

The Sat4j library, release 2.3.2

on the fly solver configuration

Daniel Le Berre et Stéphanie Roussel

CRIL-CNRS UMR 8188 - Université d'Artois

Pragmatics of SAT workshop , June 16, 2012, Trento



Motivation

Metrics

Interacting with the solver

Examples

Demo

Conclusion

SAT, a push button technology?

[Remark : SAT solver really means CDCL SAT solver in this talk]

Often heard about SAT :

- ▶ SAT solvers are black boxes
- ▶ Fed using Dimacs formatted CNF
- ▶ Many efficient implementations available
- ▶ Simply update to the SAT competition/race/challenge winner each year



SAT, a push button technology?

[Remark : SAT solver really means CDCL SAT solver in this talk]

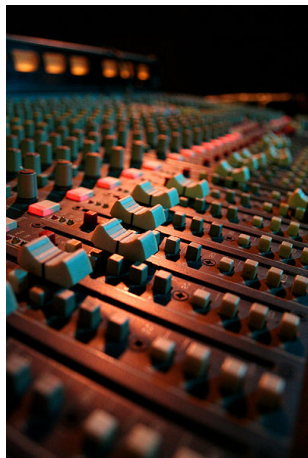
Often heard about SAT :

- ▶ SAT solvers are black boxes
- ▶ Fed using Dimacs formatted CNF
- ▶ Many efficient implementations available
- ▶ Simply update to the SAT competition/race/challenge winner each year

Not really true in practice !



SAT solvers do have a variety of buttons !



- ▶ Heuristics
- ▶ Restarts
- ▶ Minimization
- ▶ DB Cleanup
- ▶ Pre-processing
- ▶ In-processing
- ▶ ...

“Best” settings depend on the problem to solve : companies investing on SAT need a generic, flexible SAT solver [A. Nadel Talk @PoS'11]

Finding the right settings for solving a problem

- ▶ Expert knowledge of both SAT solving and the initial problem
 - ▶ Pros : ideal case, adapt the solver to a specific problem
 - ▶ Cons : cannot have a SAT expert in all companies using SAT
- ▶ Use automatic solver parameters optimization (e.g. ParamILS, Prog. by Opt.)
 - ▶ Pros : independent of the class of problem, fully automated
 - ▶ Cons : consume of lot of resources, no understanding of the settings

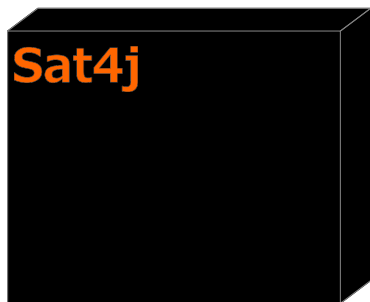
Finding the right settings for solving a problem

- ▶ Expert knowledge of both SAT solving and the initial problem
 - ▶ Pros : ideal case, adapt the solver to a specific problem
 - ▶ Cons : cannot have a SAT expert in all companies using SAT
- ▶ Use automatic solver parameters optimization (e.g. ParamILS, Prog. by Opt.)
 - ▶ Pros : independent of the class of problem, fully automated
 - ▶ Cons : consume of lot of resources, no understanding of the settings

Our approach : help the expert of the problem to better understand the effect of the SAT solver's settings on her problems

