# SAT solver essentials, SAT modeling
# Incremental SAT

**Gilles Audemard**

VTSA School - Liege - 2021

Thanks to N. Szczepanski and L. Simon

# Incremental SAT solving : an Introduction

- A surprising effect of solvers' efficiency : used as NP-Complete oracles
  - IC3 : thousands of calls on *simple* formulas [Bradley 2012]
  - MUS extraction [Belov etal. 2012]
  - MaxSAT
- Many calls on similar instances
- CDCL solvers learn form the PAST ! !

**Keep the solver alive**

# Minimum Unsatisfiable Subformula

| | | |
|---|---|---|
| $x \vee y \vee z$ | $x \vee \neg y$ | $x \vee \neg z$ |
| $\neg x \vee y \vee z$ | $x \vee w$ | $w \vee z \vee \neg y$ |
| $\neg x \vee \neg y$ | $\neg x \vee \neg z$ | $w \vee \neg x \vee \neg z$ |
| | UNSAT | |

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses

# Minimum Unsatisfiable Subformula

| | | |
|---|---|---|
| $x \vee y \vee z$ | $x \vee \neg y$ | $x \vee \neg z$ |
| $\neg x \vee y \vee z$ | $x \vee w$ | $w \vee z \vee \neg y$ |
| $\neg x \vee \neg y$ | $\neg x \vee \neg z$ | $w \vee \neg x \vee \neg z$ |

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses

# Minimum Unsatisfiable Subformula

| | | |
|---|---|---|
| $x \vee y \vee z$ | $x \vee \neg y$ | $x \vee \neg z$ |
| $\neg x \vee y \vee z$ | $x \vee w$ | $w \vee z \vee \neg y$ |
| $\neg x \vee \neg y$ | $\neg x \vee \neg z$ | $w \vee \neg x \vee \neg z$ |

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

# Minimum Unsatisfiable Subformula

$\neg x \vee y \vee z$

$\neg x \vee \neg y$

$x \vee \neg y$

$x \vee w$

$\neg x \vee \neg z$

SAT

$x \vee \neg z$

$w \vee z \vee \neg y$

$w \vee \neg x \vee \neg z$

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

# Minimum Unsatisfiable Subformula

| | | |
|---|---|---|
| $x \vee y \vee z$ | $x \vee \neg y$ | $x \vee \neg z$ |
| $\neg x \vee y \vee z$ | $x \vee w$ | $w \vee z \vee \neg y$ |
| $\neg x \vee \neg y$ | $\neg x \vee \neg z$ | $w \vee \neg x \vee \neg z$ |

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

# Minimum Unsatisfiable Subformula

| | | |
|---|---|---|
| $x \vee y \vee z$ | $x \vee \neg y$ | $x \vee \neg z$ |
| $\neg x \vee y \vee z$ | $x \vee w$ | $w \vee z \vee \neg y$ |
| $\neg x \vee \neg y$ | $\neg x \vee \neg z$ | $w \vee \neg x \vee \neg z$ |

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

# Minimum Unsatisfiable Subformula

| | | |
|---|---|---|
| $x \vee y \vee z$ | $x \vee \neg y$ | $x \vee \neg z$ |
| $\neg x \vee y \vee z$ | | $w \vee z \vee \neg y$ |
| $\neg x \vee \neg y$ | $\neg x \vee \neg z$ | $w \vee \neg x \vee \neg z$ |
| | UNSAT | |

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

# Minimum Unsatisfiable Subformula

| | | |
|---|---|---|
| $x \vee y \vee z$ | $x \vee \neg y$ | $x \vee \neg z$ |
| $\neg x \vee y \vee z$ | | |
| $\neg x \vee \neg y$ | $\neg x \vee \neg z$ | $w \vee \neg x \vee \neg z$ |
| | UNSAT | |

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

# Minimum Unsatisfiable Subformula

| | | |
|---|---|---|
| $x \vee y \vee z$ | $x \vee \neg y$ | $x \vee \neg z$ |
| $\neg x \vee y \vee z$ | | |
| $\neg x \vee \neg y$ | $\neg x \vee \neg z$ | |

MUS !

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

# Working with Assumptions

- A formula $F$
- A set of assumptions, $\ell_1, \ell_2, \ldots, \ell_n$ with $\ell_i$ are (*fresh*) literals
- Solve $F \wedge \ell_1 \wedge \ell_2 \ldots \wedge \ell_n$
- Incremental SAT solving : the process can be repeated with new assumptions

# Working with Assumptions

- A formula $F$
- A set of assumptions, $\ell_1, \ell_2, \ldots, \ell_n$ with $\ell_i$ are (*fresh*) literals
- Solve $F \wedge \ell_1 \wedge \ell_2 \ldots \wedge \ell_n$
- Incremental SAT solving : the process can be repeated with new assumptions

**First solution**

- Simplify : $F' = F \wedge \ell_1 \wedge \ell_2 \ldots \wedge \ell_n$
- Solve $F'$
- Learnt clauses can not be kept

