Diskrete Mathematik

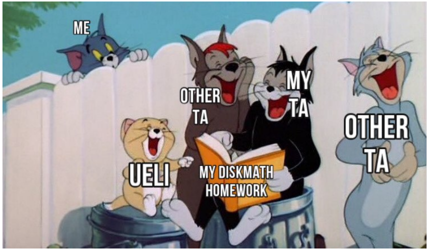
Woche 11: Logik - Proof Systems, Syntax & Semantik

Shivram Sambhus (cs.shivi.io)

ETH Zürich

Willkommen zu Kapitel 6!

- **Thema:** Kapitel 6 - Logik.
- **Der Shift:** Bisher (Algebra): Wir haben gerechnet ($x^2 + 1 = 0$). Jetzt (Logik): Wir begründen und leiten ab (Wahr $\implies$ Wahr).
- **Ziel:** Wir formalisieren den Begriff "Beweis", damit Computer mathematische Aussagen verifizieren können.
- **Warum ist das wichtig?**
  - Hardware-Verifikation (Intel Bugs vermeiden).
  - Software-Sicherheit (Beweisen, dass Code nicht crasht).
  - KI & Automated Reasoning.



Roadmap

Wir arbeiten uns durch die Sektionen des Skripts heute durch:

1. **Teil 1: Proof Systems (Die Meta-Ebene)** Was ist überhaupt ein Beweis? Die Trennung von *Wahrheit* (Gott) und *Beweisbarkeit* (Mensch/Maschine). (Kap 6.2)
2. **Teil 2: Allgemeine Konzepte** Das Vokabular: Syntax, Semantik, Freie Variablen und die vier Typen von Aussagen. (Kap 6.3 & 6.4)
3. **Teil 3: Aussagenlogik** Die Logik der Bits. Wir vertiefen **Normalformen (CNF/DNF)** und Äquivalenzumformungen. (Kap 6.5)
4. **Teil 4: Prädikatenlogik** Die Logik der Mathematik. Wir führen Strukturen und Quantoren ein ($\forall, \exists$). (Kap 6.6)

Teil 1: Proof Systems (Die Meta-Ebene)

Motivation: Was ist ein Beweis?

In der Schule: "Ein Text, der den Lehrer überzeugt." In der Informatik: "Ein String, der mechanisch verifiziert werden kann."

Wir unterscheiden zwei Welten:

1. **Die Semantik (τ):** Ist eine Aussage *tatsächlich* wahr? (Gott-Perspektive). Das ist oft schwer zu wissen.
2. **Die Syntax (ϕ):** Können wir *überprüfen*, dass sie wahr ist? (Computer-Perspektive). Das ist ein mechanischer Prozess.

Ziel: Wir wollen Systeme bauen, wo (2) uns garantiert, dass (1) gilt.

Analogie: Das Sudoku

Stell dir vor, ich behaupte: *"Dieses extrem schwere Sudoku ist lösbar."* (S)

- **Wahrheit ($\tau(S)$):** Entweder es gibt eine Lösung oder nicht. Das ist ein faktischer Zustand des Universums, unabhängig davon, ob wir ihn kennen.
- **Beweis (P):** Ich gebe dir das ausgefüllte Gitter (den "Zeugen").
- **Verifikation ($\phi(S, P)$):** Du prüfst jede Zeile, Spalte und Box auf Konflikte. Stimmt alles, akzeptierst du meine Behauptung.

Insight: Einen Beweis zu FINDEN ist schwer (benötigt Intelligenz/Suche). Ihn zu PRÜFEN ist leicht (mechanisch/effizient).

Definition 6.1: Proof System

Ein **Proof System** ist ein Tupel $\Pi = (\mathcal{S}, \mathcal{P}, \tau, \phi)$.

- $\mathcal{S}$: Menge der Aussagen (Statements, z.B. Sudoku-Gitter). *Wir nehmen an* $\mathcal{S} \subseteq \{0,1\}^*$*.*
- $\mathcal{P}$: Menge der Beweise (Proofs, z.B. Lösungen). *Wir nehmen an* $\mathcal{P} \subseteq \{0,1\}^*$*.*
- $\tau : \mathcal{S} \to \{0,1\}$: Die **Wahrheitsfunktion**. (Semantik). Gibt an, ob S *wirklich* wahr ist.
- $\phi : \mathcal{S} \times \mathcal{P} \to \{0,1\}$: Die **Verifikationsfunktion**. $\phi(S,P) = 1$ bedeutet "P ist ein gültiger Beweis für S".

Wichtig: ϕ muss **effizient** (z.B. Polynomialzeit) berechenbar sein.

