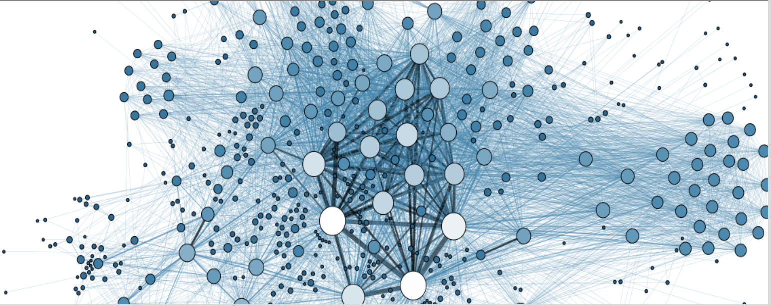
# Grundbegriffe der Informatik
# Tutorium 33

Lukas Bach, lukas.bach@student.kit.edu | 17.11.2016



# Quiz

Lukas Bach, lukas.bach@student.kit.edu

Lukas Bach, lukas.bach@student.kit.edu



# Quiz

- Was macht die Funktion $val_l$?

Lukas Bach, lukas.bach@student.kit.edu



# Quiz

- Was macht die Funktion $val_l$?
- Was bedeutet Äquivalenz?

Lukas Bach, lukas.bach@student.kit.edu



# Quiz

- Was macht die Funktion $val_I$?
- Was bedeutet Äquivalenz?
- Was bedeutet Tautologie und Erfüllbarkeit?

Lukas Bach, lukas.bach@student.kit.edu



# Quiz

- Was macht die Funktion $val_l$?
- Was bedeutet Äquivalenz?
- Was bedeutet Tautologie und Erfüllbarkeit?
- Welche dieser Aussagen sind erfüllbar?

Lukas Bach, lukas.bach@student.kit.edu



# Quiz

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

Lukas Bach, lukas.bach@student.kit.edu



# Quiz

Karlsruher Institut für Technologie

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

Lukas Bach, lukas.bach@student.kit.edu



# Wahrheitsgehalt von unendlich Aussagen

Beispielsituation:

Lukas Bach, lukas.bach@student.kit.edu



# Wahrheitsgehalt von unendlich Aussagen

Beispielsituation: Wir haben unendlich viele Dominosteine.

Lukas Bach, lukas.bach@student.kit.edu



# Wahrheitsgehalt von unendlich Aussagen

Beispielsituation: Wir haben unendlich viele Dominosteine. Behauptung:

Lukas Bach, lukas.bach@student.kit.edu



# Wahrheitsgehalt von unendlich Aussagen

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

Lukas Bach, lukas.bach@student.kit.edu



# Wahrheitsgehalt von unendlich Aussagen

Karlsruher Institut für Technologie

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

- Wir haben Aussagen: {“1. Stein fällt um”, “2. Stein fällt um”, ...}

Lukas Bach, lukas.bach@student.kit.edu



# 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?

Lukas Bach, lukas.bach@student.kit.edu



# Wahrheitsgehalt von unendlich Aussagen

Karlsruher Institut für Technologie

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:

Lukas Bach, lukas.bach@student.kit.edu



# 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:
  - $A(n) :=$ “$n$-ter Stein fällt um” $\forall n \in \mathbb{N}$.

Lukas Bach, lukas.bach@student.kit.edu



# Wahrheitsgehalt von unendlich Aussagen

Karlsruher Institut für Technologie

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:
  - $A(n) :=$ “$n$-ter Stein fällt um” $\forall n \in \mathbb{N}$.
- Aussage $A :=$ “Alle Steine fallen um” $\equiv A(i)$ ist wahr $\forall i \in \mathbb{N}$.

Lukas Bach, lukas.bach@student.kit.edu



# 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:
  - $A(n) :=$ “$n$-ter Stein fällt um” $\forall n \in \mathbb{N}$.
- Aussage $A :=$ “Alle Steine fallen um” $\equiv A(i)$ ist wahr $\forall i \in \mathbb{N}$.

Wir haben immernoch unendlich viele Aussagen...

# 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:
  - $A(n)$ := “$n$-ter Stein fällt um” $\forall n \in \mathbb{N}$.
- Aussage $A$ := “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

# 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:
  - $A(n) :=$ “$n$-ter Stein fällt um” $\forall n \in \mathbb{N}$.
- Aussage $A :=$ “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 $i \in \mathbb{N}$.

Lukas Bach, lukas.bach@student.kit.edu



# 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:
  - $A(n)$ := “$n$-ter Stein fällt um” $\forall n \in \mathbb{N}$.
- Aussage $A$ := “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:

# 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:
  - $A(n) :=$ “$n$-ter Stein fällt um” $\forall n \in \mathbb{N}$.
- Aussage $A :=$ “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:

Lukas Bach, lukas.bach@student.kit.edu



# 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:
  - $A(n) :=$ "$n$-ter Stein fällt um" $\forall n \in \mathbb{N}$.
- Aussage $A :=$ "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.

# 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:
  - $A(n) :=$ “$n$-ter Stein fällt um” $\forall n \in \mathbb{N}$.
- Aussage $A :=$ “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.

Lukas Bach, lukas.bach@student.kit.edu

# Vollständige Induktion



Karlsruher Institut für Technologie

Lukas Bach, lukas.bach@student.kit.edu



# Vollständige Induktion

- Beweisverfahren

Lukas Bach, lukas.bach@student.kit.edu



# Vollständige Induktion

- Beweisverfahren
- In der Regel zu zeigen: Eine Aussage gilt für alle $n \in \mathbb{N}_+$, manchmal auch für alle $n \in \mathbb{N}_0$

Lukas Bach, lukas.bach@student.kit.edu



# Vollständige Induktion

- 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

Lukas Bach, lukas.bach@student.kit.edu



# Vollständige Induktion

- 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



Lukas Bach, lukas.bach@student.kit.edu

Lukas Bach, lukas.bach@student.kit.edu



# Struktur des Beweises

Behauptung: (*kurz* **Beh.:**)

Beweis: (*kurz* **Bew.:**)

Lukas Bach, lukas.bach@student.kit.edu



# 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

Lukas Bach, lukas.bach@student.kit.edu



# 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]

Lukas Bach, lukas.bach@student.kit.edu



# 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

Lukas Bach, lukas.bach@student.kit.edu



# 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:

Lukas Bach, lukas.bach@student.kit.edu



# 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:

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

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Vollständiger Induktion

## 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.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu vollständiger Induktion

## Übungsaufgaben

Zeige die Wahrheit folgender Aussagen mit vollständiger Induktion:

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

Lukas Bach, lukas.bach@student.kit.edu

# Formale Sprache

Lukas Bach, lukas.bach@student.kit.edu



# Formale Sprache

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

Lukas Bach, lukas.bach@student.kit.edu



# Formale Sprache

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

Lukas Bach, lukas.bach@student.kit.edu



# Formale Sprache

Karlsruher Institut für Technologie

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

**Formale Sprache**

Eine Formale Sprache $L$

Lukas Bach, lukas.bach@student.kit.edu



# Formale Sprache

- 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 $L \subseteq A^*$.

Lukas Bach, lukas.bach@student.kit.edu



# Formale Sprache

Karlsruher Institut für Technologie

- 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:

Lukas Bach, lukas.bach@student.kit.edu



# Formale Sprache

Karlsruher Institut für Technologie

- 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:

- $A := \{b, n, a\}$.

# Formale Sprache

- 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 $A$.

