

# Aussagenlogik und Prädikatenlogik

**Zusammenfassung**

Fabian Damken

8. November 2023



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

---

# Inhaltsverzeichnis

---

<b>1</b>	<b>Aussagenlogik</b>	<b>3</b>
1.1	Grundlegende Begriffe . . . . .	3
1.2	Notation . . . . .	4
1.3	Hornklauseln . . . . .	4
1.3.1	Horn-Erfüllbarkeitstest . . . . .	4
1.4	Kalküle . . . . .	5
1.4.1	Resolutionskalkül . . . . .	5
1.4.2	Sequenzenkalkül . . . . .	6
1.5	Kompaktheitssatz . . . . .	6
1.6	Normalformen . . . . .	7
1.6.1	Disjunktive Normalformmodus ponens (DNF) . . . . .	7
1.6.2	Konjunktive Normalform (KNF) . . . . .	7
<b>2</b>	<b>Logik erster Stufe</b>	<b>8</b>
2.1	Grundlegende Begriffe . . . . .	8
2.2	Herbrand . . . . .	10
2.2.1	Herbrandstrukturen . . . . .	10
2.2.2	Herbrandmodelle . . . . .	10
2.3	Kalküle . . . . .	10
2.3.1	Grundinstanzen-Resolutionskalkül . . . . .	10
2.3.2	Sequenzenkalkül . . . . .	11
2.4	Kompaktheitssatz . . . . .	11
2.5	Normalformen . . . . .	12
2.5.1	Pränexe Normalform (PNF) . . . . .	12
2.5.2	Skolem-Normalform (SNF) . . . . .	12
2.5.3	Negationsnormalform (NNF) . . . . .	12
2.5.4	Gleichheitsfreie Normalform (GNF) . . . . .	13
2.6	Spielsemantik . . . . .	13
<b>3</b>	<b>Entscheidbarkeit</b>	<b>15</b>
<b>4</b>	<b>Abkürzungen und Begriffe</b>	<b>16</b>

---

# 1 Aussagenlogik

---

Die Aussagenlogik beschäftigt sich mit Grundlegenden Aussagen, in denen Fakten (Atome), denen ein Wahrheitswert zugeordnet werden kann, miteinander verknüpft werden.

---

## 1.1 Grundlegende Begriffe

---

**Allgemeingültigkeit** Eine Formel  $\varphi$  ist allgemeingültig gdw. alle möglichen Interpretationen  $\mathcal{I}$  die Formel  $\varphi$  erfüllen. Eine solche Formel wird auch Tautologie genannt.

**Atome** Eine Formel  $\varphi$  ist atomar (ist ein Atom) gdw. sie in der Form  $p$  oder der Form  $\neg p$  vorliegt, wobei  $p$  eine eigenständige Variable darstellt. Diese werden auch Literale genannt.

**Erfüllbarkeit** Eine Interpretation  $\mathcal{I}$  erfüllt eine Formel  $\varphi$  gdw. die Formel  $\varphi$  mit der Belegung  $\mathcal{I}$  wahr wird. Diese Aussage ist äquivalent zu den folgenden Aussagen:

- $\varphi$  ist wahr unter  $\mathcal{I}$ .
- $\mathcal{I}$  ist ein Modell von  $\varphi$ .

**Formel** Eine Formel ist eine Menge von Atomen, welche mit booleschen Junktoren und Klammern zu einer Aussage zusammengefasst werden. Formel werden häufig mit  $\varphi$  oder  $\psi$  bezeichnet.

**Interpretation** Eine Interpretation ist eine Belegung der Variablen in einer Formel mit Wahrheitswerten (wahr/falsch), sodass die Formel ausgewertet werden kann. Interpretationen werden häufig mit  $\mathcal{I}$  bezeichnet.

**Klausel** Formeln in Klauselform ist eine Schreibweise für Formeln, in denen Atome ausschließlich mit Disjunktionen verknüpft sind. Hierbei werden die Atome als Elemente einer Menge verstanden, was die Schreibweise ergibt.

**Beispiel:**  $(a \vee b \vee \neg c)$  ist eine solche Formel, welche als Klauselform folgendermaßen dargestellt wird:  $\{a, b, \neg c\}$ .

