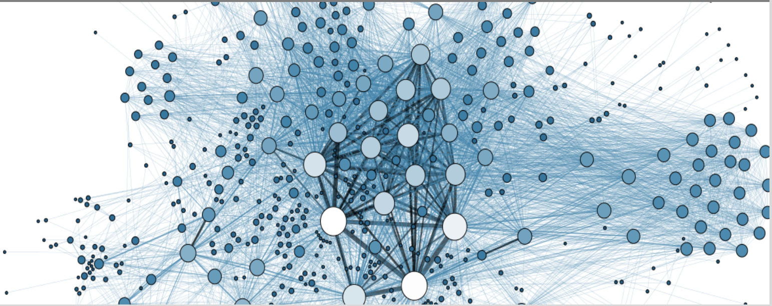
Grundbegriffe der Informatik
Tutorium 33

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



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

Gliederung

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

Termine

- Vorlesung und Übung
  - Mittwoch 9:45 - 11:15 Vorlesung
  - Freitag 9:45 - 11:15 abwechselnd Vorlesung und Übung
- Tutorium
  - Donnerstags, 14:00 - 15:30
  - 50.34 Informatikbau, -107
- Übungsblätter
  - Alle zwei Wochen
  - Ausgabe Mittwochs, Abgabe Donnerstags bis 16:00 zwei Wochen drauf

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

Übungsschein

- min. 50% aller Punkte auf Übungsblättern richtig
- Rückgabe im Tutorium
- Bestehen ist *keine* Voraussetzung für die Klausur, *aber* fürs Modul!
- Gemeinsames Abgeben, Abschreiben verboten
- Übungsblätter und später auch Musterlösungen im ILIAS

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

Tutorium

- Alle Tutorienfolien auf:

  `http://gbi.lukasbach.com`

- Bei Fragen: lukas.bach@student.kit.edu
- Keine Anwesenheitspflicht
- Möglichkeit andere Tutorien zu besuchen

Signale und Nachrichten

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

- Objekt: 101
  - Eins null eins oder 101 als Zahl oder 5 in binär oder zwei merkwürdige Striche mit einem Kreis dazwischen?
  - Vom Kontext abhängig.
  - Zunächst einfach ein konkretes Objekt.

Signale und Nachrichten

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

Karlsruher Institut für Technologie

- Signal
  - Physikalische Veränderung
  - Lässt sich verschieden interpretieren.
  - Beispiele:
    - Notfallalarm in Serverraum

    - Für Besucher nur schönes Leuchten
    - Für Security die Information, zu kommen
    - Für Techniker die Information, Ausrüstung zu holen
- Nachricht: Objekt wie oben, das von Signal unabhängig ist
  - Roter Notfallalarm ist ein anderes Signal als ein blauer Notfallalarm, aber vielleicht dieselbe Nachricht.

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

Signale und Nachrichten

- Der interessante Teil: Informationen
- Bedeutung einer Nachricht
- Der vorher fehlende Kontext.
- Im obigen Beispiel:
  - Rote Alarmleuchte ist ein Signal (blaue Signalleuchte in Raum nebendran vielleicht auch)
  - “Alarm”: Nachricht
  - Information: Security soll herkommen, Techniker sollen das Werkzeug bereit halten, Besucher sollten Platz machen.

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

Mengen

- Erster wirklich wichtiger Teil.

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

Mengen

Zeichnung

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

Mengen

Definition: Mengen

"Unter einer Menge verstehen wir jede Zusammenfassung von bestimmten wohlunterschiedenen Objekten unserer Anschauung oder unseres Denkens (welche die Elemente dieser Menge genannt werden) zu einem Ganzen."

- Beispiel: $\{a, b, c, d\} =: A$ $\{a, c, 4\} =: B$, $\{10, 11\} =: C$
- Das Objekt c ist in A enthalten: $c \in A$, $c \in B$, $c \notin C$
- Reihenfolge gleich: $\{a, b\} = \{b, a\}$
- Elemente doppelt? $\{a, a, b, a\} = \{a, b\}$

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

Mehr über Mengen

- Kardinalität oder Größe: Die Anzahl der Elemente der Menge
  - $A := \{a, b, c\}$. $|A| = 3$
  - $B := \{c, d\}$. $|B| = 2$
  - Was ist $|\{1, 2, 3, 2\}|$? 3!
  - Was ist $|\{\}|$? 0

Leere Menge

Die Menge, die nichts enthält, nennen wir die leere Menge, und schreiben sie als $\{\}$ oder $\emptyset$.

Was ist $|\{\{\}\}|$? 1! $\{\emptyset\}$ enthält eine leere Menge, die selbst ein Element ist.

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

Mehr über Mengen

Zeichnung

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

Mehr über Mengen

Karlsruher Institut für Technologie

Seien $A := \{a, b, c\}$, $B := \{b, c\}$, $C := \{c, b\}$, $D := \{b, c, d\}$.

- Teilmenge: $A \subseteq B$, also A ist Teilmenge von B genau dann, wenn alle Elemente aus A auch in B sind.
- Echte Teilmenge: $A \subset B$ genau dann, wenn $A \subseteq B$ *und* $A \neq B$.
  - Beispiele: $B \subseteq A$, sogar $B \subset A$.
    $C \subseteq B$ und $B \subseteq C$, aber $C \not\subset B$ und $B \not\subset C$.
- Schnittmenge: $A \cap B = \{b, c\}$.
  $A \cap B$ enthält *genau* die Elemente, die in A *und* in B sind.
- Vereinigungsmenge: $A \cup D = \{a, b, c, d\}$.
  $A \cup B$ enthält *genau* die Elemente, die in A *oder* in B sind.
- Mengendifferenz: $A \setminus B = \{a\}$, also alle Elemente in A, die nicht in B sind.
- Komplementärmenge: $\bar{A}$ enthält alle Elemente des *Universums*, die nicht in A sind. Angenommen, Universum = Lateinisches Alphabet: $\bar{A} = \{d, e, f, g, \ldots, y, z\}$

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

Potenzmenge

Potenzmenge

Die Potenzmenge 2^M einer Menge M enthält genau alle Mengen, die Teilmenge von M sind.