# Working with Assumptions

- A formula $F$
- A set of assumptions, $\ell_1, \ell_2, \ldots, \ell_n$ with $\ell_i$ are (*fresh*) literals
- Solve $F \wedge \ell_1 \wedge \ell_2 \ldots \wedge \ell_n$
- Incremental SAT solving : the process can be repeated with new assumptions

# Working with Assumptions

- A formula $F$
- A set of assumptions, $\ell_1, \ell_2, \ldots, \ell_n$ with $\ell_i$ are (*fresh*) literals
- Solve $F \wedge \ell_1 \wedge \ell_2 \ldots \wedge \ell_n$
- Incremental SAT solving : the process can be repeated with new assumptions

## Second Solution

- First, selects all assumptions as decision variables :
  one level => one assumption
- Second, Run the SAT solver as usual
- All learnt clauses can be kept
- One can explain unsatisfiability wrt set of assumptions !

# Forget some clauses

- Add one selector (fresh variable) $\ell_i$ per clause

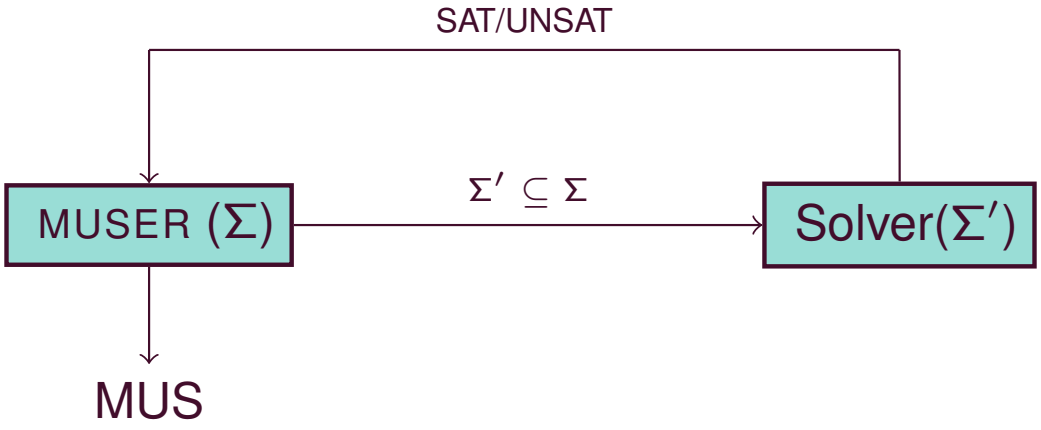
$$\ell_1 \vee x \vee y \vee z \qquad \ell_2 \vee x \vee \neg y \qquad \ell_3 \vee x \vee \neg z$$
$$\ell_4 \vee \neg x \vee y \vee z \qquad \ell_5 \vee x \vee w \qquad \ell_6 \vee w \vee z \vee \neg y$$
$$\ell_7 \vee \neg x \vee \neg y \qquad \ell_8 \vee \neg x \vee \neg z \qquad \ell_9 \vee w \vee \neg x \vee \neg z$$

$$\ell_1 \vee x \vee y \vee z \qquad \ell_2 \vee x \vee \neg y$$
$$\ell_1 \vee \ell_2 \vee x \vee z$$

- Learnt clause contains selectors of all original clauses used to generate it

# Muser Architecture

## Incremental SAT




- One of the best MUS extractor
- Successive calls to a SAT oracle
- Non independent calls
- Informations between two calls are preserved
  - Heuristics : VSIDS, phase saving, restarts...
  - **Learnt clauses**

# Forget Some Clauses

- Assign $\ell_i$ (as an assumption) to false to activate the clause $i$
- Assign $\ell_j$ (as an assumption) to true to disable the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too !

$\ell_1 \vee x \vee y \vee z$
$\ell_2 \vee x \vee \neg y$
$\ell_3 \vee x \vee \neg z$
$\ell_4 \vee \neg x \vee y \vee z$
$\ell_5 \vee x \vee w$
$\ell_6 \vee w \vee z \vee \neg y$
$\ell_7 \vee \neg x \vee \neg y$
$\ell_8 \vee \neg x \vee \neg z$
$\ell_9 \vee w \vee \neg x \vee \neg z$

$\ell_1 \vee \ell_2 \vee x \vee z$

# Forget Some Clauses

- Assign $\ell_i$ (as an assumption) to false to activate the clause $i$
- Assign $\ell_j$ (as an assumption) to true to disable the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too !