**Klauselmengen** Eine Klauselmenge ist eine Menge von Formeln in Klauselform, welche bei der Umformung in einen logischen Ausdruck mit konjunktionen Verknüpft werden.

**Beispiel:**  $((a \vee b) \wedge (b \vee c) \wedge (a \vee \neg d \vee \neg e) \wedge d)$  ist eine Formel in konjunktiver Normalform, welche als Klauselform folgendermaßen dargestellt wird:  $\{\{a, b\}, \{b, c\}, \{a, \neg d, \neg e\}, \{d\}\}$ . Hierbei stellen die inneren Mengen die Klauseln dar und die äußere Menge die Klauselmenge.

**Minimales Modell** Das minimale Modell  $\mathcal{I}$  einer Formel  $\varphi$  ist das Modell mit der minimalen Anzahl an 1-Belegungen, um  $\varphi$  zu erfüllen.

---

**Unerfüllbarkeit** Eine Formel  $\varphi$  ist unerfüllbar gdw. die Formel  $\varphi$  unter allen möglichen Interpretationen  $\mathcal{I}$  nicht wahr wird. Dies ist äquivalent dazu, dass das Negat der Formel  $\varphi$  (also  $\neg\varphi$ ) allgemeingültig ist.

---

## 1.2 Notation

---

Sei  $\mathcal{I}$  eine Interpretation, und seien  $\varphi, \psi$  Formeln.

$\models$  Das Symbol  $\models$  kann folgende Dinge beschreiben:

- $\mathcal{I} \models \varphi$  bedeutet  $\mathcal{I}$  ist ein Modell von  $\varphi$
- $\varphi \models \psi$  bedeutet  $\varphi$  impliziert  $\psi$
- $\models \varphi$  bedeutet  $\varphi$  ist allgemeingültig

$\equiv$  Das Symbol  $\equiv$  kann folgende Dinge beschreiben:

- $\varphi \equiv \psi$  bedeutet  $\varphi$  ist logisch äquivalent zu  $\psi$

---

## 1.3 Hornklauseln

---

Eine Hornklausel (oder auch Horn-Formel) ist eine Formel in Klauselform, welche maximal ein positives Literal hat.

**Warning:** Nicht jede Formel lässt sich als Hornklauselmenge darstellen!

Eine Hornklausel heißt

**negativ** wenn sie keine positiven Literale.

**positiv** wenn sie ausschließlich positive Literale besitzt.

Werden mehrere Horn-Formeln in einer Klauselmenge zusammengefasst, so spricht man von einer Hornklauselmenge.

---

### 1.3.1 Horn-Erfüllbarkeitstest

---

Der Horn-Erfüllbarkeitstest ist ein effizienter Test um zu prüfen, ob eine Hornklauselmenge erfüllbar ist.

---

**Eingabe** : Hornklauselmenge  $K$

**Ergebnis** :  $K$  erfüllbar mit gegebenem minimalen Modell oder unerfüllbar

```
1 for  $\psi \in K$  der Form  $\psi = x$  do
2   | markiere  $x$ 
3 for  $\psi \in K$  der Form  $\psi = \neg x_1 \vee \dots \vee \neg x_n$  (Typ 1) oder  $\psi = \neg x_1 \vee \dots \vee \neg x_n \vee y$  (Typ 2), wobei
    $x_1, \dots, x_n$  markiert sind und, im Falle von Typ 2,  $y$  noch nicht markiert ist do
4   | if  $\psi$  ist vom Typ 1 then
5     |  $\implies$  unerfüllbar
6   | else if  $\psi$  ist vom Typ 2 then
7     | Markiere  $y$ 
8  $\implies$  erfüllbar mit minimalem Modell gegeben durch die markierten Variablen
```

---

## 1.4 Kalküle

---

### 1.4.1 Resolutionskalkül

---

Das Resolutionskalkül ist ein Beweiskalkül, welches zum Beweisen der Unerfüllbarkeit von Klauselmengen.