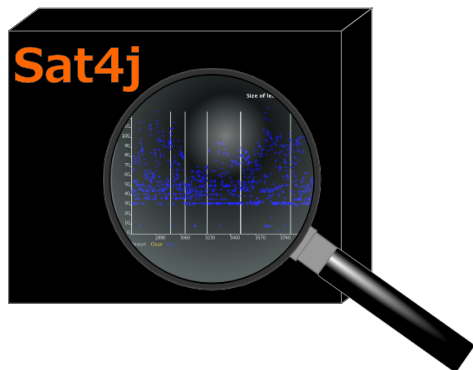
From black box to grey box

Sat4j live metrics visualization and remote control



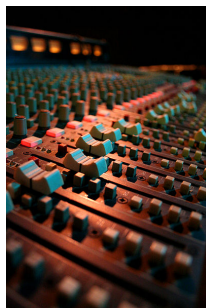
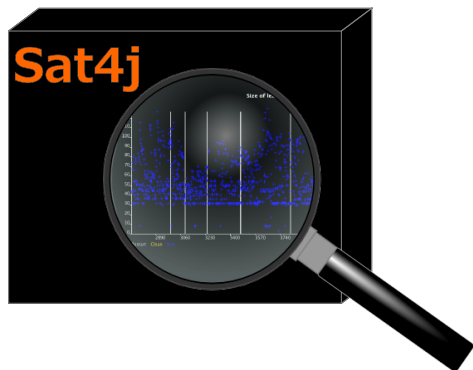
From black box to grey box

Sat4j live metrics visualization and remote control



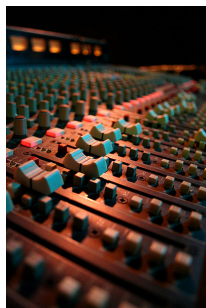
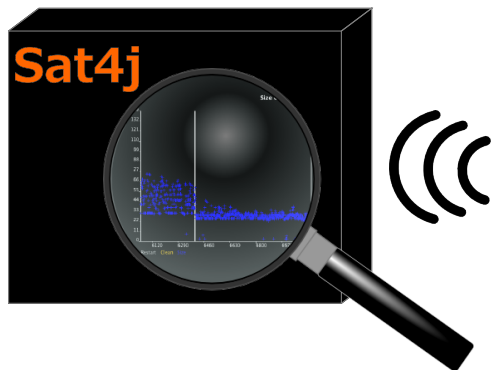
From black box to grey box

Sat4j live metrics visualization and remote control



From black box to grey box

Sat4j live metrics visualization and remote control



Motivation

Metrics

Interacting with the solver

Examples

Demo

Conclusion

Static Metrics

Sat4j on SAT-Race-Benchmarks/ibm-2002-05r-k90.cnf

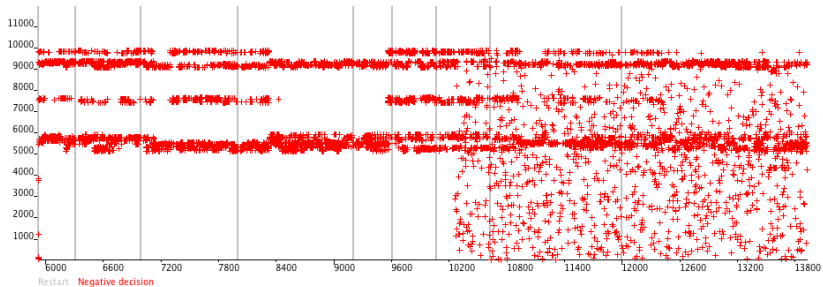
```
c starts : 53
c conflicts : 14180
c decisions : 262512
c propagations : 34406050
c inspects : 99050368
c shortcuts : 0
c learnt literals : 215
c learnt binary clauses : 1359
c learnt ternary clauses : 1526
c learnt constraints : 13965
c root simplifications : 0
c removed literals (reason simplification) : 77609
c reason swapping (by a shorter reason) : 0
c Calls to reduceDB : 2
c Number of update (reduction) of LBD : 6605
c speed (assignments/second) : 1244791.968162084
c non guided choices 69316
```

- ▶ Check which variable is selected by the heuristics
- ▶ Check the polarity used
- ▶ Bad smell? pattern, no occurrence of a polarity
- ▶ Corrective action : random walk, another heuristic or phase selection strategy

