TECHNISCHE UNIVERSITÄT DRESDEN

Artificial Intelligence

ASPARTIX-D READY FOR THE COMPETITION

Sarah Gaggl and Norbert Manthey

Dresden, 21st April 2015

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

ICCMA

International Competition on Computational Models of Argumentation

Home | Competition 2015 | Rules | Participation | Important Dates | Solvers | Organization | Contact

In cooperation with

The 2015 International Workshop on Theory and Applications of Formal Argument (TAFA'15)

Subscribe to the mailing list of ICCMA'15 argumentationcompetition@inria.fr by sending a mail with the subject

subscribe argumentationcompetition <your first name> <your surname>

to sympa_inria@inria.fr

This is the homepage of the International Competition on Computational Models of Argumentation (ICCMA). The competition aims at nurturing research and development of implementations for computational models of argumentation.

The first instance of the competition will be conducted in 2015 and is associated with the workshop "Theory and Applications of Formal Argument" (TAFA'15).

For more information on rules and news see Competition 2015.

Important dates

~~*Nov 21, 2014*~~:	~~Release of final solver interface~~
~~*Jan 30, 2015*~~:	~~Declaration of intent by participants~~
~~*Mar 30, 2015*~~:	~~Submission of solvers~~
Apr 30, 2015:	Submission of system descriptions
Jul 25-Aug 1, 2015:	Presentation of results at TAFA'15

ICCMA steering committee:

- Federico Cerutti, Department of Computing Science, University of Aberdeen, UK
- Nir Oren, Department of Computing Science, University of Aberdeen, UK
- Hannes Strass, Computer Science Institute, Leipzig University, Germany
- Matthias Thimm, Institute for Web Science and Technologies, University of Koblenz-Landau, Germany
- Mauro Vallati, School of Computing and Engineering, University of Huddersfield, UK
- Serena Villata, WIMMICS Research Team, INRIA Sophia Antipolis, France

All members of the steering committee can be reached through the common e-mail address sc@argumentationcompetition.org.

Last updated 14.04.2015, Matthias Thimm

Reasoning Tasks

Computational Tasks:

1. SE - given an AF F determine some extension
2. EE - given an AF F determine all extensions
3. DC - given an AF F and some argument a, decide whether a is credulously inferred
4. DS - given an AF F and some argument a, decide whether a is skeptically inferred

Semantics:

- CO - Complete Semantics
- PR - Preferred Semantics
- GR - Grounded Semantics
- ST - Stable Semantics

Our Goal

Make ASPARTIX ready for the 1st Argumentation Competition!

Our Goal

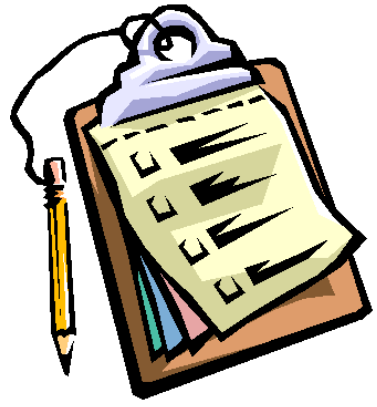
Make ASPARTIX ready for the 1st Argumentation Competition!

