2. Großübung (13.5.19)

Inhalt:
- Normalformen
- Bsp. zum Kompaktheitssatz
- Bsp.: Deduktives System

Normalformeln

Wir gehen im Folgenden davon aus, dass die Formeln nur die Operatoren $\wedge, \vee, \neg$ beinhalten. Um dies herzustellen, nutze die folgenden Regeln:

$$A \to B \;\models\!\dashv\; \neg A \vee (A \wedge B) \;\models\!\dashv\; \neg A \vee B$$
$$A \Leftrightarrow B \;\models\!\dashv\; A \to B \wedge B \to A \;\models\!\dashv\; (\neg A \wedge \neg B) \vee (A \wedge B)$$
$$\top \;\models\!\dashv\; p \vee \neg p \quad \text{(für beliebige Variable } p)$$
$$\bot \;\models\!\dashv\; p \wedge \neg p \quad (\text{" \quad " \quad " } p)$$

Hier nutzen wir aus, dass logische Äquivalenz eine Kongruenz ist: Wir dürfen in eine Formel durch eine logisch äquivalente Formel ersetzen und erhalten eine (zur Gesamtformel) logisch äquivalente Formel.

Formal:

$$A \models\!\dashv B \Rightarrow \forall C \in F(\mathcal{A}): \neg A \models\!\dashv \neg B$$
$$A \wedge C \models\!\dashv B \wedge C$$
$$A \vee C \models\!\dashv B \vee C$$
$$A \to C \models\!\dashv B \to C$$

usw.

Eine Formel ist in <u>Negationsnormalform (NNF)</u>, wenn nur Variablen negiert sind:

A in NNF: Für jede Teilformel der Form $\neg B$ gilt $B \equiv p$ für eine Variable p.

<u>Bsp:</u> $\neg(p \wedge \neg(q \vee r)) \vDash\dashv \neg p \vee \neg\neg(q \vee r) \vDash\dashv \neg p \vee (q \vee r)$

$\neg(p \wedge (q \vee r)) \vDash\dashv \neg p \vee \neg(q \vee r) \vDash\dashv \neg p \vee (\neg q \wedge \neg r)$

Das ist ein „∨", sorry

jeweils nicht in NNF

jeweils in NNF

<u>Lemma:</u> Zu jeder Formel gibt es eine logisch äquivalente Formel in NNF.

<u>Beweisidee:</u> Verwende die folgenden „Regeln", um Negationen „nach innen" zu schieben:

- $\neg(A \wedge B) \vDash\dashv \neg A \vee \neg B$
- $\neg(A \vee B) \vDash\dashv \neg A \wedge \neg B$
- $\neg\neg A \vDash\dashv A$

De Morgan

∎

Ein Literal L ist eine Variable p oder eine negierte Variable $\neg p$.

Eine Klausel $K \equiv L_1 \vee \ldots \vee L_k$ ist eine Disjunktion von ~~[illegible]~~ Literalen.

Eine Coklausel $C \equiv L_1 \wedge \ldots \wedge L_k$ ist eine Konjunktion...

Eine Formel in disjunktiver Normalform (DNF) ist eine Disjunktion von Coklauseln, $A \equiv C_1 \vee \ldots \vee C_m$.

Eine Formel in konjunktiver Normalform (KNF) ist eine Konjunktion von Klauseln, $A \equiv K_1 \wedge \ldots K_m$.

Idee: Operatoren sind sortiert

Außen $\longleftrightarrow$ Innen

KNF:	$\wedge$	$\vee$	$\neg$
DNF:	$\vee$	$\wedge$	$\neg$

Normalformen sind wichtig, wenn man Formeln speichern / algorithmisch behandeln möchte.

Theorem: Zu jeder Formel A gibt es B in DNF mit $A \mathrel{\models\!\!\!|} B$.

(Analoges Resultat für KNF.)

