Universität Augsburg
Institut für Informatik

**Übung zur Vorlesung Informatik 1** WS 2019/20
Fakultät für Angewandte Informatik
Lehrprofessur für Informatik
PROF. DR. LORENZ, MARIUS BRENDLE, JOHANNES METZGER, LISA PETRAK 19.12.2019

---

# Lösungsvorschlag zu Übungsblatt 9

---

**Abgabe: Donnerstag 09.01.2020, 12:00 Uhr** (Postkasten der Veranstaltung und E-Mail an Tutor)

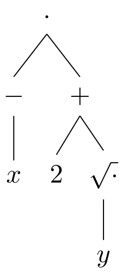
- Dieses Übungsblatt muss im Team abgegeben werden (Einzelabgaben sind nicht erlaubt!).
- Bitte zur Angabe von Namen, Übungsgruppe und Teamnummer das **Deckblatt** verwenden!
- Die **Zeitangaben** geben zur Orientierung an, wie viel Zeit für eine Aufgabe später in der Klausur vorgesehen wäre; gehen Sie davon aus, dass Sie zum jetzigen Zeitpunkt wesentlich länger brauchen und die angegebene Zeit erst nach ausreichender Übung erreichen.

* leichte Aufgabe / ** mittelschwere Aufgabe / *** schwere Aufgabe

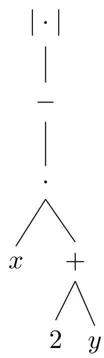
## Aufgabe 33 *(Terme, 18 Minuten)*

a) (*, Syntaxbäume zu Termen, 3 Minuten)
Geben Sie zu folgenden Termen in Infixnotation den jeweils zugehörigen Syntaxbaum an

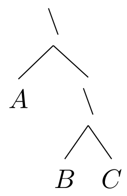
1. $((-x) \cdot (2 + \sqrt{y}))$



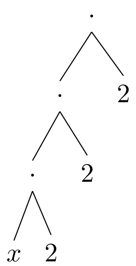

2. $|-(x \cdot (2 + y))|$



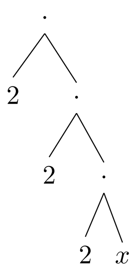

3. $(A \setminus (B \setminus C))$



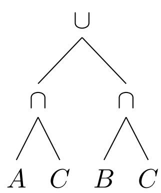

b) (*, Terme zu Syntaxbäumen, 3 Minuten)
Geben Sie zu folgenden Syntaxbäumen jeweils den zugehörigen Terme in Infixnotation an

1. 


$(((x \cdot 2) \cdot 2) \cdot 2)$

2. 


$(2 \cdot (2 \cdot (2 \cdot x)))$

3. 


$((A \cap C) \cup (B \cap C))$

c) (***, Strukturelle Induktion für Terme, 12 Minuten)
Beweisen Sie mit struktureller Induktion: *Ein Term, der Funktionen und unäre und binäre Operationen über $\mathbb{R}$ enthält, hat in der Infixnotation eine gerade Anzahl von Klammern*

**Sei $k(t)$ die Anzahl an Klammern eines Terms $t$.**

- **Induktionsanfang:**
  **Zu zeigen: Ist $t$ eine Konstante oder eine Variable, dann gilt** $k(t) \bmod 2 = 0$
  **Beweis:** $k(t) \bmod 2 = 0 \bmod 2 = 0$ **(Konstanten und Variablen enthalten keine Klammern)**

- **Induktionsschritt:**
  **Induktionsvoraussetzung (IV):** $t$ **und** $s$ **sind Terme mit** $k(s) \bmod 2 = 0$ **und** $k(t) \bmod 2 = 0$
  **Zu zeigen:**
  - **(IV)** $\Rightarrow k((op\, t)) \bmod 2 = 0$ **für jede unäre Operation** $op$
  - **(IV)** $\Rightarrow k((s\, op\, t)) \bmod 2 = 0$ **für jede binäre Operation** $op$
  - **(IV)** $\Rightarrow k(f(s)) \bmod 2 = 0$ **für jede Funktion** $f$