TODO

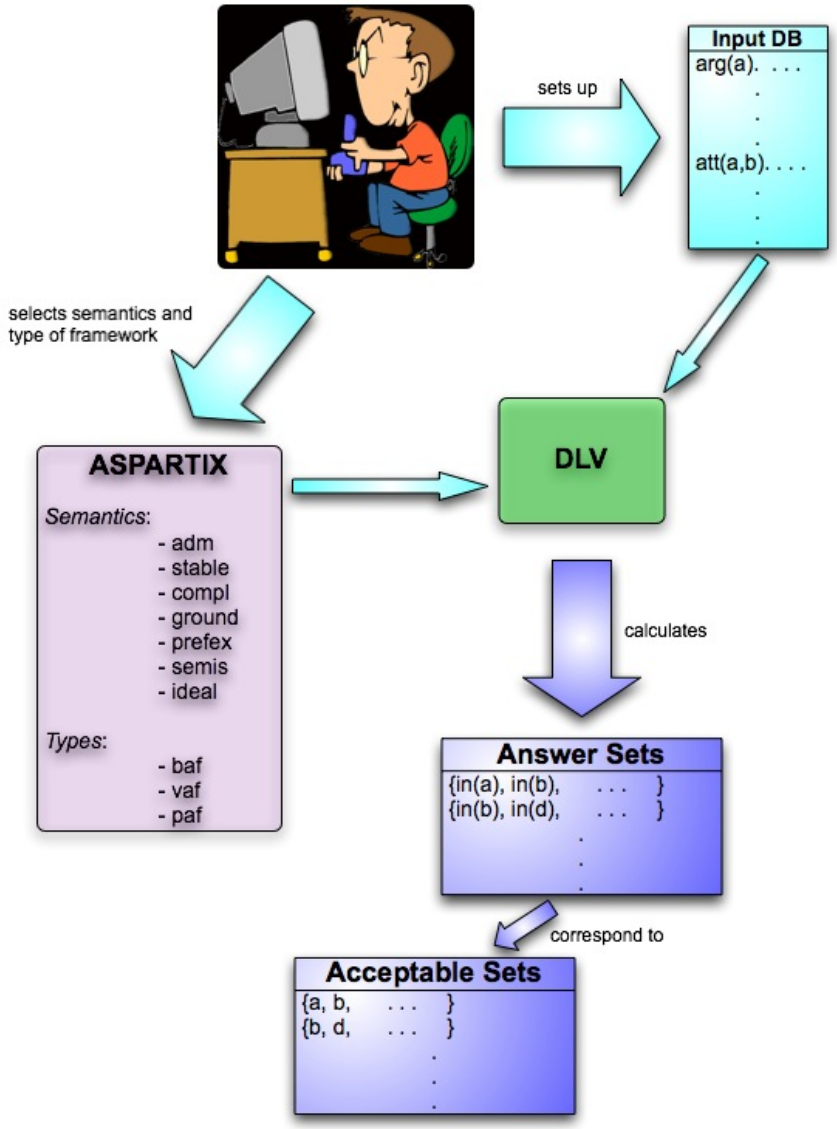
Find the "best" configuration for each reasoning task.

Agenda

1. What is ASPARTIX?
2. Modifications on Encodings
3. Which Solver and which Options?
4. Benchmarks
5. Tests
6. Results
7. Final Configuration of ASPARTIX-D
8. Future Work



ASPARTIX


sets up
Input DB
arg(a). . . .
att(a,b). . . .
selects semantics and
type of framework
ASPARTIX
Semantics:
- adm
- stable
- compl
- ground
- prefex
- semis
- ideal
Types:
- baf
- vaf
- paf
DLV
calculates
Answer Sets
{in(a), in(b), . . . }
{in(b), in(d), . . . }
correspond to
Acceptable Sets
{a, b, . . . }
{b, d, . . . }

ASP Encodings

Stable Extensions

Given an AF $F = (A, R)$. A set $S \subseteq A$ is a stable extension of F, if

- S is conflict-free in F
- for each $a \in A \setminus S$, there exists a $b \in S$, such that $(b, a) \in R$

Encoding

$$\widehat{F} = \{\text{arg}(a) \mid a \in A\} \cup \{\text{att}(a,b) \mid (a,b) \in R\}$$

$$\pi_{stable} = \left\{ \begin{array}{lcl} \text{in}(X) & \leftarrow & \textit{not } \text{out}(X), \text{arg}(X) \\ \text{out}(X) & \leftarrow & \textit{not } \text{in}(X), \text{arg}(X) \\ & \leftarrow & \text{in}(X), \text{in}(Y), \text{att}(X,Y) \\ \text{defeated}(X) & \leftarrow & \text{in}(Y), \text{att}(Y,X) \\ & \leftarrow & \text{out}(X), \textit{not } \text{defeated}(X) \end{array} \right\}$$

Modifications for EE

Clingo Syntax

$$\pi_{stable} = \left\{ \begin{array}{lll} \{\mathrm{in}(X) : \mathrm{arg}(X)\} & & \\ & \leftarrow & \mathrm{in}(X), \mathrm{in}(Y), \mathrm{att}(X,Y) \\ \mathrm{defeated}(X) & \leftarrow & \mathrm{in}(Y), \mathrm{att}(Y,X) \\ & \leftarrow & \mathit{not}\ \mathrm{in}(X), \mathit{not}\ \mathrm{defeated}(X), \mathrm{arg}(X) \\ \#\mathrm{show}\ \mathrm{in}/1 & & \end{array} \right\}$$

