Logikkalküle: Deduktion und Refutation

Logikkalküle: rein syntaktische Formate für formale Beweise.

Formale Beweise: syntaktische Zeichenketten,
nach einfach nachprüfbaren syntaktischen Regeln aufgebaut
(Regelsystem: *Kalkül*).

Ableitung: Erzeugung von (regelkonformen) formalen Beweisen.

Korrektheit nur semantisch korrekte Sachverhalte sind formal beweisbar (ableitbar).

Vollständigkeit jeder semantisch korrekte Sachverhalt ist formal beweisbar (ableitbar).

Resolution: ein *Widerlegungskalkül* für die *Unerfüllbarkeit* von KNF-Formeln.

Sequenzenkalkül: ein *Deduktionskalkül* für Allgemeingültigeit beliebiger AL-Formeln.

KNF in Klauselform

KNF: Konjunktionen von Disjunktionen von Literalen.
Notation: L für Literal; $\overline{L}$ für komplementäres Literal; $\overline{L} \equiv \neg L$.

Klausel: endliche Menge von Literalen
$C = \{L_1, \ldots, L_k\}$ steht für $\bigvee C \equiv L_1 \vee \ldots \vee L_k$
$\square$ steht für die leere Klausel.
Erinnerung: $\square \equiv \bigvee \emptyset \equiv 0$.

Klauselmenge: Menge von Klauseln
$K = \{C_1, \ldots, C_\ell\}$ steht für $\bigwedge K \equiv C_1 \wedge \ldots \wedge C_\ell$
Erinnerung: $\bigwedge \emptyset \equiv 1$.

endliche Klauselmengen $\approx$ KNF-Formeln

Resolutionskalkül arbeitet mit KNF in Klauselform
Ableitungsziel: Nachweis der Unerfüllbarkeit einer geg. Klauselmenge durch Ableitung der leeren Klausel $\square$

Resolution

Beispiele: $L, \overline{L} \in C \;\Rightarrow\; C \equiv 1$ allgemeingültig.

$C \equiv 1 \;\Rightarrow\; K \equiv K \setminus \{C\}$.

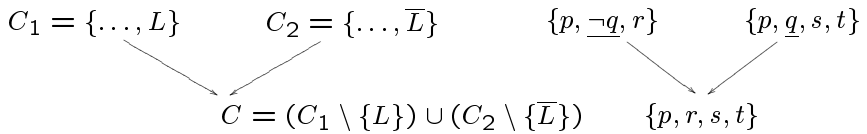
$\square \in K \;\Rightarrow\; K \equiv 0$ unerfüllbar.

Resolventen und Resolutionslemma

$L \in C_1, \overline{L} \in C_2 \;\Rightarrow\; \{C_1, C_2\} \equiv \{C_1, C_2, C\}$

wobei $C = \underbrace{(C_1 \setminus \{L\}) \cup (C_2 \setminus \{\overline{L}\})}_{\textit{Resolvente}}$

Diagrammatisch:



Resolutionslemma

Seien $C_1, C_2 \in K$, C Resolvente von C_1 und C_2.
Dann ist $K \equiv K \cup \{C\}$. [also $K \models C$]

$\mathbf{Res}(K)$ und $\mathbf{Res}^*(K)$

$\mathrm{Res}(K) := K \cup \{C\colon\ C$ Resolvente von Klauseln in $K\ \}$.

Klausel C heißt (im Resolutionskalkül) *ableitbar* aus K, gdw.
$C \in \underbrace{\mathrm{Res} \cdots \mathrm{Res}}_{n\text{-mal}}(K)$ für ein $n \in \mathbb{N}$.

$\mathrm{Res}^*(K)$: die Menge aller aus K ableitbaren Klauseln.

Korrektheit / Vollständigkeit

Korrektheit: $\square \in \mathrm{Res}^*(K) \;\Rightarrow\; K \equiv 0$ unerfüllbar. [$\rightarrow$ Lemma]

Vollständigkeit: K unerfüllbar $\;\Rightarrow\; \square \in \mathrm{Res}^*(K)$.

Resolutionskalkül: Vollständigkeit

Z.z.: K über $\mathcal{V}_n = \{p_1, \ldots, p_n\}$ unerfüllbar $\quad \Rightarrow \quad \square \in \text{Res}^*(K)$.

Beweis durch Induktion über n.

Induktionsschritt von n nach $n+1$

Aus $K = \{C_1, \ldots, C_k\}$ über $\mathcal{V}_{n+1}$ gewinne K_0 und K_1 über $\mathcal{V}_n$:

$K_0 \equiv K \cup \{\{\neg p_{n+1}\}\}$ $\qquad$ $K_1 \equiv K \cup \{\{p_{n+1}\}\}$ $\qquad$ (wie?)

K unerfüllbar $\Rightarrow$ K_0 und K_1 unerfüllbar
$\Rightarrow$ $\square \in \text{Res}^*(K_0)$ und $\square \in \text{Res}^*(K_1)$.

Dann ist $\square \in \text{Res}^*(K)$ oder $\begin{cases} \{p_{n+1}\} \in \text{Res}^*(K) \\ \text{und} \\ \{\neg p_{n+1}\} \in \text{Res}^*(K) \end{cases}$

und demnach jedenfalls $\square \in \text{Res}^*(K)$.

Resolutionsalgorithmus

breadth-first-search, Breitensuche

Eingabe: K [Klauselmenge, endlich]
$R := K$
WHILE $\big(\text{Res}(R) \neq R$ and $\square \notin R\big)$ DO $R := \text{Res}(R)$ OD
IF $\square \in R$ THEN output "unerfüllbar"
ELSE output "erfüllbar"

Beweis im Resolutionskalkül

Ableitungsbaum für $\square$:
Knoten mit Klauseln beschriftet
$\square$ an der Wurzel
Resolventen an binären Verzweigungen
Klauseln aus K an den Blättern

Hornklauseln

- interessanter Spezialfall für KI Anwendungen,
- AL-HORN-SAT-Problem effizient entscheidbar
- logische Programmierung (Prolog: FO Horn-Formeln)

Hornklausel: Klausel mit *höchstens einem positiven* Literal
z.B. $C = \{\neg q_1, \ldots, \neg q_r, q\} \equiv \big(q_1 \wedge \ldots \wedge q_r\big) \rightarrow q$;
auch $\square$ ist Hornklausel.

Spezialfälle: C besteht nur aus positivem Literal: *positiv*.
C ohne positive Literale: *negativ*.

Beobachtungen:

Mengen von negativen Hornklauseln trivial erfüllbar ($p_i \mapsto 0$).
Mengen von nicht-negativen Hornklauseln besitzen eindeutige *minimale* erfüllende Interpretationen.

Effizienter Horn-Erfüllbarkeitstest: Grundidee

H Hornklauselmenge; $H^- \subseteq H$ negative Klauseln in H
$H_0 := H \setminus H^-$ nicht negative Klauseln

1. Schritt: Berechne minimale Interpretation $\mathfrak{I}_0 \models H_0$.

2. Schritt: Prüfe, ob $\mathfrak{I}_0 \models H^-$.

Korrektheit

$\mathfrak{I}_0 \models H^- \quad \Rightarrow \quad \mathfrak{I}_0 \models H$.

$\mathfrak{I} \models H \quad \Rightarrow \quad \mathfrak{I} \models H_0$, also $\mathfrak{I}_0 \leqslant \mathfrak{I}$.
$\mathfrak{I} \models H^- \Rightarrow \mathfrak{I}_0 \models H^-$. Also $\mathfrak{I}_0 \models H$.