Die Qualitätsmerkmale eines Beweissystems

Wir wollen zwei Eigenschaften für unser System:

1. **Nichts Falsches beweisen!** (Soundness / Korrektheit) Wenn ich dir ein falsch ausgefülltes Sudoku gebe, muss ϕ "Fehler!" sagen. Ein System, das Lügen als wahr akzeptiert, ist gefährlich.
2. **Alles Wahre beweisen können!** (Completeness / Vollständigkeit) Wenn das Sudoku lösbar ist, muss es möglich sein, eine Lösung vorzulegen, die ϕ akzeptiert. Es darf keine "unerreichbaren Wahrheiten" geben.

Soundness & Completeness (Formal)

Definition 6.2 (Soundness): Ein System ist sound, wenn gilt:

$$(\exists P \in \mathcal{P} : \phi(S, P) = 1) \implies \tau(S) = 1$$

"Wenn es einen akzeptierten Beweis gibt, ist die Aussage wahr."

Definition 6.3 (Completeness): Ein System ist complete, wenn gilt:

$$\tau(S) = 1 \implies (\exists P \in \mathcal{P} : \phi(S, P) = 1)$$

"Wenn die Aussage wahr ist, gibt es einen akzeptierten Beweis."

Beispiel aus dem Skript: Hamilton-Kreis

Aussage S: Ein Graph G (repräsentiert als Adjazenzmatrix-Bits) hat einen Hamilton-Kreis (besucht jeden Knoten genau einmal).

Beweis P: Eine Sequenz von Knotenindizes $(v_1, v_2, \dots, v_n)$.

Verifikation $\phi(S, P)$:

1. Prüfe, ob (v_i, v_{i+1}) Kanten in G sind.
2. Prüfe, ob jeder Knoten genau einmal vorkommt.
3. Prüfe, ob (v_n, v_1) eine Kante ist.

Analyse: Dieses System ist **sound** und **complete**. Aber: Das *Finden* von P ist schwer (NP-Complete).

Asymmetrie der Beweise

Problem: Beweise, dass G **keinen** Hamilton-Kreis hat.

- Was wäre der Beweis P?
- "Ich habe alle Möglichkeiten probiert"? $\rightarrow$ Dieser Beweis wäre exponentiell lang!
- ϕ könnte diesen Beweis nicht effizient prüfen (da ϕ effizient sein muss).

Erkenntnis: Wahrheit und effiziente Beweisbarkeit sind oft asymmetrisch. Es ist leicht zu zeigen, dass 101 prim ist (Zertifikat), aber schwer zu zeigen, dass ein Graph *keinen* Kreis hat. Wir kennen kein effizientes Proof System für die Nicht-Existenz (Klasse co-NP).

Mini-Übung: Proof Systems

Frage 1: Ein System ist **sound**, aber nicht **complete**. Kann ich einem Beweis vertrauen, wenn das System ihn akzeptiert?

Frage 2: Ein System ist **complete**, aber nicht **sound**. Kann ich einem Beweis vertrauen?

Frage 3 (Harder): Sei Π ein Proof System. Wir bauen ein neues System Π', das genau gleich funktioniert, aber zusätzlich das leere Wort ϵ als "Joker-Beweis" für *jede* Aussage akzeptiert.

a) Ist Π' complete, wenn Π complete ist?
b) Ist Π' sound?

Lösungen Mini-Übung: Proof Systems

Zu Frage 1 (Soundness): Ja. Soundness bedeutet: Wenn der Beweis akzeptiert wird, ist die Aussage wahr. Dass manche wahren Aussagen *keinen* Beweis haben (Incompleteness), ändert nichts an der Verlässlichkeit der vorhandenen Beweise.

Zu Frage 2 (Completeness): Nein. Completeness heisst nur, dass alle wahren Aussagen beweisbar sind. Es verbietet nicht, dass auch falsche Aussagen beweisbar sind.

Zu Frage 3 (Joker-Beweis):

a) **Ja.** Da Π' alle Beweise von Π akzeptiert, sind alle wahren Aussagen weiterhin beweisbar.
b) **Nein.** Da Π' das leere Wort für *jede* Aussage akzeptiert, akzeptiert es auch Beweise für falsche Aussagen (Lügen). Damit ist die Soundness zerstört.

Teil 2: Allgemeine Logik-Konzepte

Vom Proof System zur Logik

Wir verlassen die Meta-Ebene und gehen zur Logik (Kap 6.3). Hier ist ein Statement S oft ein Paar (M, G) bestehend aus einer Menge von Formeln M (Annahmen) und einer Formel G (Schlussfolgerung).

