

# Grundbegriffe der Informatik

## Tutorium 33

Lukas Bach, [lukas.bach@student.kit.edu](mailto:lukas.bach@student.kit.edu) | 17.11.2016



Lukas Bach, lu-  
kas.bach@student.kit.edu

Vollständige  
Induktion

## 1 Vollständige Induktion

Formale Sprache

## 2 Formale Sprache

Übersetzung und  
Kodierung

## 3 Übersetzung und Kodierung

- Kodierung von Zahlen
- Repräsentation von Zahlen
- Zweierkomplement-Darstellung

Kodierung von  
Zahlen

Repräsentation von  
Zahlen

Zweierkomplement-  
Darstellung

Lukas Bach, lu-  
kas.bach@student.kit.edu

Vollständige  
Induktion

Formale Sprache

Übersetzung und  
Kodierung

Kodierung von  
Zahlen

Repräsentation von  
Zahlen

Zweierkomplement-  
Darstellung

- Was macht die Funktion  $val_l$ ?
- Was bedeutet Äquivalenz?
- Was bedeutet Tautologie und Erfüllbarkeit?
- Welche dieser Aussagen sind erfüllbar?
  - $\neg(P \wedge Q) \leftrightarrow \neg P \vee \neg Q$
  - $P \wedge P \leftrightarrow P \vee P$

# Wahrheitsgehalt von unendlich Aussagen

Beispielsituation: Wir haben unendlich viele Dominosteine. Behauptung:  
Alle Dominosteine fallen um.

- Wir haben Aussagen: {"1. Stein fällt um", "2. Stein fällt um", ...}
- Wie zeigen wir unendlich viele Aussagen?
- Stelle Aussagen in Abhängigkeit einer Laufvariable  $n$  dar:
  - $A(n) := \text{"n-ter Stein fällt um"} \forall n \in \mathbb{N}$ .
- Aussage  $A := \text{"Alle Steine fallen um"} \equiv A(i)$  ist wahr  $\forall i \in \mathbb{N}$ .

Wir haben immernoch unendlich viele Aussagen...

- Zeige:  $A(1)$  ist wahr, sowie  $A(i)$  gilt  $\rightarrow A(i+1)$  gilt für beliebiges  $i \in \mathbb{N}$ .
- Also: Der erste Stein fällt, sowie: falls der  $i$ -te Stein fällt, so fällt auch der  $i+1$ -te Stein.
- Nach dem Prinzip der vollständigen Induktion fallen dann alle Steine um.

Vollständige  
Induktion

Formale Sprache

Übersetzung und  
Kodierung

Kodierung von  
Zahlen

Repräsentation von  
Zahlen

Zweierkomplement-  
Darstellung

## Vollständige Induktion

Formale Sprache

Übersetzung und  
Kodierung

Kodierung von  
Zahlen

Repräsentation von  
Zahlen

Zweierkomplement-  
Darstellung

- Beweisverfahren
- In der Regel zu zeigen: Eine Aussage gilt für alle  $n \in \mathbb{N}_+$ , manchmal auch für alle  $n \in \mathbb{N}_0$
- Man schließt “induktiv” von einem  $n$  auf  $n+1$
- Idee: Wenn die Behauptung für ein beliebiges **festes**  $n$  gilt, dann gilt sie auch für den Nachfolger  $n+1$  (und somit auch für dessen Nachfolger und schließlich für alle  $n$ )

# Struktur des Beweises

Behauptung: (*kurz Beh.:*)

Beweis: (*kurz Bew.:*)

- Induktionsanfang: (*kurz IA:*)

- Zeigen, dass Behauptung für Anfangswert gilt (oft  $n = 1$ )
- Auch mehrere (z.B. zwei) Anfangswerte möglich

- Induktionsvoraussetzung: (*kurz IV:*)

- Sei  $n \in \mathbb{N}_+$  (bzw.  $n \in \mathbb{N}_0$ ) fest aber beliebig und es gelte [Behauptung einsetzen]

- Induktionsschritt: (*kurz IS:*)

