AL und Boolesche Funktionen → Abschnitt 3

$\mathcal{B}_n$: die Menge aller n-stelligen Booleschen Funktionen

$$\begin{array}{rcl} f : \mathbb{B}^n & \longrightarrow & \mathbb{B} \\ (b_1, \ldots, b_n) & \longmapsto & f(b_1, \ldots, b_n) \end{array}$$

speziell für $\varphi \in \mathrm{AL}_n$:

$$\left.\begin{array}{rcl} f_\varphi : \mathbb{B}^n & \longrightarrow & \mathbb{B} \\ (b_1, \ldots, b_n) & \longmapsto & \varphi[b_1, \ldots, b_n] \end{array}\right\} \in \mathcal{B}_n$$

beachte: $f_\varphi = f_\psi$ gdw. $\varphi \equiv \psi$

also: $\begin{array}{rcll} \mathrm{AL}_n / \equiv & \longrightarrow & \mathcal{B}_n & \text{injektiv!} \\ [\varphi]_\equiv & \longmapsto & f_\varphi & \end{array}$

Fragen:

- wieviele n-stellige Boolesche Funktionen gibt es?; $|\mathcal{B}_n| = ?$
- ist jedes $f \in \mathcal{B}_n$ durch AL-Formel $\varphi \in \mathrm{AL}_n$ darstellbar?

Disjunktive und konjunktive Normalformen, DNF, KNF

Nomenklatur: p bzw. $\neg p$ (für $p \in \mathcal{V}$) heißen *Literale*

Disjunktionen von Konjunktionen von Literalen: **DNF**-Formeln
Konjunktionen von Disjunktionen von Literalen: **KNF**-Formeln

"große" Konjunktion/Disjunktion (Schreibweisen):

für endliche Formelmenge $\Phi = \{\varphi_1, \ldots, \varphi_n\}$:

$\bigwedge \Phi := \bigwedge_{i=1}^n \varphi_i = \varphi_1 \wedge \ldots \wedge \varphi_n$

$\bigvee \Phi := \bigvee_{i=1}^n \varphi_i = \varphi_1 \vee \ldots \vee \varphi_n$

Konvention: auch *leere* Disjunktionen/Konjunktionen zulässig

mit der Interpretation: $\bigvee \emptyset \equiv 0$ (!)

$\bigwedge \emptyset \equiv 1$ (!)

Funktionale Vollständigkeit

Funktionale Vollständigkeit von AL_n für $\mathcal{B}_n$:

zu jedem $f \in \mathcal{B}_n$ existiert DNF-Formel $\varphi \in \mathrm{AL}_n$ mit $f = f_\varphi$.

($\Rightarrow$ bijektive Korrespondenz zw. $\mathcal{B}_n$ und $\mathrm{AL}_n/\equiv$)

Beweis:

betrachte $\varphi_f := \bigvee \{\varphi_{\mathbf{b}} : f(\mathbf{b}) = 1\}$

wo $\varphi_{\mathbf{b}} = \bigwedge \{p_i : b_i = 1\} \wedge \bigwedge \{\neg p_i : b_i = 0\}$

Korollar: Satz über DNF und KNF

zu $\varphi \in \mathrm{AL}_n$ existieren stets: $\begin{cases} \text{DNF-Formel } \varphi_1 \in \mathrm{AL}_n \text{ mit } \varphi_1 \equiv \varphi, \\ \text{KNF-Formel } \varphi_2 \in \mathrm{AL}_n \text{ mit } \varphi_2 \equiv \varphi. \end{cases}$