  **Beweis:**
  - $k((op\ t)) \bmod 2 = (2 + k(t)) \bmod 2 = ((2 \bmod 2) + (k(t) \bmod 2)) \bmod 2 \overset{(IV)}{=} (0 + 0) \bmod 2 = 0$ **für jede unäre Operation** $op$
  - $k((s\, op\, t)) \bmod 2 = (2 + k(t) + k(s)) \bmod 2 = ((2 \bmod 2) + (k(t) \bmod 2) + (k(s) \bmod 2)) \bmod 2 \overset{(IV)}{=} (0 + 0 + 0) \bmod 2 = 0$ **für jede binäre Operation** $op$
  - $k(f(s)) \bmod 2 = (2 + k(s)) \bmod 2 = ((2 \bmod 2) + (k(s) \bmod 2)) \bmod 2 \overset{(IV)}{=} (0 + 0) \bmod 2 = 0$ **für jede Funktion** $f$

## Aufgabe 34 *(Aussagenlogische Formeln, 22 Minuten)*

Die folgenden Aufgaben beschäftigen sich mit der **Äquivalenz von aussagenlogischen Formel** und der Anwendung von **Wahrheitstafeln** auf aussagenlogische Formeln.

Zwei aussagenlogische Formel $p$ und $q$ heißen **äquivalent**, wenn $p$ und $q$ für jede Belegung ihrer Variablen entweder beide erfüllt oder beide nicht erfüllt sind. Wir schreiben $p \equiv q$ für äquivalente Formeln $p$ und $q$.

a) (*, Aussagenlogische Formeln und Wahrheitstafeln, 4 Minuten)
Stellen Sie die Erfüllung der Formel $p \Rightarrow q$ in Abhängigkeit von der Erfüllung der Formeln $p$ und $q$ mit einer Wahrheitstafel dar.

*Hinweis:*

- *Bezüglich einer Belegung können $p$ und $q$ jeweils erfüllt oder nicht erfüllt sein.*
- *Mit einer Wahrheitstafel stellen Sie die Erfüllung von $p \Rightarrow q$ für alle möglichen Kombinationen der Erfüllung von $p$ und $q$ dar.*
- *Ist eine Formel erfüllt, so drücken sie das in der Wahrheitstafel durch den Wert 1 aus.*
- *Ist eine Formel nicht erfüllt, so drücken sie das in der Wahrheitstafel durch den Wert 0 aus.*

| $p$ | $q$ | $p \Rightarrow q$ |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |

b) (*, Beweis mit Wahrheitstafel, 6 Minuten)
Beweisen Sie mit Hilfe einer Wahrheitstafel das **Distributivgesetz** für aussagenlogische Formeln: *Für beliebige aussagenlogische Formeln $p, q, r$ gilt $(p \vee q) \wedge r \equiv (p \wedge r) \vee (q \wedge r)$.*

*Hinweis:*

- *Bezüglich einer Belegung können $p, q, r$ jeweils erfüllt oder nicht erfüllt sein.*
- *Mit einer Wahrheitstafel zeigen Sie für alle möglichen Kombinationen der Erfüllung von $p, q, r$, dass $(p \vee q) \wedge r$ und $(p \wedge r) \vee (q \wedge r)$ entweder beide erfüllt, oder beide nicht erfüllt sind.*
- *Ist eine Formel erfüllt, so drücken sie das in der Wahrheitstafel durch den Wert 1 aus.*
- *Ist eine Formel nicht erfüllt, so drücken sie das in der Wahrheitstafel durch den Wert 0 aus.*

| $p$ | $q$ | $r$ | $p \vee q$ | $(p \vee q) \wedge r$ | $p \wedge r$ | $q \wedge r$ | $(p \wedge r) \vee (q \wedge r)$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

c) (***, Strukturelle Induktion für aussagenlogische Formeln, 12 Minuten)
Beweisen Sie mit struktureller Induktion: *Für jede aussagenlogische Formel $p$ gibt es eine äquivalente aussagenlogische Formel $q$, die weder $\wedge$ noch $\vee$ enthält.*