$\ell_1 \vee x \vee y \vee z$
$\ell_2 \vee x \vee \neg y$
$\ell_3 \vee x \vee \neg z$
$\ell_4 \vee \neg x \vee y \vee z$
$\ell_5 \vee x \vee w$
$\ell_6 \vee w \vee z \vee \neg y$
$\ell_7 \vee \neg x \vee \neg y$
$\ell_8 \vee \neg x \vee \neg z$
$\ell_9 \vee w \vee \neg x \vee \neg z$

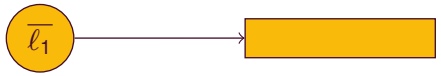
$\ell_1 \vee \ell_2 \vee x \vee z$

# Forget Some Clauses

- Assign $\ell_i$ (as an assumption) to false to activate the clause $i$
- Assign $\ell_j$ (as an assumption) to true to disable the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too !

$$
\begin{aligned}
& x \vee y \vee z \\
\ell_2 \vee & x \vee \neg y \\
\ell_3 \vee & x \vee \neg z \\
\ell_4 \vee & \neg x \vee y \vee z \\
\ell_5 \vee & x \vee w \\
\ell_6 \vee & w \vee z \vee \neg y \\
\ell_7 \vee & \neg x \vee \neg y \\
\ell_8 \vee & \neg x \vee \neg z \\
\ell_9 \vee & w \vee \neg x \vee \neg z \\
\\
\ell_1 \vee \ell_2 \vee & x \vee z
\end{aligned}
$$

DL 1



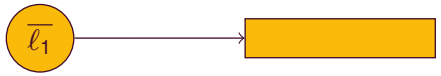
# Forget Some Clauses

- Assign $\ell_i$ (as an assumption) to false to activate the clause $i$
- Assign $\ell_j$ (as an assumption) to true to disable the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too !

$$
\begin{aligned}
& x \vee y \vee z \\
\ell_2 \vee & x \vee \neg y \\
\ell_3 \vee & x \vee \neg z \\
\ell_4 \vee & \neg x \vee y \vee z \\
\ell_5 \vee & x \vee w \\
\ell_6 \vee & w \vee z \vee \neg y \\
\ell_7 \vee & \neg x \vee \neg y \\
\ell_8 \vee & \neg x \vee \neg z \\
\ell_9 \vee & w \vee \neg x \vee \neg z
\end{aligned}
$$

$$\ell_2 \vee x \vee z$$

DL 1

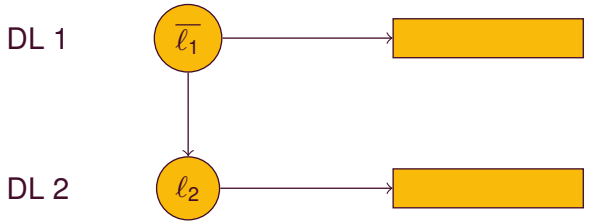


# Forget Some Clauses

- Assign $\ell_i$ (as an assumption) to false to activate the clause $i$
- Assign $\ell_j$ (as an assumption) to true to disable the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too !

$$
\begin{aligned}
& x \vee y \vee z \\
\ell_2 \vee & x \vee \neg y \\
\ell_3 \vee & x \vee \neg z \\
\ell_4 \vee & \neg x \vee y \vee z \\
\ell_5 \vee & x \vee w \\
\ell_6 \vee & w \vee z \vee \neg y \\
\ell_7 \vee & \neg x \vee \neg y \\
\ell_8 \vee & \neg x \vee \neg z \\
\ell_9 \vee & w \vee \neg x \vee \neg z
\end{aligned}
$$

$$\ell_2 \vee x \vee z$$



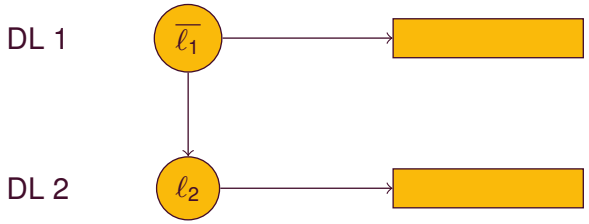
# Forget Some Clauses

- Assign $\ell_i$ (as an assumption) to false to activate the clause $i$
- Assign $\ell_j$ (as an assumption) to true to disable the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too !

$$x \vee y \vee z$$

$$\ell_3 \vee x \vee \neg z$$
$$\ell_4 \vee \neg x \vee y \vee z$$
$$\ell_5 \vee x \vee w$$
$$\ell_6 \vee w \vee z \vee \neg y$$
$$\ell_7 \vee \neg x \vee \neg y$$
$$\ell_8 \vee \neg x \vee \neg z$$
$$\ell_9 \vee w \vee \neg x \vee \neg z$$