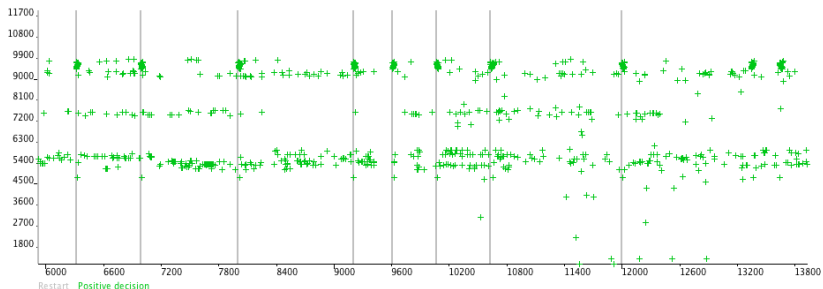


Dynamic metrics : decisions

Negative decision phases

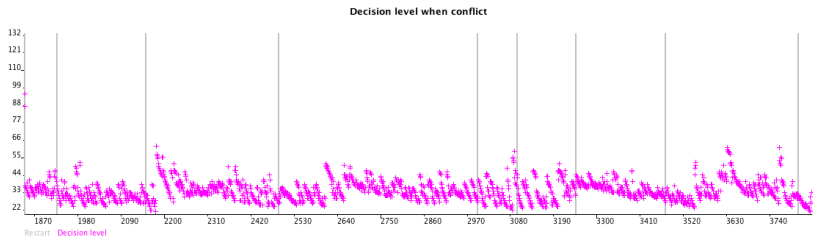


Positive decision phases



Dynamic metrics : decisions level

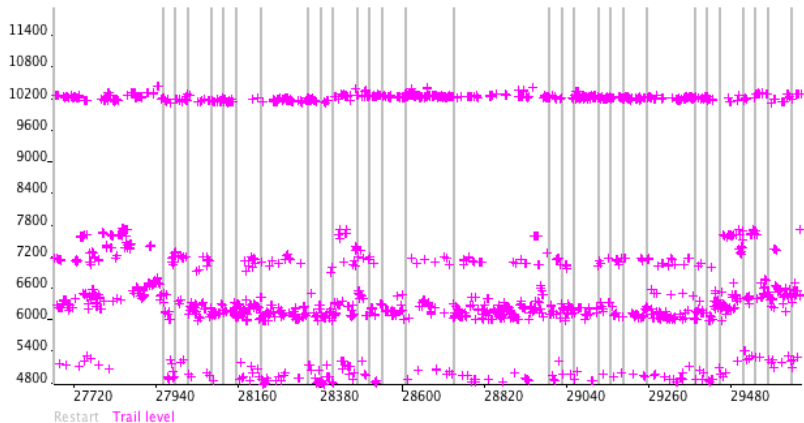
- ▶ Shows how many decisions are needed before reaching a conflict
- ▶ Bad smell? does not decrease over time
- ▶ Corrective action : restarts, ...



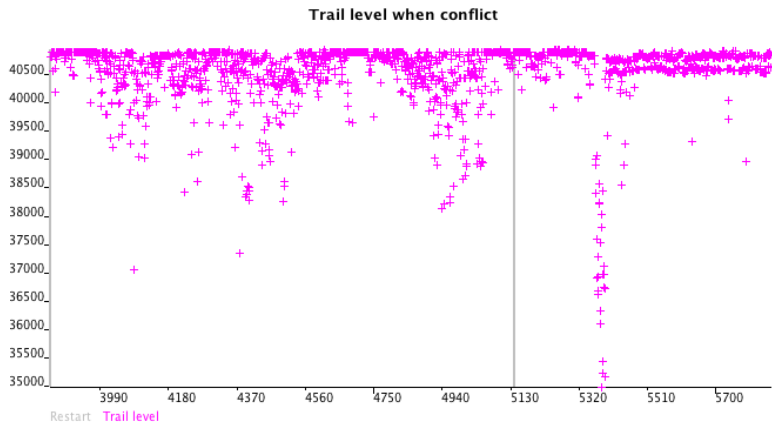
Dynamic metrics : trail level

- ▶ Shows how many variable are assigned before reaching a conflict
- ▶ Bad smell ? most variables assigned : generate and test
- ▶ Corrective action : change the model ...