- Behauptung für  $n+1$  auf  $n$  zurückführen
- Wenn induktive Definition gegeben: verwenden!
- Sonst: Versuche Ausdruck, in dem  $(n+1)$  vorkommt umzuformen in einen Ausdruck, in dem nur  $n$  vorkommt

Vorhin:

$$\underbrace{A(1) \text{ ist wahr}}_{IA}, \text{ sowie } \underbrace{A(i) \text{ gilt}}_{IV} \rightarrow \underbrace{A(i+1) \text{ gilt}}_{IS} \text{ f\"ur beliebiges } i \in \mathbb{N}$$

## Aufgabe

$$x_0 := 0$$

$$\text{Für alle } n \in \mathbb{N}_0 : x_{n+1} := x_n + 2n + 1$$

*Zeige mithilfe vollständiger Induktion, dass für alle  $n \in \mathbb{N}_0$*

$$x_n = n^2$$

**gilt.**

## Übungsaufgaben

Zeige die Wahrheit folgender Aussagen mit vollständiger Induktion:

■  $\sum_{i=1}^n i = \frac{n(n+1)}{2} \quad \forall n \in \mathbb{N}.$

■  $\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6} \quad \forall n \in \mathbb{N}$



- Was war nochmal  $A^*$ ? Menge aller Wörter *beliebiger* Länge über Alphabet  $A$ .
- Was war nochmal eine formale Sprache?

## Formale Sprache

Eine Formale Sprache  $L$  über einem Alphabet  $A$  ist eine Teilmenge  $L \subseteq A^*$ .

Als Beispiel von vorigen Folien:

- $A := \{b, n, a\}$ .
  - $L_1 := \{ban, baan, nba, aa\}$  ist eine mögliche formale Sprache über  $A$ .
  - $L_2 := \{banana, bananana, banananana, \dots\}$   
 $= \{w : w = bana(na)^k, k \in \mathbb{N}\}$  auch.
  - $L_3 := \{ban, baan, baaan, \dots\}$  auch. Andere Schreibweise?  
 $L_3 = \{w : w = ba^k n, k \in \mathbb{N}\}$

## Produkt von formalen Sprachen

Von zwei formalen Sprachen  $L_1, L_2$  lässt sich das Produkt  $L_1 \cdot L_2$  bilden mit  $L_1 \cdot L_2 = \{w_1 w_2 : w_1 \in L_1 \text{ und } w_2 \in L_2\}$ .

Sei  $A := \{a, b\}, B := \{\alpha, \beta, \gamma, \varepsilon, \delta\}$ .

- Sprache  $L_1 \subseteq A^*$ , die zuerst drei  $a$ 's enthält und dann entweder zwei  $b$ 's oder vier  $a$ 's?  $L_1 = \{aaa\} \cdot \{bb, aaaa\}$ .
- Sprache  $L_2 \subseteq A^*$ , die alle Wörter über  $A$  enthält außer  $\varepsilon$ ?  
 $L_2 = A \cdot A^* = A^* \setminus \{\varepsilon\}$ .
- Sprache  $L_3 \subseteq B^*$ , die alle Wörter über  $B$  enthält, mit:
  - Zwei beliebigen Zweichen aus  $B$ .
  - Dann einem  $\varepsilon$  oder zwei  $\delta$ 's.
  - Dann vier Zeichen aus  $A$ .
- $L_3 = B \cdot B \cdot \{\varepsilon, \delta\delta\} \cdot A \cdot A \cdot A \cdot A$ .

Vollständige  
Induktion

Formale Sprache

Übersetzung und  
Kodierung

Kodierung von  
Zahlen

Repräsentation von  
Zahlen

Zweierkomplement-  
Darstellung

Vollständige  
Induktion

Formale Sprache

Übersetzung und  
Kodierung

Kodierung von  
Zahlen

Repräsentation von  
Zahlen

Zweierkomplement-  
Darstellung

## Übung zu Produkt von formalen Sprachen

Sei  $A$  ein beliebiges Alphabet und  $M := \{L : L \text{ ist formale Sprache über } A\} = 2^A$ . Produkt von Sprachen lässt sich auch als Abbildung bzw. Verknüpfung  $\cdot : M \times M \rightarrow M$  darstellen.