- **Induktionsanfang:**
  **Zu zeigen: Seien $R \subseteq T_1 \times \ldots \times T_n$ eine $n$-stellige Relation und $A_i$ ein Term vom Typ $T_i$ für $i \in \mathbb{N}$, $1 \leq i \leq n$. Dann gibt es eine zur aussagenlogischen Formel $(A_1, \ldots, A_n) \in R$ äquivalente aussagenlogische Formel $q$, die weder $\wedge$ noch $\vee$ enthält.**
  **Beweis: $(A_1, \ldots, A_n) \in R$ enthält selbst weder $\wedge$ noch $\vee$.**

- **Induktionsschritt:**
  **Induktionsvoraussetzung (IV): $p$ und $q$ sind aussagenlogische Formeln, die weder $\wedge$ noch $\vee$ enthalten.**
  **Zu zeigen:**
  **(1) (IV) $\Rightarrow$ Es gibt eine zu $p \wedge q$ äquivalente aussagenlogische Formel $r$, die weder $\wedge$ noch $\vee$ enthält.**
  **(2) (IV) $\Rightarrow$ Es gibt eine zu $p \vee q$ äquivalente aussagenlogische Formel $r$, die weder $\wedge$ noch $\vee$ enthält.**
  **(3) (IV) $\Rightarrow$ Es gibt eine zu $\neg p$ äquivalente aussagenlogische Formel $r$, die weder $\wedge$ noch $\vee$ enthält.**

**(4) (IV) $\Rightarrow$ Es gibt eine zu $p \Rightarrow q$ äquivalente aussagenlogische Formel $r$, die weder $\wedge$ noch $\vee$ enthält.**

**(5) (IV) $\Rightarrow$ Es gibt eine zu $p \Leftrightarrow q$ äquivalente aussagenlogische Formel $r$, die weder $\wedge$ noch $\vee$ enthält.**

**Beweis:**

- **Die Formeln aus (3), (4) und (5) enthalten nach (IV) selbst weder $\wedge$ noch $\vee$.**
- $p \vee q \equiv \neg(\neg p) \vee q \equiv (\neg p) \Rightarrow q$ **(Beweise hierfür: Wahrheitstafeln - optional, da Einzelschritte einfach).**
- $p \wedge q \equiv \neg((\neg p) \vee (\neg q)) \equiv \neg(p \Rightarrow (\neg q))$ **(Beweise hierfür: Wahrheitstafeln - optional, da Einzelschritte einfach).**

## Aufgabe 35 *(Prädikatenlogische Formeln, 30 Minuten)*

In den folgenden Aufgaben ist der Wertebereich jeder gebundenen Variable wie in der Vorlesung eingeführt als Teil der Formel anzugeben.

a) (*, 5 Minuten)
Entscheiden Sie, ob im Folgenden jeweils prädikatenlogische Formeln angegeben sind oder nicht (ohne Angabe einer Begründung):

1. $\exists x \in \mathbb{R}(\forall y \in \mathbb{R})$

   **nein (es fehlt die Formel zu den Quantoren)**

2. $\forall y \in \mathbb{Z}(y > 0)$

   **ja ($(y > 0)$ ist eine Formel, also auch $\forall y \in \mathbb{Z}(y > 0)$)**

3. $\exists x \in \mathbb{N}(x + 3)$

   **nein ($(x + 3)$ ist keine Formel)**

4. $\exists x \in \mathbb{R}(x \leq \forall y \in \mathbb{R})$

   **nein ($\forall y \in \mathbb{R}$ ist kein Term, also ist $x \leq \forall y \in \mathbb{R}$ keine Formel)**

5. $\exists x \in \mathbb{R}(0 < x < 1)$

   **nein ($0 < x < 1$ ist keine Formel)**

b) (**, 4 Minuten)
Geben Sie an, ob die folgenden prädikatenlogischen Formeln erfüllt sind oder nicht (ohne Angabe einer Begründung).