Trail level when conflict

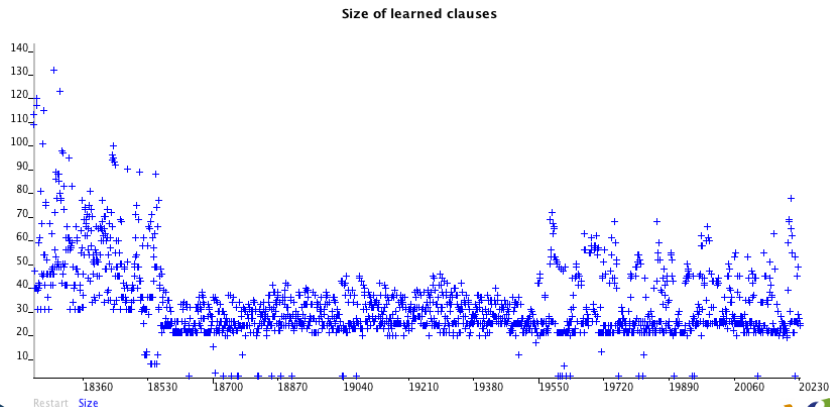


Dynamic metrics : trail level bad smell



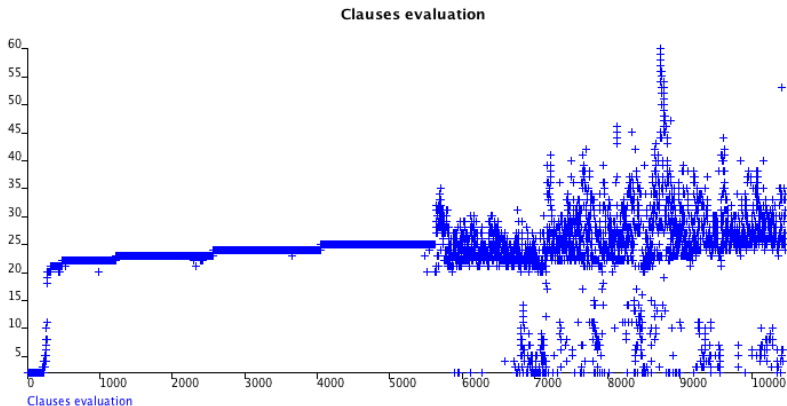
Dynamic metrics : size of learned clauses

- ▶ Shows the size of the clauses learned after conflict analysis
- ▶ Great to witness the efficiency of conflict clause minimization
- ▶ Bad smell ? no changes when different strategies are used
- ▶ Corrective action : disable clause minimization



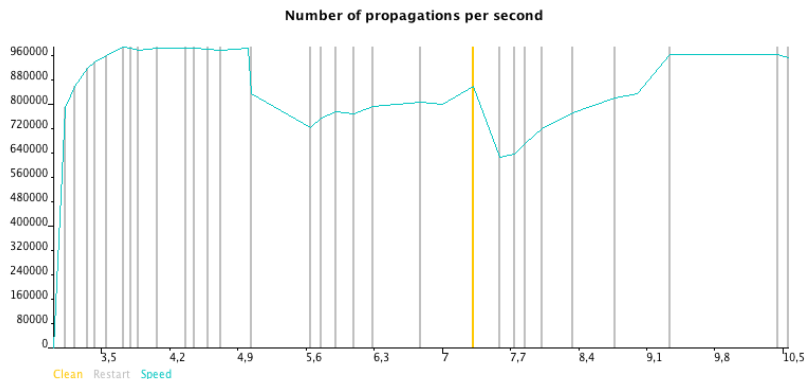
Dynamic metrics : score of learned clauses

