Prädikatenlogik

Einführung: Meistens reicht die Aussagenlogik nicht aus, um z.B. ein mathematisches, ein logisches oder ein Problem der natürlichen Sprache zu formalisieren. Die Sprache der Prädikatenlogik ist deutlich mächtiger und kann über Objekte einer 'Welt' sowie deren Eigenschaften und Beziehungen sprechen.
Damit enthält sie Bezeichnungen für:

- Objekte einer Grundmenge $\mathcal{U}$ (Konstanten, 0-stellige $Funktionen$) (notiert als $a, b, c, ..$ oder z.B. Ziffern)
- Funktionen auf den Objekten (z.B. Nachfolgerfunktion) (n-stellige $Funktionen$) $f : \mathcal{U}^n \to \mathcal{U}$ (notiert als $f, g, h, ...$)
- Aussagen (wie in der Aussagenlogik) (0-stellige $Prädikate$, notiert als $p, q, r, ..$)
- Eigenschaften von Objekten (z.B. gross, gelb,..) (Teilmengen von U) $P \subseteq \mathcal{U}$ (1-stellige Prädikate, notiert als $P, Q, R, ..$)
- Relationen zwischen Objekten (z.B. Grossvater von, kleiner als) (n-stellige $Prädikate$ (notiert als $P, Q, R, ..$) auf dem Universum $P \subseteq \mathcal{U}^n$

Prädikatenlogik – Beispiel

Beispiel aus der Zahlentheorie

- 0 ist eine gerade Zahl.
- Wenn x eine gerade Zahl ist, so ist $x + 1$ keine gerade Zahl.
- Wenn x keine gerade Zahl ist, so ist $x + 1$ eine gerade Zahl.

Zur Formalisierung benötigt: Neben den aus der Aussagenlogik bekannten Junktoren ($\rightarrow, ...$) auch Quantoren (z.B. $\forall$), Konstanten (z.B. $0, 1$), Variablen (z.B. x), Funktionssymbole (z.B. $s, +$), und Prädikatensymbole (z.B. $EVEN$)

- $EVEN(0)$
- $\forall x(EVEN(x) \rightarrow \neg EVEN(x + 1))$ oder $\forall x(EVEN(x) \rightarrow \neg EVEN(s(x)))$
- $\forall x(\neg EVEN(x) \rightarrow EVEN(x + 1))$ oder $\forall x(\neg EVEN(x) \rightarrow EVEN(s(x)))$

Sprache der Prädikatenlogik 1. Stufe (PL1)

- Alphabet („Symbole“)
- Signatur (Symbole mit Stelligkeiten)
- Wohlgeformte Ausdrücke:
  - Terme
  - Formeln

Alphabet $\mathcal{A}$: Abzählbar unendliche Menge von Objekten („Symbolen“), partitioniert in 6 disjunkte Untermengen:

1. Unendliche Menge $\mathcal{V}$ von Variablen
2. Unendliche Menge $\mathcal{F}$ von Funktionszeichen
3. Unendliche Menge $\mathcal{P}$ von Prädikatszeichen
4. Menge $\mathcal{C} = \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ von Junktoren oder Konnektiven (analog AL)
5. 2-elementige Menge $\mathcal{Q} = \{\forall, \exists\}$ von Quantoren: $\forall$ Allquantor, $\exists$ Existenzquantor
6. 3-elementige Menge $\{„(“, „)“, „,“\}$ von Hilfszeichen

Sprache der Prädikatenlogik 1. Stufe (2)

Signatur $\Sigma = \langle \mathcal{A}, \alpha \rangle$: $\mathcal{A}$ Alphabet und $\alpha : \mathcal{F} \cup \mathcal{P} \cup \mathcal{C} \cup \mathcal{Q} \longrightarrow \mathbb{N}_0$ ist eine Funktion (Stelligkeitsfunktion) derart, dass gilt:

1. Auf jedes $n \in \mathbb{N}_0$ werden unendlich viele Funktions- und Prädikatszeichen aus $\mathcal{A}$ abgebildet.
2. Negationszeichen hat Stelligkeit 1, alle anderen Junktoren und die Quantoren haben Stelligkeit 2.

Menge der Terme $\mathcal{T}$ (induktiv):

- Alle Variablen sind Terme.
- Falls f Funktionszeichen mit Stelligkeit $n \geq 0$ und $t_1, \ldots, t_n$ Terme, dann ist $f(t_1, \cdots, t_n)$ ein Term.

  Bei nullstelligen Funktionszeichen (Konstanten) schreiben wir auch a statt $a()$.

Sprache der Prädikatenlogik 1. Stufe (3)

Menge der Formeln $\mathcal{F}$ (induktiv):

1. Falls P Prädikatsszeichen mit Stelligkeit $n \geq 0$ und $t_1, \ldots, t_n$ Terme, dann ist $P(t_1, \cdots, t_n)$ eine (atomare) Formel.
   Statt $P()$ schreiben wir auch kurz P oder p (analog Aussagenlogik).

2. Falls F und G Formeln und $x \in \mathcal{V}$, dann sind auch:
   - $\neg F$, $(F \wedge G)$, $(F \vee G)$, $(F \rightarrow G)$, $(F \leftrightarrow G)$
   - und $\forall x F$, $\exists x F$

   Formeln.

Semantik der Prädikatenlogik – Interpretation

Eine Interpretation bestimmt die Bedeutung einer Formel durch die Festlegung der Trägermenge (Universum), durch Interpretation der Funktions- und Prädikatszeichen sowie durch die Belegung der freien (s.u.) Variablen einer Formel.

Interpretation $\mathcal{I} = \langle \mathcal{U}, \iota \rangle$:

- $\mathcal{U}$ Universum: Nichtleere Menge von Objekten
- ι Interpretationsfunktion:
  - jedem n-stelliges Funktionssymbol f wird eine Funktion $f' : \mathcal{U}^n \longrightarrow \mathcal{U}$ zugeordnet, d.h. $\iota(f) = f'$.
  - jedem n-stelliges Prädikatsymbol P wird eine n-stellige Relation $P' \subseteq \mathcal{U}^n$ zugeordnet, d.h. $\iota(P) = P'$.