Lukas Bach, lukas.bach@student.kit.edu



# Formale Sprache

Karlsruher Institut für Technologie

- 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 $A$.
  - $L_2 := \{banana, bananana, banananana, ...\}$

Lukas Bach, lukas.bach@student.kit.edu



# Formale Sprache

- 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 $A$.
  - $L_2 := \{banana, bananana, banananana, \ldots\}$
    $= \{w : w = bana(na)^k, k \in \mathbb{N}\}$ auch.

Lukas Bach, lukas.bach@student.kit.edu



# Formale Sprache

Karlsruher Institut für Technologie

- 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 $A$.
  - $L_2 := \{banana, bananana, banananana, \ldots\}$
    $= \{w : w = bana(na)^k, k \in \mathbb{N}\}$ auch.
  - $L_3 := \{ban, baan, baaan, \ldots\}$ auch.

Lukas Bach, lukas.bach@student.kit.edu



# Formale Sprache

- 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 $A$.
  - $L_2 := \{banana, bananana, banananana, \ldots\}$ $= \{w : w = bana(na)^k, k \in \mathbb{N}\}$ auch.
  - $L_3 := \{ban, baan, baaan, \ldots\}$ auch. Andere Schreibweise?

Lukas Bach, lukas.bach@student.kit.edu



# Formale Sprache

- 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 $A$.
  - $L_2 := \{banana, bananana, banananana, \ldots\}$
    $= \{w : w = bana(na)^k, k \in \mathbb{N}\}$ auch.
  - $L_3 := \{ban, baan, baaan, \ldots\}$ auch. Andere Schreibweise?
    $L_3 = \{w : w = ba^k n, k \in \mathbb{N}\}$

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

Karlsruher Institut für Technologie

## Produkt von formalen Sprachen

Von zwei formalen Sprachen $L_1$, $L_2$

# Produkt von Sprachen

Lukas Bach, lukas.bach@student.kit.edu



## Produkt von formalen Sprachen

Von zwei formalen Sprachen $L_1, L_2$ lässt sich das Produkt

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

Karlsruher Institut für Technologie

## Produkt von formalen Sprachen

Von zwei formalen Sprachen $L_1, L_2$ lässt sich das Produkt $L_1 \cdot L_2$

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

Karlsruher Institut für Technologie

## Produkt von formalen Sprachen

Von zwei formalen Sprachen $L_1, L_2$ lässt sich das Produkt $L_1 \cdot L_2$ bilden mit

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

Karlsruher Institut für Technologie

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

# Produkt von Sprachen

## 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? $L_1 = \{aaa\} \cdot \{bb, aaaa\}$.

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

## 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 $\varepsilon$?

# Produkt von Sprachen

## 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 $\varepsilon$? $L_2 = A \cdot A^*$

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

Karlsruher Institut für Technologie

## 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 $\varepsilon$? $L_2 = A \cdot A^* = A^* \setminus \{\varepsilon\}$.

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

## 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:

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

Karlsruher Institut für Technologie

## 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.

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

Karlsruher Institut für Technologie

## 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.

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

## 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.

# Produkt von Sprachen

Lukas Bach, lukas.bach@student.kit.edu



## 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.
- $L_3$

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

Karlsruher Institut für Technologie

## 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 $\varepsilon$? $L_2 = A \cdot A^* = A^* \backslash \{\varepsilon\}$.
- 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.
- $L_3 = B \cdot B \cdot \{\varepsilon, \delta\delta\} \cdot A \cdot A \cdot A \cdot A$.

# Produkt von Sprachen

Lukas Bach, lukas.bach@student.kit.edu



## Übung zu Produkt von formalen Sprachen

Sei $A$ ein beliebiges Alphabet und $M := \{L : L \text{ ist formale Sprache über } A\}$

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

## Übung zu Produkt von formalen Sprachen

Sei $A$ ein beliebiges Alphabet und $M := \{L : L$ ist formale Sprache über $A\} = 2^A$.

Lukas Bach, lukas.bach@student.kit.edu

# Produkt von Sprachen



## Übung zu Produkt von formalen Sprachen

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

Zeige:

Lukas Bach, lukas.bach@student.kit.edu

# Produkt von Sprachen



## 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 $\cdot : M \times M \to M$ darstellen.

Zeige:

- Die Verknüpfung $\cdot$ ist assoziativ.

Lukas Bach, lukas.bach@student.kit.edu

# Produkt von Sprachen



## Übung zu Produkt von formalen Sprachen

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

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

## 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 $\cdot : M \times M \to M$ 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)

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

Karlsruher Institut für Technologie

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

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

- Die Verknüpfung $\cdot$ ist assoziativ:

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

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

- Die Verknüpfung $\cdot$ ist assoziativ:
  - $(L_1 \cdot L_2) \cdot L_3$

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

Karlsruher Institut für Technologie

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

- Die Verknüpfung $\cdot$ ist assoziativ:
  - $(L_1 \cdot L_2) \cdot L_3 = (\{w_1 \cdot w_2 : w_1 \in L_1, w_2 \in L_2\}) \cdot L_3$

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

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

- Die Verknüpfung $\cdot$ ist assoziativ:
  - $(L_1 \cdot L_2) \cdot L_3 = (\{w_1 \cdot w_2 : w_1 \in L_1, w_2 \in L_2\}) \cdot L_3 = \{w_1 w_2 w_3 : w_1 \in L_1, w_2 \in L_2, w_3 \in L_3\}$

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

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

- Die Verknüpfung $\cdot$ ist assoziativ:
  - $(L_1 \cdot L_2) \cdot L_3 = (\{w_1 \cdot w_2 : w_1 \in L_1, w_2 \in L_2\}) \cdot L_3 = \{w_1 w_2 w_3 : w_1 \in L_1, w_2 \in L_2, w_3 \in L_3\} = L_1 \cdot (\{w_2 w_3 : w_2 \in L_2, w_3 \in L_3\})$

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

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

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

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

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)

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

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)
  - $e := \{\varepsilon\}$.

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

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

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

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

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

Karlsruher Institut für Technologie

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

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

Karlsruher Institut für Technologie

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)

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

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)
  - $o := \emptyset$

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

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

Lukas Bach, lukas.bach@student.kit.edu



# Produkt von Sprachen

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 Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

Karlsruher Institut für Technologie

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.

# Potenz von Sprachen

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
  - $L_1^0 = \{\varepsilon\}$.

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

Karlsruher Institut für Technologie

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
  - $L_1^0 = \{\varepsilon\}$. $L_1^1$

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
    - $L_1^0 = \{\varepsilon\}$. $L_1^1 = \{\varepsilon\} \cdot L_1$

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

Karlsruher Institut für Technologie

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
  - $L_1^0 = \{\varepsilon\}$. $L_1^1 = \{\varepsilon\} \cdot L_1 = L_1$.

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
  - $L_1^0 = \{\varepsilon\}$. $L_1^1 = \{\varepsilon\} \cdot L_1 = L_1$.
  - $L_1^2 = (\{\varepsilon\} \cdot L_1) \cdot L_1$

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
  - $L_1^0 = \{\varepsilon\}$. $L_1^1 = \{\varepsilon\} \cdot L_1 = L_1$.
  - $L_1^2 = (\{\varepsilon\} \cdot L_1) \cdot L_1 = \{aa\}$.

