

Logik und Diskrete Strukturen

Kapitel 6: Aussagenlogik

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für Theoretische Informatik und Theorembeweisen

Sommersemester 2026

Basierend auf Folien von PD Dr. Jan Johannsen

Übersicht

Logik

Syntax und Semantik

Beweiskalküle

SAT-Solving

Logik

Logik: die Lehre vom Schlussfolgern

Ursprünge in der Antike:

Syllogistik (Aristoteles), Geometrie (Euklid)

In der Neuzeit:

Formalisierung der Mathematik: Frege, Peano, Russell

20. Jahrhundert:

Unmöglichkeitsergebnisse: Gödel, Turing, Matijasevich

Anwendungen in der Informatik

Aussagenlogik: Syntax

Sei X eine Menge von Variablen.

Aussagenlogische Formeln über X sind induktiv definiert:

- ▶ die Konstanten 0 und 1 sind Formeln
- ▶ jede Variable $x \in X$ ist eine Formel
- ▶ ist F eine Formel, dann auch $\neg F$
- ▶ sind F und G Formeln, dann auch $(F \wedge G)$
- ▶ sind F und G Formeln, dann auch $(F \vee G)$
- ▶ sind F und G Formeln, dann auch $(F \rightarrow G)$

Notation: $(F \leftrightarrow G)$ steht für $((A \rightarrow B) \wedge (B \rightarrow A))$

Aussagenlogik: informelle Bedeutung

Variablen: elementare Aussagen, die wahr oder falsch sein können

- ▶ 1 ist wahr, 0 ist falsch
- ▶ $\neg A$ ist wahr gdw. A ist falsch
- ▶ $(A \wedge B)$ ist wahr gdw. A ist wahr und B ist wahr
- ▶ $(A \vee B)$ ist wahr gdw. A ist wahr oder B ist wahr
- ▶ $(A \rightarrow B)$ ist wahr gdw. B folgt aus A

Damit: $(A \leftrightarrow B)$ ist wahr gdw. A und B sind äquivalent.

Algebra der Wahrheitswerte

Die Menge der Wahrheitswerte ist $\mathbb{B} = \{0, 1\}$, mit

$0 \hat{=}$ falsch

$1 \hat{=}$ wahr

Auf der Menge $\mathbb{B}$ sind die Operationen $\neg, \wedge, \vee, \rightarrow$ definiert durch

x	y	$\neg x$	$x \wedge y$	$x \vee y$	$x \rightarrow y$
0	0	1	0	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	0	1	1	1

$\wedge$ und $\vee$ sind Infimum und Supremum in der Ordnung $0 \leq 1$.

$\rightsquigarrow$ Rechenregeln im Verband gelten

Aussagenlogik: Formale Semantik

Sein $\alpha : X \rightarrow \mathbb{B}$ eine Bewertung der Variablen X .

Der Wert $\llbracket F \rrbracket_\alpha$ einer Formel F ist rekursiv definiert:

- ▶ $\llbracket 0 \rrbracket_\alpha = 0$ und $\llbracket 1 \rrbracket_\alpha = 1$
- ▶ $\llbracket x \rrbracket_\alpha = \alpha(x)$
- ▶ $\llbracket \neg F \rrbracket_\alpha = \neg \llbracket F \rrbracket_\alpha$
- ▶ $\llbracket (F \wedge G) \rrbracket_\alpha = \llbracket F \rrbracket_\alpha \wedge \llbracket G \rrbracket_\alpha$
- ▶ $\llbracket (F \vee G) \rrbracket_\alpha = \llbracket F \rrbracket_\alpha \vee \llbracket G \rrbracket_\alpha$
- ▶ $\llbracket (F \rightarrow G) \rrbracket_\alpha = \llbracket F \rrbracket_\alpha \rightarrow \llbracket G \rrbracket_\alpha$

Definition: α erfüllt F , wenn $\llbracket F \rrbracket_\alpha = 1$ ist.

Man schreibt dafür auch $\alpha \models F$.

F ist **Tautologie**, wenn $\alpha \models F$ für alle α gilt.

F ist **erfüllbar**, wenn es eine Bewertung α gibt mit $\alpha \models F$.

F und G sind **äquivalent**, $F \equiv G$, wenn für jede Bewertung α gilt
 $\llbracket F \rrbracket_\alpha = \llbracket G \rrbracket_\alpha$

Lemma

F ist eine Tautologie gdw. $\neg F$ unerfüllbar ist.

Beweis.

F ist eine Tautologie gdw. $\llbracket F \rrbracket_{\alpha} = 1$ für alle Bewertungen α

gdw. $\llbracket \neg F \rrbracket_{\alpha} = 0$ für alle α

gdw. es kein α gibt, sodass $\llbracket \neg F \rrbracket_{\alpha} = 1$.

gdw. $\neg F$ unerfüllbar ist.



Lemma

$F \equiv G$ gdw. $F \leftrightarrow G$ eine Tautologie ist.

Beweis.