Semantik: Variablen- und Termzuweisung

Variablenzuweisung: Funktion $\mathcal{A} : \mathcal{V} \longrightarrow \mathcal{U}$

(wird als Zwischenschritt benötigt um den Wert komplexer Terme und Formeln zu definieren)

Termzuweisung: Fortsetzung einer Interpretation $\langle \mathcal{U}, \iota \rangle$ und einer Variablenzuweisung $\mathcal{A}$ auf Terme

- $\iota^{\mathcal{A}}(x) = \mathcal{A}(x)$
- $\iota^{\mathcal{A}}(f(t_1, \ldots, t_n)) = \iota(f)(\iota^{\mathcal{A}}(t_1), \ldots, \iota^{\mathcal{A}}(t_n))$ wobei f n-stellig

Semantik: Formelzuweisung

Formelzuweisung: Fortsetzung einer Interpretation $\langle \mathcal{U}, \iota \rangle$ und einer Variablenzuweisung $\mathcal{A}$ auf Formeln

- $\iota^{\mathcal{A}}(P(t_1, \ldots, t_n)) = \mathbf{t}$, falls $\langle \iota^{\mathcal{A}}(t_1), \ldots, \iota^{\mathcal{A}}(t_n) \rangle \in \iota(P)$ und $\mathbf{f}$ sonst (wobei P n-stellig)
- $\iota^{\mathcal{A}}(\neg F) = \mathbf{t}$, falls $\iota^{\mathcal{A}}(F) = \mathbf{f}$ und $\mathbf{f}$ sonst,
- $\iota^{\mathcal{A}}(F \wedge G) = \mathbf{t}$, falls $\iota^{\mathcal{A}}(F) = \mathbf{t}$ und $\iota^{\mathcal{A}}(G) = \mathbf{t}$ und $\mathbf{f}$ sonst,
- $\iota^{\mathcal{A}}(F \vee G) = \iota^{\mathcal{A}}(\neg(\neg F \wedge \neg G))$,
- $\iota^{\mathcal{A}}(F \to G) = \iota^{\mathcal{A}}(\neg F \vee G)$,
- $\iota^{\mathcal{A}}(F \leftrightarrow G) = \iota^{\mathcal{A}}((F \to G) \wedge (G \to F))$,
- $\iota^{\mathcal{A}}(\forall x F) = \mathbf{t}$, falls $\iota^{\mathcal{A}^u_x}(F) = \mathbf{t}$ für alle $u \in \mathcal{U}$, und $\mathbf{f}$ sonst (wobei $\mathcal{A}^u_x(x) = u$ und sonst $\mathcal{A}^u_x = \mathcal{A}$)
- $\iota^{\mathcal{A}}(\exists x F) = \iota^{\mathcal{A}}(\neg \forall x \neg F)$

Umgangssprachlich: $\exists x \ldots$: es gibt mindestens ein x mit ...
$\forall x \ldots$: für alle x gilt ...

Unterausdrücke, Vorkommnisse, Quantorenbereiche

Unterausdrücke:

- Sei $W = W_1 \hat{} \cdots \hat{} W_n$ erzeugt nach Definition Term, Formel. Falls W_i wohlgeformt, dann unmittelbarer Unterausdruck von W.
- Unterausdruckrelation: transitive Hülle der unmittelbaren Unterausdruckrelation

Vorkommnisse oder Positionen:

- Ein Ausdruck E kann an verschiedenen Stellen in einem Ausdruck W vorkommen, die i.A. unterschieden werden müssen.
- Wir annotieren die Position von E in W durch ${}^kE$, wobei das erste Symbol von E das k-te in W ist.

Quantorenbereiche: Sei ${}^kQ$ Vorkommnis eines Quantors in einem Ausdruck W.

Das Vorkommnis ${}^kQxE$ heißt Bereich von ${}^kQ$ in W.

Freie und gebundene Vorkommnisse von Variablen

Freie und gebundene Variablenvorkommnisse:

- Ein Vorkommnis einer Variablen ${}^k x$ heisst gebunden in einem Ausdruck W, falls ${}^k x$ im Bereich eines Quantorvorkommnisses ${}^{k-l}Q$, dem unmittelbar x folgt; sonst heisst ${}^k x$ frei in W. Das ${}^{k-l}Q$ mit minimalem l ist das bindende Quantorvorkommnis von ${}^k x$ in W.
- Hinweis: Variablen können in der gleichen Formel sowohl gebunden als auch frei vorkommen. Beispiel: $P(x) \vee \forall x Q(x)$
- Formeln ohne freie Variablen heissen geschlossene Formeln oder Sätze.
  Hinweis: Der Wahrheitswert von Sätzen hängt damit nicht von der Variablenbelegung ab.
  Beispiel: Man vergleiche $P(x, y)$, $\exists x P(x, y)$ und $\forall y \exists x P(x, y)$.
- Gegeben eine Formel F mit freien Variablen $x_1, \ldots, x_n$:
  Allabschluß: $\forall x_1 \cdots \forall x_n F$
  Existenzabschluß: $\exists x_1 \cdots \exists x_n F$

Erfüllbarkeit, Modellbegriff

- Modell $\mathcal{M}$ einer Formel F:
  Interpretation $\langle \mathcal{U}, \iota \rangle$ mit $\iota^{\mathcal{A}}(F) = \mathbf{t}$ für alle Variablenzuweisungen $\mathcal{A}$
  (d.h. F entspricht dem Allabschluss von F)
- Modell $\mathcal{M}$ einer Formelmenge Δ:
  $\mathcal{M}$ Modell von Δ, falls $\mathcal{M}$ Modell für alle Formeln in Δ
- Formel(menge) Δ (un)erfüllbar: es gibt (k)ein Modell für Δ
- Formel(menge) Δ allgemeingültig: alle Interpretationen für Δ sind Modelle für Δ
- Logische Folgerungsbeziehung $\Delta \models F$ („F folgt logisch aus Δ“):
  Alle Modelle für Δ sind Modelle für F.
  Hinweis: $\Delta \models F$ genau dann, wenn $\Delta \cup \{\neg F\}$ unerfüllbar
- $\emptyset \models F$ (kurz $\models F$) bedeutet die Allgemeingültigkeit von F
- F und F' logisch äquivalent (kurz $F \equiv F'$), falls $F \models F'$ and $F' \models F$
- Es gilt für alle Formeln F und alle Variablen x: $F \equiv \forall x F$

Variablensubstitutionen

Substitution: Eine Funktion $\sigma : \mathcal{V} \longrightarrow \mathcal{T}$ derart, daß $\{x \mid \sigma(x) \neq x\}$ (genannt Trägermenge der Substitution) endlich. ($\mathcal{V}$ Menge der Variablen, $\mathcal{T}$ Menge der Terme)

Notation: Für $\sigma(x)$ schreiben wir kurz $x\sigma$.