Eine Klausel  $C$  heißt Resolvente von  $C_1$  und  $C_2$ , wenn für ein Literal  $L$  gilt

$$L \in C_1, \neg L \in C_2 \text{ und } C = (C_1 \setminus \{L\}) \cup (C_2 \setminus \{\neg L\})$$

**Algorithmus** Die Funktion  $\text{Res}(M)$  produziert alle möglichen Resolventen der Klauselmenge  $M$ .

---

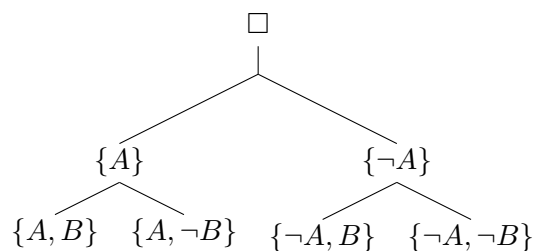
**Input** : Klauselmenge  $K$

**Ergebnis** :  $K$  erfüllbar oder unerfüllbar

```
1 while  $\text{Res}(R) \neq R \wedge \square \notin R$  do
2   |  $R \leftarrow \text{Res}(R)$ 
3 if  $\square \in R$  then
4   |  $\implies$  unerfüllbar
5 else
6   |  $\implies$  erfüllbar
```

---

**Beispiel**  $K := \{\{A, B\}, \{A, \neg B\}, \{\neg A, B\}, \{\neg A, \neg B\}\}$



**Einheitsresolution** Als Einheitsresolution wird eine Resolution benannt, bei denen ausschließlich Klauseln mit 1-elementigen Klauseln resoliert.

## 1.4.2 Sequenzenkalkül

### Schlussregeln in $\mathcal{SK}$ für AL

Das Sequenzenkalkül  $\mathcal{SK}$  definiert die Schlussregeln in Abbildung 1.1.

(Axiom)	$\frac{}{\Gamma, \varphi \vdash \Delta, \varphi}$		
(0-Axiom)	$\frac{}{\Gamma, 0 \vdash \Delta}$ $\frac{}{\Gamma \vdash \Delta, \varphi}$	(1-Axiom)	$\frac{}{\Gamma \vdash \Delta, 1}$ $\frac{}{\Gamma, \varphi \vdash \Delta}$
( $\neg L$ )	$\frac{\Gamma, \neg \vdash \Delta}{\Gamma, \varphi \vdash \Delta \quad \Gamma, \psi \vdash \Delta}$	( $\neg R$ )	$\frac{\Gamma \vdash \Delta, \neg \varphi}{\Gamma \vdash \Delta, \varphi, \psi}$
( $\vee L$ )	$\frac{\Gamma, \varphi \vee \psi \vdash \Delta}{\Gamma, \varphi, \psi \vdash \Delta}$	( $\vee R$ )	$\frac{\Gamma \vdash \Delta, \varphi \vee \psi}{\Gamma \vdash \Delta, \varphi \quad \Gamma \vdash \Delta, \psi}$
( $\wedge L$ )	$\frac{}{\Gamma, \varphi \wedge \psi \vdash \Delta}$	( $\wedge R$ )	$\frac{}{\Gamma \vdash \Delta, \varphi \wedge \psi}$

Abbildung 1.1: Schlussregeln in  $\mathcal{SK}$

### Schlussregeln in $\mathcal{SK}^+$ für AL

Das Sequenzenkalkül  $\mathcal{SK}^+$  erweitert  $\mathcal{SK}$  (1.1) um Schnittregeln, unter anderem um modus ponens, wodurch sich eine Vielzahl weiterer Regeln herleiten lässt. Die wichtigsten Regeln (inklusive der Kernschnittregeln modus ponens) sind in Abbildung 1.2 aufgezeigt.

	$\frac{\Gamma \vdash \varphi \quad \Gamma', \varphi \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta}$
(modus ponens)	$\frac{\Gamma \vdash \varphi \quad \Gamma' \vdash \neg \varphi}{\Gamma, \Gamma' \vdash \emptyset}$
(Kontradiktion)	$\frac{\Gamma, \neg \varphi \vdash \psi \quad \Gamma, \neg \varphi \vdash \neg \psi}{\Gamma \vdash \varphi}$
(Widerspruch)	

Abbildung 1.2: Schlussregeln in  $\mathcal{SK}^+$

## 1.5 Kompaktheitssatz

Eine (möglicherweise unendliche) Formelmenge  $\Sigma$  ist erfüllbar gdw. jede endliche Teilmenge von  $\Sigma$  erfüllbar ist.