$$\ell_2 \vee x \vee z$$

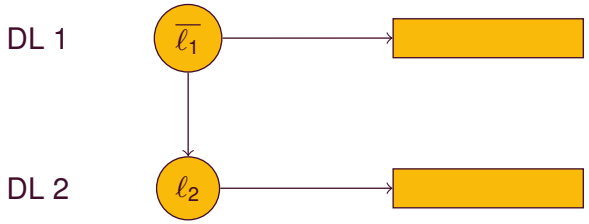


# Forget Some Clauses

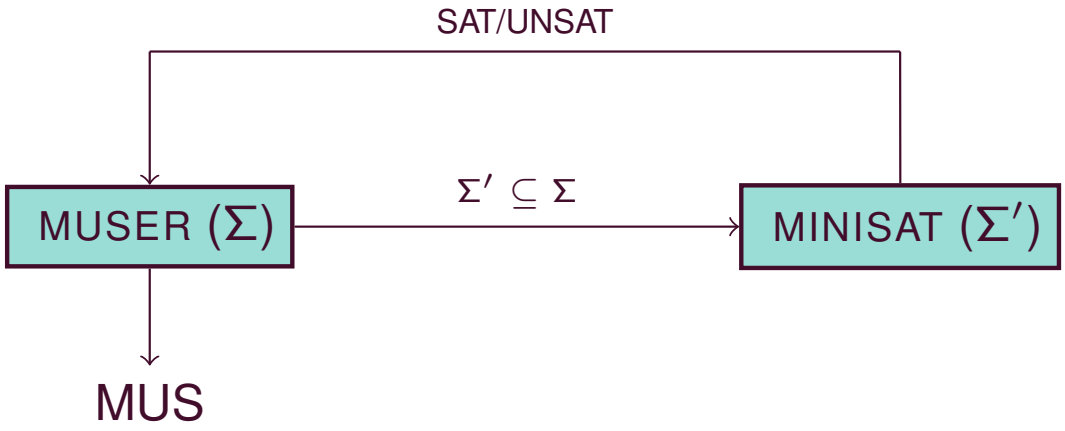
- Assign $\ell_i$ (as an assumption) to false to activate the clause $i$
- Assign $\ell_j$ (as an assumption) to true to disable the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too !

$$x \vee y \vee z$$

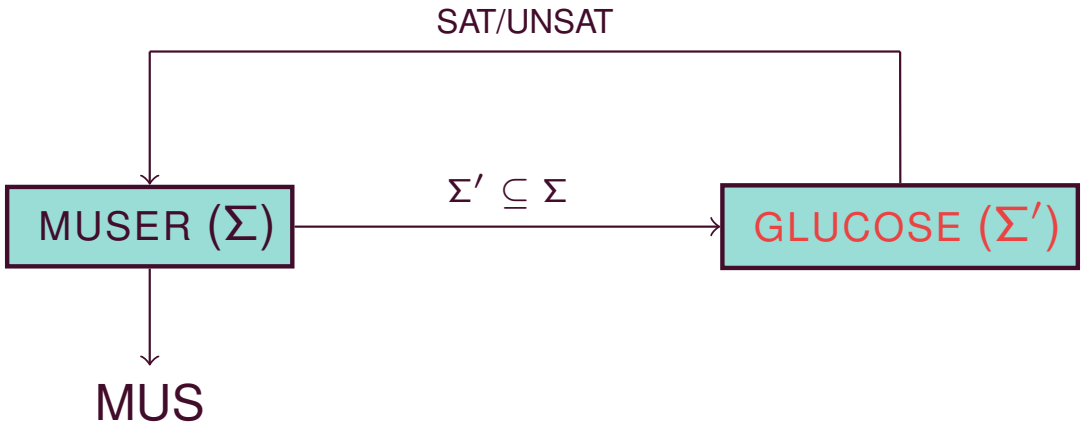
$$\ell_3 \vee x \vee \neg z$$
$$\ell_4 \vee \neg x \vee y \vee z$$
$$\ell_5 \vee x \vee w$$
$$\ell_6 \vee w \vee z \vee \neg y$$
$$\ell_7 \vee \neg x \vee \neg y$$
$$\ell_8 \vee \neg x \vee \neg z$$
$$\ell_9 \vee w \vee \neg x \vee \neg z$$



