Proof Complexity and Solving LAB

2-SAT

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 (decision tree)
  (c) CDCL
    - clause learning
    - watched literals
    - decision heuristics
    - restart strategy
  (d) QBF expansion..

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

Restrictions

$$(\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_4)$$

- a restriction is the application of an assignment to a formula
- two operations in CNF:
  - a literal may be deleted from a clause
  - a clause may be deleted from the formula
- e.g. $x_1 \mapsto 0, x_2 \mapsto 0$

$$(\neg x_3) \wedge (x_3 \vee \neg x_4)$$

Unit Propagation

$$(\neg x_1) \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_4) \wedge (x_4 \vee \neg x_5)$$

- a unit clause has just one literal: $(\neg x_1)$
- a unit assignment satisfies a unit clause: $x_1 \mapsto 0$
- unit propagation is the successive and exhaustive application of unit assignments
- unit propagation on F yields
  - the assignment $\alpha : x_1 \mapsto 0, x_2 \mapsto 0, \ldots, x_5 \mapsto 0$
  - the formula $F[\alpha] = \top$

A Feature of 2-SAT

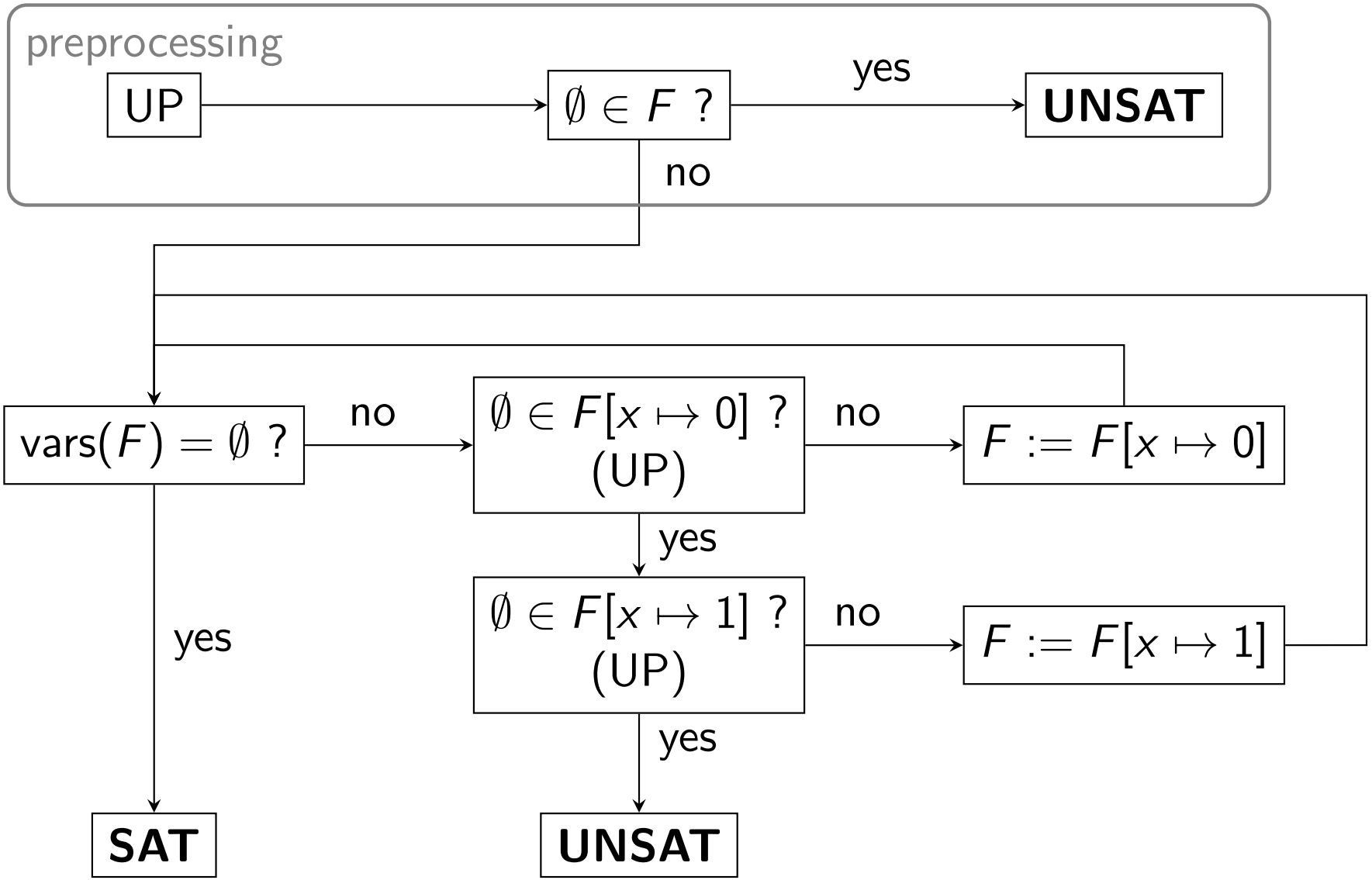
- let $F[x \mapsto b]$ denote the application of $x \mapsto b$ plus unit propagation on F
- a decision plus unit propagation preserves satisfiability if it does not yield the empty clause:

$$\emptyset \notin F[x \mapsto b] \qquad \Longrightarrow \qquad F \equiv_{\mathsf{sat}} F[x \mapsto b]$$

- so if:
  - F is satisfiable, and
  - $F[x \mapsto b]$ does not contain the empty clause,

  then
  - $F[x \mapsto b]$ is satisfiable
- gives rise to a non-backtracking algorithm

2-SAT Algorithm



2-SAT Task

- implement the 2-SAT algorithm
- include a README
- test your solver on random 2-SAT formulas
- print some statistics, e.g.
  - solving time
  - memory consumption
  - number of decisions
  - number of unit propagations
  - ..?
- at this point: don't try to make your code reusable for later projects (DPLL, CDCL, ..)