---

## 1.6 Normalformen

---

In der Aussagenlogik existieren die disjunktive Normalform und die konjunktive Normalform, welche sich in ihrer Komplexität stark unterscheiden. Ebenfalls ist ein effizientes Umrechnen von KNF  $\leftrightarrow$  DNF nicht möglich.

---

### 1.6.1 Disjunktive Normalformmodus ponens (DNF)

---

Eine Formel  $\varphi$  ist in disjunktiver Normalform gdw. sie die folgende Form hat ( $p_{ij}$  ist ein Atom):

$$\varphi = \bigvee_i \bigwedge_j p_{ij}$$

**Beispiel** Gegeben sei die Formel  $\varphi := \neg(p \vee (\neg(p \wedge q) \wedge \neg r)) \vee s$ . Diese ist logisch äquivalent zu folgender Formel in konjunktiver Normalform:  $\varphi \equiv (\neg p \vee s) \wedge (p \vee r \vee s) \wedge (q \vee r \vee s)$ .

---

### 1.6.2 Konjunktive Normalform (KNF)

---

Eine Formel  $\varphi$  ist in konjunktiver Normalform gdw. sie die folgende Form hat ( $p_{ij}$  ist ein Atom):

$$\varphi = \bigwedge_i \bigvee_j p_{ij}$$

**Beispiel** Gegeben sei die Formel  $\varphi := \neg(p \vee (\neg(p \wedge q) \wedge \neg r)) \vee s$ . Diese ist logisch äquivalent zu folgender Formel in disjunktiver Normalform:  $\varphi \equiv (r \wedge \neg p) \vee s$ .

---

## 2 Logik erster Stufe

---

### 2.1 Grundlegende Begriffe

---

**Formeln** Eine Formel der Logik erster Stufe ist eine Kombination aus Quantoren, Variablen, Funktionen und Relationen, welche mit booleschen Junktoren Verknüpft werden. Typische Namen für Formeln sind  $\varphi$  und  $\psi$ .

**freie/gebundene Variablen** Eine freie Variable ist nicht an einen Quantor gebunden ((ab-) quantifiziert). Die Menge dieser Variablen wird mit der Funktion  $\text{frei}(\varphi)$  rekursiv formalisiert:

$$\begin{aligned}\text{frei}(t_1 = t_2) &:= \text{var}(t_1) \cup \text{var}(t_2) \\ \text{frei}(Rt_1 \cdots t_n) &:= \text{var}(t_1) \cup \cdots \cup \text{var}(t_n) \\ \text{frei}(\neg\varphi) &:= \text{frei}(\varphi) \\ \text{frei}(\varphi \wedge \psi) = \text{frei}(\varphi \vee \psi) &:= \text{frei}(\varphi) \cup \text{frei}(\psi) \\ \text{frei}(\forall x\varphi) = \text{frei}(\exists x\varphi) &:= \text{frei}(\varphi) \setminus \{x\}\end{aligned}$$

#### Interpretationen

**Quantoren** In der Logik erster Stufe wird zwischen den folgenden Quantoren unterschieden:

**Allquantor** Der Allquantor ( $\forall$ ) bindet (quantifiziert) die nachstehende Variable an sich und sagt aus, dass die folgende Aussage im Bezug auf den Quantor für alle Elemente aus der Trägermenge der Signatur gilt.

**Beispiel:**  $\forall x(x > 0)$  ist semantisch äquivalent zu „alle Elemente sind größer als 0“.

**Existenzquantor** Der Existenzquantor ( $\exists$ ) bindet (quantifiziert) die nachstehende Variable an sich und sagt aus, dass die folgende Aussage im Bezug auf den Quantor für mindestens ein Element aus der Trägermenge gilt.

