Logik – Prüfungsunterlagen

Fabian Hahn, Dino Wernli

Juli 2008

Mengenlehre

Eine Menge ist eine Zusammenfassung unterschiedlicher Objekte und ist selbst wieder ein Objekt. Für Mengen gelten ausserdem die ZFC-Axiome:

1. $A = B \leftrightarrow (x \in \mathrm{A} \leftrightarrow x \in \mathrm{B})$
2. $A \subseteq B \leftrightarrow (x \in \mathrm{A} \rightarrow x \in \mathrm{B})$
3. $(A \subseteq B) \wedge (B \subseteq A) \leftrightarrow \mathrm{A} = \mathrm{B}$
4. $x \in (A \cap B) \leftrightarrow (x \in A) \wedge (x \in B)$
5. $x \in (A \cup B) \leftrightarrow (x \in A) \vee (x \in B)$
6. $x \in (A \backslash B) \leftrightarrow (x \in A) \wedge (x \notin B)$
7. $x \in (A \triangle B) \leftrightarrow (x \in A) \oplus (x \in B)$
8. $A,\ B$ gleichmächtig $\leftrightarrow (\exists f: A \rightarrow B$ bijektiv$)$

Die Potenzmenge 2^A einer Menge A ist die Menge aller möglichen Teilmengen von A.

Russel's Antinomie: $M = \{A \mid A \notin A\}$
Es folgt ein Paradoxon: $M \in M \rightarrow M \notin M \rightarrow M \in M$ …

Gödel: „Für diesen Satz gibt es im System keinen Beweis". Dieser Satz ist immer wahr, denn wenn er falsch wäre, gäbe es für ihn einen Beweis, was ihn wiedrum wahr machen würde. Daraus folgt, dass es in jedem System wahre, aber unbeweisbare Sätze gibt.

Aussagenlogik

Grundjunktoren:

A	B	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \oplus B$	$A \mid B$
0	0	0	0	1	0	1
0	1	0	1	1	1	1
1	0	0	1	0	1	1
1	1	1	1	1	0	0

Aussagenlogik in NANDs:

A "NAND" $B \equiv A \mid B \equiv \neg(A \wedge B)$

Umformungen:

- $\neg A \equiv A|A$
- $A \wedge B \equiv (A|B)|(A|B)$
- $A \vee B \equiv (A|A)|(B|B)$

Aussagenlogik in NORs:

A "NOR" $B \equiv A \# B \equiv \neg(A \vee B)$

Umformungen:

- $\neg A \equiv A\#A$
- $A \wedge B \equiv (A\#A)\#(B\#B)$
- $A \vee B \equiv (A\#B)\#(A\#B)$

Definitionen:

Aussagenlogische Aussage: sprachlicher Ausdruck, der entweder wahr oder falsch sein muss.
Beispiel: „Diese Aussage ist falsch" ist keine Aussage.

Logischer Schluss: es ist keine Welt denkbar, wo die Prämissen gelten, aber die Konklusion nicht.

Atomaussage: nicht weiter zerlegbare Aussage

Literal: sei F Atomaussage $\rightarrow$ $F,\ \neg F$ Literale

Belegung: Funktion, die jeder Atomaussage einer Formel einen Wahrheitswert zuorndet.

Modell: Belegung $\mathcal{A}$, für die eine Formel F wahr wird
Schreibweise: $\mathcal{A} \vDash F$ für $\mathcal{A}(F) = 1$

Semantisch äquivalent: gleiche Wahrheitstabelle, bzw. gleicher Wahrheitswert für alle Belegeungen.
$(F \equiv G) \leftrightarrow (\mathrm{F} \leftrightarrow \mathrm{G}$ Tautologie$)$

Tautologie: Formel, die für jede Belegung Wert 1 hat

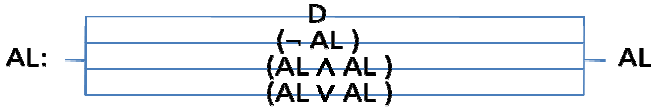
Unerfüllbar: Formel, die für jede Belegung Wert 0 hat

Beziehungen:

- F Tautologie $\leftrightarrow \neg F$ unerfüllbar
- F erfüllbar $\leftrightarrow \neg F$ keine Tautologie

Tertium non Datur: $\neg\neg \mathrm{A} \leftrightarrow \mathrm{A}$
Wenn wir den Wahrheitswert aller Atomaussagen einer Formel kennen, kennen wir auch den Wahrheitswert der gesamten Formel.

