Benchmarks and Quality Evaluation of CAS
ACA 2016 – Kassel – Germany

Albert Heinle

Symbolic Computation Group
David R. Cheriton School of Computer Science
University of Waterloo
Canada

2016–08–01
Correct Benchmarking of CAS – Case Studies and Dangers

Challenges and Vision for Benchmarking in Computer Algebra

Conclusion
Correct Benchmarking of CAS – Case Studies and Dangers
You read in a paper a sentence like the following:

*We presented a new implementation of algorithm X. Our timings show that we outperform the alternative programs when using examples we found in literature as input, and we observe that our program scales well by using randomly generated objects.*

What is/are potential problem/s?
You read in a paper a sentence like the following:

*We presented a new implementation of algorithm X. Our timings show that we outperform the alternative programs when using examples we found in literature as input, and we observe that our program scales well by using randomly generated objects.*

What is/are potential problem/s?

- Are the scripts and outputs made available? Did the authors check if the outputs were correct for the random inputs?
- Did the authors run the other programs on their machine, or did they just take the timings from the other paper?
- Did the authors also check the scalability for the other programs?
Consider the following **SINGULAR** code:

```plaintext
evaluate(read("singular_poly.txt"));
// File Content:
// ring R = 0,(x,y),dp;
// ideal I = *large polynomial system*;
timer = 1; int t = timer;
ideal g = yourCommand(I);
t = timer - t; print(g); print(t);
```

What is/are potential problem/s?
Consider the following \texttt{SINGULAR} code:

```plaintext
execute(read("singular_poly.txt"));
// File Content:
// ring R = 0,(x,y),dp;
// ideal I = *large polynomial system*;
timer = 1; int t = timer;
ideal g = yourCommand(I);
t = timer - t; print(g); print(t);
```

What is/are potential problem/s?

- \texttt{SINGULAR} sorts all input polynomials with respect to given monomial ordering. This may assist computations, but the sorting time is not taken into account.

- \texttt{SINGULAR} is open source, hence we know how the timer works. What happens if we would use \texttt{MAPLE} in a similar way?
Find the Problem – A Case Study III

Singular: | Maple:
======================================================================
ring R = 0,(x,y),lp; | with(Groebner):
ideal I = x^2 + y^2, x + y; | F:=[x^2 + y^2, x + y];
print(groebner(I)); | print(Basis(F,plex(x,y)))

What is/are potential problem/s?
Find the Problem – A Case Study III

Singular:  | Maple:
-----------------|-----------------------
ring R = 0,(x,y),lp; | with(Groebner):
ideal I = x^2 + y^2, x + y; | F:=\{x^2 + y^2, x + y\};
print(groebner(I)); | print(Basis(F,plex(x,y)))

What is/are potential problem/s?

- **SINGULAR** computes by default not a **reduced** Gröbner basis, while **MAPLE** in its current version always does.
Summarizing the Dangers of the Case Studies

- Ad Case Study I: Loosing Transparency.
- Ad Case Study II: Overlooking crucial implementation details.
- Ad Case Study III: Different facets of certain computations are overlooked.

The threat of all the above points becomes larger with the number of different implementations available.
Challenges and Vision for Benchmarking in Computer Algebra
What Makes Benchmarking for the Computer Algebra Community Difficult?

- Non-uniqueness of computation results. Sometimes checking results for “equality” is a difficult problem itself. This difficulty also transfers to checking the correctness of an output.
- Many sub-communities with their own sets of problems.
- Input formats for different computer algebra systems are differing a lot.
What We Should Not Do...

Figure: Picture Taken from http://xkcd.com/927/
SDEval for Benchmarking in Computer Algebra

- **SDEval**\(^1\) is a benchmarking framework tailored for the computer algebra community.
- **Create Benchmarks:** Using entries from the *Symbolic Data* database, one can create executable code for several different computer algebra systems.
- **Run Benchmarks:** Independent from the creation part, it provides a feasible infrastructure to run, monitor and time computations, and there are interfaces for scripts to interpret the output.

---

\(^1\)http://wiki.symbolicdata.org/SDEval
\(^2\)https://www.youtube.com/watch?v=CctmrfisZso
SDEval for Benchmarking in Computer Algebra

- **SDEval**\(^1\) is a benchmarking framework tailored for the computer algebra community.
- Create Benchmarks: Using entries from the Symbolic Data database, one can create executable code for several different computer algebra systems.
- Run Benchmarks: Independent from the creation part, it provides a feasible infrastructure to run, monitor and time computations, and there are interfaces for scripts to interpret the output.

---

\(^1\)http://wiki.symbolicdata.org/SDEval
\(^2\)https://www.youtube.com/watch?v=CctmrfsZso
A Call For Transparency: The SDEval Solution

Together with papers, authors should make so-called taskfolders available. These look like the following.

```
+ TaskFolder
  |- runTasks.py //For Running the task
  |- taskInfo.xml //Saving the Task in XML Structure
  |- machinesettings.xml//The Machine Settings in XML form
  | + classes //All classes of the SDEval project
  | + casSources //Folder containing all executable files
  |    | + SomeProblemInstance1
  |    |    | + Computer Algebra System1
  |    |    |- executablefile.sdc //Executable code for CAS
  |    |    |- template_sol.py //Script to analyze the output of the CAS
  |    |    | + Computer Algebra System2
  |    |    |- executablefile.sdc
  |    |    | + ...
  |    | + SomeProblemInstance2
  |    |    + ...
  |    + ...
```

**Figure**: Folder structure of a taskfolder
What We Could be Working Towards: StarExec

- StarExec\(^3\) is a complete benchmarking infrastructure for the satisfiability community (SAT/SMT solvers). Funded with 1.85 million USD by the NSF.
- Different kinds of computations clearly structured and standardized by SMT-LIB.

![Diagram of logical relationships](http://smtlib.cs.uiowa.edu/logics.shtml)

Figure: Image taken from http://smtlib.cs.uiowa.edu/logics.shtml

\(^3\)https://www.starexec.org/
Different from SDEval, StarExec also provides physical computation infrastructure to perform calculations and to run benchmarks (Used during conferences).

StarExec does not provide the flexibility that we would need for computer algebra computations. However, we can learn a lot from their experience and maybe one day create a similar infrastructure for computer algebra.
Conclusion
What Do We Need, What Do We Have

- The computer algebra community needs to realize the need we have for correct, reproducible, and transparent benchmarking.
- Several databases, like Symbolic Data, are available from different communities. We need a way to have a central overview of all of them.
- With SDEval, we have a starting point for creating and running benchmarks, which can be refined in the future.
- At some point, we should also introduce a computational infrastructure à la STARExec.