# Glucose inside Muser



# Glucose inside Muser



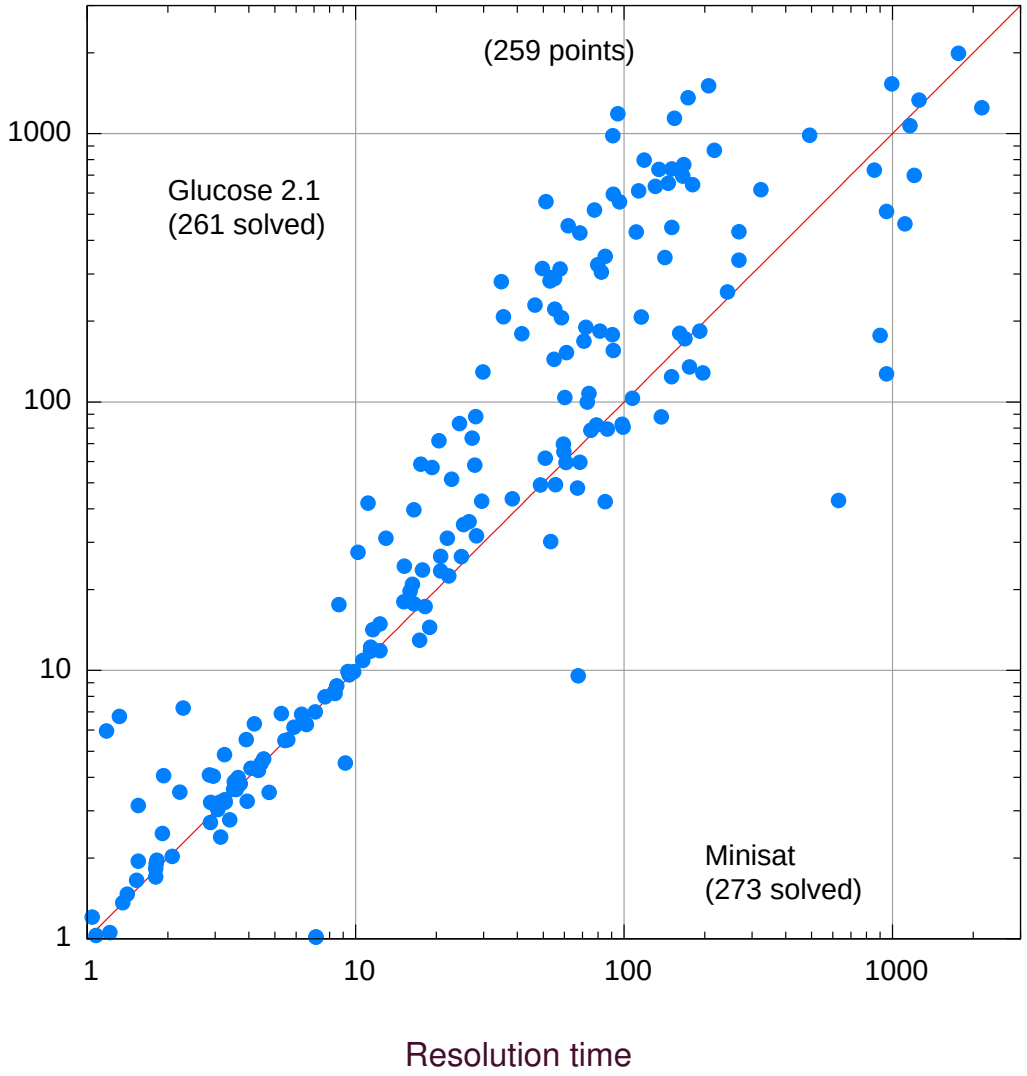

- Plug GLUCOSE in MUSER
- Adapt and modify GLUCOSE to improve MUSER performances

Improve SAT oracle in order to improve the MUSER tool

# Test Set

- 300 instances from the SAT competition 2011, MUS category
- timeout set to 2400 seconds
- MUSER is used with default options (destructive approach, model rotation)

