme: Es gaelte  $P(n) : n^3 + 5 * n$ , dass heisst  $n^3 + 5 * n = 6 * r$ , mit  $r : \mathbb{Z}$ , ist vielfaches von 6.

zu zeigen: (unter dieser Annahme)  $P(n+1) : (n+1)^3 + 5 * (n+1)$  ist vielfaches von 6.

das heisst:  $(n+1)^3 + 5 * (n+1) = 6 * s$ , mit  $s : \mathbb{Z}$

$$(n+1)^3 + 5 * (n+1)$$

<Arith>

$$= (n^3 + 3 * n^2 + 3 * n + 1) + (5 * n + 5)$$

<Arith + Kaninchen>

$$= (n^3 + 5 * n) + (3 * n^2 + 3 * n + 6)$$

<Annahme>

$$= 6 * r + 3 * n^2 + 3 * n + 6$$

<Arith + Kaninchen>

$$= 6 * r + 3 * n * (n+1) + 6$$

< $n*(n+1)$  ist gerade>

$$= 6 * r + 3 * (2 * t) + 6$$

<Arith>

$$= 6 * (r + t + 1) \text{ (Zeuge) } \checkmark$$

Modus ponens

$p$	$q$	$(p \wedge (p \Rightarrow q)) \Rightarrow q$
$f$	$f$	$w$
$f$	$w$	$w$
$w$	$f$	$w$
$w$	$w$	$w$

$$(p \Rightarrow r) \wedge (\neg p \Rightarrow s)$$

$$\equiv (p \wedge r) \vee (\neg p \wedge s)$$

Sei  $p$ . Dann

$$\begin{aligned} & (p \Rightarrow r) \wedge (\neg p \Rightarrow s) \\ = & \quad r \quad \wedge \quad true \\ = & \quad r \end{aligned}$$

$$\begin{aligned} & (p \wedge r) \vee (\neg p \wedge s) \\ = & \quad r \quad \vee \quad false \\ = & \quad r \end{aligned}$$

Sei  $\neg p$  Analog

## 6 Sechste Woche

### 6.1 Vollst ndige Induktion

Arbeitsblatt 1 - Aufgabe 1

$$(P(0) \wedge (\forall n : \mathbb{N} \mid : P(n) \Rightarrow P(n+1))) \Rightarrow (\forall n : \mathbb{N} \mid : P(n))$$

wobei  $P(0)$  : Base Case

$(\forall n : \mathbb{N} \mid : P(n) \Rightarrow P(n+1))$  : Inductive Case

$(\forall n : \mathbb{N} \mid : P(n))$  : Ziel der Induktion

Complete Induction

$$\begin{aligned}
&P(0) \\
&(P(0) \wedge (P(0) \Rightarrow P(1))) \Rightarrow P(1) \\
&(P(0) \wedge P(1) \wedge (P(0) \wedge P(1) \Rightarrow P(2))) \Rightarrow P(2) \\
&(P(0) \wedge P(1) \wedge P(2) \wedge (P(0) \wedge P(1) \wedge P(2) \Rightarrow P(3))) \Rightarrow P(3) \\
&(P(0) \wedge (\forall n : \mathbb{N} \mid : (\forall k : \mathbb{N} \mid k \leq n : P(k)) \Rightarrow P(n+1)) \Rightarrow (\forall n : \mathbb{N} \mid : P(n))
\end{aligned}$$

## Aufgabe 2

Sei  $k : \mathbb{N}$ ,  $k \geq 0$

$$(P(k) \wedge (\forall n : \mathbb{N} \mid n \geq k : P(n) \Rightarrow P(n+1)) \Rightarrow (\forall n : \mathbb{N} \mid n \geq k : P(n))$$

## Fibonacci

$$fib : \mathbb{N} \rightarrow \mathbb{N}$$

- Ⓐ  $fib(0) = 0$
- Ⓑ  $fib(1) = 1$
- Ⓒ  $fib(n) = fib(n-1) + fib(n-2)$ ,  $n \geq 2$

