4.5 Umformungen prädikatenlogischer Ausdrücke

In der Mathematik findet man häufig Aussagen wie z.B. diese:

$$\forall \varepsilon\, \exists \delta\, \forall x \; (|x-a| < \delta \Rightarrow |f(x) - f(a)| < \varepsilon)$$

(Stetigkeit von f an der Stelle a)

Es fällt auf, daß die Quantoren $\forall \exists \forall$ am Anfang der Aussage stehen, während der Kern der Aussage quantorenfrei ist.
Wir zeigen jetzt, wie man beliebige Ausdrücke in $PL(\exists, \forall)$ in diese übersichtliche Gestalt bringt.

Definition

(i) Ein Ausdruck α heißt quantorenfrei, wenn er keine Quantoren $\bigvee_x$, $\bigwedge_x$ enthält.
(quantorenfrei sind also alle Primformeln und alle Ausdrücke, die man mit $\neg, \vee, \wedge$ daraus zusammensetzen kann).

(ii) Ein Präfix Π ist eine Folge von Quantoren (also das leere Wort Λ und mit Π auch $\bigvee_x \Pi$ und $\bigwedge_x \Pi$).

(iii) Ein Ausdruck α heißt pränex, wenn es ein Präfix Π und einen quantorenfreien Ausdruck β gibt, so daß $\alpha \equiv \Pi \beta$.

Beispiel $\underbrace{\bigwedge_v \bigvee_w \bigwedge_x}_{\text{Präfix}} \underbrace{(\, K\, dxaw \rightarrow K\, dfxfav\,)}_{\text{quantorenfrei}}$ ist pränex

Wenn man diesen Ausdruck über

$\mathcal{A} = (\mathbb{R}, (d^{\mathcal{A}}, f^{\mathcal{A}}), <, 2.9)$ interpretiert, wobei

$d^{\mathcal{A}}(x, y) = |x - y|$ und $f^{\mathcal{A}}$ irgendeine 1-stellige Funktion,

so besagt er gerade, daß $f^{\mathcal{A}}$ an der Stelle 2.9 stetig ist.

$\forall x \forall y$
$f(\alpha) = f(\beta)$
$\alpha \models \beta \models \alpha$

Wir wollen zu gegebenem Ausdruck α einen Ausdruck β konstruieren, der folgende Eigenschaften aufweist:

(i) $\alpha \sim \beta$ (siehe S. 185)

(ii) $f(\alpha) = f(\beta)$ und $\mathrm{const}(\alpha) = \mathrm{const}(\beta)$

hierfür schreiben wir $\alpha \approx \beta$

d.h. nur bei gebundenen Variablen kann es Unterschiede geben

alle Individuen-, Funktions- und Prädikaten-konstanten in α

(iii) β ist pränex

Wie schon bei den Normalformen des Aussagenkalküls (S. 153) müssen zur Gewinnung von β aus α Negationszeichen nach innen gebracht werden, hier also über Quantoren hinweg. Diese "kippen dabei um":

alle Hunde sind schwarz

alle Menschen sind grün

<u>Beh 1</u> $\neg \Pi \alpha \approx \Pi^{-1} \neg \alpha$, wobei Π^{-1} rekursiv wie folgt definiert:

$$\Lambda^{-1} :\equiv \Lambda$$

$$\Big(\bigvee_x \Pi\Big)^{-1} :\equiv \bigwedge_x \Pi^{-1}$$

$$\Big(\bigwedge_x \Pi\Big)^{-1} :\equiv \bigvee_x \Pi^{-1}$$

leeres Wort: nichts zu tun!

<u>Bew 1</u> durch Induktion über die Länge $|\Pi|$ von Π:

<u>Induktionsanfang</u> $|\Pi| = 0 \Rightarrow \Pi = \Lambda \Rightarrow \Pi^{-1} = \Lambda$ und wir sind fertig.

Induktionsschritt

1. Fall:

$$\neg\left(\bigwedge_x \Pi\alpha\right) \equiv \neg\neg\bigvee_x \neg\Pi\alpha \approx \bigvee_x \neg\Pi\alpha$$

$$\underset{\text{Induktions-Vor.}}{\approx} \bigvee_x \Pi^{-1}\neg\alpha \equiv \left(\bigwedge_x \Pi\right)^{-1}\neg\alpha$$

2. Fall:

$$\neg\left(\bigvee_x \Pi\alpha\right) \approx \neg\neg\bigwedge_x \neg\Pi\alpha \approx \bigwedge_x \neg\Pi\alpha$$

$$\underset{\text{Induktions-Vor.}}{\approx} \bigwedge_x \Pi^{-1}\neg\alpha \equiv \left(\bigvee_x \Pi\right)^{-1}\neg\alpha .$$