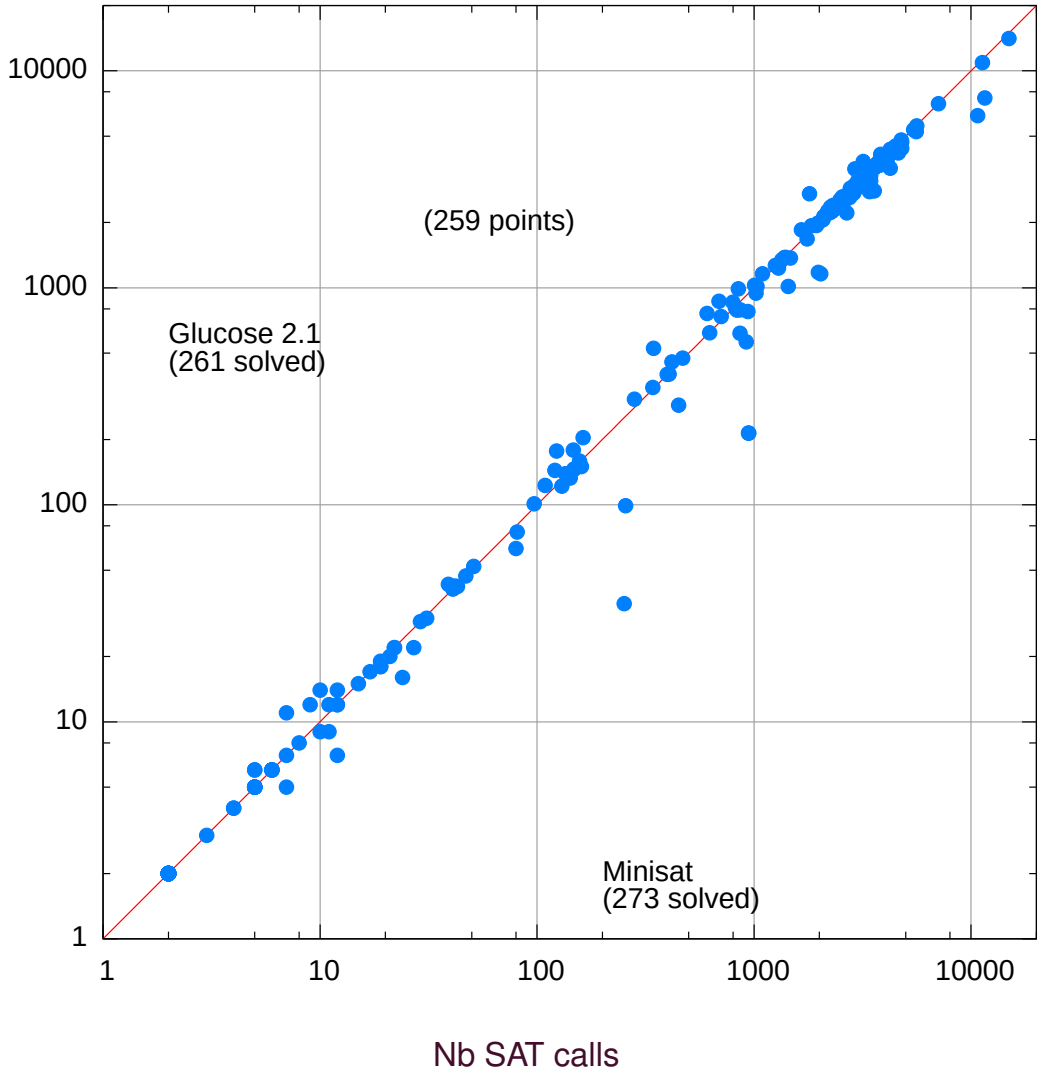
# A First Attempt



# Disappointing Results

Trying to explain these bad results

# Disappointing Results



# Disappointing Results

Trying to explain these bad results

- Comparable number of oracle calls
- Easy SAT calls
- Difficult UNSAT ones
- GLUCOSE is supposed to be good on UNSAT formulas

# Disappointing Results

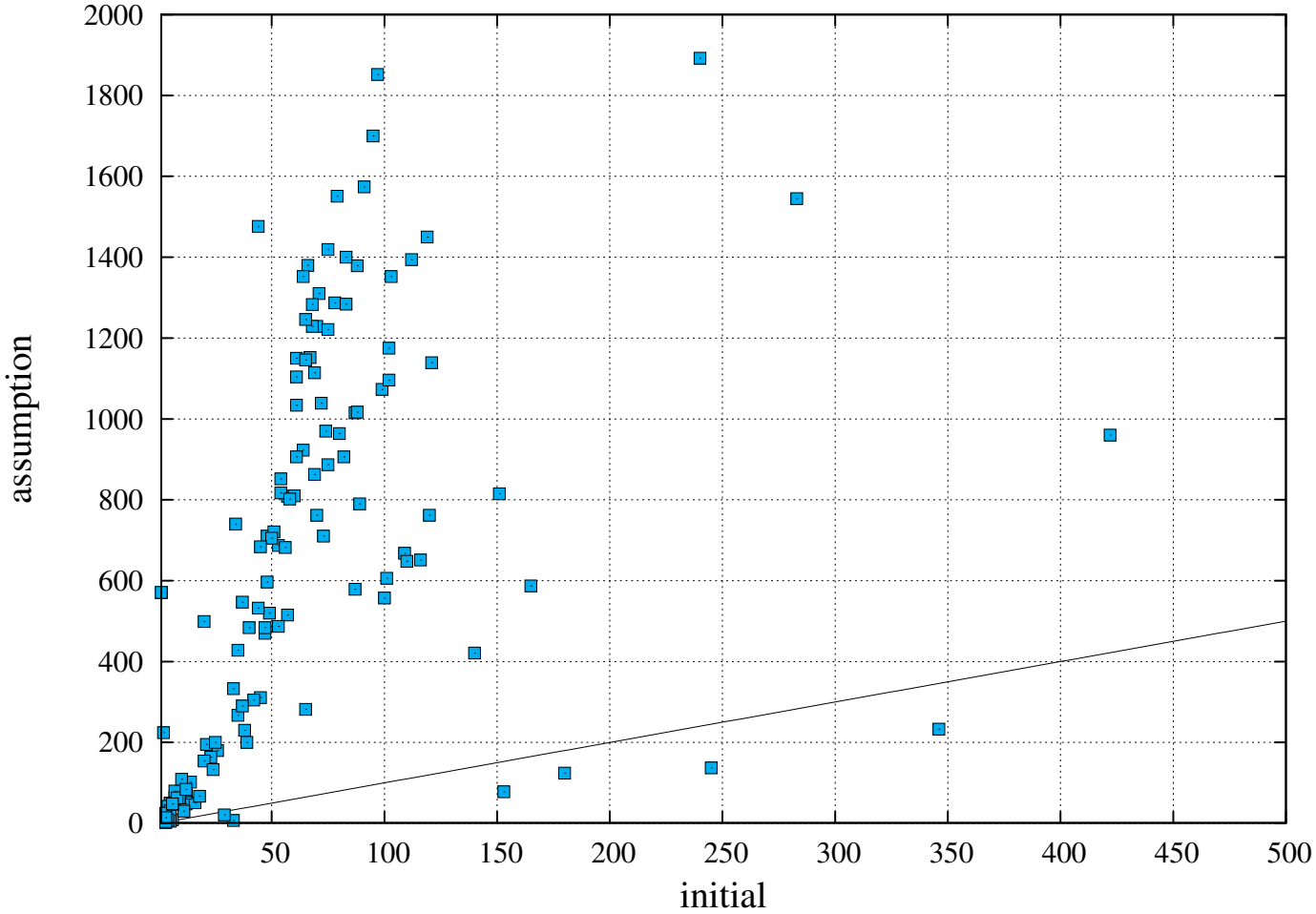
Trying to explain these bad results

- Comparable number of oracle calls
- Easy SAT calls
- Difficult UNSAT ones
- GLUCOSE is supposed to be good on UNSAT formulas