Was bedeutet das allgemein?

- $M \in 2^M$
- $\emptyset \in 2^M$
- Konkretes Beispiel: Was ist 2^M mit $M = \{0, 1\}$?
  - Natürlich $\emptyset \in 2^M$ und $\{0, 1\} \in 2^M$.
  - $\{0\} \in 2^M$ und $\{1\} \in 2^M$.
  - Weitere? Nein, diese vier Mengen sind alle möglichen Teilmengen.
  - $\Rightarrow 2^M = \{\{\}, \{0\}, \{1\}, \{0, 1\}\}$.

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

Potenzmenge

$M = \{0, 1\}$, $2^M = \{\{\}, \{0\}, \{1\}, \{0, 1\}\}$.

Was ist 2^{2^M}?

- Also $2^{\{\{\},\{0\},\{1\},\{0,1\}\}}$.
- Natürlich $\emptyset \in 2^M$ und $2^M = \{\{\}, \{0\}, \{1\}, \{0, 1\}\} \in 2^{2^M}$.

$$
\begin{aligned}
2^{2^M} = \{ & \\
& \{\}, \\
& \{\{\}\}, \{\{0\}\}, \{\{1\}\}, \{\{0, 1\}\}, \\
& \{\{\}, \{0\}\}, \{\{\}, \{1\}\}, \{\{\}, \{0, 1\}\}, \{\{0\}, \{1\}\}, \\
& \quad \{\{0\}, \{0, 1\}\}, \{\{1\}, \{0, 1\}, \\
& \{\{\}, \{0\}, \{1\}\}, \{\{\}, \{0\}, \{0, 1\}\}, \{\{\}, \{1\}, \{0, 1\}\}, \{\{0\}, \{1\}, \{0, 1\}\}, \\
& \{\{\}, \{0\}, \{1\}, \{0, 1\}\} \\
\} &
\end{aligned}
$$

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

Alphabete

Alphabet

Ein Alphabet ist eine *endliche*, *nichtleere* Menge von Zeichen.

Was davon sind Alphabete? $\{d, 34, \pi, \%\}$, $\{a, b, c, \ldots, y, z\}$, $\emptyset$, $\mathbb{N}$.

- $\{d, 34, \pi, \%\}$ und $\{a, b, c, \ldots, y, z\}$ sind Alphabete.
- $\emptyset$ ist leer und damit kein Alphabet.
- $\mathbb{N} = \{1, 2, 3, \ldots\}$ enthält alle natürlichen Zahlen und ist damit nicht endlich, also kein Alphabet.
- $\{0, 1\}$ ist das Alphabet, das alle Binärzahlen enthält.
- $\{\cdot, +, -, /\} =: R$ ist ein Alphabet von Rechenzeichen. $R \cup \{0, 1, \ldots, 9\}$ ist ein Alphabet, das ein Taschenrechner als Eingabealphabet benutzen könnte.

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

Paare und Tupel

Paar

Ein Paar ist eine geordnete Menge der Kardinalität 2.

Schreibweise mit runden Klammern ().

- Beispiel: $(a, 4) \neq (4, a)$
- Beispiel für eine Menge aus Tupeln: $\{(\text{“}\mathit{AgeOfEmpires}'', \text{“}\mathit{Strategie}''), (\text{“}\mathit{Battlefield}'', \text{“}\mathit{Shooter}''), (\text{“}\mathit{SeriousSam}'', \text{“‘}\mathit{Shooter}''))\}$

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

Tupel

Tupel

Ein Tupel ist eine geordnete Menge. Konkret ist ein n-Tupel ein Tupel der Kardinalität n.

Also wie ein Paar, nur mit beliebiger Kardinalität. Ein Paar ist spezifisch ein 2-Tupel.
Beispiel: $(4tb, 512gb, 128gb, 4mb) \neq (512gb, 4mb, 4tb, 128gb)$.

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

Kartesisches Produkt

Karlsruher Institut für Technologie

Zwei Mengen: $A := \{a, b, c\}$ und $B := \{1, 2, 3\}$.
Wir wollen alle Tupel mit erstem Element aus A und zweiten Element aus B.
$\{ \quad (a, 1), (a, 2), (a, 3), \qquad (b, 1), (b, 2), (b, 3), \qquad (c, 1), (c, 2), (c, 3) \quad \}$
$= A \times B$

Kreuzprodukt von zwei Mengen

Zu zwei Mengen A und B ist das Kreuzprodukt definiert als Menge aller Paare (a, b) mit $a \in A$ und $b \in B$.

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

Kreuzprodukt

Karlsruher Institut für Technologie

Kreuzprodukt von zwei Mengen

Zu zwei Mengen A und B ist das Kreuzprodukt $A \times B$ definiert als Menge aller Paare (a, b) mit $a \in A$ und $b \in B$.