Offensichtlich, da $\llbracket F \rrbracket_\alpha = \llbracket G \rrbracket_\alpha$.



Lemma

F ist eine Tautologie gdw. $F \equiv 1$.

Beweis.

Für jede Bewertung α gilt $\llbracket F \rrbracket_\alpha = 1 = \llbracket 1 \rrbracket_\alpha$.



Substitution

Sind F und G Formeln und $x \in X$, dann bezeichnet $F[x := G]$ die Formel F , in der jedes Vorkommen von x durch G ersetzt ist.

Substitution

Lemma

$\llbracket F[x := G] \rrbracket_\alpha = \llbracket F \rrbracket_{\alpha'}$, wobei α' definiert ist durch

$$\alpha'(x) = \llbracket G \rrbracket_\alpha \text{ und } \alpha'(y) = \alpha(y) \text{ f\"ur alle } y \neq x$$

Beweis (durch starke Induktion über die Länge von F).

► **Induktionsschritt:** Die Fälle $F = 0$ und $F = 1$ sind trivial.

Falls $F = x$ haben wir $x[x := G] = G$. Dann $\llbracket G \rrbracket_\alpha = \alpha'(x)$.

Falls $F = F_1 \wedge F_2$ ist $F[x := G] = F_1[x := G] \wedge F_2[x := G]$.

Dann $\llbracket F[x := G] \rrbracket_\alpha = \llbracket F_1[x := G] \rrbracket_\alpha \wedge \llbracket F_2[x := G] \rrbracket_\alpha$

$= \llbracket F_1 \rrbracket_{\alpha'} \wedge \llbracket F_2 \rrbracket_{\alpha'} \quad (\text{durch die Induktionshypothese})$

$= \llbracket F \rrbracket_{\alpha'}.$

Die Fälle $\neg, \vee, \rightarrow$ sind analog.



Substitution

Lemma

Sind F, G, H Formeln mit $F \equiv G$, dann gilt

1. $F[x := H] \equiv G[x := H]$
2. $H[x := F] \equiv H[x := G]$

Beweis.

1. $\llbracket F[x := H] \rrbracket_\alpha = \llbracket F \rrbracket_{\alpha'}$ (wobei $\alpha'(x) = \llbracket H \rrbracket_\alpha$ und sonst $\alpha' = \alpha$)
 $= \llbracket G \rrbracket_{\alpha'}$
 $= \llbracket G[x := H] \rrbracket_\alpha.$
2. $\llbracket H[x := F] \rrbracket_\alpha = \llbracket H \rrbracket_{\alpha'}$ (wobei $\alpha'(x) = \llbracket F \rrbracket_\alpha = \llbracket G \rrbracket_\alpha$ und sonst $\alpha' = \alpha$)
 $= \llbracket H[x := G] \rrbracket_\alpha.$



Substitution

Beispiel: Aus

$$x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$$

folgt durch Substitution, dass

$$F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$$

für alle **Formeln** F, G, H

Negations-Normalform

Ein **Literal** ist eine Variable x oder eine negierte Variable $\neg x$.

Formeln in Negations-Normalform (NNF) sind definiert durch:

- ▶ 0, 1 und Literale sind in NNF
- ▶ sind F und G in NNF, dann auch $F \wedge G$ und $F \vee G$

Also: F ist in NNF, wenn keine Implikationen und Negationen nur vor Variablen vorkommen.

De Morgansche Gesetze: für alle Formeln A und B gilt

$$\neg(A \wedge B) \equiv (\neg A \vee \neg B) \quad \text{und} \quad \neg(A \vee B) \equiv (\neg A \wedge \neg B)$$

Negation

Für eine Formel F in NNF ist die Formel $\overline{F}$ in NNF definiert als

- ▶ $\overline{0} = 1, \quad \overline{1} = 0$

- ▶ $\overline{x} = \neg x$

- ▶ $\overline{\neg x} = x$

- ▶ $\overline{G \wedge H} = \overline{G} \vee \overline{H}$

- ▶ $\overline{G \vee H} = \overline{G} \wedge \overline{H}$

Negation

Lemma

Für F in NNF gilt $\overline{F} \equiv \neg F$.

Beweis (durch starke Induktion über die Länge von F).

► **Induktionsschritt:** Fall $F = 0$: $\overline{F} = 1 \equiv \neg 0 = \neg F$.

Fall $F = 1$: $\overline{F} = 0 \equiv \neg 1 = \neg F$.

Fall $F = x$: $\overline{F} = \neg x = \neg F$.

Fall $F = \neg x$: $\overline{F} = x \equiv \neg \neg x = \neg F$.

Fall $F = G \wedge H$: Durch die Induktionshypothese gilt $\overline{G} \equiv \neg G$ und $\overline{H} \equiv \neg H$.

Das heißt, $\overline{F} = \overline{G} \vee \overline{H} \equiv \neg G \vee \neg H \equiv \neg(G \wedge H) = \neg F$.

Fall $F = G \vee H$: Analog.



