# Künstliche Intelligenz

Prof. Dr. Dirk Krechel
Hochschule RheinMain

Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim Geisenheim

# Inhalte

- Einführung
- **Symbolische Verfahren, Logik**
  - Aussagenlogik, **Prädikatenlogik**
  - Horn Logik, Prolog
- Suchen und Bewerten
  - Problemlösen durch Suche
    - Uninformierte Suche
    - Heuristische Suche
    - Spielbäume
  - Information Retrieval
- Lernen
  - Entscheidungstheorie
  - Naive Bayes
  - Entscheidungsbäume
  - Neuronale Netze
  - unüberwachtes Lernen

# Prädikatenlogik – Syntax

- Prädikatensymbole
  - p, q, r, s, P, Q, R, S, $P_1$, $Q_1$, $R_1$, $S_1$, Vater, Katze …
  - Jedes Prädikatensymbole hat Stelligkeit (Anzahl Argumente)
  - Stelligkeit 0 sind Konstanten true and false
  - Stelligkeit 1 sind die Propositionen der Aussagenlogik
- Funktionssymbole
  - f, g, h, $f_1$, $g_1$, $h_1$, add, succ, 0, 1, 2, 3, …
  - Jedes Funktionssymbol hat Stelligkeit
  - Funktionssymbole mit Stelligkeit 0 heißen Konstanten
- Variablen
  - x, y, z, $x_1$, $y_1$, $z_1$, …
- Quantoren
  - $\exists$ (Existenzquantor), $\forall$ (Allquantor)
- Logische Verknüpfungen
  - $\Leftrightarrow$, $\Rightarrow$, $\vee$, $\wedge$, $\neg$, ()

# Definition: Formel

- Definition: *Formel*
  - Wenn P n-stelliges Prädikatssymbol und $t_1, t_2, \ldots, t_n$ Terme sind, dann ist $P(t_1, t_2, \ldots, t_n)$ eine *atomare Formel*. Eine atomare Formel ist eine Formel
  - Wenn $\alpha$ und $\beta$ Formeln sind, dann sind $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \Leftrightarrow \beta)$, $(\alpha \Rightarrow \beta)$, $(\neg \alpha)$ ebenfalls Formeln
  - Wenn x eine Variable ist und $\alpha$ eine Formel ist, dann sind $\forall x{:}\ \alpha$ und $\exists x{:}\ \alpha$ ebenfalls Formeln
- Formeln werden nur durch die Regeln 1. und 2. gebildet
- Formeln ohne Variablen heißen auch *Grundformeln*
- Atomare oder negierte atomare Formeln heißen auch *Literale*
  - Atomare Formeln sind positive Literale
  - Negierte atomare Formeln sind negative Literale
- Einführung von Bindungsregeln, Vermeidung von Klammern
  - Bindungsstärke (aufsteigend): $\forall$, $\exists$, $\Leftrightarrow$, $\Rightarrow$, $\wedge$, $\vee$, $\neg$
  - Gleicher Operator: Annahme Bindung von links nach rechts

# Definition: Semantische Folgerung

- Eine Formel $\beta$ *folgt semantisch* aus einer Formel $\alpha$ gdw für jedes Modell M und jede Interpretation $\mathrm{I}$ gilt, dass wenn $\mathrm{I}(\alpha)=1$ dann $\mathrm{I}(\beta) = 1$.
  Wir schreiben: $\alpha \models \beta$
- Es gilt: $\alpha \models \beta$ gdw $\alpha \Rightarrow \beta$ ist allgemeingültig

- Eine Formel $\beta$ *folgt semantisch* aus einer Menge von Formeln $\Sigma = \{ \alpha_1 ,..., \alpha_n \}$ gdw
  für jedes Modell M und jede Interpretation $\mathrm{I}$ gilt, dass wenn $\mathrm{I}(\alpha_1)=1$ und ... und $\mathrm{I}(\alpha_n)=1$ dann ist auch $\mathrm{I}(\beta) = 1$.
  Wir schreiben: $\Sigma \models \beta$
- Es gilt: $\Sigma \models \beta$ gdw $\alpha_1 \wedge \alpha_2 \wedge ... \wedge \alpha_n \Rightarrow \beta$ ist allgemeingültig

# Widerspruchsvoll

- Eine Menge von prädikatenlogischen Formeln $\Sigma = \{\alpha_1 ,..., \alpha_n\}$ ist *erfüllbar* gdw
  es existiert ein Modell und eine Interpretation I unter der alle Formeln $\alpha_i$ wahr sind, das heißt $I(\alpha_i)=1$ für $i=1...n$

- Eine Menge von prädikatenlogischen Formeln $\Sigma = \{\alpha_1 ,..., \alpha_n\}$ ist *allgemeingültig* (ist eine *Tautologie*) gdw
  jede Formel $\alpha_i$ allgemeingültig ist
- Wenn $\Sigma = \emptyset$, dann ist $\Sigma$ allgemeingültig