Π ist kürzer als $\bigwedge_x \Pi$ oder $\bigvee_x \Pi$

$\boxed{\text{Beh 1}}$

Beispiele

$$\neg\bigvee_x\bigvee_y Pxy \approx \bigwedge_x\bigwedge_y \neg Pxy$$

$$\neg\bigwedge_x\bigvee_y \, myy = x \approx \bigvee_x\bigwedge_y \, myy \neq x$$

$$\neg\bigwedge_v\bigvee_w\bigwedge_x (Kdxaw \to Kdfxfav) \approx \bigvee_v\bigwedge_w\bigvee_x \neg(Kdxaw \to Kdfxfav)$$

$$\equiv \bigvee_v\bigwedge_w\bigvee_x \neg(\neg Kdxaw \vee Kdfxfav)$$

$$\approx \bigvee_v\bigwedge_w\bigvee_x (Kdxaw \wedge \neg Kdfxfav)$$

diese Formel ist schon pränex, aber wir arbeiten im Kern noch weiter

bei der Interpretation von S. 186.1: f^A ist nicht stetig in 2.9.

Als nächstes zeigen wir, wie man die $\vee$-Verknüpfung von zwei pränexen Ausdrücken pränex macht.

In manchen Fällen ist das ganz einfach:

$$\bigwedge_x mxx = c \ \vee \ z=w \quad \approx \quad \bigwedge_x (mxx = c \ \vee \ \underbrace{z=w})$$

denn x kommt hier nicht vor!

$$\bigwedge_x mxx = c \ \vee \ \bigvee_x fx=y \quad \approx \quad \bigwedge_x (mxx = c \ \vee \ \underbrace{\bigvee_x fx=y})$$

denn x kommt hier nicht frei vor

Allgemein lassen sich diese einfachen Fälle so beschreiben:

Beh 2 $\quad x \notin f(\beta) \Rightarrow$

$$\bigvee_x \alpha \vee \beta \approx \bigvee_x (\alpha \vee \beta)$$
$$\bigwedge_x \alpha \vee \beta \approx \bigwedge_x (\alpha \vee \beta)$$

Bew 2 Sei $\mathcal{J}$ eine Interpretation.

$$\mathcal{J}(\bigvee_x \alpha \vee \beta) = w \iff \mathcal{J}(\bigvee_x \alpha) = w \text{ oder } \mathcal{J}(\beta) = w$$

$$\iff (\text{es gibt } \varphi \in A_s: \ \mathcal{J}_x^{\varphi}(\alpha) = w) \text{ oder } (\mathcal{J}(\beta) = w)$$

$$\iff \text{es gibt } \varphi \in A_s: (\mathcal{J}_x^{\varphi}(\alpha) = w \text{ oder } \mathcal{J}(\beta) = w)$$

$\parallel$ da $x \notin f(\beta)$: $\mathcal{J}_x^{\varphi}(\beta)$

$$\iff \text{es gibt } \varphi \in A_s \ (\mathcal{J}_x^{\varphi}(\alpha) = w \text{ oder } \mathcal{J}_x^{\varphi}(\beta) = w)$$

$$\iff \mathcal{J}(\bigvee_x (\alpha \vee \beta)) = w.$$

Der zweite Teil wird analog bewiesen. $\boxed{\text{Beh 2}}$

Es gibt aber auch schwierigere Fälle:

$\bigwedge_x x \neq x \ \vee \ x = c$ ist erfüllbar durch $\mathcal{Y}(x) := c^{\mathcal{A}}$

aber $\bigwedge_x (x \neq x \ \vee \ x = c)$ ist nur in solchen $\mathcal{A}$ mit $|A_S| = 1$ erfüllbar

Die Ausdrücke sind also nicht äquivalent.

Abhilfe: das x unter dem Quantor durch eine "neue" Variable y ersetzen –
dann geht es wegen $y \notin f(x=c)$ problemlos nach Beh 2:
(man könnte hier auch das zweite x ersetzen, durch Subst)

$$\bigwedge_x \underbrace{x \neq x}_{\alpha} \ \vee \ \underbrace{x = c}_{\beta} \approx \bigwedge_y y \neq y \ \vee \ x = c$$

$$\underset{\text{Beh 2}}{\approx} \bigwedge_y (y \neq y \ \vee \ x = c) \quad , \text{ da } y \notin f(\beta)$$

Um die quantifizierte Variable x durch y zu ersetzen, können wir nicht direkt Subst anwenden, denn x kommt ja nicht frei vor (vgl. S. 169).
Wir müssen in dem Ausdruck α hinter dem Quantor $\bigwedge_x$ das x durch y ersetzen, dann $\bigvee_y$ davorschreiben.

Beh 3 $\quad y \notin \mathcal{V}(\alpha) \Longrightarrow$ es gibt β mit Subst(α, x, y, β)
und Subst(β, y, x, α)
und $\bigvee_x \alpha \approx \bigvee_y \beta$

"gebundene Umbenennung"

$\longrightarrow$ Beweis liefert Algorithmus.