Satz Fuer alle  $n : \mathbb{N}$  gilt:

$$P(n) : (\sum i : \mathbb{N} \mid 1 \leq i \leq n : fib(i)) = f(n+2) - 1$$

## Beweis

1) Induktionsanfang: zu zeigen:  $P(0)$ , also

$$(\sum i : \mathbb{N} \mid 1 \leq i \leq 0 : fib(i)) = f(0+2) - 1$$

$$(\sum i : \mathbb{N} \mid 1 \leq i \leq 0 : fib(i))$$

< empty range, neutral + >

$$= 0$$

$$fib(0+2) - 1$$

< arith >

$$= fib(2) - 1$$

< Ⓒ mit  $n = 2$  >

$$= fib(0) + fib(1) - 1$$

< Ⓐ, Ⓑ >

$$= 0 + 1 - 1$$

< arith >

$$= 0$$

2) Induktionsschritt:

Sei  $n$  eine beliebige natuerliche Zahl

Annahme:  $(\sum i : \mathbb{N} \mid 1 \leq i \leq n : fib(i)) = fib(n+2) - 1$

zu zeigen:  $(\sum i : \mathbb{N} \mid 1 \leq i \leq n+1 : fib(i)) = fib(n+3) - 1$

$$(\sum i : \mathbb{N} \mid 1 \leq i \leq n+1 : fib(i))$$

< range split (split-off term) >

$$= (\sum i : \mathbb{N} \mid 1 \leq i \leq n : fib(i)) + fib(n+1)$$

< Annahme >

$$= fib(n+2) - 1 + fib(n+1)$$

< arith >

$$= (fib(n+2) + fib(n+1)) - 1$$

<  $\textcircled{C}$ , mit  $n+3 \geq 2$  >

$$= fib(n+3) - 1$$

□

---

Satz:

Fuer alle  $n : \mathbb{N}$  mit  $n \geq 3$  gilt:

$$2n+1 < 2^n$$

Beweis

1) IA

$$2 * 3 + 1 < 2^3 \equiv 7 < 8 \checkmark$$

2) IS

Sei  $n$  eine beliebige natuerliche Zahl mit  $n \geq 3$

Annahme Es gelte:  $2n+1 < 2^n$

zu zeigen: Es gilt:  $2(n+1)+1 < 2^{(n+1)}$

Beweis



$$2(n+1)+1$$

< arith >

$$= (2n+1)+2$$

< Annahme >

$$\Rightarrow (2n+1)+2 < 2^n+2$$

$\leq$

$$2^n+2^n$$

< arith >

$$= 2^n+1 \checkmark$$

$$2 * 2 * \dots * 2 = 2^n$$

(n mal)

$$pow1(n) = (\prod i : \mathbb{N} \mid 1 \leq i \leq n : 2)$$

$$pow2(n) \begin{cases} 2^0 = 1 \\ 2^n = 2 * 2^{n-1}, \text{ fuer } n > 0 \end{cases}$$

```

int pow1(int n){
    int p=1;
    for(int i=1; i<=n; i++){
        p*=2;
    }
    return p;
}

int pow2(int n){
    return (n==0)? 1: 2*pow2(n-1);
}

```

Satz: Fuer alle  $n : \mathbb{N}$  gilt:  $pow1(n) = pow2(n)$

Beweis Aufgabe!

$$x < y \Rightarrow x \leq y$$

$$x \leq y \Rightarrow x < y \vee x = y$$

$$p \Rightarrow p \vee q$$

$$x \leq y \nRightarrow x < y$$

$$15 \leq 15 \nRightarrow 15 < 15$$

---

## 7 Siebte Woche

### 7.1 Zahlentheorie

#### Teilbarkeit

$$\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}, c, b : \mathbb{Z}$$

$$c \setminus b \equiv (\exists k : \mathbb{Z} \mid : b = k * c)$$

$$c \setminus b : \text{"}c \text{ teilt } b\text{"}$$

$$\text{"}b \text{ ist teilbar durch } c\text{"}$$

$$\text{"}c \text{ ist Teiler von } b\text{"}$$

$$\text{"}b \text{ ist Vielfaches von } c\text{"}$$

#### Beispiel:

$$7 \setminus 13 = false$$

$$(-7) \setminus 14 = true$$

$$\text{nicht einheitlich in Literatur } \begin{cases} 0 \setminus 14 = false \text{ (es existiert kein } k) \\ 0 \setminus 0 = true \end{cases}$$