Konstruktion der NNF

Theorem

Für jede Formel F gibt es $n(F)$ in NNF mit $n(F) \equiv F$.

Beweis (durch starke Induktion über die Länge von F).

► **Induktionsschritt:** $F = x$, $F = 0$ oder $F = 1$ sind in NNF, also setzen wir $n(F) := F$.

Für $F = G \wedge H$ gibt es durch die Induktionshypothese $n(G)$ und $n(H)$ in NNF.

Dann ist $n(F) := n(G) \wedge n(H)$.

Analog für $F = G \vee H$.

Für $F = \neg G$ gibt es durch die Induktionshypothese $n(G)$ in NNF.

Dann ist $n(F) := \overline{n(G)}$.

Für $F = G \rightarrow H$ gibt es durch die Induktionshypothese $n(G)$ und $n(H)$ in NNF.

Dann ist $n(F) := \overline{n(G)} \vee n(H)$.



Disjunktive Normalform

Formeln in disjunktiver Normalform (DNF) sind definiert durch:

- ▶ Ein **Term** ist eine Konjunktion $a_1 \wedge \cdots \wedge a_k$ von Literalen
- ▶ Eine Formel in **DNF** ist eine Disjunktion $T_1 \vee \cdots \vee T_m$ von Termen

Jede Formel in DNF ist in NNF.

Disjunktive Normalform

Theorem

Für jede Formel F gibt es eine Formel G in DNF mit $G \equiv F$.

Beweis.

Seien $x_1, \dots, x_n$ die Variablen in F .

Für zwei Bewertungen α, β schreiben wir $\alpha =_n \beta$, wenn $\alpha(x_i) = \beta(x_i)$ für $i \in \{1, \dots, n\}$.

Für eine Variable x setzen wir $x^1 := x$ und $x^0 := \neg x$. Somit gilt $\alpha \models x^\varepsilon$ gdw. $\alpha(x) = \varepsilon$.

Für eine Bewertung α definieren wir $T_\alpha := x_1^{\alpha(x_1)} \wedge \dots \wedge x_n^{\alpha(x_n)}$.

Es gilt $\beta \models T_\alpha$ gdw. $\beta =_n \alpha$ für alle β .

Wir wählen Bewertungen $\alpha_1, \dots, \alpha_m$, sodass

(1) $\alpha_i \models F$ für alle $i \in \{1, \dots, m\}$ und (2) es gibt $j \in \{1, \dots, m\}$ mit $\alpha_j =_n \beta$ für alle $\beta \models F$.

Wir setzen $G := T_{\alpha_1} \vee \dots \vee T_{\alpha_m}$.

Dann $\beta \models F$ gdw. $\beta =_n \alpha_j$ für ein $j \in \{1, \dots, m\}$

gdw. $\beta \models T_{\alpha_j}$ für ein $j \in \{1, \dots, m\}$ gdw. $\beta \models G$. Somit $F \equiv G$. □

Konjunktive Normalform

Formeln in konjunktiver Normalform (KNF) sind definiert durch:

- ▶ Eine **Klausel** ist eine Disjunktion $a_1 \vee \cdots \vee a_k$ von Literalen
- ▶ Eine Formel in **KNF** ist eine Konjunktion $C_1 \wedge \cdots \wedge C_m$ von Klauseln

Jede Formel in KNF ist in NNF.

Konjunktive Normalform

Theorem

Für jede Formel F gibt es eine Formel H in KNF mit $H \equiv F$.

Beweis.

Wir konstruieren zuerst die Formel G in DNF, sodass $G \equiv \neg F$.

Dann ist $H := \overline{G}$ offensichtlich in KNF.

Es gilt $H = \overline{G} \equiv \neg G \equiv \neg \neg F \equiv F$.



Normalformen und Semantik

Semantische Eigenschaften sind für DNF und KNF einfach.

Für DNF:

- ▶ Ein Term $a_1 \wedge \dots \wedge a_k$ ist erfüllbar gdw. er keine komplementären Literale enthält, d.h. für alle $i, j \in \{1, \dots, k\}$ ist $a_i \neq \overline{a_j}$.
- ▶ Eine Formel in DNF ist erfüllbar gdw. einer ihrer Terme erfüllbar ist.

Für KNF:

- ▶ Eine Klausel $a_1 \vee \dots \vee a_k$ ist eine Tautologie gdw. sie komplementäre Literale enthält, d.h. es gibt $i, j \in \{1, \dots, k\}$ mit $a_i = \overline{a_j}$.
- ▶ Eine Formel in KNF ist eine Tautologie gdw. jede ihrer Klauseln eine Tautologie ist.

Exponentieller Blow-Up

Theorem

Es gibt Formeln F_n in KNF mit $4n$ Literalen, sodass jede Formel $G_n \equiv F_n$ in DNF mindestens $n2^n$ Literale enthält.

Beweis.

Sei $F_n := A_1 \wedge \dots \wedge A_n$, wobei $A_i = (x_{i0} \vee x_{i1}) \wedge (\neg x_{i0} \vee \neg x_{i1})$ für $i \in \{1, \dots, n\}$.