Angabe einer Substitution durch Trägermenge (mit $t_i = x_i\sigma$):

$$\sigma = \{\langle x_1, t_1\rangle, \ldots, \langle x_n, t_n\rangle\}$$

oder kürzer:

$$\{x_1/t_1, \ldots, x_n/t_n\}$$

Erweiterung der Anwendung einer Substitution σ auf beliebige Ausdrücke:

$E\sigma$: Resultat der simultanen Ersetzung jedes freien Vorkommnisses einer Variablen x in E durch $x\sigma$

Eigenschaften von Substitutionen

Eine Substitution σ ist frei für einen Ausdruck E, falls für jedes freie Vorkommnis x einer Variablen in E gilt: alle Variablenvorkommnisse in $x\sigma$ sind frei in $E\sigma$.

Sätze: Sei σ frei für einen Ausdruck E.

- Jede Variable kommt gebunden an denselben Positionen in E und $E\sigma$ vor.
- $(E\sigma)\tau = E(\sigma\tau)$
- $\emptyset\sigma = \sigma = \sigma\emptyset$ ($\emptyset$ ist die Substitution mit leerer Trägermenge)
- $(\sigma\tau)\theta = \sigma(\tau\theta)$

Substitutionskorrektheit: $F \models F\sigma$, falls die Substitution σ frei für F ist.

Beispiel:

Sei $F := \exists x(P(x, y, z) \wedge \neg P(y, y, z))$, weiterhin $\sigma = \{y/z\}$ und $\tau = \{y/x\}$.

Dann gile: $F \models F\sigma$, aber $F \not\models F\tau$.

Wichtige Äquivalenzen in der Prädikatenlogik

$\neg\forall x F$	$\equiv$	$\exists x \neg F$	$\neg\exists x F$	$\equiv$	$\forall x \neg F$
$\forall x F \wedge \forall x G$	$\equiv$	$\forall x (F \wedge G)$	$\exists x F \vee \exists x G$	$\equiv$	$\exists x (F \vee G)$
$\forall x \forall y F$	$\equiv$	$\forall y \forall x F$	$\exists x \exists y F$	$\equiv$	$\exists y \exists x F$

Wegen letzterer Zeile schreiben wir oft für $Qx_1 Qx_2 \cdots Qx_n$ kürzer $Qx_1 x_2 \cdots x_n$ wobei $Q \in \{\forall, \exists\}$

Falls x nicht frei in G vorkommt:

$\forall x F \wedge G$	$\equiv$	$\forall x (F \wedge G)$	$\forall x F \vee G$	$\equiv$	$\forall x (F \vee G)$
$\exists x F \wedge G$	$\equiv$	$\exists x (F \wedge G)$	$\exists x F \vee G$	$\equiv$	$\exists x (F \vee G)$

Falls y nicht frei in F vorkommt:

$\forall x F$	$\equiv$	$\forall y F\{x/y\}$
$\exists x F$	$\equiv$	$\exists y F\{x/y\}$

Wichtige Normalformen in PL 1

- Die für die Aussagenlogik eingeführten Normalformen sind auch in der Prädikatenlogik einsetzbar
- Zusätzliches Problem in der Prädikatenlogik: Quantoren und Variablen
- Deshalb stellen wir zur Erzeugung einer Normalform in der Prädikatenlogik einige Schritte voran
- Der erste solche Schritt ist die Bereinigung der Formel. Eine Formel F heißt bereinigt, falls in F keine Variable sowohl frei als auch gebunden vorkommt und keine Variable mehr als einmal unmittelbar hinter einem Quantor steht.
- Beispiel: $F = \forall x \exists y P(x, f(y)) \wedge \forall y (Q(x, y) \vee R(x))$
  ergibt bereinigt: $F' = \forall u \exists y P(u, f(y)) \wedge \forall z (Q(x, z) \vee R(x))$
- Als nächstes bilden wir den Allabschluss der Formel, falls sie freie Variablen enthält: $F'' = \forall x \forall u \exists y P(u, f(y)) \wedge \forall z (Q(x, z) \vee R(x))$

Normalformen: Pränexform

Der nächste Schritt befasst sich mit dem Schieben der Quantoren nach außen.

- Eine Formel F' heißt pränex oder in Pränexform oder Pränexformel, falls sie die Form
$$F' := Q_1 x_1 Q_2 x_2 \cdots Q_n x_n F$$
hat, wobei $Q_i \in \{\forall, \exists\}$ und $x_1, \ldots, x_n$ die in F vorkommenden Variablen sind (F' ist also geschlossen) und F frei von Quantoren ist.
- In F' heisst F die Matrix der Formel, $Q_1 x_1 \cdots Q_n x_n$ das Präfix der Formel.
- Transformation einer Formel in Pränexform durch (wiederholte) Anwendung der aufgeführten Äquivalenzen (im worst-case exponentiell):
  - Zunächst Elimination der Junktoren $\leftrightarrow$ und $\rightarrow$ mittels:
    $(F \leftrightarrow G) \equiv ((F \rightarrow G) \wedge (G \rightarrow F))$ und $(F \rightarrow G) \equiv (\neg F \vee G)$
  - Dann Anwendung von $\neg\forall x F \equiv \exists x \neg F$ und $\neg\exists x F \equiv \forall x \neg F$ bzw.
    $(QxF \circ G) \equiv Qy(F[x/y] \circ G)$ und $(G \circ QxF) \equiv Qy(G \circ F[x/y])$
    für $\circ \in \{\vee, \wedge\}$ wobei y neu in F und G
    (falls Formel bereinigt, muss keine Umbenennung vorgenommen werden)

Skolemform

Eine prädikatenlogische Formel Φ heißt Skolemformel oder in *Skolem-Form*, falls Φ eine Pränexformel der Struktur $\forall x_1 \cdots \forall x_n F$ ist $(n \geq 0)$, wobei F quantorenfrei ist.

Wie können die Existenzquantoren beseitigt werden bei Beibehaltung der Pränexform und welche Eigenschaften können dabei erhalten werden?

- Beispiel: Man betrachte die Formel $\forall x \exists y S(x, y)$.
  y kann hier nicht frei gewählt werden, es hängt von der vorherigen Wahl des x ab.
- Es besteht also eine funktionale Abhängigkeit zwischen x und y
- Diese funktionale Abhängigkeit kann in der Prädikatenlogik durch Skolemfunktionen ausgedrückt werden.

