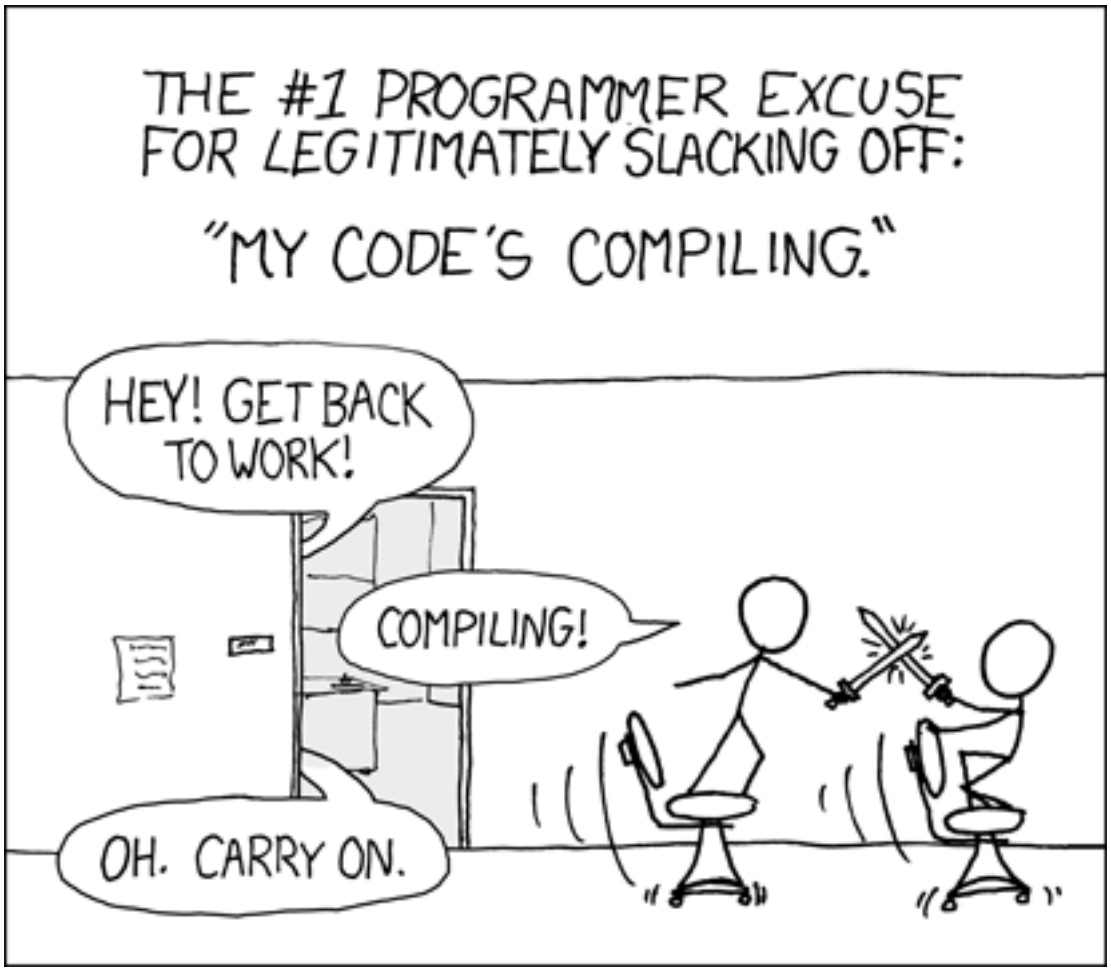
GBI Zusammenfassung

Jessica Ochs und Andreas Mai

17.02.2016

GBI Klausur am 02.03.2016
14:00 - 16:00



Kein Anspruch auf Vollständigkeit ;)

Inhaltsverzeichnis

1 Mengen

1.1 Kartesisches Produkt

- $M \times N = \{(m, n) \mid m \in M, n \in N\}$
- $A \times \varnothing = \varnothing$

2 Relationen

2.1 Funktionen

- Funktion = Rechtseindeutige und Linkstotale Relation
- Injektive Funktion = Linkseindeutige Funktion
- Surjektive Funktion = Rechtstotale Funktion
- Bijektive Funktion = Injektive und Surjektive Funktion