- Eine Menge von prädikatenlogischen Formeln $\Sigma = \{\alpha_1 ,..., \alpha_n\}$ ist *widerspruchsvoll* (*inkonsistent*) gdw
  es gibt kein Modell und keine Interpretation I
  mit $I(\alpha_i)=1$ für $i=1...n$

# Äquivalenzen

- Definition: Zwei prädikatenlogische Formeln $\alpha$ und $\beta$ sind äquivalent gdw $\alpha \Leftrightarrow \beta$ ist allgemeingültig
  Wir schreiben $\alpha \approx \beta$
- Alle Äquivalenzen der Aussagenlogik sind gültig
- Weitere wichtige Äquivalenzen:
  - Quantorenaustausch: $\neg (\forall x\ \alpha) \approx (\exists x\ \neg\alpha)$ $\neg (\exists x\ \alpha) \approx (\forall x\ \neg\alpha)$
  - Quantorenvertauschung: $\forall x\ \forall y\ \alpha \approx \forall y\ \forall x\ \alpha$ $\exists x\ \exists y\ \alpha \approx \exists y\ \exists x\ \alpha$
  - Quantorenelimination: Wenn x nicht frei in $\alpha$ vorkommt
    $\forall x\ \alpha \approx \alpha$ $\exists x\ \alpha \approx \alpha$
  - Quantoreneinführung: Wenn x nicht frei in $\beta$ vorkommt
    $(\forall x\ \alpha) \wedge \beta \approx (\forall x\ \alpha \wedge \beta)$ $(\exists x\ \alpha) \wedge \beta \approx (\exists x\ \alpha \wedge \beta)$
    $(\forall x\ \alpha) \vee \beta \approx (\forall x\ \alpha \vee \beta)$ $(\exists x\ \alpha) \vee \beta \approx (\exists x\ \alpha \vee \beta)$
  - Variablenumbenennung: Wenn x nicht in $\alpha$ vorkommt
    $\exists x\ \alpha \approx \exists y\ \alpha[x/y]$

# Pränexe Normalform

- Eine quantorenfreie prädikatenlogische Formel ist in
  - *konjunktiver Normalform,* wenn sie als eine Konjunktion von Disjunktionen von Literalen geschrieben ist
  - *disjunktiver Normalform*, wenn sie als eine Disjunktionen von Konjunktionen von Literalen geschrieben ist
- Eine prädikatenlogische Formel ist in *pränexer Normalform* (PNF)
  - wenn sie von der Form $Q_1 x_1\ Q_2 x_2 \ldots Q_n x_n\ \alpha$ ist und
  - die $Q_i$ Quantoren sind ($Q_i \in \{\forall, \exists\}$) und die $x_i$ Variablen sind ($Q_1 x_1\ Q_2 x_2 \ldots Q_n x_n$ nennt man Präfix) und
  - $\alpha$ (die Matrix) eine quantorfreie prädikatenlogische Formel ist
- Satz: Jede Formel ist logisch äquivalent zu einer Formel in pränexer Normalform mit einer Matrix in konjunktiver (beziehungsweise disjunktiver) Normalform

# Beispiel

- $\forall x \ \forall y\ P(x, y)$ ist in PNF
- $(\forall x\ P(x)) \Rightarrow (\exists x\ P(x))$ ist nicht in PNF
- $\exists x\ \exists y\ (\neg P(x) \vee P(y))$ ist in PNF
  ist äquivalent zu zweiter Formel
- true ist in PNF
  ist äquivalent zu zweiter Formel

# Transformation in PNF

- Transformation jeder prädikatenlogischen Formel in eine äquivalente Formel in PNF

1. Umbenennung der Variablen
   - Alle gebundenen Variablen haben voneinander verschiedene Namen
   - Keine gebundene Variable kommt auch frei vor
2. Bringe alle Quantoren auf die linke Seite
   Wende folgenden Äquivalenzen solange möglich von links nach rechts an
   - $(\forall x\ \alpha) \wedge \beta \approx \forall x\ (\alpha \wedge \beta)$ $(\exists x\ \alpha) \wedge \beta \approx \exists x\ (\alpha \wedge \beta)$
   - $(\forall x\ \alpha) \vee \beta \approx \forall x\ (\alpha \vee \beta)$ $(\exists x\ \alpha) \vee \beta \approx \exists x\ (\alpha \vee \beta)$
   - $\neg\ (\forall x\ \alpha) \approx \exists x\ (\neg\alpha)$ $\neg\ (\exists x\ \alpha) \approx \forall x\ (\neg\alpha)$

     Wegen Kommutativität für $\wedge$ und $\vee$ kann man Formel mit Quantor nach vorne bringen und Regel anwenden
   - Formel liegt in PNF vor
3. Transformiere Matrix der PNF-Formel gegebenenfalls in CNF (oder DNF), wie in der Aussagenlogik