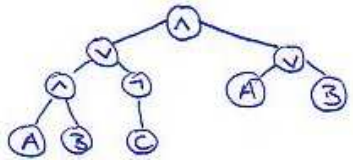
Strenge Syntax der Aussagenlogik:
(über eine Menge D atomarer Formeln):



Beispiele bei der strengen Syntax:

- $(A \wedge \mathrm{B}) \rightarrow$ korrekt
- $A \wedge B \wedge \mathrm{C} \rightarrow$ nicht korrekt

Syntaxbaum der Formel: $\big(\big((A \wedge B) \vee \neg \mathrm{C}\big) \wedge (A \vee \mathrm{B})\big)$



Rein syntaktisch existieren natürlich unendlich viele Formeln, allerdings ist die Anzahl semantisch äquivalenter Formeln beschränkt.

Semantische Folgerung:
G ist semantische Folgerung von $F_1, F_2, \ldots, F_n$ genau dann, wenn (alle Bedingungen äquivalent):

1. $F_1 \wedge F_2 \wedge \ldots \wedge F_n \wedge (\neg G)$ unerfüllbar ist
2. $F_1 \wedge F_2 \wedge \ldots \wedge F_n \rightarrow G$ Tautologie ist

3. $\forall \mathcal{A}: (\mathcal{A} \vDash F_1, \dots, \mathcal{A} \vDash F_n) \rightarrow \mathcal{A} \vDash G$

Man schreibt: $F_1, F_2, \dots, F_n \vDash G$

Junktorregeln (symmetrisch):

Konjunktiv	Disjunktiv
$x \wedge y \equiv y \wedge x$	$x \vee y \equiv y \vee x$
$x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$	$x \vee (y \vee z) \equiv (x \vee y) \vee z$
$x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$	$x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$
$\neg(x \wedge y) \equiv \neg x \vee \neg y$	$\neg(x \vee y) \equiv \neg x \wedge \neg y$
$x \wedge x \equiv x$	$x \vee x \equiv x$
$x \wedge 0 \equiv 0$	$x \vee 1 \equiv 1$
$x \wedge 1 \equiv x$	$x \vee 0 \equiv x$
$x \wedge y \equiv x$, falls $x \rightarrow y$	$x \vee y \equiv x$, falls $y \rightarrow x$
$F \wedge (\mathrm{F} \vee \mathrm{G}) \equiv F$	$F \vee (\mathrm{F} \wedge \mathrm{G}) \equiv F$

Junktorregeln (asymmetrisch):

- $A \,\mathrm{XOR}\, B \;\equiv\; A \oplus B \;\equiv\; (A \wedge \neg B) \vee (\neg A \wedge B)$
- Implikation: $A \rightarrow B \equiv \neg \mathrm{A} \vee \mathrm{B}$
- Äquivalenz: $A \leftrightarrow B \equiv (\neg A \vee B) \wedge (\neg B \vee A)$
- Konsensusregel: $(x \wedge y) \vee (\neg x \wedge z) \vee (y \wedge z) \equiv (x \wedge y) \vee (\neg x \wedge z)$
- Shannon-Expansion: $F \equiv x \wedge F[x/1] \vee \neg x \wedge F[x/0]$, wobei $F[x/0]$: alle x durch 0 ersetzen

Tseitin-Trnasformationen:

- $\boldsymbol{x \leftrightarrow \neg y} \equiv (\neg x \vee \neg y) \wedge (x \vee y)$
- $\boldsymbol{x \leftrightarrow (y \vee z)} \equiv (x \vee \neg y) \wedge (x \vee \neg z) \wedge (\neg x \vee y \vee z)$
- $\boldsymbol{x \leftrightarrow (y \wedge z)} \equiv (\neg x \vee y) \wedge (\neg x \vee z) \wedge (x \vee \neg y \vee \neg z)$
- $\boldsymbol{x \leftrightarrow (y \leftrightarrow z)} \equiv (\neg x \vee \neg y \vee z) \wedge (\neg x \vee y \vee \neg z) \wedge (x \vee \neg y \vee \neg z) \wedge (x \vee y \vee z)$
- $\boldsymbol{x \leftrightarrow (y \rightarrow z)} \equiv (\neg x \vee \neg y \vee z) \wedge (y \vee z) \wedge (\neg z \vee x)$

Normalformen:

Disjunktive Normalform (DNF):

$F = (\dots \wedge \dots) \vee (\dots \wedge \dots) \vee (\dots \wedge \dots)$

Unerfüllbar genau dann, wenn alle Blöcke unerfüllbar.
Erfüllbar, falls ein Block erfüllbar.