Dualität Kunjunktion/Disjunktion

→ Abschnitt 3.2

nützliche Umformungen/Rechenregeln

$\neg(\varphi_1 \wedge \varphi_2) \equiv \neg\varphi_1 \vee \neg\varphi_2$ verallgemeinert sich zu $\boxed{\neg\big(\bigwedge \Phi\big) \equiv \bigvee \Phi^\neg}$

wobei $\Phi^\neg := \{\neg\varphi : \varphi \in \Phi\}$

$\neg(\varphi_1 \vee \varphi_2) \equiv \neg\varphi_1 \wedge \neg\varphi_2$ verallgemeinert sich zu $\boxed{\neg\big(\bigvee \Phi\big) \equiv \bigwedge \Phi^\neg}$

für **KNF** $\longleftrightarrow$ **DNF:**

$$\boxed{\neg \underbrace{\bigwedge_{i=1}^{k} \big(\bigvee C_i\big)}_{\text{KNF}} \equiv \underbrace{\bigvee_{i=1}^{k} \big(\bigwedge C_i^\neg\big)}_{\text{DNF }(*)}}$$

$C_1, \ldots, C_k$ (endl.) Mengen von Literalen

* Doppelnegationen in den $C_i^\neg$ eliminieren

Beispiel für exponentiellen "blow-up"

$$\varphi_m = \varphi_m(p_1, \ldots, p_{2m}) \;:=\; \bigwedge_{i=1}^{m} \neg(p_{2i-1} \leftrightarrow p_{2i}) \;\in \mathrm{AL}_{2m}$$

- φ_m hat genau 2^m erfüllende Interpretationen in $\mathbb{B}^{2m}$
- KNF von Länge $\sim m$ (linear in m):
  $\varphi_m \equiv \bigwedge_{i=1}^{m}\big((p_{2i-1} \vee p_{2i}) \wedge (\neg p_{2i-1} \vee \neg p_{2i})\big)$
- DNF in Länge $\sim 2m2^m$ (exponentiell in m):
  $\varphi_m \equiv \bigvee\{\varphi_{\mathbf{b}} \colon \mathbf{b} \in \mathbb{B}^{2m}, \varphi_m[\mathbf{b}] = 1\}$
- **keine kürzere DNF:** keine kürzeren Disjunktionsglieder! / keine redundanten Disjunktionsglieder!

Vollständige Systeme von Junktoren → Abschnitt 3.3

Für $n \geqslant 1$ ist jede Funktion in $\mathcal{B}_n$ darstellbar durch AL_n-Formel, die nur die Junktoren $\neg$ und $\wedge$ (nur $\neg$ und $\vee$) benutzt.

Begr.: Eliminiere $\vee$ oder $\wedge$ mit $\begin{cases} \varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2) \\ \varphi_1 \wedge \varphi_2 \equiv \neg(\neg\varphi_1 \vee \neg\varphi_2) \end{cases}$

Systeme von Junktoren (Booleschen Funktionen)
mit dieser Eigenschaft heißen *vollständig*.

weitere Beispiele vollständiger Systeme:

- $\mid$ mit der Definition $p \mid q := \neg(p \wedge q)$ (NAND)
  benutze z.B.: $\neg p \equiv p \mid p$; $p \wedge q \equiv \neg(p \mid q) \equiv (p \mid q) \,\big|\, (p \mid q)$.
- $\to$ zusammen mit 0
  benutze z.B.: $\neg p \equiv p \to 0$; $p \vee q \equiv \neg p \to q \equiv (p \to 0) \to q$.

nicht vollständig sind z.B. $\begin{cases} \{\wedge, \vee\} & \text{(Monotonie);} \\ \{\to\} & (0 \in \mathcal{B}_n \text{ nicht darstellbar}). \end{cases}$

Kompaktheitssatz (Endlichkeitssatz) (Satz 4.1)

Erfüllbarkeit von unendlichen Formelmengen
hängt nur von je endlich vielen ab, i.d.S.d.

für alle $\Phi \subseteq \mathrm{AL}$ gilt:

$$\boldsymbol{\Phi} \text{ erfüllbar} \quad \Leftrightarrow \quad \text{jedes endliche } \boldsymbol{\Phi_0} \subseteq \boldsymbol{\Phi} \text{ erfüllbar} \qquad (*)$$

für alle $\Phi \subseteq \mathrm{AL}, \psi \in \mathrm{AL}$ gilt:

$$\boldsymbol{\Phi} \models \psi \quad \Leftrightarrow \quad \boldsymbol{\Phi_0} \models \psi \text{ für ein endliches } \boldsymbol{\Phi_0} \subseteq \boldsymbol{\Phi} \qquad (**)$$

Konsequenz:
Unerfüllbarkeit einer unendlichen Formelmenge lässt sich durch ein endliches Zertifikat nachweisen. (Warum?)