Kreuzprodukt von n Mengen

Zu n Mengen $M_1, M_2, \dots, M_n$ ist das Kreuzprodukt $M_1 \times M_2 \times \dots \times M_n$ definiert als Menge aller n-Tupel $(e_1, e_2, \dots, e_n)$ mit $e_1 \in M_1$, $e_2 \in M_2$, …, $e_n \in M_n$.

Mengenpotenz

$$\underbrace{A \times A \times \dots \times A}_{n \times mal} = A^n.$$

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

Kreuzprodukt: Beispiele

Karlsruher Institut für Technologie

Kreuzprodukt von zwei Mengen

Zu zwei Mengen A und B ist das Kreuzprodukt $A \times B$ definiert als Menge aller Paare (a, b) mit $a \in A$ und $b \in B$.

$A := \{a, b\}, B := \{1, 2\}$. $A \times B = \{(a, 1), (a, 2), (b, 1), (b, 2)\}$.

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

Kreuzprodukt: Beispiele

Kreuzprodukt von n Mengen

Zu n Mengen $M_1, M_2, \ldots, M_n$ ist das Kreuzprodukt $M_1 \times M_2 \times \cdots \times M_n$ definiert als Menge aller n-Tupel $(e_1, e_2, \ldots, e_n)$ mit $e_1 \in M_1, e_2 \in M_2, \ldots, e_n \in M_n$.

$A := \{a, b\}, B := \{1, 2\}, C := \{\omega\}$. $A \times B \times C$
$= \{(a, 1, \omega), (a, 2, \omega), (b, 1, \omega), (b, 2, \omega)\}$.

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

Kreuzprodukt: Beispiele

Mengenpotenz

$$\underbrace{A \times A \times \cdots \times A}_{nmal} = A^n.$$

- $A := \{a, b\}$. $A^2 = \{(a, b), (b, a), (a, a), (b, b)\}$
  $A^3 = \{(a, a, a), (a, a, b), (a, b, b), \ldots\}$.
- A beliebige Menge. $A^0? = \emptyset$
- Achtung! $2^M \neq M^2$. Potenzmengen nicht mit Mengenpotenz verwechseln!

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

Relation

Binäre Relation

Eine binäre Relation auf zwei Mengen A und B ist eine Menge $R \subseteq A \times B$.

- Für die Mengen $M_{Spiele} = \{''Battlefield'', ''AgeOfEmpires'', ''SeriousSam''\}$, $M_{Genre} = \{''Shooter'', ''Strategie''\}$ sind folgendes mögliche Relationen:
  - $\{(''AgeOfEmpires'', ''Strategie''), (''Battlefield'', ''Shooter''), (''SeriousSam'', '''Shooter'')\}$
  - $\{(''AgeOfEmpires'', ''Strategie''), (''AgeOfEmpires'', ''Shooter'')$
  - $\emptyset$
- "Kleinergleichrelation" auf $M = \{1, 2, 3\}$: $R_{\leq} = \{(1,1), (1,2), (1,3), (2,2), (2,3), (3,3)\} \in M \times M$

Relation

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

Binäre Relation

Eine binäre Relation auf zwei Mengen A und B ist eine Menge $R \subseteq A \times B$.

Ternäre Relation

Eine ternäre Relation auf drei Mengen A, B und C ist eine Menge $R \subseteq A \times B \times C$.

n-äre Relation

Eine n-äre Relation auf n Mengen M_1, M_2 ... M_n ist eine Menge $R \subseteq M_1 \times M_2 \times \cdots \times M_n$.

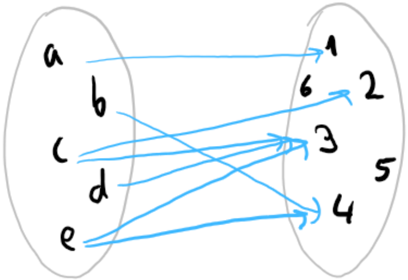
Lukas Bach, lukas.bach@student.kit.edu

Linkstotalität

Linkstotale Relation

Eine Relation $R \subseteq A \times B$ heißt linkstotal, wenn für jedes $a \in A$ ein $b \in B$ existiert mit $(a, b) \in R$.

Die linke Seite der Relation ist also “total” aufgefüllt.



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

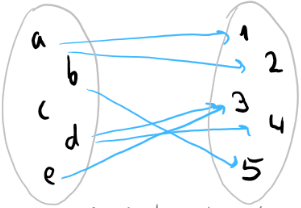
Rechtstotalität

Rechtstotale Relation

Eine Relation $R \subseteq A \times B$ heißt rechtstotal, wenn für jedes $b \in B$ ein $a \in A$ existiert mit $(a, b) \in R$.

Die rechte Seite der Relation ist also "total" aufgefüllt.
Wenn die Relation zusätzlich eine Abbildung ist, heißt diese dann surjektiv.

Rechtstotalität



„jedes b hat min. einen Partner a"

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

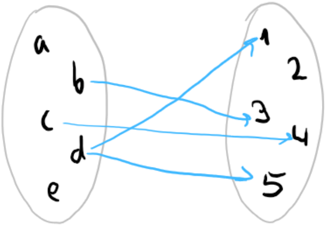
Linkseindeutigkeit

Karlsruher Institut für Technologie

Linkseindeutige Relation

Eine Relation $R \subseteq A \times B$ heißt linkseindeutig, wenn für zwei beliebige Elemente $(a, \alpha) \in R, (b, \beta) \in R$ aus der Relation R gilt: wenn $a \neq b$, dann gilt auch $\alpha \neq \beta$.