Konjunktive Normalform (KNF):

$F = (\dots \vee \dots) \wedge (\dots \vee \dots) \wedge (\dots \vee \dots)$

Klauselmengenschreibweise: $F = \{\{\dots\}, \{\dots\}, \{\dots\}\}$

Tautologie genau dann, wenn alle Blöcke Tautologien.
Falsifizierbar, falls ein Block falsifizierbar.

NP-vollständig sind:

- Erfüllbarkeitstest in KNF (3SAT)
- Tautologietest in DNF
- Umwandlung KNF ⟷ DNF

Resolution:

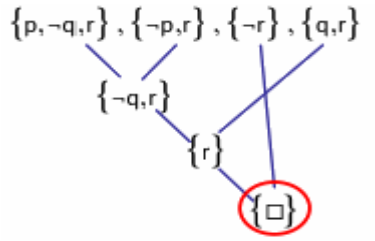
Resolvent: $R = (K_1 \backslash \{L\}) \cup (K_2 \backslash \{\neg L\})$ Resolvent von K_1 und K_2 , falls $\exists L: (L \in K_1) \wedge (\neg L \in K_2)$.
R ist dann eine semantische Folgerung: $K_1, K_2 \vDash R$

Resolventenmenge: alle Klauseln, die durch den nächsten Resolutionsschritt der Formel F entstehen können.
$Res(F) = \{R \mid R \text{ ist Resolvent zweier Klauseln in } F\} \cup F$
Es gilt: $Res(F) \equiv F$

Maximale Resolventenmenge $Res^*(F)$**:** alle möglichen Klauseln, die durch Resolution von F entstehen können.

Die Resolution ist ein Test auf Unerfüllbarkeit. Es gilt nämlich: F unerfüllbar ⟷ $\square \in Res^*(F)$

Beispiel:
Versuch, $\{r\}$ aus den anderen 3 Klauseln zu folgern



Beachte:

- mit Tautologien nicht weiterresolvieren
- pro Klammer nur EIN Literal streichen

Falls die Resolution fehlschlagen sollte, erhält man sofort ein Modell der Formel, indem man alle Klauseln aus $Res^*(F)$ wahr werden lässt.

Hornlogik:

Hornklausel: Klausel mit maximal einem positiven Literal

Hornformel: konjunkte Hornklauseln

Aus einer Hornklausel entsteht nach einem Resolutionsschritt immer wieder eine Hornklausel.

Man unterscheidet 3 Hornformelarten:

1. Faktum: positives Literal
2. Regel: mind. 1 positives Literal, rest negativ
   $\{\neg A, \neg B, C\} \leftrightarrow (A \wedge B \rightarrow C)$
3. Query: nur negative Literale
   Negation der Anfrage

Resolvieren wir diese Klauseln zu einer Unerfüllbarkeit, wissen wir, dass die Anfrage stimmt.

Durch Umbenennung von Literalen ist es oft möglich, aus gewöhnlichen Formeln Hornformeln herzustellen.

Es existieren 2 Möglichkeiten zur Hornresolution:

Markierungsalgorithmus:

$\{\neg A\}, \{\neg D, \neg C, A\}, \{\neg C, D\}, \{\neg B, \neg E, C\}, \{\neg E, B\}, \{E\}$

Fakten: E
Regeln: $D \wedge C \rightarrow A,\ C \rightarrow D,\ B \wedge E \rightarrow C,\ E \rightarrow B$
Query: $\{\neg A\}$ mit dem Ziel, dass es unerfüllbar wird

Der Reihe nach alle Vorkommen von E markieren, nun sieht man, dass B impliziert wird. Jetzt markiert man alle Bs. Dies führt man so lange weiter, bis die ganze Query markiert ist und somit eine Unerfüllbarkeit entsteht.

SLD Resolution:
Resolviere sequentiell die Query mit Klauseln aus der Grundmenge bis alle Fakten und Regelen aufgebraucht sind oder die leere Menge entsteht.
Dabei ist jede Klausel nur ein mal zu verwenden. Aus diesem Grund ist der Algorithmus linear.

Prädikatenlogik

In der Prädikatenlogik existiert Unentscheidbarkeit. Zudem existieren erfüllbare Formeln, die nur unendliche Modelle besitzen.

Die Unerfüllbarkeit ist in der Prädikatenlogik mittels Resolution semi-entscheidbar: ein positiver Unerfüllbarkeitstest is zuverlässig, aber wenn der Algorithmus nicht terminiert, kann man keine Aussage über die Unerfüllbarkeit machen.

Für Erfüllbarkeit gibt es keinen Algorithmus.

