SAT modulo the theory of linear arithmetic: Exact, inexact and commercial solvers

Germain Faure, Robert Nieuwenhuis,

Albert Oliveras and Enric Rodríguez-Carbonell

11th International Conference, SAT 2008

Guangzhou, China

May 14th, 2008

Departament de Llenguatges i Sistemes Informatics

UNIVERSITAT POLITECNICA DE CATALUNYA

Overview of the talk

- SAT Modulo Theories (SMT)
  - DPLL(T) = Boolean engine + T-Solver
  - What is needed from T-Solver?
- Use of OR solvers for DPLL(LA)
  - Existing and non-existing functionalities
  - Adapting OR solvers
- Experimental evaluation
- New prospects and conclusions

Overview of the talk

- SAT Modulo Theories (SMT)
  - DPLL(T) = Boolean engine + T-Solver
  - What is needed from T-Solver?
- Use of OR solvers for DPLL(LA)
  - Existing and non-existing functionalities
  - Adapting OR solvers
- Experimental evaluation
- New prospects and conclusions

UPC
Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

SAT Modulo Theories (SMT)

- Some problems are more naturally expressed in other logics than propositional logic, e.g:
  - Software verification needs reasoning about equality, arithmetic, data structures, ...
- SMT consists of deciding the satisfiability of a (ground) FO formula with respect to a background theory
- Example (Equality with Uninterpreted Functions – EUF):

$$g(a)=c \;\wedge\; (\, f(g(a))\neq f(c) \vee g(a)=d \,) \;\wedge\; c\neq d$$

- Wide range of applications:
  - Predicate abstraction
  - Model checking
  - Equivalence checking
  - Static analysis
  - Scheduling
  - ...

UPC
Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

The Theory of Linear Arithmetic

- Plenty of applications:
  - System verification
  - Scheduling and planning
  - ...

The Theory of Linear Arithmetic

- Plenty of applications:
  - System verification
  - Scheduling and planning
  - ...
- Several variants:
  - $\mathbb{R}$ / $\mathbb{Z}$ / mixed linear arithmetic
  - First-order quantifier free / quantified formulas
  - Difference logic ($x - y \leq 4$)
    / UTVPI constraints ($x - y \leq 2$, $x + y \leq 7$)
    / General linear constraints (e.g., $2x + y - z \leq 3$)

UPC
Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

The Theory of Linear Arithmetic

- Plenty of applications:
  - System verification
  - Scheduling and planning
  - ...
- Several variants:
  - $\mathbb{R}$ / $\mathbb{Z}$ / mixed linear arithmetic
  - First-order quantifier free / quantified formulas
  - Difference logic ($x - y \leq 4$)
    / UTVPI constraints ($x - y \leq 2,\; x + y \leq 7$)
    / General linear constraints (e.g., $2x + y - z \leq 3$)

THIS TALK: general quantifier-free formulas in $\mathbb{R}$

UPC
Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Solving SMT with DPLL(T)

Methodology:

$$\underbrace{x \leq 2}_{1} \;\wedge\; (\underbrace{x + y \geq 10}_{2} \vee \underbrace{2x + 3y \geq 30}_{3}) \;\wedge\; \underbrace{y \leq 4}_{4}$$

- SAT solver returns model $[1, 2, 4]$

UPC
Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITECNICA DE CATALUNYA

Solving SMT with DPLL(T)

Methodology:

$$\underbrace{x \leq 2}_{1} \;\wedge\; (\underbrace{x+y \geq 10}_{2} \vee \underbrace{2x+3y \geq 30}_{3}) \;\wedge\; \underbrace{y \leq 4}_{4}$$

- SAT solver returns model $[1, 2, 4]$
- Theory solver says T-inconsistent

UPC
Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITECNICA DE CATALUNYA

Solving SMT with DPLL(T)