- ▶ Shows the score of the clauses learned for deletion policy
- ▶ Best old clauses first, then new ones (unordered)
- ▶ Bad smell ? all the clauses are good, deletion policy is too aggressive
- ▶ Corrective action : change cleanup strategy



Dynamic metrics : speed (velocity) of the solver

- ▶ Displays the number of assignment per second
- ▶ Bad smell ? solver velocity is seriously decreasing
- ▶ Corrective action : cleanup learned clauses



Theory does not meet practice here !

Motivation

Metrics

Interacting with the solver

Examples

Demo

Conclusion

How to apply corrective actions

- ▶ **Stopping the solver and change the settings** : disabling conflict minimization for instance.
- ▶ **When the solver is running**

In the latter case :

- ▶ Need to allow the user to control the solver
 - ▶ restart
 - ▶ cleanup learned clauses
- ▶ Need to allow the user to change the settings of the solver on-the-fly
 - ▶ clause minimization strategy (NONE, SIMPLE, EXPENSIVE from Minisat 1.13)
 - ▶ phase selection strategy (Positive, Negative, Random, RSAT)
 - ▶ restart strategy (MiniSAT, Armin, Luby, **Glucose21**)
 - ▶ cleanup strategy (conservative/activity, aggressive/lbd)
 - ▶ random walk (random selection of an unassigned variable)

- ▶ Main features of the CDCL engine in Sat4j are customizable thanks to the **strategy design pattern**
- ▶ The remote control implements those strategies
- ▶ Sat4j CDCL engine calls listeners in case of specific events (propagation, conflicts, etc)
- ▶ Each metric is managed by a listener
- ▶ Small modifications in the code to allow the settings to be changed while the solver is running : initialization, concurrency.
- ▶ Two visualizations available :
 - ▶ gnuplot : data is logged into text files, displayed by gnuplot 4.6 (clumsy but “efficient”)
 - ▶ jchart2d : pure java solution (default, runs out-of-the-box, slow right now)