Die Skolemisierung

Skolemisierung einer Formel F in bereinigter Pränexform:

Solange F einen Existenzquantor enthält, wiederhole den folgenden Schritt:

1. F habe die Form $\forall x_1 \cdots x_n \exists y G$, wobei G in Pränexform und $n \geq 0$.
2. Sei f_{sko} ein bisher nicht in F vorkommendes n-stelliges Funktionssymbol.
3. Bilde $F' = \forall x_1 \cdots \forall x_n G\{y/f_{\mathsf{sko}}(x_1, \ldots, x_n)\}$

Die Skolemisierung ist i.A. keine Äquivalenztransformation, aber sie erhält die Unerfüllbarkeit und die Erfüllbarkeit (die Menge der Modelle kann evtl. schrumpfen, es bleibt aber auf jeden Fall eines erhalten, falls die Eingabeformel erfüllbar war).

Beispiel:

Die Formel $\forall x \exists y \forall z \exists w(\neg P(a, w) \vee Q(f(x), y))$

kann skolemiziert werden zu:

$\forall x \forall z(\neg P(a, f_{\mathsf{sko}_2}(x, z)) \vee Q(f(x), f_{\mathsf{sko}_1}(x)))$

Die Klauselnormalform

- Wie in der Aussagenlogik kann die Matrix einer Skolemformel in Klauselform transformiert werden, d.h. in eine beliebigstellige Konjunktion von beliebigstelligen Disjunktionen von Literalen
- Diese Darstellung nennen wir Klauselnormalform
- Oft werden Skolemformeln, deren Matrix in Klauselform ist, auch einfach als eine Menge von Mengen von Literalen angegeben
- Da alle Variablen allquantifiziert sind, können wir das Quantorenpräfix weglassen
- Beispiel:

  $\forall xy((\neg P(x, f(y)) \vee R(y, f(x))) \wedge (P(x, g(x)) \vee R(x, f(x))))$

  wird zu

  $\{\{\neg P(x, f(y)), R(y, f(x))\}, \{Q(x, g(x)), R(x, f(x))\}\}$

Die Vollständigkeit der Prädikatenlogik 1. Stufe

Im Unterschied zur Aussagenlogik gibt es für die Prädikatenlogik erster Stufe keine Entscheidungsverfahren für den logischen Status einer Formelmenge, sondern lediglich Semi-Entscheidungsverfahren.

Genauer: Es gibt effektive Methoden, die logische Gültigkeit einer Formel bzw. die Unerfüllbarkeit einer Formel(menge) zu verifizieren.

Wir werden Semi-Entscheidungsverfahren für die Unerfüllbarkeit von Formelmengen betrachten.

Die Existenz solcher Verfahren ist auf die Kompaktheit der Prädikatenlogik zurückzuführen, d.h. auf folgenden

Satz: Jede unerfüllbare Menge von Skolemformeln (prädikatenlogischen Formeln) hat eine endliche unerfüllbare Teilmenge.

Herbrand-Interpretationen

Herbrand-Universum/Herbrand-Basis einer Formel(menge) S:

Menge aller variablenfreien Terme/Atomformeln über der Signatur Σ von S bzw. falls S keine Konstante enthält: über $\Sigma \cup \{a\}$ für eine beliebige Konstante a

Herbrand-Interpretation einer Formel(menge) S:

Sei $\mathcal{U}$ das Herbrand-Universum von S.
Eine Herbrand-Interpretation für S ist eine Interpretation $\langle \mathcal{U}, \iota \rangle$ mit:

1. ι bildet jede Konstante aus $\mathcal{U}$ auf sich selbst ab
2. ι bildet jedes Funktionszeichen f mit Stelligkeit $n \geq 0$, das in $\mathcal{U}$ vorkommt, auf die n-stellige Funktion ab, die jedes n-Tupel von Termen $\langle t_1, \ldots, t_n \rangle \in \mathcal{U}^n$ auf den Term $f(t_1, \ldots, t_n)$ abbildet.

Satz: Falls eine Skolemformel ein Modell hat, dann hat sie ein Herbrand-Modell.

Herbrand-Interpretationsbaum

Herbrand-Interpretationsbaum einer Formel(menge) S:

Sei $\prec$ eine strikte lineare Ordnung auf der Herbrand-Basis von S.
Der Herbrand-Interpretationsbaum für S und $\prec$ ist der folgende semantische Baum T:

- Alle Zweige in T sind gleich lang und haben die Mächtigkeit der Herbrand-Basis von S und
- die beiden Kanten, die in jedem Knoten der Tiefe i $(i \geq 0)$ entspringen, sind mit dem $(i+1)$-ten Atom in $\prec$ bzw. seiner Negation markiert.

Es gibt eine eindeutige Entsprechung (Bijektion) zwischen den Herbrand-Interpretationen für S und den Mengen von (Formeln auf den) Ästen in T.

Beweis der Kompaktheit

Sei T der Herbrand-Interpretationsbaum für S und eine totale Ordnung $\prec$ auf der Herbrand-Basis von S.

Ein Knoten N in T heisse falsch, falls es eine Formel $F_N \in S$ gibt derart, dass keine Herbrand-Interpretation eines Astes durch N die Formel F_N erfüllt.
Wir markieren jeden falschen Knoten N mit einer solchen Formel F_N.

Wegen der Unerfüllbarkeit von S muss jeder Ast von T einen falschen Knoten passieren.

Nun entfernen wir alle Unterbäume unterhalb falscher Knoten aus T.

- Der erhaltene Baum T' ist endlich (Lemma von König).
- Weiterhin gilt: Die endliche Menge S' der Formeln an falschen Knoten in T' ist unerfüllbar.

Herbrand-Instanzen

Herbrand-Instanz einer Formel aus einer Menge von Skolemformeln S:

Sei $\Phi \in S$ und F die Matrix von Φ.
Eine Formel F' ist eine Herbrand-Instanz von Φ *bzgl.* S, falls F' aus F erhalten werden kann, indem man alle Variablen in F durch Terme aus dem Herbrand-Universum von S ersetzt.

Hilfssatz: Eine Menge von Skolemformeln S ist unerfüllbar gdw die Menge der Herbrand-Instanzen der Formeln in S unerfüllbar ist.

Herbrand-Instanz-Theorem: Zu jeder unerfüllbaren Menge von Skolemformeln S gibt es eine endliche unerfüllbare Menge S' von Formeln derart, daß jede Formel in S' eine Herbrand-Instanz einer Formel in S bzgl. S ist.

Beweis: Durch Hilfssatz und Kompaktheitssatz

Direkte Herbrand-Verfahren

Erste Implementationen von Semi-Entscheidungsverfahren für die Prädikatenlogik von Gilmore bzw. von Davis und Putnam Anfang der 60er Jahre sind direkte algorithmische Umsetzungen des Herbrand-Instanz-Theorems.

