

# Kapitel 4 Formale Logik

- 4.1 Aussagenlogik
- 4.2 Prädikatenlogik




---

---

---

---

---

---

---

---

## 4 Ziele

1/53

- Präzise Beschreibung von Sachverhalten so, daß
    - mit mechanisch durchführbaren, **syntaktischen** Schlüssen inhaltliche, **semantische** Eigenschaften festgestellt und Schlußfolgerungen gezogen werden können
    - Konsistenz/Widersprüchlichkeit bzw. Vollständigkeit von Aussagen festgestellt werden kann
    - nachgeprüft werden kann, ob das logische Gedanken-Gebäude ein Modell in der realen Welt besitzt
  - Formale Logik ist
    - das allgemeinste und genaueste Verfahren zur Spezifikation von Aufgaben und den Eigenschaften von Lösungen, insbesondere der Korrektheit von Programmen
    - **Beispiel:** Zustände vor/nach einem Zustandsübergang durch Vor-/Nachbedingungen (prädikatenlogische Formel) charakterisieren
    - formale Logik ist Grundlage des logischen Programmierens, des automatischen Theorem-Beweisens und anderer Zweige der KI
  - digitaler Entwurf von Schaltkreisen ist eine Anwendung der Aussagenlogik
- Mathematische und formale Logik sind Synonyme, philosophische und andere Logiken unterscheiden sich davon*




---

---

---

---

---

---

---

---

## 4 Fußgänger-Ampel

2/53

### Beispiel:

Eine Fußgänger-Ampel darf nur ganz bestimmte Kombinationen an Lichtzeichen zeigen. Die Funktionsweise kann mittels formaler Logik beschrieben werden.

Die Fußgängerampel zeigt genau dann grün ( $F_G$ ), wenn die Verkehrsampel nicht grün ( $\neg A_G$ ) zeigt:

$$F_G = \neg A_G$$

$$F_R = A_G \vee A_E$$



Die Verkehrsampel zeigt genau dann grün, wenn sie weder rot oder gelb zeigt:

$$A_G = \neg A_E \wedge \neg A_R \quad \text{usw.}$$

### Spezifikation mittels formaler Logik




---

---

---

---

---

---

---

---

#### 4.1 Aussagenlogik

3/53

**Beispiel:** Bauernregeln

- Gibt es Eis und Schnee im Januar, so beginnt sehr kalt das Jahr
- Es gibt Eis und Schnee im Januar
- Also beginnt das Jahr sehr kalt.

**Beispiel:**

- Wenn es schneit, dann sind die Straßen glatt.
- Es schneit.
- Also sind die Straßen glatt.



---

---

---

---

---

---

---

---

#### 4.1 Beobachtungen

4/53

- Wir verknüpfen Aussagen:
  - es gibt Eis und Schnee im Januar  $\rightarrow$  das Jahr beginnt sehr kalt
  - es gibt Eis und Schnee im Januar
  - wir haben die Aussage „das Jahr beginnt sehr kalt“ gefolgert.
  - es schneit  $\rightarrow$  die Straßen sind glatt
  - es schneit
  - wir haben die Aussage „die Straßen sind glatt“ gefolgert
- die Art des Schlusses ist dieselbe.
  - aus  $(a \rightarrow b)$  und  $a$  wurde  $b$  gefolgert.
  - insbesondere ist der Schluß unabhängig von konkreten Inhalten.
  - konkrete Inhalte sind **Interpretationen**.
    - ➔ Schlußfolgern heißt Formeln syntaktisch herleiten (**Kalkül**)
    - ➔ trotzdem ist das auch semantisch richtig! (**Korrektheit**)
  - können wir auch **alles** Richtige folgern? (**Vollständigkeit**)



---

---

---

---

---

---

---

---

#### 4.1 Aussagenlogische Formeln

5/53

**aussagenlogische Formel:** Term mit Variablen in der initialen Termalgebra  $\mathcal{T} = (\Sigma, V)$  zur Signatur  $\Sigma = \{ O/0, L/0, \neg/1, \vee/2, \wedge/2 \}$

▪ die Signatur entspricht der booleschen Algebra  $\mathcal{B}$ .