Methodology:

$$\underbrace{x \le 2}_{1} \;\wedge\; (\underbrace{x + y \ge 10}_{2} \vee \underbrace{2x + 3y \ge 30}_{3}) \;\wedge\; \underbrace{y \le 4}_{4}$$

- SAT solver returns model $[1, 2, 4]$
- Theory solver says T-inconsistent
- Send $\{1,\; 2 \vee 3,\; 4,\; \overline{1} \vee \overline{2} \vee \overline{4}\}$ to SAT solver

Solving SMT with DPLL(T)

Methodology:

$$\underbrace{x \leq 2}_{1} \;\wedge\; (\underbrace{x+y \geq 10}_{2} \vee \underbrace{2x+3y \geq 30}_{3}) \;\wedge\; \underbrace{y \leq 4}_{4}$$

- SAT solver returns model $[1, 2, 4]$
- Theory solver says T-inconsistent
- Send $\{1,\ 2 \vee 3,\ 4,\ \overline{1} \vee \overline{2} \vee \overline{4}\}$ to SAT solver
- SAT solver returns model $[1, \overline{2}, 3, 4]$

Solving SMT with DPLL(T)

Methodology:

$$\underbrace{x \leq 2}_{1} \;\wedge\; (\underbrace{x+y \geq 10}_{2} \vee \underbrace{2x+3y \geq 30}_{3}) \;\wedge\; \underbrace{y \leq 4}_{4}$$

- SAT solver returns model $[1, 2, 4]$
- Theory solver says T-inconsistent
- Send $\{1,\ 2 \vee 3,\ 4,\ \overline{1} \vee \overline{2} \vee \overline{4}\}$ to SAT solver
- SAT solver returns model $[1, \overline{2}, 3, 4]$
- Theory solver says T-inconsistent

UPC
Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Solving SMT with DPLL(T)

Methodology:

$$\underbrace{x \le 2}_{1} \;\wedge\; (\underbrace{x+y \ge 10}_{2} \vee \underbrace{2x+3y \ge 30}_{3}) \;\wedge\; \underbrace{y \le 4}_{4}$$

- SAT solver returns model $[1, 2, 4]$
- Theory solver says T-inconsistent
- Send $\{1,\ 2 \vee 3,\ 4,\ \overline{1} \vee \overline{2} \vee \overline{4}\}$ to SAT solver
- SAT solver returns model $[1, \overline{2}, 3, 4]$
- Theory solver says T-inconsistent
- SAT solver detects $\{1,\ 2 \vee 3,\ 4,\ \overline{1} \vee \overline{2} \vee \overline{4},\ \overline{1} \vee 2 \vee \overline{3} \vee \overline{4}\}$ UNSAT

Solving SMT with DPLL(T)

Methodology:

$$\underbrace{x \le 2}_{1} \;\wedge\; (\underbrace{x+y \ge 10}_{2} \vee \underbrace{2x+3y \ge 30}_{3}) \;\wedge\; \underbrace{y \le 4}_{4}$$

- SAT solver returns model $[1,2,4]$
- Theory solver says T-inconsistent
- Send $\{1,\; 2 \vee 3,\; 4,\; \overline{1} \vee \overline{2} \vee \overline{4}\}$ to SAT solver
- SAT solver returns model $[1,\overline{2},3,4]$
- Theory solver says T-inconsistent
- SAT solver detects $\{1,\; 2 \vee 3,\; 4,\; \overline{1} \vee \overline{2} \vee \overline{4},\; \overline{1} \vee 2 \vee \overline{3} \vee \overline{4}\}$ UNSAT

Two components: Boolean engine DPLL(X) + T-Solver

UPC
Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Solving SMT with DPLL(T) (2)

Several optimizations for enhancing efficiency:

- Check T-consistency only of full prop. models (at a leaf)

Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITECNICA DE CATALUNYA

Solving SMT with DPLL(T) (2)

Several optimizations for enhancing efficiency:

- ~~Check T-consistency only of full prop. models (at a leaf)~~
- Check T-consistency of partial assignment while being built

Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITECNICA DE CATALUNYA

Solving SMT with DPLL(T) (2)

Several optimizations for enhancing efficiency:

- ~~Check T-consistency only of full prop. models (at a leaf)~~
- Check T-consistency of partial assignment while being built
- Given a T-inconsistent assignment M, add $\neg M$ as a clause

UPC
Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Solving SMT with DPLL(T) (2)

Several optimizations for enhancing efficiency:

- ~~Check T-consistency only of full prop. models (at a leaf)~~
- Check T-consistency of partial assignment while being built

- ~~Given a T-inconsistent assignment M, add $\neg M$ as a clause~~
- Given a T-inconsistent assignment M, identify a T-inconsistent subset $M_0 \subseteq M$ and add $\neg M_0$ as a clause

Solving SMT with DPLL(T) (2)

Several optimizations for enhancing efficiency:

- ~~Check T-consistency only of full prop. models (at a leaf)~~
- Check T-consistency of partial assignment while being built

- ~~Given a T-inconsistent assignment M, add $\neg M$ as a clause~~
- Given a T-inconsistent assignment M, identify a T-inconsistent subset $M_0 \subseteq M$ and add $\neg M_0$ as a clause

- Upon a T-inconsistency, add clause and restart

Solving SMT with DPLL(T) (2)

Several optimizations for enhancing efficiency:

- ~~Check T-consistency only of full prop. models (at a leaf)~~
- Check T-consistency of partial assignment while being built

- ~~Given a T-inconsistent assignment M, add $\neg M$ as a clause~~
- Given a T-inconsistent assignment M, identify a T-inconsistent subset $M_0 \subseteq M$ and add $\neg M_0$ as a clause

- ~~Upon a T-inconsistency, add clause and restart~~
- Upon a T-inconsistency, bactrack to some point where the assignment was still T-consistent

Solving SMT with DPLL(T) (2)

Several optimizations for enhancing efficiency:

- ~~Check T-consistency only of full prop. models (at a leaf)~~
- Check T-consistency of partial assignment while being built

- ~~Given a T-inconsistent assignment M, add $\neg M$ as a clause~~
- Given a T-inconsistent assignment M, identify a T-inconsistent subset $M_0 \subseteq M$ and add $\neg M_0$ as a clause

- ~~Upon a T-inconsistency, add clause and restart~~
- Upon a T-inconsistency, bactrack to some point where the assignment was still T-consistent

THIS TALK: obtain LA-solvers that are incremental, backtrackable and produce inconsistency explanations

UPC
Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Overview of the talk

- SAT Modulo Theories (SMT)
  - DPLL(T) = Boolean engine + T-Solver
  - What is needed from T-Solver?
- Use of OR solvers for DPLL(LA)
  - Existing and non-existing functionalities
  - Adapting OR solvers
- Experimental evaluation
- New prospects and conclusions

Current state of LA-solvers in SMT

- Different techniques have been tried:
  - Simplex-based procedures:
    ARGOLIB, BARCELOGIC, CVC3, MATHSAT, YICES, Z3
  - Fourier-Motzkin:
    CVC, CVCLITE, SVC
  - Automata
- So far, clear winner is Simplex
- SMT-oriented Simplex implementations are recent (3 years)
- Why not trying mature OR Linear Programming tools?

UPC
Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITECNICA DE CATALUNYA

Use of OR tools - Good news

- Provide incremental addition / removal of constraints
- Provide support for explanations of inconsistencies
- All these functionalities available through an API
- Can handle millions of constraints, millions of vars
- Many years of application to real-life problems
- These features both in commercial and publicly available solvers, e.g.:
  - Commercial: CPLEX 11
  - Publicly available: GLPK 4.25

Use of OR tools - Bad news

- Take the set of constraints:

$$\begin{array}{rcl} -x-y+u & \leq & 0 \\ -u+z & \leq & 0 \\ -t+y & \leq & 0 \end{array} \qquad \begin{array}{rcl} -11z+v+11t & \leq & 0 \\ 11x-v & \leq & -10^{-5} \\ x & \geq & 10^{-5} \end{array}$$

and give it to CPLEX 11. Answer: **SAT**.