Zeige:

- Die Verknüpfung  $\cdot$  ist assoziativ.
- Es gibt (mindestens) ein Element  $e \in M$ , sodass für alle  $x \in M$  gilt:  
 $x \cdot e = e \cdot x = x$ . (Neutrales Element)
- Es gibt ein Element  $o \in M$ , sodass für alle  $x \in M$  gilt:  
 $x \cdot o = o = o \cdot x$ . (Absorbierendes Element)

Seien  $L_1, L_2, L_3 \in M$ .

- Die Verknüpfung  $\cdot$  ist assoziativ:

- $(L_1 \cdot L_2) \cdot L_3 = (\{w_1 \cdot w_2 : w_1 \in L_1, w_2 \in L_2\}) \cdot L_3 = \{w_1 w_2 w_3 : w_1 \in L_1, w_2 \in L_2, w_3 \in L_3\} = L_1 \cdot (\{w_2 w_3 : w_2 \in L_2, w_3 \in L_3\}) = L_1 \cdot (L_2 \cdot L_3).$

- Es gibt (mindestens) ein Element  $e \in M$ , sodass für alle  $x \in M$  gilt:  
 $x \cdot e = e \cdot x = x$ . (neutrales Element)

- $e := \{\varepsilon\}.$
  - $L_1 \cdot \{\varepsilon\} = L_1 = \{\varepsilon\} \cdot L_1$

- Es gibt ein Element  $o \in M$ , sodass für alle  $x \in M$  gilt:  
 $x \cdot o = o = o \cdot x$ . (Absorbierendes Element)

- $o := \emptyset$
  - $L_1 \cdot \emptyset = \emptyset = \emptyset \cdot L_1$

$(M, \cdot)$  ist damit trotzdem keine Gruppe, denn es existieren keine Invers-Element.

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$  für  $i \in \mathbb{N}_0$ .
- $L_1 := \{a\}$ .
  - $L_1^0 = \{\varepsilon\}$ .  $L_1^1 = \{\varepsilon\} \cdot L_1 = L_1$ .
  - $L_1^2 = (\{\varepsilon\} \cdot L_1) \cdot L_1 = \{aa\}$ .
- $L_2 := \{ab\}^3 \{c\}^4$ 
  - $L_2^0 = \{\varepsilon\}$ ,  $L_2^1 = \dots$
  - $L_2^2 = (\{ab\}^3 \{c\}^4)^2 = (\{ab\}^3 \{cccc\})^2 = \{abababcccc\}^2 = \{abababccccabababcccc\}$ .
- $L_3 := (\{a\} \cup \{b\})^2 = \{aa, ab, ba, bb\}$

Vollständige  
Induktion

Formale Sprache

Übersetzung und  
Kodierung

Kodierung von  
Zahlen

Repräsentation von  
Zahlen

Zweierkomplement-  
Darstellung

## Konkatenationsabschluss

Zu einer formalen Sprache  $L$  ist der Konkatenationsabschluss  $L^*$  definiert als  $L^* := \bigcup_{i \in \mathbb{N}_0} L^i$ .

## $\varepsilon$ -freie Konkatenationsabschluss

Zu einer formalen Sprache  $L$  ist der  $\varepsilon$ -freie Konkatenationsabschluss  $L^+$  definiert als  $L^+ := \bigcup_{i \in \mathbb{N}_+} L^i$ .

- Warum gilt  $\varepsilon \notin L^+$  bei formalen Sprache  $L \subseteq A^* \setminus \{\varepsilon\}$ ?
- $L := \{a, b, c\}.L^* = \{\varepsilon, a, aa, ab, ac, aaa, aab, \dots, b, ba, bb, \dots\}$
- $L := \{aa, bc\}.L^* = \{\varepsilon, aa, bc, aa \cdot aa, aa \cdot bc, bc \cdot aa, bc \cdot bc, aa \cdot aa \cdot aa, \dots\}$

Sei  $A := \{a, b\}$ ,  $B := \{A, B, C, D, E, F\}$ .