Also: Keine zwei Elemente der linken Seite der Relation haben dasselbe rechte Element.
Angenommen, $a \neq b$ und $\alpha = \beta$. $\Rightarrow$ offenbar nicht linkseindeutig.
Wenn die Relation zusätzlich eine Abbildung ist, heißt diese dann injektiv.



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

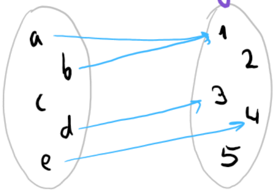
Rechtseindeutig

Rechtseindeutige Relation

Eine Relation $R \subseteq A \times B$ heißt rechtseindeutig, wenn für zwei beliebige Elemente $(a, \alpha) \in R, (b, \beta) \in R$ aus der Relation R gilt: wenn $\alpha \neq \beta$, dann gilt auch $a \neq b$.

Also: Keine zwei Elemente der rechten Seite der Relation haben dasselbe linke Element.

Rechtseindeutigkeit

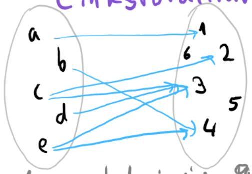


„jedes a hat max einen Partner b"

Eigenschaften von Relationen

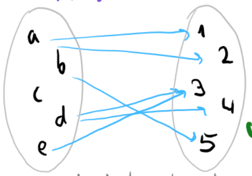
Lukas Bach, lukas.bach@student.kit.edu

Linkstotalität



„jedes a hat min einen Partner b“

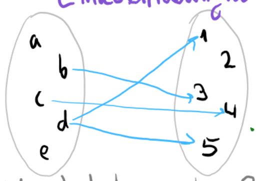
Rechtstotalität



Surjektiv

„jedes b hat min. einen Partner a“

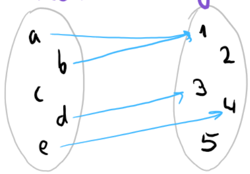
Linkseindeutigkeit



injektiv

„jedes b hat max einen Partner a“

Rechtseindeutigkeit



„jedes a hat max einen Partner b“

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

Abbildung

Abbildung

Eine Relation R heißt eine Abbildung, wenn sie linkstotal *und* rechtseindeutig sind.

- Injektive Funktion: linkstotal, rechtseindeutig, linkseindeutig
- Surjektive Funktion: linkstotal, rechtseindeutig, rechtstotal

Bijektivität

Eine Relation heißt bijektiv, wenn sie injektiv und surjektiv ist.

Damit ist sie linkstotal und rechtseindeutig (weil es eine Abbildung ist) und linkseindeutig (injektiv) und rechtstotal (surjektiv).
Tolle Eigenschaft: Für jedes Element $(a, b) \in R$ der bijektiven Relation R ist *jedem a genau ein b* zugeordnet.

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

Abbildungen Schreibweise

Seien $A = B = \mathbb{R}$, $f \subseteq A \times B$. Wir suchen Relation, die für jedes $a \in A$ ein Element $(a, b) \in f$ enthält mit $b = a^2$.
$f = \{(0, 0), (0.1, 0.01), (2, 4), \dots\}$
Unendlich viele Elemente, und unmöglich alle zu nennen.
(Mathematischere) Schreibweise für Abbildungen:
$f : A \to B, a \mapsto a^2$, also Quadratfunktion.
Ist diese Funktion injektiv oder surjektiv?

- Nicht injektiv, da z.B. $f(1) = f(-1)$, also $(1, 1) \in f$ und $(-1, 1) \in f$.
- Nicht surjektiv, da z.B. -1 nie als Funktionswert angenommen wird, daher $(a, -1) \notin f$ für beliebige $a \in A$.

Wiederholung

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

$A := \{a, b, c\}, B := \{b, c, d\}, C := \{a, d\}$

- $A \cap B = \{b, c\}$
- $A \cup B = \{a, b, c, d\}$
- $A \backslash B = \{a\}$
- $C^2 = C \times C = \{(a, a), (a, d), (d, a), (d, d)\}$
- $2^C = \{\emptyset, \{a, d\}, \{a\}, \{d\}\}$
- Unterschied zwischen $\{a, b\}$ und (a, b)?
- Definition von...
  - Alphabet?
  - Abbildung?

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

Wörter

Karlsruher Institut für Technologie

Konkatenation

Durch Konkatenation werden einzelne Buchstaben aus einem Alphabet miteinander verbunden.

- Symbol: $\cdot$, also zwei Buchstaben a und b miteinander konkateniert: $a \cdot b$.
- Nicht kommutativ: $a \cdot b \neq b \cdot a$
- Aber assoziativ: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
- Kurzschreibweise: Ohne Punkte, also $a \cdot b = ab$

Wörter

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

Wörter: Intuitivere Definition

Ein Wort w entsteht durch die Konkatenation durch Buchstaben aus einem Alphabet.

Also Abfolge von Zeichen.
Sei $A := \{a, b, c\}$.

- Mögliche Worte: $w_1 := a \cdot b$, $w_2 = b \cdot c \cdot c$, $w_3 = a \cdot c \cdot c \cdot b \cdot a$.
- Keine möglichen Worte: d.
- Konkatenation nicht kommutativ: Wort *abc* ist ungleich dem Wort *bca*.

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

Wörter

Wörter: Abstraktere Definition

Ein Wort w über dem Alphabet A ist definiert als surjektive Abbildung $w : \mathbb{Z}_n \to A$. Dabei heißt n die Länge $|w|$ des Wortes.