UPC
Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Use of OR tools - Bad news

- Take the set of constraints:

$$\begin{array}{rclcrcl} -x-y+u & \leq & 0 & \quad & -11z+v+11t & \leq & 0 \\ -u+z & \leq & 0 & & 11x-v & \leq & -10^{-5} \\ -t+y & \leq & 0 & & x & \geq & 10^{-5} \end{array}$$

  and give it to CPLEX 11. Answer: **SAT**.

- Now, multiply the 3 constraints on the left by 11

UPC
Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITECNICA DE CATALUNYA

Use of OR tools - Bad news

- Take the set of constraints:

$$\begin{array}{rcl} -x-y+u & \leq & 0 \\ -u+z & \leq & 0 \\ -t+y & \leq & 0 \end{array} \qquad \begin{array}{rcl} -11z+v+11t & \leq & 0 \\ 11x-v & \leq & -10^{-5} \\ x & \geq & 10^{-5} \end{array}$$

  and give it to CPLEX 11. Answer: **SAT**.

- Now, multiply the 3 constraints on the left by 11
- Add them all except the bound on x

UPC
Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITECNICA DE CATALUNYA

Use of OR tools - Bad news

- Take the set of constraints:

$$\begin{array}{rcl} -x-y+u & \leq & 0 \\ -u+z & \leq & 0 \\ -t+y & \leq & 0 \end{array} \qquad \begin{array}{rcl} -11z+v+11t & \leq & 0 \\ 11x-v & \leq & -10^{-5} \\ x & \geq & 10^{-5} \end{array}$$

  and give it to CPLEX 11. Answer: **SAT**.

- Now, multiply the 3 constraints on the left by 11
- Add them all except the bound on x
- We obtain $0 \leq -10^{-5}$. Hence, answer should be **UNSAT!!!!**.

UPC
Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITECNICA DE CATALUNYA

Use of OR tools - Bad news

- Take the set of constraints:

$$\begin{array}{rcl} -x-y+u & \leq & 0 \\ -u+z & \leq & 0 \\ -t+y & \leq & 0 \end{array} \qquad \begin{array}{rcl} -11z+v+11t & \leq & 0 \\ 11x-v & \leq & -10^{-5} \\ x & \geq & 10^{-5} \end{array}$$

  and give it to CPLEX 11. Answer: **SAT**.

- Now, multiply the 3 constraints on the left by 11
- Add them all except the bound on x
- We obtain $0 \leq -10^{-5}$. Hence, answer should be **UNSAT!!!!**.
- What is the PROBLEM here?

UPC
Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Use of OR tools - Bad news (2)

The problem is that

- Most LP solvers work with floating-point arithmetic

In addition,

- They cannot handle natively strict inequalities
- They cannot handle natively disequalities

Last two problems, have en easy “fix”:

- Transform $< k$ into $\leq k - \varepsilon$ (ε “small”)
- Split input disequalities $x \neq y$ into $x < y \vee x > y$ (formula level)

at the expense of introducing more inaccuracies

Use of OR tools - Bad news (3)

How to fix the inaccuracies due to floating-point arithmetic?
Use an exact T-Solver (Ex-Solver in the following)

- Only two critical situations:
  1. We are at a leaf an T-Solver says model is consistent
  2. T-Solver says assignment is inconsistent

- They both have a successful solution:
  1. Check whether solution satisfies all literals.
     If not, send model to an exact solver and proceed.
  2. Ask T-Solver for an explanation and send it to Ex-Solver.
     If inconsistent **then** continue.
     Else, **If** in a leaf **then** check assignment with Ex-Solver.
     Else continue as if assignment was consistent.

