

Verified SAT Solver Competition

Ruben Martins
October 23, 2018

Carnegie
Mellon
University

The Success of SAT

Satisfiability Modulo Theories
Quantified Boolean Formulas

Software Model Checking
Haplotyping
Test Pattern Generation

Software Testing
Network Security Management

Maximum Satisfiability Configuration

Package Management
Filter Design
Equivalence Checking

Hardware Model Checking
Model Finding
Planning
Power Estimation
Circuit Delay Computation
Test Suite Minimization

Constraint Programming
Binate Covering
Fault Localization

Noise Analysis
Pedigree Consistency

Technology Mapping Games
Function Decomposition

Termination Analysis

FPGA Routing
Abstract Argumentation

Switching Network Verification
Minimum Satisfiability

Resource Constrained Scheduling

Symbolic Trajectory Evaluation
Telecom Feature Subscription

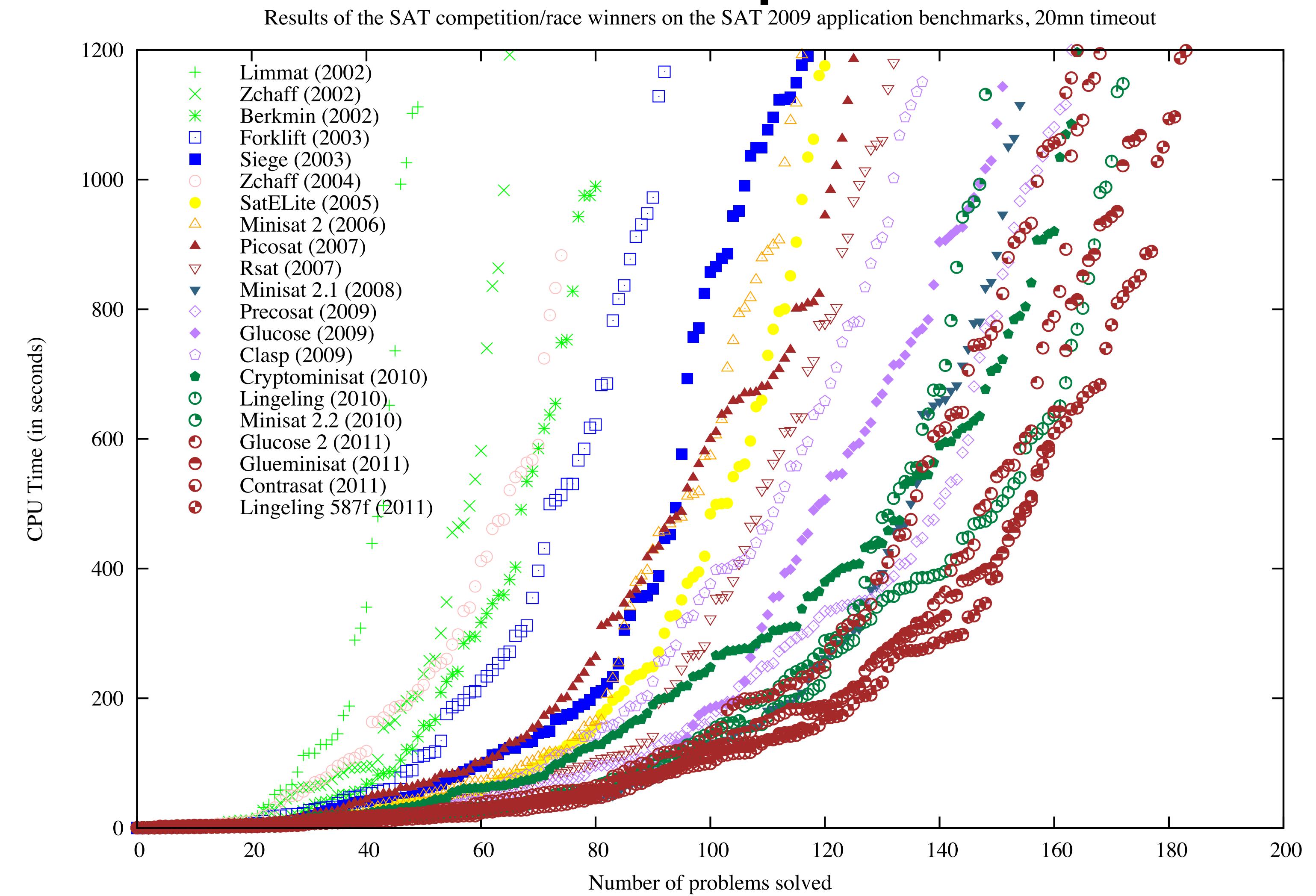
Timetabling

Design Debugging
Genome Rearrangement

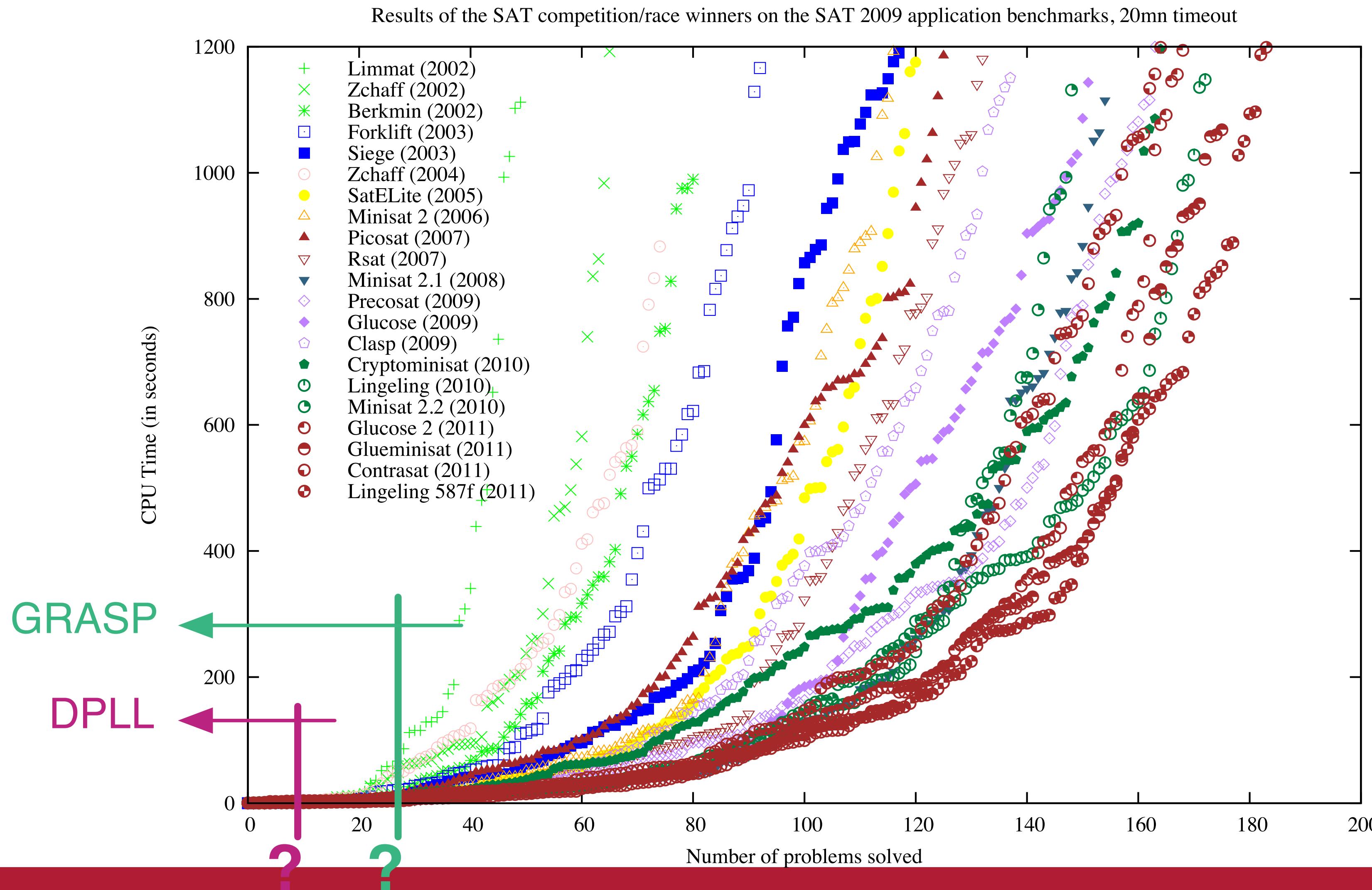
Lazy Clause Generation
Pseudo-Boolean Formulas



SAT Solver Improvement