# Transformation in PNF – Beispiel

- Formel: $(\forall x\, P(x, y)) \vee P(a,z) \vee (\neg\ \forall x\, Q(x, z))$
- Variablenumbenennung
  $(\forall x\, P(x, y)) \vee P(a,z) \vee (\neg\ \forall y\, Q(y, z))$
- Quantoren nach links schieben
  $\forall x\, (P(x, y) \vee P(a,z)) \vee (\neg\ \forall u\, Q(u, z))$
  $\forall x\, (P(x, y) \vee P(a,z)) \vee (\exists u\, \neg Q(u, z))$
  $\forall x\, (P(x, y) \vee P(a,z)\ \vee (\exists u\, \neg Q(u, z)))$
  $\forall x\, \exists u\, (P(x, y) \vee P(a,z) \vee \neg Q(u, z))$
  $\forall x\, \exists u\, P(x, y) \vee P(a,z) \vee \neg Q(u, z)$

- Wir haben PNF mit Matrix in DNF und CNF

# Skolem Normalform

- Eine prädikatenlogische Formel ist in *Skolem Normalform*, wenn
  - sie in Pränex Normalform ist, also in der Form $Q_1x_1\ Q_2x_2 \ldots Q_nx_n\ \alpha$ und
  - die $Q_i$ alle Allquantoren sind ($Q_i = \forall$)
- *Skolemisierung*: Wir können für jede Formel in PNF die Skolem Normalform erzeugen
  - Entferne den am weitesten links stehenden Existenzquantor $\exists x$
  - Ersetze in der Matrix die Variable x durch den Term $f(y_1, \ldots, y_n)$ wobei
    f ein neues Funktionssymbol ist und
    $y_1, \ldots, y_n$ alle Variablen der Allquantoren links des entfernten Existenzquantors sind
- Das Ergebnis ist in Skolem Normalform
  Die neuen Funktionen heißen *Skolemfunktionen*
  - Idee: Der Zeuge der Existenz des Objekts in einem Universum kann nur davon abhängen welche Objekte den Variablen in den Allquantoren links des entsprechenden Existenzquantors vorher zugewiesen wurde. Die Soklemfunktion „berechnet“ eines dieser Objekte

# Skolem Normalform – Beispiele

- Formel $\forall x\ \exists u\ P(x, y) \vee P(a, z) \vee \neg Q(u, z)$
  Skolemisierung mit $u = f(x)$
  $\forall x\ P(x, y) \vee P(a, z) \vee \neg Q(f(x), z)$

- Formel $\forall x\ \forall y\ \exists z\ P(x, f(y)) \vee P(a, z)$
  Skolemisierung mit $z = g(x, y)$
  $\forall x\ \forall y\ \exists z\ P(x, f(y)) \vee P(a, g(x, y))$

- Formel $\forall x\ \forall y\ \exists z\ \forall u\ \exists v\ P(x, z) \wedge P(y, v)$
  Skolemisierung mit $z = f(x, y)$, $v = g(x, y, u)$
  $\forall x\ \forall y\ \forall u\ P(x, f(x,y)) \wedge P(y, g(x, y, u))$

# Entscheidungsverfahren

- Satz:
  - Eine Formel ist inkonsistent gdw die Skolem Normalform ist inkonsistent
  - Eine Formel ist erfüllbar gdw ihre Skolem Normalform ist erfüllbar
- Praktische Aufgabenstellung:
  Zeige, dass
  $\Sigma = \{\alpha_1 ,..., \alpha_n\} \models \beta$
- Umsetzung (wie in der Aussagenlogik):
  Zeige, dass
  $\Sigma = \{\alpha_1 ,..., \alpha_n\} \cup \{\neg\beta\}$ inkonsistent
  was äquivalent ist
- Es gilt, dass $\{\alpha_1 ,..., \alpha_n\} \cup \{\neg\beta\}$ inkonsistent gdw
  $\alpha_1 \wedge ... \wedge \alpha_n \wedge \neg\beta$ inkonsistent gdw
  Soklem Normalform von $\alpha_1 \wedge ... \wedge \alpha_n \wedge \neg\beta$ inkonsistent

# Klauselnormalform

- Eine Klausel ist eine endliche Menge von Literalen $\{L_1, \ldots, L_n\}$, die als Disjunktion dieser Literale interpretiert wird
  - Mehrfachvorkommen desselben Literals in der Disjunktion sind ausgeschlossen
- Die leere Klausel wird immer auf false abgebildet und mit $\square$ bezeichnet
  - Die leere Klausel ist der Widerspruch
- Eine Menge von Klauseln wird interpretiert als Konjunktion dieser Klauseln
  - Mehrfachvorkommen derselben Klausel in der Konjunktion sind ausgeschlossen
- Jede Formel in Skolem Normalform mit einer Matrix in CNF kann direkt auf eine endliche Menge von Klauseln abgebildet werden
  - Wir nehmen dabei immer an, dass keine freien Variablen vorkommen, bzw. alle (freien) Variablen sind allquantifiziert
  - Die endliche Menge von Klauseln ist die *Klauselnormalform*