Verfahren bestehen aus zwei weitgehend unabhängigen Unterprozeduren:

1. Oberprozedur zählt endliche Mengen von Herbrand-Instanzen der gegebenen Eingabeformeln auf
2. Unterprozedur entscheidet den Erfüllbarkeitsstatus der jeweiligen endlichen Menge aussagenlogisch

Nachteil der 2-stufigen Verfahren am Beispiel:

$$\begin{array}{ll}\{ & \forall x_1 \cdots \forall x_n \; P(x_1, \ldots, x_n, a_1, \ldots, a_n), \\ & \forall y_1 \cdots \forall y_n \neg P(a_1, \ldots, a_n, y_1, \ldots, y_n) \quad \}\end{array}$$

Jede Klausel in S hat n^n Herbrand-Instanzen, aber nur eine (per Klausel) trägt zur Widerlegung bei.

Unifikation

Falls Mengen von Grundinstanzen blind aufgezählt werden, d.h. ohne die Strukturzusammenhänge der Formeln in S zu betrachten, ist die Wahrscheinlichkeit sehr gering, die richtigen Instanzenmengen zu finden.

Unifikator (einer Menge S) von quantorenfreien Ausdrücken $L_1, \ldots, L_n$: Substitution σ derart, daß $L_1\sigma = \cdots = L_n\sigma$, d.h. $S\sigma$ ist eine Einermenge. S bzw. $L_1, \ldots, L_n$ heißen dann unifizierbar.

Beispiele:

1. $\{x_1/a_1, \ldots, x_n/a_n\}$ ist ein Unifikator für $P(x_1, \ldots, x_n)$ und $P(a_1, \ldots, a_n)$

2. $\{y/a, x/h(a), z/a\}$ ist ein Unifikator für $P(a, f(x), g(x))$ und $P(y, f(h(z)), g(h(y)))$

3. $\{x/f(g(a)), y/g(a)\}$ ist ein Unifikator für x und $f(y)$

Spezielle Unifikatoren

Allgemeinster Unifikator (einer Menge S) von quantorenfreien Ausdrücken $L_1, \ldots, L_n$: Unifikator σ von S derart, daß es für jeden Unifikator τ von S eine Substitution θ gibt mit $\tau = \sigma\theta$.

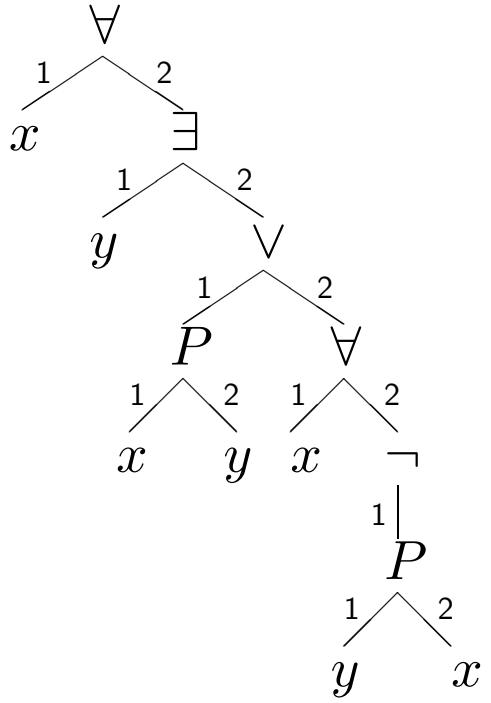
Minimaler Unifikator (einer Menge S) von quantorenfreien Ausdrücken $L_1, \ldots, L_n$: Unifikator von S, dessen Trägermenge kleinste Kardinalität hat.

Satz: Jeder minimale Unifikator für eine Menge S ist ein allgemeinster Unifikator für S.

Wie findet man einen minimalen/allgemeinsten Unifikator?

Positionen im Symbolbaum

Symbolbaum der Formel $\forall x \exists y (P(x,y) \vee \forall x \neg P(y,x))$



Position im Symbolbaum: Kantenmarkierungen

Positionen (Vorkommnisse) von x in obiger Formel: ${}^{1}x, {}^{2,2,1,1}x, {}^{2,2,2,1}x, {}^{2,2,2,2,1,2}x$

Differenzmengen

Differenzmenge für zwei quantorenfreie Ausdrücke K und L: Seien

$$^{s_1,\ldots,s_n}K_n \text{ und } \ ^{s_1,\ldots,s_n}L_n$$

zwei Vorkommnisse von verschiedenen Unterausdrücken an derselben Position in K und L derart, daß

$$^{s_1,\ldots,s_i}K_i = {}^{s_1,\ldots,s_i}L_i$$

für alle $1 \leq i < n$.

Dann ist $\{K_n, L_n\}$ eine Differenzmenge für K und L.

Beispiel: Die Literale $P(a, f(x), g(x))$ und $P(y, f(h(z)), g(h(y)))$ haben die folgenden Differenzmengen: $\{a, y\}$, $\{x, h(z)\}$, $\{x, h(y)\}$.

Satz: Falls zwei quantorenfreie Ausdrücke K und L unifizierbar sind, dann enthält jede Differenzmenge von K und L eine Variable und einen Term, in dem die Variable nicht vorkommt.

Unifikationsalgorithmus

Unifikationsalgorithmus: Gegeben zwei quantorenfreie Ausdrücke K und L, setze am Anfang $\sigma_0 = \emptyset$ und $i = 0$.

1. Falls $K\sigma_i = L\sigma_i$, gebe σ_i als berechneten Unifikator von K und L aus; sonst wähle eine Differenzmenge D_i für $K\sigma_i$ und $L\sigma_i$ und gehe zu 2.

2. Wähle eine Variable x aus D_i und falls x nicht in dem anderen Ausdruck E in D_i vorkommt, setze $\sigma_{i+1} = \sigma_i\{x/E\}$, inkrementiere i um 1 und gehe zu 1; ansonsten gebe aus „nicht unifizierbar“.

Unifikationstheorem: Jeder berechnete Unifikator zweier quantorenfreier Ausdrücke ist ein minimaler Unifikator der Ausdrücke und umgekehrt.

Zur Komplexität der Unifikation

Beispiel: $P(x_1, x_2, \ldots, x_n)$ und
$P(f(x_0, x_0), f(x_1, x_1), \ldots, f(x_{n-1}, x_{n-1}))$

Für jeden Unifikator σ gilt: $x_n\sigma$ enthält mehr als 2^n Symbolvorkommnisse.

