Proof Complexity and Solving LAB

Restart Strategies

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($\Phi$)                                   #assuming $\Phi$ is preprocessed
    decision-level ← 0
    while there are unassigned variables
        decision-level++
        decide()                                               #adds assignment to trail
        $C_{\text{conflict}}$ ← propagate()                    #returns conflict clause or null
        while $C_{\text{conflict}}$ is not null
            if decision-level = 0 return UNSAT
            $C_{\text{learned}}$ ← analyse-conflict($C_{\text{conflict}}$)
            if $C_{\text{conflict}}$ is unit
                backtrack(0)
                assign unit literal
            else
                backtrack(asserting-level($C_{\text{learned}}$))   #changes trail and DL
                $\Phi \leftarrow \Phi \wedge C_{\text{learned}}$
            $C_{\text{conflict}}$ ← propagate()
        apply-restart-policy()
    return SAT
```

Why Restart?

- Sometimes we make bad decisions
- Bad decisions lead to slower solving
- Restarts: fresh chance to make better decisions

- "But my decision heuristic is deterministic!?"
- Vital: initial decision must be randomised
- To restart: simply delete assignment and start again with first decision
- Recommended: delete learned clauses as well

Geometric Restarts

$$a\,,\, ar\,,\, ar^2\,,\, ar^3\,,\, ar^4\,,\, \ldots$$

- Use a geometric progression to determine when to restart:
  - choose values for a and r, e.g. $a = 100, r = 2$
  - first restart after a decisions
  - second restart after ar decisions
  - third restart after ar^2 decisions
  - and so on..
- Termination is guaranteed provided $a > 0$ and $r > 1$

Restarts Task

- Implement a geometric restart strategy
- Check correctness and termination
- Try to find values of a and r that improve your solver compared to no restarts

- optional/extra: Luby restarts
  - similar to geometric restarts
  - better (optimal) sequence
  - see the following for the details:

Haim, S., Heule, M.: *Towards Ultra Rapid Restarts.* CoRR 2014.