$O \mapsto \perp, L \mapsto \top, \neg \mapsto \neg, \vee \mapsto \vee, \wedge \mapsto \wedge$

- lies:  $O$  als ‚falsch‘  
 $L$  als ‚wahr‘  
 $\neg$  als ‚nicht‘  
 $\wedge$  als ‚und‘  
 $\vee$  als ‚oder‘

- **aber: wir wissen noch nicht, ob auch die Gesetze V1-V10 gelten!**
- **atomare Aussage:**  $O, L, p \in V$ , **Literal:**  $O, L, p, \neg O, \neg L, \neg p$
- syntaktisch korrekt aufgebaute Terme heißen (**syntaktisch**) **korrekt**
  - hat nichts damit zu tun, ob sie semantisch wahr sind

**Beispiele:**

- $\neg(\text{es gibt Eis und Schnee im Januar}) \vee (\text{das Jahr beginnt sehr kalt})$
- $x \wedge (y \vee z)$
- $O \vee x$



---

---

---

---

---

---

---

---

#### 4.1 Abkürzungen und Konventionen

6/53

**Abkürzungen:**  $p \rightarrow q \triangleq \neg p \vee q$   
**impliziert** (auch: **wenn..., dann..., aus ... folgt ...**)

$p \equiv q$  oder  $p \leftrightarrow q \triangleq (p \rightarrow q) \wedge (q \rightarrow p)$   
**äquivalent** (auch: **genau dann..., wenn...**)

$\bigvee_{i=1..n} p_i \triangleq p_1 \vee \dots \vee p_n$  (auch bei abzählbar vielen  $p_i$ )

$\bigwedge_{i=1..n} p_i \triangleq p_1 \wedge \dots \wedge p_n$  (auch bei abzählbar vielen  $p_i$ )

$\bigvee_{i=1..0} p_i \triangleq \text{O}$

$\bigwedge_{i=1..0} p_i \triangleq \text{L}$

▪  $\neg, \vee, \wedge, \rightarrow, \equiv$  heißen **Junktoren**



---

---

---

---

---

---

---

---

---

---

#### 4.1 Klammern

7/53

Konventionen zum Weglassen von Klammern:

$\neg$   
bindet stärker als  $\wedge$   
bindet stärker als  $\vee$   
bindet stärker als  $\rightarrow$   
bindet stärker als  $\leftrightarrow$

Klammerung von  $\vee, \wedge, \rightarrow, \leftrightarrow$  ist linksassoziativ, z.B.

$x \wedge y \wedge z = (x \wedge y) \wedge z$

**Beispiel:**

$p \wedge q \rightarrow r$   
 $= (p \wedge q) \rightarrow r$   
 $= \neg(p \wedge q) \vee r$   
 $= (\neg(p \wedge q)) \vee r$



---

---

---

---

---

---

---

---

---

---

#### 4.1 Interpretation von Formeln (I)

8/53

**Ziel:** Formeln inhaltlich betrachten, prüfen, ob

- Formeln immer wahr sind (**allgemeingültig, Tautologie**)
- Formeln wahr sein können (**erfüllbar**)
- Formeln nie wahr sein können (**unerfüllbar**)

dazu müssen Formeln interpretiert werden.

**Grundidee:**

Interpretation von Formeln durch Abbildung  $J$  in eine boolesche Algebra mit Signatur  $\Sigma = \{f/0, w/0, \neg/1, \vee/2, \wedge/2\}$

- Abbildung durch Umbenennung der Signatur :  
 $O \rightarrow f, L \rightarrow w, \neg \rightarrow \neg, \vee \rightarrow \vee, \wedge \rightarrow \wedge$
- Trägermenge  $B = \{f, w\}$ ,  $f$  ist „falsch“,  $w$  ist „wahr“,  $f \leq w$
- **Belegung:** Variablenabbildung durch Substitution  $\sigma = [w/v]$  bzw.  $\sigma = \frac{f/v}{f/v}$
- immer nur endlich viele Variable abbilden!  
◦ in endlich vielen Formeln kommen nur endlich viele Variable vor