**Beispiel:**  $\exists x(x > 0)$  ist semantisch äquivalent zu „mindestens ein Element ist größer als 0“.

**Quantorenrang** Als *Quantorenrang* wird die Anzahl der Quantoren bezeichnet, was durch die Funktion



$qr(\varphi)$  rekursiv formalisiert wird:

$$\begin{aligned}qr(\varphi) &:= 0 && (\varphi \text{ atomar}) \\qr(\neg\varphi) &:= qr(\varphi) \\qr(\varphi \wedge \psi) = qr(\varphi \vee \psi) &:= \max(qr(\varphi), qr(\psi)) \\qr(\forall x\varphi) = qr(\exists x\varphi) &:= qr(\varphi) + 1\end{aligned}$$

**Symbole** In Formeln der Logik erster Stufe können die Folgenden Typen von Symbolen genutzt werden (sei  $A$  die Trägermenge):

**Konstantensymbole** Konstantensymbole, meist mit  $a, b, c, \dots$  bezeichnet, sind Elemente aus  $A$ , welche Konstant in einer Formel stehen und nicht von einer Interpretation belegt oder an Quantoren gebunden werden können.

**Funktionssymbole** Funktionssymbole, meist mit  $f, g, h, \dots$  bezeichnet, sind  $n$ -stellige Funktionen der Form  $A^n \rightarrow A$ .

**Notation:** Sei  $f$  ein 2-stelliges Funktionssymbol, so ist  $fxy$  als ein Aufruf dieser mit den Parametern  $x, y$  zu verstehen.

**Relationssymbole** Relationssymbole, meist mit  $R, L, P, \dots$  bezeichnet, sind  $n$ -stellige Teilmengen von  $A^n$ .

**Prefix-Notation:** Sei  $R$  ein 3-stelliges Relationssymbol, so ist  $Rxyz$  als ein Test zu verstehen, ob  $x, y, z$  in Relation stehen.

**Infix-Notation:** Sei  $\leq$  ein 2-stelliges Relationssymbol, so ist  $x \leq y$  als ein Test zu verstehen, ob  $x, y$  in Relation stehen.

**Sätze** Eine Formel  $\varphi$  wird als *Satz* bezeichnet, wenn gilt  $\text{frei}(\varphi) = \emptyset$ . In anderen Worten: Dass die Formel keine freien Variablen enthält.

$\varphi(t/x)$  besagt, dass die freie Variable  $x$  in  $\varphi$  durch  $c$  ersetzt wird.

**Beispiel:**

$$\begin{aligned}\varphi(x) &:= \forall n(n = x \vee n \neq x) \\ \varphi(c/x) &:= \forall n(n = c \vee n \neq c)\end{aligned}$$

---

## 2.2 Herbrand

---

### 2.2.1 Herbrandstrukturen

---

### 2.2.2 Herbrandmodelle

---

## 2.3 Kalküle

---

### 2.3.1 Grundinstanzen-Resolutionskalkül

---

Die Grundinstanzen-Resolution ist eine Erweiterung des Resolutionskalküls der Aussagenlogik (1.4.1). Somit kann es ebenfalls genutzt werden, um die Erfüllbarkeit und Unerfüllbarkeit einer Klauselmengen  $\Phi$  zu zeigen. Hierzu müssen alle Formeln, welche die Klauselmengen produzieren, in Skolem-Normalform vorliegen. Die eigentliche Formel ( $\varphi$  für  $\forall x_1 \dots \forall x_n \varphi$ ) entspricht dann einer Klausel der Klauselmengen. Die Klauselmengen wird nun für die Resolution vorbereitet, indem die folgenden Substitutionen vorgenommen werden können:

- Ersetzung einer Variable durch eine Konstante:  $P(x) \rightarrow P(c)$
- Ersetzung einer Variable durch eine andere Variable:  $P(x) \rightarrow P(y)$
- Ersetzung einer Variable durch eine Funktion:  $P(x) \rightarrow P(fx)$

Hierbei müssen stets alle Variablen einer Unterklauselmengen, welche aus einer Formel entstanden ist, substituiert werden!