Bemerkung: Aussagen $(*)$ und $(**)$ sind äquivalent.

Kompaktheitssatz: Beweis → Abschnitt 4

für $\Phi \subseteq \mathrm{AL}(\mathcal{V})$, $\mathcal{V} = \{p_i : i \geqslant 1\}$

Sei jedes endliche $\Phi_0 \subseteq \Phi$ erfüllbar.

Konstruiere induktiv $\mathfrak{I}_0, \mathfrak{I}_1, \mathfrak{I}_2, \ldots$ so, dass für jedes n:

- $\mathfrak{I}_n$ eine $\mathcal{V}_n$-Interpretation ist.
- $\mathfrak{I}_{n+1}$ verträglich ist mit $\mathfrak{I}_n$: $\mathfrak{I}_{n+1}(p_i) = \mathfrak{I}_n(p_i)$ für $1 \leqslant i \leqslant n$.
- alle endlichen $\Phi_0 \subseteq \Phi$ erfüllbar sind durch $\mathfrak{I}$, die mit $\mathfrak{I}_n$ verträglich sind.

Dann ist $\mathfrak{I} \models \Phi$ für die Interpretation $\left\{ \begin{array}{rcl} \mathfrak{I} : \mathcal{V} & \longrightarrow & \mathbb{B} \\ p_n & \longmapsto & \mathfrak{I}_n(p_n) \end{array} \right.$

Frage: Wie kommt man von $\mathfrak{I}_n$ zu $\mathfrak{I}_{n+1}$?

Kompaktheitssatz: Konsequenzen

vgl. auch Skript u. Aufgaben

Lemma von König (Lemma 4.4)

Ein endlich verzweigter Baum mit unendlich vielen Knoten muss einen unendlichen Pfad haben. beachte Voraussetzung!

k-Färbbarkeit

Ein Graph ist genau dann k-färbbar, wenn jeder endliche Teilgraph k-färbbar ist.

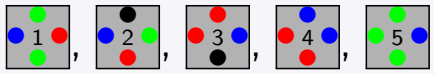
Domino-Parkettierungen

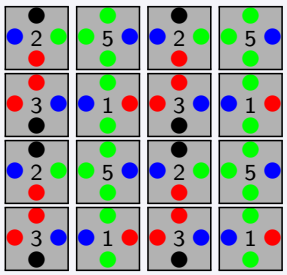
Ein endliches Domino-System erlaubt genau dann eine Parkettierung der Ebene, wenn sich beliebig große endliche Quadrate parkettieren lassen.

Domino-Parkettierung

ein interessantes, algorithmisch unentscheidbares Problem

> Zu gegebener Menge von Kacheln mit gefärbten Rändern:
> Kann man damit beliebig große Quadrate kacheln?

Beispiel: 

$\longrightarrow$


Mit AL-Kompaktheit lässt sich zeigen:

Ein endlicher Kachel-Satz erlaubt genau dann eine Parkettierung der unendlichen $\mathbb{N} \times \mathbb{N}$-Ebene (oder auch der $\mathbb{Z} \times \mathbb{Z}$-Ebene), wenn sich beliebig große endliche Quadrate parkettieren lassen. (wie?)

Lemma von König aus AL-Kompaktheit

Betrachte $\mathcal{T} = (V, E, \lambda)$ Baum mit

– Wurzel λ und abzählbar unendlicher Knotenmenge V,

– endlich verzweigter Kantenrelation E:
$E[u] = \{v \in V : (u, v) \in E\}$ endlich f.a. $u \in V$.