---

---

---

---

---

---

---

---

---

---

#### 4.1 Interpretation von Formeln (II)

9/53

##### Beispiel:

•  $p \wedge q$  ist **erfüllbar**:

die Variablen (Elementaroperanden)  $p$  und  $q$  werden jeweils durch  $w$  interpretiert:  $\sigma = [w/p, w/q]$  anwenden

•  $p \vee \neg p$  ist **allgemeingültig**:

unabhängig von der Interpretation des Elementaroperanden  $p$  ergibt sich  $w$  nach Definition des Komplements

•  $p \wedge \neg p$  ist **unerfüllbar**:

unabhängig von der Interpretation des Elementaroperanden  $p$  ergibt sich  $f$  nach der Definition des Komplements



---

---

---

---

---

---

---

---

#### 4.1 Belegungen und Interpretationen

10/53

**Belegung:** Substitution  $\sigma: V' \rightarrow \{w, f\}$ ,  
endliche Teilmenge  $V' \subseteq V$  wird abgebildet

**Interpretation  $\mathcal{J}_\sigma$ :** Abbildung von Formeln nach  $\{f, w\}$  unter Belegung  $\sigma$ :

$$\mathcal{J}_\sigma(O) = f \qquad \mathcal{J}_\sigma(L) = w$$

$$\mathcal{J}_\sigma(a) = \sigma(a) \text{ für } a \in V \qquad \mathcal{J}_\sigma(\neg F) = \begin{cases} w & \text{wenn } \mathcal{J}_\sigma(F) = f \\ f & \text{wenn } \mathcal{J}_\sigma(F) = w \end{cases}$$

$$\mathcal{J}_\sigma(F \wedge G) = \begin{cases} w & \text{wenn } \mathcal{J}_\sigma(F) = w \text{ und } \mathcal{J}_\sigma(G) = w \\ f & \text{sonst} \end{cases}$$

$$\mathcal{J}_\sigma(F \vee G) = \begin{cases} w & \text{wenn } \mathcal{J}_\sigma(F) = w \text{ oder } \mathcal{J}_\sigma(G) = w \\ f & \text{sonst} \end{cases}$$

**Beachte:** **und** und **oder** werden jetzt inhaltlich (semantisch) interpretiert!

☛ Die Regel für  $\neg F$  interpretiert „ $\neg$ “ als „es gilt **nicht**“



---

---

---

---

---

---

---

---

#### 4.1 Modell

11/53

Eine Interpretation  $\mathcal{J}_\sigma$  heißt **Modell** für eine Formel  $F$ , wenn  $\mathcal{J}_\sigma(F) = w$  gilt.  
 $\mathcal{J}_\sigma$  heißt **Modell** für die Formelmengemenge  $\mathcal{F}$ , wenn  $\mathcal{J}_\sigma$  Modell für alle  $F \in \mathcal{F}$  ist.

**erfüllbare Formel  $F$ :**

es gibt eine Belegung  $\sigma$ , so daß  $\mathcal{J}_\sigma(F) = w$  oder:  
es gibt ein Modell  $\mathcal{J}_\sigma$  für die Formel  $F$ .

**allgemeingültige Formel  $F$ :**

für alle Belegungen  $\sigma$  ist  $\mathcal{J}_\sigma(F) = w$  oder:  
alle Interpretationen  $\mathcal{J}_\sigma$  sind Modelle für  $F$ .

**unerfüllbare Formel  $F$ :**

für alle Belegungen  $\sigma$  ist  $\mathcal{J}_\sigma(F) = f$  oder:  
es gibt kein Modell für  $F$ .

