Lemma: Wenn β die Klauselmenge $\{C_1, \dots, C_m\}$ erfüllt, und R Resolvente von C_1 und C_2 ist, dann erfüllt β auch $\{C_1, \dots, C_m, R\}$.

Beweis: $C_1 = \{L_1, \dots, L_i, A_n\}$

$C_2 = \{L_1', \dots, L_j', \neg A_n\}$

1. Fall $\beta(A_n) = 1$, also (da β C_2 erfüllt) $\beta(L_k') = 1$ für ein k

2. Fall $\beta(\neg A_n) = 1$, also $\beta(L_\ell) = 1$ für ein ℓ

$R = \{L_1, \dots, L_i, L_1', \dots, L_j'\}$ wird von β erfüllt. $\square$

Beweis vom Satz:

Gegeben endliche Klauselmenge $\mathcal{C}$, schließe $\mathcal{C}$ unter Resolution ab, Ergebnis heiße $\mathcal{R}$

Lemma: $\mathcal{C}$ erfüllbar $\Rightarrow$ $\mathcal{R}$ erfüllbar, also $\{\} \notin \mathcal{R}$

$\{\} = \emptyset$

Dies zeigt „$\Rightarrow$"

„$\Leftarrow$" Angenommen $\{\} \notin \mathcal{R}$, zeige: $\mathcal{R}$ erfüllbar.

Induktion über die Anzahl n der Aussagenvariablen in $\mathcal{R}$, o.E. $A_0, \dots, A_{n-1}$

Induktionsanfang $n=0$: einzige mögliche Klausel $\{\}$

1. Fall $\mathcal{R} = \{\{\}\}$, dann $\mathcal{R}$ unerfüllbar

$\{\{\}\} = \{\emptyset\}$

2. Fall $\mathcal{R} = \emptyset$, dann $\mathcal{R}$ erfüllbar

Induktionsschritt $n \to n+1$

$\mathcal{R}$ unter Resolution abgeschlossen, $\{\} \notin \mathcal{R}$, AV $A_0, \dots, A_n$

Es kann nicht sein, dass $\{A_n\} \in \mathcal{R}$ und $\{\neg A_n\} \in \mathcal{R}$

sonst $\{\} \in \mathcal{R}$ per Resolution

1. Fall $\{\neg A_n\} \notin \mathcal{R}$ (2. Fall geht analog)

Konstruiere Klauselmenge $\mathcal{R}'$ in A.V. $A_0, \dots, A_{n-1}$

- lasse alle Klauseln weg, in denen A_n (als Literal) vorkommt
- lasse in allen Klauseln, in denen $\neg A_n$ vorkommt, dieses Literal weg
- alle anderen Klauseln bleiben

$\mathcal{R}'$ ist Klauselmenge in den A.V. $A_0, \dots A_{n-1}$, $\{\} \notin \mathcal{R}'$

$\mathcal{R}'$ ist unter Resolution abgeschlossen

Induktion: $\mathcal{K}'$ ist erfüllbar durch β

setze $\beta(A_n) = 1$. Dann erfüllt β $\mathcal{K}$. ☒

Viele Varianten und Zusatzuntersuchungen zur Resolution!
(vgl. Übungsblatt: Hornformel)

Satz: Zu jeder aussagenlogischen Formel F gibt es eine Formel F^+ in KNF mit den Eigenschaften:

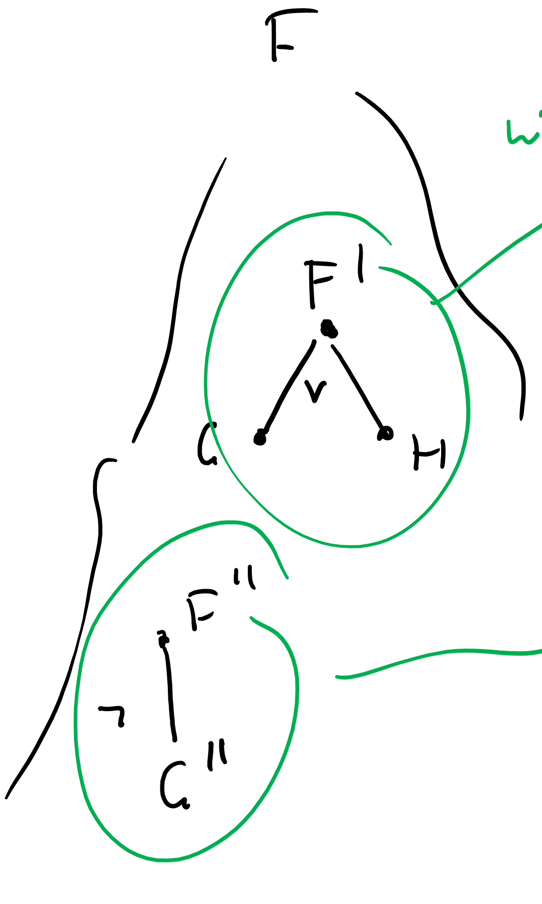
- F ist erfüllbar $\Leftrightarrow$ F^+ ist erfüllbar
- $\lg(F^+)$ ist polynomial beschränkt in $\lg(F)$

„Länge"

