

# clsat

- clsat = SMT-LIB format parser + CNF converter + SAT solver + QF\_IDL solver
- Started as a half semester class project for "Intro. to Automated Theorem Proving" in Fall 2007
- In 2008, Duckki Oe, Timothy Simpson and Terry Tidwell continued to work to finish up and improve it.

# The SAT solver in clsat

- Modern features
  - Watched literals
  - Lemma learning and back jumping
  - Conflict clause simplification (tinisat-like)
  - Restarts
- Resolution proof generation

# SAT Performance

Benchmark	Size	clsat	minisat	tinisat
een-tipb-sr06-par1	8.4M	1.54	1.46	1.43
een-tipb-sr06-tc6b	1.9M	0.38	0.22	0.34
manol-pipe-c10ni_s	10M	4.94	43.42	7.14
manol-pipe-c6nid_s	7.4M	13.81	162.01	93.56
manol-pipe-f6b	1.7M	16.03	4.02	5.41
manol-pipe-f6n	1.7M	12.22	4.57	6.58
manol-pipe-g6bid	1.8M	15.59	3.60	3.99
manol-pipe-g7n	1.1M	11.27	2.75	6.46
velev-eng-uns-1.0-04	1.0M	19.37	5.19	5.63
velev-sss-1.0-cl	184K	2.86	0.41	0.21

# Resolution Proof

- The sequence of resolutions to derive a conflict clause can be recorded as a proof.
- Conflict clause simplification also can be modeled by resolution.
- topologically sort eliminated literals w.r.t the dependency between them to determine a correct resolution sequence.

# Decision Heuristic

- Based on VSIDS
- IDL solver provides a light-weight function that tests if a literal is suspected false.
- see if one of the top variables in score is suspected false positively or negatively and the negation is asserted (prefer true lits)
- a simpler form of Theory Propagation ?

# More to come in clsat

- Proof generation for QF\_IDL
- Catching up with mature solvers
  - Refining conflict clauses (more precisely)
  - Better handling equalities
  - Leveraging structure of original formula