- GLUCOSE uses LBD for cleaning, restarts...
- Each assumption uses its own decision level

# Disappointing Results

- Each point represents an instance
- x-axis is the average number of initial variables in learnt clauses
- y-axis is the average number of selector variables in learnt clauses



# Disappointing Results

| | | LBD | | | | |
|---|---|---|---|---|---|---|
| | | | size | | LBD | |
| Instance | #C | time | avg | max | avg | max |
| fdmus_b21_96 | 8541 | 29 | 1145 | 5980 | 1095 | 5945 |
| longmult6 | 8853 | 46 | 694 | 3104 | 672 | 3013 |
| dump_vc950 | 360419 | 110 | 522 | 36309 | 498 | 35873 |
| g7n | 70492 | 190 | 1098 | 16338 | 1049 | 16268 |

- LBD looks like size
- Clauses are very long

# Disappointing Results

Trying to explain these bad results

- Comparable number of oracle calls
- Easy SAT calls
- Difficult UNSAT ones
- GLUCOSE is supposed to be good on UNSAT formulas

- GLUCOSE uses LBD for cleaning, restarts...
- Each assumption uses its own decision level
- **The LBD of a clause looks like its size !**

# Disappointing Results

Trying to explain these bad results

- Comparable number of oracle calls
- Easy SAT calls
- Difficult UNSAT ones
- GLUCOSE is supposed to be good on UNSAT formulas

- GLUCOSE uses LBD for cleaning, restarts...
- Each assumption uses its own decision level
- **The LBD of a clause looks like its size !**

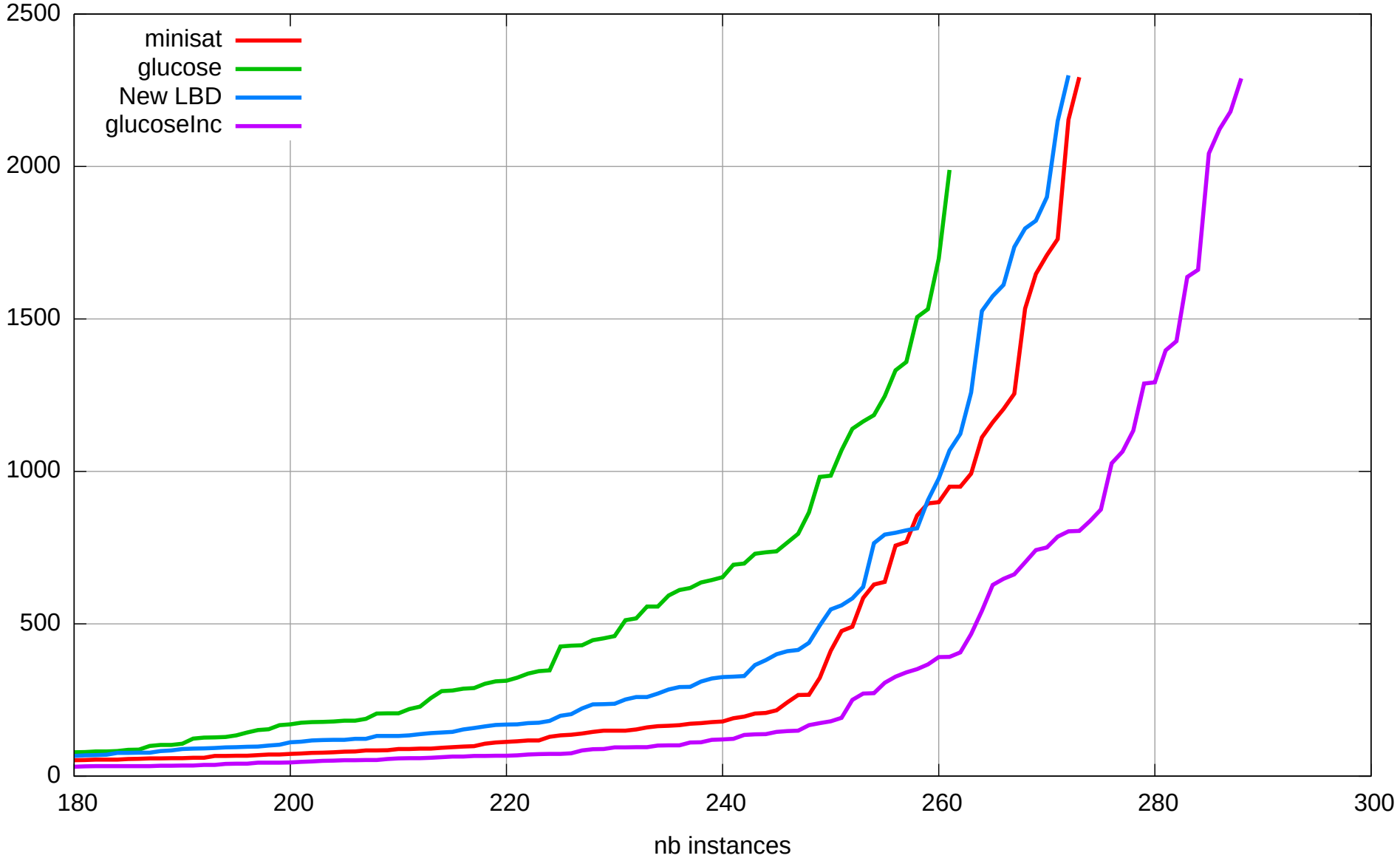
**Refine LBD** : Do not take into account selectors

