Daraus folgt für die Äquivalenz aussagenlogischer Ausdrücke (S. 136):

$$\alpha \sim \beta \underset{\text{nach Def. } \sim}{\iff} \begin{matrix} \alpha \models \beta \\ \text{und} \\ \beta \models \alpha \end{matrix} \underset{\text{nach Def. } \models}{\iff} \models \alpha \leftrightarrow \beta$$

$$\underset{\text{Vollst./Korr}}{\iff} \vdash \alpha \leftrightarrow \beta \iff \begin{matrix} \alpha \vdash \beta \\ \text{und} \\ \beta \vdash \alpha \end{matrix}$$

"$\Leftarrow$" nach Lemma 24
"$\Rightarrow$" nach MP

Damit ist zumindest Frage 2 von S. 137 beantwortet: Wenn α und β dieselben Wahrheitstafeln haben, so kann man sie im Kalkül voneinander ableiten, und umgekehrt.

Anwendung: Kompaktheitssatz $M \models \alpha \Rightarrow \exists E \subseteq M$ endlich: $E \models \alpha$

hier noch nicht so erstaunlich

4.3. Normalformen der Aussagenlogik

Frage 1 von S. 135 ist noch offen: Wie "sieht man", ob ein Ausdruck α erfüllbar oder gültig ist?

Auf jeden Fall hilft es, α "in Form zu bringen"!

Beispiel: $\alpha \equiv \neg(p \wedge (q \vee \neg r)) \wedge (s \wedge \neg p \rightarrow t)$

Ist α gültig?

Kann man so kaum sehen

Definition Ein Ausdruck $\beta \in AL(\Pi)$ ist in Konjunktiver Normalform (KNF), falls es ein $m \in \mathbb{N}$ und $\alpha_1, \ldots, \alpha_m \in AL(\Pi)$ gibt mit

$$\beta \equiv \alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_m \,, \quad \text{wobei}$$

$$\alpha_i \equiv \lambda_{i,1} \vee \lambda_{i,2} \vee \lambda_{i,3} \vee \ldots \vee \lambda_{i,j_i} \quad \text{mit}$$

$$\lambda_{i,j} \in \Pi \quad \text{oder} \quad \lambda_{i,j} = \neg \mu_{i,j} \quad \text{mit } \mu_{i,j} \in \Pi.$$

Die α_i heißen Klauseln, die $\lambda_{i,j}$ nennt man Literale.

Definition

Anders gesagt: Ein Ausdruck in KNF ist
eine Konjunktion (= UND-Verknüpfung)
von Disjunktionen (= ODER-Verknüpfungen)
von Literalen (= negierten oder nicht negierten Boole'schen Variablen.

Beispiel: $\alpha' \equiv \underbrace{(\neg p \vee \neg q)} \wedge \underbrace{(\neg p \vee r)} \wedge \underbrace{(p \vee \neg s \vee t)}$

Klauseln α_i

Lemma 26 β in KNF ist genau dann gültig, wenn es in jeder Klausel α_i ein Literal $\lambda_{i,j}$ gibt dessen Negation ebenfalls in α_i vorkommt.

Beweis β gültig $\Longleftrightarrow$ jede Klausel α_i ist gültig

$\Longleftrightarrow$ in jeder Klausel α_i kommt ein $\lambda_{i,j}$ zusammen mit $\lambda_{i,j'} \equiv \neg \lambda_{i,j}$ vor.

Lemma 26

Wie bringt man einen Ausdruck in KNF?

