Proof Complexity and Solving LAB

Clause Learning

Dr. Joshua Blinkhorn

Friedrich-Schiller-Universität Jena
https://github.com/JoshuaBlinkhorn/SAT-LAB

Goals

- Implemenatation of SAT solving algorithms

  (a) 2-SAT (polynomial time)

  (b) DPLL

  (c) CDCL

    - watched literals
    - clause learning
    - decision heuristics
    - restart strategy

  (d) QBF expansion..

- Practical programming experience

  - use your favourite language (Python, C, C++, Java, ..)
  - recommended: Python

CDCL Pseudocode

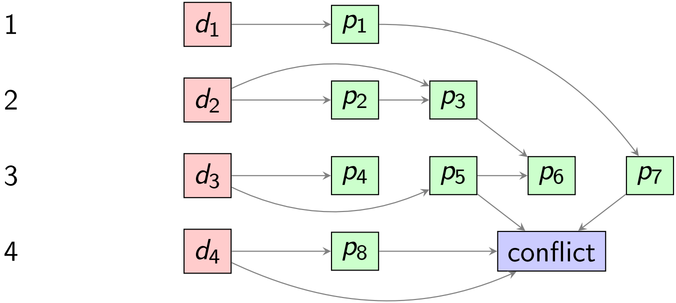
```
function CDCL-solver(Φ)                               #assuming Φ is preprocessed
  decision-level ← 0
  while there are unassigned variables
    decision-level++
    decide()                                          #adds assignment to trail
    C_conflict ← propagate()                          #returns conflict clause or null
    while C_conflict is not null
      if decision-level = 0 return UNSAT
      C_learned ← analyse-conflict(C_conflict)
      if C_conflict is unit
        backtrack(0)
        assign unit literal
      else
        backtrack(asserting-level(C_learned))         #changes trail and DL
        Φ ← Φ ∧ C_learned
      C_conflict ← propagate()
    apply-restart-policy()
  return SAT
```

Resolution

$$(C \vee x), (D \vee \neg x) \quad \vdash \quad (C \vee D)$$

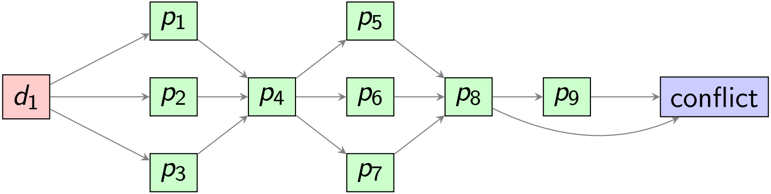
- $(C \vee D)$ is the resolvent
- x is the pivot
- logically correct rule: premises imply resolvent
- sound and complete proof system for UNSAT

Clause Learning - Cutting the Implication Graph



Unique Implication Points

- A unique implication point (UIP) is:
  - a node at the highest decision level
  - every path from highest decision to conflict passes through it



Common Implementation

- advantage: clauses learned from UIPs are always asserting
- asserting means 'becomes unit at a previous decision level'
- the asserting level is second highest decision level in learned clause
- hence: backtrack to asserting level ('backjumping')
- remember: we don't learn unit clauses – we make assignments instead

- easy implementation:
- resolve conflict clause with reason clauses until there is exactly one variable at highest decision level
- This is the 1UIP learning scheme

Thoughts

- Don't worry too much about why 1UIP schemes work well
- My opinion: the community has not agreed on a clear reason
- Theoreticians: SAT solver are automated decision procedures
- Practitioners: SAT solvers perform local search

Clause Learning Task

- implement a 1UIP clause learning scheme in your CDCL solver
- check correctness
- compare the solving time to DPLL-style learning