UPC
Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Overview of the talk

- SAT Modulo Theories (SMT)
  - DPLL(T) = Boolean engine + T-Solver
  - What is needed from T-Solver?
- Use of OR solvers for DPLL(LA)
  - Existing and non-existing functionalities
  - Adapting OR solvers
- Experimental evaluation
- New prospects and conclusions

Experimental evaluation

Setting used:

- Used BARCELOGIC as SMT solver
- Integrated both CPLEX 11 and GLPK 4.25 (the former substantially faster)
- Used our own Simplex-like T-Solver as exact solver
- Ran experiments over the QF_LRA division of SMT-LIB (500 problems)

Questions to answer:

1. How frequent are wrong results?
2. How expensive is result checking?
3. Which is the overall gain in performance?

UPC
Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITECNICA DE CATALUNYA

Experimental evaluation - Analysis

QUESTION: How frequent are wrong results?

Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITECNICA DE CATALUNYA

Experimental evaluation - Analysis

QUESTION: How frequent are wrong results?
ANSWER: Not too frequent. In more than 75% of the benchmarks, less than 2% of wrong answers.

Experimental evaluation - Analysis

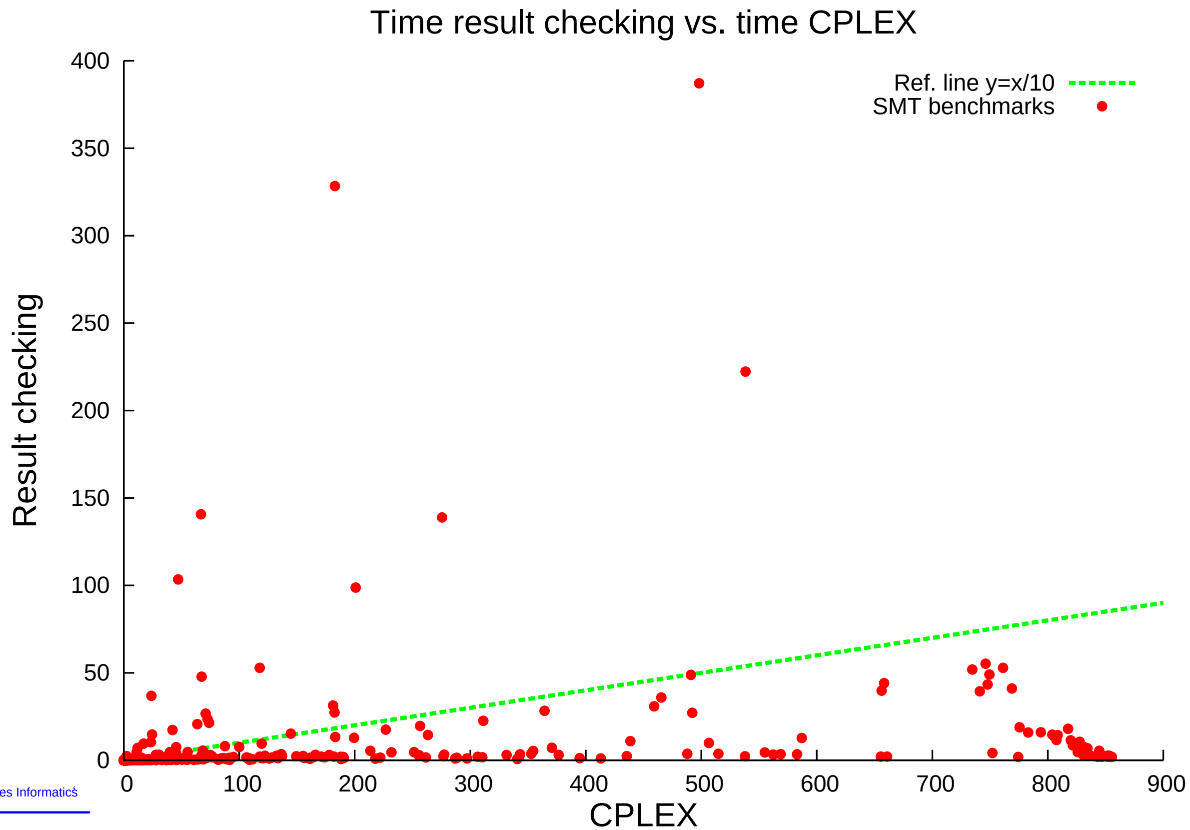
QUESTION: How frequent are wrong results?
ANSWER: Not too frequent. In more than 75% of the benchmarks, less than 2% of wrong answers.