Run with clingo option `--project`

Modifications for DC

DC-ST

For an AF $F = (A, R)$, is $a \in A$ contained in some extension $E \in stable(F)$

Modifications for DC

DC-ST

For an AF $F = (A, R)$, is $a \in A$ contained in some extension $E \in \mathit{stable}(F)$

$$\pi_{\mathit{stable}} = \left\{ \begin{array}{lcl} \mathrm{in}(X) & \leftarrow & \mathit{not}\ \mathrm{out}(X), \mathrm{arg}(X) \\ \mathrm{out}(X) & \leftarrow & \mathit{not}\ \mathrm{in}(X), \mathrm{arg}(X) \\ & \leftarrow & \mathrm{in}(X), \mathrm{in}(Y), \mathrm{att}(X, Y) \\ \mathrm{defeated}(X) & \leftarrow & \mathrm{in}(Y), \mathrm{att}(Y, X) \\ & \leftarrow & \mathrm{out}(X), \mathit{not}\ \mathrm{defeated}(X) \\ \mathrm{in}(a) & & \end{array} \right\}$$

Let Clingo compute ONE Answer-Set

If $\pi_{\mathit{stable}}(\widehat{F})$ is satisfiable, then return YES;
if $\pi_{\mathit{stable}}(\widehat{F})$ is unsatisfiable, then return NO.

Modifications for DS

DS-ST

For an AF $F = (A, R)$, is $a \in A$ contained in each extension $E \in stable(F)$

$$\pi_{stable} = \left\{ \begin{array}{lll} \text{in}(X) & \leftarrow & \textit{not } \text{out}(X), \text{arg}(X) \\ \text{out}(X) & \leftarrow & \textit{not } \text{in}(X), \text{arg}(X) \\ & \leftarrow & \text{in}(X), \text{in}(Y), \text{att}(X, Y) \\ \text{defeated}(X) & \leftarrow & \text{in}(Y), \text{att}(Y, X) \\ & \leftarrow & \text{out}(X), \textit{not } \text{defeated}(X) \\ \text{out}(a) & & \end{array} \right\}$$

Let Clingo compute ONE Answer-Set

If $\pi_{stable}(\widehat{F})$ is satisfiable, then return NO;
if $\pi_{stable}(\widehat{F})$ is unsatisfiable, then return YES.

Solver Options

clingo 4.3

- NOPARAM
- –project
- –rand-freq=0.05
- –configuration=auto
- –configuration=frumpy
- –configuration=tweety
- –configuration=handy
- –configuration=crafty
- –configuration=trendy

gringo305/clasp

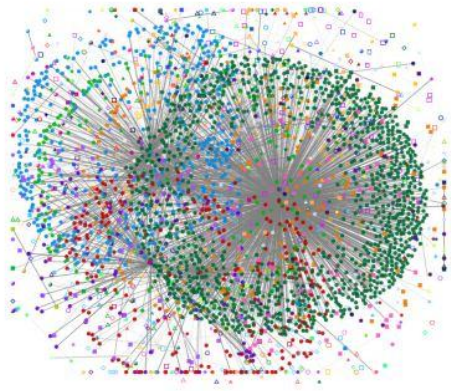
metasp encodings for preferred and grounded

lp2sat[Gebser et al.(2014)], riss[Manthey(2014), Manthey(2012)]

SAT Solver for DC-{ST,CO,GR} and DS-{ST,CO,GR}

Benchmarks

- Benchmarks_SCCs/ECAI* (≈8900 instances)
- Benchmarks_SCCs/KR/* (≈1500 instances)
- dynpartix/examples/benchmarks_1-2011/* (≈4800 instances)
- iccma15_testcases/apx/* (≈100 instances)
- ICCMAtest_cases2/apx/* (≈90 instances)



Total 15490 instances.
Second benchmark set 5830 instances.