Wir unterscheiden Semantik und Syntax:

1. **Syntax:** Welche Zeichenketten sind "wohlgeformte Formeln"? (Rein mechanisch: Klammern, Symbole).
2. **Semantik:** Was bedeuten die Symbole? Wann ist eine Formel wahr? (Braucht Interpretation).

Das Konzept der “Freien Variablen” (Kap 6.3.3)

Bevor wir über Wahrheit reden können, müssen wir klären, welche Teile einer Formel “Input” von aussen brauchen.

Definition 6.5 (Freie Symbole/Variablen): Die Semantik definiert eine Funktion `free(F)`, die angibt, welche Symbole in einer Formel F **nicht gebunden** sind.

- In der **Aussagenlogik**: Alle atomaren Variablen $(A, B, P \dots)$ sind immer frei. sie haben keinen Wert, bis wir ihnen einen geben.
- In der **Prädikatenlogik**: Variablen können durch Quantoren $(\forall x, \exists x)$ gebunden werden. Nur die *nicht* gebundenen Variablen sind “frei” und müssen von der Interpretation festgelegt werden.

Beispiel (Prädikatenlogik): In $P(x) \wedge \forall y Q(y)$ ist x frei (Input nötig), aber y ist gebunden (lokal).

Semantik: Interpretation und Modell

Um “Wahrheit” zu definieren, müssen wir alle freien Symbole interpretieren.

Definition 6.6 & 6.8 (Interpretation): Eine Interpretation $\mathcal{A}$ weist den freien Symbolen einer Formel Werte zu. $\mathcal{A}(F) \in \{0, 1\}$ ist der Wahrheitswert von Formel F unter dieser Interpretation.

Definition 6.9 (Modell): Eine Interpretation $\mathcal{A}$ heisst **Modell** für F, wenn $\mathcal{A}(F) = 1$. Man schreibt: $\mathcal{A} \models F$.

Für eine Menge M: $\mathcal{A} \models M$ gdw. $\mathcal{A} \models F$ für alle $F \in M$.

Die 4 Typen von Aussagen in der Logik (Kap 6.3.8)

Es ist wichtig zu unterscheiden, auf welcher Ebene wir argumentieren.

1. **Theoreme innerhalb einer Theorie:** Sei T eine Menge von Axiomen (z.B. Gruppenaxiome). F ist ein Theorem, wenn $T \models F$.
2. **Eigenschaften einer Formel (Meta-Ebene):** z.B. "F ist erfüllbar" oder "F ist eine Tautologie". Dies ist eine Aussage *über* F.
3. **Wahrheit in einer spezifischen Interpretation:** $\mathcal{A} \models F$. (Beispiel: "In der Gruppe $\mathbb{Z}$ ist $x + 0 = x$ wahr").
4. **Aussagen über das Logiksystem selbst:** z.B. "Der Kalkül ist korrekt (sound)" oder "Die Aussagenlogik ist entscheidbar".

Logische Folgerung & Äquivalenz

Definition 6.12 (Logische Folgerung / Consequence): Eine Formel G folgt logisch aus M, geschrieben $M \models G$, wenn **jedes** Modell von M auch ein Modell von G ist.

Definition 6.13 (Äquivalenz): Zwei Formeln F und G sind äquivalent $(F \equiv G)$, wenn sie in **jeder** Interpretation denselben Wahrheitswert haben.

$$F \equiv G \iff (F \models G \text{ und } G \models F)$$

Logische Kalküle (Kap 6.4)

Wie formalisieren wir "Beweisen"? Wir wollen reine Syntax-Manipulation.

Definition 6.17 (Ableitungsregel): Eine Regel erlaubt es, aus einer Menge von Formeln (Prämissen) eine neue Formel (Konklusion) abzuleiten.

$$\{F_1, \dots, F_k\} \vdash_R G$$

Definition 6.19 (Kalkül K): Eine endliche Menge solcher Regeln. Ein Computer kann diese Regeln stur anwenden.

Definition 6.20 (Herleitung): Eine Sequenz von Formeln, wobei jede Formel entweder eine Annahme aus M ist oder durch eine Regel aus vorherigen Formeln entsteht. Symbol: $M \vdash_K G$.

Soundness und Completeness von Kalkülen

Das Ziel der Logik ist es, $\vdash$ (Syntax) so zu definieren, dass es $\models$ (Semantik) entspricht.

Ein Kalkül K ist:

1. **Sound (Korrekt):**
$$M \vdash_K F \implies M \models F$$
(Der Kalkül produziert keinen Unsinn. Wenn wir es ableiten können, stimmt es auch).
2. **Complete (Vollständig):**
$$M \models F \implies M \vdash_K F$$
(Der Kalkül ist mächtig genug, alle logischen Folgerungen zu finden).