QUESTION: How expensive is result checking?

Experimental evaluation - Analysis

QUESTION: How frequent are wrong results?
ANSWER: Not too frequent. In more than 75% of the benchmarks, less than 2% of wrong answers.

QUESTION: How expensive is result checking?
ANSWER:



UPC
Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Experimental evaluation - Analysis (2)

- So far, experiments reveal that:
  - CPLEX usually gives correct answers
  - Result checking is cheap comparing to CPLEX time
- Conclusion: replacing our exact T-Solver by CPLEX + result checking will improve performance

Experimental evaluation - Analysis (2)

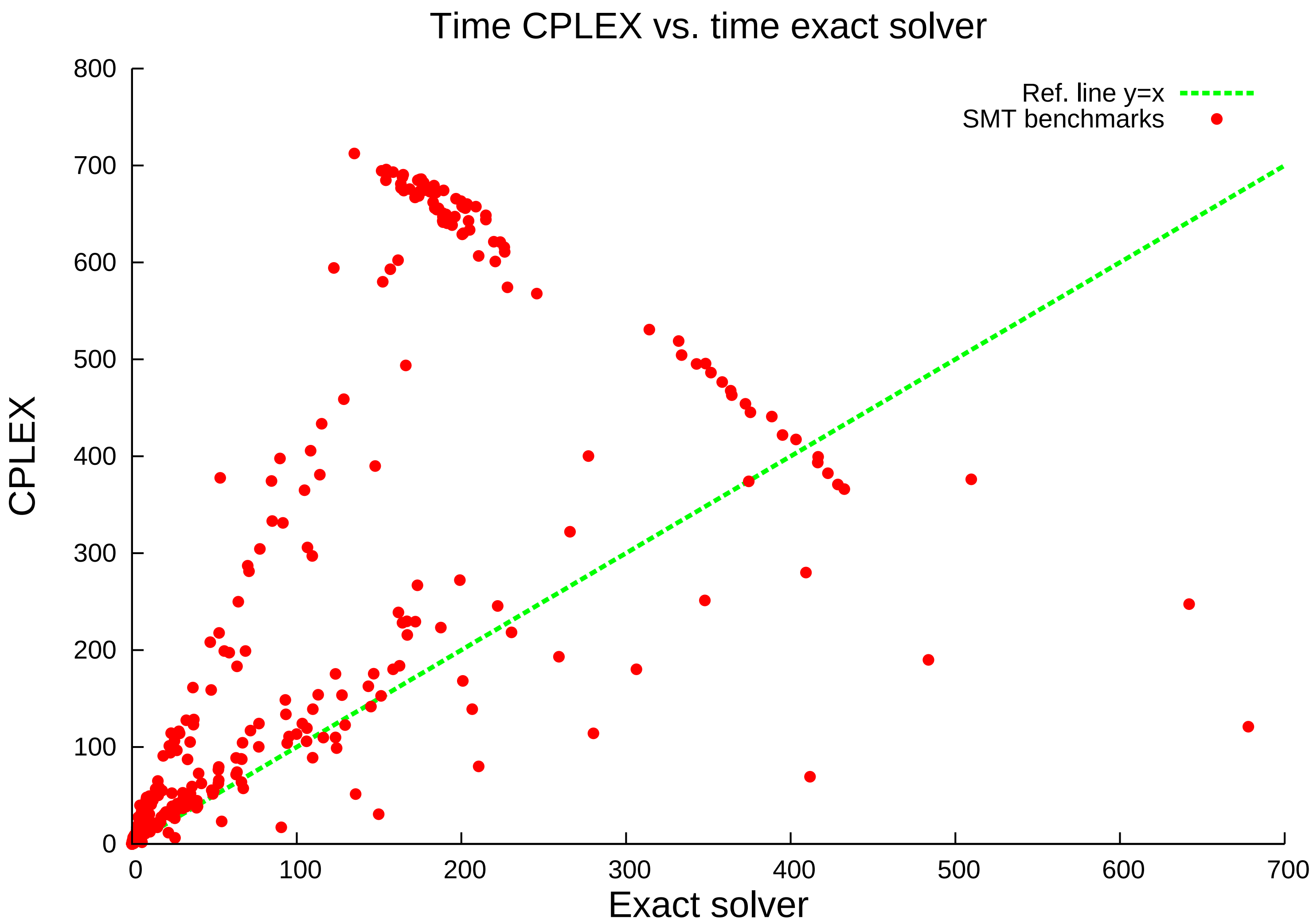
- So far, experiments reveal that:
  - CPLEX usually gives correct answers
  - Result checking is cheap comparing to CPLEX time
- Conclusion: replacing our exact T-Solver by CPLEX + result checking will improve performance
- Reality: no speed up is obtained. Sometimes even slower!!!!
- Analysis: could it be due to luck or parameter tuning?