1. $\forall x \in \mathbb{N}(\exists y \in \mathbb{N}(y > x))$

   **erfüllt (zu jeder natürlichen Zahl gibt es eine größere natürliche Zahl)**

2. $\exists x \in \mathbb{N}(\forall y \in \mathbb{N}(y \geq x))$

   **erfüllt (für $x = 1$)**

3. $\forall x \in \mathbb{Z}((x > 0) \vee (x < 0))$

   **nicht erfüllt (für $x = 0$)**

4. $(\forall x \in \mathbb{N}(x \bmod 4 = 0)) \Rightarrow (\forall x \in \mathbb{N}(x \bmod 2 = 0))$

   **erfüllt (die Formel entspricht $\neg(\forall x(x \bmod 4 = 0)) \vee (\forall x(x \bmod 2 = 0))$; da nicht jede Zahl $x$ durch $4$ teilbar ist, ist $\neg(\forall x(x \bmod 4 = 0))$ erfüllt)**

c) (**, 6 Minuten)
Drücken Sie die folgenden natürlichsprachlichen Aussagen als prädikatenlogische Formeln aus.

1. $y$ ist der größte Teiler von $x$, der nicht gleich $x$ ist.
   $((x \bmod y) = 0)) \wedge (y \neq x) \wedge ((k > y \wedge (x \bmod k) = 0) \Rightarrow (k = x))$
2. Das Produkt von zwei ungeraden natürlichen Zahlen ist eine ungerade Zahl.
   $\forall x, y \in \mathbb{N}(((x \bmod 2 = 1) \wedge (y \bmod 2 = 1)) \Rightarrow (x \cdot y \bmod 2 = 1))$
3. Für jede natürliche Zahl $x$ gibt es eine natürliche Zahl $y$ mit $\log(y) > x$.
   $\forall x \in \mathbb{N}(\exists y \in \mathbb{N}(\log(y) > x))$

d) (**, 6 Minuten)
Finden Sie für die folgenden prädikatenlogischen Formeln möglichst prägnante und intuitive natürlichsprachliche Aussagen.

1. $\forall x, y \in \mathbb{Z}((x > 0) \wedge (y > 0) \Rightarrow (x \cdot y > 0))$.
   **Das Produkt von zwei positiven ganzen Zahlen ist positiv**
2. $\forall x \in \mathbb{R}(\exists z \in \mathbb{R}(x \cdot z = 1))$
   **Zu jeder reellen Zahl gibt es eine bzgl. der Mulitplikation inverse reelle Zahl (Anmerkung: trifft nicht für $x = 0$ zu)**
3. $\forall x \in \mathbb{Z}((x \mod 2 = 0) \vee (x \mod 2 = 1))$
   **Jede ganze Zahl ist gerade oder ungerade**

e) (***, 4 Minuten)
Zwei prädikatenlogische Formel $p$ und $q$ heißen **äquivalent**, wenn $p$ und $q$ für jede Belegung ihrer freien Variablen entweder beide erfüllt oder beide nicht erfüllt sind. Wir schreiben $p \equiv q$ für äquivalente Formeln $p$ und $q$.

Beweisen Sie unter Benutzung der auf Folie 571, Beispiel 10.48 dargestellten Technik: *Für eine beliebige prädikatenlogische Formel $p$ gilt $\exists x(p) \equiv \neg(\forall x(\neg p))$*

**Beweise: ($\exists x(p)$ ist erfüllt) $\Leftrightarrow$ ($\neg(\forall x(\neg p))$ ist erfüllt)**
**($\exists x(p)$ ist erfüllt) $\overset{(2)}{\Leftrightarrow}$ ($p$ ist für eine Belegung von $x$ erfüllt) $\Leftrightarrow$ ($\neg p$ ist nicht für alle Belegungen von $x$ erfüllt) $\overset{(1)}{\Leftrightarrow}$ ($\neg(\forall x(\neg p))$ ist erfüllt)**

- (1)**: Definition von Erfüllbar für den All-Quantor**
- (2)**: Definition von Erfüllbar für den Existiert-Quantor**

f) (***, 5 Minuten)
Beweisen Sie unter Benutzung der auf Folie 564, Beispiel 10.42 dargestellten Technik für ein Alphabet $A$: $\forall u, v, w \in A^*$ *((u ist Teilwort von v) $\wedge$ (v ist Teilwort von w) $\Rightarrow$ (u ist Teilwort von w))*

*Hinweis: Benutzen Sie folgende Definition von Teilworten:*
*a ist Teilwort von b* $:\Leftrightarrow \exists x, y \in A^*(b = xay)$