# Klauselnormalform – Beispiele

- Formel $\forall x\, \exists u\, P(x, y) \vee P(a, z) \vee \neg Q(u, z)$
  Skolemisierung mit $u = f(x)$
  $\forall x\, P(x, y) \vee P(a, z) \vee \neg Q(f(x), z)$
  Klauselnormalform
  $\{ \{P(x, y), P(a, z), \neg Q(f(x), z)\} \}$
- Formel $\forall x\, \forall y\, \exists z\, P(x, f(y)) \vee P(a, z)$
  Skolemisierung mit $z = g(x, y)$
  $\forall x\, \forall y\, P(x, f(y)) \vee P(a, g(x, y))$
  Klauselnormalform
  $\{ \{P(x, f(y)), P(a, g(x, y))\} \}$
- Formel $\forall x\, \forall y\, \exists z\, \forall u\, \exists v\, P(x, z) \wedge P(y, v)$
  Skolemisierung mit $z = f(x, y)$, $v = g(x, y, u)$
  $\forall x\, \forall y\, \forall u\, P(x, f(x,y)) \wedge P(y, g(x, y, u))$
  Klauselnormalform
  $\{ \{P(x, f(x,y))\}, \{P(y, g(x, y, u))\} \}$

## Beispiel

- Sei $\Sigma = \{ \forall x\, (\neg P(x) \vee \exists y\, \neg Q(y)), P(a) \}$
  Gilt $\Sigma \models \exists x\, \neg Q(x)$ ?
- Gilt gdw
  $\forall x\, (\neg P(x) \vee \exists y\, \neg Q(y)) \wedge P(a) \wedge \neg \exists x\, \neg Q(x)$ inkonsistent
- Transformation in PNF
  $\forall x\, (\neg P(x) \vee \exists y\, \neg Q(y)) \wedge P(a) \wedge \forall x\, Q(x)$
  $\forall x\, (\neg P(x) \vee \exists y\, \neg Q(y)) \wedge P(a) \wedge \forall z\, Q(z)$
  $\forall x \exists y\, (\neg P(x) \vee \neg Q(y)) \wedge P(a) \wedge \forall z\, Q(z)$
  $\forall x \exists y \forall z\, ((\neg P(x) \vee \neg Q(y)) \wedge P(a) \wedge Q(z))$
- Skolemisierung
  $\forall x \forall z\, ((\neg P(x) \vee \neg Q(f(x))) \wedge P(a) \wedge Q(z))$
- Klauselnormalform: $\{ \{\neg P(x), \neg Q(f(x))\}, \{P(a)\}, \{Q(z)\} \}$
- Klauselnormalform ist inkonsistent, also gilt $\Sigma \models \exists x\, \neg Q(x)$
  - Q(z) gilt immer, also darf P(x) für kein x gelten, aber P(a) gilt

# Inferenzkalkül

- Ziel:
  - Prädikatenlogische Formeln direkt syntaktisch manipulieren
  - Erzeugen von *korrekten* prädikatenlogische Formeln
- *Inferenzkalkül*
  - Vorschriften oder *Inferenzregeln*
  - Aus gegebenen prädikatenlogischen Formeln *neue* prädikatenlogische Formen generieren
- Definition (wie Aussagenlogik): Eine Formel $\beta$ *folgt syntaktisch* aus einer Formelmenge $\Sigma=\{\alpha_1,\ldots,\ \alpha_n\}$ und Inferenzregeln IR gdw
  - Es gibt eine Folge von $\Sigma=\Sigma_0, \Sigma_1, \Sigma_2, \ldots$ mit $\beta$ aus einem $\Sigma_i$
  - $\Sigma_{i+1} = \Sigma_i \cup \{\gamma_i\}$; und $\gamma_i$ und entsteht aus Anwendung einer Regel in IR auf $\Sigma_i$
  - Wir schreiben: $\Sigma_i \vdash \gamma_i$, $\Sigma \vdash^* \beta$ oder kurz $\Sigma \vdash \beta$ und $\Sigma_i \vdash \Sigma_j$ für $i \leq j$; um explizit auf IR hinzuweisen schreibt man auch $\vdash_{IR}$ statt $\vdash$

# Korrektheit und Vollständigkeit

- Ziel
  - Ein Kalkül soll *vollständig* sein:
    Alles was (semantisch) korrekt ist soll (syntaktisch) herleitbar sein
  - Ein Kalkül soll *korrekt* sein:
    Alles was (syntaktisch) hergleitet werden kann soll (semantisch) korrekt sein
- Korrektheit und Vollständigkeit: $\vdash = \models$
  - Korrektheit: Für alle $\Sigma$, $\beta$ gilt: Falls $\Sigma \vdash \beta$ gilt, dann gilt $\Sigma \models \beta$
  - Vollständigkeit: Für alle $\Sigma$, $\beta$ gilt: Falls $\Sigma \models \beta$ gilt, dann gilt $\Sigma \vdash \beta$
- Satz von Gödel: Es gibt einen korrekten und vollständigen Kalkül für die Prädikatenlogik erster Stufe (PL1)
- Aber: PL1 ist unentscheidbar
  - Es gibt kein Verfahren, das für ein gegebene Formel entscheidet ob die Formel eine Tautologie ist oder nicht
  - Für ein vollständiges korrektes Kalkül gibt es keine maximale Länge der Ableitungsfolge