- Sprache  $L_1 \subseteq A^*$ , die das Teilwort  $ab$  nicht enthält?  $L_1 = \{b\}^* \{a\}^*$ .
- Sprache  $L_2 \subseteq B^*$ , die alle erlaubten Java Variablennamen enthält.
  - $B := \{\_, a, b, \dots, z, A, B, \dots, Z\}$
  - $C := B \cup \mathbb{Z}_9$
  - $L_2 \subseteq C = (B \cdot C^*) \setminus \{if, class, while, \dots\}$

Sei  $L := \{a\}^* \{b\}^*$ .

- Was ist alles in  $L$  drin?
  - $aaabbabbbaaabbba$ ? Nein.
  - $aaabb, abbaaabbba$ ? Ja, nein.
  - $aaabb, abb, aaabba$ ? Ja, ja, nein.
  - $aaabb, abb, aaabb, a$ ? Alles drin.
- Was ist alles in  $L^*$  drin?
  - $aaabbabbbaaabbba$ ? Ja.
  - $aaabb, abbaaabbba$ ? Ja.
  - $aaabb, abb, aaabba$ ? Ja.
  - $aaabb, abb, aaabb, a$ ? Ja.
  - Alle Wörter aus  $\{a, b\}^* \rightarrow L^* = \{a, b\}^*$ .



## Erinnerung

$$L^* := \bigcup_{i \in \mathbb{N}_0} L^i \quad L^+ := \bigcup_{i \in \mathbb{N}_+} L^i$$

## Beweisaufgabe

Beweise:  $L^* \cdot L = L^+$ .

$\subseteq$ :

**Voraussetzung:**  $w \in L^* \cdot L$  mit  
 $w = w'w''$ ,  $w' \in L^*$  und  
 $w'' \in L$ .

Dann existiert ein  $i \in \mathbb{N}_0$  mit  
 $w' \in L^i$ , also  
 $w = w'w'' \in L^i \cdot L = L^{i+1}$ .

Weil  $i+1 \in \mathbb{N}_+$ , gilt:  
 $L^{i+1} \subseteq L^+$ , also  $w \in L^+$ .

$\supseteq$ :

**Voraussetzung:**  $w \in L^* \cdot L$ .

Dann existiert ein  $i \in \mathbb{N}_+$  mit  $w \in L^i$ . Da  
 $i \in \mathbb{N}_+$ , existiert ein  $j \in \mathbb{N}_0$  mit  $i = j+1$ , also  
für ein solches  $j \in \mathbb{N}_0$ :  $w \in L^{j+1} = L^j \cdot L$ .

Also  $w = w'w''$  mit  $w' \in L^j$  und  $w'' \in L$ .

Wegen  $L^j \subseteq L^*$  ist  $w = w'w'' \in L^* \cdot L$ .

Vollständige  
Induktion

Formale Sprache

Übersetzung und  
Kodierung

Kodierung von  
Zahlen

Repräsentation von  
Zahlen

Zweierkomplement-  
Darstellung

$L_1, L_2$  seien formale Sprachen.

- Wie sieht  $L_1 \cdot L_2$  aus?
- Wie sieht  $L_1^3$  aus?
- Wie sieht  $L_1^2 \cdot L_2 \cdot L_2^0 \cdot L_1^*$  aus?
- Wie sieht  $(L_1^*)^0 \cdot L_2^+$  aus?

Wir betrachten die Alphabete  $A_{dez} := \mathbb{Z}_{10}$ ,  $A_{bin} := \{0, 1\}$ ,  $A_{oct} := \mathbb{Z}_8$ .

- Was können wir daraus machen?
- $A_{dez}^* \supset \{42, 1337, 999\}$ .
- $A_{bin}^* \supset \{101010, 10100111001, 1111100111\}$ .
- $A_{oct}^* \supset \{52, 2471, 1747\}$ .
- Wir suchen eine Möglichkeit, diese **Zahlen** zu **deuten**.
- Aber irgendwie so, dass  $42_{\in A_{dez}} \stackrel{\text{Deutung}}{=} 101010_{\in A_{bin}} \stackrel{\text{Deutung}}{=} 52_{\in A_{oct}}$ .