F_n ist offensichtlich in KNF und hat $4n$ Literale.

Sei $G_n = T_1 \vee \dots \vee T_m$ in DNF mit $G_n \equiv F_n$.

Ohne Beschränkung der Allgemeinheit ist jeder Term T_j erfüllbar für $j \in \{1, \dots, m\}$.

Wenn T_j weder x_{i0} noch x_{i1} enthält, dann gibt es $\alpha \models T_j$ mit $\alpha(x_{i0}) = \alpha(x_{i1}) = 0$.

Wenn T_j sowohl x_{i0} als auch x_{i1} enthält, dann gibt es $\alpha \models T_j$ mit $\alpha(x_{i0}) = \alpha(x_{i1}) = 1$.

In beiden Fällen gilt $\alpha \models G_n$, aber $\alpha \not\models F_n$.

Beobachtung: Das heißt, T_j enthält genau eine der Variablen x_{i0} und x_{i1} .

Exponentieller Blow-Up

Theorem

Es gibt Formeln F_n in KNF mit $4n$ Literalen, sodass jede Formel $G_n \equiv F_n$ in DNF mindestens $n2^n$ Literale enthält.

Beweis (Fortsetzung).

Für jeden Bitvektor $(\varepsilon_1, \dots, \varepsilon_n) \in \{0, 1\}^n$ gibt es einen Term T_j in G_n mit $x_{i\varepsilon_i}$ als Literal in T_j für jedes $i \in \{1, \dots, n\}$.

Sonst gäbe es $(\varepsilon_1, \dots, \varepsilon_n)$, sodass es für jeden Term T_j ein $i \in \{1, \dots, n\}$ gibt, sodass T_j das Literal $x_{i(1-\varepsilon_i)}$ enthält (durch die Beobachtung).

Wähle die Bewertung α mit $\alpha(x_{i\varepsilon_i}) = 1$ und $\alpha(x_{i(1-\varepsilon_i)}) = 0$ für alle $i \in \{1, \dots, n\}$.

Dann $\alpha \models F_n$, aber $\alpha \not\models G_n$. Widerspruch.

Daher gibt es mindestens 2^n Terme in G_n mit je n Literalen. □

Exponentieller Blow-Up

Theorem

*Es gibt Formeln F'_n in DNF mit $4n$ Literalen,
sodass jede Formel $H_n \equiv F'_n$ in KNF mindestens $n2^n$ Literale enthält.*

Beweis.

Die Formel $F'_n := \overline{F_n}$ ist offensichtlich in DNF. Es gilt $F'_n \equiv \neg F_n$.

Sei H_n in KNF mit $H_n \equiv F'_n$.

Dann ist $\overline{H_n}$ in DNF mit $\overline{H_n} \equiv \neg H_n \equiv \neg F'_n \equiv F_n$.

Daher enthält $\overline{H_n}$ mindestens $n2^n$ Literale.

Dies gilt auch für H_n .



Ende der 9. Vorlesung

Sequenzenkalkül

Eine **Sequenz** ist $\Gamma \vdash \Delta$, wobei Γ und Δ Listen von Formeln sind.

Γ ist **Antezedens** und Δ ist **Sukzedens** der Sequenz.

Für eine Sequenz gilt $\alpha \models \Gamma \vdash \Delta$, wenn

$\llbracket A \rrbracket_\alpha = 0$ für eine Formel A in Γ , oder
 $\llbracket B \rrbracket_\alpha = 1$ für eine Formel B in Δ

Zur Sequenz $A_1, \dots, A_m \vdash B_1, \dots, B_n$ äquivalente Formel:

$$(A_1 \wedge \dots \wedge A_m) \rightarrow (B_1 \vee \dots \vee B_n)$$

Schlussregeln

Die Schlussregeln des Sequenzenkalküls erlauben aus bekannten Sequenzen (**Prämissen**) eine neue Sequenz (**Konklusion**) herzuleiten.

Es gibt **strukturelle** und **logische** Schlussregeln.

Strukturelle Schlussregeln

Permutation:

$$\frac{\Gamma_1, A, B, \Gamma_2 \vdash \Delta}{\Gamma_1, B, A, \Gamma_2 \vdash \Delta} (Perm-L)$$

$$\frac{\Gamma \vdash \Delta_1, A, B, \Delta_2}{\Gamma \vdash \Delta_1, B, A, \Delta_2} (Perm-R)$$

Kontraktion:

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta} (Contr-L)$$

$$\frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A} (Contr-R)$$

Logische Schlussregeln

Konjunktion:

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} (\wedge\text{-L}) \quad \frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B} (\wedge\text{-R})$$

Disjunktion:

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta} (\vee\text{-L}) \quad \frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B} (\vee\text{-R})$$

Implikation:

$$\frac{\Gamma \vdash \Delta, A \quad \Gamma, B \vdash \Delta}{\Gamma, A \rightarrow B \vdash \Delta} (\rightarrow\text{-L}) \quad \frac{\Gamma, A \vdash \Delta, B}{\Gamma \vdash \Delta, A \rightarrow B} (\rightarrow\text{-R})$$

Negation:

$$\frac{\Gamma \vdash \Delta, A}{\Gamma, \neg A \vdash \Delta} (\neg\text{-L}) \quad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \Delta, \neg A} (\neg\text{-R})$$

Herleitungen und Beweise

Ein **Axiom** ist eine Sequenz $\Gamma \vdash \Delta$ mit $\Gamma \cap \Delta \neq \emptyset$.