Mini-Übung: Syntax vs. Semantik

Frage 1: Welche der folgenden Aussagen sind syntaktisch (über Formeln) und welche semantisch (über Wahrheit)?

1. F ist eine Tautologie.
2. F hat weniger als 5 Zeichen.
3. $M \vdash F$.
4. $M \models F$.

Frage 2: In der Formel $\forall x(P(x) \to Q(y))$, welche Variablen sind frei?

Frage 3 (Harder): Sei F eine unerfüllbare Formel (z.B. $A \land \neg A$). Gilt $F \models G$ für eine beliebige Formel G? Begründe.

Lösungen Mini-Übung: Syntax vs. Semantik

Zu Frage 1:

1. Semantisch (bezieht sich auf Modelle).
2. Syntaktisch (bezieht sich auf Zeichenkette).
3. Syntaktisch (Ableitbarkeit im Kalkül).
4. Semantisch (Logische Folgerung).

Zu Frage 2:

y ist **frei**. x ist **gebunden** (durch $\forall x$).

Zu Frage 3 (Ex Falso Quodlibet):

Ja.

Definition von $\models$: "In jedem Modell, in dem F wahr ist, muss auch G wahr sein." Da es *kein* Modell gibt, in dem F wahr ist, ist die Voraussetzung nie erfüllt (leere Menge an Modellen). Die Aussage ist damit trivialerweise wahr ("vacuously true"). Aus einem Widerspruch folgt alles.

Teil 3: Aussagenlogik

Syntax der Aussagenlogik

Wir werden konkret.

Definition 6.23 (Syntax):

1. **Atome:** A_i (oder A, B, P, Q, R) sind atomare Formeln. *In der AL sind alle Atome "freie Variablen".*
2. **Verknüpfungen:** Sind F, G Formeln, dann auch:
   - $\neg F$ (Negation / NOT)
   - $(F \wedge G)$ (Konjunktion / AND)
   - $(F \vee G)$ (Disjunktion / OR)

Syntactic Sugar:

- $F \rightarrow G$ steht für $\neg F \vee G$.
- $F \leftrightarrow G$ steht für $(F \wedge G) \vee (\neg F \wedge \neg G)$.

Semantik: Belegungen

Eine Formel wie $A \wedge B$ ist nur Syntax. sie braucht eine **Belegung**, um wahr oder falsch zu sein.

Definition 6.24 (Belegung / Truth Assignment): Eine Belegung $\mathcal{A}$ ist eine Funktion $\mathcal{A} : \{A, B, ...\} \to \{0, 1\}$. *(Dies ist die "Interpretation" in der Aussagenlogik).*

Die Semantik einer Formel F unter $\mathcal{A}$ ist rekursiv definiert:

- $\mathcal{A}(\neg F) = 1 \iff \mathcal{A}(F) = 0$.
- $\mathcal{A}(F \wedge G) = 1 \iff \mathcal{A}(F) = 1$ UND $\mathcal{A}(G) = 1$.
- $\mathcal{A}(F \vee G) = 1 \iff \mathcal{A}(F) = 1$ ODER $\mathcal{A}(G) = 1$.

Äquivalenzumformungen (Lemma 6.1)

Bevor wir zu Normalformen kommen: Wir können Formeln umformen wie in der Algebra.

1. **De Morgan:** $\neg(A \wedge B) \equiv \neg A \vee \neg B$.
2. **Distributivität:**
   - $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$.
   - $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$.
3. **Doppelnegation:** $\neg\neg A \equiv A$.
4. **Implikation:** $A \to B \equiv \neg A \vee B$.

Diese Regeln sind essentiell, um Formeln "aufzuräumen" und in Standardformen zu bringen.

Normalformen: Definitionen

Für Computerbeweise brauchen wir Standardformate.

Definition 6.25 (Literal): Ein Atom A oder dessen Negation $\neg A$.

Definition 6.26 (CNF - Konjunktive Normalform): Eine Konjunktion von Disjunktionen von Literalen (“Und von Oders”).

$$F = (L_{1,1} \vee L_{1,2} \ldots) \wedge (L_{2,1} \vee \ldots) \wedge \ldots$$

Vorteil: Gut für SAT-Solver (Resolution). Um zu zeigen, dass F wahr ist, muss jede Klammer wahr sein.

Definition 6.27 (DNF - Disjunktive Normalform): Eine Disjunktion von Konjunktionen (“Oder von Unds”).

$$F = (L_{1,1} \wedge L_{1,2} \ldots) \vee (L_{2,1} \wedge \ldots) \vee \ldots$$

Vorteil: Gut für Erfüllbarkeit. Wenn eine Klammer wahr ist, ist alles wahr.