**Eigenschaft:** Eine Formel  $F$  ist genau dann allgemeingültig, wenn  $\neg F$  unerfüllbar ist.



---

---

---

---

---

---

---

---









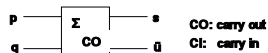


#### 4.1 Einsichten und Anwendungen

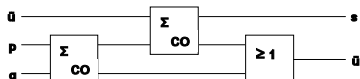
27/53

- Der Sheffersche Strich, die Peirce-Funktion, sowie die Paare  $\neg$ ,  $\wedge$  und  $\neg$ ,  $\vee$  genügen jeweils für sich, um alle Junktoren auszudrücken

- Halbaddierer:**  $p, q$  als Ziffern 0/1, Ergebnis ist Summe und Übertrag (CO),  $p + q = (s, \bar{u}) = (p \text{ xor } q, p \wedge q)$



- Volladdierer:**  $p + q + \bar{u} = (s, \bar{u}) = (p \text{ xor } q \text{ xor } \bar{u}, ((p \text{ xor } q) \wedge \bar{u}) \text{ nor } (p \wedge q))$



---

---

---

---

---

---

---

---

---

---

#### 4.1 Zusammenfassung

28/53

- Aussagenlogik dient der Formalisierung von Aussagen
- Syntax von Formeln basiert auf aussagenlogischen Variablen und den Operatoren  $\neg$  (nicht),  $\wedge$  (und),  $\vee$  (oder); Variable  $\underline{a}$  Aussage
- Die boolesche Algebra ist Grundlage der Interpretation von Formeln.
- Den Wahrheitsgehalt von Formeln kann man mit Wahrheitstafeln ausrechnen.
- Allgemeingültige Formeln sind immer wahr, erfüllbare Formeln können wahr werden, unerfüllbare Formeln sind nie wahr.
- Formeln kann man aus Formelmengen semantisch folgern ( $\mathcal{F} \models F$ ).
- Auch dies kann man syntaktisch mit Wahrheitstafeln nachrechnen.
- Das Nachrechnen mit Wahrheitstafeln ist sehr aufwendig.
- Mit einem korrekten und vollständigen Kalkül kann man syntaktische Folgerungen einfacher nachweisen ( $\mathcal{F} \vdash F$ ).
- DNF und KNF sind aufwendig herzustellen, aber Grundlage vieler Anwendungen
- Die Junktoren spielen in der technischen Informatik eine große Rolle



---

---

---

---

---

---

---

---

---

---

#### 4.2 Prädikatenlogik

29/53

- Problem:
  - in der Aussagenlogik bezeichnen Variable Aussagen
    - keine Möglichkeit Aussagen  $p(X)$  über Objekte  $X$  zu machen
- Prädikatenlogik:
  - Variable bezeichnen Objekte, nur in Logik höherer Stufe auch Aussagen
    - Objekte werden repräsentiert durch Terme einer Termalgebra
  - Aussagen  $p(X)$  über Objekte heißen **Prädikate** (über  $X$ )
  - auch Aussagen  $q(X, Y, \dots)$  über mehrere Objekte möglich: Prädikatssymbole  $p, q, \dots$  haben Stelligkeit  $p/1, q/2, \dots$
  - Prädikate sind **atomare Formeln**
  - atomare Formeln kann man mit Junktoren verknüpfen
    - Gesetze und Regeln wie in der Aussagenlogik
  - zusätzlich: mit **Quantoren  $\forall, \exists$**  kann man Prädikate über alle Objekte formulieren oder die Existenz eines Objekts mit  $p(X)$  postulieren:
    - $\forall X: p(X)$  bzw.  $\exists X: p(X)$



---

---

---

---

---

---

---

---

---

---

## 4.2 Prädikatenlogik: Beispiel

30/53

**Beispiel:** Weisheiten

- Von je zwei Politikern einer Versammlung von Politikern ist wenigstens einer bestechlich.
- Ein Politiker der Versammlung ist unbestechlich.
- Also sind alle Politiker bis auf einen bestechlich.

Aus *Smullyan: Dame oder Tiger, Fischer, 1992*

**Beobachtungen:**

- Diese Weisheiten können nicht in Aussagenlogik formalisiert werden, denn
  - wir wissen nicht, wie viele Politiker an der Versammlung teilnehmen.
  - wir wissen insbesondere nicht, welcher Politiker unbestechlich ist.
  - wir reden von der Eigenschaft, daß Politiker bestechlich sein können
  - wir reden von **allen** Politikern und von der **Existenz** eines unbestechlichen Politikers.