  ANSWER: try fair comparison, i.e.

  1. Simultaneously call our exact solver and CPLEX at consistency checks
  2. Measure time taken by each solver
  3. Let exact solver guide the search

UPC
Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Experimental evaluation - Analysis (3)



Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Experimental evaluation - Analysis (4)

How to explain these results?

- We have tried various CPLEX settings, algorithms, parameters
- We have worked on many hypothesis (see paper)
- OUR CONCLUSION: CPLEX is not designed for being used as a solver in DPLL(T) nor the kind of problems arising in SMT
  - SMT problems are small (thousands constraints, hundreds variables)
  - Adding/removing constraints is not determinant in OR
  - Inconsistency explanations are not crucial in OR
  - Unlike with OR problems, SMT problems are easy and can be solved with few iterations of the simplex

UPC
Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITECNICA DE CATALUNYA

Overview of the talk

- SAT Modulo Theories (SMT)
  - DPLL(T) = Boolean engine + T-Solver
  - What is needed from T-Solver?
- Use of OR solvers for DPLL(LA)
  - Existing and non-existing functionalities
  - Adapting OR solvers
- Experimental evaluation
- New prospects and conclusions

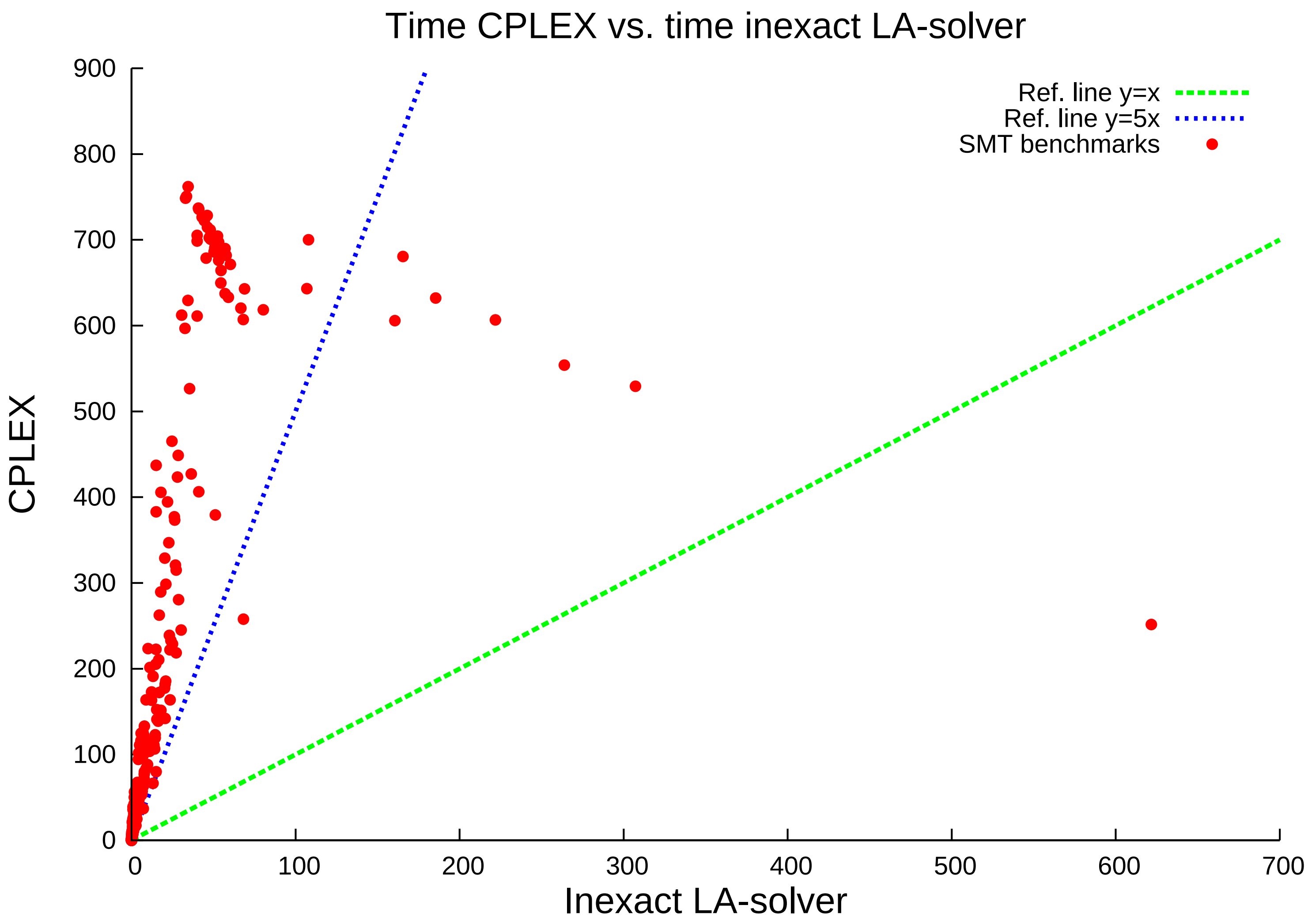
New prospects

- Experiments reveal that OR solvers are not competitive in SMT problems
- This negative result suggests new line of research:

> Combine result checking with implementations of our SMT T-Solver's using floating-point arithmetic

- Our initial experiments are promising, but result checking could be further improved
- The following figure shows a comparison of using CPLEX and a floating-point version of our T-Solver on the same sequence of calls.

New prospects (2)



UPC
Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITECNICA DE CATALUNYA

Conclusions

- Use of OR solvers inside an SMT system
- Result checking policies are cheap
- OR solvers are slower than SMT-dedicated solvers
- Possibility of improving performance by SMT-dedicated floating-point solvers

UPC
Departament de Llenguatges i Sistemes Informatics
UNIVERSITAT POLITECNICA DE CATALUNYA

Thank you!

Departament de Llenguatges i Sistemes Informatics̀
UNIVERSITAT POLITEC̀NICA DE CATALUNYA