0

Äquivalenzrelation R refl. symm. transitiv Relation auf M

Äquivalenzklassen: $[\![a]\!]_R := \{ b \in M \mid bRa \}$

- disjunkte Zerlegung von M
- Arithmetik / Abbildung

BSP **$\equiv_7$, $\mathbb{Z}_7$, $\mathbb{Z}/7\mathbb{Z}$** $\quad [\![2]\!]_{\equiv_7} = \{ \ldots, -12, -5, 2, 9, 16, \ldots \} = \overline{2}$

$$\overline{-12} \otimes_7 \overline{2} = \overline{-24} = \overline{-3} = \overline{4} = \overline{2} \otimes_7 \overline{2}$$

Abbildung $\otimes_7 : \mathbb{Z}_7 \times \mathbb{Z}_7 \longrightarrow \mathbb{Z}_7$ **wohldefiniert**

$\overline{a} \otimes_7 \overline{b} := \overline{a \cdot b}$ Unabhängig von Repräsentanten

1

Theorem 2.18 Sei $n \in \mathbb{N}$ beliebig und $a, a', b, b' \in \mathbb{Z}$ mit

$$[\![a]\!] = [\![a']\!], \quad [\![b]\!] = [\![b']\!]$$

Dann gilt:

$$[\![(a+b)]\!] = [\![(a'+b')]\!] \qquad \text{I.}$$

$$[\![(a \cdot b)]\!] = [\![(a' \cdot b')]\!] \qquad \text{II.}$$

Beweis:

I. $[\![a]\!] = [\![a']\!]$ $\quad [\![b]\!] = [\![b']\!]$

n. Def. so ex $r, q \in \{0, 1, 2, 3, \ldots, n-1\}$

so ex $x, x', y, y' \in \mathbb{Z}$

$$a = x \cdot n + r \quad a' = x' \cdot n + r \quad b = y \cdot n + q \quad b' = y' \cdot n + q$$

②

$$(a\cdot b) = (x\cdot n + r)\cdot(y\cdot n + q)$$
$$= (x\cdot y\cdot n + \underbrace{x\cdot q + y\cdot r}_{=: \ell_1})\,n + r\cdot q$$

$$(a'\cdot b') = (x' y' n + \underbrace{x'\cdot q + y'\cdot r}_{=: \ell_2})\,n + r\cdot q$$

Eindeutig Darstellung modulo n

$\Rightarrow$ $a\cdot b$, $a'\cdot b'$ haben modulo n den gleichen Rest, nämlich den Rest von $\underline{r\cdot q}$ modulo n

$\square$

Ergänzung:

$r\cdot q$ modulo n eindeutige $\ell\cdot n + r'$ Darstellung

$\Rightarrow$ $a\cdot b = (\ell_1 + \ell)\cdot n + r'$ und $a'\cdot b' = (\ell_2 + \ell)\cdot n + r'$

$r' \in \{0,1,2,\dots,n-1\}$

Modulo Rechnung:

BSP

1. Tageszeit Modulo 24

10 Std nach 21 Uhr

$$[\![21]\!] \oplus_{24} [\![10]\!] = [\![7]\!]$$

2. Programmierung 8-Bit Register

	2^7			2^4	2^3	2^2	2^1	2^0
~~X~~	0	0	0	0	1	0	1	0
~~X~~	1	1	1	1	0	1	1	0
1	1	1	1	1	1	1		
1	0	0	0	0	0	0	0	0

$$[\![10]\!] \oplus_{256} [\![246]\!] = [\![0]\!]$$

<u>Begriffe</u>: <u>Hinreichend, notwendig</u>

für Implikation $(A \Rightarrow B)$

<u>Annahme</u> $(A \Rightarrow B)$ ist <u>wahr</u>

	A	B	$(A \Rightarrow B)$
a)	0	0	1
~~b)~~	~~1~~	~~0~~	~~0~~
c)	0	1	1
d)	1	1	1

(Gültigkeit von) A ist <u>hinreichend</u> für (Gültigkeit von) B

(Gültigkeit von) B ist <u>notwendig</u> für (Gültigkeit von) A

3. Endliche Automaten und Formale Sprachen

Motivation Automaten

Ausgangsbeispiele

+ Komplexität von Rechenaufgabe
+ Modell eines Rechners
+ Einfaches Modell
+ Gültigkeit des Ausdrücken
+ Getränkautomat (Zustände)

Formale Sprachen Programmiersprache ⇝ Compiler

1. Schlüsselwörtererkennung IF THEN ELSE
   Lexikalische Analyse Token
2. Zusammensetzung korrekt IF (X>0) THEN ... END
   Syntaktische Analyse Strukturbaum

3.1 Sprachen und Grammatiken

Σ endliche Mengen von Zeichen, Alphabet $\Sigma \neq \emptyset$

Wort w ist eine endliche Konkatenation (Folge) von Zeichen bzw. Elemente aus Σ

$\Sigma^* :=$ Menge aller Wörter über Σ

$\varepsilon :=$ leere Wort $\quad \varepsilon \in \Sigma^* \qquad \Sigma^+ := \Sigma^* \setminus \{\varepsilon\}$

Definition 3.1: Eine Menge $L \subseteq \Sigma^*$ heißt (formale) Sprache über Σ.