2.2 Potenzmengen

- $\mathcal{P}(M)$ ist die Menge aller möglichen Teilmengen von M
- $\forall M_i \subseteq M : M_i \in \mathcal{P}(M)$

2.3 Mengengleichheit

- $A = B \Leftrightarrow A \subseteq B \wedge B \subseteq A$
- $A \setminus B \Leftrightarrow \{x \in A \wedge x \notin B\}$

3 Wörter

- Alphabet = Endliche Menge von Zeichen
- Wort w aus dem Alphabet A ist eine Folge von konkatenierten Zeichen aus A
- A^* = Menge aller Wörter über A

3.1 Konkatenation

- $w_1 \circ w_2 \neq w_2 \circ w_1$
- $w = w_1 \circ w_2$ und $w_1 \in A^*, w_2 \in B^* \Rightarrow w \in (A \cup B)^*$

3.2 Das Leere Wort

- $\varepsilon := P_0 \to A$
- $\varepsilon : \{\} \to \{\}$

3.3 Länge eines Wortes

- $|w^k| = k * |w|$
- $|\varepsilon| = 0$
- $|a \circ b| = |a| + |b|$
- A^n ist die Menge aller Wörter der Länge n
- $A^* = \bigcup_{i=0}^{\infty} A^i$

4 Binäre Operationen

- Eine Binäre Operation auf einer Menge M ist eine Abbildung $\diamond : M \times M \to M$
- kommutativ: $\forall x, y \in M : x \diamond y = y \diamond x$
- assoziativ: $\forall x, y, z \in M : (x \diamond y) \diamond z = x \diamond (y \diamond z)$

5 Aussagenlogik

- Var_{AL} = Menge aller Aussagevariablen
  Beispiel: $Var_{AL} = \{A, B, C\}$
- For_{AL} = Menge aller möglichen Formeln über Var_{AL}
  Beispiel: $For_{AL} = \{A \to B, ...\}$

6 Induktion

- Behauptung
- Induktionsanfang: Zeige: Behauptung gilt für $n = 0$
- Induktionsvoraussetzung: Die Beh. gelte für ein beliebiges aber festes $n \in \mathbb{N}_0$
- Induktionsschritt: Zeige: Behauptung gilt für $n + 1$

7 Formale Sprachen

Sei A ein Alphabet. Die Formale Sprache $L \subseteq A^*$ ist eine Sprache, die alle laut L syntaktisch korrekten Gebilde enthält

7.1 Produkt formaler Sprachen

$L_1 * L_2 = \{w_1 * w_2 \mid w_1 \in L_1 \wedge w_2 \in L_2\}$

7.2 Potenz Formaler Sprachen

$L^0 = \{\varepsilon\}$
$L^{i+1} = L^i * L \ (i \in \mathbb{N}_0)$

7.3 Konkatenationsabschluss

$L^* = \bigcup_{i=0}^{\infty} L^i$

7.3.1 ε -freier Konkatenationsabschluss:

$L^+ = \bigcup_{i=1}^{\infty} L^i$
Achtung: Allgemein gilt nicht $\varepsilon \notin L^+$

8 Übersetzung von Wörtern in Zahlen

Ich will ja nicht klugscheißen, aber meinst du nicht Codierung??

- Zahlenbasis b
- $Num_b(\varepsilon) = 0$
- $\forall w \in \mathbb{Z}_0^*, \forall x \in \mathbb{Z}_0 : Num_b(w * x) = b * Num_b(w) + num_b(x)$
- Num_b ist die Umrechnung aus dem Binärsystem in das Dezimalsystem
- $Repr_k(n)$ ist das kürzeste Wort $w \in \mathbb{Z}_k^*$ mit $Num_k(w) = n$,
  also $Num_k(Repr_k(n)) = n$
  Achtung: Im allgemeinen: $Repr_k(Num_k(w)) \neq w$
- $$Repr_k : \mathbb{N}_0 \to \mathbb{Z}_k$$
  $$n \mapsto \begin{cases} repr_k(n) & \text{wenn } n < k \\ Repr_k(n \ div \ k) * repr_k(n \ mod \ k) & \text{wenn } n \geq k \end{cases}$$