Voraussetzungen polynomieller Unifikation:

1. Repräsentation von Ausdrücken als gerichtete azyklische Graphen (reduziert Platzkomplexität von exponentiell auf linear).

2. „Merken“ bereits unifizierter Paare und in welchen Ausdrücken eine Variable nicht vorkommt (für Vorkommnistest) (reduziert exponentielle Zeitkomplexität von exponentiell auf quadratisch oder sogar linear).

Bekannte polynomielle Unifikationsalgorithmen gibt es von Huet, Martelli/Montanari, Paterson/Wegman, Corbin-Bidoit, Jaffar, etc.

Prädikatenlogische Resolution

Binäre Resolutionsregel: Seien $c_1 \cup \{L\}$ und $c_2 \cup \{\neg K\}$ zwei Klauseln:

$$\frac{c_1 \cup \{L\} \qquad\qquad c_2 \cup \{\neg K\}}{(c_1\sigma \setminus \{L\sigma\}) \ \cup \ (c_2\sigma \setminus \{\neg K\sigma\})}$$

wobei σ ein minimaler Unifikator der Literale L und K ist

Probleme:

- Betrachte zwei Klauseln: $\{P(x)\}$ und $\{\neg P(f(x))\}$
- Um möglichst allgemeine Resolventen zu erhalten: Umbenennen der Variablen
- Weiteres Problem: Klauseln $\{P(x), P(y)\}$ und $\{\neg P(x), \neg P(y)\}$

Faktorisierungsregel: Sei $c_1 \cup c_2$ eine Klausel:

$$\frac{c_1 \cup c_2}{(c_1 \cup c_2)\sigma}$$

wobei σ ein minimaler Unifikator von c_2 ist

Resolution (nach Robinson, 1965)

Resolutionsregel (Robinson): Seien $c_1 \cup c_3$ und $c_2 \cup c_4$ zwei variablenfremde Klauseln (evtl. ist Umbenennen erforderlich) und $\overline{c_4}$ die Menge der Komplemente der Literale in c_4:

$$\frac{c_1 \cup c_3 \qquad\qquad c_2 \cup c_4}{(c_1\sigma \setminus c_3\sigma) \;\cup\; (c_2\sigma \setminus c_4\sigma)}$$

wobei σ ein minimaler Unifikator von $c_3 \cup \overline{c_4}$ ist

Robinsons Verfahren integriert Binäre Resolution und Faktorisierung

Satz: Robinsons Resolutionsregel ist vollständig für Prädikatenlogische Klauselmengen (ebenso das System aus binärer Resolution (mit Umbenennen) und Faktorisierung).

Resolutionsbeweiser

Eigenschaften der Resolution

- Wegen flacher Formelstruktur starke Methoden der Suchraumreduktion
- Ermöglicht Integration geordneter Gleichheitsbehandlung (geordnete Paramodulation)
- Effiziente Implementierungen: Vampire, E, SPASS, Gandalf
- Forschungsrichtungen: Datenstrukturen und Suchheuristiken
- Problem: Zielorientierung

Tableaukalkül [Smullyan 1968]

Uniforme Dekompositionsregeln für die Quantoren:

Universell	
γ	γ-Subformelmenge
$\forall x F$	$F\{x/t\}$
$\neg\exists x F$	$\neg F\{x/t\}$

wobei t beliebiger Grundterm
$\gamma(t)$: γ-Subformel von γ

Existentiell	
δ	δ-Subformelmenge
$\exists x F$	$F\{x/c\}$
$\neg\forall x F$	$\neg F\{x/c\}$

wobei c beliebige *neue* Konstante
$\delta(c)$: δ-Subformel von δ

Tableau: Mit geschlossenen Formeln markierter Baum (Satztableau)

Tableaukalkül: System von Regeln zum Aufbau eines Satztableaus

Tableaukalkül für geschlossene Formeln (Sätze)

Regeln des Tableaukalküls für Sätze:

α-Regel: $$\frac{\alpha}{\alpha_i}$$

β-Regel: $$\frac{\beta}{\beta_1 \mid \cdots \mid \beta_n}$$

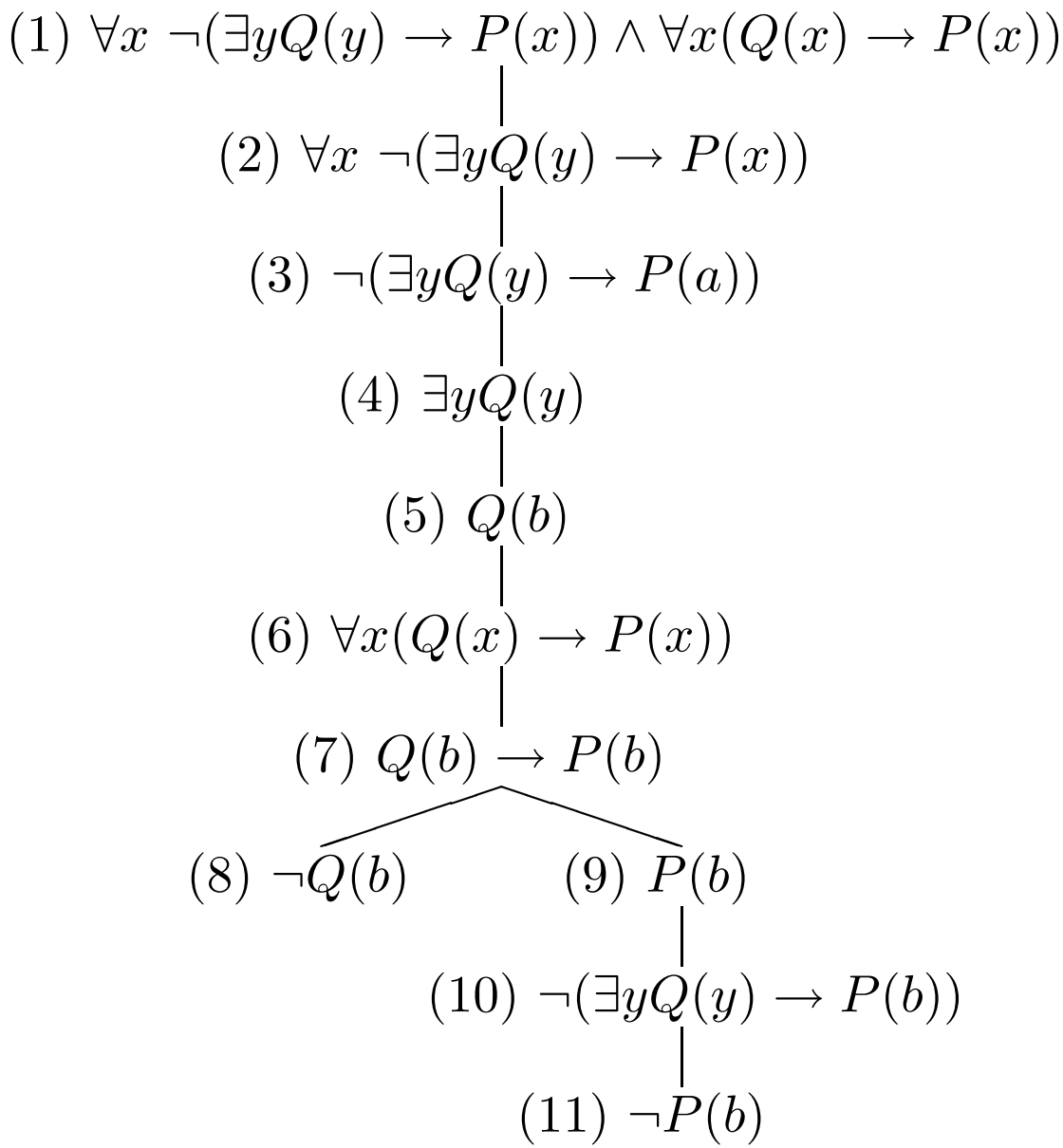
γ-Regel: $$\frac{\gamma}{\gamma(t)}$$

(**Signaturrestriktion:** t darf nur Symbole des Astes enthalten)

δ-Regel: $$\frac{\delta}{\delta(c)}$$

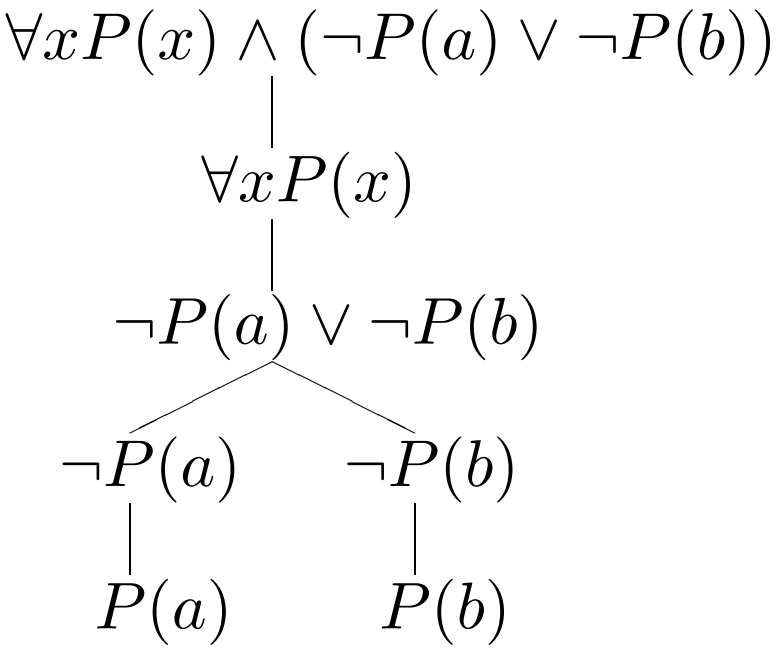
Tableau geschlossen: Jeder Ast enthält eine Formel und ihre Negation

Smullyans Tableaukalkül (Beispiel)



Probleme mit der γ-Regel

Beobachtung: Es gibt unendlich viele γ-Unterformeln.



Problem: Wie erreicht man Fairneß der γ-Regel?

Lösung: Zähle systematisch alle $\gamma(t)$ auf. Definiere dazu eine bijektive Abbildung $\pi : \mathbb{N}_0 \longrightarrow \mathcal{G}$ ($\mathcal{G}$ Menge aller Grundterme).
Ein Grundterm s ist kleiner als t modulo π, falls $\pi^{-1}(s) < \pi^{-1}(t)$.

Smullyans Tableaukalkül (Eigenschaften)

Eigenschaften des Smullyanschen Tableaukalküls:

- Geeignet für beweistheoretische, didaktische Zwecke
- Nicht geeignet als Basiskalkül für automatische Beweiser
- Problem: γ-Regel erfordert Raten der „richtigen“ Grundinstanzen
- Beweisfindung durch sog. systematisches Verfahren: faire („blinde“) Aufzählung aller Instanzen auf allen Ästen
- Modellgenerierung: saturierter offener Ast charakterisiert ein Modell
- Mit Signaturrestriktion: Entscheidungsverfahren für Bernays-Schönfinkel-Klasse

Freier-Variablen-Tableaukalkül [Fitting 1990]

Tableau: Mit beliebigen Formeln markierter Baum

Inferenzregeln für Formeltableaux:

α-Regel: $$\frac{\alpha}{\alpha_i}$$

β-Regel: $$\frac{\beta}{\beta_1 \mid \cdots \mid \beta_n}$$

γ'-Regel: $$\frac{\gamma}{\gamma(x)} \quad x \text{ neu}$$

δ'-Regel: $$\frac{\delta}{\delta(f(x_1, \ldots, x_n))}$$

f neu, $x_1, \ldots, x_n$ freie Variablen in δ (entspricht Skolemisierung)

Instantiierungsregel: $T \Rightarrow T\sigma$, falls zwei Formeln/Literale F und $\neg F'$ auf einem Ast und $F\sigma = F'\sigma$ (entspricht Unifikation)

Freier-Variablen-Tableaukalkül (2)

Vollständigkeit: Konstruktiv durch schrittweise Simulation des Smullyanschen Kalküls $\rightsquigarrow$ Beweislängen bleiben erhalten

Problem:

- Kalkül ist (beweis)konfluent (sackgassenfrei), aber destruktiv
- Wie findet man Beweise (systematisches Verfahren)?