Beweis: Bringe A in NNF, verwende dann die Distributivgesetze,

$$A \wedge (B \vee C) \mathrel{\models\!\!\!|} (A \wedge B) \vee (A \wedge C)$$
$$A \vee (B \wedge C) \mathrel{\models\!\!\!|} (A \vee B) \wedge (A \vee C)$$

um die Operatoren zu „sortieren".

Formal: Ang. A ist in NNF (siehe Lemma)

Induktion nach (Struktur von) A

Basisfall $\underline{A \equiv p}$: ist in DNF ✓

" $\underline{A \equiv \neg p}$: ist in DNF ✓ ← Dies ist ein Basisfall, weil wir NNF annehmen

IV: Ang. zu B, C gibt es B', C' mit

$B \mathrel{\models\!\!\!|} B'$, $C \mathrel{\models\!\!\!|} C'$, B und C in DNF.

Induktionsschritt:

$\underline{A \equiv B \vee C}$: Es gilt $A \mathrel{\models\!\!\!|} B \vee C \overset{IV}{\mathrel{\models\!\!\!|}} B' \vee C'$.

Da B', C' in DNF ist auch $B' \vee C'$ in DNF (Disjunktion ist außen.)

$\underline{A \equiv B \wedge C}$: Analog: $A \mathrel{\models\!\!\!|} B' \wedge C'$. Keine DNF: „$\wedge$" außen.

Sei $B' \equiv \bigvee_{i=1,\dots,l} C_i$, $C' \equiv \bigvee_{j=1,\dots,m} C_j'$ für Coklauseln C_i / C_j'.

Es gilt $A \mathrel{\models\!\!\!|} B' \wedge C' \mathrel{\models\!\!\!|} \bigvee_{\substack{i=1,\dots,l \\ j=1,\dots,m}} \underbrace{(C_i \wedge C_j')}_{\text{Coklausel}}$ in DNF ■

Wiederholte Anwendung des Distributivgesetzes.

Wichtigstes algorithmisches Problem der Aussagenlogik: Erfüllbarkeit.

Erfüllbarkeit

Gegeben: $A \in \mathcal{F}(\mathcal{A})$.

Frage: Ist A erfüllbar (d.h. $\exists \varphi: \mathcal{A} \to \mathbb{B}$ mit $\hat{\varphi}(A) = 1$)?

Hier: Effizienter Algorithmus für Erfüllbarkeit von Formeln in DNF

erfüllbar (DNF A) mit $A \equiv C_1 \vee \dots \vee C_m$

```
for (i=1,…,m)
    if (erfüllbar(C_i) return true;
return false;
```

Idee: Disjunktion erfüllbar gdw. mind 1 Disjunkt erfüllbar.

mit

erfüllbar (Coklausel C) mit $C \equiv L_1 \wedge \dots \wedge L_k$

```
for (Variable p∈A)
    if (∃ j,j' ∈ {1,…,k}: L_j ≡ p, L_j' ≡ ¬p) return false;
return true
```

Idee: Coklausel erfüllbar gdw. sie keinen Widerspruch ($p \wedge \dots \wedge \neg p$) enthält.

Dieser Algorithmus entscheidet Erfüllbarkeit für Formeln in DNF in Linearzeit (d.h. wenn $|A| = n$, in $\sim n$ Schritten) Wenn man zum Speichern der Coklauseln eine geeignete Datenstruktur verwendet.

Betrachte den folgenden Algorithmus für beliebige Formeln:

erfüllbar (Formel $A \in f(\mathcal{A})$)

Berechne B mit B in DNF, $A \models\mid B$

return erfüllbar (B)

Löst der Algorithmus das Problem: Ja!

Ist er effizient?: Nein!

Problem: Umwandeln in DNF teuer

Im Induktionschritt wird die Formelgröße quadriert

Dies geschieht 1x pro Konjunktion außen

$\Rightarrow$ Wiederholte Quadrierung $\Rightarrow$ Exponentielles Wachstum

Man sieht dies auch am Distributivgesetz:

$$A \wedge (B \vee C) \models\mid (\underline{A} \wedge B) \vee (\underline{A} \wedge C)$$

Vorkommen von A verdoppelt.