1. Schritt: alle Negationszeichen $\neg$ ganz nach innen bringen,

d.h. $\neg(\alpha \wedge \beta) \longrightarrow (\neg\alpha \vee \neg\beta)$

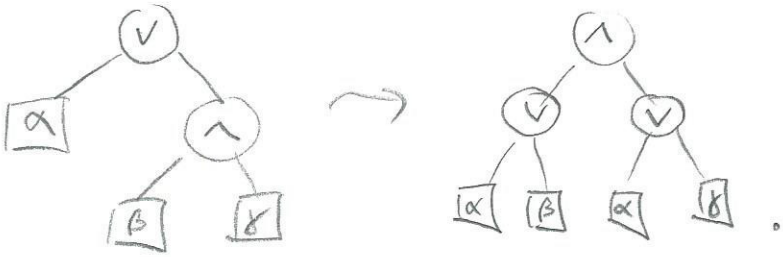
$\neg(\alpha \vee \beta) \longrightarrow (\neg\alpha \wedge \neg\beta)$

2. Schritt alle $\vee$-Zeichen vor Klammern "ausmultiplizieren".

d.h. $\alpha \vee (\beta \wedge \gamma) \rightarrow (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$

Diese Regeln könnte man natürlich auch im Kalkül ableiten!

Dabei wird der Ausdruck länger, aber die $\wedge$-Zeichen wandern im Syntaxbaum des Ausdrucks weiter nach oben:



Beispiel $\alpha \equiv \underbrace{\neg(p \wedge (q \vee \neg r))}\wedge (\underbrace{s \wedge \neg p \rightarrow t})$

$\neg(p \wedge (q \vee \neg r))$ 1. Schritt $\neg p \vee \neg(q \vee \neg r)$ 1. Schritt $\neg p \vee (\neg q \wedge r)$ 2. Schritt $(\neg p \vee \neg q) \wedge (\neg p \vee r)$

$s \wedge \neg p \rightarrow t \equiv \underbrace{\neg(s \wedge \neg p) \vee t}$ 1. Schritt $(\neg s \vee p) \vee t$ überflüssige ( ) weglassen $\neg s \vee p \vee t$

ergibt $\alpha' \equiv (\neg p \vee \neg q) \wedge (\neg p \vee r) \wedge (p \vee \neg s \vee t)$ in KNF

(vgl. S. 152)

Zur konjunktiven Normalform (KNF) dual ist die disjunktive Normalform:

<u>Definition</u> Ein Ausdruck $\gamma \in AL(\Pi)$ ist in Disjunktiver Normalform (DNF), falls es ein $m \in \mathbb{N}$ und $\kappa_1, \dots, \kappa_m \in AL(\Pi)$ gibt mit

$$\gamma \equiv \kappa_1 \vee \kappa_2 \vee \dots \vee \kappa_m ,$$

wobei $\kappa_i \equiv \lambda_{i,1} \wedge \lambda_{i,2} \wedge \dots \wedge \lambda_{i,j_i}$

für Literale $\lambda_{i,j}$.

<u>Beispiel</u>

$$\alpha'' \equiv (\neg p \wedge \neg s) \vee (\neg p \wedge t) \vee (p \wedge \neg q \wedge r) \vee (\neg q \wedge r \wedge \neg s) \vee (\neg q \wedge r \wedge t)$$

Wie bringt man ein α in DNF?

<u>1. Schritt</u> alle Negationszeichen nach innen, wie oben

<u>2. Schritt</u> alle $\wedge$-Zeichen vor Klammern ausmultiplizieren, also

$$\alpha \wedge (\beta \vee \gamma) \rightarrow (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

<u>Beispiel</u>

$$\alpha \equiv \underbrace{\neg (p \wedge (q \vee \neg r))} \wedge (\underbrace{s \wedge \neg p \rightarrow t})$$

$\downarrow$ 1. Schritt

$\neg p \vee \neg (q \vee \neg r)$

$\downarrow$ 1. Schritt

$\neg p \vee (\neg q \wedge r)$

$\equiv \underbrace{\neg (s \wedge \neg p) \vee t}$

$\downarrow$ 1. Schritt

$(\neg s \vee p) \vee t$

$\downarrow$

$\neg s \vee p \vee t$

wie auf S. 153

übertrag: $\left(\neg p \vee (\neg q \wedge r)\right) \wedge \left(p \vee \neg s \vee t\right)$

$$\Big\downarrow \text{2. Schritt}$$

$$\underbrace{(\neg p \wedge p)}_{\text{überflüssig}} \vee (\neg p \wedge \neg s) \vee (\neg p \wedge t) \vee (\neg q \wedge r \wedge p) \vee (\neg q \wedge r \wedge \neg s) \vee (\neg q \wedge r \wedge t)$$

$$\Big\downarrow \text{Terme sortieren}$$

$$(\neg p \wedge \neg s) \vee (\neg p \wedge t) \vee (p \wedge \neg q \wedge r) \vee (\neg q \wedge r \wedge \neg s) \vee (\neg q \wedge r \wedge t)$$

$\equiv \alpha''$ von S. 154.