8.1 Zweierkomplement

- Bietet die Möglichkeit, negative Zahlen Binär darzustellen
- Vorteilhaft bei Berechnungen im Prozessor
- $Zkpl_k(x) = \begin{cases} 0 * bin_{k-1}(n) & \text{wenn } x \geq 0 \\ 1 * bin_{k-1}(2^{k1} + x) & \text{wenn } x < 0 \end{cases}$

9 Homomorphismus

- Strukturerhaltene Abbildung
- Kann Präfixfrei und ε-frei sein

9.1 Strukturerhaltend

$\forall x, y \in A^* : A(xy) = h(x) \circ h(y)$
Beispiel: $h(a) = 2, h(b) = 3 \Rightarrow h(aba) = 232$

9.2 ε-frei

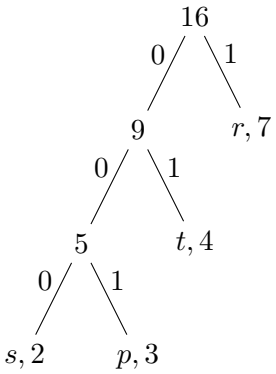
$\forall x \in A : h(x) \neq \varepsilon$
Beispiel: $Kannichnichtlesen$

9.3 Präfixfrei

$\forall x \in A^* \nexists v, z \in A^* \wedge w \neq vz : h(w) = h(v) \circ h(z)$
Beispiel: $Kannichauchnedlesen$

9.4 Huffman-Codierung

Beispiel: $w = strrprrrstprprtt$



Buchstabe aus w	x	r	t	p	s
Anzahl des Buchstabens in w	$N_x(w)$	7	4	3	2
Huffman-Codierung	$h(x)$	1	01	001	000

Aus der Tabelle folgt: $h(w) = 00001110011110000100110011010 1$

10 Speicher

- Eine Speichereinheit, 0 oder 1, wird Bit genannt
- Ein Wort aus 8 Bits wird Byte genannt
- Ein Speicher bildet Adressen (adr) auf Werte (val) ab.
- Methoden
  - $memread(m, adr)$: Liest den Wert val einer Zelle adr aus dem Speicher m
  - $memwrite(m, adr, val)$ Schreibt einen Wert val in die Zelle adr eines Speichers m

11 MIMA (Minimalmaschiene)

- idealisierter Prozessor
- Adressen adr sind 20-Bit Wörter
- Werte val sind 24-Bit Wörter
- Befehlscodierungen
  - 4-Bit Befehl + 20-Bit Parameter
  - 8-Bit Befehl + irrelevanter Rest

11.1 Wichtige Register

Register		**Beschreibung**
IAR	InductionAddressRegister	Speichert Adresse des aktuell auszuführenden Befehls
IR	InductionRegister	Speichert den aktuell auszuführenden Befehl
SAR	StorageAddressRegister	Enthält Adresse eines Wertes, der aus dem Speicher gelesen werden soll
SDR	StorageDataRegister	Enthält den Wert, der aus dem Speicher geladen wurde

11.2 Befehle

Befehl		**Beschreibung**	**Funktion**
LDIV	*adr*	Load indirect value from address	$M(M(adr)) \rightarrow Akku$
STIV	*adr*	Store indirect value at address	$Akku \rightarrow M(M(adr))$
LDC	*const*	Load constant	$const \rightarrow Akku$
LDV	*adr*	Load value from address	$M(adr) \rightarrow Akku$
STV	*adr*	Store value at address	$Akku \rightarrow M(adr)$