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

Karlsruher Institut für Technologie

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
    - $L_1^0 = \{\varepsilon\}$. $L_1^1 = \{\varepsilon\} \cdot L_1 = L_1$.
    - $L_1^2 = (\{\varepsilon\} \cdot L_1) \cdot L_1 = \{aa\}$.
- $L_2 := \{ab\}^3\{c\}^4$

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

Karlsruher Institut für Technologie

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
  - $L_1^0 = \{\varepsilon\}$. $L_1^1 = \{\varepsilon\} \cdot L_1 = L_1$.
  - $L_1^2 = (\{\varepsilon\} \cdot L_1) \cdot L_1 = \{aa\}$.
- $L_2 := \{ab\}^3\{c\}^4$
  - $L_2^0 = \{\varepsilon\}, L_2^1 = \ldots$

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

Karlsruher Institut für Technologie

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
  - $L_1^0 = \{\varepsilon\}$. $L_1^1 = \{\varepsilon\} \cdot L_1 = L_1$.
  - $L_1^2 = (\{\varepsilon\} \cdot L_1) \cdot L_1 = \{aa\}$.
- $L_2 := \{ab\}^3\{c\}^4$
  - $L_2^0 = \{\varepsilon\}, L_2^1 = ...$
  - $L_2^2$

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
  - $L_1^0 = \{\varepsilon\}$. $L_1^1 = \{\varepsilon\} \cdot L_1 = L_1$.
  - $L_1^2 = (\{\varepsilon\} \cdot L_1) \cdot L_1 = \{aa\}$.
- $L_2 := \{ab\}^3\{c\}^4$
  - $L_2^0 = \{\varepsilon\}, L_2^1 = \dots$
  - $L_2^2 = (\{ab\}^3\{c\}^4)^2$

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
  - $L_1^0 = \{\varepsilon\}$. $L_1^1 = \{\varepsilon\} \cdot L_1 = L_1$.
  - $L_1^2 = (\{\varepsilon\} \cdot L_1) \cdot L_1 = \{aa\}$.
- $L_2 := \{ab\}^3\{c\}^4$
  - $L_2^0 = \{\varepsilon\}, L_2^1 = \dots$
  - $L_2^2 = (\{ab\}^3\{c\}^4)^2 = (\{ab\}^3\{cccc\})^2$

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
  - $L_1^0 = \{\varepsilon\}$. $L_1^1 = \{\varepsilon\} \cdot L_1 = L_1$.
  - $L_1^2 = (\{\varepsilon\} \cdot L_1) \cdot L_1 = \{aa\}$.
- $L_2 := \{ab\}^3\{c\}^4$
  - $L_2^0 = \{\varepsilon\}, L_2^1 = ...$
  - $L_2^2 = (\{ab\}^3\{c\}^4)^2 = (\{ab\}^3\{cccc\})^2 = \{abababcccc\}^2$

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen



## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
  - $L_1^0 = \{\varepsilon\}$. $L_1^1 = \{\varepsilon\} \cdot L_1 = L_1$.
  - $L_1^2 = (\{\varepsilon\} \cdot L_1) \cdot L_1 = \{aa\}$.
- $L_2 := \{ab\}^3\{c\}^4$
  - $L_2^0 = \{\varepsilon\}, L_2^1 = ...$
  - $L_2^2 = (\{ab\}^3\{c\}^4)^2 = (\{ab\}^3\{cccc\})^2 = \{abababcccc\}^2 = \{abababccccabababcccc\}$.

Lukas Bach, lukas.bach@student.kit.edu



# Potenz von Sprachen

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
  - $L_1^0 = \{\varepsilon\}$. $L_1^1 = \{\varepsilon\} \cdot L_1 = L_1$.
  - $L_1^2 = (\{\varepsilon\} \cdot L_1) \cdot L_1 = \{aa\}$.
- $L_2 := \{ab\}^3\{c\}^4$
  - $L_2^0 = \{\varepsilon\}, L_2^1 = ...$
  - $L_2^2 = (\{ab\}^3\{c\}^4)^2 = (\{ab\}^3\{cccc\})^2 = \{abababcccc\}^2 = \{abababccccabababcccc\}$.
- $L_3 := (\{a\} \cup \{b\})^2$

# Potenz von Sprachen

Karlsruher Institut für Technologie

## Potenz von Sprachen

Potenz von formellen Sprachen ist wie folgt definiert:

- $L^0 := \{\varepsilon\}$
- $L^{i+1} := L^i \cdot L$ für $i \in \mathbb{N}_0$.

- $L_1 := \{a\}$.
  - $L_1^0 = \{\varepsilon\}$. $L_1^1 = \{\varepsilon\} \cdot L_1 = L_1$.
  - $L_1^2 = (\{\varepsilon\} \cdot L_1) \cdot L_1 = \{aa\}$.
- $L_2 := \{ab\}^3\{c\}^4$
  - $L_2^0 = \{\varepsilon\}, L_2^1 = ...$
  - $L_2^2 = (\{ab\}^3\{c\}^4)^2 = (\{ab\}^3\{cccc\})^2 = \{abababcccc\}^2 = \{abababccccabababcccc\}$.
- $L_3 := (\{a\} \cup \{b\})^2 = \{aa, ab, ba, bb\}$

Lukas Bach, lukas.bach@student.kit.edu



# Konkatenationsabschluss bei formalen Sprachen

Lukas Bach, lukas.bach@student.kit.edu



# Konkatenationsabschluss bei formalen Sprachen

Karlsruher Institut für Technologie

## Konkatenationsabschluss

Zu einer formalen Sprache $L$

Lukas Bach, lukas.bach@student.kit.edu



# Konkatenationsabschluss bei formalen Sprachen

Karlsruher Institut für Technologie

## Konkatenationsabschluss

Zu einer formalen Sprache $L$ ist der Konkatenationsabschluss $L^*$ definiert

Lukas Bach, lukas.bach@student.kit.edu



# Konkatenationsabschluss bei formalen Sprachen

## Konkatenationsabschluss

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

Lukas Bach, lukas.bach@student.kit.edu



# Konkatenationsabschluss bei formalen Sprachen

## 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 $L$

Lukas Bach, lukas.bach@student.kit.edu



# Konkatenationsabschluss bei formalen Sprachen

## 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

Lukas Bach, lukas.bach@student.kit.edu



# Konkatenationsabschluss bei formalen Sprachen

## 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 $L^+ := \bigcup_{i \in \mathbb{N}_+} L^i$.

# Konkatenationsabschluss bei formalen Sprachen

## 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 $L \subseteq A^* \setminus \{\varepsilon\}$?

Lukas Bach, lukas.bach@student.kit.edu



# Konkatenationsabschluss bei formalen Sprachen

Karlsruher Institut für Technologie

## 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 $L \subseteq A^* \setminus \{\varepsilon\}$?
- $L := \{a, b, c\}.L^*$

Lukas Bach, lukas.bach@student.kit.edu



# Konkatenationsabschluss bei formalen Sprachen

## 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 $L \subseteq A^* \backslash \{\varepsilon\}$?
- $L := \{a, b, c\}. L^* = \{\varepsilon, a, aa, ab, ac, aaa, aab, \dots, b, ba, bb, \dots\}$

Lukas Bach, lukas.bach@student.kit.edu



# Konkatenationsabschluss bei formalen Sprachen

## 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 $L \subseteq A^* \setminus \{\varepsilon\}$?
- $L := \{a, b, c\}. L^* = \{\varepsilon, a, aa, ab, ac, aaa, aab, \ldots, b, ba, bb, \ldots\}$
- $L := \{aa, bc\}. L^*$

Lukas Bach, lukas.bach@student.kit.edu



# Konkatenationsabschluss bei formalen Sprachen

## 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 $L \subseteq A^* \setminus \{\varepsilon\}$?
- $L := \{a, b, c\}. L^* = \{\varepsilon, a, aa, ab, ac, aaa, aab, \dots, b, ba, bb, \dots\}$
- $L := \{aa, bc\}. L^* = \{\varepsilon, aa, bc, aa \cdot aa, aa \cdot bc, bc \cdot aa, bc \cdot bc, aa \cdot aa \cdot aa, \dots\}$