## $Num_k$

Einer Zeichenkette  $Z_k$  aus Ziffern wird mit  $Num_k$  eine eindeutige Zahl zugeordnet:

$$Num_k(\varepsilon) = 0$$

$$Num_k(wx) = k \cdot Num_k(w) + num_k(x) \text{ mit } w \in Z_k^* \text{ und } x \in Z_k.$$

## $num_k$

Einer einzelnen Ziffer  $x \in Z_k$  aus einem Alphabet von Ziffern  $Z_k$  wird mit  $num_k(x)$  der Wert der Zahl zugewiesen.

- Wichtig:  $Num_k(w) \neq num_k(w)$ !
- Was ist:  $num_{10}(3) = 3$ ,  $num_{10}(7) = 7$ ,  $num_{10}(11) =$  nicht definiert.
- Für Zahlen  $\geq k$ : Benutze  $Num_k$ !

## Beispiel zu Zahlendarstellungen

$$Num_k(\varepsilon) = 0.$$

$$Num_k(wx) = k \cdot Num_k(w) + num_k(x) \text{ mit } w \in Z_k^* \text{ und } x \in Z_k.$$

Was ist  $Num_{10}(123)$ ?

$$\begin{aligned} \blacksquare \quad Num_{10}(123) &= 10 \cdot Num_{10}(12) + num_{10}(3) = \\ &= 10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) = \\ &= 10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot (1 + 10 \cdot 2) + 3 = 123. \end{aligned}$$

Yay?

Was ist der dezimale Zahlenwert der Binärzahl 1010? Diesmal Basis  $k = 2$ .

$$\begin{aligned} \blacksquare \quad Num_2(1010) &= 2 \cdot Num_2(101) + num_2(0) = \\ &= 2 \cdot (2 \cdot Num_2(10) + num_2(1) + num_2(0)) = \\ &= 2 \cdot (2 \cdot (2 \cdot Num_2(1) + num_2(0)) + num_2(1)) + num_2(0) = \\ &= 2 \cdot (2 \cdot (2 \cdot num_2(1) + num_2(0)) + num_2(1)) + num_2(0) = \\ &= 2 \cdot (2 \cdot (2 \cdot 1 + 0) + 1) + 0 = 10. \end{aligned}$$

Yay!

Vollständige  
Induktion

Formale Sprache

Übersetzung und  
Kodierung

Kodierung von  
Zahlen

Repräsentation von  
Zahlen

Zweierkomplement-  
Darstellung

$$Num_k(\varepsilon) = 0.$$

$$Num_k(wx) = k \cdot Num_k(w) + num_k(x) \text{ mit } w \in Z_k^* \text{ und } x \in Z_k.$$

## Übungen zu Zahlendarstellungen

Berechne den numerischen Wert der folgenden Zahlen anderer Zahlensysteme nach dem vorgestellten Schema:

- $Num_8(345)$ .
- $Num_2(11001)$ .
- $Num_2(1000)$ .
- $Num_4(123)$ .
- $Num_{16}(4DF)$ . (Zusatz)

Anmerkung: Hexadezimalzahlen sind zur Basis 16 und verwenden als Ziffern (in aufsteigender Reihenfolge: 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

## Lösungen:

- $Num_8(345) = 229.$
- $Num_2(11001) = 25.$
- $Num_2(1000) = 8.$
- $Num_4(123) = 27.$
- $Num_{16}(4DF) = 1247.$

Es gilt:

$$2(2(2(2(2 \cdot 1 + 0) + 1) + 0) + 1) + 0 = 2^4 \cdot 1 + 2^4 \cdot 0 + 2^3 \cdot 1 + 2^2 \cdot 0 + 2^1 \cdot 1 + 2^0 \cdot 0.$$

Daher, einfachere Rechenweise:

$$Num_k(w) = k^0 \cdot w(0) + k^1 \cdot w(1) + k^2 \cdot w(2) + \dots$$

Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

- $Num_2(10101) = 21.$
- $Num_2(11101) = 29.$
- $Num_2(1111111111) = 1023.$



$$Num_k(w) = k^0 \cdot w(0) + k^1 \cdot w(1) + k^2 \cdot w(2) + \dots$$

Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

- $Num_{16}(A1) = 161.$
- $Num_{16}(BC) = 188.$
- $Num_{16}(14) = 20.$

## *div* Funktion

Die Funktion *div* **dividiert ganzzahlig**. (Schneidet also den Rest ab).