11.2.1 Fetch (Befehlsholphase)

- $IAR \rightarrow X$
- $Eins \rightarrow Y$
- $Z \rightarrow IAR$

11.2.2 LDC (Load Constant)

- Fetch
- $IR \rightarrow Akku$

11.2.3 LDV (Load Value)

- Fetch
- $IR \rightarrow SAR$
- $SDR \rightarrow Akku$

11.2.4 LDIV (Load Indirect Value)

- Fetch
- $IR \rightarrow SAR$
- $SDR \rightarrow SAR$
- $SDR \rightarrow Akku$

11.2.5 STV (Store Value)

- Fetch
- $IR \rightarrow SAR$
- $Akku \rightarrow SDR$

11.2.6 STIV (Store Indirect Value)

- Fetch
- $IR \rightarrow SAR$
- $SDR \rightarrow SAR$
- $Akku \rightarrow SDR$

11.2.7 JMP (Jump)

- Fetch
- $IR \rightarrow IAR$

11.2.8 EQL (Vergleich)

- Fetch
- $IR \rightarrow SAR$
- $Akku \rightarrow X$
- $SDR \rightarrow Y$
- $ALU \rightarrow Z$ (Bei Gleichheit -1, ansonsten 0)
- $Z \rightarrow Akku$

11.2.9 ADD (Addition)

- Fetch
- $IR \rightarrow SAR$
- $Akku \rightarrow X$
- $SDR \rightarrow Y$
- $ALU \rightarrow Z$ (Addition)
- $Z \rightarrow Akku$

12 Kontextfreie Grammatik

$G = (N, T, S, P)$ ist eine kontextfreie Grammatik

- N: Nichtterminalsymbole
- T: Terminalsymbole, disjunkt zu N
- S: Startsymbol $S \in N$
- P: Produktionsmenge $P \subseteq N \times (N \cup T)^*$

12.1 Ableitung

Annahme: $w \in V^*, v \in V^*$ und es gibt eine Aufspaltung in $w = w_1 X w_2$ und $v = v_1 w v_2$
Mit $w_1, w_2 \in V^*$ und der Produktion $(X, w) \in P$ ist v aus w ableitbar.
Wir schreiben nun $w \Rightarrow v$
Mit $w \Rightarrow^i v$ für $i \in \mathbb{N}$ bezeichnen wir zwei Wörter, wenn zwischen ihnen i (gleiche) Ableitungsschritte liegen.
$\Rightarrow^*$ ist die reflexiv-transitive Hülle der Relation $\Rightarrow$

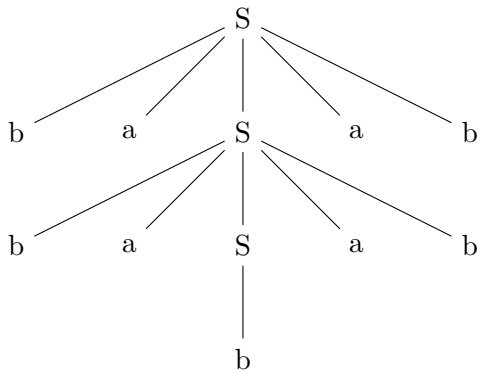
12.2 Erzeugte Sprache

- $L = L(G)$ mit $L = \{w \in T^* \mid S\} \Rightarrow^* w$
- $L(G)$ ist die von der Grammatik G erzeugte Sprache

12.3 Ableitungsbaum

Beispiel: $G = (\{S\}, \{a, b\}, S, \{S \to baSab \mid b\})$

Ableitung für $w = bababab$:



$S \Rightarrow baSab \Rightarrow babaSabab \Rightarrow bababab$

13 Relationen Part 2

13.1 Produkt

$S \circ R = \{(x, z) \in M_1 \times M_3 \mid \exists y \in M_2 \text{ mit } (x, y) \in R \text{ und } (y, z) \in S\}$
für $R \subseteq M_1 \times M_2$ und $S \subseteq M_2 \times M_3$

13.2 Potenzen

- $R^0 = I_\mu$
- $R^{i+1} = R^i \circ R$ für alle $i \in \mathbb{N}$

13.3 Reflexiv-transitive Hülle einer Relation R

- $R^* = \bigcup_{i \in \mathbb{N}_\nvDash} R^i$
- Reflexiv: $\forall x \in M : (x, x) \in R$
- Transitiv: $\forall x, y, z \in M : (x, y) \in R \wedge (y, z) \in R \rightarrow (x, z) \in R$