- $\mathbb{Z}_n = \{i \in \mathbb{N} : 0 \leq i < n\}$
  $\mathbb{Z}_3 = \{0, 1, 2\}, \mathbb{Z}_2 = \{0, 1\}, \mathbb{Z}_0 = \emptyset$.
- Länge oder Kardinalität eines Wortes: $|w|$. $|abcde| = 5$.
- Wort $w = abdec$ als Relation aufgeschrieben:
  $w = \{(0, a), (1, b), (2, d), (3, e), (4, c)\}$. Also
  $w(0) = a, w(1) = b, w(2) = d, \ldots$
  Damit sieht man auch:
  $|w| = |\{(0, a), (1, b), (2, d), (3, e), (4, c)\}| = 5$.

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

Wörter

Karlsruher Institut für Technologie

- Wort der Kardinalität 0?

Das leere Wort

Das leere Wort ε ist definiert ein Wort mit Kardinalität 0, also mit 0 Zeichen.

- Leere Wort wird interpretiert als “nicht sichtbar” und kann überall platziert werden: $aabc = a\varepsilon abc = \varepsilon\varepsilon a\varepsilon bc\varepsilon$.
- $|\{\varepsilon\}| = 1$, die Menge ist nicht leer! Das leere Wort ist nicht *nichts*! (Vergleiche leere Menge)
- $|\varepsilon| = 0$.

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

Mehr über Wörter

A^n

Zu einem Alphabet A ist A^n definiert als die Menge aller Wörter der Länge n über dem Alphabet A.

- Nicht mit Mengenpotenz verwechseln!
- $A := \{a, b, c\}$, $A^2 = \{aa, ab, ac, ba, bb, bc, ca, cb, cc\}$. $A^1 = A$, $A^0 = \{\varepsilon\}$.

Die Menge aller Wörter *beliebiger* Länge:

- $A^* := \bigcup_{i \in \mathbb{N}_0} A_i$
- $A := \{a, b, c\}$. $aa \in A^*$, $abcabcabc \in A^*$, $aaaa \in A^*$, $\varepsilon \in A^*$.

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

Mehr über Wörter

Konkatenation von Wörtern:

- $lager \cdot regal = lagerregal$
- $lag \cdot erregal = lagerregal$

Konkatenation von Wörtern.

$$w_1 \cdot w_2 : \mathbb{Z}_{m+n} \to A_1 \cup A_2$$
$$i \mapsto \begin{cases} w_1(i) & \text{falls } 0 \leq i < m \\ w_2(i-m) & \text{falls } m \leq i < m+n \end{cases}$$

- Warum $\mathbb{Z}_{m+n}$? Wörter w_1 und w_2 mit $|w_1| = m$ und $|w_2| = n$ werden konkateniert, also neues Wort hat Länge $m + n$.

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

Mehr über Wörter

Konkatenation von Wörtern.

$$w_1 \cdot w_2 : \mathbb{Z}_{m+n} \to A_1 \cup A_2$$

$$i \mapsto \begin{cases} w_1(i) & \text{falls } 0 \leq i < m \\ w_2(i-m) & \text{falls } m \leq i < m+n \end{cases}$$

$|w_1| = 5 := m$ $\quad$ $|w_2| = 8 := n$

$w_1(1)$ $w_1(2)$ $w_1(3)$ $w_1(4)$ $w_1(5)$ $\quad$ $w_2(1)$ $w_2(2)$ $w_2(3)$ $w_2(4)$ $w_2(5)$ $w_2(6)$ $w_2(7)$ $w_2(8)$

grund · begriffe

Wert von i: 0 1 2 3 4 $\quad$ 5 6 7 8 9 10 11 12

Fall 1: $0 \leq i < m = 5$

Fall 2: $m \leq i < m+n = 13$

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

Mehr über Wörter

- Immernoch: Reihenfolge ist wichtig!
  $OTT \cdot O = OTTO \neq OOTT = O \cdot OTT$
- Auf wieviele Weisen kann man *abc* als Konkatenation nichtleerer Wörter schreiben? abc, $a \cdot bc$, $ab \cdot c$, $a \cdot b \cdot c$.
- Wortkonkatenation mit dem leeren Wort: $w \cdot \varepsilon = w = \varepsilon \cdot w$.

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

Mehr über Wörter

Wort Potenzen

Sich direkt wiederholende Teilworte kann man als Wortpotenz darstellen, daher $w_i^n = w_i \cdot w_i \cdots w_i$ (n $\times$ mal).

- $a^4 = aaaa$, $b^3 = bbb$, $c^0 = \varepsilon$, $d^1 = d$.
- $a^3c^2b^6 = aaaccbbbbbb$.
- $b \cdot a \cdot (n \cdot a)^2 = banana$.
- $(a^3b^2)^2c(a^2bcb^3)^3dd = (aaabb)^2c(aabcbbb)^3dd$
  $= aaabb \cdot aaabb \cdot \quad c \quad \cdot aabcbbb \cdot aabcbbb \cdot aabcbbb \cdot \quad dd.$

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

Übung zu Wörter

Sei A ein Alphabet.

Übung zu Wörter

1. Finde Abbildung $f : A^* \to A^*$, sodass für alle $w \in A^*$ gilt:
   $2 \cdot |w| = |f(w)|$.
2. Finde Abbildung $g : A^* \to A^*$, sodass für alle $w \in A^*$ gilt:
   $|w| + 1 = |g(w)|$.
3. Finde Abbildung $h : A^* \to A^*$, sodass für alle $w \in A^*$ gilt:
   $\lfloor \frac{|w|}{2} \rfloor = |h(w)|$. (Zusatz)
4. Sind f, g, h injektiv und/oder surjektiv?