Fitting-Verfahren: 2-stufig

1. Faire Beschränkung der γ-Regel-Anwendungen (stufenweise erhöht)
2. Aufzählung aller Tableaus für die jeweilige Stufe (Σ_2^P-vollständig)

Schwächen des Verfahrens:

- Kein kalkülinherentes Terminierungskriterium, keine Modellgenerierung
- Verfahren suboptimal für automatisches Beweisen

Gegenwärtig erfolgreiche Tableau-Paradigmen

Alternative: Destruktive vs. nichtdestruktive Methoden

Destruktive Methoden

- Konsequenz: Wenn schon destruktiv, keine Modellgenerierung etc., dann auch Konfluenz überflüssig
- Prinzip: Maximale Einschränkung des Kalküls (d.h. Einschränkung der Beweisstruktur)
- Erfolgreich: Konnektionstableaus (entspricht Verallgemeinerung der Modellelimination [Loveland 1968])

Nichtdestruktive Methoden

- Prinzip: Beibehaltung des Smullyanschen Prinzips der Saturierung von Tableauästen und damit im negativen Terminierungsfall auch Modellgenerierung
- Problem: Wie bettet man die Unifikation effizient ein?
- Erfolgreich: Diskonnektionstableaus (relativ neuer Ansatz [Billon 1996])

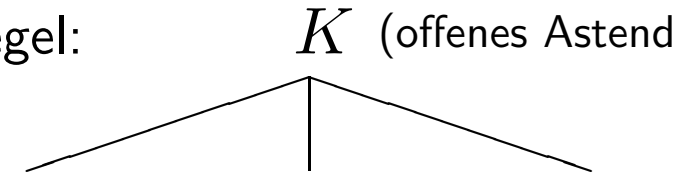
Konnektionstableaus [Letz 1992]

Inferenzregeln für Konnektionstableaus:

Gegeben sei eine Menge S prädikatenlogischer Klauseln

1. Startregel $=$ β-Regel (einmalig)

2. sog. Extensionsregel:



und modifiziere T zu $T\sigma$, falls es ein L_i gibt mit $L_i\sigma = {\sim}K\sigma$

(entspricht einer Kombination von Regeln aus Fittings System)

3. sog. Reduktionsregel: Modifiziere T zu $T\sigma$, falls ein Blattliteral K einen Vorgänger L hat und ${\sim}L\sigma = K\sigma$

Konnektionstableaus (2)

Vollständigkeit: Der Konnektionstableaukalkül ist vollständig für jede „relevante“ Startklausel.

Eigenschaften des Konnektionstableaukalküls:

- Nichtkonfluent und destruktiv
- Sehr hohe Inferenzraten durch Prolog-Technologie
- Grosses Potential zur Suchraumreduktion
- Zielorientierung mit allen wichtigen Verfeinerungen verträglich
- Pfadsaturierung und Modellgenerierung prinzipiell unmöglich
- Integration von Gleichheitsbehandlung schwierig

Konnektionstableaus (Beispiel)

Beispiel: 4 Klauseln

$(1)\ \forall xy(\leq(x,y) \vee \leq(y,x))$
$(2)\ \neg \in (a,m)$
$(3)\ \forall xy(\neg \leq(x,y) \vee \in (y,m)$
$(4)\ \leq(a,b)$

$$\begin{array}{ccc} \neg \leq(x,y) & | & \in (y,m) \\ \leq(x',y') \quad | \quad \leq(y',x') & & \neg \in (b,m) \\ x'/x, y'/y \qquad x/y & & y/b \end{array}$$

Konnektionstableaus (2)

Beispiel: 4 Klauseln

(1) $\forall xy(\leq(x,y) \vee \leq(y,x))$
(2) $\neg \in (a,m)$
(3) $\forall xy(\neg \leq(x,y) \vee \in(y,m)$
(4) $\leq(a,b)$

$\neg \leq(x,y)$ | $\in(y,m)$
$\leq(a,b)$ | Sackgasse
$x/a, y/b$

Diskonnektionstableaus [Billon 1996]

Ein Tableaupfad P is **grund-geschlossen** bzgl. einer Konstante a, falls $P\sigma$ geschlossen ist, wobei σ alle freien Variablen auf a abbildet.

Inferenzregeln für Diskonnektionstableaus (a beliebige feste Konstante):

S Eingabeklauselmenge, P aktueller Pfad, P_S Klauselmenge auf P,

C Konnektion zwischen Klausel $c, d \in P_S \cup S$ mit Unifikator σ

1. Pfaderweiterungsregel: $$\frac{P}{L_1 \mid \cdots \mid L_n}$$

wobei $L_1 \vee \cdots \vee L_n$ eine variablendisjunkte Variante von $c\sigma$ ist;
anschliessend Streichen der Konnektion für c

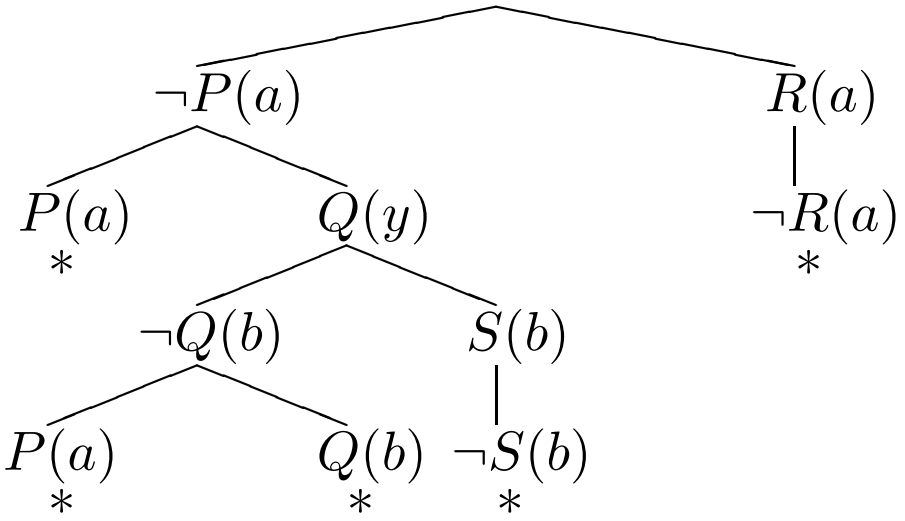
2. Pfadabschluss: falls P grund-geschlossen bzgl. a.

Mögliche Einschränkung: Variantenfreiheit, d.h. keine zwei Varianten einer Klausel auf einem Ast

Diskonnektionstableaus (2)

Beispiel:

$$
\begin{array}{rcl}
P(x) & \vee & Q(y) \\
\neg P(x) & \vee & R(x) \\
\neg Q(x) & \vee & S(x) \\
\neg R(a) & & \\
\neg S(b) & &
\end{array}
$$



Diskonnektionstableaus (3)

Eigenschaften des Diskonnektionskalküls:

- Konfluent und nichtdestruktiv
- Ermöglicht Pfadsaturierung wie in Smullyans Tableaukalkül
- Elegante und kompakte Charakterisierung von Modellen
- Entscheidet Bernays-Schönfinkel-Klasse
- Weitere Einschränkungen möglich: Auswahl eines initialen Pfades
- Ermöglicht Integration geordneter Gleichheitsbehandlung
- Problem: Zielorientierung