Automatisches Beweisen

– Kap 2.4 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

SR

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 $f \in \mathcal{F}$ (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)

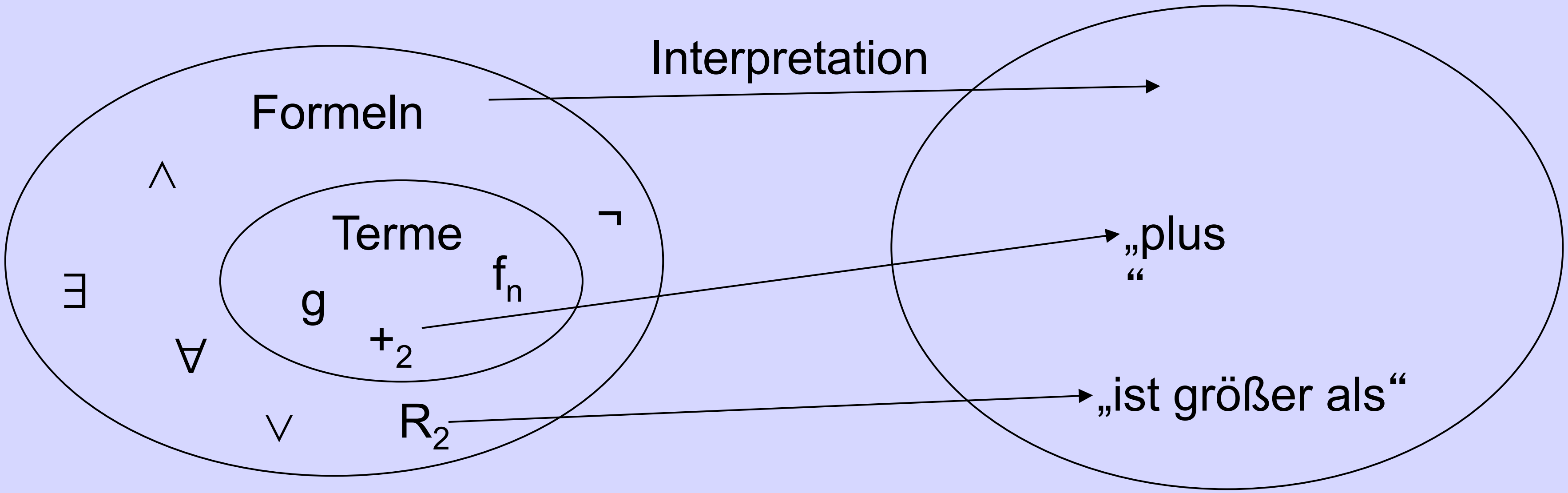
- **Relationssymbole** $\mathcal{R}$ bezeichnen Relationen (Bool-wertige Funktionen)
- **Formeln** $\Phi(\mathcal{V},\mathcal{F},\mathcal{R})$: Definiert als kleinste Menge, so dass (schreibe Φ anstelle von $\Phi(\mathcal{V},\mathcal{F},\mathcal{R})$)
  - $\bot \in \Phi$
  - Falls $R \in \mathcal{R}$ (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)

- Eine passende Semantic Domain benötigt Funktionen und Prädikate (Relationen) passend zu den Funktionssymbolen $\mathcal{F}$ und den Relationssymbolen $\mathcal{R}$.

- **($\mathcal{F}$, $\mathcal{R}$)-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 β: $\mathcal{V} \rightarrow$A.

- Notation: β[x/a]

  die an Stelle x auf a abgeänderte Funktion β

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

Semantik der Prädikatenlogik (4)

- **Interpretation**

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

- **Interpretation eines Terms**

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

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

Beispiel zur Interpretation eines Terms

- $A=(N, \mu)$ mit
  - $\mu(f): N \times N \to N: (x,y) \mapsto x+y$
  - $\mu(g): N \to N : x \mapsto x+1$
  - $\mu(P) \subseteq N^2: (x,y) \in \mu(P)$ 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 $\models_I$ F sagen wir
  - $\mathcal{I}$ erfüllt F *(satisfies* F, *validates* F*)*
  - F gilt unter $\mathcal{I}$ *(is valid under* $\mathcal{I}$*)*
  - F ist wahr unter $\mathcal{I}$ *(is valid under* $\mathcal{I}$*)*
  - $\mathcal{I}$ ist ein Modell von F ($\mathcal{I}$ models F)
- Existiert ein $\mathcal{I}$, so dass $\models_I$ F, so heißt F erfüllbar.
- Gilt $\models_I$ F für alle $\mathcal{I}$, so heißt F allgemeingültig, $\models$F
- G $\models_I$ F bedeutet: Falls $\models_I$ G, dann auch $\models_I$ F
- Folgendes ist möglich:
  - $\not\models$ F (im allgemeinen), aber $\models_I$ F (im speziellen $\mathcal{I}$)
  - G $\not\models$ F (im allgemeinen), aber G $\models_I$ F (im speziellen $\mathcal{I}$)

SR

Substitutionen (1)

- **Freie / gebundene Variablen**
  - Die Quantoren ∃ und ∀ binden Variablen.
  - Fr(F) = Menge der freien Variablen von F
  - Bd(F) = Menge der gebundenen Variablen von F
  - Beispiel: $F = \forall x\, (\exists y\, Pxyz \vee Qfu) \wedge \exists z\, Rax$

    $Fr(F) = \{z, u, x\}$

    $Bd(F) = \{x, y, z\}$

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:
    - $x[x_1,\ldots,x_r/t_1,\ldots,t_r] = \begin{cases} x & \text{falls } x \notin \{x_1,\ldots,x_r\} \\ t_i & \text{falls } x = x_i \end{cases}$
    - $f_0[x_1,\ldots,x_r/t_1,\ldots,t_r] = f_0$
    - $f(y_1,\ldots,y_k)[x_1,\ldots,x_r/t_1,\ldots,t_r] = f(y_1[x1,\ldots,x_r/t_1,\ldots,t_r], \ldots, y_k[x_1,\ldots,x_r/t_1,\ldots,t_r])$
    - $R(s_1,\ldots,s_k)[x_1,\ldots,x_r/t_1,\ldots,t_r] = R(s_1[x_1,\ldots,x_r/t_1,\ldots,t_r], \ldots, s_k[x_1,\ldots,x_r/t_1,\ldots,t_r])$
  - $(Qx\,F)[x_1,\ldots,x_r/t_1,\ldots,t_r] = Qu(F[x_{i_1},\ldots,x_{i_s},x/t_{i_1},\ldots,t_{i_s},u])$

    Dabei $x_{ik} \in Fr(Qx\,F)$ 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

    (beschränke die Subst. auf Fr(F) und benenne x in neue Variable u um, falls x in einem der Terme t_{ik} vorkommt)

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 $A^* \triangleq A$.
- Wir beschränken uns im Folgenden auf geschlossene Formeln (ohne freie Variablen). Falls zunächst eine freie Variable vorkommt, wird diese gemäß unserer Intention quantifiziert.
  - Bsp: $\forall x$ Pxy. Wollen wir beweisen, dass dies für alle y gilt, oder dass es ein y gibt, sodass dies gilt?