14 Prädikatenlogik

3 Schritte für den Aufbau Prädikatenlogischer Formeln

14.1 Terme

- **Konstantensymbole** $Const_{PL}$
  c_i (für endlich viele $i \in \mathbb{N}_\nvDash$, kurz c, d)
- **Variablensymbole** Var_{PL}
  x_i (für endlich viele $i \in \mathbb{N}_\nvDash$, kurz x, y, z)
- **Funktionssymbole** Fun_{PL}
  f_i (für endlich viele $i \in \mathbb{N}_\nvDash$, kurz f, g, h)
  jedes $f_i \in Fun_{PL}$ hat Stelligkeit $ar(f_i \in \mathbb{N}_+)$

14.2 Atomare Formeln

- **Relationssymbole** Rel_{PL}
  $\doteq$ immer dabei
  R_i (für endlich viele $i \in \mathbb{N}_\nvDash$, kurz: R, S
  jedes $R_i \in Rel_{PL}$ hat Stelligkeit $ar(R_i \in \mathbb{N}_+)$

14.3 Prädikatenlogische Formeln

Bestehen aus Atomaren Formeln und aussagenlogischen Konnektiven und Quantoren
Aussagenlogische Konnektive: $\{, \neg \wedge, \vee, \rightarrow, \forall, \exists\}$
Beispiele (aus dem Tutorium):

- $\forall x(x \doteq a \vee x \doteq b \vee x \doteq c)$
- $\forall x, \forall y(kills(x, y) \rightarrow \neg richer(x, y))$ mit:

Freie und gebundene Variablen können vorkommen

- $\forall x(p_0(x, y) \rightarrow \forall z(\exists y p_1(y, z) \vee \forall x p_2(f(x), x)))$
- $\forall x(R(x, y) \wedge \exists y(R(x, y))$

15 Hoare-Kalkül

15.1 Hoare Tripel

- Tripel $(\{P\}S\{Q\})$ mit einem Programmstück S und prädikatenlogischen Zusicherungen P, Q
- P: Bedingung vor Ausführung (Vorbedingung)
- Q: Bedingung nach Ausführung (Nachbedingung)

15.2 Hoare Regeln

15.2.1 HT1

Wenn $(\{P\}S\{Q\})$ gilt, dann gilt auch $(\{P'\}S\{Q'\})$ mit:

- $P' \Rightarrow P$
- $Q \Rightarrow Q'$

$\Rightarrow$ Vorbedingungen können stärker und Nachbedingungen können schwächer werden

15.2.2 HT2

Wenn $(\{P\}S_1\{Q\})$ und $(\{P\}S_2\{Q\})$ gilt, dann gilt auch $(\{P\}S_1S_2\{Q\})$
$\Rightarrow$ Hoare-Tripel können transitiv zusammengefasst werden

15.2.3 HT3

$\{\sigma_{x/E}(Q)\}x \leftarrow E\{Q\}$
$\Rightarrow$ Nach der Zuweisung gilt jede Aussage für die Variable, die vorher für die linke Seite galt.
$\sigma_{x/E}(Q)$ ist die Aussage, die dadurch entsteht, dass man in Q jedes freie Vorkommen von x durch E ersetzt.
Beispiel: $\{x + 1 = 43\}y := x + 1\{y = 43\}$

15.2.4 HT4

Wenn $\{P \wedge B\}S_1\{Q\}$ und $\{P \wedge \neg B\}S_2\{Q\}$ gilt, dann gilt auch
$\{P\}$ if B then S_1 else S_2 fi $\{Q\}$

15.2.5 Schleifeninvarianten

- Aussagen, die bei jedem Schleiendurchgang gleich sind
- helfen Korrektheit eines Programms zu beweisen
- beweist man durch vollständige Induktion

Beispiel:

$x \leftarrow a \in \mathbb{N}_{\not\vdash}$
$y \leftarrow b \in \mathbb{N}_{\not\vdash}$
for i in 1 to b do
  $x \leftarrow x + 1$
  $y \leftarrow y + 1$
od
Output x