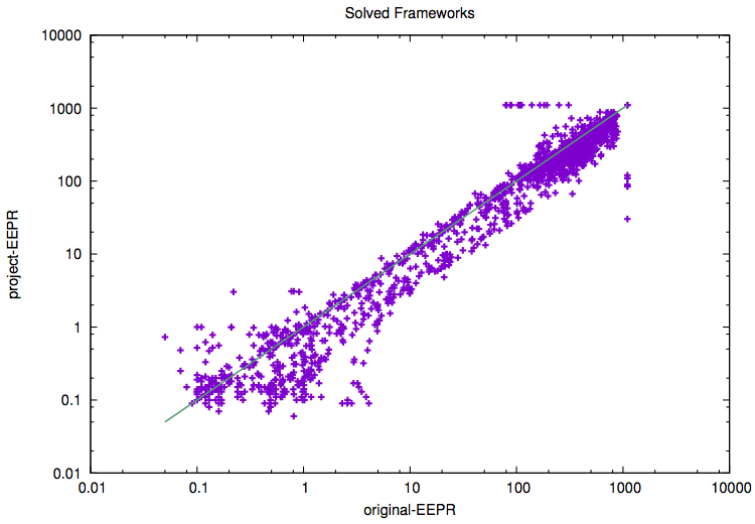
Tests

Bull HPC-Cluster (Taurus)

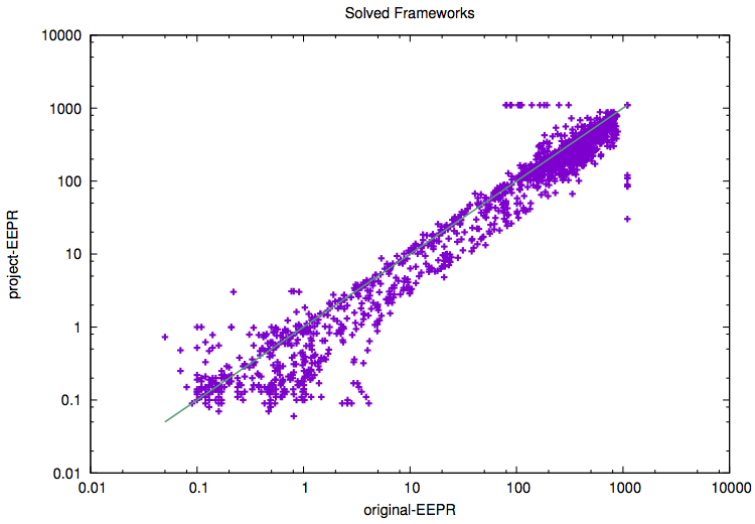
- Intel Xeon CPU (E5-2670) with 2.60GHz
- 6.5 GB Ram, 900 seconds
- from 16 cores we used every 4th

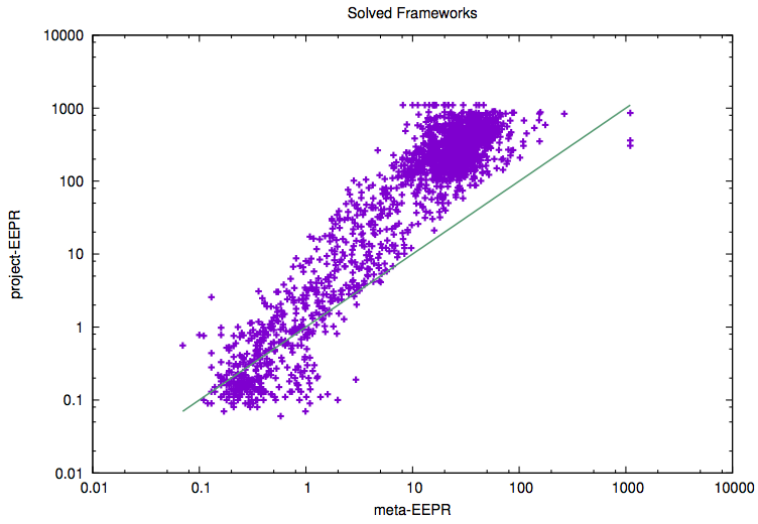
We thank the Center for Information Services and High Performance Computing (ZIH) at TU Dresden for generous allocations of computer time.