– Pfaden $\lambda \xrightarrow{E} \ldots \xrightarrow{E} u$ jeder endlichen Länge,
da sonst V endlich.

Lemma von König aus AL-Kompaktheit

Kodierung in $\mathrm{AL}(\mathcal{V})$ mit $\mathcal{V} := \{p_u : u \in V\}$:

$$\varphi_u := p_u \to \bigvee\{p_v : v \in E[u]\}$$

"wenn u gewählt wird,
dann auch mindestens ein direkter Nachfolger von u"

Für $\Phi := \{p_\lambda\} \cup \{\varphi_u : u \in V\}$ gilt:

- jedes endliche $\Phi_0 \subseteq \Phi$ ist erfüllbar, also auch Φ insgesamt.
- wenn $\mathfrak{I} \models \Phi$, so existiert ein unendlicher Pfad
$\lambda = u_0 \xrightarrow{E} u_1 \xrightarrow{E} u_2 \xrightarrow{E} \ldots$ mit $\mathfrak{I}(u_i) = 1$.

Bem.: mit $\varphi'_u := p_u \to$ "... genau ein direkter Nachfolger von u"
beschreibt jedes $\mathfrak{I} \models \Phi'$ *exakt einen* unendlichen Pfad.

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 *Allgemeingültigkeit* beliebiger AL-Formeln.

KNF in Klauselform → Abschnitt 5.1

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 $\bigvee C \equiv L_1 \vee \ldots \vee L_k$
$\Box$ steht für die leere Klausel.
Erinnerung: $\Box \equiv \bigvee \emptyset \equiv 0$.

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 $\Box$

Resolution → Abschnitt 5.2

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

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

$\Box \in K \;\Rightarrow\; K \equiv 0$ (unerfüllbar).

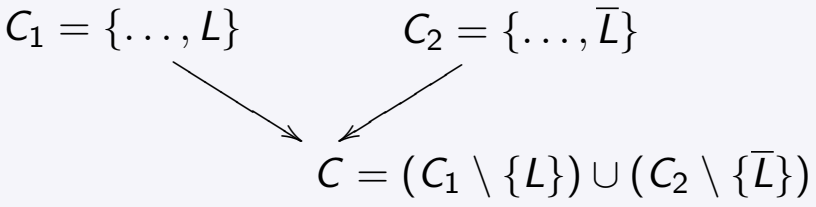
Resolventen und Resolutionslemma

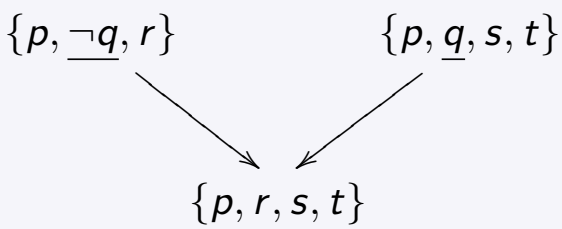
$$L \in C_1, \overline{L} \in C_2 \;\Rightarrow\; \{C_1, C_2\} \equiv \{C_1, C_2, C\}$$

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

Resolution

diagrammatisch:





Resolutionslemma (Lemma 5.5)

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

Res(K) und Res*(K)

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

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). [R-Lemma]

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

Resolutionskalkül: Vollständigkeit → Abschnitt 5.3

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

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 $\mathcal{V}_n$:

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

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

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

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

Resolutionsalgorithmus

breadth-first-search, Breitensuche

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

Beweis im Resolutionskakül

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

Hornklauseln → Abschnitt 5.4

- interessanter Spezialfall für KI Anwendungen,
- $\mathrm{AL\text{-}HORN\text{-}SAT}$-Problem effizient entscheidbar
- logische Programmierung (Prolog: FO Horn-Formeln)

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