Normalformen

- **Negationsnormalform** (NNF)

  Formel ist in NNF, wenn $\neg$ nur noch vor Relationssymbolen oder vor $\bot$ 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:

  $$Q_1 x_1 \ldots Q_n x_n \; F_0$$

- Algorithmus
  - $F \vee \exists x G \equiv> \exists y(F \vee G[x/y])$, wobei $y \notin Var(F) \cup Fr(G)$
  - $F \wedge \exists x G \equiv> \exists y(F \wedge G[x/y])$, wobei $y \notin Var(F) \cup Fr(G)$
  - $F \wedge \forall x G \equiv> \forall y(F \wedge G[x/y])$, wobei $y \notin Var(F) \cup Fr(G)$
  - $F \vee \forall x G \equiv> \forall y(F \vee G[x/y])$, 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:
  - $\forall x_1 \ldots \forall x_k \exists x_{k+1} Q_{k+2} x_{k+2} \ldots Q_n x_n F_0 \overset{sat}{\equiv}>$
    $\forall x_1 \ldots \forall x_k Q_{k+2} x_{k+2} \ldots Q_n x_n (F_0[x_{k+1}/f x_1 \ldots x_k])$
    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

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: $\forall y (Rc_0 y) \wedge \forall z (\neg Rzg_1 z)$
  - PNF3: $\forall y \forall z (Rc_0 y \wedge \neg Rzg_1 z)$

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 all-quantifiziert
  - 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

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

  - normalerweise in Postfix geschrieben: $x\sigma$

- Umbenennung:

  bijektiver Substitutor

- Separator von K_1 und K_2:

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

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: $\sigma(\mu(x))=\nu(x)$)

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. $\mu‘=\{u \mapsto x, v \mapsto g(y)\}$

Unifikationsalgorithmus nach J.R. Robinson

1. Falls in L verschiedene Prädikatssymbole auftauchen, STOP mit „L ist nicht unifizierbar."
2. $i:=0$; $\mu_i:=id$;
3. Falls $L\mu_i$ einelementig, STOP mit „μ_i ist mgu"
4. Wähle F_1, F_2 aus $L\mu_i$ mit $F_1 \neq F_2$. Seien s_1 und s_2, die ersten unterschiedlichen Symbole.

   Falls s_1 und s_2 Funktionssymbole, STOP mit „L ist nicht unifzierbar."
5. Falls s_1 Variable, bestimme Term t in F_2, der an Position von s_2 beginnt. // Falls s_2 Variable, entsprechendes mit s_2 und F_1
6. Falls $s_1 \in t$, // *(occurrence check)*

   STOP mit „L ist nicht unifizierbar"
7. $\mu_{i+1}:=\mu_i\{s_1 \mapsto t\}$; $i:=i+1$;
8. Goto 3;

Prädikatenlogische Resolution

- Klauseln $K_1=M_1\cup L_1$ und $K_2=M_2\cup L_2$.

$$\overbrace{M_1 \cup L_1}^{K_1} \qquad \overbrace{M_2 \cup L_2}^{K_2}$$

$$\xi$$

$$\mu$$

$$K$$

ξ Separator von K_1 und K_2

$L_1\xi\cup\neg L_2$ unifizierbar mit mgu μ

Beispiel PL1-Resolution

$$\{\overbrace{\neg Pxyc}^{M_1}, \overbrace{Rygfx}^{L_1}\} \qquad \{\overbrace{\neg Rfxgy}^{L_2}\} \quad M2=\{\ \}$$

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

$\{\neg Puvc,\ Rvgfu\}$

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

$$\{\neg Pufxc\}$$

Korrektheit und Vollständigkeit der PL-Resolution

- Korrektheit:

  $F \vdash_{Res} C$ impliziert $F \models C$

  - relativ leicht zu zeigen. Beide Elternklauseln müssen in allen Modellen gelten. Da die komplementären Literale nicht gleichzeitig gelten können, muss die Resolvente gelten.

- Widerlegungsvollständigkeit:

  $F \models \square$ impliziert $F \vdash_{Res} \square$.

  - Beweis ist aufwändig. Die Prädikatenlogischen Modelle sind komplex, da Funktions- und Prädikatssymbole durch beliebige Funktionen und Relationen passender Stelligkeit interpretiert werden können.
  - Man zeigt zuerst: Herbrand-Modelle sind allgemeinste Modelle
    - Es gibt ein Modell gdw. es gibt ein Herbrand Modell.
  - Man zeigt danach: Falls es kein Herbrand-Modell gibt, folgt $F \vdash_{Res} \square$.

Herbrand Universum HU(S)

- Das **Herbrand Universum** HU(S) ist eine allgemeinste Domäne für eine Interpretation der Terme
- Sei S eine Menge von Klauseln in SNF.

  HU(S) ist wie folgt definiert
  - Alle Konstanten-Symbole von S sind in HU(S).
    - Gibt es ein solches nicht, sei ein beliebiges Symbol a in HU(S). Hierdurch beschränkt man sich o.B.d.A. auf nicht-leere Domänen.
  - Falls Terme $t_1, \ldots, t_n$ in HU(S), so auch $f(t_1, \ldots, t_n)$ für jedes n-stellige Funktionssymbol f in $\mathcal{F}$.
  - Der Term fab als Zeichenreihe wird durch den Syntaxbaum f(a,b) interpretiert
  - HU(S) enthält Terme, die die Elemente der Interpretations-Domäne *(semantic domain elements)* darstellen. Falls ein Funktionssymbol (mit Stelligkeit > 0) existiert, ist HU(S) bereits unendlich.

Herbrand Basis HB(S)

- Die **Herbrand Basis** HB(S) enthält alle Grund-Instanzen aller atomaren Formeln aus S. (Einsetzen des Herbrand-Universums in die Variablen.)
  - Manchmal auch: alle atomaren Formeln, die sich mit den Prädikatssymbolen und Termen aus HU(S) bilden lassen

- Beispiel:

  $S := \{P(x) \vee Q(y), \neg P(a), \neg Q(b)\}$

  $HU(S) = \{a, b\}$

  $HB(S) = \{P(a), P(b), Q(a), Q(b)\}$

Herbrand Interpretation

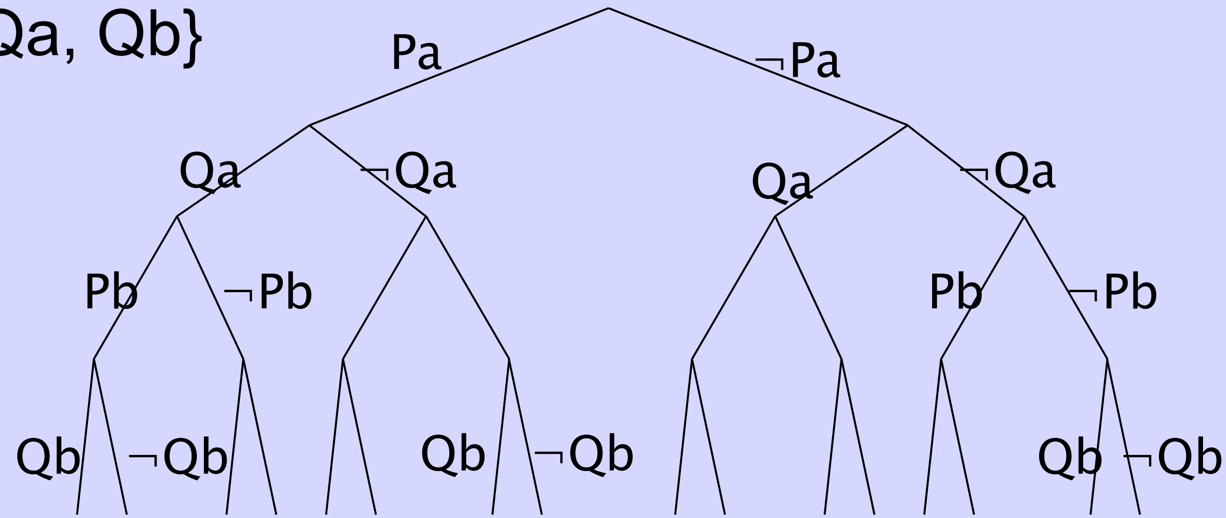
- Eine allgemeinste Interpretation der Formel S
- Die symbolischen Terme werden durch die Elemente von HU(S) interpretiert.
- Interpretation der Funktionssymbole wird erweitert um (minimale) Interpretation der Prädikatssymbole, sodass die Elemente von HB(S) auf 1 oder 0 abgebildet werden
  - Jede Interpretation der Prädikatssymbole durch Relationen muss jeder atomaren Formel aus HB(S) einen Wert zuordnen
  - Hierfür gibt es viele Möglichkeiten, also gibt es auch viele Herbrand-Interpretationen. Diese unterscheiden sich aber nur in der Zuordnung von 1 bzw. 0 zu den Elementen von HB(S).

Semantische Bäume

- Ein **Semantischer Baum** **ST(S)** repräsentiert alle möglichen Herbrand-Interpretationen von S.
  - Jeder von der Wurzel ausgehende Pfad ist eine mögliche H-Interpretation
  - Je nach Aufzählung von HB(S) ein anderer (äquivalenter) Baum
  - Im Allgemeinen ist der Baum unendlich
- Beispiel (endlich): S={Px v Qy, ¬Pa, ¬Pb},

HB(S)={Pa, Pb, Qa, Qb}

HU(S)={a, b}



Herbrand-Modell

- Eine Herbrand-Interpretation, durch die S erfüllt wird, ist ein Herbrand-Modell von S.
- Ein Herbrand Modell von S ist eindeutig charakterisiert durch diejenigen Elemente der Herbrand-Basis, deren Wert =1 ist.
- Eine Klauselmenge S hat genau dann ein Modell, wenn sie ein Herbrand-Modell hat.
  - „=>": Das benötigte Herbrand-Modell besteht aus denjenigen Elementen von HB(S), die im Modell zu 1 evaluieren.
  - Es ist essentiell, dass S skolemisiert wurde, sodass die nötigen Elemente in HU(S) und damit in HB(S) vorhanden sind.
  - Korollar: kein Modell $\Leftrightarrow$ kein Herbrand-Modell

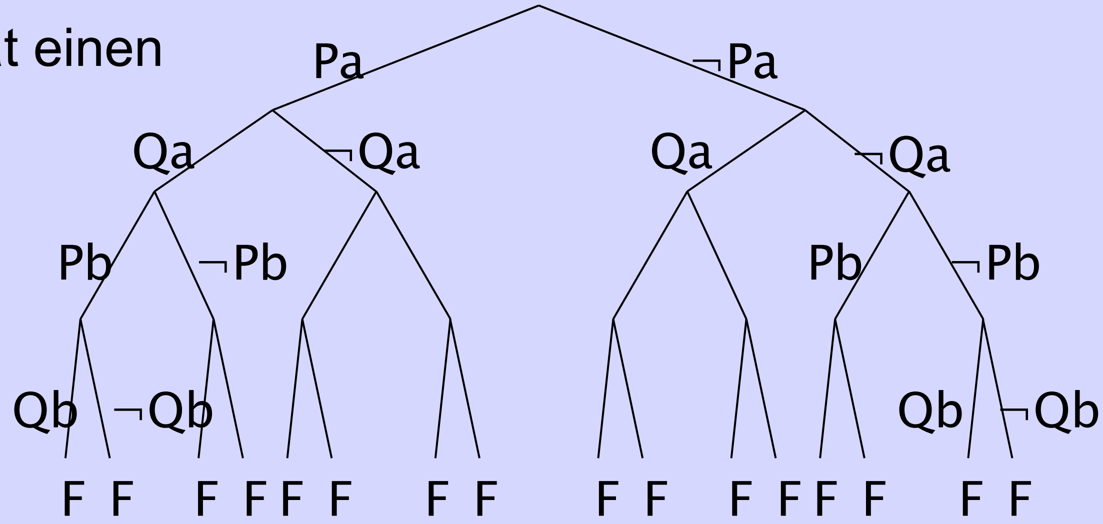
Semantische Bäume und Herbrand-Modelle

- Eine Interpretation (=Pfad) kann Modell sein oder nicht.
  - Sie ist ein Modell, wenn sie alle Klauseln in S erfüllt
  - Sie ist kein Modell, wenn sie (mind.) eine Klausel K falsifiziert.
    - Es gibt einen **Fehlerknoten** endlicher Tiefe mit einer **Fehlerklausel,** die durch die endliche Teilmenge von HB(S) von der Wurzel zum Fehlerknoten falsifiziert wird
    - Ein Pfad mit Fehlerknoten heißt **geschlossen**, sonst **offen**
    - Ein geschlossener Pfad hat einen höchsten Fehlerknoten
- Beispiel 1