Definitionen:

Elemente der Prädikatenlogik:

1. Universum: Nicht-leere Menge aller Dinge, über die wir sprechen
2. Variablen: Elemente des Universums
3. Funktionen: Zuordnungen und Konstanten
4. Prädikate: Elemente mit Wahrheitswert
5. Quantoren: ∀ für alle ; ∃ es existiert
6. Aussagenlogische Junktoren zwischen Prädikaten

Funktionen: nullstellige Funktionen heissen Konstanten, alle anderen Funktionen sind Zuordnungen.

Term: jede Zusammensetzung von Konstanten und Funktionen ist ein Term.

Prädikate: nullstellige Prädikate sind logische Konstanten $\in \{0,1\}$, einstellige heissen Eigenschaften, zweistellige heissen Relationen.

Formel: Junktionen von Prädikaten von Termen.

Struktur: $\mathcal{A}$ legt Universum und Interpretation (Festlegung aller Funktionen, freien Variablen und Prädikaten) der Formel F fest $\rightarrow \mathcal{A}(F) = (U, I)$

Modell: $\mathcal{A}$ Modell von F, falls $\mathcal{A}$ zu F passt und $\mathcal{A}(F) = 1$. F ist erfüllbar, wenn mindestens ein Modell existiert. Man schreibt: $\mathcal{A} \vDash F$

Gültig: Eine Formel heisst gültig, falls alle zu ihr passenden Strukturen Modelle sind. Eine beliebte Methode, um Gültigkeit zu beweisen, ist die Zurückführung auf eine Aussagenlogische Tautologie. Dies kann durch Textsubstitution erreicht werden.

Man unterscheidet freie und gebundene Variablen:

- **gebunden** (Laufvariablen): ein Quantor bindet alle freien vorkommen seiner Variable in seinem Unterbaum
- **frei** (dürften umbenannt werden, ohne Quantoren zu verändern): ein Vorkommen einer Variable heisst „frei“ wenn es nicht gebunden ist

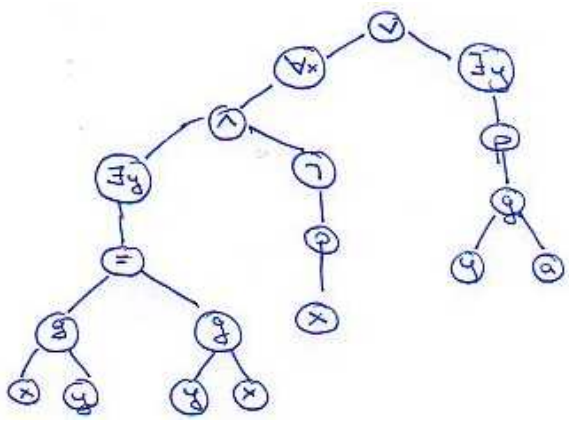
Bindungsregel: Eine Variable wird immer vom tiefsten Quantor über ihr gebunden. Also gilt:

$$\forall x \exists x\ P(x) \equiv \exists x\ P(x)$$

Die Menge $\boldsymbol{Frei(F)}$ der freien Variablen einer Formel F beinhaltet alle Variablen, die mindestens ein Mal frei in F vorkommen. es gilt:

- F heisst **geschlossen**, falls $Frei(F) = \emptyset$
- $Frei(F) = Frei(\neg F)$

Syntaxbäume existieren auch in der Prädikatenlogik:
$F \equiv \forall x \left(\exists y \left(g(x,y) = g(x,y)\right) \wedge \neg P(x)\right) \vee \exists y \left(P\left(g(y,a)\right)\right)$



Substitution: $F[x\ /a]$ bedeutet, dass alles freien Auftreten von x in F durch a ersetzt werden

Es gilt: t Term, t enthält keine Variable, über die F oder in F quantisiert wird: $\forall x(F) \rightarrow F[x/t]$ gültig.