#### Satz $b, c, d : \mathbb{Z}$

$$(1) c \setminus c \text{ (Reflexivitaet)}$$

$$(2) c \setminus 0$$

$$(3) 1 \setminus b$$

$$(4) c \setminus 1 \Rightarrow c = 1 \vee c = -1$$

$$(5) d \setminus c \wedge c \setminus b \Rightarrow d \setminus b \text{ (Transitivitaet)}$$

$$(6) b \setminus c \wedge c \setminus b \Rightarrow b = c \vee b = -c$$

auf  $\mathbb{Z}$  also nicht antisymmetrisiert, auf  $\mathbb{N}$  aber schon

$$(7) b \setminus c \Rightarrow b \setminus (c * d)$$

$$(8) b \setminus c \Rightarrow (b * d) \setminus (c * d)$$

$$(9) 1 < b \wedge b \setminus c \Rightarrow \neg(b \setminus (c + 1))$$

(1) und (5) und (6) heisst:  $\setminus$ -Relation ist eine partielle Ordnung auf  $\mathbb{N}$  (aber nicht auf  $\mathbb{Z}$ )

#### Beweis:

$$(1) \text{ zu zeigen: es gibt } k : \mathbb{Z} \text{ mit } c = k * c$$

$$\text{Wähle } k = 1 : \mathbb{Z}$$

$$(2) 0 = k * c, k = 0 : \mathbb{Z}$$

$$(3) b = k * 1, k = b : \mathbb{Z}$$

$$(4) 1 = k * c = 1 * 1 = (-1) * (-1) \Rightarrow c = 1 \vee c = -1$$

$$(5)$$

$$d \setminus c, \text{ also gibt es } k_1 : \mathbb{Z} \text{ mit } c = k_1 * d$$

$$c \setminus b, \text{ also gibt es } k_2 : \mathbb{Z} \text{ mit } b = k_2 * c$$

$$\text{aber } b = k_2 * c = k_2 * (k_1 * d) = (k_2 * k_1) * d$$

also gibt es  $k = k_1 * k_2 : \mathbb{Z}$  mit  $b = k * d$ , also  $d \setminus b$   
 (6)  
 $b \setminus c$ , also gibt es  $k_1 : \mathbb{Z}$  mit  $c = k_1 * b$   
 $c \setminus b$ , also gibt es  $k_2 : \mathbb{Z}$  mit  $b = k_2 * c$   
 also  $b = k_2 * c = k_2 * (k_1 * b) = (k_2 * k_1) * b$   
 also  $b - (k_2 * k_1) * b = 0$   
 also  $b * (1 - k_2 * k_1) = 0$   
 also  $b = 0 \vee 1 - k_2 * k_1 = 0$   
 also  $b = 0 \vee k_2 * k_1 = 1$   
 also  $b = 0 \vee k_2 = k_1 = 1 \vee k_2 = k_1 = -1$   
 also  $b = 0 \vee c = b \vee c = -b$   
 mit  $b = 0$  ist  $c = b$

Satz Seien  $a, b, c : \mathbb{Z}$

Dann  $a \setminus b \wedge a \setminus c \Rightarrow a \setminus (b + c)$

$a \setminus b \Rightarrow b = k_1 * a$

$a \setminus c \Rightarrow c = k_2 * a$

$a \setminus (b + c) \Rightarrow a \setminus (k_1 * a + k_2 * a) \Rightarrow a \setminus a * (k_1 + k_2) \checkmark$  (Satz (7))

Beweis

Annahme 1:  $a \setminus b$ , also nach Definition existiert  $k_1 : \mathbb{Z}$  mit  $b = k_1 * a$

Annahme 2:  $a \setminus c$ , also nach Definition existiert  $k_2 : \mathbb{Z}$  mit  $c = k_2 * a$

zu zeigen: Es existiert  $k_3 : \mathbb{Z}$  mit  $b + c = k_3 * a$

$b + c$

< Annahme 1 und 2 >

$= k_1 * a + k_2 * a$

< arith >

$= (k_1 + k_2) * a$

< mit  $k_3 = k_1 + k_2 : \mathbb{Z}$  >

$= k_3 * a$

□

## 8 Achte Woche

### 8.1 Euklidische Division

Satz (Euklidische Division auf  $\mathbb{N}$ )

Seien  $a, b : \mathbb{N}$ ,  $b \neq 0$ .