1. $f : A^* \to A^*, w \mapsto w \cdot w$.
2. $g : A^* \to A^*, w \mapsto w \cdot x, x \in A$.
3. $h : A^* \to A^*, w \mapsto \hat{w}$ mit $\hat{w}_i = \left\{ \begin{array}{ll} w_i & \text{wenn } i \leq \lfloor \frac{|w|}{2} \rfloor \\ \varepsilon & \text{sonst} \end{array} \right\}$ und $i \in \mathbb{Z}_{|w|}$.

Übung zu Wörter

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

1. $f : A^* \to A^*, w \mapsto w \cdot w$.
   - f ist injektiv, denn jedes w aus der Bildmenge wird von maximal einem Wort abgebildet.
   - f ist nicht surjektiv, denn z.B. bildet nichts auf $x \in A$ ab (oder auf andere Wörter mit ungerader Anzahl an Buchstaben).
2. $g : A^* \to A^*, w \mapsto w \cdot x, x \in A$.
   - g ist injektiv.
   - g ist nicht surjektiv, denn z.B. bildet nichts auf ε ab.
3. $h : A^* \to A^*, w \mapsto \hat{w}$ mit $\hat{w}_i = \left\{ \begin{array}{ll} w_i & \text{wenn } i \leq \lfloor \frac{|w|}{2} \rfloor \\ \varepsilon & \text{sonst} \end{array} \right\}$ und $i \in \mathbb{Z}_{|w|}$.
   - h ist nicht injektiv, denn z.B. $x = h(xy) = h(xz)$ mit $x, y, z \in A$.
   - h ist surjektiv, denn für jedes $w \in A^*$ existiert ein $\hat{w} \in A^*$ mit $\hat{w} = w \cdot w$ sodass $h(\hat{w}) = w$.

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

Formale Sprache

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

Formale Sprache

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

- Zufälliges Beispiel: $A := \{b, n, a\}$.
  - $L_1 := \{ban, baan, nba, aa\}$ 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?
    $L_3 = \{w : w = ba^k n, k \in \mathbb{N}\}$
- Formale Sprachen sind also nicht zwangsweise endliche Mengen.
- Praktischeres Beispiel: $A := \{w : w$ ist ein ASCII Symbol $\}$.
  - $L_4 := \{class, if, else, while, for, \ldots\}$ ist eine formale Sprache über A.
  - $L_5 := \{w : w = a \cdot b$ mit a als Großbuchstabe und b als Groß- oder Kleinbuchstabe $\} \setminus L_4$ ist eine formale Sprache von korrekten Klassennamen in Java.

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

Übung zu formalen Sprachen

Karlsruher Institut für Technologie

$A := \{a, b\}$

- Sprache L aller Wörter über A, die nicht das Teilwort ab enthalten?
  - Was passiert wenn ein solches Wort ein a enthält? Dann keine b's mehr!
  - $L = \{w_1 \cdot w_2 : w_1 \in \{b\}^* \text{ und } w_2 \in \{a\}^*\}$
  - Andere Möglichkeit: Suche Wörter mit ab und nehme diese Weg.
  - $L = \{a, b\}^* \setminus \{w_1 \cdot ab \cdot w_2 : w_1, w_2 \in \{a, b\}^*\}$

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

Übung zu formalen Sprachen

Sei $A := \{a, b\}, B := \{0, 1\}$.

Aufgabe zu formalen Sprachen

1. Sprache $L_1 \subseteq A^*$ von Wörtern, die mindestens drei b's enthalten.
2. Sprache $L_2 \subseteq A^*$ von Wörtern, die gerade Zahl von a's enthält.
3. Sprache $L_3 \subseteq B^*$ von Wörtern, die, interpretiert als Binärzahl eine gerade Zahl sind.

1. $L_1 = \{w = w_1 b w_2 b w_3 b w_4 : w_1, w_2, w_3, w_4 \in A^*\}$
2. $L_2 = \{w = (w_1 a w_2 a w_3)^* : w_1, w_2, w_3 \in \{b\}^*\}$ (Ist da ε drin?)
3. $L_3 = \{w = w \cdot 0 : w \in B^*\}$

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

Aussagenlogik

- Das wars erst mal zu formalen Sprachen.
- Heute ist Freitag.
- Die Menge aller Männer dieser Welt ist disjunkt zur Menge aller Frauen dieser Welt.

Das sind alles Aussagen. Aussagen sind entweder *wahr* oder *falsch*.

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

Aussagenlogik

Wir kapseln Aussagen und verwendet Variablen dafür.
Zum Beispiel:

- $A :=$ "Die Straße ist nass."
- $B :=$ "Es regnet."

Aussagen lassen sich verknüpfen:

- Logisches Und: $A \wedge B = A$ und $B =$ Die Straße ist nass und es regnet.
- Logisches Oder: $A \vee B = A$ oder $B =$ Die Straße ist nass oder es regnet. Es kann auch beides wahr sein.
- Negierung: $\neg A =$ nicht $A =$ Die Straße ist nicht nass.
- Implikation: $A \rightarrow B =$ Aus A folgt $B =$ Wenn die Straße nass ist, dann regnet es.
- Äquivalenz: $A \leftrightarrow B = A$ und B sind äquivalent $=$ Die Straße ist *genau dann* nass, *wenn* es regnet.
  - $A \leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A)$, also die Straße ist nass wenn es regnet *und* es regnet wenn die Straße nass ist.