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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

# Übung zu Konkatenationsabschluss

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

- Sprache $L_1 \subseteq A^*$, die das Teilwort *ab* nicht enthält?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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

- Sprache $L_1 \subseteq A^*$, die das Teilwort *ab* nicht enthält? $L_1 = \{b\}^*\{a\}^*$.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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.
  - $B := \{\_, a, b, ..., z, A, B, ..., Z\}$

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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.
  - $B := \{\_, a, b, ..., z, A, B, ..., Z\}$
  - $C := B \cup \mathbb{Z}_9$

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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.
  - $B := \{\_, a, b, ..., z, A, B, ..., Z\}$
  - $C := B \cup \mathbb{Z}_9$
  - $L_2 \subseteq C$

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

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.
  - $B := \{\_, a, b, ..., z, A, B, ..., Z\}$
  - $C := B \cup \mathbb{Z}_9$
  - $L_2 \subseteq C = (B \cdot C^*) \backslash \{if, class, while, ...\}$

# Übung zu Konkatenationsabschluss

Lukas Bach, lukas.bach@student.kit.edu

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

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

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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

- Was ist alles in $L$ drin?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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

- Was ist alles in $L$ drin?
  - $aaabbabbaaabba$?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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

- Was ist alles in $L$ drin?
  - *aaabbabbaaabba*? Nein.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

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

- Was ist alles in $L$ drin?
  - *aaabbabbaaabba*? Nein.
  - *aaabb*, *abbaaabba*?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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

- Was ist alles in $L$ drin?
  - *aaabbabbaaabba*? Nein.
  - *aaabb*, *abbaaabba*? Ja, nein.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

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

- Was ist alles in $L$ drin?
  - $aaabbabbaaabba$? Nein.
  - $aaabb$, $abbaaabba$? Ja, nein.
  - $aaabb$, $abb$, $aaabba$?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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

- Was ist alles in $L$ drin?
  - *aaabbabbaaabba*? Nein.
  - *aaabb*, *abbaaabba*? Ja, nein.
  - *aaabb*, *abb*, *aaabba*? Ja, ja, nein.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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

- Was ist alles in $L$ drin?
  - $aaabbabbaaabba$? Nein.
  - $aaabb$, $abbaaabba$? Ja, nein.
  - $aaabb$, $abb$, $aaabba$? Ja, ja, nein.
  - $aaabb$, $abb$, $aaabb$, $a$?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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

- Was ist alles in $L$ drin?
  - $aaabbabbaaabba$? Nein.
  - $aaabb$, $abbaaabba$? Ja, nein.
  - $aaabb$, $abb$, $aaabba$? Ja, ja, nein.
  - $aaabb$, $abb$, $aaabb$, $a$? Alles drin.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

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

- Was ist alles in $L$ drin?
  - $aaabbabbaaabba$? Nein.
  - $aaabb$, $abbaaabba$? Ja, nein.
  - $aaabb$, $abb$, $aaabba$? Ja, ja, nein.
  - $aaabb$, $abb$, $aaabb$, $a$? Alles drin.
- Was ist alles in $L^*$ drin?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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

- Was ist alles in $L$ drin?
  - *aaabbabbaaabba*? Nein.
  - *aaabb*, *abbaaabba*? Ja, nein.
  - *aaabb*, *abb*, *aaabba*? Ja, ja, nein.
  - *aaabb*, *abb*, *aaabb*, *a*? Alles drin.
- Was ist alles in $L^*$ drin?
  - *aaabbabbaaabba*?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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

- Was ist alles in $L$ drin?
  - *aaabbabbaaabba*? Nein.
  - *aaabb*, *abbaaabba*? Ja, nein.
  - *aaabb*, *abb*, *aaabba*? Ja, ja, nein.
  - *aaabb*, *abb*, *aaabb*, *a*? Alles drin.
- Was ist alles in $L^*$ drin?
  - *aaabbabbaaabba*? Ja.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

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

- Was ist alles in $L$ drin?
  - $aaabbabbaaabba$? Nein.
  - $aaabb$, $abbaaabba$? Ja, nein.
  - $aaabb$, $abb$, $aaabba$? Ja, ja, nein.
  - $aaabb$, $abb$, $aaabb$, $a$? Alles drin.
- Was ist alles in $L^*$ drin?
  - $aaabbabbaaabba$? Ja.
  - $aaabb$, $abbaaabba$?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

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

- Was ist alles in $L$ drin?
  - $aaabbabbaaabba$? Nein.
  - $aaabb$, $abbaaabba$? Ja, nein.
  - $aaabb$, $abb$, $aaabba$? Ja, ja, nein.
  - $aaabb$, $abb$, $aaabb$, $a$? Alles drin.
- Was ist alles in $L^*$ drin?
  - $aaabbabbaaabba$? Ja.
  - $aaabb$, $abbaaabba$? Ja.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

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

- Was ist alles in $L$ drin?
  - *aaabbabbaaabba*? Nein.
  - *aaabb*, *abbaaabba*? Ja, nein.
  - *aaabb*, *abb*, *aaabba*? Ja, ja, nein.
  - *aaabb*, *abb*, *aaabb*, *a*? Alles drin.
- Was ist alles in $L^*$ drin?
  - *aaabbabbaaabba*? Ja.
  - *aaabb*, *abbaaabba*? Ja.
  - *aaabb*, *abb*, *aaabba*?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

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

- Was ist alles in $L$ drin?
  - *aaabbabbaaabba*? Nein.
  - *aaabb*, *abbaaabba*? Ja, nein.
  - *aaabb*, *abb*, *aaabba*? Ja, ja, nein.
  - *aaabb*, *abb*, *aaabb*, *a*? Alles drin.
- Was ist alles in $L^*$ drin?
  - *aaabbabbaaabba*? Ja.
  - *aaabb*, *abbaaabba*? Ja.
  - *aaabb*, *abb*, *aaabba*? Ja.

# Übung zu Konkatenationsabschluss

Lukas Bach, lukas.bach@student.kit.edu



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

- Was ist alles in $L$ drin?
  - $aaabbabbaaabba$? Nein.
  - $aaabb$, $abbaaabba$? Ja, nein.
  - $aaabb$, $abb$, $aaabba$? Ja, ja, nein.
  - $aaabb$, $abb$, $aaabb$, $a$? Alles drin.
- Was ist alles in $L^*$ drin?
  - $aaabbabbaaabba$? Ja.
  - $aaabb$, $abbaaabba$? Ja.
  - $aaabb$, $abb$, $aaabba$? Ja.
  - $aaabb$, $abb$, $aaabb$, $a$?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

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

- Was ist alles in $L$ drin?
  - $aaabbabbaaabba$? Nein.
  - $aaabb$, $abbaaabba$? Ja, nein.
  - $aaabb$, $abb$, $aaabba$? Ja, ja, nein.
  - $aaabb$, $abb$, $aaabb$, $a$? Alles drin.
- Was ist alles in $L^*$ drin?
  - $aaabbabbaaabba$? Ja.
  - $aaabb$, $abbaaabba$? Ja.
  - $aaabb$, $abb$, $aaabba$? Ja.
  - $aaabb$, $abb$, $aaabb$, $a$? Ja.