Definition: Eine **Herleitung** einer Sequenz $\Gamma \vdash \Delta$ aus einer Menge $\mathcal{P}$ von Sequenzen (**Prämissen**) ist eine Folge $S_1, \dots, S_k$ von Sequenzen mit

- ▶ $S_k = \Gamma \vdash \Delta$
- ▶ Für jedes $i \leq k$ gilt eines der Folgenden:
 - ▶ S_i ist ein Axiom
 - ▶ $S_i \in \mathcal{P}$
 - ▶ es gibt $j < i$, sodass S_i Konklusion einer Regel mit Prämisse S_j ist
 - ▶ es gibt $j_1, j_2 < i$, sodass S_i Konklusion einer Regel mit Prämissen S_{j_1} und S_{j_2} ist

Ein **Beweis** für $\Gamma \vdash \Delta$ ist eine Herleitung von $\Gamma \vdash \Delta$ aus $\emptyset$.

Sequenzkalkül: Korrektheit

Lemma

*Für jede Schlussregel und jede Bewertung α gilt:
erfüllt α die Prämissen der Regel, dann auch die Konklusion.*

Beweis.

Wir beschränken uns auf die Regeln ($\wedge$ -R) und ($\neg$ -L).

Für die ($\wedge$ -R)-Regel: Wir nehmen an, dass $\alpha \models \Gamma \vdash \Delta, A$ und $\alpha \models \Gamma \vdash \Delta, B$ und werden zeigen, dass $\alpha \models \Gamma \vdash \Delta, A \wedge B$.

- ▶ **Fall 1:** $\llbracket F \rrbracket_\alpha = 0$ für ein $F \in \Gamma$. Offensichtlich.
- ▶ **Fall 2:** $\llbracket G \rrbracket_\alpha = 1$ für ein $G \in \Delta$. Offensichtlich.
- ▶ **Fall 3:** Sonst $\llbracket A \rrbracket_\alpha = 1$ und $\llbracket B \rrbracket_\alpha = 1$.
Also $\llbracket A \wedge B \rrbracket_\alpha = 1$.
Also $\alpha \models \Gamma \vdash \Delta, A \wedge B$.

Sequenzenkalkül: Korrektheit

Lemma

*Für jede Schlussregel und jede Bewertung α gilt:
erfüllt α die Prämissen der Regel, dann auch die Konklusion.*

Beweis (Fortsetzung).

Für die $(\neg\text{-L})$ -Regel: Wir nehmen an, dass $\alpha \models \Gamma \vdash \Delta, A$ und werden zeigen, dass $\alpha \models \Gamma, \neg A \vdash \Delta$.

- ▶ **Fall 1:** $\llbracket F \rrbracket_\alpha = 0$ für ein $F \in \Gamma$. Offensichtlich.
- ▶ **Fall 2:** $\llbracket G \rrbracket_\alpha = 1$ für ein $G \in \Delta$. Offensichtlich.
- ▶ **Fall 3:** Sonst $\llbracket A \rrbracket_\alpha = 1$.
Also $\llbracket \neg A \rrbracket_\alpha = 0$.
Also $\alpha \models \Gamma, \neg A \vdash \Delta$.



Sequenzenkalkül: Korrektheit

Theorem

Sei $S_1, \dots, S_n$ eine Herleitung von S_n aus den Prämissen $\mathcal{P}$ gegeben.

Dann gilt für jede Bewertung α :

wenn $\alpha \models P$ für alle $P \in \mathcal{P}$, dann auch $\alpha \models S_n$.

Beweis (durch starke Induktion über n).

► Induktionsschritt:

- **Fall 1:** S ist ein Axiom, also eine Tautologie.
- **Fall 2:** $S \in \mathcal{P}$. Dann gilt $\alpha \models S$ aus der Annahme.
- **Fall 3:** S ist die Konklusion mit Prämissen aus $\{S_1, \dots, S_{n-1}\}$.
Durch die Induktionshypothese gilt $\alpha \models S_i$ für alle $i \in \{1, \dots, n-1\}$.
Nach dem letzten Lemma gilt auch $\alpha \models S$.



Sequenzkalkül: Korrektheit

Korollar

Gibt es einen Beweis für $\vdash F$, dann ist F eine Tautologie.

Sequenzkalkül: Vollständigkeit

Lemma

*Für jede Schlussregel und jede Bewertung α gilt:
erfüllt α die Konklusion der Regel, dann auch alle Prämissen.*

Beweis.

