Automatisches Beweisen

– Prädikatenlogik –

Prof. Dr. Wolfgang Küchlin

Dipl.-Inform., Dr. sc. techn. (ETH)

Arbeitsbereich Symbolisches Rechnen
Wilhelm-Schickard-Institut für Informatik
Fakultät für Informations- und Kognitionswissenschaften

Universität Tübingen

Steinbeis Transferzentrum
Objekt- und Internet-Technologien (OIT)

Wolfgang.kuechlin@uni-tuebingen.de
http://www-sr.informatik.uni-tuebingen.de

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Prädikatenlogik (PL1)

Prädikatenlogik (PL1)

- Sprache der Mathematik

- Neu im Vergleich zur Aussagenlogik
  - Funktions- und Relationssymbole *(predicate symbols)*
  - Existenz- und All-Quantoren
  - Atomare Formeln werden ersetzt durch Relationen (Prädikate) über Termen
  - Terme bezeichnen Individuen explizit
  - Es lassen sich unbeschränkt viele Terme bauen (z. Bsp. 0, F(0), F(F(0)), F(F(F(0))), …)

Syntax der Prädikatenlogik (1)

- Sprache:
  - Menge von (Individuen-) Variablen $\mathcal{V} = \{v_0, v_1, \ldots\}$
  - aussagenlogische Junktoren
  - Funktionssymbole: $\mathcal{F} = \{f, g, h, \ldots\}$ (Stelligkeit ≥ 0)
  - Prädikatssymbole: $\mathcal{P} = \{R, S, T, \ldots\}$ (Stelligkeit ≥ 0)
  - Quantoren: $\forall$, $\exists$
  - Hilfssymbole: Klammern, Komma

Syntax der Prädikatenlogik (2)

- Terme $T(\mathcal{V}, \mathcal{F})$: die kleinste Menge mit
  - $\mathcal{V} \subseteq T(\mathcal{V}, \mathcal{F})$
  - Falls $f \in F$ (mit Stelligkeit n) und $t_1, \ldots, t_n \in T(\mathcal{V}, \mathcal{F})$, so auch $ft_1 \ldots t_n \in T(\mathcal{V}, \mathcal{F})$.
- Beispiel:
  - $\mathcal{V} = \{x, y, z\}$
  - $\mathcal{F} = \{c, f, g\}$ (c 0-stellig, f 2-stellig, g 1-stellig}
  - Terme: fxfgyc oder gffcgxgz
  - Erweiterung mit Klammern: f(x, f(g(y), c)) oder g(f(f(c,g(x)),g(z)))
  - Achtung: Variablen stehen für Individuen, nicht mehr für Aussagen (anders als in der AL).

Syntax der Prädikatenlogik (3)