Theorem 6.4: Existenz von Normalformen

Theorem: Jede Formel ist äquivalent zu einer Formel in CNF und auch zu einer Formel in DNF.

Wir schauen uns zwei Methoden an, dies zu erreichen:

1. **Algebraische Umformung** (Syntax-basiert).
2. **Wahrheitstabelle** (Semantik-basiert).

Methode 1: Algebraische Umformung

Kochrezept für CNF:

1. **Implikationen eliminieren:** $A \to B \rightsquigarrow \neg A \lor B$.
2. **Negationen nach innen (De Morgan):** $\neg(A \land B) \rightsquigarrow \neg A \lor \neg B$. Bis $\neg$ nur noch vor Atomen steht.
3. **Distributivgesetz:** Wir wollen $\land$ aussen. Wende $A \lor (B \land C) \rightsquigarrow (A \lor B) \land (A \lor C)$ an.

Beispiel: $F = \neg(A \land \neg B) \lor C$.

1. $\neg A \lor \neg\neg B \lor C$ (De Morgan)
2. $\neg A \lor B \lor C$ (Doppelnegation) Dies ist bereits CNF (eine einzige Klausel).

Methode 2: Konstruktion aus Wahrheitstabelle

Diese Methode funktioniert immer, kann aber zu grossen Formeln führen.

Für DNF (Oder von Unds):

1. Erstelle Wahrheitstabelle.
2. Markiere alle Zeilen, wo Ergebnis $= 1$.
3. Für jede solche Zeile: Bilde ein UND der Literale (A wenn 1, $\neg$ A wenn 0).
4. Verknüpfe diese Terme mit ODER.

Für CNF (Und von Oders):

1. Markiere alle Zeilen, wo Ergebnis $= \mathbf{0}$.
2. Für jede solche Zeile: Bilde ein ODER der **negierter** Literale (A wenn 0, $\neg$ A wenn 1).
3. Verknüpfe diese Klauseln mit UND.

Beispiel: Wahrheitstabelle zu DNF/CNF

Betrachte XOR: $A \oplus B$. (Wahr wenn ungleich).

A	B	Resultat
0	0	0
0	1	1
1	0	1
1	1	0

DNF (Zeilen mit 1):

- Zeile 2 $(A = 0, B = 1)$: $\neg A \wedge B$.
- Zeile 3 $(A = 1, B = 0)$: $A \wedge \neg B$. $\implies F_{DNF} = (\neg A \wedge B) \vee (A \wedge \neg B)$.

CNF (Zeilen mit 0):

- Zeile 1 $(A = 0, B = 0)$: Negiere Literale $\to A \vee B$.
- Zeile 4 $(A = 1, B = 1)$: Negiere Literale $\to \neg A \vee \neg B$.
  $\implies F_{CNF} = (A \vee B) \wedge (\neg A \vee \neg B)$.

Übungsblock 1: Aussagenlogik

Aufgaben:

1. **Algebraische Umformung:** Bringe $F = \neg(A \to (B \land C))$ durch Umformung in CNF.
2. **Wahrheitstabelle:** Konstruiere die DNF für die Implikation $A \to B$ mithilfe der Wahrheitstabelle.
3. **Verständnis (Semantik):** Sei $M = \{A \lor B, \neg A\}$. Gilt $M \models B$? Begründe mit Modellen.

Lösungen 1 (Teil 1)

1. Algebraisch zu CNF:

Wir formen $F = \neg(A \rightarrow (B \wedge C))$ um:

$$\begin{aligned} F &= \neg(\neg A \vee (B \wedge C)) && \text{(Implikation)} \\ &= \neg\neg A \wedge \neg(B \wedge C) && \text{(De Morgan)} \\ &= A \wedge (\neg B \vee \neg C) && \text{(De Morgan)} \end{aligned}$$

Das ist bereits CNF mit zwei Klauseln: $\{A\}$ und $\{\neg B, \neg C\}$.

Lösungen 1 (Teil 2)

2. Tabelle zu DNF:

$A \to B$ ist wahr für die Belegungen $(0,0),(0,1),(1,1)$.

- $(0,0) \implies \neg A \wedge \neg B$
- $(0,1) \implies \neg A \wedge B$
- $(1,1) \implies A \wedge B$

$$F_{DNF} = (\neg A \wedge \neg B) \vee (\neg A \wedge B) \vee (A \wedge B)$$

3. Logische Folgerung:

Sei $M = \{A \vee B, \neg A\}$.