Wir beschränken uns auf die $(\wedge\text{-R})$ - und $(\rightarrow\text{-L})$ -Regeln.

Für die $(\wedge\text{-R})$ -Regel: Wir nehmen an, dass $\alpha \models \Gamma \vdash \Delta, A \wedge B$ und werden zeigen, dass $\alpha \models \Gamma \vdash \Delta, A$ und $\alpha \models \Gamma \vdash \Delta, B$.

Der interessante Fall ist, wenn $\llbracket A \wedge B \rrbracket_\alpha = 1$.

Dann gilt auch $\llbracket A \rrbracket_\alpha = 1$ und $\llbracket B \rrbracket_\alpha = 1$.

Deshalb gilt $\alpha \models \Gamma \vdash \Delta, A$ und $\alpha \models \Gamma \vdash \Delta, B$.

Sequenzenkalkül: Vollständigkeit

Lemma

*Für jede Schlussregel und jede Bewertung α gilt:
erfüllt α die Konklusion der Regel, dann auch alle Prämissen.*

Beweis (Fortsetzung).

Für die $(\rightarrow -L)$ -Regel: Wir nehmen an, dass $\alpha \models \Gamma, A \rightarrow B \vdash \Delta$ und werden zeigen, dass $\alpha \models \Gamma \vdash \Delta, A$ und $\alpha \models \Gamma, B \vdash \Delta$.

Der interessante Fall ist, wenn $\llbracket A \rightarrow B \rrbracket_\alpha = 0$.

Dann gilt auch $\llbracket A \rrbracket_\alpha = 1$ und $\llbracket B \rrbracket_\alpha = 0$.

Deshalb gilt $\alpha \models \Gamma \vdash \Delta, A$ und $\alpha \models \Gamma, B \vdash \Delta$.



Sequenzenkalkül: Vollständigkeit

Theorem

Ist F eine Tautologie, dann gibt es einen Beweis für $\vdash F$.

Beweis.

Wir konstruieren Schritt für Schritt eine Folge $S_1, \dots, S_n$ von Sequenzen.

Wir fangen mit der einelementigen Folge mit $S_1 := \vdash F$ an. Wir lassen S_1 unmarkiert.

Sei S_i die erste unmarkierte Sequenz in der Folge.

- ▶ **Fall 1:** S_i ist ein Axiom. Markiere S_i als erledigt.
- ▶ **Fall 2:** S_i ist kein Axiom und besteht nur aus Variablen. Markiere S_i als erledigt.
- ▶ **Fall 3:** S_i ist die Konklusion einer logischen Regel.
Hänge die Prämissen als unmarkierte Sequenzen an und markiere S_i als erledigt.
- ▶ **Fall 4:** Sonst: Wende eine Permutationsregel an und markiere S_i als erledigt.

Sequenzenkalkül: Vollständigkeit

Theorem

Ist F eine Tautologie, dann gibt es einen Beweis für $\vdash F$.

Beweis (Fortsetzung).

Da jede logische Regel ein logisches Symbol entfernt, muss dieser Prozess terminieren.

Fall 2 kann tatsächlich nicht auftreten.

Nehme wir an, er sei bei der Sequenz $S_i := x_1, \dots, x_m \vdash y_1, \dots, y_n$ aufgetreten.

Wir definieren die Bewertung α mit $\alpha(x_i) = 1$ für jedes $i \in \{1, \dots, m\}$
und $\alpha(y_j) = 0$ für jedes $j \in \{1, \dots, n\}$.

Dann $\alpha \not\models S_i$.

Durch das letzte Lemma sind alle Konklusionen einer Regel mit S_i als Prämisse unerfüllt von α und so weiter bis $\alpha \not\models S_1 = \vdash F$. Aber F ist eine Tautologie. Widerspruch.

Da Fall 2 nicht auftreten kann, ist die Folge $S_n, \dots, S_1$ ein Beweis für $\vdash F$. □

Ende der 10. Vorlesung

Resolution

Beweissystem zum Beweis der Unerfüllbarkeit von Formeln in KNF.

Klauseln aufgefasst als **Mengen** von Literalen.

Einzigste Schlussregel: **Resolution**

$$\frac{C \vee a \quad D \vee \bar{a}}{C \vee D}$$

$C \vee D$ heißt **Resolvente** von $C \vee a$ und $D \vee \bar{a}$.

Resolution

Lemma

Für jede Bewertung α gilt:

falls $\alpha \models C \vee a$ und $\alpha \models D \vee \bar{a}$, dann auch $\alpha \models C \vee D$.

Beweis.

► **Fall 1:** $\llbracket a \rrbracket_\alpha = 0$.

Dann gilt $\llbracket C \rrbracket_\alpha = 1$ aus $\alpha \models C \vee a$.

Somit $\llbracket C \vee D \rrbracket_\alpha = 1$.

► **Fall 2:** $\llbracket a \rrbracket_\alpha = 1$.