Quantorregeln:

- $\neg \forall x\ P(x) \equiv \exists x \neg P(x)$
- $\neg \exists x\ P(x) \equiv \forall x \neg P(x)$
- $\forall x \forall y\ F \equiv \forall y \forall x\ F$
- $\exists x \exists y\ F \equiv \exists y \exists x\ F$
- !! $\forall x \exists y\ F \not\equiv \exists y \forall x\ F$!!
  Aber: $\exists y \forall x\ F \rightarrow \forall x \exists y\ F$
- $(\forall x\ F) \wedge (\forall x\ G) \equiv \forall x\ (F \wedge G)$
- $(\exists x\ F) \vee (\exists x\ G) \equiv \exists x\ (F \vee G)$

- !! $(\forall x\, F) \vee (\forall x\, G) \not\equiv \forall x\, (F \vee G)$!!
  Aber: $(\forall x\, F) \vee (\forall x\, G) \rightarrow \forall x\, (F \vee G)$
- !! $\exists x\, (F \wedge G) \not\equiv (\exists x\, F) \wedge (\exists x\, G)$!!
  Aber: $\exists x\, (F \wedge G) \rightarrow (\exists x\, F) \wedge (\exists x\, G)$

Resolution in der Prädikatenlogik:

F normalisieren und auf erfüllbarkeitsäquivalente Aussagenlogische Formel F^*zurückführen. Danach AL-Resolution auf F^* anwenden.
Beachte: F und F^* sind NICHT äquivalent.

Vorgehen:

1. **Bereinigen:** alle Variablenkollisionen beseitigen.
   $\forall x \exists y\, P(x,y) \vee \exists x Q(x,y) \rightarrow \forall u \exists v\, P(u,v) \vee \exists x Q(x,y)$
2. **Pränexform** (BPF): Alle Quantoren an Formelanfang setzen und Reihenfolge beibehalten
3. Alle freien Variablen durch ∃-Quantoren **binden:**
   $\forall x P(x,y) \rightarrow \exists y \forall x P(x,y)$

Nach Schritt 3 ist das aktuelle, vereinfachte F' nicht mehr äquivalent zu F.

4. **Skolemform** (SKF): ∃-Quantoren loswerden und ihre Variablen durch Funktionen der Variablen aller ∀-Quantoren links vom ∃-Quantor ersetzen:
   $\exists y \forall x P(x,y) \rightarrow \forall x P\big(x, f(x)\big)$
5. **Matrixklauselform** (F^*): Streichen aller ∀-Quantoren: $\forall x P\big(x, f(x)\big) \rightarrow P\big(x, f(x)\big)$
6. **Klauselmenge:** F^* in KNF bringen und als Klauselmenge schreiben
7. **Resolution:** Geeignete Substitution der Variablen ermöglichen Resolutionsschritte.

Vorgehen bei einem Resolutionsschritt:
In den Klauseln treten Prädikate von Funktionen, Konstanten und Variablen auf.
Resolutionsschritt:

- 2 resolvierbare Klauseln suchen
- Substituionen aller Variablen in den einzelnen Klauseln durchführen, bis keine Variable in beiden Klauseln vorkommt
- Gemeinsame Substitution der Vairablen in beiden Klauseln mit einem Unifikator sub der Literalmenge zweier nichtleeren unifizierbaren Teilmengen der beiden Klauseln
- Der Resolvent ist die Verinigung der beiden Klauseln ohne die unifizierte Literalmenge

Analog zur Aussagenlogik resolviert man bis eine leere Menge auftritt oder keine Resolutionsschritte mehr möglich sind.

Beispiel:
$\{P(f(x)), \neg Q(z), P(z)\}$ $\quad\quad$ $\{\neg P(x), R(g(x), a)\}$
$\{P(f(x)), \neg Q(z), P(z)\}$ $\quad\quad$ $\{\neg P(u), R(g(u), a)\}$

$$sub = [z/f(x)][u/f(x)]$$
$$R = \{\neg Q(f(x)), R(g(f(x)), a)\}$$

Der „variabelste“ Unifikator einer Literalmenge heisst allgemeinster Unifikator.

Logikprogrammierung: Prolog

Syntax:

	Normale Syntax	Prolog Syntax
Fakten	$P(a)$	`P(a).`
Regeln	$R(a) \wedge P(b) \rightarrow Q(c)$	`Q(c):-R(a),P(b).`
Query	gilt $P(c)$?	`?-P(c).`

Beispiel:

`Vat(a,b):` a ist Vater von b
`Mut(a,b):` a ist Mutter von b
Gesucht: Prozedurklauseln für

- X ist Elternteil von Y:
  ```
  Elt(X,Y) :- Mut(X,Y)
  Elt(X,Y) :- Vat(X,Y)
  ```
- X und Y sind Geschwister :
  ```
  Gesch(X,Y) :- Elt(Z,X), Elt(Z,Y)
  ```
- X und Y sind verwandt:
  ```
  Verw(X,Y) :- Vorf(X,Y), Vorf(Z,Y)
  ```
  mit Klausel für Vorfahren:
  ```
  Vorf(X,Y) :- X = Y
  Vorf(X,Y) :- Elt(X,Z), Vorf(Z,Y)
  ```