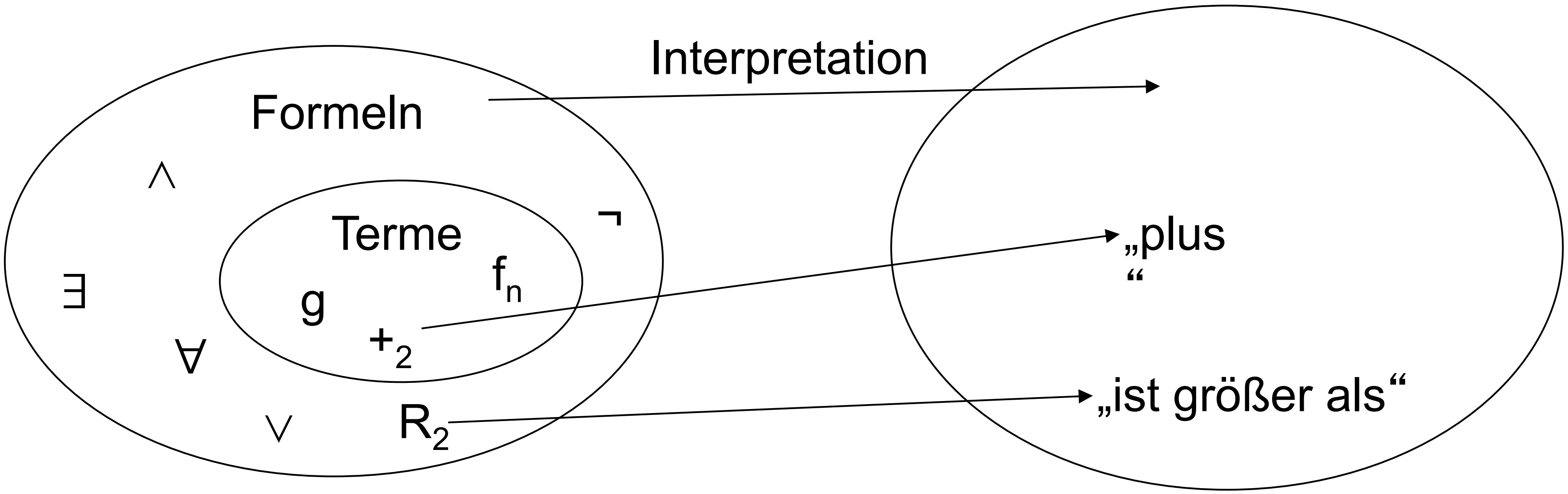
- **Relationssymbole** $\mathcal{R}$ bezeichnen Relationen (Bool-wertige Funktionen)
- **Formeln** $\Phi(\mathcal{V},\mathcal{F},\mathcal{R})$: Definiert als kleinste Menge, so dass (schreibe Φ anstelle von $\Phi(\mathcal{V},\mathcal{F},\mathcal{R})$)
  - $\perp \in \Phi$
  - Falls $R \in \mathcal{R}$ (Stelligkeit n) und $t_1, \ldots, t_n \in T(\mathcal{V},\mathcal{F})$, so ist $Rt_1 \ldots t_n \in \Phi$.
    - dieses sind die **atomaren Formeln**
  - Falls $F,G \in \Phi$, so auch $(F \vee G) \in \Phi$, $(F \wedge G) \in \Phi$ und $\neg F \in \Phi$.
  - Falls $x \in \mathcal{V}$ und $F \in \Phi$, so auch $\exists x F \in \Phi$ und $\forall x F \in \Phi$.

  - Beispiel: $\mathcal{V} = \{x, y\}$, $\mathcal{F} = \{f_2, g_1\}$, $\mathcal{R} = \{P_2, Q_2\}$
    Formel: $\forall x(\exists y Pxy \wedge Qfxgxy)$. Atomare Formel: $Qfxgxy$

Semantik der Prädikatenlogik

Logik

Strukturen (semantic domains)



Semantik der Prädikatenlogik (2)

- Semantic Domain benötigt Funktionen und Prädikate (Relationen)

- **($\mathcal{F}$, $\mathcal{R}$)-Struktur**:

  Tupel (A,μ), mit Universum $A \neq \{\ \}$ und Funktion

  μ (meaning function), die jedem $f_n \in \mathcal{F}$ eine

  n-stellige Funktion und jedem $R_m \in \mathcal{R}$ eine

  m-stellige Relation auf A zuweist.

Semantik der Prädikatenlogik (3)

- **Variablenbelegung**:

  $\mathcal{V}$ eine Variablenmenge, (A,μ) eine $(\mathcal{F}, \mathcal{R})$-Struktur

  Eine Variablenbelegung β ist eine Abbildung $\beta: V \rightarrow A$.

- Notation: $\beta[x/a]$

  die an Stelle x auf a abgeänderte Funktion β

$$\beta[x/a](y) = \begin{cases} a & \text{falls } y = x, \\ \beta(y) & \text{sonst} \end{cases}$$

Semantik der Prädikatenlogik (4)

➢ Interpretation

Tupel $(\mathcal{A}, \beta)$, bestehend aus $(\mathcal{F}, \mathcal{R})$-Struktur $\mathcal{A}$ und Variablenbelegung β für $\mathcal{V}$ in $\mathcal{A}$.

➢ Interpretation eines Terms

Sei $\mathcal{I} = (\mathcal{A}, \beta)$ eine Interpretation. $\mathcal{I}(t)$ ist rekursiv definiert durch:

- $\mathcal{I}(t) = \beta(t)$ falls $t \in \mathcal{V}$.
- $\mathcal{I}(ft_1 \ldots t_n) = \mu[f](\mathcal{I}(t_1), \ldots, \mathcal{I}(t_n))$.