# Clauses are too Long

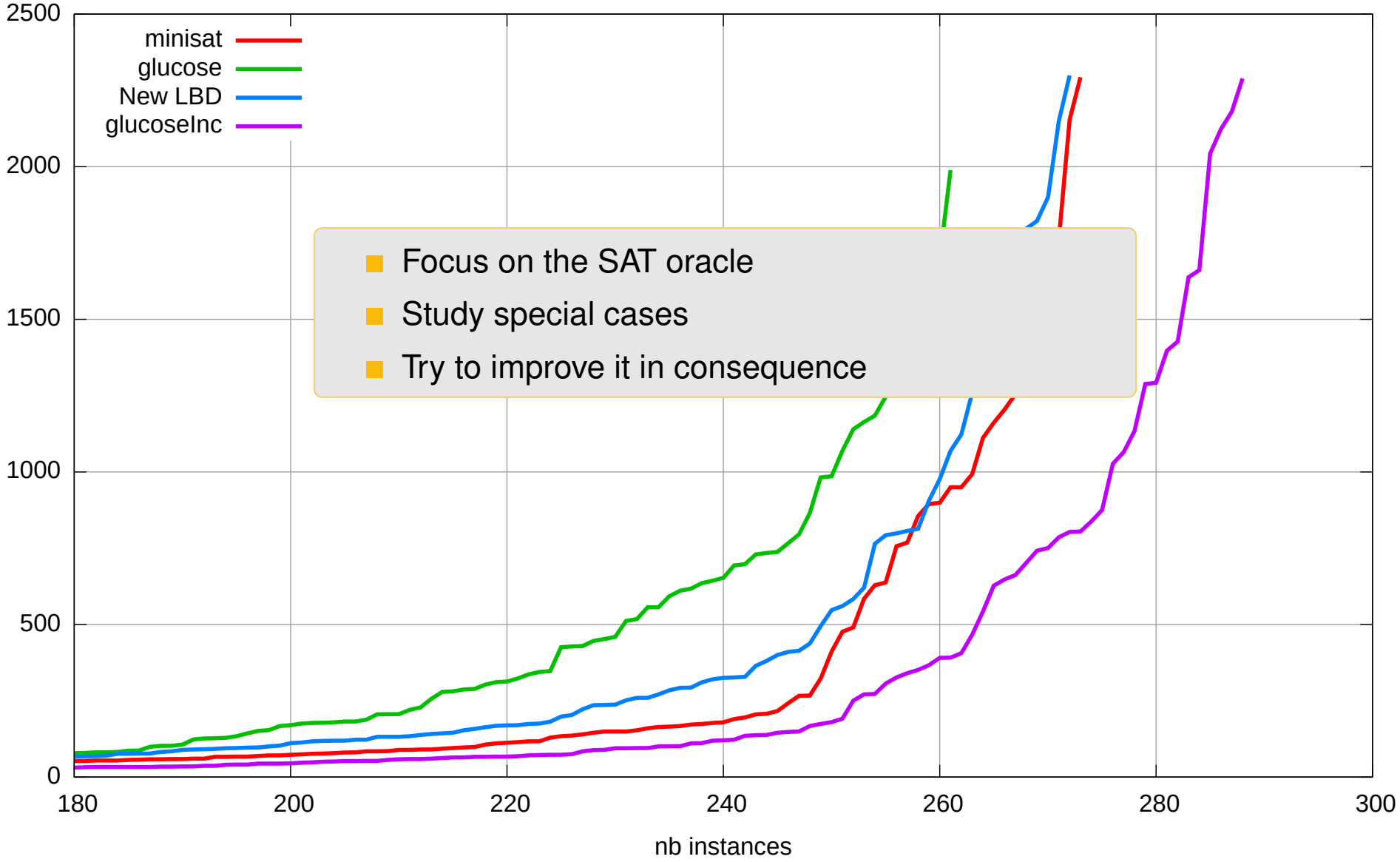
Many algorithms need to traverse clauses

- Dynamic computing of LBD (useful but costly)
  - → Store the number of selectors in the clause
  - → Stop when all initial literals have been tested
- Conflict analysis
  - → Force initial literals to be placed at the beginning
- Unit propagation
  - → Look for a non selector literal or a satisfied one
  - → Push selectors at the end of the clause
- Deleting satisfiable clauses
  - → Take only watched literals into account

# Comparison


minisat
glucose
New LBD
glucoseInc
2500
2000
1500
1000
500
0
180
200
220
240
260
280
300
nb instances

# Comparison




- Focus on the SAT oracle
- Study special cases
- Try to improve it in consequence

- Take a look at a CDCL solver (the essentials of Minisat in fact) : assumptions branch

# Exercise
# A simple MUS extractor