  S={Px ∨ Qy, ¬Pa, ¬Pb},

  HB(S)={Pa, Pb, Qa, Qb}

  HU(S)={a, b}



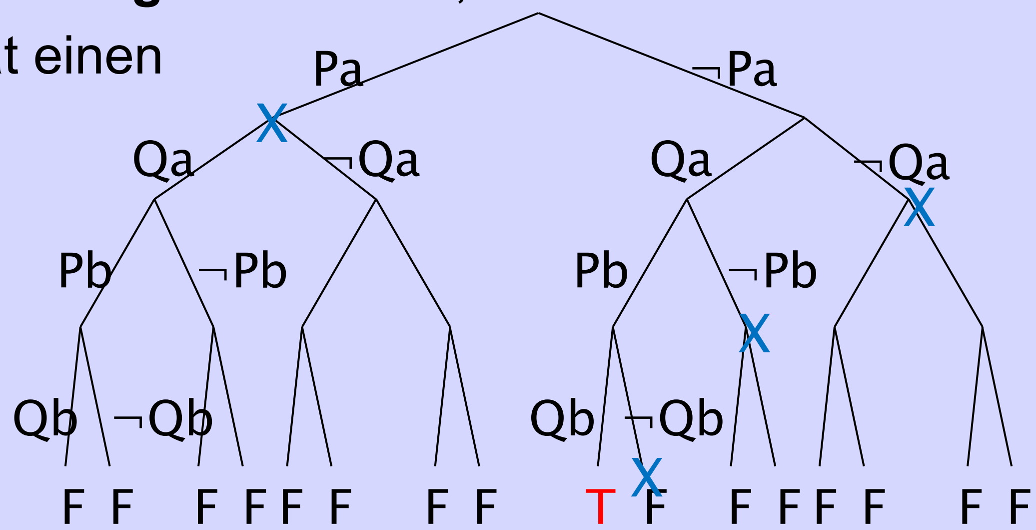
Semantische Bäume und Herbrand-Modelle

- Eine Interpretation (=Pfad) kann Modell sein oder nicht.
  - Sie ist ein Modell, wenn sie alle Klauseln in S erfüllt
  - Sie ist kein Modell, wenn sie (mind.) eine Klausel K falsifiziert.
    - Es gibt einen **Fehlerknoten** endlicher Tiefe mit einer **Fehlerklausel,** die durch die endliche Teilmenge von HB(S) von der Wurzel zum Fehlerknoten falsifiziert wird
    - Ein Pfad mit Fehlerknoten heißt **geschlossen**, sonst **offen**
    - Ein geschlossener Pfad hat einen höchsten Fehlerknoten
- Beispiel 2

  S={Px v Qy, ¬Pa, Pb},

  HB(S)={Pa, Pb, Qa, Qb}

  HU(S)={a, b}



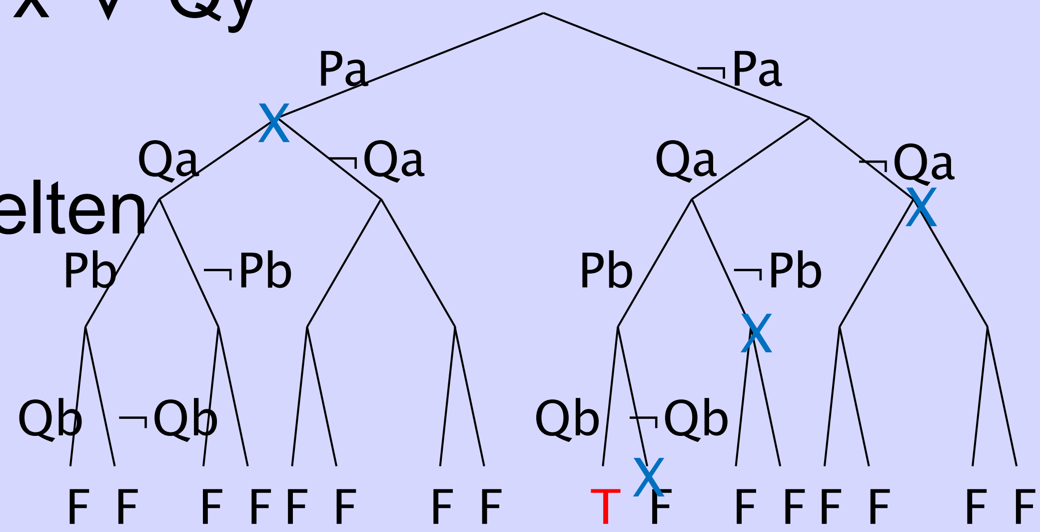
Einschub: Allquantifizierung

- Es gilt: $\forall x Px \lor \forall y Qy \equiv \forall x \forall y\,(Px \lor Qy)$
- Beispiel 2: $S=\{Px \lor Qy, \neg Pa, Pb\}$,

  $HB(S)=\{Pa, Pb, Qa, Qb\}$, $HU(S)=\{a, b\}$

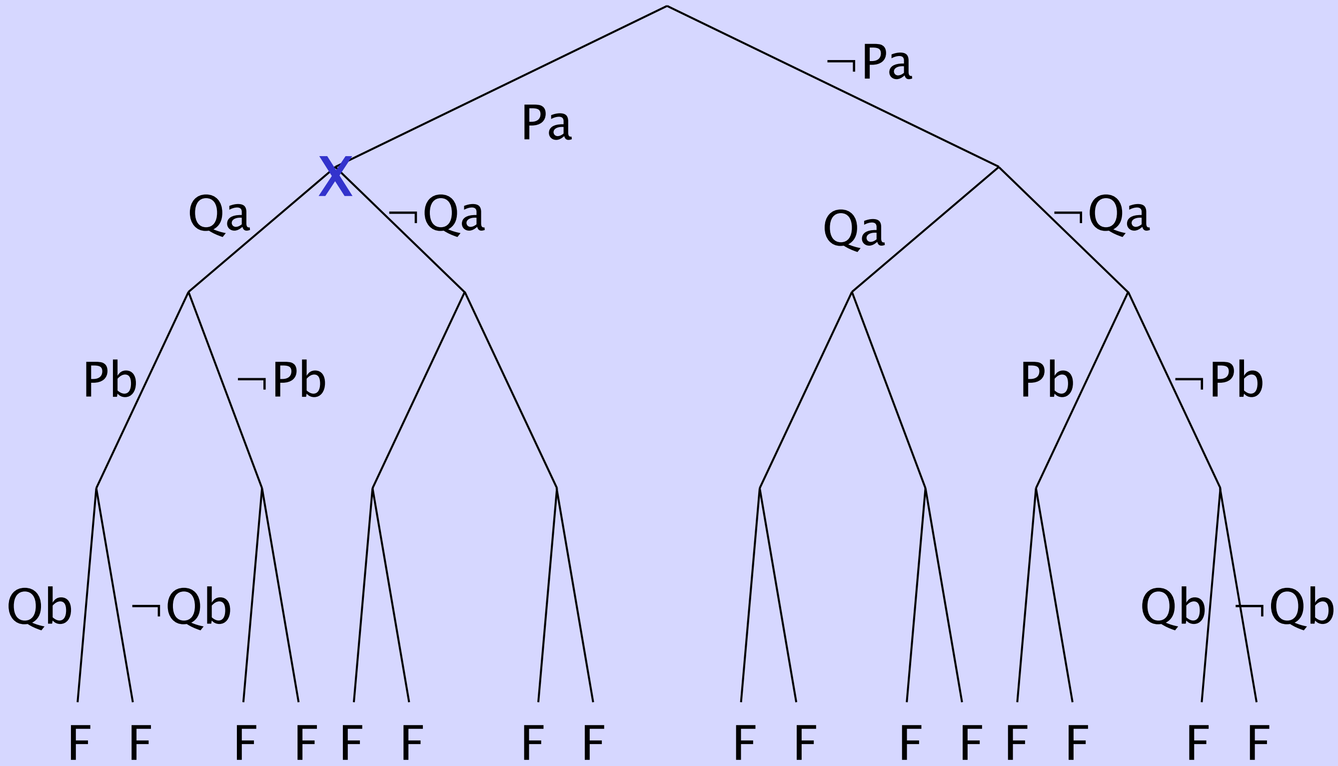
  $\forall x \forall y\,(Px \lor Qy) \equiv \forall y\,(Pa \lor Qy) \land \forall y\,(Pb \lor Qy)$