# Inferenzregeln

- Notation wie in der Aussagenlogik
  - Prämisse: Ein Muster, dem bestehende Formeln aus E entsprechen
  - Konklusion: Die neue hergeleitete Formel

$$\frac{\text{Prämisse}}{\text{Konklusion}}$$

- Beispiele
  - Modus Ponens

$$\frac{\alpha, \alpha \Rightarrow \beta}{\beta}$$

  - Instanziierung

$$\frac{\forall x \ \alpha}{\alpha \ [x/t]}$$

  - Existenzielle Aussage

$$\frac{\alpha \ [x/t]}{\exists x \ \alpha}$$

- Definition: Beweise (unter IR)
  - Ein *Beweis* von $\alpha$ aus $\Sigma$ ist eine Folge von Formeln $\alpha_i$ (i=1, …,n), so dass
  - $\alpha_i$ in $\Sigma$ oder $\alpha_i$ ist Konklusion aus Anwendung einer Inferenzregel mit $\{ \alpha_j \mid 0 < j < i \} \cup \Sigma$ als Quelle für die Prämisse
  - $\alpha_n = \alpha$

# Beweis – Beispiel

- Sei $\Sigma = \{ \exists x\ P(x) \Rightarrow \forall x\ Q(x), P(a) \}$
  Gilt $\Sigma \vdash Q(a)$?
- Beweis unter Verwendung der Inferenzregeln
  - $\alpha_1 = \exists x\ P(x) \Rightarrow \forall x\ Q(x)$ (aus $\Sigma$)
  - $\alpha_2 = P(a)$ (aus $\Sigma$)
  - $\alpha_3 = \exists x\ P(x)$ (Existenzielle Aussage mit $\alpha_1$ und $\alpha_2$)
  - $\alpha_4 = \forall x\ Q(x)$ (Modus Pones mit $\alpha_1$ und $\alpha_3$)
  - $\alpha_5 = Q(a)$ (Instantziierung mit $\alpha_4$)

# Resolution für die Prädikatenlogik

- Satz: Die Resolution ist korrekt und vollständig in PL1
- Resolutionsbeweis für $\Sigma \models \beta$
  Wir zeigen Inkonsistenz von $\Sigma \cup \{\neg\beta\}$

- Konvertiere $\Sigma \cup \{\neg\beta\}$ in Klauselnormalform
- Verwende die folgenden beiden Inferenzregeln
  solange wie möglich
  - Resolutionsregel für die Prädikatenlogik
  - Faktorisierungsregel für die Prädikatenlogik
- Wenn die leere Klausel abgeleitet werden kann, dann ist die Klauselmenge inkonsistent und folglich

# Resolutionsregel für PL1

- Resolutions-regel $$\frac{\{L_1, L_2, \ldots, L_n, A\} \qquad \{\neg A', K_1, K_2, \ldots, K_m\}}{\{\sigma(L_1), \sigma(L_2), \ldots, \sigma(L_n), \sigma(K_1), \sigma(K_2), \ldots, \sigma(K_m)\}}$$
- Annahme: Die Klauseln $\{L_1, \ldots, L_n, A\}$ und $\{\neg A', K_1, \ldots, K_m\}$ haben disjunkte Variablenmengen, gegebenenfalls können Variablen vorher umbenannt werden
- Die $L_i$ und die $K_j$ sind beliebige (positive oder negative) Literale
- A und A' sind positive Literale
- A und A' sind unifizierbar und mgu(A, A') = $\sigma$

- Die Konklusion heißt auch *Resolvente*
- Resolutionsregel nur anwendbar wenn A und A' unifzierbar
- Ähnlich zur Aussagenlogik, plus Unifikation und Substitution

# Faktorisierungsregel für PL1

- Faktorisierungs-regel

$$\frac{\{L_1, L_2, ..., L_n\}}{\{\sigma(L_1), \sigma(L_2), ..., \sigma(L_n)\}}$$