---

---

---

---

---

---

---

---

---

---

## 4.2 Beispiel in Formeln

31/53

**Beispiel:** Weisheiten

- Von je zwei Politikern einer Versammlung von Politikern ist wenigstens einer bestechlich.
- Ein Politiker der Versammlung ist unbestechlich.
- Also sind alle Politiker bis auf den einen bestechlich.

**Weisheiten als Formeln:**

- $\forall p_1, \forall p_2: \text{inVersammlung}(p_1) \wedge \text{inVersammlung}(p_2) \wedge \neg(p_1 = p_2) \rightarrow \text{istBestechlich}(p_1) \vee \text{istBestechlich}(p_2)$
- $\exists p: \text{inVersammlung}(p) \wedge \neg \text{istBestechlich}(p)$
- $\exists p_1: \text{inVersammlung}(p_1) \wedge \neg \text{istBestechlich}(p_1) \rightarrow \forall p_2: \text{inVersammlung}(p_2) \wedge \neg(p_1 = p_2) \rightarrow \text{istBestechlich}(p_2)$

Abkürzungen:  $\forall x \in U: p$  statt  $\forall x: x \in U \rightarrow p$   
 $\forall x, y: p$  statt  $\forall x \forall y: p$  (U implizit bekannt)  
(entsprechend für  $\exists$ , oft ohne Doppelpunkt oder durch Punkt ersetzt)



---

---

---

---

---

---

---

---

---

---

## 4.2 Beispiel: Sortieren

32/53

**Eingabe:** Daten (beliebiger Anzahl) in irgendeiner Reihenfolge

**Ausgabe:** dieselben Daten wie in der Eingabe, aber sortiert

**Probleme:**

- Wie sehen diese Daten aus?
- Was heißt *Reihenfolge*?
- Was heißt *sortiert*?
- Was heißt *dieselben* Daten?



---

---

---

---

---

---

---

---

---

---

#### 4.2 Beispiel Sortieren: Antworten

33/53

##### Wie sehen die Daten aus?

- Daten stammen aus einem Universum  $U$ . ( $x \in U$ )

##### Was heißt Reihenfolge?

- Reihenfolge heißt, daß die Daten in einer Liste (Monoid) angeordnet sind und die Elemente dieser Liste bilden. ( $q \in U^*$ ,  $x \in q$ )

##### Was heißt sortiert?

- sortiert verlangt, daß  $U$  total geordnet ist mit einer Ordnung  $\leq$ :

$$\forall x: x \in U \rightarrow x \leq x$$

$$\forall x \forall y: x \in U \wedge y \in U \wedge x \leq y \wedge y \leq x \rightarrow x = y$$

$$\forall x \forall y \forall z: x \in U \wedge y \in U \wedge z \in U \wedge x \leq y \wedge y \leq z \rightarrow x \leq z$$

$$\forall x \forall y: x \in U \wedge y \in U \rightarrow x \leq y \vee y \leq x$$

- sortiert bedeutet, daß die Elemente in einer Liste vorliegen, so daß alle Elemente  $z$ , die vor einem beliebigen Element  $x$  kommen,  $\leq x$  sind:

$$\text{isSorted}([])$$

$$\forall a: a \in U \rightarrow \text{isSorted}([a])$$

$$\forall a \forall q: a \in U \wedge q \in U^* \wedge \neg(q = []) \wedge (\forall x: x \in q \rightarrow a \leq x) \wedge \text{isSorted}(q)$$

$$\rightarrow \text{isSorted}(\text{append}([a], q))$$



---

---

---

---

---

---

---

---

---

---

#### 4.2 Beispiel Sortieren: Antworten

34/53

##### Was heißt dieselben Daten?

- dieselben Daten heißt, daß alle Elemente der Eingabeliste auch in der Ausgabeliste vorkommen und umgekehrt, auch mit der selben Vielfachheit anz. Dazu Prädikat „ist Permutation“:

$$\forall x \in U: \text{anz}(x, []) = 0$$

$$\forall x \in U: \text{anz}(x, [x]) = 1$$

$$\forall x \in U \forall y \in U: x \neq y \rightarrow \text{anz}(x, [y]) = 0$$

$$\forall x \in U \forall q, q', q'' \in U^*: q = \text{append}(q', q'') \rightarrow \text{anz}(x, q) = \text{anz}(x, q') + \text{anz}(x, q'')$$

$$\forall q \forall q': \text{isPerm}(q, q') \leftrightarrow \forall x \in U: \text{anz}(x, q) = \text{anz}(x, q')$$