  $\equiv [(Pa \lor Qa) \land (Pa \lor Qb)] \land [(Pb \lor Qa) \land (Pb \lor Qb)]$
- Jede Kombination von $Px \lor Qy$ muss gelten
- $\forall x Px \lor \forall y Qy$ muss gelten
  - ansonsten Kombination der Gegenbeisp. möglich



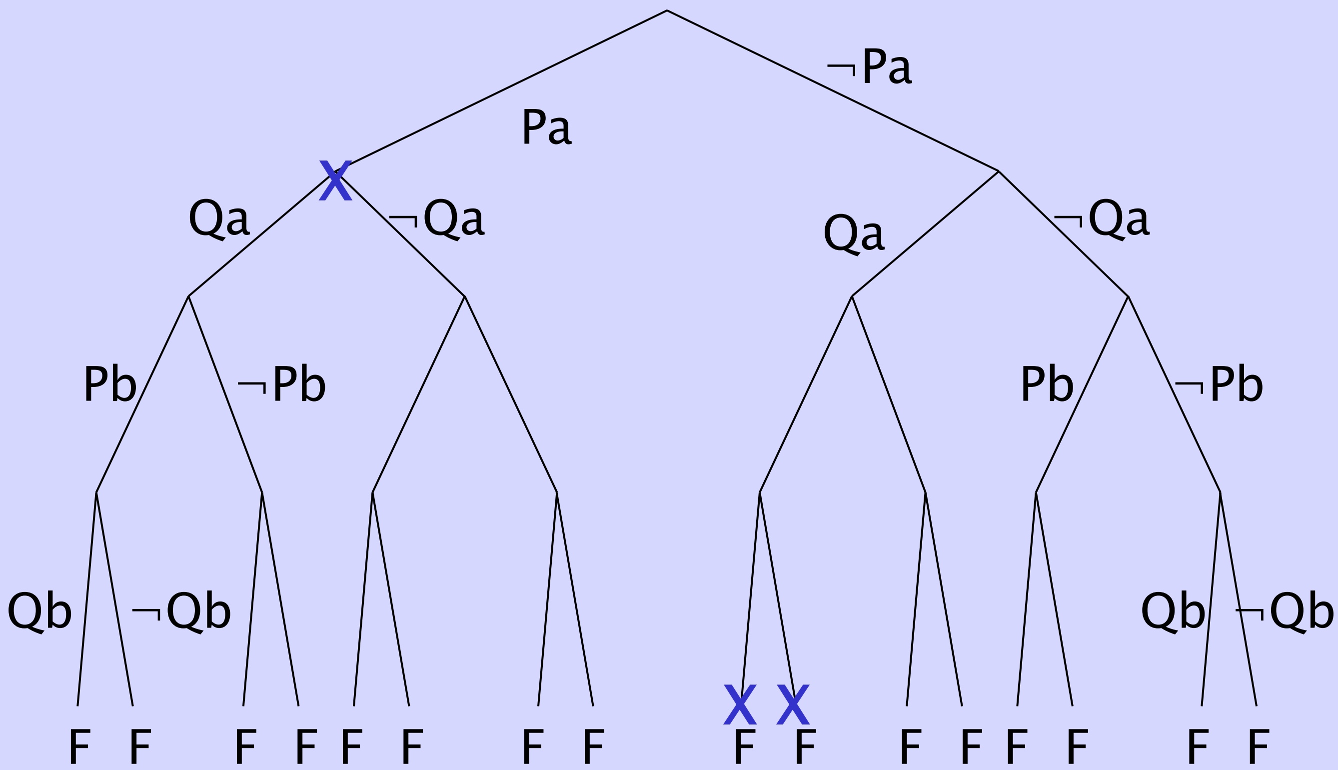
Semantische Bäume, Fehlerknoten

- Bsp. 3: S={Px ∨ Qy, ¬Pa, ¬Qb}, H(S)={a, b}, H-B(S)={Pa, Pb, Qa, Qb}



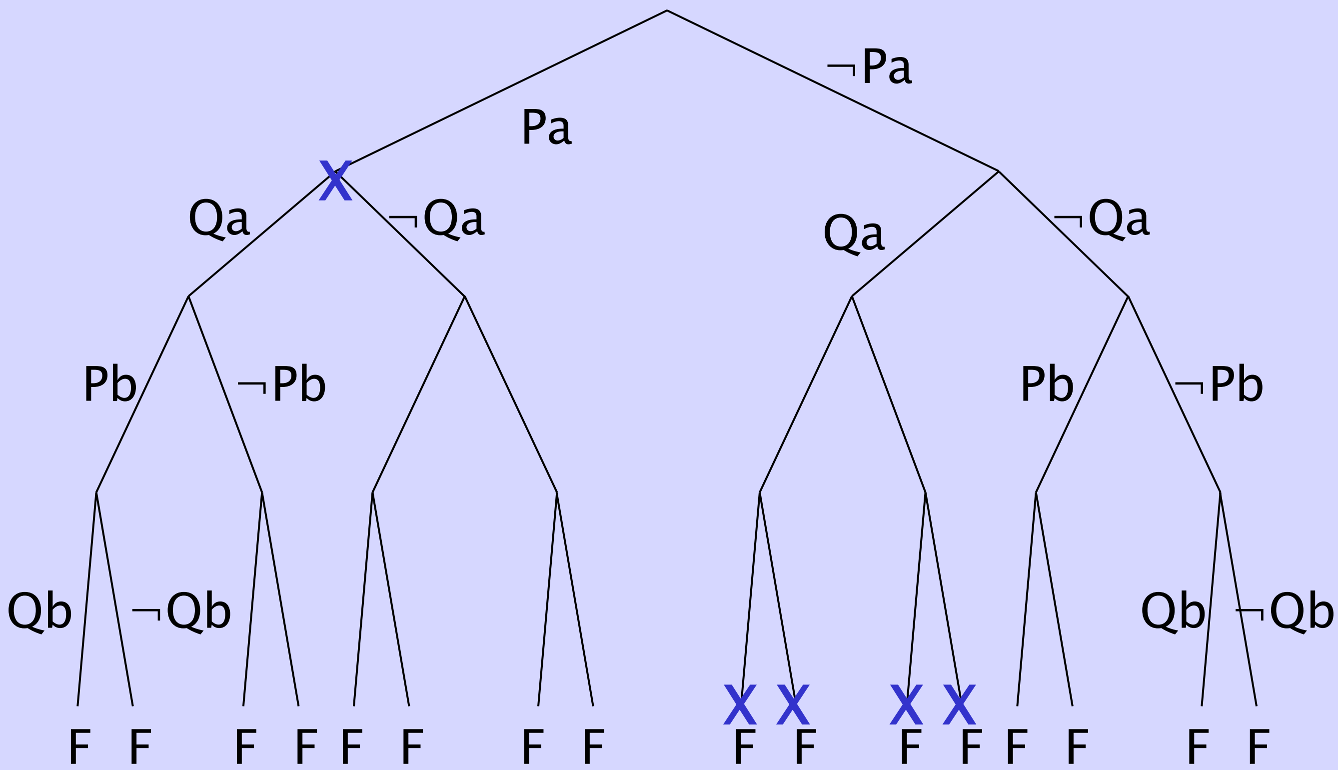
Semantische Bäume, Fehlerknoten

- Bsp. 3: S={Px ∨ Qy, ¬Pa, ¬Qb}, H(S)={a, b}, H-B(S)={Pa, Pb, Qa, Qb}



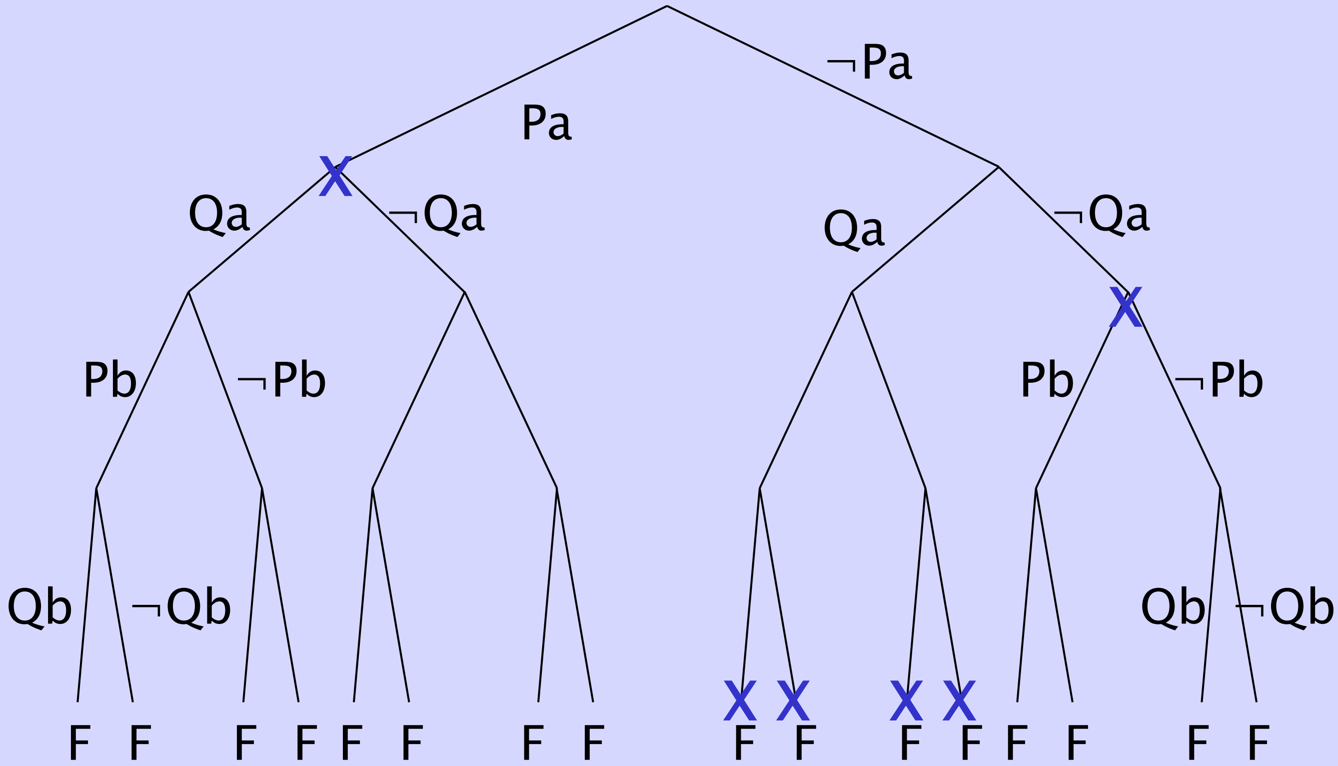