Results EE-PR

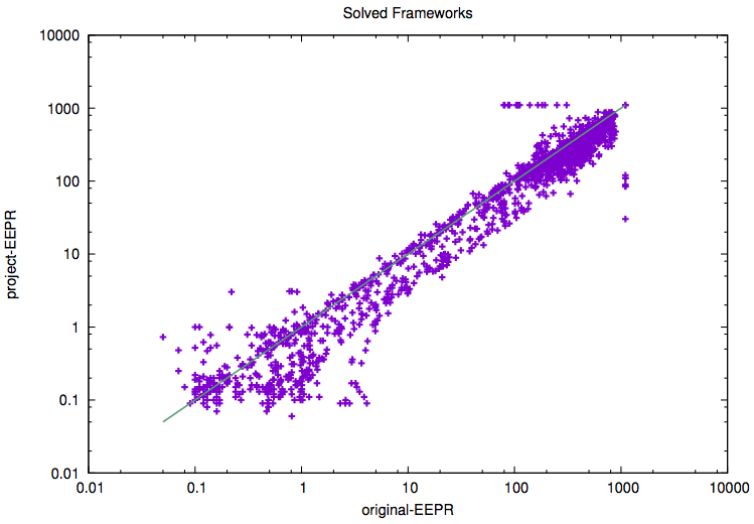


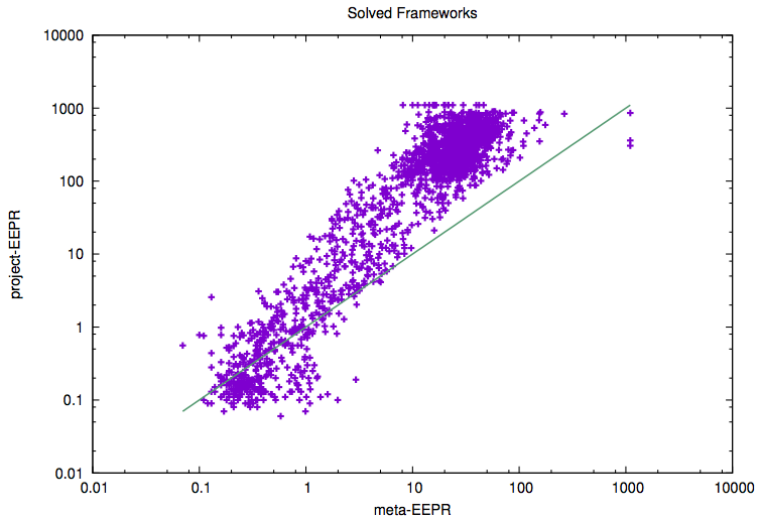
Results EE-PR

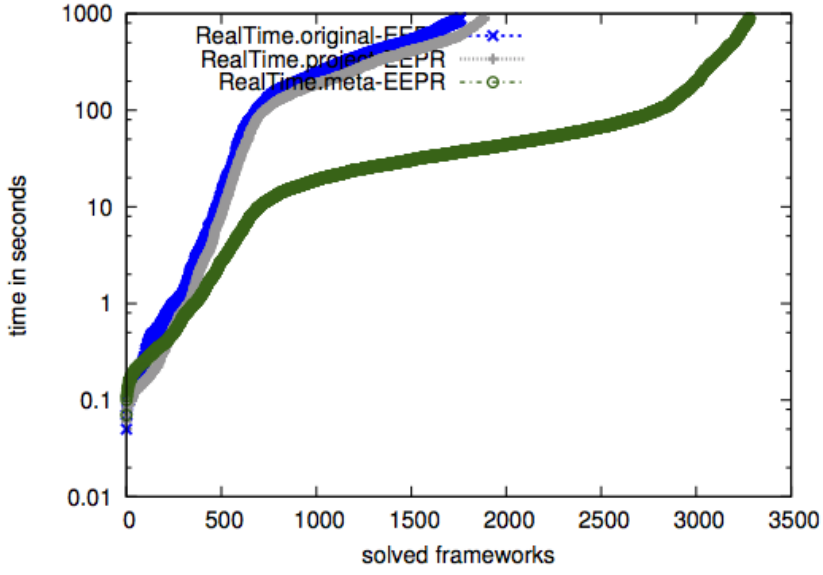




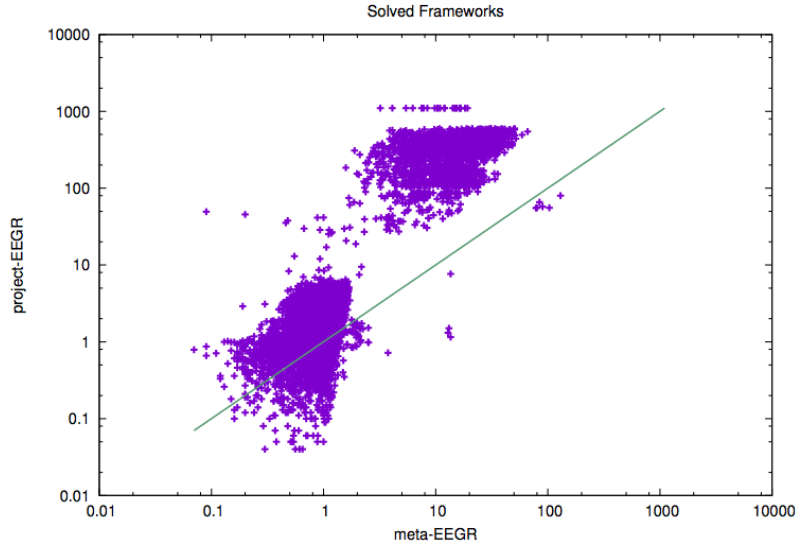
Results EE-PR



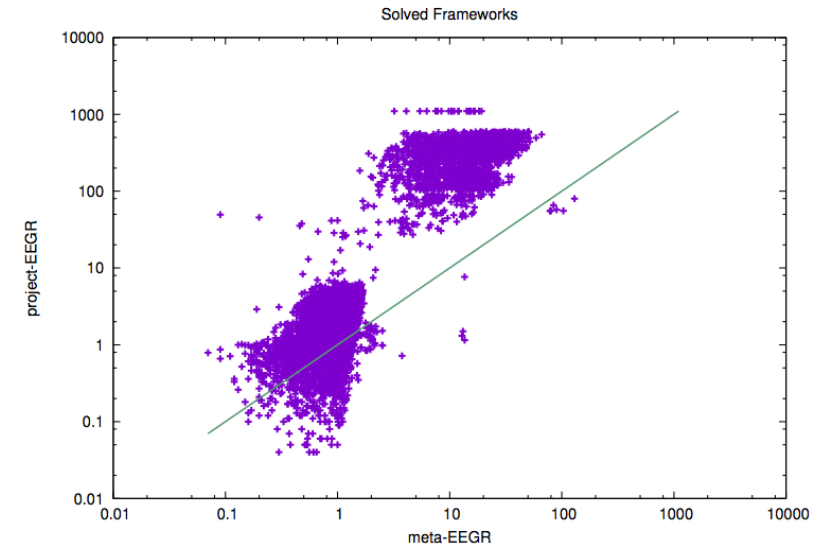


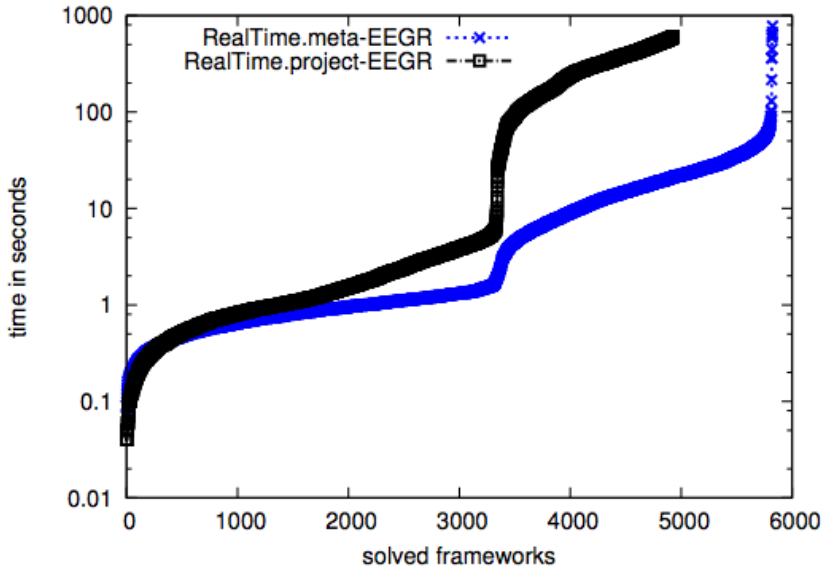


Results EE-GR

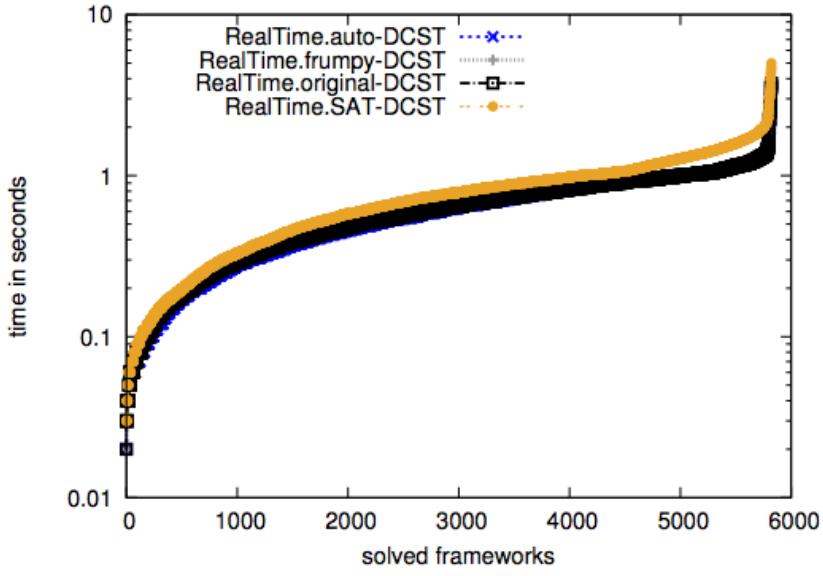


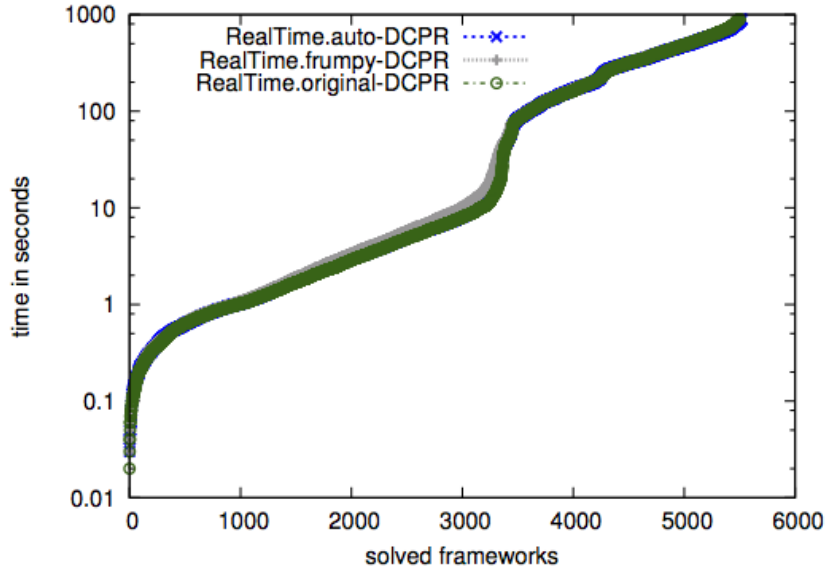
Results EE-GR