## *mod* Funktion

Die Modulo Funktion *mod* gibt den **Rest einer ganzzahligen Division** zurück.

■  $22 \text{ div } 8 = 2 \left( \frac{22}{8} = 2,75 \right).$

■  $22 \text{ mod } 8 = 6.$

Fülle die Tabelle aus:

x	0	1	2	3	4	5	6	7	8	9	10	11	12
$x \text{ div } 4$	0	0	0	0	1	1	1	1	2	2	2	2	3
$x \text{ mod } 4$	0	1	2	3	0	1	2	3	0	1	2	3	0

$11101_2$  ist also  $29_{10}$ . Was ist  $29_{10}$  in binär?

## $k$ -äre Darstellung

Die Repräsentation einer Zahl  $n$  zur Basis  $k$  lässt sich wie folgt ermitteln:

$$\mathbf{Repr}_k(n) = \begin{cases} \mathbf{repr}_k(n) & \text{falls } n < k \\ \mathbf{Repr}_k(n \text{ div } k) \cdot \mathbf{repr}_k(n \bmod k) & \text{falls } n \geq k \end{cases}$$

Achtung! Das  $\cdot$  Symbol steht für Konkatenation, nicht für Multiplikation!

$$\text{Repr}_k(n) = \begin{cases} \text{repr}_k(n) & \text{falls } n < k \\ \text{Repr}_k(n \text{ div } k) \cdot \text{repr}_k(n \bmod k) & \text{falls } n \geq k \end{cases}$$

Zum Beispiel:

$$\begin{aligned} \text{Repr}_2(29) &= \text{Repr}_2(29 \text{ div } 2) \cdot \text{repr}_2(29 \bmod 2) \\ &= \text{Repr}_2(14) \cdot \text{repr}_2(1) \\ &= \text{Repr}_2(14 \text{ div } 2) \cdot \text{repr}_2(14 \bmod 2) \cdot 1 \\ &= \text{Repr}_2(7) \cdot \text{repr}_2(0) \cdot 1 \\ &= \text{Repr}_2(7 \text{ div } 2) \cdot \text{repr}_2(7 \bmod 2) \cdot 01 \\ &= \text{Repr}_2(3) \cdot \text{repr}_2(1) \cdot 01 \\ &= \text{Repr}_2(3 \text{ div } 2) \cdot \text{repr}_2(3 \bmod 2) \cdot 101 \\ &= \text{Repr}_2(1) \cdot \text{repr}_2(1) \cdot 101 \\ &= 11101 \end{aligned}$$

$$\mathbf{Repr}_k(n) = \begin{cases} \mathbf{repr}_k(n) & \text{falls } n < k \\ \mathbf{Repr}_k(n \text{ div } k) \cdot \mathbf{repr}_k(n \bmod k) & \text{falls } n \geq k \end{cases}$$

Beispiel mit Hexadezimalzahlen:

$$\begin{aligned} \mathbf{Repr}_{16}(29) &= \mathbf{Repr}_{16}(29 \text{ div } 16) \cdot \mathbf{repr}_{16}(29 \bmod 16) \\ &= \mathbf{Repr}_{16}(1) \cdot \mathbf{repr}_{16}(13) \\ &= 1 \cdot D = 1D \end{aligned}$$

$$\text{Repr}_k(n) = \begin{cases} \text{repr}_k(n) & \text{falls } n < k \\ \text{Repr}_k(n \text{ div } k) \cdot \text{repr}_k(n \bmod k) & \text{falls } n \geq k \end{cases}$$

## Übung zu $\text{Repr}_k$

Berechne die Repräsentationen folgender Zahlen in gegebenen Zahlensystemen:

- $\text{Repr}_2(13)$ .
- $\text{Repr}_4(15)$ .
- $\text{Repr}_{16}(268)$ .