Beispiel zur Interpretation eines Terms

- $A=(N, \mu)$ mit
  - $\mu(f)$: N x N → N: $(x,y) \mapsto x+y$
  - $\mu(g)$: N → N : $x \mapsto x+1$
  - $\mu(P) \subseteq N^2$: $(x,y) \in \mu(P)$ gdw $x=y$
  - $\mu(Q) \subseteq N^2$: $(x,y) \in \mu(Q)$ gdw $x<y$
- $\beta(x)=2$

$$\mathcal{I}(fxgx) = \mu(f)(\mathcal{I}(x), \mathcal{I}(gx)) = \beta(x)+\mu(g)(\mathcal{I}(x))$$

$$= 2+(2+1) = 5$$

Semantik der Prädikatenlogik (5)

- **Erfüllbarkeitsrelation**

  Sei $\mathcal{I}=(\mathcal{A}, \beta)$ Interpretation, F Formel.

  $\models_{\mathcal{I}} F$ definiert durch:

  - $\not\models_{\mathcal{I}} \bot$
  - $\models_{\mathcal{I}} Rt_1 \ldots t_n$ gdw. $(\mathcal{I}(t_1), \ldots, \mathcal{I}(t_n)) \in \mu(R)$
  - $\models_{\mathcal{I}} F \vee G$ gdw. $\models_{\mathcal{I}} F$ oder $\models_{\mathcal{I}} G$
  - $\models_{\mathcal{I}} F \wedge G$ gdw. $\models_{\mathcal{I}} F$ und $\models_{\mathcal{I}} G$
  - $\models_{\mathcal{I}} \neg F$ gdw. $\not\models_{\mathcal{I}} F$
  - $\models_{\mathcal{I}} \forall x F$ gdw. $\models_{\mathcal{I}[x/a]} F$ für alle $a \in \mathcal{A}$
  - $\models_{\mathcal{I}} \exists x F$ gdw. es gibt ein $a \in \mathcal{A}$ mit $\models_{\mathcal{I}[x/a]} F$

Sprechweisen

- Für $\models_{\mathcal{I}}$ F sagen wir
  - $\mathcal{I}$ erfüllt F *(satisfies, validates)*
  - F gilt unter $\mathcal{I}$ *(is valid)*
  - F ist wahr unter $\mathcal{I}$ *(is valid)*
  - $\mathcal{I}$ ist ein Modell von F
- Existiert ein $\mathcal{I}$, so dass $\models_{\mathcal{I}}$ F, so heißt F erfüllbar.
- Gilt $\models_{\mathcal{I}}$ F für alle $\mathcal{I}$, so heißt F allgemeingültig, $\models$F
- G $\models_{\mathcal{I}}$ F bedeutet: Falls $\models_{\mathcal{I}}$ G, dann auch $\models_{\mathcal{I}}$ F
- Folgendes ist möglich:
  - $\not\models$ F (im allgemeinen), aber $\models_{\mathcal{I}}$ F (im speziellen $\mathcal{I}$)
  - G $\not\models$ F (im allgemeinen), aber G $\models_{\mathcal{I}}$ F (im speziellen $\mathcal{I}$)

Substitutionen (1)

➢ Freie / gebundene Variablen

- Die Quantoren $\exists$ und $\forall$ binden Variablen.

- Fr(F) = Menge der freien Variablen von F
- Bd(F) = Menge der gebundenen Variablen von F

- Beispiel: $F = \forall x\ (\exists y\ Pxyz \lor Qfu) \land \exists z\ Rax$
  $Fr(F) = \{z, u, x\}$
  $Bd(F) = \{x, y, z\}$

Substitutionen (2)

- Die simultane Substitution $t[x_1,\ldots,x_r / t_1,\ldots,t_r]$ bzw. $F[x_1,\ldots,x_r / t_1,\ldots,t_r]$ ist für Terme $t, t_1, \ldots, t_r$, paarweise verschiedene Variablen $x_1, \ldots, x_r$ und Formeln F rekursiv definiert.
  - Basis:
    - $x[x_1,\ldots,x_r/t_1,\ldots,t_r] = \begin{cases} x & \text{falls } x \notin \{x_1,\ldots,x_r\} \\ t_i & \text{falls } x = x_i \end{cases}$
    - $f_0[x_1,\ldots,x_r/t_1,\ldots,t_r] = f$
    - $f(y_1,\ldots,y_k)[x_1,\ldots,x_r/t_1,\ldots,t_r] = f(y[x1,\ldots,x_r/t_1,\ldots,t_r], \ldots, y_k[x_1,\ldots,x_r/t_1,\ldots,t_r])$
    - $R(s1,\ldots,sk)[x_1,\ldots,x_r/t_1,\ldots,t_r] = R(s_1[x_1,\ldots,x_r/t_1,\ldots,t_r], \ldots, s_k[x_1,\ldots,x_r/t_1,\ldots,t_r])$
  - $(Qx\ F)[x_1,\ldots,x_r/t_1,\ldots,t_r] = Qu(F[x_{i_1},\ldots,x_{i_s}, x/t_{i_1},\ldots,t_{i_s}, u])$