Übung zu Aussagenlogik

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

- $A :=$ “Die Straße ist nass.”
- $B :=$ “Es regnet.”
- $C :=$ “π ist gleich 3.”

- Was ist $B \rightarrow C$? “Wenn es regnet, ist π gleich 3.”

x_1	x_2	$\neg x_1$	$x_1 \wedge x_2$	$x_1 \vee x_2$	$x_1 \rightarrow x_2$
f	f	w	f	f	w
f	w	w	f	w	w
w	f	f	f	w	f
w	w	f	w	w	w

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

Syntax der Aussagenlogik

Menge der Aussagevariablen:

$Var_{AL} \subseteq \{P_i : i \in \mathbb{N}_0\}$ oder $\{P, Q, R, S, \ldots\}$

Alphabet der Aussagenlogik:

$A_{AL} = \{(,), \neg, \wedge, \vee, \rightarrow, \leftrightarrow\} \cup Var_{AL}$

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

Boolesche Funktionen

Karlsruher Institut für Technologie

Boolesche Funktionen

Eine boolsche Funktion ist eine Abbildung der Form $f : \mathbb{B}^n \to \mathbb{B}$ mit $\mathbb{B} = \{w, f\}$.

Typische Boolsche Funktionen: $b_\neg(x) = \neg x$, $b_\vee(x_1, x_2) = x_1 \vee x_2 \ldots$

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

Interpretationen

Karlsruher Institut für Technologie

Interpretation

Eine Interpretation ist eine Abbildung $I : V \to \mathbb{B}$, die einer Variablenmenge eine "Interpretation", also wahr oder falsch zuordnet.

Weiter legt man $val_I(F)$ als Auswertung einer aussagenlogischer Formel F fest.

$$\begin{aligned}
val_I(X) &= I(X) \\
val_I(\neg G) &= b_\neg(val_I(G)) \\
val_I(G \wedge H) &= b_\wedge(val_I(G), val_I(H)) \\
val_I(G \vee H) &= b_\vee(val_I(G) val_I(H)) \\
val_I(G \to H) &= b_\to(val_I(G), val_I(H))
\end{aligned}$$

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

Übung zu Interpretationen

- Wie viele Interpretationen gibt es bei $k = 1, 2, 3$ Variablen?
- Wie viele Interpretationen gibt es bei $k+1$ Variablen im Vergleich zu k Variablen?

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

Übung zur Aussagenlogik

Karlsruher Institut für Technologie

Sei $A := w, B := w, C := f$.

- Ist $(A \wedge B) \vee \neg C$ wahr oder falsch?
  $(A \wedge B) \vee \neg C = (w \wedge w) \vee \neg f = w \vee \neg f = w \vee w = w$, die Aussage ist also wahr.
- Ist $\neg(A \vee A)$ wahr oder falsch? Falsch! Wann ist $\neg(A \vee A)$ im allgemeinen wahr? Genau dann, wenn $\neg A$ wahr ist.

Aussagen Äquivalenz

Erinnerung: $A \leftrightarrow B$ heißt: $A \rightarrow B \wedge B \rightarrow A$.
Wenn zwei Aussagen äquivalent sind, sind ihre Wahrheitswerte immer gleich, wenn die Wahrheitswerte, von denen sie abhängen, gleich sind.
Mann sagt und schreibt dann: A ist *genau dann* wahr, *wenn* B wahr ist.

- $\neg(A \vee A)$ ist genau dann wahr, wenn $\neg A$ wahr ist, also gilt:
  $\neg(A \vee A) \leftrightarrow \neg A$.

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

Mehr zu Äquivalenz

Karlsruher Institut für Technologie

Alternative Definition zu Äquivalenz

Zwei Formeln G und H heißen äquivalent, wenn für jede Interpretation gilt $val_I(G) = val_I(H)$.

Vorher Äquivalenz von Formeln unter gegebener Interpretation, diesmal Äquivalenz von Formeln unter beliebiger Interpretation.

Bemerkung

- Man schreibt $G \equiv H$
- $\mathbb{B}^V \to \mathbb{B} : I \mapsto val_I(G)$

Beispiele

$(\neg(\neg P))$ ist äquivalent zu P

$(\neg(P \wedge Q))$ ist äquivalent zu $((\neg P) \vee (\neg Q))$

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

Beispiele zu Äquivalenz

- Ein Wort w hat die Länge $n \leftrightarrow |w| = n$.
- Die Vereinigung zweier Mengen A und B hat die Kardinalität $|A| + |B|$ $\leftrightarrow A \cap B = \emptyset \leftrightarrow A$ und B sind disjunkt.
- p ist eine rationale Zahl $\leftrightarrow$ p lässt sich darstellen als $p = \frac{a}{b}, a \in \mathbb{Z}, b \in \mathbb{N} \leftrightarrow p \in \mathbb{Q}$.