Semantische Bäume, Fehlerknoten

- Bsp. 3: S={Px v Qy, ¬Pa, ¬Qb}, H(S)={a, b}, H-B(S)={Pa, Pb, Qa, Qb}



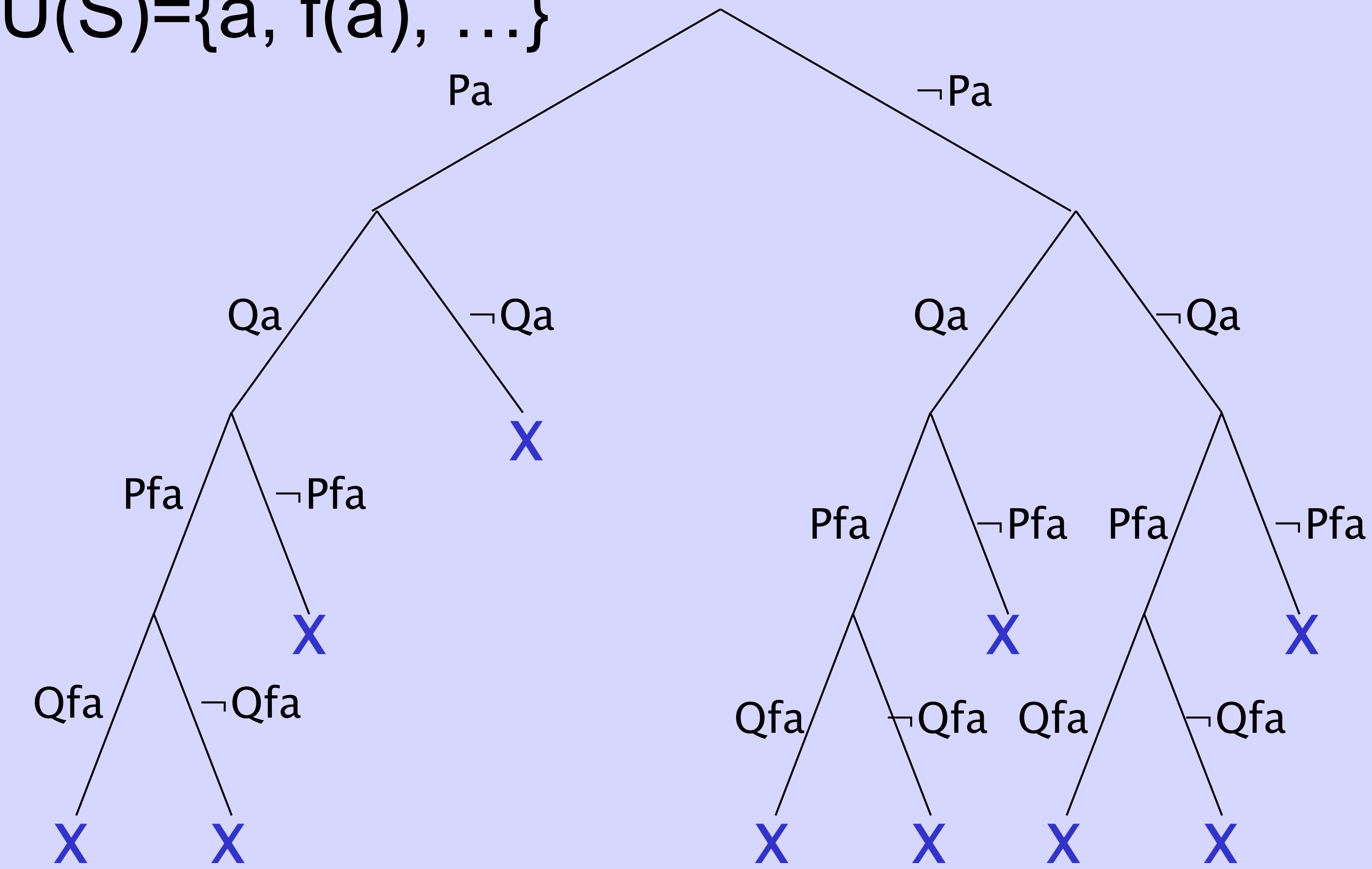
Semantische Bäume, Fehlerknoten

- Bsp. 3: S={Px ∨ Qy, ¬Pa, ¬Qb}, H(S)={a, b}, H-B(S)={Pa, Pb, Qa, Qb}



Semantische Bäume, Fehlerknoten

- Bsp. 4: $S=\{\neg Px \vee Qx, Pfy, \neg Qfz\}$, ST(S) ist unendlich
  HU(S)={a, f(a), …}



Satz von Herbrand

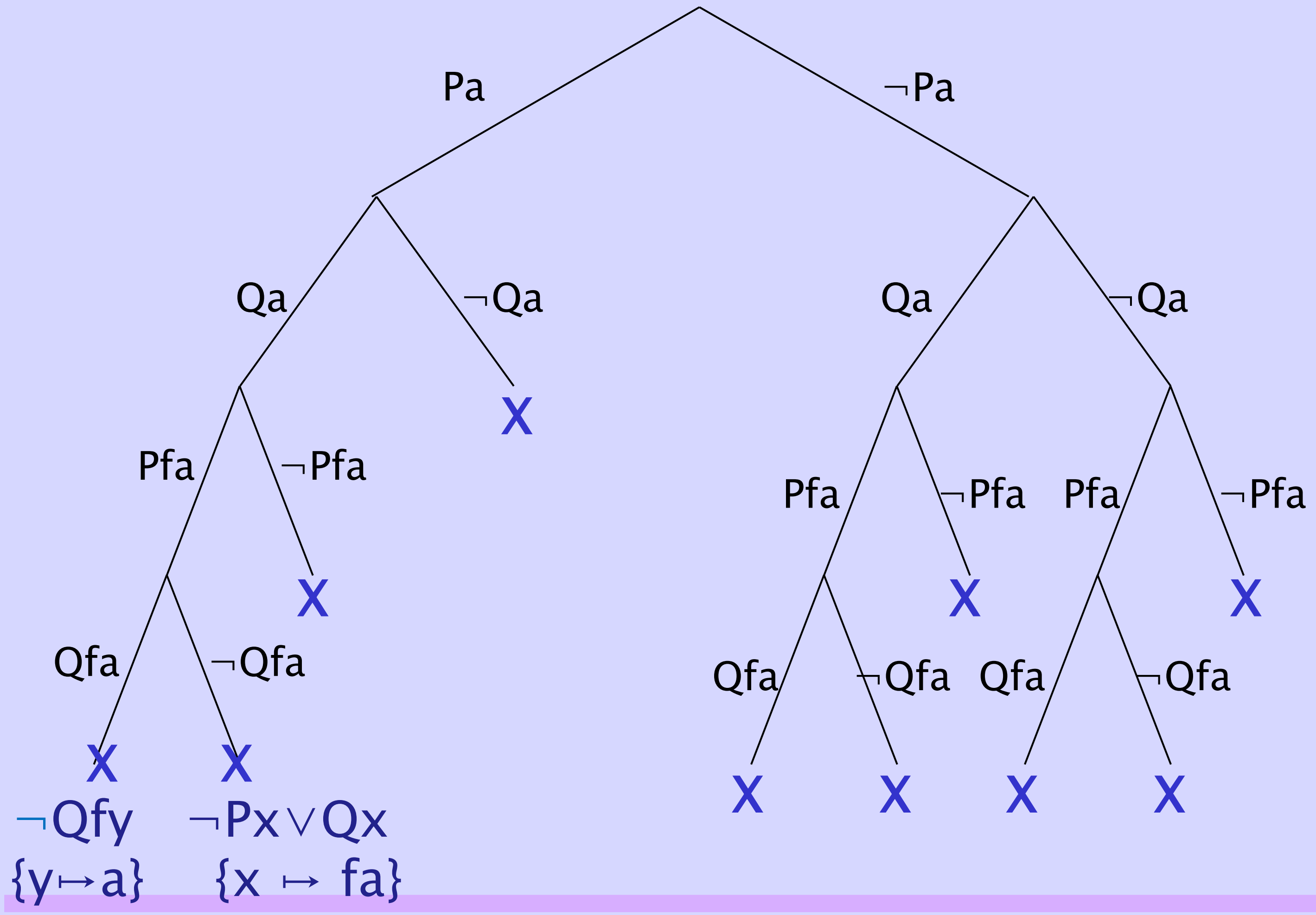
- Ein semantischer Baum heißt **geschlossen**, wenn alle seine Pfade geschlossen sind
  - Jeder Pfad hat einen Fehlerknoten (in endlicher Tiefe)
- S ist unerfüllbar gdw. ST(S) ist geschlossen
  - Ein Pfad ohne Fehlerknoten liefert ein Herbrand-Modell.
  - Zu jedem Modell gibt es ein Herbrand-Modell auf offenem Pfad.
- Satz (Herbrand): S ist unerfüllbar gdw. eine endl. Menge von Grund-Instanzen von Klauseln in S ist unerfüllbar.
  - Die Grundinstanzen ergeben sich aus den Elementen von HB(S) oberhalb der (höchsten) Fehlerknoten
  - **Korollar**: PL-Unerfüllbarkeit rein aussagenlogisch beweisbar
    - Bem.: Das war die ursprüngliche Motivation von Davis-Putnam !