Dann gilt $\llbracket D \rrbracket_\alpha = 1$ aus $\alpha \models D \vee \bar{a}$.

Somit $\llbracket C \vee D \rrbracket_\alpha = 1$.



Resolutionsbeweise

Eine Resolutions-Herleitung einer Klausel C aus einer Formel F in KNF ist eine Folge $C_1, \dots, C_k$ von Klauseln mit:

- ▶ $C_k = C$
- ▶ für jedes $i \leq k$ gilt eines der Folgenden:
 - ▶ $C_i \in F$
 - ▶ es gibt $j, k < i$, sodass C_i Resolvente von C_j und C_k ist

Eine **Widerlegung** von F ist eine Herleitung der leeren Klausel $\perp$ aus F .

Resolution: Korrektheit

Theorem

Sei eine Resolutions-Herleitung von C aus F gegeben.

Dann gilt für jede Bewertung α :

ist $\alpha \models F$, dann auch $\alpha \models C$.

Beweis (durch starke Induktion über die Länge der Herleitung).

- ▶ **Induktionsschritt:** Sei $C_1, \dots, C_k$ die Herleitung mit $C = C_k$.
 - ▶ **Fall 1:** $C \in F$. Dann $\alpha \models C$ folgt trivial.
 - ▶ **Fall 2:** C ist die Resolvente von C_i, C_j für $i, j < k$.
Durch die Induktionshypothese gilt $\alpha \models C_i$ und $\alpha \models C_j$.
Daher gilt $\alpha \models C$ durch das letzte Lemma.



Resolution: Korrektheit

Korollar

Gibt es eine Widerlegung von F , dann ist F unerfüllbar.

Resolution: Vollständigkeit

Theorem

Ist F unerfüllbar, dann gibt es eine Resolutions-Widerlegung von F .

Beweis (durch vollständige Induktion über die Anzahl der Variablen von F).

- ▶ **Induktionsanfang:** Die einzige Klausel mit null Variablen ist $\perp$.
Falls $\perp \in F$, dann ist $\perp$ eine Widerlegung von F .
Sonst ist $F = \emptyset$ und erfüllbar.
- ▶ **Induktionsschritt:** Falls F $n + 1$ Variablen enthält, wählen wir ein beliebiges x aus F .
Wir entfernen alle Klauseln, die $\bar{x}$ enthalten und entfernen x aus allen Klauseln, und nennen das Ergebnis F' .
Genauso wie F ist F' unerfüllbar, denn wenn $\alpha \models F'$, dann $\alpha' \models F$, wobei $\alpha'(x) = 0$ und $\alpha'(y) = \alpha(y)$ für $y \neq x$. Widerspruch.
Zudem hat F' nur n Variablen. Durch die Induktionshypothese gibt es eine Widerlegung $C'_1, \dots, C'_k = \perp$ von F' .

Resolution: Vollständigkeit

Theorem

Ist F unerfüllbar, dann gibt es eine Resolutions-Widerlegung von F .

Beweis (Fortsetzung).

Aus der Widerlegung $C'_1, \dots, C'_k$ werden wir eine Herleitung $C_1, \dots, C_k$ aus F konstruieren.

Für jedes $i \in \{1, \dots, k\}$ steigend:

- **Fall 1:** $C'_i \in F$. Dann entweder $C'_i \in F$ oder $C'_i \vee x \in F$.

Wir setzen C_i dementsprechend zu C'_i bzw. $C'_i \vee x$.

- **Fall 2:** C'_i ist eine Resolvente von C'_{j_1} und C'_{j_2} .

Wir setzen C_i zu C'_i , falls C_{j_1} und C_{j_2} keine x enthalten, sonst zu $C'_i \vee x$.

Entweder ist $C_1, \dots, C_k$ eine Widerlegung von F (falls $C_k = \perp$) oder eine Herleitung von x aus F (falls $C_k = x$).

Resolution: Vollständigkeit

Theorem

Ist F unerfüllbar, dann gibt es eine Resolutions-Widerlegung von F .

Beweis (Fortsetzung).

Zur Erinnerung: Entweder ist $C_1, \dots, C_k$ eine Widerlegung von F oder eine Herleitung von x aus F .

Das Ganze kann man analog mit $\bar{x}$ statt x durchführen.

Entweder ist die so konstruierte Herleitung eine Widerlegung von F oder eine Herleitung von $\bar{x}$ aus F .

Falls eine der beiden Folgen eine Widerlegung ist, haben wir unser Ziel erreicht.

Sonst verketteten wir die beiden Folgen und hängen einen Resolutionsschluss dran, der aus x und $\bar{x}$ die leere Klausel $\perp$ herleitet.



Tseitin-Expansion

Für eine Formel F konstruiere eine Formel $E(F)$ mit

- ▶ $E(F)$ ist in KNF
- ▶ $|E(F)| \leq c \cdot |F|$ für eine Konstante $c \in \mathbb{N}$
- ▶ $E(F)$ ist erfüllbar gdw. F erfüllbar ist.

Ohne Beschränkung der Allgemeinheit ist F in NNF.