    Dabei $x_i \in Fr(Qx\ F)$ und $x_i \neq t_i$. u neue Variable mit $u \notin Fr(F) \cup Var(t_{i_1}) \cup \ldots \cup Var(t_{i_s})$

    Falls $x \notin Var(t_{i_1}) \cup \ldots \cup Var(t_{i_s})$ kann u=x gewählt werden

    (benenne x in neue Variable u um, falls x in einem der Terme t_i vorkommt, und ersetze freie Variablen)

Beispiele zur Substitution

- $fyz[y,z,u/z,x,y] = fzx$
- $\forall x\ Pxy\ [y/x] = \forall u(Pxy[x,y/u,x]) = \forall u\ Pux$
- $(\exists x\ Pxfyz)[x,z/u,fyy] = \exists x\ Pxfyfyy$

Normalformen

- Wie schon die AL-Resolution benötigt auch die PL-Resolution eine Formel in Normalform: Klausel-Form
- Eine geschlossene Formel ist in **Klausel-Form**, falls sie von der Bauart

$$Qx_1 \ldots x_n : M$$

ist. Hierbei ist $Qx_1 \ldots x_n$ ein **Präfix** aus allquantifizierten Variablen und M ist eine quantorfreie **Matrix** in konjunktiver Normalform.
- Satz (Skolem): Zu jeder geschlossenen Formel A existiert eine erfüllbarkeits-äquivalente Formel A* in Klausel-Form, also $A^* \cong A$.

Normalformen

- **Negationsnormalform** (NNF)

  Formel ist in NNF, wenn $\neg$ nur noch vor Relationssymbolen oder vor $\perp$ vorkommt.

- Algorithmus:
  - $\neg\forall x\ F \quad \equiv> \quad \exists x\ \neg F$
  - $\neg\exists x\ F \quad \equiv> \quad \forall x\ \neg F$
  - AL-NNF-Transformationen

Normalformen

- **Pränexe-Normalform** (PNF, *prenex normal form*)
  Formel ist in PNF, falls sie von der folgenden Form ist:
  $$Q_1x_1 \ldots Q_nx_n\ F_0$$
- Algorithmus
  - $F \vee \exists\, xG \equiv> \ \exists y(F \vee G[x/y])$, wobei $y \notin Var(F) \cup Fr(G)$
  - $F \wedge \exists\, xG \equiv> \ \exists y(F \wedge G[x/y])$, wobei $y \notin Var(F) \cup Fr(G)$
  - $F \wedge \forall xG \equiv> \ \forall y(F \wedge G[x/y])$, wobei $y \notin Var(F) \cup Fr(G)$
  - $F \vee \forall xG \equiv> \ \forall y(F \vee G[x/y])$, wobei $y \notin Var(F) \cup Fr(G)$
- F ist in **Pränex-CNF (PCNF)**, falls F_0 in CNF
- Algorithmus: Distributivgesetz anwenden

Normalformen

- **Skolem-Normalform** (SNF)

  Formel ist in SNF, wenn sie in PCNF ist und wenn ihr Präfix nur universelle Quantoren enthält.

- Algorithmus:
  - $\forall x_1 \ldots \forall x_k \exists x_{k+1} Q_{k+2} x_{k+2} \ldots Q_n x_n F_0 \;\stackrel{\mathrm{s}}{=}\!>$
    $\forall x_1 \ldots \forall x_k Q_{k+2} x_{k+2} \ldots Q_n x_n (F_0[x_{k+1}/f x_1 \ldots x_k])$
    wobei f neues k-stelliges Funktionssymbol ist, das eine Skolem-Funktion bezeichnet, die zu jeder Kombination $x_1 \ldots x_k$ einen der für x_{k+1} existierenden Werte auswählt.