Semantische Bäume und Resolution

- Ein **Inferenzknoten** ist ein Knoten im Semantischen Baum, dessen beide Kinder Fehlerknoten sind
- Satz: Jeder geschlossene Semantische Baum hat einen Inferenzknoten.
  - Ansonsten gäbe es einen unendlichen nicht geschlossenen Pfad und damit ein Modell
- Satz: Sei n ein Inferenzknoten mit Kindern n_1 und n_2. Seien G_1 und G_2 die beiden Grund-Fehlerklauseln an n_1 und n_2. Dann gibt es eine AL-Resolvente G zwischen G_1 und G_2 und diese schlägt an n fehl (oder oberhalb von n)
  - G_1 und G_2 enthalten komplementäre Literale, da sie nur wegen der komplementären Elemente von HB(S) an n_1 und n_2 fehlschlagen

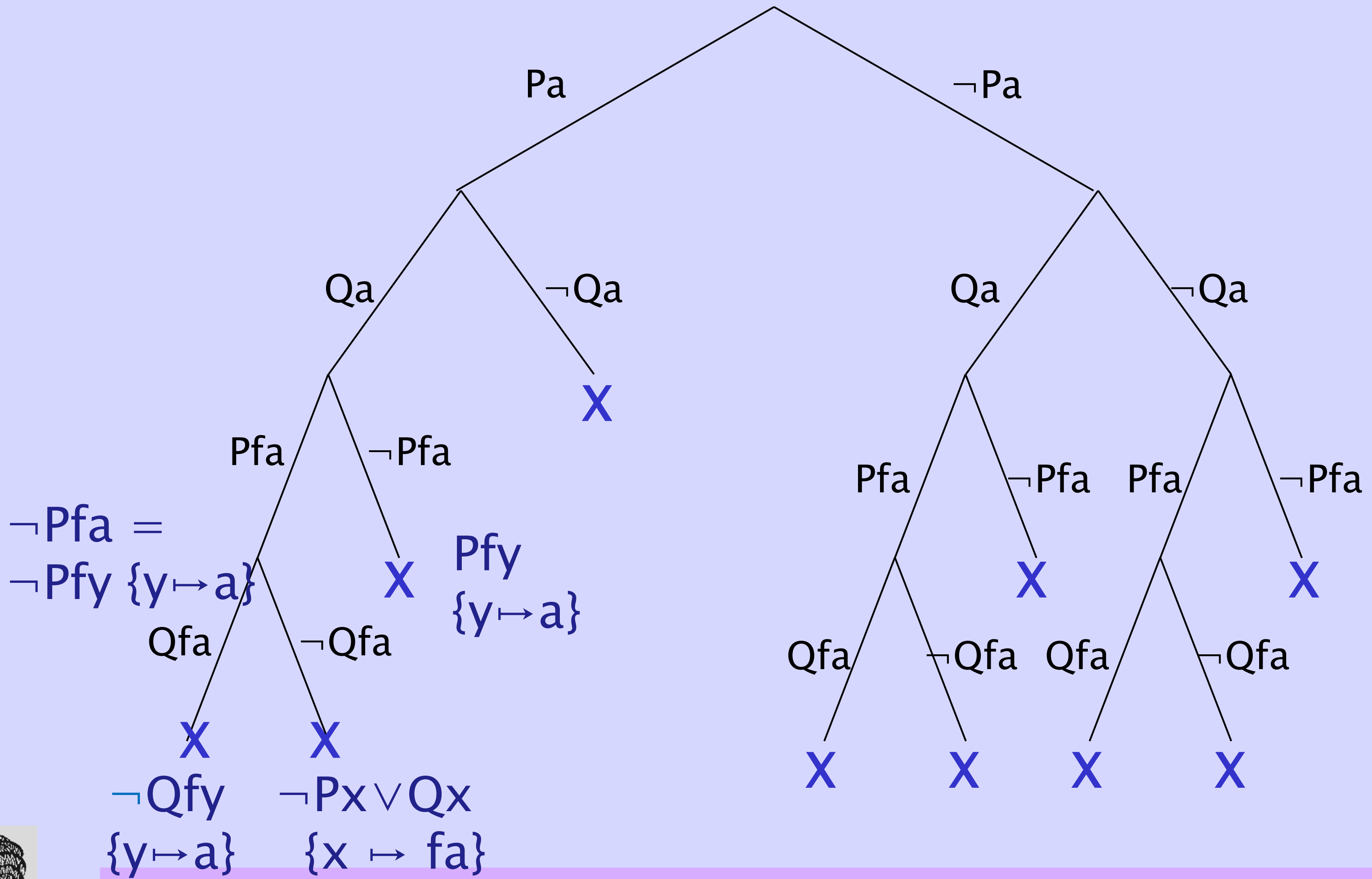
Semantische Bäume, Inferenzknoten

- $S=\{\neg Px \vee Qx,\ Pfy,\ \neg Qfy\}$



Semantische Bäume, Inferenzknoten

- $S=\{\neg Px \vee Qx,\ Pfy,\ \neg Qfy\}$



Widerlegungsvollständigkeit der Resolution

- Sei T ein geschlossener semantischer Baum von S; sei n ein Inferenzknoten mit Kindern n_1 und n_2 und Grund-Fehlerklauseln G_1 und G_2.
- Es gibt eine Grund-Resolvente (AL-Resolvente) G, die bei n (oder schon oberhalb n) falsifiziert wird.
- Nach endlich vielen AL-Resolutionsschritten werden alle Inferenzknoten zu Fehlerknoten, inklusive der Wurzel. Dort schlägt die leere Klausel fehl.
- Theorem (Herbrand): Eine PL-Formel S ist beweisbar gdw. eine Formel aus Grundinstanzen von S ist aussagenlogisch beweisbar

Widerlegungsvollständigkeit der Resolution

- Jetzt noch zu zeigen: die PL-Resolution mit Unifikation „deckt die Grundresolution vollständig ab". Nach endl. vielen PL-Resolutionen wird die leere Klausel erzeugt.
- **Lifting Lemma** (Robinson): Seien G_1 und G_2 Grundinstanzen der Klauseln K_1 und K_2, und sei G eine (Grund-)Resolvente von G_1 und G_2. Dann gibt es auch eine allgemeine Resolvente K von K_1 und K_2,

  und G ist eine Grundinstanz von K.

$$\begin{tikzcd}
K_1 \arrow[r] \arrow[d] & K \arrow[d] & K_2 \arrow[l] \arrow[d] \\
G_1 \arrow[r] & G & G_2 \arrow[l]
\end{tikzcd}$$

Widerlegungsvollständigkeit der Resolution

- Nach dem Satz von Herbrand kann man die Grund-Instanzen von S enumerieren (mittels Enumeration von HB(S)) und jeweils auf Erfüllbarkeit prüfen (z.B. mittels AL-Resolution oder mittels DPLL)
- Nach dem Lifting Lemma genügt es, statt vieler AL-Resolventen „summarisch" die PL-Resolventen direkt auf der PL-Formel zu bilden.
- Vollständigkeits-Theorem (Robinson): Falls S unerfüllbar ist, kann auf S mittels PL-Resolution in endlicher Zeit die leere Klausel abgeleitet werden.
  - Prädikatenlogik ist nur semi-entscheidbar!
    - Falls S erfüllbar ist werden i.A. unendlich viele Resolventen erzeugt