Dann gibt es eindeutige  $q, r : \mathbb{N}$  mit

$$a = b * q + r \wedge 0 \leq r < b$$

□

Satz (Euklidische Division auf  $\mathbb{Z}$ )

Seien  $a, b : \mathbb{Z}$ ,  $b \neq 0$ .

Dann gibt es eindeutige  $q : \mathbb{Z}$ ,  $r : \mathbb{N}$  mit

$$a = b * q + r \wedge 0 \leq r < |b|$$

□

---

$$a, b : \mathbb{N}, b \neq 0$$

$$a = 17, b = 5$$

$$17 = 5 * 0 + 17 \wedge 0 \leq 17$$

$$17 = 5 * 1 + 12 \wedge 0 \leq 12$$

$$17 = 5 * 2 + 7 \wedge 0 \leq 7$$

$$17 = 5 * 3 + 2 \wedge 0 \leq 2$$

$$a = b * q + r \wedge 0 \leq r \text{ (Invariante)}$$

Stop bei  $r < b$

$q, r$  seien Variablen einer imperativer Programmiersprache (keine Mathematische Variablen)

---

$$\{a \geq 0 \wedge b > 0\}$$

VC3  $\Rightarrow$

$$\{a = b * 0 + a \wedge 0 \leq a\}$$

$$q, r := 0, a;$$

$$\{a = b * q + r \wedge 0 \leq r\}$$

while  $r \geq b$  do

$$\{a = b * q + r \wedge 0 \leq r \wedge r \geq b\}$$

VC1  $\Rightarrow$

$$\{a = b * (q + 1) + (r - b) \wedge 0 \leq r - b\} \text{ (Precondition)}$$

$$q, r := q + 1, r - b$$

$$\{a = b * q + r \wedge 0 \leq r\}$$

endwhile

$$\{a = b * q + r \wedge 0 \leq r \wedge \neg r \geq b\}$$

VC2  $\Rightarrow$

$$\{a = b * q + r \wedge 0 \leq r < b\}$$

VC: Verification Condition

$$\text{VC3: } a = b * 0 + a \text{ (true)}$$

$$a \geq 0 \wedge b > 0 \Rightarrow \text{true} \wedge 0 \leq a$$

$$\text{VC2: } \neg r \geq b \equiv r < b$$

$$\text{VC1: } a = b(q + 1) + (r - b)$$

$\Leftarrow$

$$a = b * q + b + r - b$$

$\Leftarrow$

$$a = b * q + r$$

(von unten nach oben)

Warum gilt  $0 \leq r - b$ ?

Weil  $r \geq b$ !

$$r \geq b \Rightarrow r - b \geq 0 \Rightarrow 0 \leq r - b$$

---


$$\{x + 1 > 5\} \text{ (Vorbedingung)}$$

$$x := x + 1$$

$$\{x > 5\} \text{ (Nachbedingung)}$$

---

## 8.2 Fundamentaler Schleifen-Satz

Satz (Fundamentaler Schleifen-Satz)

Seien  $B$  und  $I$  boolesche Ausdrücke und  $C$  ein Kommando.

Es gelte  $\{I \wedge B\} C \{I\}$ , d.h.  $I$  ist eine Invariante der Schleife while  $B$  do  $C$  end.

Die Ausführung der Schleife beginne im Zustand der  $I$  erfüllt. Dann gilt  $I$  nach jedem Schleifendurchlauf.

□

Beweis

Durch Induktion. Wir zeigen, dass "Invariante ist nach  $n$  Schleifendurchläufe erfüllt" fuer alle  $n \in M$ . Dabei ist  $M = \mathbb{N}$ , falls  $\infty$ -Schleife, und  $M = \{0 \dots N\}$  mit  $N : \mathbb{N}$  Anzahl der Schleifendurchläufe (SDL).