Normalformen - Beispiel

- $F = \exists x \forall y\ Rxy \wedge \neg \exists z \forall u\ Rzu$
- NNF: $\exists x \forall y\ Rxy \wedge \forall z \exists u\ \neg Rzu$
- PNF: $\exists x \forall y \forall z \exists u\ (Rxy \wedge \neg Rzu)$
- SNF: $\forall y \forall z (Rc_0 y \wedge \neg Rzf_2 yz)$

Normalformen

- Einführung von Skolemfkt. erhält nur die Erfüllbarkeit
- Je nachdem, wie die Quantoren extrahiert wurden, bekommt man unterschiedliche Skolemfunktionen
  - NNF: $\exists x \forall y\ Rxy \wedge \forall z \exists u\ \neg Rzu$
  - PNF1: $\exists x \forall y \forall z \exists u\ (Rxy \wedge \neg Rzu)$
  - SNF1: $\forall y \forall z (Rc_0 y \wedge \neg Rzf_2 yz)$
  - PNF2: $\forall z \exists u \exists x \forall y\ (Rxy \wedge \neg Rzu)$
  - SNF2: $\forall z \forall y (Rf_1 zy \wedge \neg Rzg_1 z)$
- Wir sind an den einfachsten Skolemfunktionen (ohne überflüssige Parameter) interessiert
  - $\exists$−Quantoren im engsten (innersten) Kontext ersetzen
  - danach die Allquantoren nach außen ziehen

Normalformen

➢ Skolem-Normalform und freie Variablen

- Nach der Skolemisierung werden alle Variablen als allquantifiziert angenommen
- Was passiert mit freien Variablen?
  - Erfüllbarkeit: Implizit existenzquantifiziert
  - Allgemeingültigkeit: Implizit allquantifiziert
- Wir wollen Erfüllbarkeit überprüfen ⇒ Freie Variablen sind implizit existenzquantifiziert auf äußerstem Level

➢ Beispiel:

- $\forall x \forall y \exists a\, P(f(a,b),g(x,y) \;\stackrel{\frown}{=}\!> \exists b \forall x \forall y \exists a\, P(f(a,b),g(x,y) \;\stackrel{\frown}{=}\!> \forall x \forall y \exists a\, P(f(a,c_1),g(x,y) \;\stackrel{\frown}{=}\!> \forall x \forall y\, P(f(c_2(x,y),c_1),g(x,y)$
- Alle vorkommenden Variablen (x,y) sind allquantifiziert

Zusammenfassung: Klausel-Form

- empfohlene Transformationsschritte zur Klausel-Form
  - Gebundene Variablen umbenennen (separieren)
  - Abgeleitete aussagenlog. Operatoren durch $\vee$, $\wedge$, $\neg$ ersetzen
  - NNF herstellen ($\neg$ nach innen schieben)
  - Quantoren nach innen schieben ($\exists$-Kontexte minimieren)
  - $\exists$-Quantoren eliminieren (Skolemfunktionen einführen)
  - $\forall$-Quantoren extrahieren (Reihenfolge egal)
  - Matrix in CNF konvertieren
- Am Beispiel
  - NNF: $\exists x \forall y\ Rxy \wedge \forall z \exists u\ \neg Rzu$
  - SNF3: $\forall y(Rc_0y) \wedge \forall z(\neg Rzg_1z)$
  - PNF3: $\forall y \forall z(Rc_0y \wedge \neg Rzg_1z)$

Unifikation

- AL-Resolution arbeitet auf komplementären Literalen
- Für PL-Resolution müssen komplementäre Literale i.A. durch **Unifikation** hergestellt werden.
- Beispiel: $\{\{P(x, f(x,y))\}, \{\neg P(g(c), f(z,c))\}\}$
  - Zunächst keine komplementären Literale vorhanden
  - Wegen Klausel-Form sind x, y, z allquantifiziert
  - Formel gilt also auch für $x \mapsto g(c)$, $z \mapsto g(c)$, $y \mapsto c$, also im Spezialfall: $\{P(g(c), f(g(c),c)), \neg P(g(c), f(g(c),c))\}$
  - Jetzt ist Resolution anwendbar und liefert $\square$
- Der Unifikations-Algorithmus sucht eine *allgemeinste* Substitution, die einen gemeinsamen Spezialfall liefert

Unifikation

- Hintergrund:
  syntaktisches Lösen von Termgleichungs-
  systemen

- Unifikation im Allgemeinen nicht eindeutig:
- Beispiel: $\{f(x,y)=z,\ g(y)=g(g(x))\}$
  - $y \mapsto g(x),\ z \mapsto f(x,g(x))$ oder
  - $x \mapsto a,\ y \mapsto g(a),\ z \mapsto f(a,g(a))$

Unifikation

- **Substitutor:**

  Abbildung $\sigma: \mathcal{V} \rightarrow T(\mathcal{V}, \mathcal{F})$, mit $\sigma(x)=x$ für fast alle x