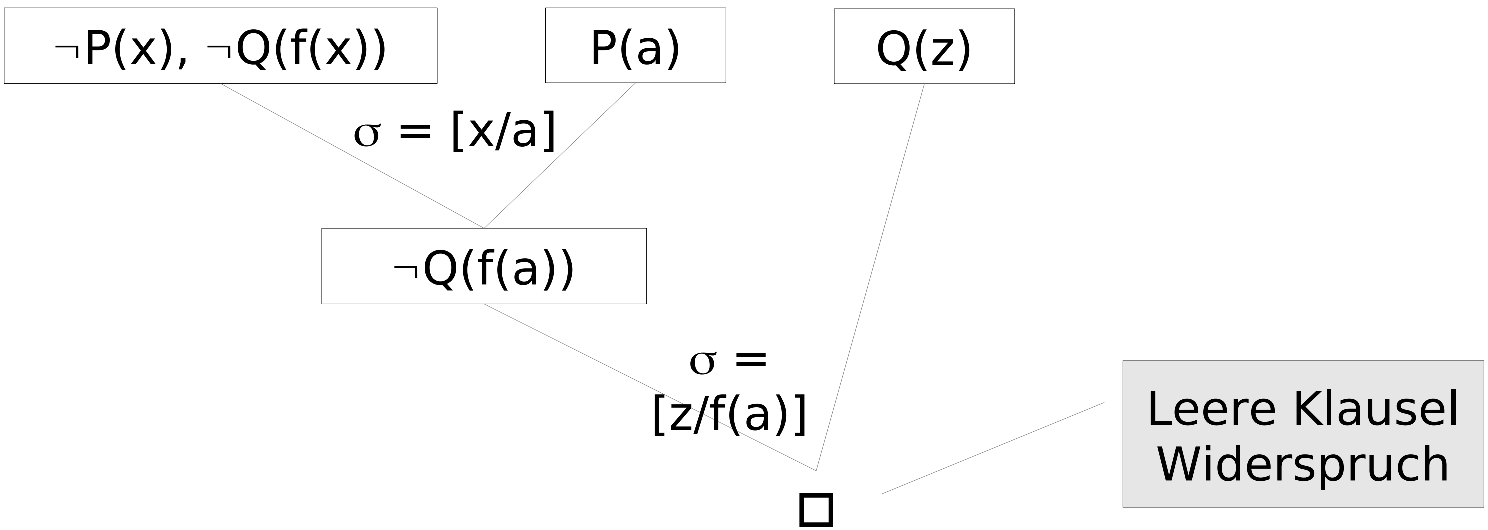
- Die $L_i$ und sind beliebige (positive oder negative) Literale
- Es gibt zwei unifizierbare Literale $L_i$ und $L_j$ mit $i \neq j$
- $\mathrm{mgu}(L_i, L_j) = \sigma$

- Durch Anwendung der Faktorisierungsregel wird die resultierende Klausel kürzer als die Ausgangsklausel

- Keine Entsprechung in der Aussagenlogik

# Resolution – Beispiel

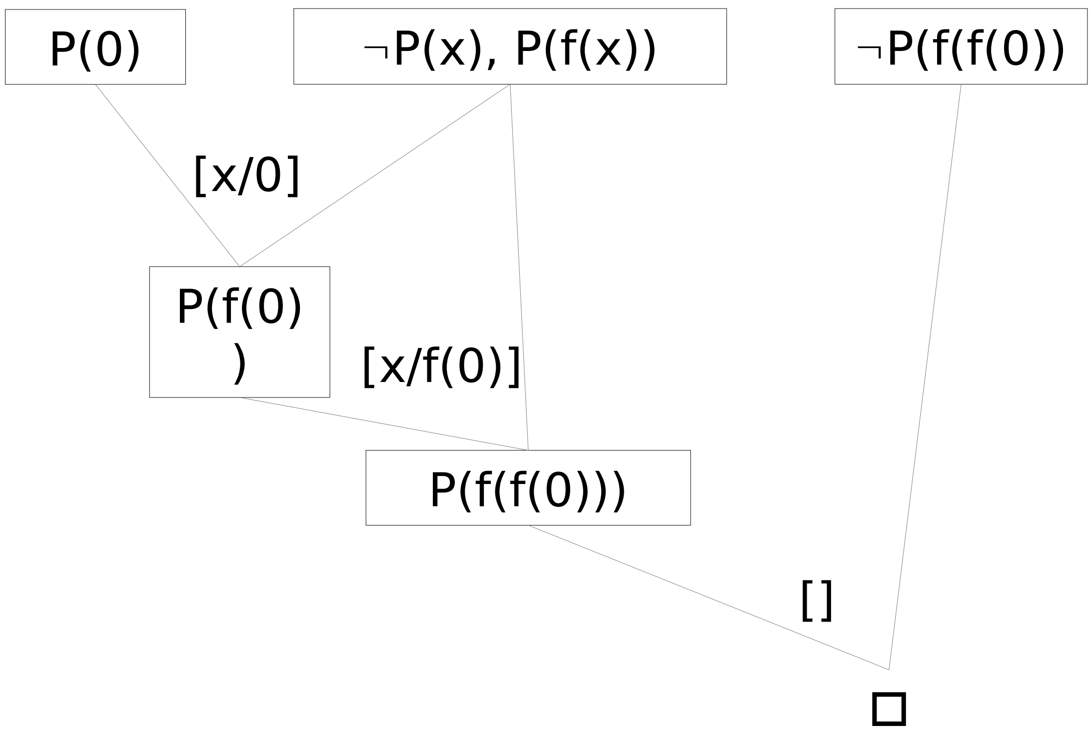
- Sei $\Sigma = \{\ \forall x\ (\neg P(x) \vee \exists y\ \neg Q(y)),\ P(a)\ \}$
  Gilt $\Sigma \vdash \exists x\ \neg\ Q(x)$ ?
- Klauselnormalform wie vorher
  $\{\ \{\neg P(x), \neg Q(f(x))\}, \{P(a)\}, \{Q(z)\}\ \}$
- Resolutionsbeweis




- [Im Beispiel nur Resolutionsregel ausreichend ]