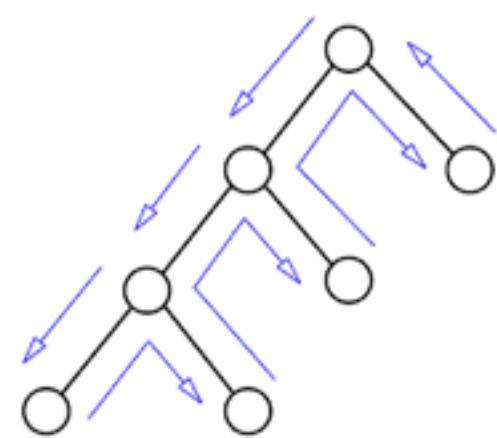
SAT Solver Improvement



Building a Verified SAT Solver



Lab 3: Sheep SAT - enumerate all interpretations



Lab 4: DPLL - improve the solver with unit propagation



Verified SAT Competition



Improve your **verified** solver with:

- Simplification techniques (pure literal, ...)
- More efficient data structures for unit propagation
- Clause learning
- ...



How can I participate?



1. Improve your verified solver
2. Write a 1 page description of your techniques
3. Give your solver a cool name and solve real problems!



How can I participate?



1. Improve your verified solver
2. Write a 1 page description of your techniques
3. Give your solver a cool name and solve real problems!

If you improve your solver you will **not need** to do Lab 5 (Bounded Model Checking)!



What can I win?



December 3 - December 7

1. Fame and glory!
2. Prizes for:
 - (1) Most efficient verified solver
 - (2) Most original verified techniques
 - (3) Random winner
 - (4)
3. Food for everyone!