$\boxed{\text{Sogar: Klauseln in } F^+ \text{ bestehen aus maximal 3 Literalen}}$

polynomial: n^k k fest
exponentiell: 2^n

Beweis: Für jede Teilformel F' von F, die nicht eine Aussagenvariable ist, nimmt man neue Aussagenvariable $A_{F'}$



wird zu $\rightarrow$ $(A_{F'} \leftrightarrow (A_G \vee A_H))$

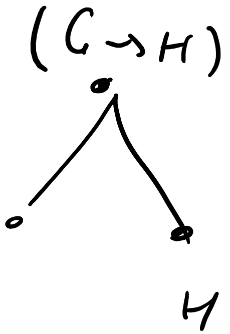
Falls G oder H Aussagenvariable A_i:
ersetze z.B. A_G durch A_i

wird zu $(A_{F''} \leftrightarrow \neg A_{G''})$

F erfüllbar $\Leftrightarrow$ alle diese Formeln gemeinsam erfüllbar sind, also $\bigwedge \ldots$

Übersetzen die Äquivalenten in Klauseln

Bsp $(A_{F'} \leftrightarrow (A_G \to A_H))$



$$\sim ((A_{F'} \to (A_G \to A_H)) \wedge ((A_G \to A_H) \to A_{F'}))$$

$$\sim ((A_{F'} \to (\neg A_G \vee A_H)) \wedge ((\neg A_G \vee A_H) \to A_{F'}))$$

$$\sim ((\neg A_{F'} \vee \neg A_G \vee A_H) \wedge (\underbrace{\neg(\neg A_G \vee A_H) \vee A_{F'}}))$$

$$((A_G \wedge \neg A_H) \vee A_{F'})$$

$$((A_G \vee A_{F'}) \wedge (\neg A_H \vee A_{F'}))$$

Klauselmenge: $\{\{\neg A_{F'}, \neg A_G, A_H\}, \{A_G, A_{F'}\}, \{\neg A_H, A_{F'}\}\}$

(D) Hornformeln: Übungsblatt
oder „Markierungsalgorithmus"
(siehe Schöning oder selbst überlegen)

Schneller Algorithmus!

<u>Kompaktheitssatz</u> (Endlichkeitssatz)

T (evtl unendliche) Formelmenge

T erfüllbar $\iff$ T ist „endlich erfüllbar", d.h. jede endl. Teilmenge von T ist erfüllbar

<u>Beweis</u> „$\Rightarrow$" klar

„$\Leftarrow$" (a) Setze $T_0 := T$ und definiere induktiv endlich erfüllbare T_{n+1}

$$T_{n+1} := \begin{cases} T_n \cup \{A_n\} & \text{falls endlich erfüllbar,} \\ T_n \cup \{\neg A_n\} & \text{sonst} \end{cases}$$

Beh: ist dann endl. erfüllbar

Sei $S \subseteq_{\text{endl.}} T_n$, zeige: $S \cup \{\neg A_n\}$ ist erfüllbar

nicht erfüllbar, also ex. $S_0 \subseteq_{\text{endl.}} T_n$ mit $\underbrace{S_0 \cup \{A_n\} \text{ nicht erfüllbar}}_{\Leftrightarrow \quad S_0 \vdash \neg A_n}$

Monotonie: $S_0 \cup S \vdash \neg A_n$

$S_0 \cup S \subseteq_{\text{end}} T_n$, also erfüllbar durch ein β, also $\beta(\neg A_n) = 1$

$\Rightarrow$ β erfüllt $S_0 \cup S \cup \{\neg A_n\}$ $\square$ Beh.

(b) Alle T_n endlich erfüllbar, also $T_\infty := \bigcup_{n \in \mathbb{N}} T_n$ endl. erfüllbar.

(beachte: $T_n \subseteq T_{n+1}$)

$S \subseteq_{\text{endl.}} T_\infty$, dann ex. n. mit $S \subseteq T_n$

(c) Definiere $\beta(A_i) := \begin{cases} 1 & \text{falls } A_i \in T_\infty \text{, und setze } L_i := A_i \\ 0 & \text{falls } \neg A_i \in T_\infty \text{, und setze } L_i := \neg A_i \end{cases}$

Beh: β erfüllt T_∞

Denn für jedes $F \in T_\infty$ mit Auss. variablen von $F \subseteq \{A_0, \ldots, A_m\}$

betrachte $\{F, L_0, \ldots, L_m\} \underset{\text{endl.}}{\subseteq} T_\infty$, also erfüllbar.

Dies geht nur durch β' mit $\beta'(A_i) = \beta(A_i)$

für $i = 0, \ldots, m$ $\boxtimes$

Satz von Cook und Levin

satisfiability

propositional logic

Das Erfüllbarkeitsproblem SAT (geg. aussl. Formel, ist die erfüllbar?) ist NP-vollständig.

sogar: 3-SAT ist NP-vollständig

(Erfüllbarkeitsproblem für KNF-Formeln mit ≤ 3 Literalen pro Klausel)

Definitionen:

Ein Problem ist eine Teilmenge $M \subseteq A^*$ A endliches Alphabet, z.B. $\{0,1\}$

A^*: Menge der Wörter über A

genauer: das Entscheidungsproblem ist gegebenes $w \in A^*$ in M oder nicht?

Ein Problem ist in P (= PTIME) – polynomial, falls es ein Polynom $t \in \mathbb{R}[X]$ und eine Maschine gibt, die in $\le t(\lg(w))$ Berechnungsschritten entscheidet, ob $w \in M$.

d.h. richtige Antwort wird ausgegeben

- Speicherplatz ist unbeschränkt
- in der Regel: (deterministische) Turing-Maschine

Ein Problem ist in NP (nicht-deterministisch polynomial) ...

nicht-deterministische Turing-Maschine

$P \subseteq NP$

$=, \neq$? großes offenes Problem

$SAT \in NP$