**($u$ Teilwort von $v$) $\wedge$ ($v$ Teilwort von $w$)** $\overset{(1)}{\Rightarrow} \exists x, y \in A^*(v = xuy) \wedge \exists a, b \in A^*(w = avb)$ $\Rightarrow \exists x, y, a, b \in A^*(w = axuyb) \Rightarrow \exists m, n \in A^*(w = num)$ **(wähle $n = ax$ und $m = yb$)** $\overset{(1)}{\Rightarrow}$ **($u$ Teilwort von $w$)**

- (1)**: Definition von Teilworten**

## Aufgabe 36 *(Problemspezifikationen, 27 Minuten)*

a) (15 Minuten)
Geben Sie für jede der folgenden natürlichsprachlichen Problemstellungen eine formale Problemspezifikation an.

1. (*, 3 Minuten)
   Berechne die größte ganze Zahl, die kleiner ist als eine reelle Zahl $x$.

   **Eingabe:** $x \in \mathbb{R}$
   **Ausgabe:** $z \in \mathbb{Z}$
   **Funktionaler Zusammenhang:** $z = \lfloor x \rfloor - 1$

2. (*, 4 Minuten)
   Berechne den Umfang eines Kreises mit reellem Radius $r$, wobei der Rechenfehler höchstens $2^{-10}$ beträgt.

   **Eingabe:** $r \in \mathbb{R}, r > 0$
   **Ausgabe:** $u \in \mathbb{R}$
   **Funktionaler Zusammenhang:** $|u - 2 \cdot \pi \cdot r| \leq 2^{-10}$

3. (**, 4 Minuten)
   Sei $A$ ein geordnetes Alphabet und $<$ die lexikographische Ordnung auf $A^*$. Bestimme für Wörter $u, v \in A^*$, ob $u < v$, $u = v$ oder $u > v$.

   **Eingabe:** $u, v \in A^*$
   **Ausgabe:** $n \in \{-1, 0, 1\}$
   **Funktionaler Zusammenhang:** $((n = -1) \wedge (u < v)) \vee ((n = 0) \wedge (u = v))) \vee ((n = 1) \wedge (u > v))$

4. (**, 4 Minuten)
   Sei $A$ ein Alphabet. Berechne, wie oft ein Zeichen $a \in A$ in einer Zeichenfolge $a_1 \dots a_n \in A^*$ vorkommt.

   **Eingabe:** $a_1 \dots a_n \in A^*, a \in A$
   **Ausgabe:** $k \in \mathbb{N}_0$
   **Funktionaler Zusammenhang:** $k = |\{i \mid a_i = a\}|$

b) (12 Minuten)
Welches Problem löst jeweils die folgende C-Funktion? Geben Sie eine zugehörige formale Problemspezifikation an.

1. (*, 3 Minuten)
   Die Funktion `strlen` aus `string.h`.

   **Eingabe**: $w \in ASCII^*$, $ASCII =$ Menge der ASCII-Zeichen
   **Ausgabe**: $k \in \mathbb{N}_0$
   **Funktionaler Zusammenhang**: $k = |w|$ (oder $k = |w| - 1$, falls man davon ausgeht, dass die abschließende binäre Null zu $w$ gehört; allerdings ist das ein technisches prgrammiersprachenspezifisches Detail, das man als solches bei der Problemspezifikation in der Regel nicht berücksichtigt.)

2. (*, 4 Minuten)
   Die Funktion `sqrt` aus `math.h`.

   **Eingabe**: $x \in \mathbb{R}$
   **Ausgabe**: $y \in \mathbb{R}$
   **Funktionaler Zusammenhang**: $x \geq 0 \Rightarrow y = \sqrt{x}$

3. (**, 5 Minuten)
   Die Funktion `isdigit` aus `ctype.h`.

   **Eingabe**: $c \in ASCII$, $ASCII =$ Menge der ASCII-Zeichen
   **Ausgabe**: $k \in \mathbb{Z}$
   **Funktionaler Zusammenhang**: $((k \neq 0) \wedge (a(c) \geq 48) \wedge (a(c) \leq 57)) \vee ((k = 0) \wedge ((a(c) < 48) \vee (a(c) > 57)))$ ($a(c)$ bezeichnet den ACSII-Code von $c$)