# Übung zu Konkatenationsabschluss



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

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

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

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

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

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

## Erinnerung

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

## Beweisaufgabe

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

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

## Erinnerung

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

## Beweisaufgabe

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

$\subseteq$:

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

## Erinnerung

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

## Beweisaufgabe

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

$\subseteq$:

Voraussetzung:

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

## Erinnerung

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

## Beweisaufgabe

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

$\subseteq$:

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

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

## Erinnerung

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

## 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 $w' \in L^i$

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

## Erinnerung

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

## 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 $w = w'w'' \in L^i \cdot L$

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

## Erinnerung

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

## 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 $w = w'w'' \in L^i \cdot L = L^{i+1}$.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

## Erinnerung

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

## 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 $i + 1 \in \mathbb{N}_+$

# Übung zu Konkatenationsabschluss

Lukas Bach, lukas.bach@student.kit.edu



## Erinnerung

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

## 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: $L^{i+1} \subseteq L^+$

# Übung zu Konkatenationsabschluss

## Erinnerung

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

## 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 $w \in L^+$.

$\supseteq$:

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

## Erinnerung

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

## 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:

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

## Erinnerung

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

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

# Übung zu Konkatenationsabschluss

Lukas Bach, lukas.bach@student.kit.edu



## Erinnerung

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

## 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 $w \in L^i$.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

## Erinnerung

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

## 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 $i \in \mathbb{N}_+$

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

## Erinnerung

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

## 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 $i = j + 1$

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

## Erinnerung

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

## 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 $j \in \mathbb{N}_0$

# Übung zu Konkatenationsabschluss

Lukas Bach, lukas.bach@student.kit.edu



## Erinnerung

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

## 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 $j \in \mathbb{N}_0$: $w \in L^{j+1}$

# Übung zu Konkatenationsabschluss

Lukas Bach, lukas.bach@student.kit.edu



## Erinnerung

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

## 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 $j \in \mathbb{N}_0$: $w \in L^{j+1} = L^j \cdot L$.

# Übung zu Konkatenationsabschluss

Lukas Bach, lukas.bach@student.kit.edu



## Erinnerung

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

## 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 $w'' \in L$.

# Übung zu Konkatenationsabschluss

Lukas Bach, lukas.bach@student.kit.edu



## Erinnerung

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

## 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 $L^j \subseteq L^*$

# Übung zu Konkatenationsabschluss

Lukas Bach, lukas.bach@student.kit.edu



## Erinnerung

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

## 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 $w = w'w''$

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu Konkatenationsabschluss

Karlsruher Institut für Technologie

## Erinnerung

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

## 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 $w = w'w'' \in L^* \cdot L$.

# Übung zu formalen Sprachen

$L_1, L_2$ seien formale Sprachen.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu formalen Sprachen

Karlsruher Institut für Technologie

$L_1, L_2$ seien formale Sprachen.

- Wie sieht $L_1 \cdot L_2$ aus?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu formalen Sprachen



$L_1, L_2$ seien formale Sprachen.

- Wie sieht $L_1 \cdot L_2$ aus?
- Wie sieht $L_1^3$ aus?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu formalen Sprachen

Karlsruher Institut für Technologie

$L_1, L_2$ 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?

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu formalen Sprachen

$L_1, L_2$ 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?

Lukas Bach, lukas.bach@student.kit.edu

# Herführung zu Zahlendarstellungen

Lukas Bach, lukas.bach@student.kit.edu



# Herführung zu Zahlendarstellungen

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

# Herführung zu Zahlendarstellungen

Lukas Bach, lukas.bach@student.kit.edu



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

- Was können wir daraus machen?

Lukas Bach, lukas.bach@student.kit.edu



# Herführung zu Zahlendarstellungen

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

- Was können wir daraus machen?
- $A_{dez}^* \supset \{42, 1337, 999\}$.

Lukas Bach, lukas.bach@student.kit.edu



# Herführung zu Zahlendarstellungen

Karlsruher Institut für Technologie

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

- Was können wir daraus machen?
- $A_{dez}^* \supset \{42, 1337, 999\}$.
- $A_{bin}^* \supset \{101010, 10100111001, 1111100111\}$.

Lukas Bach, lukas.bach@student.kit.edu



# Herführung zu Zahlendarstellungen

Karlsruher Institut für Technologie

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

- Was können wir daraus machen?
- $A_{dez}^* \supset \{42, 1337, 999\}$.
- $A_{bin}^* \supset \{101010, 10100111001, 1111100111\}$.
- $A_{oct}^* \supset \{52, 2471, 1747\}$.

Lukas Bach, lukas.bach@student.kit.edu



# Herführung zu Zahlendarstellungen

Karlsruher Institut für Technologie

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.

Lukas Bach, lukas.bach@student.kit.edu



# Herführung zu Zahlendarstellungen

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 $42_{\in A_{dez}} \overset{Deutung}{=} 101010_{\in A_{bin}} \overset{Deutung}{=} 52_{\in A_{oct}}$.

# Definition von Zahlendarstellungen

Karlsruher Institut für Technologie

Lukas Bach, lukas.bach@student.kit.edu

Lukas Bach, lukas.bach@student.kit.edu



# Definition von Zahlendarstellungen

Karlsruher Institut für Technologie

**$Num_k$**

Einer Zeichenkette $Z_k$ aus Ziffern

Lukas Bach, lukas.bach@student.kit.edu



# Definition von Zahlendarstellungen

Karlsruher Institut für Technologie

## $Num_k$

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

Lukas Bach, lukas.bach@student.kit.edu



# Definition von Zahlendarstellungen

Karlsruher Institut für Technologie

## $Num_k$

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

$$Num_k(\varepsilon) = 0$$

Lukas Bach, lukas.bach@student.kit.edu



# Definition von Zahlendarstellungen

## $Num_k$

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

$$Num_k(\varepsilon) = 0$$

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

# Definition von Zahlendarstellungen

Lukas Bach, lukas.bach@student.kit.edu



## $Num_k$

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.

Lukas Bach, lukas.bach@student.kit.edu



# Definition von Zahlendarstellungen

**$Num_k$**

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

$Num_k(\varepsilon) = 0$

$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ 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: $Num_k(w) \neq num_k(w)$!

Lukas Bach, lukas.bach@student.kit.edu



# Definition von Zahlendarstellungen

Karlsruher Institut für Technologie

## $Num_k$

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

$$Num_k(\varepsilon) = 0$$

$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ 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: $num_{10}(3)$

Lukas Bach, lukas.bach@student.kit.edu



# Definition von Zahlendarstellungen

Karlsruher Institut für Technologie

## $Num_k$

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: $num_{10}(3) = 3$

Lukas Bach, lukas.bach@student.kit.edu



# Definition von Zahlendarstellungen

Karlsruher Institut für Technologie

## $Num_k$

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

$$Num_k(\varepsilon) = 0$$

$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ 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: $num_{10}(3) = 3$, $num_{10}(7)$

Lukas Bach, lukas.bach@student.kit.edu



# Definition von Zahlendarstellungen

Karlsruher Institut für Technologie

## $Num_k$

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

$$Num_k(\varepsilon) = 0$$

$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ 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: $num_{10}(3) = 3$, $num_{10}(7) = 7$

Lukas Bach, lukas.bach@student.kit.edu



# Definition von Zahlendarstellungen

## $Num_k$

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

$$Num_k(\varepsilon) = 0$$

$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ 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: $num_{10}(3) = 3$, $num_{10}(7) = 7$, $num_{10}(11) =$

Lukas Bach, lukas.bach@student.kit.edu



# Definition von Zahlendarstellungen

Karlsruher Institut für Technologie

## $Num_k$

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.