Schleifeninvariante: $x_i + y_i = a + b$

16 Graphen

- $G = (V, E)$
- V: Menge aller Knoten im Graph G
- E: Menge aller Kanten im Graph G
  - Gerichteter Graph: $E \subseteq V \times V$
    Tupel, da Reihenfolge wichtig
  - Ungerichteter Graph: $E \subseteq \{\{x, y\} \mid x, y \in V\}$
    Mengen, da Reihenfolge unwichtig
- Schlinge: Kante zu sich Selber. Schlinge von V_0: (V_0, V_0)

16.1 Teilgraph

Ein Graph $T = (V', E')$ ist ein Teilgraph von G, wenn:

- $V' \subseteq V$
- $E' \subseteq E \cap V' \times V'$
- Also dürfen keine Kanten aus dem Teilgraphen hinausführen

16.2 Knotengrad

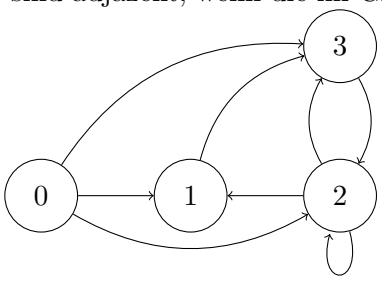
- Eingangsgrad: $d^-(k) = |\{x \mid (x,k) \in E\}|$
  Anzahl aller Eingehenden Kanten
- Ausgangsgrad: $d^+(k) = |\{x \mid (k,x) \in E\}|$
  Anzahl aller Ausgehenden Kanten
- Grad: $d^-(k) + d^+(k)$ Eingangsgrad + Ausgangsgrad
  Anzahl aller Kanten
- bei Ungerichteten Graphen gilt: $d^-(k) = d^+(k)$ Eingangsgrad = Ausgangsgrad

16.3 Pfad

- Folge von Knoten, die über Kanten erreichbar sind
  $p = (v_0, v_1, ..., v_n)$ mit $(v_i, v_{i+1}) \in E$
- Länge des Pfades = Anzahl der Knoten
- Geschlossener Pfad: $v_0 = v_n$
- Wiederholungsfreier Pfad: Alle Knoten sind Paarweise Verschieden (außer v_0 und v_n)
- einfacher Zyklus: geschlossen und wiederholngsfreier Pfad

17 Wege in Graphen finden

Zwei Knoten x, y sind adjazent, wenn die im Graphen durch eine Kante verbunden sind



Beispielgraph:

17.1 Adjazenzliste

Alle Knoten y, die zu einem Knoten x adjazent sind, werden eingetragen

Beispiel:

x	y
0	$1, 2, 3$
1	3
2	$1, 2, 3$
3	2

17.2 Adjazenzmatrix

- Graph $G = (V, E)$ mit n Knoten

  $A_{ij} = \begin{cases} 0 & \text{wenn } (i,j) \notin E \\ 1 & \text{wenn } (i,j) \in E \end{cases}$

- Beispiel: $A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
- Schlingen: $A_{ii} = 1$
- Ungerichteter Graph: A ist Symmetrisch
- Potenzen der Adjazenzmatrix
  - $(A^n)_{ij}$ gibt Auskunft darüber, ob es einen Weg der Länge n von i nach j gibt.
  - Beispiel: $A^2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

17.3 Wegematrix

- $W_{ij} = \begin{cases} 0 & \text{wenn } (i,j) \notin E^* \\ 1 & \text{wenn } (i,j) \in E^* \end{cases}$
- lässt sich berechnen über: $sgn((\sum_{k=0}^{n} A^k)_{ij})$
  - Signum Funktion: $sgn(x) = \begin{cases} 1 & \text{wenn } x > 0 \\ 0 & \text{wenn } x = 0 \\ -1 & \text{wenn } x < 0 \end{cases}$
- Die Wegematrix ist die reflexiv-transitive Hülle der Adjazenzmatrix
- Beispiel: $W = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$

18 Relationen (21 im Skript)

18.1 Äquivalenzrelation $\equiv$

Eine Äquivalnzrelation $\equiv$ besitzt immer folgende 3 Eigenschaften:

- reflexiv: $\forall x \in M : x \equiv x$
- symmetrisch: $\forall x, y \in M : x \equiv y \Leftrightarrow y \equiv x$
- transitiv: $\forall x, y, z \in M : x \equiv y \vee y \equiv z \Leftrightarrow x \equiv z$

18.2 Äquivalenzklasse

- Eine Äquivalenzklasse $[x]_{\equiv}$ von x für $x \in M$ ist definiert durch $\{y \in M \mid x \equiv y\}$
- Die Faktormenge $M_{/\equiv}$ bezeichnet die Menge aller Äquivalenzklassen $\{[x]_{\equiv} \mid x \in M\}$

18.3 Verträglichkeit von Relaionen und Operationen

- $\equiv$ ist die Äquivalenzrelation auf M
- $f : M \to M$ ist eine Abbildung
  verträglich mit $\equiv$: $\forall x_1, x_2 \in M : x_1 \equiv x_2 \Rightarrow f(x_1) \equiv f(x_2)$
- $\diamond$ sei eine binäre Operation auf M
  verträglich mit $\equiv$: $\forall x_1, x_2, y_1, y_2 \in M : x_1 \equiv x_2 \vee y_1 \equiv y_2 \Rightarrow x_1 \diamond y_1 \equiv x_2 \diamond y_2$

18.4 Kongruenzrelation

Eine Kongruenzrelation ist eine Äquivalenzrelation, die mit interessierenden Funktionen, Operationen oder beidem verträglich ist.

18.5 Antisymmetrie

- Eine Relation $R \subseteq M \times M$ ist antisymmetrisch, wenn gilt:
- $\forall x, y \in M : xRy \wedge yRx \Rightarrow x = y$

18.6 Halbordnung

Eine Relation ist eine Halbordnung, wenn sie reflexiv, antisymmetrisch und transitiv ist.

18.7 Hasse-Diagram

- “Skelett der Halbordnung”
- Hasse-Diagramm $\Rightarrow$ Halbordnung: reflexiv-transitive Hülle bilden
- Halbordnung $\Rightarrow$ Hasse-Diagramm: Kanten, die wegen Reflexivität und Transitivität klar sind, weglassen

- R ist die Halbordnung und H_R das dazugehörige Hasse-Diagramm. Dann gilt: $H_R^* = R$

- DAG = Directed Acyclic Graph
  = gerichteter zyklenfreier Graph
  = Hasse-Diagramm einer endlichen Halbordnung

18.8 Nerode Äquivalenzelation

- Für jede formale Sprache L ist $\equiv_L$ eine Äquivalenzrelation
  $w_1 \equiv_L w_2 \Leftrightarrow \forall w \in A^* : w_1 w \in L \Leftrightarrow w_2 w \in L$

- $w_1 \not\equiv_L w_2 \Leftrightarrow$ Es gibt ein $w \in A^*$, sodass genau eines der wörter $w_1 w$ und $w_2 w$ in L liegt und das jeweils andere nicht

18.9 Minimale und Maximale Elemente

- Sei $(M, \sqsubseteq)$ eine Halbordnung und $T \subseteq M$

- $x \in T$ heißt **Maximales Element** von T, wenn es kein $y \in T$ gibt, mit $x \sqsubseteq y$ und $x \neq y$

- $x \in T$ heißt **Minimales Element** von T, wenn es kein $y \in T$ gibt, mit $y \sqsubseteq x$ und $y \neq x$

18.10 Kleinste und Größte Elemente

- Sei $(M, \sqsubseteq)$ eine Halbordnung und $T \subseteq M$

- $x \in T$ heißt **Größtes Element** von T, wenn $\forall y \in T : y \sqsubseteq x$

- $x \in T$ heißt **Kleinstes Element** von T, wenn $\forall y \in T : x \sqsubseteq y$

- Kleinstes und Größtes Element sind eindeutig

18.11 Untere und Obere Schranke

- Sei $(M, \sqsubseteq)$ eine Halbordnung und $T \subseteq M$

- $x \in M$ heißt **Obere Schranke** von T, wenn $\forall y \in T : y \sqsubseteq x$