# Resolution – Beispiel

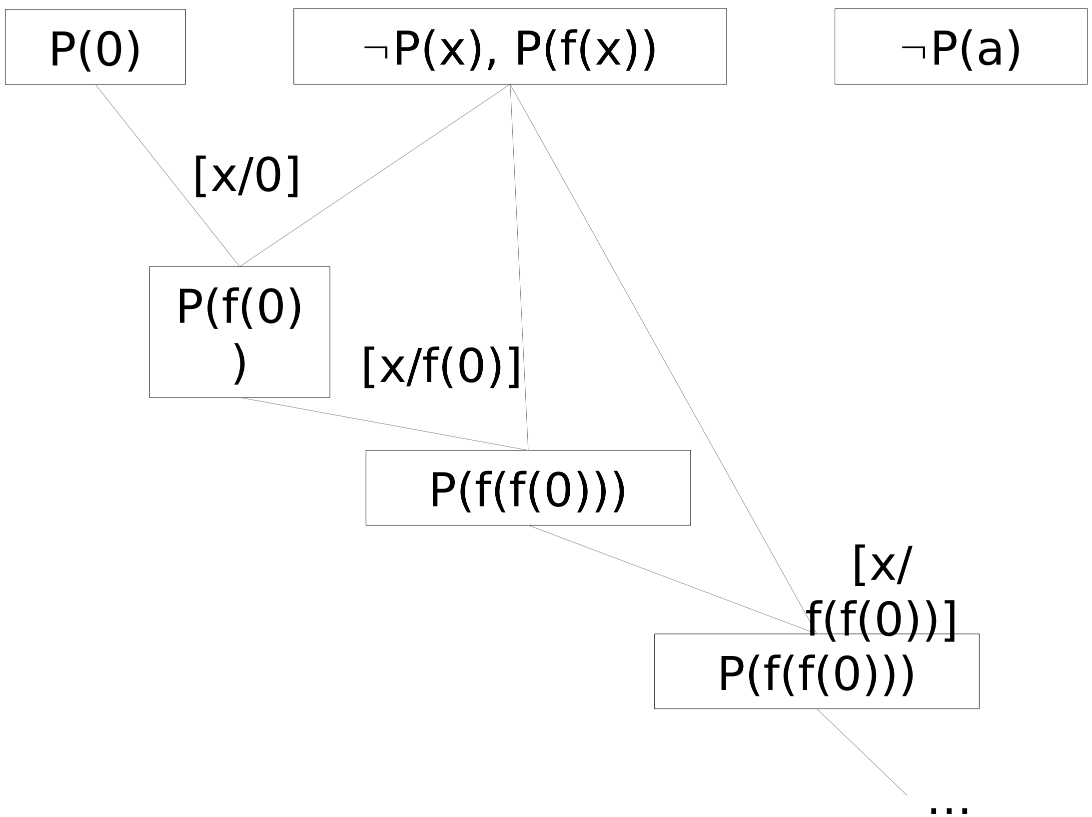
- { P(0), ∀x (P(x) ⇒ P(f(x))) } ├ P(f(f(0))




- [In diesem Beispiel auch nur Resolutionsregel, aber eine Klausel  wurde mehrfach angewendet]

# Resolution – Beispiel

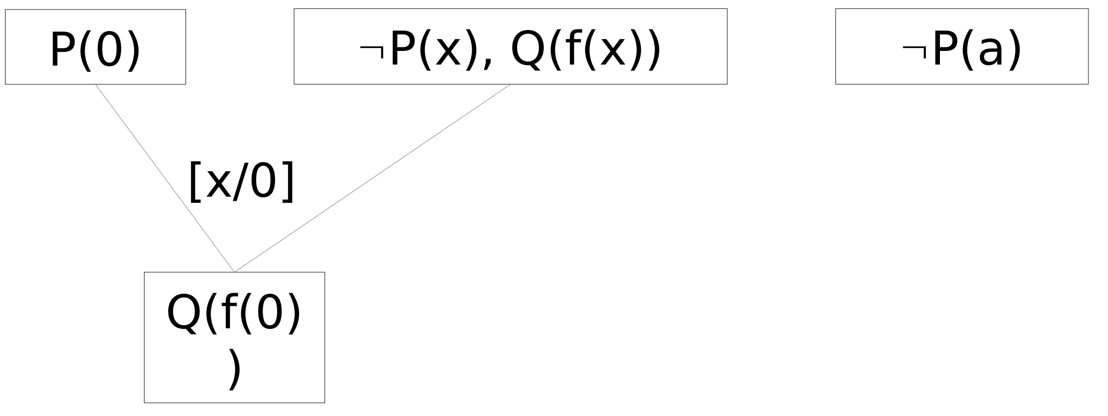
- $\{ P(0), \forall x\, (P(x) \Rightarrow P(f(x))) \} \vdash P(a)$




- [Unendliche Ableitung. P(a) folgt nicht, aber Resolution findet das nicht heraus]