Lukas Bach, lukas.bach@student.kit.edu



# Definition von Zahlendarstellungen

Karlsruher Institut für Technologie

## $Num_k$

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 $Num_k$!

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

$Num_k(\varepsilon) = 0$.
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123)$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

Karlsruher Institut für Technologie

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3)$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) =$ $10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3)$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) = 10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) = 10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3)$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) =$
  $10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) =$
  $10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot (1 + 10 \cdot 2) + 3$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

Karlsruher Institut für Technologie

$Num_k(\varepsilon) = 0$.
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) =$
  $10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) =$
  $10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot (1 + 10 \cdot 2) + 3 = 123$.

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) = 10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) = 10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot (1 + 10 \cdot 2) + 3 = 123.$

Yay?

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

Karlsruher Institut für Technologie

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) =$
  $10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) =$
  $10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot (1 + 10 \cdot 2) + 3 = 123.$

Yay?

Was ist der dezimale Zahlenwert der Binärzahl 1010?

# Beispiel zu Zahlendarstellungen



$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) = 10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) = 10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot (1 + 10 \cdot 2) + 3 = 123.$

Yay?

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

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

$Num_k(\varepsilon) = 0$.
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) =$
  $10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) =$
  $10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot (1 + 10 \cdot 2) + 3 = 123.$

Yay?

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

- $Num_2(1010)$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) = 10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) = 10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot (1 + 10 \cdot 2) + 3 = 123.$

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

- $Num_2(1010) = 2 \cdot Num_2(101) + num_2(0)$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) = 10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) = 10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot (1 + 10 \cdot 2) + 3 = 123.$

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

- $Num_2(1010) = 2 \cdot Num_2(101) + num_2(0) = 2 \cdot (2 \cdot Num_2(10) + num_2(1) + num_2(0)$

# Beispiel zu Zahlendarstellungen

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) =$
  $10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) =$
  $10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot (1 + 10 \cdot 2) + 3 = 123.$

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

- $Num_2(1010) = 2 \cdot Num_2(101) + num_2(0) =$
  $2 \cdot (2 \cdot Num_2(10) + num_2(1) + num_2(0) =$
  $2 \cdot (2 \cdot (2 \cdot Num_2(1) + num_2(0)) + num_2(1)) + num_2(0)$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

Karlsruher Institut für Technologie

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) =$
  $10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) =$
  $10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot (1 + 10 \cdot 2) + 3 = 123.$

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

- $Num_2(1010) = 2 \cdot Num_2(101) + num_2(0) =$
  $2 \cdot (2 \cdot Num_2(10) + num_2(1) + num_2(0) =$
  $2 \cdot (2 \cdot (2 \cdot Num_2(1) + num_2(0)) + num_2(1)) + num_2(0) =$
  $2 \cdot (2 \cdot (2 \cdot num_2(1) + num_2(0)) + num_2(1)) + num_2(0)$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

Karlsruher Institut für Technologie

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) =$
  $10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) =$
  $10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot (1 + 10 \cdot 2) + 3 = 123.$

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

- $Num_2(1010) = 2 \cdot Num_2(101) + num_2(0) =$
  $2 \cdot (2 \cdot Num_2(10) + num_2(1) + num_2(0) =$
  $2 \cdot (2 \cdot (2 \cdot Num_2(1) + num_2(0)) + num_2(1)) + num_2(0) =$
  $2 \cdot (2 \cdot (2 \cdot num_2(1) + num_2(0)) + num_2(1)) + num_2(0) =$
  $2 \cdot (2 \cdot (2 \cdot 1 + 0) + 1) + 0) = 10.$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu Zahlendarstellungen

Karlsruher Institut für Technologie

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ und $x \in Z_k$.

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

- $Num_{10}(123) = 10 \cdot Num_{10}(12) + num_{10}(3) =$
  $10 \cdot (Num_{10}(1) + num_{10}(2)) + num_{10}(3) =$
  $10 \cdot (num_{10}(1) + 10 \cdot num_{10}(2)) + num_{10}(3) = 10 \cdot (1 + 10 \cdot 2) + 3 = 123.$

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

- $Num_2(1010) = 2 \cdot Num_2(101) + num_2(0) =$
  $2 \cdot (2 \cdot Num_2(10) + num_2(1) + num_2(0) =$
  $2 \cdot (2 \cdot (2 \cdot Num_2(1) + num_2(0)) + num_2(1)) + num_2(0) =$
  $2 \cdot (2 \cdot (2 \cdot num_2(1) + num_2(0)) + num_2(1)) + num_2(0) =$
  $2 \cdot (2 \cdot (2 \cdot 1 + 0) + 1) + 0) = 10.$

Yay!

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu Zahlendarstellungen

$Num_k(\varepsilon) = 0.$
$Num_k(wx) = k \cdot Num_k(w) + num_k(x)$ mit $w \in Z_k^*$ 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: $1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$.

Lukas Bach, lukas.bach@student.kit.edu

# Aufgaben zu Zahlendarstellungen

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu Zahlendarstellungen

Lösungen:

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu Zahlendarstellungen

Lösungen:

- $Num_8(345)$

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu Zahlendarstellungen

Lösungen:

- $Num_8(345) = 229$.

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu Zahlendarstellungen

Lösungen:

- $Num_8(345) = 229$.
- $Num_2(11001)$

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu Zahlendarstellungen

Lösungen:

- $Num_8(345) = 229$.
- $Num_2(11001) = 25$.

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu Zahlendarstellungen

Lösungen:

- $Num_8(345) = 229.$
- $Num_2(11001) = 25.$
- $Num_2(1000)$

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu Zahlendarstellungen

Karlsruher Institut für Technologie

Lösungen:

- $Num_8(345) = 229$.
- $Num_2(11001) = 25$.
- $Num_2(1000) = 8$.

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu Zahlendarstellungen

Lösungen:

- $Num_8(345) = 229.$
- $Num_2(11001) = 25.$
- $Num_2(1000) = 8.$
- $Num_4(123)$

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu Zahlendarstellungen

Lösungen:

- $Num_8(345) = 229$.
- $Num_2(11001) = 25$.
- $Num_2(1000) = 8$.
- $Num_4(123) = 27$.

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu Zahlendarstellungen

Karlsruher Institut für Technologie

Lösungen:

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

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu Zahlendarstellungen

Karlsruher Institut für Technologie

Lösungen:

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

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

Es gilt:
$2(2(2(2(2\cdot 1+0)+1)+0)+1)+0$

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

Es gilt:

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

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

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:

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

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

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:
$Num_k(w) = k^0 \cdot w(0) + k^1 \cdot w(1) + k^2 \cdot w(2) + ....$
Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

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:
$Num_k(w) = k^0 \cdot w(0) + k^1 \cdot w(1) + k^2 \cdot w(2) + ....$
Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

- $Num_2(10101)$

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

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:

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

Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

- $Num_2(10101) = 21.$

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

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:
$Num_k(w) = k^0 \cdot w(0) + k^1 \cdot w(1) + k^2 \cdot w(2) + ....$
Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

- $Num_2(10101) = 21.$
- $Num_2(11101)$

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

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:
$Num_k(w) = k^0 \cdot w(0) + k^1 \cdot w(1) + k^2 \cdot w(2) + ....$
Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

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

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

Karlsruher Institut für Technologie

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:
$Num_k(w) = k^0 \cdot w(0) + k^1 \cdot w(1) + k^2 \cdot w(2) + ....$
Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

- $Num_2(10101) = 21.$
- $Num_2(11101) = 29.$
- $Num_2(111111111)$

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

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:

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

Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

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

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

Karlsruher Institut für Technologie

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

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

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

Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

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

Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

- $Num_{16}(A1)$

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

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

Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

- $Num_{16}(A1) = 161$.