Definition 3.2: Notationen / Bezeichnungen Σ, $a \in \Sigma$

$a^0 := \varepsilon$, $a^n := \underbrace{aa \cdots a}_{n\text{-mal}} \quad n \in \mathbb{N}_0$

$\underline{w \in \Sigma^*}$, $|w| :=$ Länge des Wortes, # Buchstaben

$|\varepsilon| = 0$ $\quad w = a_1 a_2 \cdots a_n \quad w^R := a_n a_{n-1} \cdots a_1$

$\underline{a \in \Sigma}$ $\quad |w|_a :=$ # Vorkommen von a in w

BSP $\quad \Sigma = \{1\} \qquad \Sigma^* = \{\varepsilon, 1, 11, 111, \dots\}$

$$= \{1^i \mid i \in \mathbb{N}_0\}$$

$\Sigma = \{0,1\} \qquad \Sigma^* = \{\varepsilon, 1, 0, 00, 01, 10, 11, 000, \dots\}$

$L_1 = \{0, 01, 111, \varepsilon\}$

$L'' = \{0^m 1^n \mid m, n \in \mathbb{N}_0\}$

$L' = \{0^n 1^n \mid n \in \mathbb{N}_0\} \qquad L' \subset L''$

Sprachen durch Grammatiken erzeugen

Definition 3.3: Grammatik G ist 4-Tupel

$$G = (\Sigma, V, S, P)$$

derart, dass

- Σ ist endliches Alphabet, enthält Terminalsymbole
- V endliche Menge von Nichtterminalsymbole $\Sigma \cap V = \emptyset$
- $S \in V$ ist Startsymbol
- P endliche Menge von Ableitungsregeln: Paare $(l_1, r_1), (l_2, r_2), \ldots, (l_n, r_n)$
  $l_i \in V^*$, $r_i \in (V \cup \Sigma)^*$ Schreibweise: $l_i \to r_i$ $i = 1, \ldots, n$ $n \in \mathbb{N}$

Γ $l \to r_1,\ l \to r_2,\ \dots,\ l \to r_k$ <u>Abkürzung</u>!

$l \to r_1, r_2, r_3, \dots, r_k$ $\quad$ $l \to r_1 \mid r_2 \mid \dots \mid r_k$

BSP 1. $\Sigma = \{0, 1\}$ $\quad V = \{S, A, B\}$ $\quad \boxed{\{0^n 1^m \mid n, m \in \mathbb{N}_0\} = L(G)}$

$G = (\Sigma, V, S, P)$ $\quad P = \{S \to AB,\ A \to \varepsilon \mid 0A,\ B \to \varepsilon \mid 1B\}$

(Informale)-<u>Definition</u>: Wörter erzeugen / <u>ableiten</u>

+ Beginne mit Startsymbol
+ Anwenden der Ableitungsregeln (Ersetzen der Nichtterminalsymbol
+ Solange bis nur noch Terminalsymbole vorhanden sind.

BSP 011 erzeugen Ableitungskette:

$$S \to AB \to 0AB \to 0\varepsilon B = 0B \to 01B \to 011B.$$
$$\to 011\varepsilon = 011$$

Definition 3.4 Sei $G = (\Sigma, V, S, P)$
Grammatik

Dann heißt

$$L(G) := \{ w \in \Sigma^* \mid w \text{ ist mit } G \text{ aus } S \text{ ableitbar} \}$$

die von G erzeugte Sprache.

BSP

2. $V = \{S\}$ $\Sigma = \{\cancel{0}, 1\}$ $P = \{S \to \varepsilon \mid 11S\}$

$L(G) = \{1^n \mid n \in \mathbb{N}_0, n \text{ gerade}\}$

3. $V = \{S\}$ $\Sigma = \{0, 1\}$ $P = \{S \to \varepsilon \mid 0S1\}$

$L(G) = \{0^n 1^n \mid n \in \mathbb{N}_0\}$

Grammatik für syntaktisch korrekte Programme

Sätze bauen

4. $V = \{$ SATZ, SUBJEKT, PRÄDIKAT, OBJEKT $\}$

$\Sigma = \{$ Willi, ißt, betrachtet, den␣Mond, einen␣Apfel $\}$

$P = \{$ SATZ $\to$ SUBJEKT␣PRÄDIKAT␣OBJEKT,

SUBJEKT $\to$ Willi, PRÄDIKAT $\to$ ißt | betrachtet,

OBJEKTE $\to$ den␣Mond, einen␣Apfel $\}$

SATZ $\rightsquigarrow$ Willi␣ißt␣den␣Mond <u>ableitbar</u>

<u>Wortproblem</u>: G und Wort $w \in \Sigma^*$ gegeben:

$$w \in L(G)\,?$$

Spezielle Grammatiken sind "besser" geeignet!
Komplexität der Sprache $L(G)$ hängt von P ab!

<u>Definition 3.5</u>: (Chomsky-Hierarchie)

<u>Chomsky-0</u>: Grammatiken ohne Einschränkungen in P

<u>Chomsky-2</u>: "kontextfreie" Grammatiken

$$A \to v \quad A \in V \quad v \in (V \cup \Sigma)^*$$

<u>Chomsky-3</u>: "regulär" (rechtslinear) Grammatiken

$A \to v$ $A \in V$ mit $v = \varepsilon$ oder $v = aB$ $a \in \Sigma$, $B \in V$

Wortproblem für reguläre Sprachen lösen:

effektiv / effizient

Chomsky-1: "kontextsensitiven" Grammatiken

$$\alpha A \beta \to \alpha \gamma \beta \quad \text{oder} \quad S \to \varepsilon$$

$$A \in V, \ \alpha, \beta \in V^*, \ \gamma \in (V \cup \Sigma)^*$$

S steht nie auf der rechten Seite.