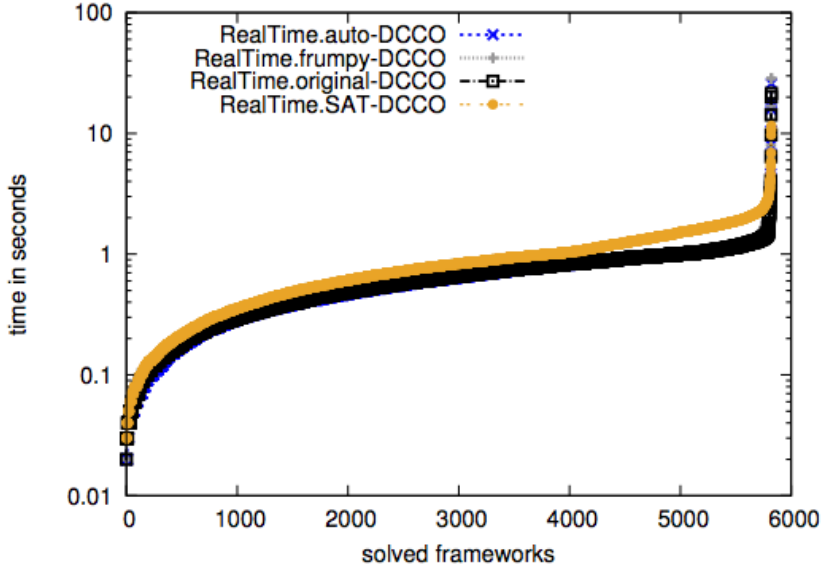


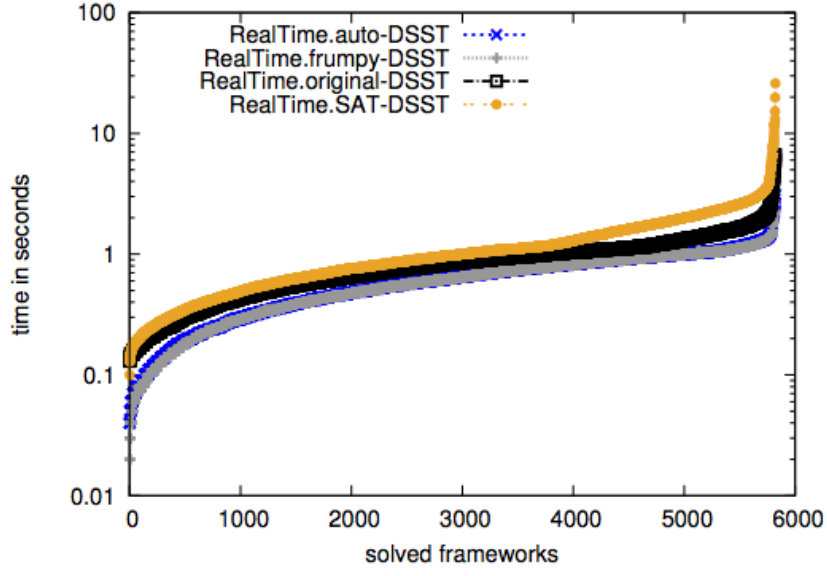


Results with no Significant Difference









Final Configuration

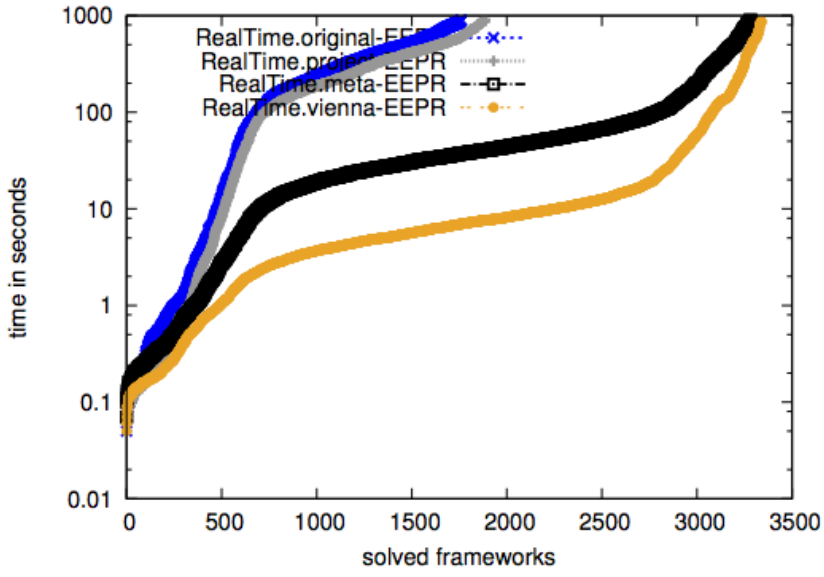
ASPARTIX-D

- GR: metasp encodings for all reasoning tasks
- DC-ST original
- DC-CO SAT
- DC-PR -configuration=auto
- DS-ST original
- DS-CO SAT
- DS-PR original
- EE-ST –project
- EE-CO –project
- EE-PR metasp
- SE-ST –project
- SE-CO –project
- SE-PR metasp

Future Work

- Fix configurations for other semantics
- GUI for ASPARTIX-D
- Study visualization methods for big frameworks
- Representation of many solutions
- Navigation in solution space

Comparison with Vienna Encodings



- Solved by metasp: 3244 (out of 5830)
- Solved by Vienna: 3325 (out of 5830)

References

Martin Gebser, Tomi Janhunen, and Jussi Rintanen.

Answer Set Programming as SAT modulo Acyclicity.
ECAI 2014. IOS Press. 2014.

Norbert Manthey.

Riss 4.27.
Proceedings of SAT Competition 2014. 2014.

Norbert Manthey.

Coprocessor 2.0 – A Flexible CNF Simplifier.
Theory and Applications of Satisfiability Testing – SAT 2012. Springer Berlin Heidelberg, 2012.

Martin Gebser, Benjamin Kaufmann Roland Kaminski, and Torsten Schaub.

Answer Set Solving in Practice.
Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2012.
doi=10.2200/S00457ED1V01Y201211AIM019.