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

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

Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

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

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

Karlsruher Institut für Technologie

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

Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

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

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

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

Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

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

Lukas Bach, lukas.bach@student.kit.edu



# Einfachere Umrechnung von Zahlendarstellungen

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

Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

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

Lukas Bach, lukas.bach@student.kit.edu

# Rechnen mit *div* und *mod*

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

## *div* Funktion

Die Funktion *div* dividiert ganzzahlig.

# Rechnen mit *div* und *mod*

## *div* Funktion

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

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*



## *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.

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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 div 8

# Rechnen mit *div* und *mod*



## *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 div 8 $= 2$

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*



## *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.

- $22 \operatorname{div} 8 = 2 \; (\frac{22}{8} = 2,75)$.

# Rechnen mit *div* und *mod*



## *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.

- $22 \text{ div } 8 = 2$ ($\frac{22}{8} = 2,75$).
- $22 \text{ mod } 8$

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \; (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

# Rechnen mit *div* und *mod*

## *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.

- $22 \text{ div } 8 = 2$ ($\frac{22}{8} = 2,75$).
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | | | | | | | | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \; (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | | | | | | | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \; (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | | | | | | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

## *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.

- $22 \text{ div } 8 = 2$ ($\frac{22}{8} = 2,75$).
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | | | | | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*



## *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.

- $22 \text{ div } 8 = 2$ ($\frac{22}{8} = 2,75$).
- $22 \bmod 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | | | | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*



## *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.

- $22 \text{ div } 8 = 2 \ (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | | | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2$ $(\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*



## *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.

- $22 \text{ div } 8 = 2$ ($\frac{22}{8} = 2,75$).
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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 *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \; (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \ (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | | | |

# Rechnen mit *div* und *mod*

## *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.

- $22 \text{ div } 8 = 2$ ($\frac{22}{8} = 2,75$).
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \; (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | |

# Rechnen mit *div* und *mod*

## *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.

- $22 \text{ div } 8 = 2 \; (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | | | | | | | | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2$ ($\frac{22}{8} = 2,75$).
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | 0 | | | | | | | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*



## *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.

- $22 \text{ div } 8 = 2 \ (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | 0 | 1 | | | | | | | | | | | |

# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \; (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | 0 | 1 | 2 | | | | | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \; (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | 0 | 1 | 2 | 3 | | | | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \ (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | 0 | 1 | 2 | 3 | 0 | | | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2$ $(\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | 0 | 1 | 2 | 3 | 0 | 1 | | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \; (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \; (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2$ ($\frac{22}{8} = 2,75$).
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \; (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

### *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.

- $22 \text{ div } 8 = 2$ ($\frac{22}{8} = 2,75$).
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | | |

Lukas Bach, lukas.bach@student.kit.edu



# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \; (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | |

# Rechnen mit *div* und *mod*

Karlsruher Institut für Technologie

## *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.

- $22 \text{ div } 8 = 2 \; (\frac{22}{8} = 2,75)$.
- $22 \text{ mod } 8 = 6$.

Fülle die Tabelle aus:

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x *div* 4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 |
| x *mod* 4 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 |

Lukas Bach, lukas.bach@student.kit.edu



# Von Zeichen zu Zahlen zurück zu Zahlen

$11101_2$ ist also $29_{10}$.

Lukas Bach, lukas.bach@student.kit.edu



# Von Zeichen zu Zahlen zurück zu Zahlen

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

Lukas Bach, lukas.bach@student.kit.edu



# Von Zeichen zu Zahlen zurück zu Zahlen

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

**$k$-äre Darstellung**

Die Repräsentation einer Zahl $n$

Lukas Bach, lukas.bach@student.kit.edu



# Von Zeichen zu Zahlen zurück zu Zahlen

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

## *k*-äre Darstellung

Die Repräsentation einer Zahl $n$ zur Basis $k$

Lukas Bach, lukas.bach@student.kit.edu



# Von Zeichen zu Zahlen zurück zu Zahlen

$11101_2$ 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:

Lukas Bach, lukas.bach@student.kit.edu



# Von Zeichen zu Zahlen zurück zu Zahlen

Karlsruher Institut für Technologie

$11101_2$ 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:

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

Lukas Bach, lukas.bach@student.kit.edu



# Von Zeichen zu Zahlen zurück zu Zahlen

$11101_2$ 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:

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

Achtung!

Lukas Bach, lukas.bach@student.kit.edu



# Von Zeichen zu Zahlen zurück zu Zahlen

$11101_2$ 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:

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

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

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Zum Beispiel:

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Zum Beispiel:

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

# Beispiel zu $Repr_k$

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

Zum Beispiel:

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

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Zum Beispiel:

$$\mathbf{Repr}_2(29)$$

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

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Zum Beispiel:

$$\mathbf{Repr}_2(29)$$

$$\begin{aligned}
&= \mathbf{Repr}_2(7) \cdot \mathbf{repr}_2(0) \cdot 1 \\
&= \mathbf{Repr}_2(7 \textbf{ div } 2) \cdot \mathbf{repr}_2(7 \textbf{ mod } 2) \cdot 01 \\
&= \mathbf{Repr}_2(3) \cdot \mathbf{repr}(1) \cdot 01 \\
&= \mathbf{Repr}_2(3 \textbf{ div } 2) \cdot \mathbf{repr}(3 \textbf{ mod } 2) \cdot 101 \\
&= \mathbf{Repr}_2(1) \cdot \mathbf{repr}(1) \cdot 101 \\
&= 11101
\end{aligned}$$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Zum Beispiel:

$$\mathbf{Repr}_2(29)$$

$$\begin{aligned}
&= \mathbf{Repr}_2(7 \;\mathbf{div}\; 2) \cdot \mathbf{repr}_2(7 \;\mathbf{mod}\; 2) \cdot 01 \\
&= \mathbf{Repr}_2(3) \cdot \mathbf{repr}(1) \cdot 01 \\
&= \mathbf{Repr}_2(3 \;\mathbf{div}\; 2) \cdot \mathbf{repr}(3 \;\mathbf{mod}\; 2) \cdot 101 \\
&= \mathbf{Repr}_2(1) \cdot \mathbf{repr}(1) \cdot 101 \\
&= 11101
\end{aligned}$$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Zum Beispiel:

$$\mathbf{Repr}_2(29)$$

$$\begin{aligned} &= \mathbf{Repr}_2(3) \cdot \mathbf{repr}(1) \cdot 01 \\ &= \mathbf{Repr}_2(3 \textbf{ div } 2) \cdot \mathbf{repr}(3 \textbf{ mod } 2) \cdot 101 \\ &= \mathbf{Repr}_2(1) \cdot \mathbf{repr}(1) \cdot 101 \\ &= 11101 \end{aligned}$$

# Beispiel zu $Repr_k$

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

Zum Beispiel:

$$\mathbf{Repr}_2(29)$$

$$= \mathbf{Repr}_2(3 \ \mathbf{div}\ 2) \cdot \mathbf{repr}(3 \ \mathbf{mod}\ 2) \cdot 101$$
$$= \mathbf{Repr}_2(1) \cdot \mathbf{repr}(1) \cdot 101$$
$$= 11101$$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu *Repr$_k$*

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

Zum Beispiel:

$$\mathbf{Repr}_2(29)$$

$$= \mathbf{Repr}_2(1) \cdot \mathbf{repr}(1) \cdot 101$$
$$= 11101$$

# Beispiel zu $Repr_k$

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

Zum Beispiel:

$$\mathbf{Repr}_2(29)$$

$$= 11101$$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $\mathit{Repr}_k$