- $\neg A$ muss wahr sein $\implies A = 0$.
- Damit $A \vee B$ wahr ist (bei $A = 0$), muss $B = 1$ sein.
- Das einzige Modell ist $\mathcal{A}(A) = 0, \mathcal{A}(B) = 1$.
- In diesem Modell ist B wahr. Also ja, $M \models B$.

Exam Challenge: Aussagenlogik (HS18)

Aufgabe 1 (CNF): Finde eine äquivalente Formel in CNF für:

$$F = ((A \to B) \to C) \lor (C \leftrightarrow \neg A)$$

Aufgabe 2 (NOR-Operator): Der Operator $\downarrow$ (NOR) ist definiert als $A \downarrow B \equiv \neg(A \lor B)$ (wahr gdw. beide falsch). Zeige, dass man jede Formel nur mit $\downarrow$ ausdrücken kann, indem du Konstruktionen für $\neg F$, $F \lor G$ und $F \land G$ findest.

Lösung Challenge 1 (Teil 1: CNF)

Aufgabe: CNF für $F = ((A \to B) \to C) \vee (C \leftrightarrow \neg A)$.

Schritt 1: Vereinfachung der Teile

- $C \leftrightarrow \neg A \equiv (C \wedge \neg A) \vee (\neg C \wedge A)$
- $(A \to B) \to C \equiv \neg(\neg A \vee B) \vee C \equiv (A \wedge \neg B) \vee C$

Schritt 2: Zusammensetzen

$$F \equiv (A \wedge \neg B) \vee C \vee (C \wedge \neg A) \vee (\neg C \wedge A)$$

Da $C \vee (C \wedge \dots) \equiv C$ (Absorption), fällt der dritte Term weg:

$$F \equiv (A \wedge \neg B) \vee C \vee (\neg C \wedge A)$$

Lösung Challenge 1 (Teil 2: CNF Finale)

Schritt 3: Distributivität Wir nutzen $C \vee (\neg C \wedge A) \equiv (C \vee \neg C) \wedge (C \vee A) \equiv C \vee A$.

$$
\begin{aligned}
F &\equiv (A \wedge \neg B) \vee (C \vee A) \\
&\equiv ((A \wedge \neg B) \vee A) \vee C \\
&\equiv A \vee C \quad \text{(Absorption: } (A \wedge X) \vee A \equiv A\text{)}
\end{aligned}
$$

Ergebnis: $A \vee C$.

Lösung Challenge 1 (Teil 3: NOR)

Aufgabe: Drücke $\neg, \vee, \wedge$ durch $\downarrow$ (NOR) aus. Erinnerung: $A \downarrow B \equiv \neg(A \vee B)$.

1. **Negation:**
$$\neg F \equiv \neg(F \vee F) \equiv F \downarrow F$$

2. **Disjunktion (OR):**
$$F \vee G \equiv \neg(\neg(F \vee G)) \equiv \neg(F \downarrow G) \equiv (F \downarrow G) \downarrow (F \downarrow G)$$

3. **Konjunktion (AND):**
$$F \wedge G \equiv \neg(\neg F \vee \neg G) \equiv \neg F \downarrow \neg G$$

   Einsetzen von (1):
$$\equiv (F \downarrow F) \downarrow (G \downarrow G)$$

Teil 4: Prädikatenlogik (First-Order Logic)

Motivation: Warum reicht Aussagenlogik nicht?

Aussagenlogik ist zu schwach für Mathematik und komplexe Software-Spezifikationen. Satz: *"Für alle Zahlen x gibt es ein $y > x$."*

In AL müssten wir unendlich viele Klauseln schreiben: $(x_1 < x_2) \vee (x_1 < x_3) \vee$ ...

Die Erweiterung:

1. **Terme:** Um über Objekte zu sprechen ($x, f(x), c$).
2. **Prädikate:** Um Eigenschaften zu beschreiben ($P(x)$).
3. **Quantoren:** Um über Mengen zu sprechen ($\forall, \exists$).

Syntax I: Terme

Terme repräsentieren **Objekte** im Universum.

Definition 6.31:

1. Jede **Variable** x_i ist ein Term.
2. Jedes **Funktionssymbol** f mit Stelligkeit k bildet mit k Termen einen neuen Term $f(t_1, \ldots, t_k)$. *(Konstanten sind einfach Funktionen mit Stelligkeit 0, z.B. $c()$)*.

Beispiel: $f(g(x, a), y)$ ist ein Term. *Wichtig:* Ein Term hat keinen Wahrheitswert! Er bezeichnet ein Ding (z.B. eine Zahl).

Syntax II: Formeln

Formeln repräsentieren **Wahrheitswerte** (Aussagen).