Motivation

Metrics

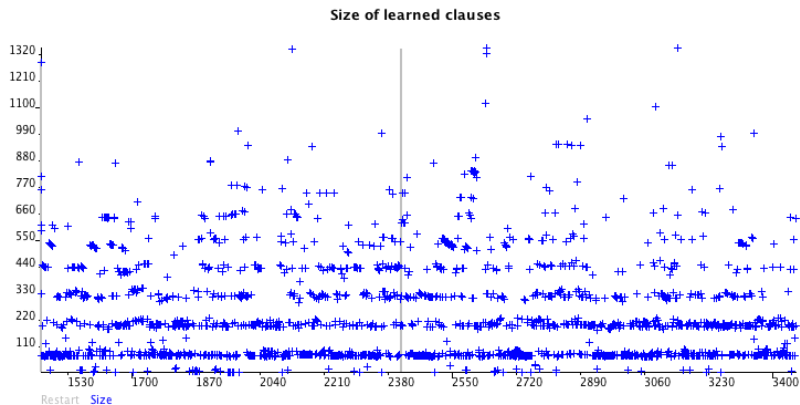
Interacting with the solver

Examples

Demo

Conclusion

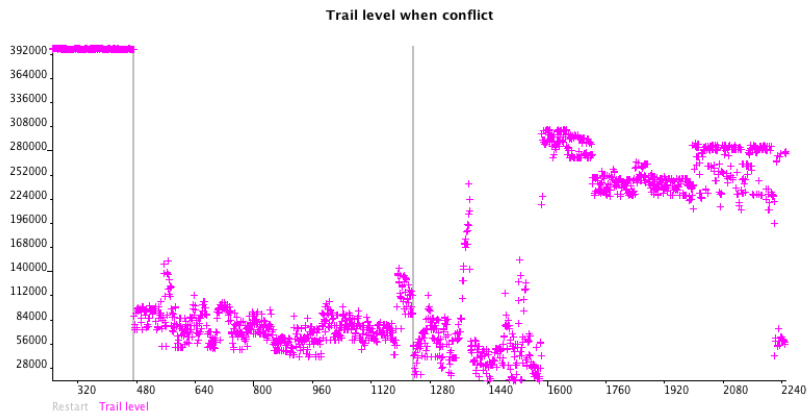
Pattern in the size of learned clauses



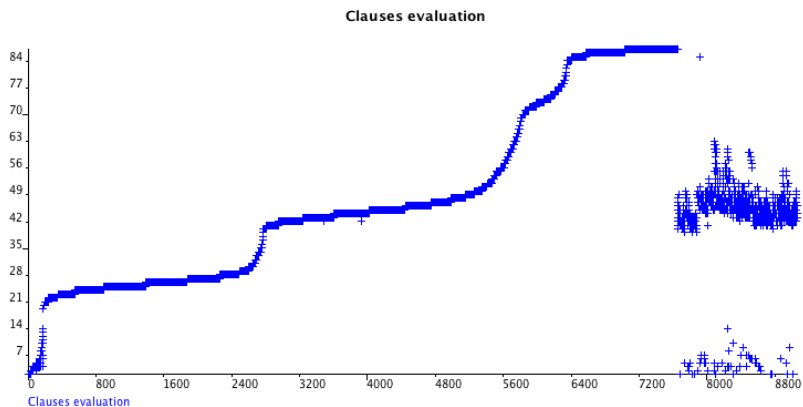
Pattern in the trail Level



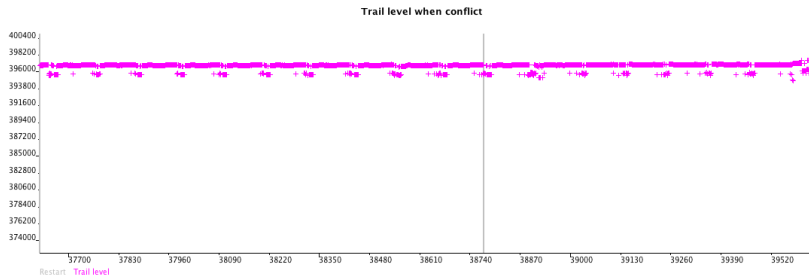
Pattern in the trail Level again



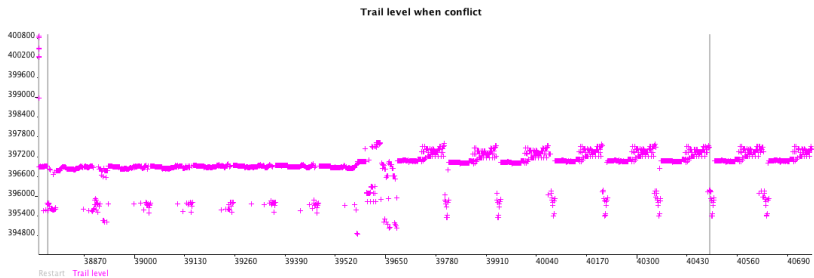
Pattern in the learned clauses evaluation



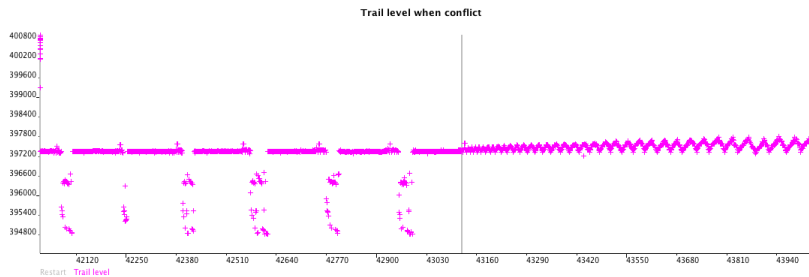
Saturday night fever !