IA  $I$  ist nach 0 SDL erfüllt, nach Voraussetzung

Sei  $M = \mathbb{N}$  (Fall 1: Sei  $M = \{0 \dots N\}$ )

IS Sei  $n$  eine beliebige natuerliche Zahl. (Fall 2:  $n \leq N - 1$ )

Annahme: Es gelte:  $I$  ist nach  $n$  SDL erfüllt.

zu zeigen: Es gilt:  $I$  ist nach  $n + 1$  SDL erfüllt.

Beweis

$I$  ist nach  $n$  SDL erfüllt. Weiterer SDL bedeutet, dass Schleifenbedingung  $B$  gilt.  $I \wedge B$  gilt also, also nach Ausführung von  $C$  wieder  $I$ , wegen  $\{I \wedge B\} C \{I\}$ , also  $I$  nach  $n + 1$  SDL erfüllt.

□

Satz Falls Ausführung terminiert, gilt am Ende  $I \wedge \neg B$

Beweis

Nach obigem Satz, gilt  $I$  am Ende. Die Schleife terminiert, gilt auch  $\neg B$ .

---

$\{I \wedge B\} C \{I\}$

$\{I\} \text{ while } \{B\} \text{ do } C \text{ end } \{I \wedge \neg B\}$

---

## 9 Neunte Woche

### 9.1 Euklid auf $\mathbb{Z}$ , $\mathbb{N}$

Satz (Euklid auf  $\mathbb{Z}$ )

$a, b : \mathbb{Z}, b \neq 0$ . Dann gibt es eindeutige  $q : \mathbb{Z}, r : \mathbb{N}$  mit

$$a = b * q + r \wedge 0 \leq r < |b|$$

□

---

Truncated Division

$$a = b * q + r \wedge \begin{cases} 0 \leq r < |b|, & \text{if } a \geq 0 \\ -|b| < r \leq 0, & \text{if } a \leq 0 \end{cases}$$

$r : \mathbb{Z}$

---

Satz (Euklid auf  $\mathbb{N}$ )

$a, b : \mathbb{N}, b \neq 0$ . Dann gibt es eindeutige  $q : \mathbb{N}, r : \mathbb{N}$  mit

$$a = b * q + r \wedge 0 \leq r < b$$

□

Beweis fuer Euklid auf  $\mathbb{Z}$

4 Faelle

1. Fall  $a \geq 0, b > 0$

Trivial

2. Fall  $a \geq 0, b < 0$

Euklid  $\mathbb{N}$  anwenden auf  $(a, -b)$ . Liefert

$$a = (-b) * q + r \wedge 0 \leq r < -b$$

Waehle  $(q', r') = (-q, r) - b = |b|$

$$a = b * (-q) + r \wedge 0 \leq r < |b|$$

$$a = b * q' + r' \wedge 0 \leq r' < |b|$$

3. Fall  $a < 0 \wedge b > 0$

Euklid  $\mathbb{N}$  anwenden auf  $(-a, b)$  Liefert

$$-a = b * q + r \wedge 0 \leq r < b$$

Sei  $r = 0$  Wähle  $(q', r') = (-q, 0)$   $b = |b|$

$$a = b * (-q) - r \wedge 0 \leq r < |b|$$

$$a = b * q' + r \wedge 0 \leq r < |b|$$

Sei  $1 \leq r \leq b - 1$  Wähle  $(q', r') = (-(q + 1), b - r)$

$$-a = b * q + r$$

$$a = b * (-q) - r$$

$$= b * (-q) - b + b - r$$

$$= b * (-q - 1) + (b - r)$$

$$= b * (-(q + 1)) + (b - r)$$

$$= b * q' + r'$$

$$1 \leq \underbrace{b - r}_{r'} \leq b - 1 \begin{cases} 1 \leq r \rightarrow -r \leq -1 \rightarrow b - r \\ r \leq b - 1 \rightarrow 1 \leq b - r \end{cases}$$

4. Fall  $a < 0, b < 0$  (??)

Ähnlich