Bew 3 Sollte $y \equiv x$ sein, ist $\alpha \equiv \beta$, und die Behauptung ist trivial. Sei also $y \not\equiv x$.

Dann kann man x in α durch y ersetzen und erhält als Ergebnis einen Ausdruck β (evtl. α, falls $x \notin f(\alpha)$). Umgekehrt kann man diese Vorkommen von y – andere gibt es nicht wegen $y \notin v(\alpha)$ – in β wieder zu x machen und bekommt α zurück.

Die Äquivalenz wird so bewiesen:

1.	$\alpha \to \bigvee_x \alpha$	P1
2.	$\beta \to \bigvee_x \alpha$	S: 1
3.	$\bigvee_y \beta \to \bigvee_x \alpha$	P2: 2, denn $y \notin f(\bigvee_x \alpha)$
4.	$\beta \to \bigvee_y \beta$	P1
5.	$\alpha \to \bigvee_y \beta$	S: 4
6.	$\bigvee_x \alpha \to \bigvee_y \beta$	P2: 5, denn $x \notin f(\bigvee_y \beta)$

alle freien Vorkommen von x in β wurden durch y ersetzt!

Mit 3. und 6. folgt

$$\bigvee_x \alpha \approx \bigvee_y \beta.$$

aus dem Korrektheitssatz

$\boxed{\text{Beh 3}}$

Jetzt können wir allgemeine $\vee$-Ausdrücke pränex machen:

Beh 4 $\forall \Pi_1 \alpha_1, \Pi_2 \alpha_2$ pränex $\exists \Pi, \beta$ quantorenfrei

$$\Pi_1 \alpha_1 \vee \Pi_2 \alpha_2 \approx \Pi \beta \text{ pränex.}$$

Bew 4 Induktion über $|\Pi_1| + |\Pi_2|$:

das hatten wir noch nicht, oder?

Induktionsanfang $|\Pi_1| + |\Pi_2| = 0 \Rightarrow \Pi_1, \Pi_2$ sind leer

$\Rightarrow \Pi :\equiv \Lambda$, $\beta :\equiv \alpha_1 \vee \alpha_2$ erfüllen die Behauptung.

Induktionsschritt $|\Pi_1| + |\Pi_2| \geq 1$.

Ohne Einschränkung sei $|\Pi_1| \geq 1$, $\Pi_1 \equiv \bigvee_x \Pi_1'$.

Für ein "neues" y folgt nach Beh 3: es gibt δ mit

$$\mathrm{Subst}(\Pi_1' \alpha_1, x, y, \delta), \quad \mathrm{Subst}(\delta, y, x, \Pi_1' \alpha_1)$$

und $\bigvee_x \Pi_1' \alpha_1 \approx \bigvee_y \delta$

Weil δ durch Subst aus $\Pi_1' \alpha_1$ entsteht, muß es die Form $\delta \equiv \Pi_1' \delta$ haben

das müßte man streng genommen auch beweisen...

Also bekommen wir

$$\Pi_1 \alpha_1 \equiv \bigvee_x \Pi_1' \alpha_1 \approx \bigvee_y \delta \equiv \bigvee_y \Pi_1' \delta$$

und damit

$$\Pi_1 \alpha_1 \vee \Pi_2 \alpha_2 \approx \bigvee_y \Pi_1' \delta \vee \Pi_2 \alpha_2 \approx \bigvee_y (\Pi_1' \delta \vee \Pi_2 \alpha_2)$$

nach Beh. 2, da $y \notin f(\Pi_2 \alpha_2)$, da y "neu"

Wegen $|\Pi_1'| + |\Pi_2| < |\Pi_1| + |\Pi_2|$ folgt aus der Induktionsvoraussetzung

es gibt Π', β mit $\Pi_1'\alpha \vee \Pi_2\alpha \approx \Pi'\beta$ pränex.

quantorenfrei

Insgesamt folgt

$$\Pi_1\alpha_1 \vee \Pi_2\alpha_2 \approx \underbrace{\bigvee_{\exists} \Pi'}_{\equiv:\Pi}\beta \equiv \Pi\beta \text{ pränex.}$$

$\boxed{\text{Beh 4}}$

Jetzt sind wir mit den Vorbereitungen fertig:

Lemma 31 (Pränexe Normalform)

$\forall\alpha \ \exists\Pi, \beta$ quantorenfrei : $\alpha \approx \Pi\beta$.

Beweis: durch Induktion über α:

Für $\alpha \equiv t_1 = t_2$ oder $\alpha \equiv Pt_1 \ldots t_n$ ist nichts zu zeigen.