Lösungen:

- $\text{Repr}_2(13) = 1101$ .
- $\text{Repr}_4(15) = 33$ .
- $\text{Repr}_{16}(268) = 10C$ .

## $\text{bin}_\ell$

Die Funktion  $\text{bin}_\ell: \mathbb{Z}_{2^\ell} \rightarrow \{0, 1\}^\ell$  bringt eine gegebene Binärzahl auf eine feste Länge, indem sie mit Nullen vorne aufgefüllt wird. Formell wird sie definiert als:

$$\text{bin}_\ell(n) = 0^{\ell - |\text{Repr}_2(n)|} \text{Repr}_2(n)$$

Beispiel:

- $\text{bin}_8(3) = 0^{8 - |\text{Repr}_2(3)|} \text{Repr}_2(3) = 0^{8 - |11|} \cdot 11 = 0^{8-2} \cdot 11 = 0^6 \cdot 11 = 00000011.$
- $\text{bin}_{16}(3) = 0000000000000011.$

## Was ist mit negative Zahlen?

- Idee: Verwende das erste Bit, um zu speichern, ob die Zahl positiv oder negativ ist.
- Beispiel:  $5 = 0101_{zkpl}$ ,  $-5 = 1101_{zkpl}$ .

## Zweierkomplement Darstellung

Die Zweierkomplementdarstellung einer Zahl  $x$  mit der Länge  $\ell$  ist wie folgt definiert:

$$\mathbf{zkpl}_{\ell}(x) = \begin{cases} 0\mathbf{bin}_{\ell-1}(x) & \text{falls } x \geq 0 \\ 1\mathbf{bin}_{\ell-1}(2^{\ell-1} + x) & \text{falls } x < 0 \end{cases}$$

- Wieso  $\ell - 1$ ?

Vollständige  
Induktion

Formale Sprache

Übersetzung und  
Kodierung

Kodierung von  
Zahlen

Repräsentation von  
Zahlen

Zweierkomplement-  
Darstellung



# Aufgaben zu Zweierkomplement-Darstellung

Vollständige  
Induktion

Formale Sprache

Übersetzung und  
Kodierung

Kodierung von  
Zahlen

Repräsentation von  
Zahlen

Zweierkomplement-  
Darstellung

$$\mathbf{Zkpl}_\ell(x) = \begin{cases} 0\mathbf{bin}_{\ell-1}(x) & \text{falls } x \geq 0 \\ 1\mathbf{bin}_{\ell-1}(2^{\ell-1} + x) & \text{falls } x < 0 \end{cases}$$

Was sind folgende Zahlen in Zweierkomplement-Darstellung?

- $\mathbf{Zkpl}_4(3) = 0011$ .
- $\mathbf{Zkpl}_4(7) = 0111$ .
- $\mathbf{Zkpl}_4(-5) = 1101$ .
- $\mathbf{Zkpl}_8(13) = 00001101$ .
- $\mathbf{Zkpl}_8(-34) = 10100010$ .
- $\mathbf{Zkpl}_8(-9) = 10001001$ .

Lukas Bach, lu-  
kas.bach@student.kit.edu

Vollständige  
Induktion

Formale Sprache

Übersetzung und  
Kodierung

Kodierung von  
Zahlen

Repräsentation von  
Zahlen

Zweierkomplement-  
Darstellung

## Zum Tutorium

- Lukas Bach
- Tutorienfolien auf:
  - <http://gbi.lukasbach.com>
- Tutorium findet statt:
  - Donnerstags, 14:00 - 15:30
  - 50.34 Informatikbau, -107

## Mehr Material

- Ehemalige GBI Webseite:
  - <http://gbi.ira.uka.de>
  - Altklausuren!

## Zur Veranstaltung

- Grundbegriffe der Informatik
- Klausurtermin:
  - 06.03.2017, 11:00
  - Zwei Stunden Bearbeitungszeit
  - 6 ECTS für Informatiker und Informationswirte, 4 ECTS für Mathematiker und Physiker

## Zum Übungsschein

- Übungsblatt jede Woche
- Ab 50% insgesamt hat man den Übungsschein
- Keine Voraussetzung für die Klausur, aber für das Modul