# Resolution – Beispiel

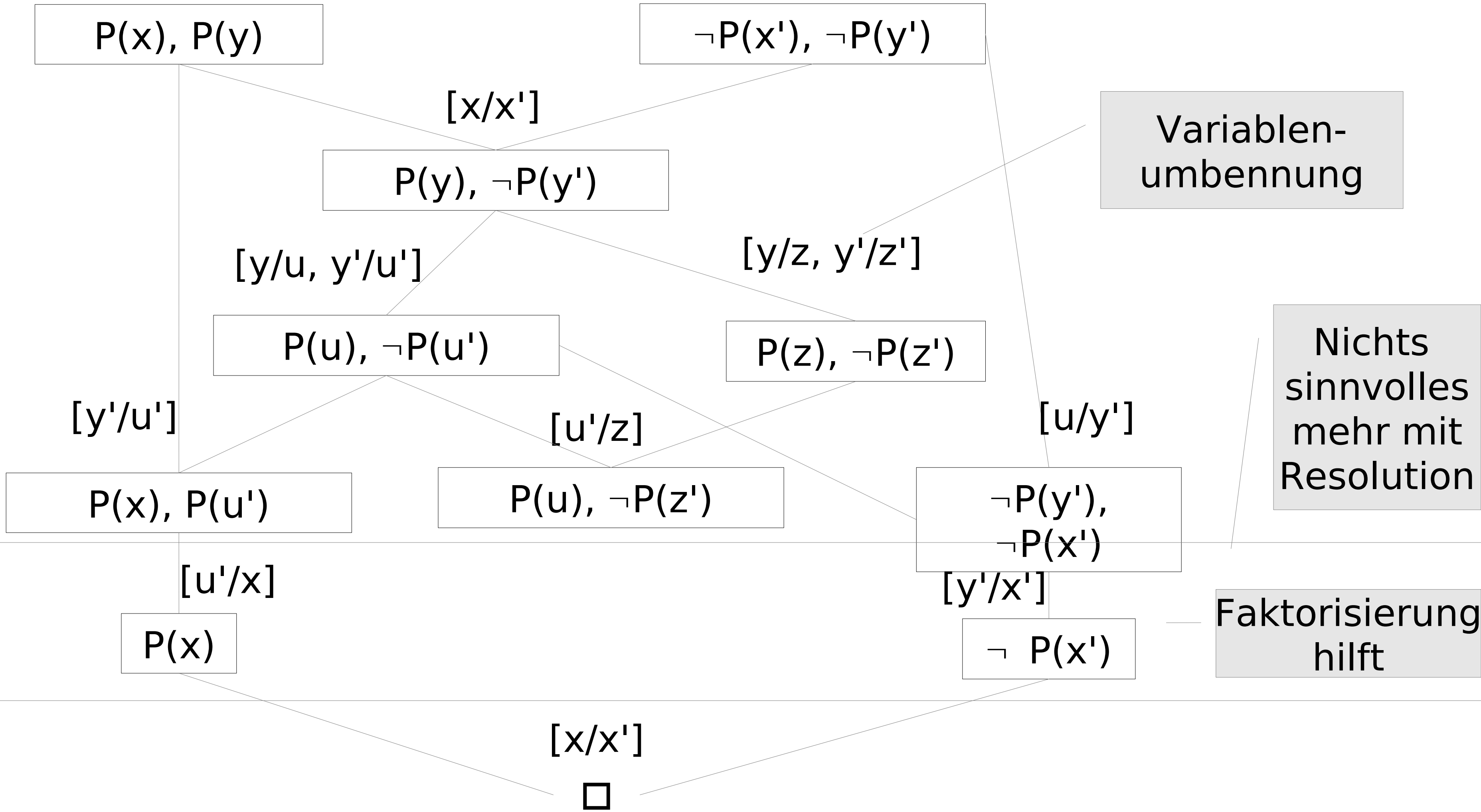
- { P(0), ∀x (P(x) ⇒ Q(f(x))) } ├ P(a)




- [Keine weitere Ableitung möglich (keine Resolution, keine Faktorisierung). Die leere Klausel kann nicht abgeleitet werden. P(a) folgt nicht, und die Resolution findet das heraus]

# Resolution – Beispiel

- $\{ \forall x \ \forall y \ P(x) \vee P(y) \} \vdash \exists x' \exists y' P(x') \wedge P(y')$ ?



# Resolution für Prädikatenlogik

- Prädikatenlogik ist unentscheidbar
  - Es gibt kein berechenbares Entscheidungsverfahren das entscheidet, ob eine Formel eine logische Folgerung aus einer Formelmenge ist oder nicht
- Trotzdem ist die Resolution ein vollständiger und korrekter Kalkül
- Drei Situationen können beim Beweisen von $\Sigma \vdash \beta$ auftreten
  - Die leere Klausel wird abgeleitet: Es gilt $\Sigma \vdash \beta$
  - Wir haben alle Klauseln konstruiert, die man mit Resolution und Faktorisierung ableiten kann und die leere Klausel konnte dabei nicht erzeugt werden: Es gilt **nicht** $\Sigma \vdash \beta$
  - Der Ableitungsvorgang dauert unendlich, beziehungsweise wir können noch keine Aussage zur Terminierung machen.
    Wir können keine Aussage zu $\Sigma \vdash \beta$ machen