

# Algorithmisches Beweisen LAB

## CDCL

Luc Spachmann

FSU Jena

16.05.2024

- Implementierung von SAT-Lösern
  - 2-SAT
  - Hornformeln
  - DPLL
  - CDCL

- Gemeinsamkeiten
  - Unit propagation
  - Backtracking
- Algorithmische Verbesserungen
  - **Klausellernen**
  - Nichtlineares Backtracking (“Backjumping”)
- Weitere Verbesserungen
  - “Watched Literals”
  - Entscheidungsheuristiken
  - Neustarts

# CDCL Pseudocode

**Eingabe:** CNF  $\varphi$

```
1: decision-level  $\leftarrow$  0
2: while Es existieren nicht belegte Variablen do
3:   decision-level  $\leftarrow$  decision-level + 1
4:   decide()
5:    $C_{\text{conflict}} \leftarrow$  propagate()
6:   while  $C_{\text{conflict}}$  is not null do
7:     if decision-level = 0 then return UNSAT
8:      $C_{\text{learned}}, \text{new\_dec\_lvl} \leftarrow$  analyze-conflict( $C_{\text{conflict}}$ )
9:      $\varphi \leftarrow \varphi \wedge C_{\text{learned}}$ 
10:    backtrack(new_dec_lvl)
11:     $C_{\text{conflict}} \leftarrow$  propagate()
12:  apply-restart-policy()
13: return SAT
```

- decision-level
  - Setzen einer Variable auf einen Wert
  - Hier: Beliebige Variable
- propagate()
  - Unit Propagation falls möglich
- apply-restart-policy()
  - Entscheidung ob neu gestartet werden soll
  - Meist anhand von Anzahl an Konflikten
  - Hier: Kein Neustart
- backtrack()
  - Backtrack auf von Konfliktklausel abhängige Stufe
  - Hier: Eine Stufe zurück
  - Funktion ändert auch das decision-level

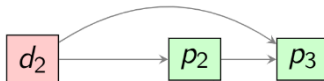
# Konfliktanalyse

decision  
level

1



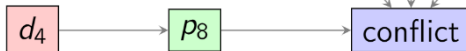
2



3



4



- Zentral in CDCL
- Ermittelt auch das neue decision-level
- Hier:
  - Gelernte Klausel: Negation aller Entscheidungsliterale
  - Neues decision-level: Eins zurück

- Beweis in zusätzliches Dokument
- Dateiendung \*.drat
- Jede gelernte Klausel in DIMACs Format anhängen
- Bei UNSAT leere Klausel anhängen
- Beweis kann in drat proof checker getestet werden (für UNSAT)
- Empfohlen: Drat-Trim (Heule et al)
- <https://github.com/marijnheule/drat-trim>



# Aufgabe: CDCL

- Implementierung des Algorithmus für CDCL mit o.g. Funktionen
- Schreiben Sie das Programm so, dass Sie es wiederverwenden können!
- Nächste Wochen: Erweiterung der einzelnen Funktionen
- Output file für Beweis
- Ausgabe einiger Statistiken:
  - Zeit
  - Speicherbedarf
  - Anzahl Unit Propagations
  - Anzahl Entscheidungen
  - Anzahl Konflikte
  - etc.