Übungen ab 4.11.2011!

1.6. Strukturelle Induktion

Bei Beweisen über Formeln kann über die Formellänge gehen. Oft intuitiver: Über den Aufbau der Formeln.

<u>Bsp</u>

Aussagenlogische Formeln:

IA → A – eine Variable A

IS $\begin{cases} (F \wedge F') & \text{– wenn } F \text{ und } F' \text{ aussagenlog. Formeln sind} \\ (F \vee F') & \text{– ''} \\ \neg F & \text{– wenn } F \text{ eine aussagenlog. Formel ist} \end{cases}$

Beh. In jeder aussagenlog. Form G gibt es genauso viel öffnende wie schließende Klammern.

IA: $G = A$ ✓

IS: 1. $G = (F \wedge F')$, stimmt, da die Beh. für F und F' nach IV gilt und wir eine öffnende und eine schließende Klammer dazu bekommen.

2. $G = (F \vee F')$, analog wie 1

3. $G = \neg F$, stimmt da die Beh. für F stimmt. ◻

Oft werden die Bew. nicht explizit ausbuchstabiert!

1.7 Beweis durch Verschärfung

Oft sieht man: "Es genügt zu zeigen ...."

Statt einer Beh. A beweisen wir B, wobei

B $\Rightarrow$ A gilt. D.h. wir beweisen eine stärkere Aussage.

Wird oft bei Ungleichungen $f(x) \leq g(x)$ benutzt.

Bsp:

Beh: $x^2 + \sin(x) \leq (x+1)^2 - \cos(x)$ für $x \geq 1/2$

$$x^2 + \sin(x) \leq x^2 + 1 \leq (x+1)^2 - 1 \leq (x+1)^2 - \cos(x)$$

$$x^2 + 1 \leq (x^2 + 2x + 1) - 1$$

$$x^2 + 1 \leq x^2 + 2x \quad \text{wahr } x \geq \frac{1}{2}$$

1.8 Beweis durch Abschwächung (OBdA)

Man beweist eine schwächere Aussage, die aber oft nur eine notationelle Vereinfachung

Bsp: OBdA nehmen wir an der Graph habe n Knoten und die Knoten sind von 1 bis n durchnummeriert.

Bsp: Gegeben drei nat. Zahlen a, b, c. **OBdA** $a \leq b \leq c$ (da wir die Zahlen ja umbenennen können)

1.9. Beweis durch Konstruktion

Man zeigt eine Beh. indem man ein formales Objekt konstruiert, das die gewünschte Eigenschaften hat.

<u>Bsp:</u>

<u>Beh.</u>: KNF Formeln, bei denen jede Klausel <u>höchstens 1 pos. Literal</u> und <u>mind. 1 neg. Literal</u> enthält, sind immer erfüllbar.

<u>Bew</u>: $K_1 \wedge K_2 \wedge \dots \wedge K_n$

$K_i = \cancel{A}_1 \vee L_2 \vee L_3 \dots \vee L_m$

$A \vee \neg B \vee \neg C \vee \neg D \dots \vee \neg E$

$\cancel{A}$

$\neg A$

$\neg A \vee B \qquad \neg A \vee \neg B$

$\neg A \vee \neg B \vee C$

$\neg A \vee \neg B \vee \neg C \vee D \qquad \neg A \vee \neg B \vee \neg C \vee \dots$

Konstruktion einer Belegung:

Alle aussagenlog. Var. sollen den Wert "false" erhalten.

→ alle Klauseln werden wahr

→ d.h. es gibt mind. eine erfüllende Belegung der Formel

2. Automaten und formale Sprache

<u>Def</u> (Alphabet)

Alphabet: bel. endliche nicht-leere Menge. Die Elemente werden Symbole oder Zeichen genannt.

<u>Def</u> (Worte)

Gegeben ein Alphabet Σ, ein <u>Wort</u> über Σ ist eine endliche Folge von Symbolen $\langle x_1, \ldots, x_n \rangle$, geschrieben $x_1 x_2 \ldots x_n$. Das <u>leere Wort</u> wird mit ε bezeichnet. Σ^* ist die Menge aller Wörter über Σ.

<u>Bsp.</u>: $\Sigma = \{a, b, c\}$

$$\Sigma^* = \{\varepsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, \ldots\}$$

Def (Konkatenation)

Für zwei Worte $v, w \in \Sigma^*$ ist mit $v \cdot w$ die Konkatenation von v und w genannt. D.h. $v = x_1 x_2 \dots x_n$ und $w = y_1 y_2 \dots y_m$ ist, dann ist

$$v \cdot w = x_1 x_2 \dots x_n y_1 y_2 \dots y_m .$$

Bem.: Oft wird der Punkt weggelassen

Bem.: $(\Sigma, \cdot)$ ist eine Halbgruppe mit neutr. El.

Def. (n-fache Produkt)

$$w^n = \underbrace{w \cdot w \cdot \dots \cdot w}_{n\text{-fach}} \quad \text{für } w \in \Sigma^*, n \in \mathbb{N}$$

Def. (Länge eines Wortes)

$|w|$ bezeichnet die Anzahl der Zeichen in w

Def (formale Sprache)

Eine formale Sprache L über einem Alphabet Σ ist eine bel. Teilmenge von Σ^*.

Bsp: $\Sigma = \{a, b\}$

$L \subseteq \Sigma^*$ seien die Worte, in denen gleich viele a's und b's vorkommen.

$L = \{\varepsilon, ab, ba, abba, abab, aabb, \ldots\}$

$abb \notin L$

Seien L und L' Sprache über Σ. Dann sind $L \cap L'$, $L \cup L'$ und $L - L'$ wohldefiniert.

Def (Konkatenation von Spr.)

Seien L und L' Spr. über Σ. Dann ist

$L \cdot L' = \{ww' \mid w \in L, w' \in L'\}$.

Def (n-fache Produkt von Spr.)

Sei L Spr. über Σ, Dann ist

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

$$L^n = L \cdot L^{n-1} \text{ für } n \geq 1$$

$$L - L' = L \cap \overline{L'} = \{w \in L \mid w \notin L'\}$$

Zur endl. Beschreibung von unendl. Sprache setzt man Grammatiken und Automaten ein.

2.1.1 Grammatiken

- Terminalsymbolen (genau die Symb. aus Σ)
- Variablensymbole (Hilfssymbole)
- Startsymbol
- Produktionsregeln: LS → RS
  mit der Bedeutung: wenn LS vorhanden, dann darf LS durch RS ersetzt werden.

<u>Idee</u>: Alle so erzeugbaren Worte gehören zu der Spr.

<u>Bsp</u>: $\Sigma = \{a, b\}$, $P = \{S \to \varepsilon, S \to ab, S \to ba, S \to SS, S \to aSb, S \to bSa\}$

$V = \{S\}$

Startsymbol S

$S \to \underline{S}S \to \underline{a\underline{S}b}S \to a\underline{ab}b\underline{S} \to aabbba$

<Satz> → <Subjekt> <Prädikat> <Objekt>
<Subjekt> → <Artikel> <Attribut> <Substantiv>
<Artikel> → ε
<Artikel> → der
<Artikel> → die
<Artikel> → das
<Attribut> → ε
<Attribut> → <Adjektiv>
<Attribut> → <Adjektiv> <Attribut>
<Adjektiv> → kleine
<Adjektiv> → bissige
<Adjektiv> → große
<Substantiv> → Hund
<Substantiv> → Katze
<Prädikat> → jagt
<Objekt> → <Artikel> <Attribut> <Substantiv>