**Beispiel** Seien die folgenden Sätze in FO gegeben:

$$\begin{aligned}\varphi_1 &:= \forall x \forall y (Rxy \rightarrow (Px \leftrightarrow Qy)) \\ &= \forall x \forall y ((\neg Rxy \vee Px \vee \neg Qy) \wedge (\neg Rxy \vee \neg Px \vee Qy)) \\ &\triangleq \{ \{ \neg Rxy, Px, \neg Qy \}, \{ \neg Rxy, \neg Px, Qy \} \} \\ \varphi_2 &:= \forall x \exists y (Rxy \wedge Py) \\ &\simeq \forall x (Rxfx \wedge Pfx) \\ &\triangleq \{ \{ Rxfx \}, \{ Pfx \} \} \\ \varphi_3 &:= \exists x (\neg Px \wedge \forall y (Qy \wedge (Py \rightarrow Rxy))) \\ &= \exists x \forall y (\neg Px \wedge (Qy \wedge (Py \rightarrow Rxy))) \\ &\simeq \forall y (\neg Pg \wedge (Qy \wedge (Py \rightarrow Pgy))) \\ &= \forall y (\neg Pg \wedge Qy \wedge (\neg Py \vee Rxy)) \\ &\triangleq \{ \{ \neg Pg \}, \{ Qy \}, \{ \neg Py, Rxy \} \}\end{aligned}$$

Dies ergibt, nach überlegten Substitutionen, die folgende Grundinstanzen-Klauselmenge:

$$\{ \begin{array}{l} \{\neg Rgfc, Pg, \neg Qfc\}, \{\neg Rgfc, \neg Pg, \neg Qfc\}, \\ \{Rgfc\}, \{Pfg\}, \\ \{\neg Pg\}, \{Qfc\}, \{\neg Pfc, Rgfc\} \end{array} \}$$

Wodurch eine Resolution möglich wird.

## 2.3.2 Sequenzenkalkül

### Schlussregeln in $\mathcal{SK}$ für FO

Das Sequenzenkalkül  $\mathcal{SK}$  für die Logik erster Stufe erweitert das Sequenzenkalkül  $\mathcal{SK}$  für die Aussagenlogik (siehe 1.1) auf die Logik erster Stufe aus. Es gelten somit auch alle Regeln aus dem Sequenzenkalkül  $\mathcal{SK}$  der Aussagenlogik. Die Schlussregeln sind in der Abbildung 2.1 aufgeführt.

$\frac{\Gamma, \varphi(t/x) \vdash \Delta}{\Gamma, \forall x \varphi(x) \vdash \Delta} \quad \forall L$	$\frac{\Gamma \vdash \Delta, \varphi(c/x)}{\Gamma \vdash \Delta, \forall x \varphi(x)} \quad \forall R$
$\frac{\Gamma, \varphi(c/x) \vdash \Delta}{\Gamma, \exists x \varphi(x) \vdash \Delta} \quad \exists L$	$\frac{\Gamma \vdash \Delta, \varphi(t/x)}{\Gamma \vdash \Delta, \exists x \varphi(x)} \quad \exists R$

Abbildung 2.1: Schlussregeln in  $\mathcal{SK}$

Achtung: Für die Regeln  $\exists L$ ,  $\forall R$  darf das  $c$  noch nicht in  $\Gamma, \Delta, \varphi(x)$  vorhanden sein.

### Schlussregeln in $\mathcal{SK}^=$ für FO

Das Sequenzenkalkül  $\mathcal{SK}^=$  erweitert  $\mathcal{SK}$  (siehe 2.1) um Schlussregeln für die Gleichheit ( $=$ ). Die Schlussregeln sind in 2.2 aufgeführt.

$\frac{\Gamma, t = t' \vdash \Delta}{\Gamma \vdash \Delta} \quad (=)$	$\frac{\Gamma \vdash \Delta, \varphi(t/x)}{\Gamma, t = t' \vdash \Delta, \varphi(t'/x)} \quad \text{(Sub-R)}$
$\frac{\Gamma, \varphi(t/x) \vdash \Delta}{\Gamma, t = t', \varphi(t'/x) \vdash \Delta} \quad \text{(Sub-L)}$	$\frac{\Gamma \vdash \Delta, \varphi(t'/x)}{\Gamma, t = t' \vdash \Delta, \varphi(t'/x)} \quad \text{(Sub-R)}$

Abbildung 2.2: Schlussregeln in  $\mathcal{SK}^=$

## 2.4 Kompaktheitssatz

Entspricht dem Kompaktheitssatz der Aussagenlogik (siehe 1.5).