$\Rightarrow$ B hat Größe bis zu $2^{|A|} = \underbrace{2 \cdot 2 \cdot \ldots \cdot 2}_{|A| \text{ mal}}$

(z.B. falls A in KNF)

$\Rightarrow$ erfüllbar(B) ist linear in $|B|$,

aber exponentiell in $|A|$, da $|B| = 2^{|A|}$ im Worst-Case.

Kompaktheitssatz

Äquivalente Formulierungen: Sei $\Sigma \subseteq F(\mathcal{A})$ Formelmenge
$A \in F(\mathcal{A})$ Formel

- Σ erfüllbar $\Leftrightarrow$ $\forall \Sigma' \subseteq \Sigma$ endliche Teilmenge: Σ' erfüllbar
- Σ unerfüllbar $\Leftrightarrow$ $\exists \Sigma' \subseteq \Sigma$ " " : Σ' unerfüllbar
- $\Sigma \models A$ $\Leftrightarrow$ $\exists \Sigma' \subseteq \Sigma$ " " : $\Sigma' \models A$

Beispiel zur Anwendung des K.S.

Lemma: Sei $\Sigma_0 \subseteq \Sigma_1 \subseteq \Sigma_2 \subseteq \dots$ eine unendliche Kette von Formelmengen mit $\Sigma_i \subseteq \Sigma_{i+1}$ $\forall i$.

Dann: $\bigcup_{i \in \mathbb{N}} \Sigma_i$ erfüllbar $\Leftrightarrow$ $\forall i \in \mathbb{N}$: Σ_i erfüllbar.

Beweis: „$\Rightarrow$" Einfach. Angenommen $\varphi: \mathcal{A} \to \mathbb{B}$ mit $\hat{\varphi}(A) = 1$ $\forall A \in \bigcup \Sigma_i$

$\Rightarrow \hat{\varphi}(A) = 1$ $\forall A \in \Sigma_i$ $\forall i \in \mathbb{N}$, da $\Sigma_i \subseteq \bigcup_{i \in \mathbb{N}} \Sigma_i$

$\Rightarrow \forall i: \Sigma_i$ erfüllbar.

„$\Leftarrow$" Ang. $\forall i \in \mathbb{N}: \Sigma_i$ erfüllbar.

Nehme $\bigcup \Sigma_i$ unerfüllbar an, erzeuge Widerspruch.

$\overset{K.S.}{\Longrightarrow}$ $\exists \Sigma' \subseteq \bigcup \Sigma_i$ endlich, Σ' unerfüllbar.

Sei $\Sigma' = \{A_1, -, A_k\}$.

$\forall j: A_j \in \bigcup \Sigma_i$. Also gibt es zu jedem A_j ein i_j mit $A_j \in \Sigma_{i_j}$.

Def. $i = \max_{j=1,\dots,k} i_j$.

Es gilt $\forall j: A_j \in \Sigma_i$, denn die $(\Sigma_i)_i$ sind eine aufsteigende Kette: $A_j \in \Sigma_{i_j} \Rightarrow A_j \in \Sigma_i$, denn $i_j \leq i$.

Also $\Sigma' \subseteq \Sigma_i$.

Σ' unerfüllbar $\Rightarrow \Sigma_i$ unerfüllbar $\lightning$ Widerspruch zu alle Σ_i erfüllbar $\blacksquare$

Brauchen wir die Bedingung, dass die Σ_i eine aufsteigende Kette bilden? Ja!

<u>Bsp.</u> Definiere $\Sigma_i = \{p_i \land \neg p_{i+1}\}$ $\forall i \in \mathbb{N}$.

Jedes Σ_i ist erfüllbar $\left(\varphi: \begin{array}{l} p_i \mapsto 1 \\ p_{i+1} \mapsto 0 \end{array}\right)$,

aber $\bigcup \Sigma_i$ enthält $\{p_0 \land \neg p_1, p_1 \land \neg p_2\}$ und ist damit unerfüllbar.