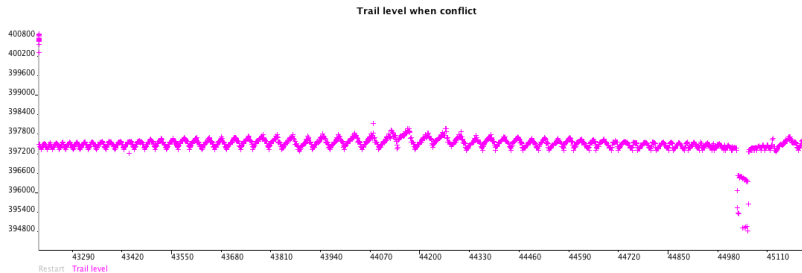
Saturday night fever !



Saturday night fever !



Saturday night fever !



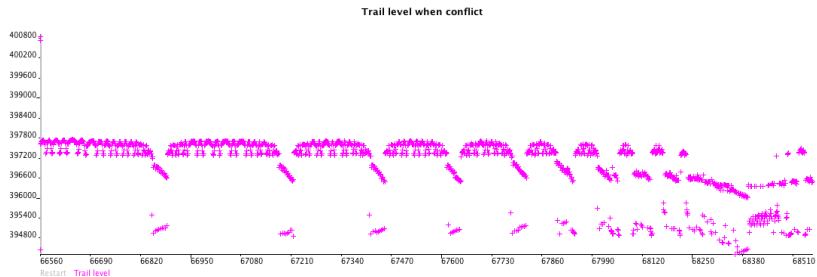
Saturday night fever !



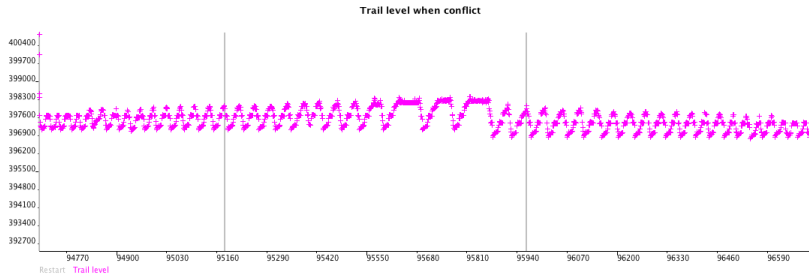
Saturday night fever !



Saturday night fever !



Saturday night fever !



Motivation

Metrics

Interacting with the solver

Examples

Demo

Conclusion

- ▶ Pickup the tool from
<http://www.satcompetition.org/PoS12/sat4j-sat.jar>
- ▶ `java -jar sat4j-sat.jar -remote -r yourfavoriteinstance.cnf`

Motivation

Metrics

Interacting with the solver

Examples

Demo

Conclusion

Conclusion

- ▶ Presented some metrics to observe live to better understand what the solver is doing
- ▶ Proposed some corrective actions against bad smell on those metrics
- ▶ Two implementations available on top of Sat4j : one in pure java (jchard2d) and one using gnuplot 4.6
- ▶ Proven useful for us (bug detection) and one of our users (better settings)
- ▶ Could be useful to teach CDCL in the classroom or tutorials
- ▶ New type of SAT solver : **User-Driven Clause Learning solver**
- ▶ Main drawback : the solver is at best 10 times slower ...

- ▶ Find a way to accurately monitor solver's velocity
- ▶ Suggested by a reviewer : logging the actions on the remote control for post analysis
- ▶ To go further : **allow a solver to replay a scenario**
- ▶ Study more problems to discover more patterns, faulty behavior
- ▶ Provides a strategy to act when a given pattern is recognized

Thanks for your attention

<http://www.sat4j.org/>



Questions ?



31/31



UNIVERSITÉ D'ARTOIS