$\alpha \equiv \neg\alpha'$: nach Induktionsvoraussetzung gibt es Π, β' quantorenfrei mit $\alpha' \approx \Pi\beta'$

$$\Longrightarrow \alpha \underset{\text{Beh 1}}{\approx} \neg\Pi\beta' \approx \Pi^{-1}\neg\beta' \text{ pränex.}$$

$\alpha \equiv \alpha' \vee \alpha''$: nach Induktionsvoraussetzung gibt es $\Pi_1, \Pi_2, \alpha_1, \alpha_2$ quantorenfrei mit

$$\alpha' \approx \Pi_1\alpha_1$$
$$\alpha'' \approx \Pi_2\alpha_2$$

Beh 4 $\Longrightarrow$ es gibt Π, β quantorenfrei:

$$\Pi_1 \alpha_1 \vee \Pi_2 \alpha_2 \approx \Pi \beta,$$

also folgt

$$\alpha \equiv \alpha' \vee \alpha'' \approx \Pi_1 \alpha_1 \vee \Pi_2 \alpha_2 \approx \Pi \beta \text{ pränex.}$$

$\boxed{\text{Lemma 31}}$

$\alpha \equiv \bigvee_x \alpha'$: nach Ind. Vor.: $\alpha' \approx \Pi \beta$ pränex

$\Rightarrow \alpha \approx \bigvee_x \Pi \beta$ pränex.

In diesem Beweis steckt ein Algorithmus, d.h., ein Verfahren, mit dem man einen gegebenen Ausdruck in pränexe Form bringen kann.

<u>Beispiel</u> $\quad \bigvee_x fxy = z \longrightarrow \bigwedge_y mzy = x \wedge \bigvee_z fvy = z$

$$\equiv \quad \neg \bigvee_x fxy = z \vee \neg \left(\neg \bigwedge_y mzy = x \vee \neg \bigwedge_z fvy = z \right)$$

$$\underset{\text{Beh 1}}{\approx} \quad \bigwedge_x fxy \neq z \vee \neg \left(\bigvee_y mzy \neq x \vee \bigvee_z fvy \neq z \right)$$

z hier frei

y hier frei

⇒ kein einfacher Fall

$$\underset{\substack{\text{Beh 3:}\\ w \text{ neue Variable}}}{\approx} \quad \bigwedge_x fxy \neq z \vee \neg \underbrace{\left(\bigvee_w mzw \neq x \vee \bigvee_z fvy \neq z \right)}$$

$$\underset{\text{Beh 2}}{\approx} \quad \bigvee_w \left(mzw \neq x \vee \bigvee_z fvy \neq z \right)$$

noch eine Umbenennung nötig ...

$$\underset{\substack{\text{Beh 3}\\ u \text{ neue Var.}}}{\approx} \quad \bigwedge_x fxy \neq z \vee \neg \underbrace{\left(\bigvee_w mzw \neq x \vee \bigvee_u fvy \neq u \right)}$$

$$\underset{\text{Beh 2}}{\approx} \quad \bigvee_u \bigvee_w \left(mzw \neq x \vee fvy \neq u \right.$$

$$\underset{\text{übertrag}}{\approx} \quad \bigwedge_x fxy \neq z \vee \neg \underbrace{\bigvee_u \bigvee_w (mzw \neq x \vee fvy \neq u)}$$

$$\underset{\text{Beh 1}}{\approx} \quad \bigwedge_u \bigwedge_w \underbrace{\neg (mzw \neq x \vee fvy \neq u)}_{\equiv\, mzw = x \,\wedge\, fvy = u}$$

$$\approx \quad \bigwedge_x fxy \neq z \vee \bigwedge_u \bigwedge_w (mzw = x \wedge fvy = u)$$

umbenennen, weil hier noch x steht

$$\underset{\substack{\text{Beh 3} \\ t \text{ neue Var}}}{\approx} \quad \bigwedge_t fty \neq z \vee \bigwedge_u \bigwedge_w (mzw = x \wedge fvy = u)$$

$$\underset{\text{Beh 2}}{\approx} \quad \bigwedge_t \left(fty \neq z \vee \bigwedge_u \bigwedge_w (mzw = x \wedge fvy = u) \right)$$

$$\underset{\text{Beh 2}}{\approx} \quad \bigwedge_t \bigwedge_u \bigwedge_w \left(fty \neq z \vee (mzw = x \wedge fvy = u) \right)$$

4.6 Einige Ausblicke

4.6.1 Unentscheidbarkeit von $\vdash$, $\models$ im PL

Eine naheliegende Frage lautet:

Wie kann man testen, ob ein Ausdruck $\alpha \in PL(\mathcal{S}, V)$ gültig ($\Leftrightarrow$ herleitbar) ist?

In der Aussagenlogik genügte es, die Wahrheitstafel aufzustellen; das war zwar mühsam, aber immerhin möglich.