**Eingabe:**  $q \in U^*$

**Ausgabe:**  $\text{sort}(q) \in U^*$

**Ausgabebedingungen:**  $\text{isPerm}(q, \text{sort}(q)) \wedge \text{isSorted}(\text{sort}(q))$



---

---

---

---

---

---

---

---

---

---

#### 4.2 Formeln der Prädikatenlogik 1. Stufe (I)

35/53

**Alphabet** ist ein Tripel  $\mathcal{A} = (V, \Sigma, \Pi)$  mit

- $V$  ist Menge von **Variablen**
- $\Sigma$  ist Signatur von **Funktionssymbolen**
- $\Pi = \Pi^{(1)} \cup \Pi^{(2)} \cup \dots$  ist Menge von Prädikats- oder Relationssymbolen
  - $p \in \Pi^{(n)}$  heißt **n-stelliges Prädikatsymbol**

**Terme** entstammen der initialen Grundtermalgebra  $\mathcal{T}(\Sigma, V)$  über der Signatur  $\Sigma$  mit Variablen  $V$ :

- Jede Variable  $x \in V$  ist ein Term.
- Wenn  $f/n \in \Sigma$ , und  $t_1, t_2, \dots, t_n$  Terme sind, dann ist auch  $f(t_1, \dots, t_n)$  ein Term



---

---

---

---

---

---

---

---

---

---





## 4.2 Rechenregeln für Quantoren

42/53

Aus der Definition einer Interpretation folgt, daß die folgenden Umformungen bei Formeln mit Quantoren semantisch korrekt sind:

- eine gebundene Variable  $x$  darf in einer Formel  $Qx: F$  durch eine Variable  $y$  ersetzt werden, **wenn  $y$  in  $F$  nicht frei vorkommt**.  
Neue Form:  $Qy: F[y/x]$  (mit  $Q \in \{\forall, \exists\}$ )
- es gilt  $\neg \exists x: F \equiv \forall x: \neg F$ ,  $\neg \forall x: F \equiv \exists x: \neg F$
- $F_1 \vee Qx: F_2 \equiv Qx: F_1 \vee F_2$ ,  $F_1 \wedge Qx: F_2 \equiv Qx: F_1 \wedge F_2$ ,  
**wenn  $x$  in  $F_1$  nicht frei vorkommt**
- $\forall x \forall y: F \equiv \forall y \forall x: F$
- **Achtung:**  $(\exists y \forall x: F) \rightarrow (\forall x \exists y: F)$  **aber nicht umgekehrt**
  - **Beispiel:** Stetigkeitsdefinition in der Analysis

• also: man darf Variable umbenennen, Negation mit Vertauschung  $\forall \leftrightarrow \exists$  durchziehen, Quantoren vorziehen und Allquantoren untereinander vertauschen

• **man darf nicht immer Existenz- und Allquantoren vertauschen**



---

---

---

---

---

---

---

---

---

---

## 4.2 Normalformen (I)

43/53

Eine Folge von Quantoren  $P = Qx_1 \dots Qx_k$ ,  $Q \in \{\forall, \exists\}$  heißt **Präfix** einer Formel  $F = PF' = Qx_1 \dots Qx_k F'$ .  
 $F$  befindet sich in **pränexer Normalform**, wenn  $F'$  quantorenfrei ist.

Es gilt der Satz:

Jede prädikatenlogische Formel  $F$  kann in eine äquivalente Formel  $G$  in pränexer Normalform überführt werden.

**Beispiel:**

- $F = \forall x \exists y (p(x) \wedge q(x, y)) \vee \neg \forall x r(x)$   
Beseitigung der negierten Quantoren und Bereinigung von  $F$ :
- $F' = \forall x \exists y (p(x) \wedge q(x, y)) \vee \exists z \neg r(z)$   
Quantoren vorziehen:
- $F'' = \forall x \exists y \exists z (p(x) \wedge q(x, y)) \vee \neg r(z)$   
 $F''$  ist in pränexer Normalform.