Wahrheitstabellen

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

- $(((P \to Q) \vee Q) \to (P \wedge \neg Q))$

P	Q	$(P \wedge Q)$	$\vee Q$	$\to$	$(P \wedge \neg Q)$
w	w	w	w	f	f
w	f	f	f	w	w
f	w	f	w	f	f
f	f	f	f	w	f

Übungen zu Aussagenlogik

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

Übungen zu Aussagenlogik

- Schreibe Wahrheitstabellen zu den Formeln um den Wahrheitsgehalt festzustellen.
- $\neg(P \wedge Q) \wedge \neg(Q \wedge P)$
- $(P \wedge Q \wedge R) \leftrightarrow (\neg P \vee Q)$
- $(A \wedge (B \vee C)) \leftrightarrow ((A \wedge B) \vee (A \wedge C))$
- Welche dieser Aussagen sind wahr?
- $\neg(P \wedge Q) = \neg P \vee \neg Q$
- $P \wedge P = P \vee P$
- $(P \vee Q) \wedge R = (P \wedge R) \vee (Q \wedge R)$

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

Wahrheitstabellen

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
w	w	f	w	w	w	w
w	f	f	f	w	f	f
f	w	w	f	w	w	f
f	f	w	f	f	w	w

Aufgabe

Finde einen logischen Ausdruck in A und B unter Verwendung von $\wedge$, $\vee$ und $\neg$, der die Aussage “Entweder A oder B” repräsentiert

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

Wahrheitstabellen

Aufgabe

Finde einen logischen Ausdruck in A und B unter Verwendung von $\wedge$, $\vee$ und $\neg$, der die Aussage "Entweder A oder B" repräsentiert

Lösung

A	B	$A \wedge \neg B$	$\neg A \wedge B$	$(A \wedge \neg B) \vee (\neg A \wedge B)$
w	w	f	f	f
w	f	w	f	w
f	w	f	w	w
f	f	f	f	f

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

Weitere Begriffe

Karlsruher Institut für Technologie

Tautologie

Die Formel G ist eine Tautologie (oder allgemeingültig), wenn G für alle Interpretationen wahr ist.

Erfüllbarkeit

Eine Formel G ist erfüllbar, wenn sie für mindestens eine Interpretation wahr ist.

Lemma

Wenn $G \equiv H$ ist, dann ist $G \leftrightarrow H$ eine Tautologie.

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

Übung zu Tautologien

Sind das Tautologien?

- $(G \rightarrow (H \rightarrow K)) \rightarrow ((G \rightarrow H) \rightarrow (G \rightarrow K))$ Ja
- $(\neg P \rightarrow Q) \wedge R \vee P$ Nein
- $G \rightarrow (H \rightarrow G)$ Ja
- $(\neg P \rightarrow \neg Q) \rightarrow ((\neg P \rightarrow Q) \rightarrow P)$ Ja

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

Übung zu Erfüllbarkeit

Karlsruher Institut für Technologie

Sind die folgenden Ausdrücke erfüllbar?

- $\neg(A \vee \neg A)$ nein
- $(P \wedge \neg Q) \vee (\neg P \wedge R)$ Ja

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 Tautologien, welche sind erfüllbar?
  - $\neg(P \wedge Q) = \neg P \vee \neg Q$
  - $P \wedge P = P \vee P$

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.
- Nach dem Prinzip der vollständigen Induktion fallen dann alle Steine um.

Vollständige Induktion

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

- 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)

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

- 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?
    $L_3 = \{w : w = ba^k n, k \in \mathbb{N}\}$

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

- 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 ε? $L_2 = A \cdot A^* = A^* \setminus \{\varepsilon\}$.
- Sprache $L_3 \subseteq B^*$, die alle Wörter über B enthält, mit:
  - Zwei beliebigen Zweichen aus B.
  - Dann einem C oder zwei D's.
  - Dann vier Zeichen aus A.
- $L_3 = B \cdot B \cdot \{C, DD\} \cdot A \cdot A \cdot A \cdot A$.

Produkt von Sprachen

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

Übung zu Produkt von formalen Sprachen

Sei A ein beliebiges Alphabet und $M := \{L : L$ 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

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.

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

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

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

Konkatenationsabschluss bei formalen Sprachen

Konkatenationsabschluss

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

ε-freie Konkatenationsabschluss

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

- Warum gilt $\varepsilon \notin L^+$ von beliebiger formeller Sprache L?
- $L := \{a, b, c\}. L^* = \{\varepsilon, a, aa, ab, ac, aaa, aab, \ldots, b, ba, bb, \ldots\}$
- $L := \{aa, bc\}. L^* = \{\varepsilon, aa, bc, aa \cdot aa, aa \cdot bc, bc \cdot aa, bc \cdot bc, aa \cdot aa \cdot aa, \ldots\}$

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

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

Erinnerung

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

Beweisaufgabe

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

$\subseteq$:

Vorraussetzung: $w \in L^* \cdot L$ 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 $w \in L^+$.

$\supseteq$:

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

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

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

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.
- Aber irgendwie so, dass $42_{\in A_{dez}} \overset{Deutung}{=} 101010_{\in A_{bin}} \overset{Deutung}{=} 52_{\in A_{oct}}$.

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.

- 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

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

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:
$2(2(2(2(2\cdot 1+0)+1)+0)+1)+0 = 2^4\cdot 1+2^4\cdot 0+2^3\cdot 1+2^2\cdot 0+2^1\cdot 1+2^0\cdot 0.$
Daher, einfachere Rechenweise:
$Num_k(w) = k^0 \cdot w(0) + k^1 \cdot w(1) + k^2 \cdot w(2) + \ldots.$
Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

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

Einfachere Umrechnung von Zahlendarstellungen

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

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

Was sind folgende Zahlen in Dezimal im Kopf gerechnet?

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

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	0

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 $Repr_k$

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

Zum Beispiel:

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

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

Beispiel zu $Repr_k$

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

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:

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

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

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

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 ℓ 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 $\ell - 1$?

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

Aufgaben zu Zweierkomplement-Darstellung

Karlsruher Institut für Technologie

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

Was sind folgende Zahlen in Zweierkomplement-Darstellung?

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

Informationen

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

Zum Tutorium

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

Mehr Material

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

Zur Veranstaltung

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

Zum Übungsschein

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