Karlsruher Institut für Technologie

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

Zum Beispiel:

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

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Beispiel mit Hexadezimalzahlen:

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Beispiel mit Hexadezimalzahlen:

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

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Beispiel mit Hexadezimalzahlen:

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

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Beispiel mit Hexadezimalzahlen:

$$\mathbf{Repr}_{16}(29)$$

$$= 1 \cdot D = 1D$$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Beispiel mit Hexadezimalzahlen:

$$\mathbf{Repr}_{16}(29)$$

$$= 1D$$

Lukas Bach, lukas.bach@student.kit.edu



# Beispiel zu $Repr_k$

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

Beispiel mit Hexadezimalzahlen:

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

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu *Repr*$_k$

Karlsruher Institut für Technologie

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

## Übung zu *Repr*$_k$

Berechne die Repräsentationen folgender Zahlen in gegebenen Zahlensystemen:

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

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu $Repr_k$

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

## Übung zu $Repr_k$

Berechne die Repräsentationen folgender Zahlen in gegebenen Zahlensystemen:

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

Lösungen:

# Übung zu $Repr_k$

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

## Übung zu $Repr_k$

Berechne die Repräsentationen folgender Zahlen in gegebenen Zahlensystemen:

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

Lösungen:

- $\mathbf{Repr}_2(13)$

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu $Repr_k$

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

## Übung zu $Repr_k$

Berechne die Repräsentationen folgender Zahlen in gegebenen Zahlensystemen:

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

Lösungen:

- $\mathbf{Repr}_2(13) = 1101$.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu $Repr_k$

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

## Übung zu $Repr_k$

Berechne die Repräsentationen folgender Zahlen in gegebenen Zahlensystemen:

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

Lösungen:

- $\mathbf{Repr}_2(13) = 1101$.
- $\mathbf{Repr}_4(15)$

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu $Repr_k$

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

## Übung zu $Repr_k$

Berechne die Repräsentationen folgender Zahlen in gegebenen Zahlensystemen:

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

Lösungen:

- $\mathbf{Repr}_2(13) = 1101$.
- $\mathbf{Repr}_4(15) = 33$.

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu $Repr_k$

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

## Übung zu $Repr_k$

Berechne die Repräsentationen folgender Zahlen in gegebenen Zahlensystemen:

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

Lösungen:

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

Lukas Bach, lukas.bach@student.kit.edu



# Übung zu $Repr_k$

Karlsruher Institut für Technologie

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

## Übung zu $Repr_k$

Berechne die Repräsentationen folgender Zahlen in gegebenen Zahlensystemen:

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

Lösungen:

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

Lukas Bach, lukas.bach@student.kit.edu



# Feste Länge von Binärzahlen

Lukas Bach, lukas.bach@student.kit.edu



# Feste Länge von Binärzahlen

Karlsruher Institut für Technologie

## $bin_\ell$

Die Funktion $\mathbf{bin}_\ell : \mathbb{Z}_{2^\ell} \to \{0, 1\}^\ell$

Lukas Bach, lukas.bach@student.kit.edu

# Feste Länge von Binärzahlen



*$bin_\ell$*

Die Funktion $\mathbf{bin}_\ell : \mathbb{Z}_{2^\ell} \to \{0, 1\}^\ell$ bringt eine gegebene Binärzahl auf eine feste Länge

# Feste Länge von Binärzahlen

Lukas Bach, lukas.bach@student.kit.edu



## *bin*$_\ell$

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

Lukas Bach, lukas.bach@student.kit.edu

# Feste Länge von Binärzahlen



### *bin*$_\ell$

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

Lukas Bach, lukas.bach@student.kit.edu

# Feste Länge von Binärzahlen





## *bin*$_\ell$

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

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

Lukas Bach, lukas.bach@student.kit.edu



# Feste Länge von Binärzahlen

*$bin_\ell$*

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

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

Beispiel:

Lukas Bach, lukas.bach@student.kit.edu



# Feste Länge von Binärzahlen

Karlsruher Institut für Technologie

## $bin_\ell$

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

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

Beispiel:

- $\mathbf{bin}_8(3)$

Lukas Bach, lukas.bach@student.kit.edu



# Feste Länge von Binärzahlen

Karlsruher Institut für Technologie

## *bin*$_\ell$

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

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

Beispiel:

- $\mathbf{bin}_8(3) = 0^{8-|\mathbf{Repr}_2(3)|}\mathbf{Repr}_2(3)$

Lukas Bach, lukas.bach@student.kit.edu



# Feste Länge von Binärzahlen

**$bin_\ell$**

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

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

Beispiel:

- $\mathbf{bin}_8(3) = 0^{8-|\mathbf{Repr}_2(3)|}\mathbf{Repr}_2(3) = 0^{8-|11|} \cdot 11$

Lukas Bach, lukas.bach@student.kit.edu



# Feste Länge von Binärzahlen

*$bin_\ell$*

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

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

Beispiel:

- $\mathbf{bin}_8(3) = 0^{8-|\mathbf{Repr}_2(3)|}\mathbf{Repr}_2(3) = 0^{8-|11|} \cdot 11 = 0^{8-2} \cdot 11$

Lukas Bach, lukas.bach@student.kit.edu



# Feste Länge von Binärzahlen

Karlsruher Institut für Technologie

## *bin*$_\ell$

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

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

Beispiel:

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

Lukas Bach, lukas.bach@student.kit.edu



# Feste Länge von Binärzahlen

Karlsruher Institut für Technologie

*bin*$_\ell$

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

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

Beispiel:

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

Lukas Bach, lukas.bach@student.kit.edu



# Feste Länge von Binärzahlen



*$bin_\ell$*

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

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

Beispiel:

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

Lukas Bach, lukas.bach@student.kit.edu

# Zweierkomplement

Lukas Bach, lukas.bach@student.kit.edu



# Zweierkomplement

Was ist mit negative Zahlen?

Lukas Bach, lukas.bach@student.kit.edu



# Zweierkomplement

Was ist mit negative Zahlen?

- Idee: Verwende das erste Bit, um zu speichern, ob die Zahl positiv oder negativ ist.

Lukas Bach, lukas.bach@student.kit.edu



# Zweierkomplement

Karlsruher Institut für Technologie

Was ist mit negative Zahlen?

- Idee: Verwende das erste Bit, um zu speichern, ob die Zahl positiv oder negativ ist.
- Beispiel:

Lukas Bach, lukas.bach@student.kit.edu



# Zweierkomplement



Was ist mit negative Zahlen?

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

Lukas Bach, lukas.bach@student.kit.edu



# Zweierkomplement

Karlsruher Institut für Technologie

Was ist mit negative Zahlen?

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

Lukas Bach, lukas.bach@student.kit.edu



# Zweierkomplement

Karlsruher Institut für Technologie

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 $x$

Lukas Bach, lukas.bach@student.kit.edu



# Zweierkomplement

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:

Lukas Bach, lukas.bach@student.kit.edu



# Zweierkomplement

Karlsruher Institut für Technologie

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:

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

Lukas Bach, lukas.bach@student.kit.edu



# Zweierkomplement

Karlsruher Institut für Technologie

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 $\ell - 1$?

# Aufgaben zu 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?

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

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu 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?

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

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu 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?

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

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu 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?

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

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu 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?

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

Lukas Bach, lukas.bach@student.kit.edu



# Aufgaben zu 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?

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


That's all Folks!