Daraus folgt, dass für jede Formelmeng  $\Phi \subseteq FO$  und Formel  $\psi \in FO$  gilt:

$$\Phi \models \psi \iff \text{es existiert eine endliche Teilmenge } \Phi_0 \subseteq \Phi, \text{ sodass } \Phi_0 \models \psi$$

---

## 2.5 Normalformen

---

### 2.5.1 Pränexe Normalform (PNF)

---

Eine Formel  $\varphi$  ist in der *pränexen Normalform*, wenn in dieser alle Quantoren vor der eigentlichen Formel stehen, das heißt, sie hat die Form  $\forall x_1 \dots \forall x_n \varphi$ .

Um eine beliebige Formel  $\varphi$  in die pränexe Normalform umzuformen, müssen alle Quantoren nach vorne gezogen. Ist eine Variable nicht eindeutig an einen Quantor gebunden, so muss diese zunächst umbenannt werden.

**Beispiel** Sei  $\varphi := \forall x \exists y (x < y \wedge \exists y (x > y))$ .

Diese Formel wird nun schrittweise in die pränexe Normalform umgeformt:

$$\begin{aligned}\varphi &= \forall x \exists y (x < y \wedge \exists y (x > y)) && \text{(Umbenennung)} \\ &= \forall x \exists y (x < y \wedge \exists z (x > z)) && \text{(Quantoren-Verschiebung)} \\ &= \forall x \exists y \exists z (x < y \wedge x > z) && \text{(Pränexe Normalform)}\end{aligned}$$

---

### 2.5.2 Skolem-Normalform (SNF)

---

Eine Formel  $\varphi$  ist in der *Skolem-Normalform*, wenn in dieser ausschließlich Allquantoren existieren und diese vor der eigentlichen Formel stehen, das heißt, sie hat die Form  $\forall x_0 \dots \forall x_n \varphi$ .

Um eine Formel in pränexer Normalform  $\varphi$  in die Skolem-Normalform umzuwandeln, müssen alle Variablen, welche an Existenzquantoren gebunden sind, in (neue!)  $n$ -stellige Funktionen geändert werden, welche von allen vorherigen freien und an Allquantoren gebundenen Variablen abhängen. Existiert keine solche vorherige Variable, so wird die Variable durch ein 0-stelliges Funktionssymbol substituiert, d.h. durch eine (neue!) Konstante. Dieser Prozess wird *Skolemisierung* genannt.

**Beispiel** Sei  $\varphi := \exists x \forall y \exists z (x > y \vee y < z)$  in pränexer Normalform.

Diese Formel wird nun schrittweise in die Skolem-Normalform umgewandelt (skolemisiert):

$$\begin{aligned}\varphi &= \exists x \forall y \exists z (x > y \vee y < z) && \text{(1. Existenzquantor)} \\ &= \forall y \exists z (c > y \vee y < z) && \text{(2. Existenzquantor)} \\ &= \forall y (c > y \vee y < f y) && \text{(Skolem-Normalform)}\end{aligned}$$

---

### 2.5.3 Negationsnormalform (NNF)

---

Eine Formel  $\varphi$  ist in der *Negationsnormalform*, wenn sie ausschließlich aus atomaren und negiert atomaren Formel mit  $\forall, \exists, \wedge, \vee$  aufgebaut ist.

Um eine beliebige Formel  $\varphi$  in die Negationsnormalform umzuwandeln, müssen die booleschen Gesetze angewandt werden und mögliche Negationen auf die unterste Ebene geschoben werden.

---

**Beispiel** Sei  $\varphi := \forall x \exists y (\neg(x > y \rightarrow (x = y)))$ .

Diese Formel wird nun schrittweise in die Negationsnormalform umgewandelt:

$$\begin{aligned}\varphi &= \forall x \exists y (\neg(x > y \rightarrow (x = y))) \\ &= \forall x \exists y (\neg(\neg(x > y) \vee (x = y))) \\ &= \forall x \exists y (x > y \wedge \neg(x = y))\end{aligned}$$

Dies entspricht der Negationsnormalform, da  $\neg(x = y)$  als atomar gilt, da  $(x = y)$  eine Relation darstellt.

---

#### 2.5.4 Gleichheitsfreie Normalform (GNF)

---

Eine Formel  $\varphi$  ist in der *gleichheitsfreien Normalform*, wenn in dieser kein Gleichheitszeichen  $=$  existiert.