Deduktives System $K = (\mathcal{F}, R)$

„Formeln" → $\mathcal{F}$; R ← Beweisregeln

Objekte, über die K spricht

$T(K)$ = Theoreme von K $\subseteq \mathcal{F}$
= In K beweisbare Dinge

Bsp: $K = (\mathcal{F}, R)$ mit

Dies ist eine vollständige Operatormenge

$\mathcal{F}$ = Aussagenlogische Formeln über Operatoren $\{\to, \neg\}$

$R = \{ (; \neg A \to \neg A), (; (\neg A \to \neg B) \to (B \to A)), (A \to B, A; B) \mid A, B \in \mathcal{F} \}$

Axiome: Regeln mit leerer Prämisse

$\Rightarrow$ $\neg A \to \neg A \in T(K) \quad \forall A \in \mathcal{F}$
$(\neg A \to \neg B) \to (B \to A) \in T(K) \quad \forall A, B \in \mathcal{F}$

$(A \to B, A; B)$ Wenn die Prämisse bewiesen ist (also $A \to B \in T(K)$, $A \in T(K)$), dann auch die Konklusion (also $B \in T(K)$).

Oder andersrum: Um B zu beweisen, reicht es, $A \to B \in T(K)$ und $A \in T(K)$ zu beweisen.

Notation üblicherweise als Regelschemata der Form

$\dfrac{\text{Prämissen}}{\text{Konklusion}}$, im Beispiel:

$$\dfrac{\text{leere Prämisse}}{\neg A \to \neg A}\ \text{Ax.0} \qquad \dfrac{}{(\neg A \to \neg B) \to (B \to A)}\ \text{Ax.1}$$

$$\dfrac{A \to B \qquad A}{B}\ \text{MP (Modus Ponens)}$$

Bsp: Beweise $p \to p \in T(K)$

2 Mögliche Notationen:

- Sequentiell, von oben nach unten // besser zu lesen

Zeile	Formel	Regel, die angewandt wurde
0:	$\neg p \to \neg p$	Ax.0 (mit $A \equiv p$)
1:	$(\neg p \to \neg p) \to (p \to p)$	Ax1 (mit $A \equiv B \equiv p$)
2:	$p \to p$	MP(0 in 1)

$\square$

- Baumartig, von unten nach oben konstruiert // besser zu schreiben

$$\dfrac{\text{Ax0}\ \dfrac{}{\neg p \to \neg p}\ \ (\text{mit } A \equiv p) \qquad \dfrac{}{(\neg p \to \neg p) \to (p \to p)}\ \text{Ax1 mit } A \equiv B \equiv p}{p \to p}\ \text{MP (mit } A \equiv \neg p \to \neg p,\ B \equiv p \to p)$$

$\square$

Zunächst haben ~~dektive~~ deduktive Systeme nichts mit der Semantik der Aussagenlogik zu tun
(Deduktive Systeme ~~[illegible]~~ manipulieren lediglich Syntax.)

Unser Beispiel-System ist „sound" (korrekt)

$A \in T(K) \Rightarrow A$ ist eine Tautologie

Beweis: (Skizze) Induktion nach der Länge des Beweises.

Basisfall: Zeige, dass die Axiome sound sind:

$\forall A, B \in F$: $\neg A \to \neg A$, $(\neg A \to \neg B) \to (B \to A)$

sind Tautologien (Bew. durch Wahrheitstabelle)

Induktionsschritt: Zeige, dass MP sind ist:

Wenn $A, B \in F$, so dass $A \to B$, A Tautologien, dann ist auch B eine Tautologie.

$\square$

Unser Beispiel-System ist nicht „complete" (vollständig).

Es gibt Formeln $A \in F$, die Tautologien sind, aber nicht bewiesen werden können (also $A \notin T(K)$).

Beispielsweise $p \to (q \to p)$

(Hausaufgabe: Zeige dies! Beweise, dass $\forall A \in T(K)$ gilt, dass jede Variable gerade oft in A vorkommt.)

In der Vorlesung: System, dass sound & complete ist.