---

---

---

---

---

---

---

---

---

---

## 4.2 Normalformen (II)

44/53

Eine pränex Normalform kann leider nicht so standardisiert werden, daß zu Beginn die Allquantoren und dahinter die Existenzquantoren kommen. Es gibt aber eine semantisch äquivalente Normalform, die nur noch die Allquantoren aufweist.

**Satz:** Zu jeder prädikatenlogischen Formel  $F$  gibt es eine Formel  $G$  mit

- $G = \forall x_1 \dots \forall x_k G'$ ,  $k \geq 0$
- $G'$  quantorenfrei
- $F$  ist genau dann erfüllbar, wenn  $G$  erfüllbar ist.

•  $G$  heißt **Skolemform** von  $F$ .  
Die Skolemform erreicht man, indem man die durch Existenzquantoren gebundenen Variablen nacheinander durch **Skolemfunktionen** ersetzt, die von den Variablen der vorausgehenden Allquantoren abhängt.

**Beispiel:**  $\forall u \exists v \forall x \exists y: p(u, v, x, y)$   
wird ersetzt durch die Skolemform  $\forall u \forall x: p(u, sk_v(u), x, sk_y(u, x))$



---

---

---

---

---

---

---

---

---

---



## 4.2 Prolog: Schreibweise

48/53

- Konventionen:
  - Variablenbezeichner beginnen mit Großbuchstaben, alle anderen Bezeichner mit Kleinbuchstaben
  - `_` ist ein Variablenbezeichner wechselnder Identität
- Hornklauseln  $p_1 \wedge p_2 \wedge \dots \wedge p_m \rightarrow q$ 
  - Implikation:  $q :- p_1, \dots, p_m$  · Komma statt  $\wedge$ , abschließender Punkt
  - Faktum:  $q$  · abschließender Punkt
  - Anfrage:  $?- G$  · abschließender Punkt



---

---

---

---

---

---

---

---

## 4.2 Beispiel: formales Differenzieren

49/53

Prolog-Programm:

```
diff(X,X,1).
diff(C,X,0)      :- atomic(C), not(C=X).
diff(U+V,X,DU+DV) :- diff(U,X,DU), diff(V,X,DV).
diff(U-V,X,DU-DV) :- diff(U,X,DU), diff(V,X,DV).
diff(U*V,X,U*DV+V*DU) :- diff(U,X,DU), diff(V,X,DV).
diff(U/V,X,(V*DU-U*DV)/(V*V)) :- diff(U,X,DU), diff(V,X,DV).
```

Anfragen:

```
?- diff(x*x,x,E).
E = x*1 + 1*x
?- diff(a*x+b,x,E).
E = a*1+x*0+0
?- diff(x^2,x,E)
no
```



---

---

---

---

---

---

---

---

## 4.2 Beispiel: formales Differenzieren

50/53

```
diff(X,X,1).
diff(C,X,0)      :- atomic(C), not(C=X).
diff(C+U,X,D)    :- atomic(C), not(C=X), diff(U,X,D).
diff(U+C,X,D)    :- atomic(C), not(C=X), diff(U,X,D).
diff(U+V,X,DU+DV) :- diff(U,X,DU), diff(V,X,DV).
diff(U-V,X,DU-DV) :- diff(U,X,DU), diff(V,X,DV).
diff(C*X,X,C)    :- atomic(C), not(C=X).
diff(C*U,X,C*D)  :- atomic(C), not(C=X), diff(U,X,D).
diff(U*C,X,C*D)  :- atomic(C), not(C=X), diff(U,X,D).
diff(U*V,X,U*DV+V*DU) :- diff(U,X,DU), diff(V,X,DV).
diff(U/V,X,(V*DU-U*DV)/(V*V)) :- diff(U,X,DU), diff(V,X,DV).
```

```
?- diff(x*x,x,E).
E = x*1 + x*1
?- diff(a*x+b,x,E).
E = a
```



---

---

---

---

---

---

---

---

