

# Algorithmisches Beweisen LAB

Hornformeln

Luc Spachmann

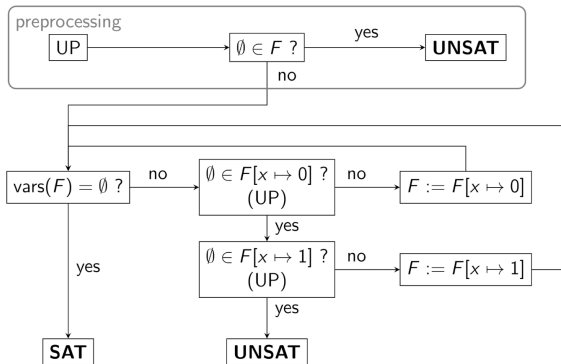
FSU Jena

02.05.2022

- Implementierung von SAT-Lösern
  - 2-SAT
  - Hornformeln
  - DPLL
  - CDCL (Schrittweise)

# Nachtrag

- 2-SAT Algorithmus letzter Woche im worst-case nicht linear
- Kann aber modifiziert werden



# Horn Formeln

- Spezielle Art von KNF
- Eine **Hornklausel** ist eine Klausel mit maximal einem positiven Literal
- Eine **Hornformel** ist eine KNF bestehend aus Horn Klauseln
- Hornformeln sind in Linearzeit lösbar (heutiger Algorithmus aber quadratisch)

$$(\neg x_1 \vee \neg x_2 \vee x_3)$$

Hornklausel

$$(x_1 \vee \neg x_2 \vee x_3)$$

Keine Hornklausel

$$(\neg x_1)$$

Hornklausel

$$(x_1)$$

Hornklausel

# Alternative Sicht auf Hornformeln

Jede Hornformel ist äquivalent zu einer Implikation:

$(\neg x_1 \vee \neg x_2 \vee x_3)$	äquivalent zu	$x_1 \wedge x_2 \rightarrow x_3$
$(x_1 \vee \neg x_2)$	äquivalent zu	$x_2 \rightarrow x_1$
$(\neg x_1 \vee \neg x_2)$	äquivalent zu	$x_1 \wedge x_2 \rightarrow 0$
$(x_1)$	äquivalent zu	$1 \rightarrow x_1$

# Algorithmus für Hornformeln

**Eingabe:** Hornformel  $F$

- 1: Sei  $\alpha$  die leere Belegung
- 2: **while** Positive Unit-Klausel  $(x_i)$  existiert **do**
- 3:      $\alpha = \alpha \cup \{x_i\}$
- 4:      $F = F[\alpha]$
- 5:     **if**  $\emptyset \in F$  **then**
- 6:         **return** UNSAT
- 7:     **end if**
- 8: **end while**
- 9: Alle nicht in  $\alpha$  enthaltenen Variablen auf 0 setzen
- 10: **return** SAT,  $\alpha$

# Aufgabe: Hornformeln

- Erweiterung des Random-CNF Programms zur Erzeugung von Hornformeln
  - Wichtig: Für Hornformeln müssen auch schmalere Klauseln erlaubt sein (insbesondere Unit Klauseln)
  - Alternativ auch separates Programm möglich
- Implementierung des Algorithmus für Hornformeln
- Ausgabe einiger Statistiken:
  - Zeit
  - Speicherbedarf
  - Anzahl Unit Propagations
  - etc.