Definition 6.31 (Fortsetzung):

1. **Atomare Formel:** $P(t_1, \ldots, t_k)$, wobei P ein Prädikatssymbol ist. *Beispiel:* $LessThan(3, 5)$.
2. **Logische Verknüpfung:** $\neg F, (F \wedge G), (F \vee G)$.
3. **Quantoren:**
   - $\forall x F$ (Für alle x gilt F)
   - $\exists x F$ (Es gibt ein x, für das F gilt).

Beispiel: $\forall x(P(x) \rightarrow \exists y Q(x, y))$.

Freie und Gebundene Variablen

Der Gültigkeitsbereich (Scope) von Variablen ist wie in der Programmierung.

Definition 6.32: Eine Variable x in F heisst **gebunden**, wenn sie im Bereich eines Quantors $\forall x$ oder $\exists x$ steht. Sonst heisst sie **frei**.

Beispiel:

$$F = P(x) \wedge \forall x Q(x)$$

- Das erste x (in P) ist **frei** (Globale Variable, Wert kommt von aussen).
- Das zweite x (in Q) ist **gebunden** (Lokale Variable, nur innerhalb des Quantors gültig).

*Eine Formel ohne freie Variablen heisst **geschlossen** (Satz).* Nur geschlossene Formeln haben einen festen Wahrheitswert (unter einer Struktur).

Semantik: Die Struktur

Was bedeutet $P(f(x))$? Ohne Kontext nichts. Wir brauchen eine **Struktur** (Interpretation), um den Symbolen Bedeutung zu geben.

Definition 6.34 (Struktur $\mathcal{A}$): Eine Struktur $\mathcal{A} = (U, \phi, \psi, \xi)$ besteht aus:

1. **Universum U:** Eine nicht-leere Menge von Objekten (z.B. $\mathbb{N}$).
2. **Interpretation der Symbole:**
   - $\phi(f) : U^k \to U$ (Funktion, z.B. $+$).
   - $\psi(P) : U^k \to \{0, 1\}$ (Relation, z.B. $\leq$).
3. **Variablenbelegung ξ:** Weist freien Variablen Werte in U zu.

Prozedur: Evaluation einer Formel

Wie prüfen wir, ob $\mathcal{A} \models F$?

1. Term-Auswertung: Berechne den Wert aller Terme (von innen nach aussen). $\mathcal{A}(f(t)) = \phi(f)(\mathcal{A}(t))$.

2. Formel-Auswertung: Prüfe die atomaren Formeln: $\mathcal{A}(P(t)) = 1 \iff \mathcal{A}(t) \in \psi(P)$. Verknüpfe mit AND/OR/NOT.

3. Quantoren (Das Herzstück): Sei $\mathcal{A}_{[x \to u]}$ die Struktur, wo x als $u \in U$ interpretiert wird.

- $\mathcal{A}(\forall x G) = 1 \iff$ Für **alle** $u \in U$ gilt $\mathcal{A}_{[x \to u]}(G) = 1$.
- $\mathcal{A}(\exists x G) = 1 \iff$ Es gibt **mindestens ein** $u \in U$ mit $\mathcal{A}_{[x \to u]}(G) = 1$.

Intuition: Spiel-Semantik

Stell dir Quantoren als Spiel gegen einen Gegner vor. Ich behaupte, die Formel ist wahr.

- $\forall x$: Der **Gegner** darf ein x wählen. Ich muss zeigen, dass die Formel trotzdem stimmt. (Worst-Case).
- $\exists x$: **Ich** darf ein x wählen (ein Zeuge/Witness). Ich muss nur eines finden, das funktioniert.

Beispiel: $\forall x \exists y (y > x)$ in $\mathbb{N}$. Gegner wählt $x = 1000$. Ich wähle $y = 1001$. $1001 > 1000$ ist wahr. Ich gewinne. Da ich für *jedes* x des Gegners eine Antwort habe, ist die Formel wahr.

Beispiel: Evaluation

Formel $F = \exists x \forall y P(x, y)$.

Struktur $\mathcal{A}$:

- $U = \mathbb{N}$ (Natürliche Zahlen).
- $P(x, y) \equiv$ "$x \leq y$".

Auswertung: $\mathcal{A}(F) = 1$, wenn es ein x gibt, das kleiner gleich alle y ist. Teste $x = 0$: Ist $\forall y(0 \leq y)$ wahr? Prüfe $y = 0 \to 0 \leq 0$ (Ja). Prüfe $y = 1 \to 0 \leq 1$ (Ja). ... Da 0 das kleinste Element ist, gilt es für alle y. $\implies \mathcal{A}(F) = 1$.

Hinweis: In $U = \mathbb{Z}$ wäre die Formel falsch (kein kleinstes Element).