  - Mappt Variablen auf Terme, mit denen sie substituiert werden
  - normalerweise in Postfix geschrieben: $x\sigma$

- **Umbenennung:**

  bijektiver Substitutor

- **Separator** von K_1 und K_2:

  Umbenennung ξ mit $Fr(K_1\xi) \cap Fr(K_2) = \{\,\}$

  - „Trennt“ die freien Variablen von K_1 und K_2
  - Benennt alle freien Variablen in K_1 um, die in K_2 vorkommen

Unifikation

- Unifikator einer Literalmenge $\mathcal{L}$:
  Substitutor σ mit $\mathcal{L}\sigma$ einelementig
  ($\mathcal{L}$ heißt unifizierbar, falls es einen Unifikator gibt.)

- allgemeinster Unifikator (mgu) von $\mathcal{L}$:
  Unifikator μ, so dass es für jeden anderen Unifikator ν einen Substitutor σ gibt mit $\mu\sigma=\nu$
  (d.h. für alle x gilt: $\sigma(\mu(x))=\nu(x)$))

Unifikationsalgorithmus nach J.R. Robinson

1. Falls in $\mathcal{L}$ verschiedene Prädikatssymbole auftauchen:
   STOP mit „$\mathcal{L}$ ist nicht unifizierbar.“
2. i:=0; μ_i:=id;
3. Falls $\mathcal{L}\mu_i$ einelementig, **STOP mit „μ_i ist mgu“**
4. Wähle P_1, P_2 (Literale) aus $\mathcal{L}\mu_i$ mit $P_1 \neq P_2$. Seien s_1 und s_2, die ersten unterschiedlichen Symbole (von links gelesen).
   Falls s_1 und s_2 Funktionssymbole: **STOP mit „$\mathcal{L}$ ist nicht unifzierbar.“**
5. Falls $s=s_1$ Variable, bestimme Term t in P_2, der an Position von s_2 beginnt.
   Falls $s=s_2$ Variable, entsprechendes mit s_2 und P_1
6. Falls $s \in t$ *(occurrence check)*: **STOP mit „$\mathcal{L}$ ist nicht unifizierbar“**
7. $\mu_{i+1} := \mu_i \cup \{s \mapsto t\}$;
   i:=i+1;
8. Goto 3;

Unifikation - Beispiele

- $\{x, a\}$ und $\{R(x,g(y)), R(g(a),g(a))\}$ unifizierbar

- $\{R(y,y), R(g(x),x)\}$ und $\{\ \}$ nicht unifizierbar

- $\{P(f(x,y),g(y)), P(z,g(g(x)))\}$ unifizierbar mit
  $\nu=\{y \mapsto g(a),\ z \mapsto f(x,g(x)),\ x \mapsto a\}$

- $\{R(x,g(y)), R(u,v)\}$ unifzierbar mit
  $\mu=\{x \mapsto u,\ v \mapsto g(y)\}$ bzw. $\mu'=\{u \mapsto x,\ v \mapsto g(y)\}$

Prädikatenlogische Resolution

- Klauseln $K_1=M_1 \cup L_1$ und $K_2=M_2 \cup L_2$.
  - L_1, L_2 zwei Literale in denen der Widersprurch erzeugt werden soll
  - M_1, M_2, die restlichen Literalmengen der Klauseln
- Bestimme Seperator ξ von K_1 und K_2
- Bestimme mgu μ von $L_1\xi \cup \neg L_2$
- Resolvente von K1 und K2 ist dann $(M_1\xi \cup M_2)\mu$

$$\begin{tikzcd}
\overbrace{M_1 \cup L_1}^{K_1} \arrow[dr, dash, "\xi"'] & & \overbrace{M_2 \cup L_2}^{K_2} \arrow[dl, dash] \\
& (M_1\xi \cup M_2)\mu &
\end{tikzcd}$$

Beispiel PL1-Resolution

$$\{\overbrace{\neg P(x,y,c)}^{M_1},\ \overbrace{R(y,g(f(x)))}^{L_1}\} \qquad \{\overbrace{\neg R(f(x),g(y))}^{L_2}\} \quad M2=\{\,\}$$

$\xi = \{x \mapsto u,\ y \mapsto v\}$

$$K_1\xi = \{\neg P(u,v,c),\ R(v,g(f(u)))\}$$

$\mu = \{v \mapsto fx,\ y \mapsto fu\}$

$$(M_1\xi \cup M_2)\mu = \{\neg P(u,f(x),c)\}$$