Um eine beliebige Formel  $\varphi$  in die gleichheitsfreie Normalform umzuwandeln, müssen alle Gleichheitszeichen durch ein (neues!) Relationssymbol  $\sim$  ersetzt werden, welches die Funktionalität des Gleichheitszeichens übernimmt.

**Beispiel** Sei  $\varphi := \forall x \exists y (x = y)$ .

Diese Formel wird nun in die gleichheitsfreie Normalform umgeformt:

$$\begin{aligned}\varphi &= \forall x \exists y (x = y) && \text{(Substitution)} \\ &= \forall x \exists y (x \sim y) && \text{(Gleichheitsfreie Normalform)}\end{aligned}$$

---

### 2.6 Spielsemantik

---

Die Spielsemantik ist eine Methode, um die Allgemeingültigkeit und auch die Unerfüllbarkeit einer Formel zu beweisen. Hierbei wird eine in Negationsnormalform vorliegende Formel  $\varphi$  von links nach rechts durchlaufen, wobei die in ?? gelisteten Spielregeln angewandt werden.

**Semantik-Spiel**  $[\mathcal{A}; \text{SF}(\varphi)]$ :

Spieler: **Verifizierer** gegen **Falsifizierer**

Spielpositionen:  $(\psi, a) \in \text{SF}(\varphi) \times A^n$

Züge in Position  $(\psi, a), a = (a_1, \dots, a_n)$ :

$\psi = \psi_1 \wedge \psi_2$    **F** am Zug - zieht nach  $(\psi_1, a)$  oder  $(\psi_2, a)$

$\psi = \psi_1 \vee \psi_2$    **V** am Zug - zieht nach  $(\psi_1, a)$  oder  $(\psi_2, a)$

$\psi = \forall x_i \psi_0$    **F** am Zug - zieht nach einem  $(\psi_0, a')$  mit  $a' = (a_1, \dots, a'_i, \dots, a_n)$

$\psi = \exists x_i \psi_0$    **V** am Zug - zieht nach einem  $(\psi_0, a')$  mit  $a' = (a_1, \dots, a'_i, \dots, a_n)$

Spielende: In Positionen  $(\psi, a)$ , wobei  $\psi$  atomar oder negiert atomar ist.

Gewinner:  $\begin{cases} \text{Verifizierer} & \text{falls } \mathcal{A} \models \psi[a] \\ \text{Falsifizierer} & \text{falls } \mathcal{A} \not\models \psi[a] \end{cases}$

Abbildung 2.3: Spielsemantik

### 3 Entscheidbarkeit

---

- $\text{SAT}(\text{AL}) := \{\varphi \in \text{AL} : \varphi \text{ erfüllbar}\}$   
entscheidbar
- $\overline{\text{SAT}(\text{AL})} := \{\varphi \in \text{AL} : \varphi \text{ unerfüllbar}\}$   
entscheidbar
- $\text{VAL}(\text{AL}) := \{\varphi \in \text{AL} : \varphi \text{ allgemeingültig}\}$   
entscheidbar
- $\text{SAT}(\text{FO}) := \{\varphi \in \text{FO} : \varphi \text{ erfüllbar}\}$   
unentscheidbar, rekursiv aufzählbar
- $\overline{\text{SAT}(\text{FO})} := \{\varphi \in \text{FO} : \varphi \text{ unerfüllbar}\}$   
unentscheidbar, rekursiv aufzählbar
- $\text{VAL}(\text{FO}) := \{\varphi \in \text{FO} : \varphi \text{ allgemeingültig}\}$   
unentscheidbar, rekursiv aufwählbar
- $\text{FINSAT}(\text{FO}) := \{\varphi \in \text{FO} : \varphi \text{ hat ein endliches Model}\}$   
unentscheidbar, nicht rekursiv aufwählbar
- $\text{INFSAT}(\text{FO}) := \{\varphi \in \text{FO} : \varphi \text{ hat nur unendliche Modelle}\}$   
unentscheidbar, nicht rekursiv aufzählbar



---

## 4 Abkürzungen und Begriffe

---

**Endlichkeitssatz** Kompaktheitssatz.