Offenbar gilt das Gegenstück zu Lemma 26:

Lemma 27 γ in DNF ist genau dann erfüllbar, wenn es eine Klausel κ_i gibt, in der kein Literal $\lambda_{i,j}$ zusammen mit seiner Negation $\neg \lambda_{i,j}$ auftritt.

$\boxed{\text{Lemma 27}}$

Man kann wegen Lemma 26 und Lemma 27 also ganz einfach feststellen, ob

- ein Ausdruck in KNF gültig
- ein Ausdruck in DNF erfüllbar ist.

Wie testet man einen beliebigen Ausdruck α auf Erfüllbarkeit?

Man könnte α in DNF bringen und dann die Erfüllbarkeit leicht feststellen.

<u>Aber:</u> Diese Umformung kann – bezogen auf die Länge von α als Zeichenreihe – exponentiell viele Schritte erfordern!

<u>Beispiel:</u> Besonders schlimmer Fall: α in KNF, z.B.

$$\alpha \equiv (\lambda_1 \vee \lambda_2 \vee \dots \vee \lambda_a) \wedge (\mu_1 \vee \mu_2 \vee \dots \vee \mu_a) \wedge \dots \wedge (\varrho_1 \vee \varrho_2 \vee \dots \vee \varrho_a)$$

k Klauseln

α besteht aus k Klauseln mit je a vielen Literalen, hat also etwa die Länge $a \cdot k$.

Beim "Ausmultiplizieren" ergeben sich aber

$a \cdot a \cdots a = a^k$ viele Klauseln $(\lambda_i \wedge \mu_j \wedge \dots \wedge \varrho_m)$

in der DNF, weil i, j und m von 1 bis a laufen.

Das Ergebnis hat also etwa die Länge $a^k \cdot k$.

Berechnung der Wahrheitstafel von α würde, wenn die Literale $\lambda_i, \mu_j, \varrho_m$ alle verschieden wären, $2^{a \cdot k}$ Schritte benötigen, wäre also noch mühsamer.

<u>Beobachtung:</u> Sei $\alpha \equiv (\lambda_1 \vee \lambda_2 \vee \dots \vee \lambda_a) \wedge \dots \wedge (\varrho_1 \vee \varrho_2 \vee \dots \vee \varrho_a)$ in KNF, wie oben.

Wenn man erst einmal eine Bewertung von den

Variablen in α geraten hat, kann man sehr schnell überprüfen, ob α durch diese Bewertung erfüllt wird.

Beispiel $\alpha' \equiv (\neg p \vee \neg q) \wedge (\neg p \vee r) \wedge (p \vee \neg s \vee t)$

p	q	r	s	t	α
W	F	F	F	W	F

bzw

p	q	r	s	t	α
F	W	W	F	W	W

man muß jeweils nur eine Zeile der Wahrheitstafel ausrechnen, und das geht bei KNF ganz schnell

Salopp gesagt:

Sei NP (= nichtdeterministisch-polynomiell)
die Klasse aller Probleme, für die sich eine geratene Lösung schnell überprüfen läßt (wie oben die Erfüllbarkeit)

und sei P (= polynomiell)
die Klasse aller Probleme, die sich schnell lösen lassen

Klar ist: $P \subseteq NP$

Die große Frage lautet: Ist $P \subsetneq NP$?

Dies ist ein Milennium-Problem des Clay Mathematics Institute in Cambridge. Auf die Lösung ist ein Preis von 10^6 US \$ ausgesetzt.

schnell $\hat{=}$ polynomiell