

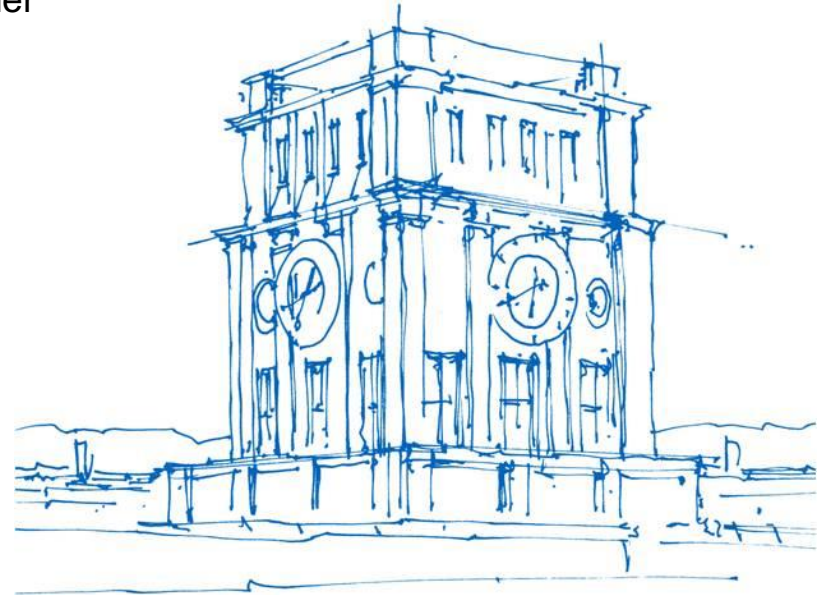
# Diskrete Strukturen – Tutorium KW 04

Jeremias Bohn, Evghenii Beriozchin/Manuela Poschenrieder

Technische Universität München

Fakultät für Informatik

Garching, 21./22. Januar 2021



*Uhrenturm der TUM*

# Erfüllbarkeitsäquivalente KNF-Formeln

Wir wissen bereits, dass wir jede aussagenlogische Formel  $F$  in eine KNF umwandeln können

- Berechnen aller Wahrheitswerte für viele Variablen dauert zu lang und ist unnötig, wenn wir nur wissen wollen, ob die Formel erfüllbar ist
- Idee: Berechne eine erfüllbarkeitsäquivalente Formel  $E_F$  (d.h.  $F$  ist erfüllbar genau dann wenn  $E_F$  erfüllbar ist)
- Für jede Teilformel  $G \in \mathcal{T}_F$  von  $F$ :
  - Wenn  $G$  atomar ist (d.h.  $true$ ,  $false$ ,  $x$  oder  $\neg x$  für  $x \in V_F$ ), setze  $q_G := G$
  - Sonst führe eine neue Variable  $q_G$  ein und bilde eine Formel  $K_G$ :
    - Falls  $G \equiv \neg H$  ist, setze  $K_G := (q_G \leftrightarrow q_H)$
    - Falls  $G \equiv H \text{ op } H'$  ist (op steht für einen bel. Operator), setze  $K_G := (q_G \leftrightarrow (q_H \text{ op } q_{H'}))$
- Es gilt dann  $E_F = q_F \wedge \bigwedge_{G \in \mathcal{T}_F \setminus V_F} K_G$
- Es müssen nur noch die Teilformeln  $K_G$  in Standardform gebracht werden

# Davis-Putnam-Logemann-Loveland-Algorithmus

Der DPLL-Algorithmus ist einer der wichtigsten Algorithmen zum Lösen des Erfüllbarkeitsproblems

- Gegeben: Formel in KNF bzw. Klauselmengendarstellung (z.B.  $(p \wedge (\neg q \vee r))$ ) wird zu  $\{\{p\}, \{\neg q, r\}\}$ , wobei  $\emptyset$  für *true* steht und  $\{\emptyset\}$  für *false*)
- Ablauf: One-Literal-Rule (OLR) über Pure Literal Rule (PLR) über Fallentscheidung (lexikographisch aufsteigend)
- One-Literal-Rule: Gibt es in der Klauselmenge eine Menge, die aus einem einzigen Literal besteht, so wird diese Variable entsprechend auf *true* oder *false* gesetzt. Mengen, die dieses Literal enthalten, werden gelöscht, aus allen anderen wird das Komplement entfernt.
- Pure-Literal-Rule: Kommt eine Variable nur in einer Literalform in allen Klauselmengen vor (d.h.  $x$  oder  $\neg x$ ), so wird sie entsprechend auf *true* oder *false* gesetzt. Mengen, die dieses Literal enthalten, werden gelöscht, aus allen anderen wird das Komplement entfernt.
- Fallentscheidung: Wir machen eine Fallunterscheidung zwischen *true* und *false* und werten beide aus
- Enden alle Wege mit  $\{\emptyset\}$ , so ist die Formel unerfüllbar. Gibt es einen Weg, der mit  $\emptyset$  endet, so ist die Formel erfüllbar.

# Resolution

Die Resolution ist sozusagen das „Gegenstück“ zum DPLL-Algorithmus, sie überprüft, ob eine Formel unerfüllbar ist

- Eingabe: Eine Formel in KNF bzw. Klauselmengendarstellung
- Wir können eine neue Klausel an diese KNF hinzufügen, indem wir logische Schlüsse ziehen:
  - Wir können aus zwei Klauseln  $K_1, K_2$  einen neuen **Resolventen** erzeugen, wenn es ein Literal  $L$  gibt, sodass  $\{L\} \subseteq K_1$  und  $\{\neg L\} \subseteq K_2$ . Der neue Resolvent ist dann  $(K_1 \setminus \{L\}) \cup (K_2 \setminus \{\neg L\})$  und kann der Menge der Klauseln hinzugefügt werden.
  - **VORSICHT!** Immer nur ein Literal gleichzeitig! Es dürfen nicht zwei Literale gleichzeitig gelöscht werden!
  - Ziel: Kombiniere geschickt, sodass die Menge  $\{\emptyset\}$  entsteht, denn so schlussfolgern wir die Klausel *false*