- $x \in M$ heißt **Untere Schranke** von T, wenn $\forall y \in T : x \sqsubseteq y$

- Können auch außerhalb von T sein

18.12 Supremum und Infimum

Besitzt eine Menge T aller $\begin{matrix}\text{unteren}\\\text{oberen}\end{matrix}$ Schranken ein $\begin{matrix}\text{kleinstes}\\\text{größtes}\end{matrix}$ Element, so heißt dies $\begin{matrix}\text{Infimum}\\\text{Supremum}\end{matrix}$ von T

19 Quantitative Aspekte

19.1 Asymptotosches Wachstum: $f \asymp g$

- Eine Funktion $g : \mathbb{N}_0 \to \mathbb{R}_0^+$ wächst größenordnungsmäßig genauso schnell wie eine Funktion $f : \mathbb{N}_0 \to \mathbb{R}_0^+$, wenn:

$$\begin{aligned}&\exists c, c' \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : cf(n) \leq g(n) \leq c'f(n)\\ &\Leftrightarrow f \asymp g\\ &\Leftrightarrow cf(n) \leq g(n) \wedge g(n) \leq c'f(n)\end{aligned}$$

- $\forall a, b \in \mathbb{R}_+ : af(n) \asymp bf(n)$
- $f \asymp g$ ist eine Äquivalenzrelation
- Beispiel: $f : n \mapsto 3n^2$ und $g : n \mapsto 10^{-2}n^2$
  Behauptung: $f \asymp g$
  - $cf(n) \leq g(n)$: für $c = 10^{-3}$ und $n_0 = 0$ gilt:
    $\forall n \leq n_0 : cf(n) = 10^{-3} * 3n^2 \leq 10^{-2}n^2 = g(n)$
  - $g(n) \leq c'f(n)$: für $c' = 1$ und $n_0 = 0$ gilt:
    $\forall n \leq n_0 : g(n) = 10^{-2}n^2 \leq 3n^2 = c'f(n)$
- Beispiel 2: $f : n \mapsto n^3 + 5n^2$ und $g : n \mapsto 3n^3 - n$
  Behauptung: $f \asymp g$
  - für $n \leq 0$ gilt:

$$\begin{aligned}f(n) &= n^3 + 5n^2\\ &\leq n^3 + n^3 = 6n^3\\ &= 9n^3 - 3n^3\\ &\leq 9n^3 - 3n\\ &= 3(3n^3 - n) = 3g(n)\\ \text{also } &\frac{1}{3}f(n) = g(n)\end{aligned}$$

  - Sowie: $g(n) = 3n^3 - n \leq 3n^3 \leq 3(n^3 + 5n^2) = 3f(n)$

19.2 Groß-O Notation Θ

- $\Theta(f) = \{g \mid f \asymp g\}$
  $\Theta(f) = \{g \mid \exists c, c' \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : cf(n) \leq g(n) \leq c'f(n)\}$
  - aus $\forall a, b \in \mathbb{R}_+ : af(n) \asymp bf(n)$ folgt $\forall a, b \in \mathbb{R}_+ : \Theta(af) = \Theta(bf)$
- $O(f) = \{g \mid \exists c \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \leq cf(n)$
  - $g \preceq f$, falls $g \in O(f) \Leftrightarrow g$ wächst asymptotisch höchstens so schnell wie f
- $\Omega(f) = \{g \mid \exists c \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \geq cf(n)$
  - $g \succeq f$, falls $g \in \Omega(f) \Leftrightarrow g$ wächst asymptotisch mindestens so schnell wie f
- $g \in O(f) \Leftrightarrow f \in \Omega(g)$, somit $g \preceq f \Leftrightarrow f \succeq g$
- $\Theta(f) = O(f) \cap \Omega(f)$, somit $g \asymp f \Leftrightarrow g \preceq f \wedge g \succeq f$

$log_2\ n$	1	2	3	4	5	6
n	2	4	8	16	32	64
n^2	4	16	64	256	1024	4096
n^3	8	64	512	4096	32768	262144
2^n	4	16	256	65536	viel	extrem viel