Für jede Teilformel G von F führe neue Variable v_G und definierende Formeln D_G ein:

- ▶ $G = a$ Literal:

$$\begin{aligned} D_a &= (v_a \leftrightarrow a) = (v_a \rightarrow a) \wedge (a \rightarrow v_a) \\ &= (\bar{v}_a \vee a) \wedge (\bar{a} \vee v_a) \end{aligned}$$

Konstruktion der Formel $E(F)$

- ▶ $G = H_1 \vee H_2$ Disjunktion:

$$\begin{aligned} D_G &= (v_G \leftrightarrow (v_{H_1} \vee v_{H_2})) \\ &= (\overline{v}_G \vee v_{H_1} \vee v_{H_2}) \wedge (\overline{v}_{H_1} \vee v_G) \wedge (\overline{v}_{H_2} \vee v_G) \end{aligned}$$

- ▶ $G = H_1 \wedge H_2$ Konjunktion:

$$\begin{aligned} D_G &= (v_G \leftrightarrow (v_{H_1} \wedge v_{H_2})) \\ &= (\overline{v}_G \vee v_{H_1}) \wedge (\overline{v}_G \vee v_{H_2}) \wedge (\overline{v}_{H_1} \vee \overline{v}_{H_2} \vee v_G) \end{aligned}$$

$$E(F) := v_F \wedge \bigwedge_{G \text{ Teilformel von } F} D_G$$

Tseitin-Expansion

Theorem

Sei F eine Formel. Es gilt:

1. $E(F)$ ist in KNF.
2. $|E(F)| \leq c \cdot |F|$ für eine Konstante $c \in \mathbb{N}$.
3. $E(F)$ ist erfüllbar gdw. F erfüllbar ist.

Beweis.

Punkt 1 und 2 sind offensichtlich. Wir fokussieren uns auf Punkt 3.

Richtung $\leftarrow$: Sei also $\alpha \models F$.

Dann setzen wir α zu einer Bewertung α' der Variablen von $E(F)$ fort mittels $\alpha'(v_G) := \alpha(G)$.

Dann ist leicht zu sehen, dass α' alle Formeln D_G erfüllt.

Da $\alpha(F) = 1$ ist, ist auch $\alpha'(v_F) = 1$.

Somit $\alpha' \models E(F)$.

Tseitin-Expansion

Theorem

Sei F eine Formel. Es gilt:

1. $E(F)$ ist in KNF.
2. $|E(F)| \leq c \cdot |F|$ für eine Konstante $c \in \mathbb{N}$.
3. $E(F)$ ist erfüllbar gdw. F erfüllbar ist.

Beweis (Fortsetzung).

Richtung $\rightarrow$: Ist $\beta \models E(F)$, dann definieren wir β' als die Einschränkung von β auf die Variablen von F .

Durch starke Induktion kann man zeigen, dass wegen der Erfülltheit von D_G für jede Teilformel G von F gilt $\beta(v_G) = \beta'(G)$.

Also gilt auch $\beta(v_F) = \beta'(F) = 1$.

Das heißt, $\beta' \models F$.



SAT-Solving

Ein **SAT-Löser** ist ein Programm mit:

Input: Formel F in KNF

Output:

- ▶ α mit $\alpha \models F$, falls F erfüllbar ist
- ▶ UNSAT, sonst

Entwicklung von effizienten SAT-Lösern seit ca. 1996

Aktuelle SAT-Löser können mit Formeln aus Anwendungen mit > 100.000 Variablen und $> 1.000.000$ Klauseln umgehen.

Industrie-Standard bei Hardware-Verifikation

Der grundlegende DPLL-Algorithmus

Nach Davis, Putnam, Logemann und Loveland, 1960

Formel F , partielle Bewertung $\alpha \rightsquigarrow$ Formel $F\alpha$

- ▶ lösche erfüllte Klauseln
- ▶ lösche falsifizierte Literale aus den übrigen Klauseln

$\text{DPLL}(F, \alpha)$

if $F\alpha = 0$ then return UNSAT

if $F\alpha = 1$ then return α

pick variable x in $F\alpha$

$\beta := \text{DPLL}(F, \alpha \cup [x := 0])$

if $\beta \neq \text{UNSAT}$

then return β

else return $\text{DPLL}(F, \alpha \cup [x := 1])$

Verbesserungen an DPLL

Kommt in $F\alpha$ Klausel (a) der Größe 1 (unit clause) vor

setze $a := 1$

Iterieren bis nicht mehr anwendbar:

$\text{UnitProp}(F, \alpha)$

while $F\alpha$ enthält Klausel (a)

$\alpha := \alpha \cup [a := 1]$

Literal a ist **rein** in $F\alpha$, wenn $\bar{a}$ in $F\alpha$ nicht vorkommt.

Reine Literale können auf $a := 1$ gesetzt werden:

$\text{PureLit}(F, \alpha)$

while $F\alpha$ contains pure literal a

$\alpha := \alpha \cup [a := 1]$