Logische Äquivalenzen (Lemma 6.7)

Wie in der Aussagenlogik können wir Formeln umformen.

1. **De Morgan für Quantoren:** $\neg \forall x F \equiv \exists x \neg F$ (“Nicht alle sind weiss” $\equiv$ “Es gibt einen nicht-weissen”) $\neg \exists x F \equiv \forall x \neg F$ (“Es gibt keinen weissen” $\equiv$ “Alle sind nicht-weiss”)
2. **Vertauschung:** $\forall x \forall y F \equiv \forall y \forall x F$ $\exists x \exists y F \equiv \exists y \exists x F$ *Achtung:* $\forall x \exists y F \not\equiv \exists y \forall x F$! (Reihenfolge ist wichtig!)
3. **Quantoren-Verschiebung:** $(\forall x F) \wedge G \equiv \forall x (F \wedge G)$, falls x nicht frei in G vorkommt.

Ableitungsregel: Universelle Instanziierung

Wir haben auch in der Prädikatenlogik Kalküle. Eine zentrale Regel ist:

Lemma 6.11 (Universelle Instanziierung):

$$\forall x F \vDash F[x/t]$$

Wenn etwas für alle gilt, gilt es auch für das spezifische Objekt t.

Beispiel: $\forall x \text{Sterblich}(x) \vDash \text{Sterblich}(\text{Sokrates})$.

Dies erlaubt uns, den Quantor zu entfernen und mit Termen zu arbeiten.

Übungsblock 2: Prädikatenlogik

Aufgaben

1. **Semantik:** Finde ein Modell für $F = \exists x \forall y (P(x,y) \rightarrow \neg P(y,x))$. (Hinweis: Asymmetrie).
2. **Gegenbeispiel:** Zeige, dass $\exists x P(x) \land \exists x Q(x) \not\models \exists x (P(x) \land Q(x))$. (Warum folgt das Rechte nicht aus dem Linken?)
3. **Exam Challenge (HS18):** Ist die folgende Formel eine Tautologie? Beweise es.

$$F = \forall x (P(x, f(x)) \lor \exists y \neg P(x,y))$$

Lösungen 2

1. Modell: Wir brauchen eine Relation, die nicht symmetrisch ist. $U = \mathbb{N}, P(x, y) \equiv x < y$. $\exists x \forall y (x < y \to \neg(y < x))$. Wähle $x = 0$. $0 < y \to y \not< 0$ (also $y \geq 0$). Das ist wahr für alle $y \in \mathbb{N}$.

2. Gegenbeispiel: Intuition: Jemand mag Pizza, Jemand mag Sushi $\not\Rightarrow$ Jemand mag Pizza UND Sushi. $U = \mathbb{N}$. $P(x)$: "x ist gerade", $Q(x)$: "x ist ungerade". Links: Es gibt Gerade und Ungerade (Wahr). Rechts: Es gibt eine Zahl, die gerade UND ungerade ist (Falsch).

Lösung Challenge (Tautologie)

3. Tautologie: Zu prüfen: $F = \forall x(P(x, f(x)) \vee \exists y \neg P(x, y))$.

Wir formen um:

$$\exists y \neg P(x, y) \equiv \neg \forall y P(x, y)$$

Damit ist die Formel äquivalent zu:

$$\forall x(P(x, f(x)) \vee \neg \forall y P(x, y))$$

Dies ist logisch äquivalent zur Implikation:

$$\forall x(\forall y P(x, y) \rightarrow P(x, f(x)))$$

Beweis: Dies ist eine Instanz der **Universellen Instanziierung**. Wenn $P(x, y)$ für *alle* y gilt, dann muss es insbesondere auch für das spezifische Element $y = f(x)$ gelten. $\Longrightarrow$ Die Formel ist eine Tautologie.

Zusammenfassung

Was du mitnehmen solltest

1. **Proof Systems:** Formalisierung von "Wahrheit" (τ, Semantik) und "Beweis" (ϕ, Syntax). Soundness und Completeness sind die Brücke.
2. **Aussagenlogik:** Wahrheitstabellen, Normalformen (CNF/DNF).
3. **Prädikatenlogik:** Unterscheidung Terme (Objekte) vs. Formeln (Aussagen). Quantoren $(\forall, \exists)$ erlauben uns, über Unendlichkeiten zu sprechen.

Viel Erfolg bei der Serie!

Offene Fragen & Feedback

- Feedback zur heutigen Session? (https://forms.gle/LPrQfoZNsAHVeKoM9)
- E-Mail: dm@shivi.io

Schöne Pause und bis nächste Woche!

DM K (🇨🇭 Swiss German)

WhatsApp group