DEFINITIONS OF TERMS USED IN THE SDEVAL PROJECT

ALBERT HEINLE

1. Definitions

Definition 1.1 (Task)**.** A task is always associated to a certain computation problem (see Definition 1.3). A task contains a list of SD-Tables (see Definition 1.7), that contain entries that are suitable as input for the associated computation problem.

From those SD-Tables, a task contains a set of problem instances (see Definition 1.5). Additionally, a task contains a set of computer algebra systems (see Definition 1.9), that provide algorithms for solving the associated computation problem. Every set in a task is not empty. Furthermore, a task has a name.

Example 1.2 (Task)**.** Let us call a task `xyz`, and let its associated computation problem be `GB_Z_lp`.

This is an entry in the Symbolic-Data table `COMP` and represents the commutative Gröbner basis computation in a commutative ring using the lexicographical ordering and given generators that have coefficients in $\mathbb{Z}$.

The only existing SD-Table, that provides an entries that can be used as inputs for the Gröbner basis algorithms, is meanwhile `INTPS` (see Symbolic-Data). From this table, we choose as one of our problem instances the instance `Amrhein`. A computer algebra system, that provides an algorithm for `GB_Z_lp` is for example Singular.

Definition 1.3 (Computation Problem)**.** A computation problem is a problem, which is specified in the SD-Table (see Definition 1.7) `COMP` of the Symbolic Data project. In the context of SDEval we are using it to specify which computations we want to perform on certain problem instances (see Definition 1.5), resp. which algorithm shall be used.

Example 1.4 (Computation Problem)**.** A computation problem is for example `GB_Z_lp`.

This is an entry in the SD-Table table `COMP` and represents the commutative Gröbner basis computation in a commutative ring using the lexicographical ordering and given generators that have coefficients in $\mathbb{Z}$.

Definition 1.5. A problem instance is in our context a representation – specified by Symbolic-Data – of a concrete input for suitable algorithms. Suitable means that the entries for the chosen algorithms can be read from this problem instance. A problem instance is always contained in a SD-Table (see Definition 1.7)

Example 1.6 (Problem Instance)**.** A problem instance is for example the entry `Amrhein` in the SD-Table `INTPS`. It contains variables and a basis of

polynomials, and those can be used for Gröbner basis computations, for example.

Definition 1.7 (SD-Table)**.** A SD-Table denotes the folder structure of a chosen subfolder in the `XMLRessources` folder in the Symbolic-Data project.

Example 1.8 (SD-Table)**.** There are several folders in `XMLRessources`. Some examples:

- `INTPS`
- `COMP`
- `ModPS`
- `CAS`

Definition 1.9 (Computer Algebra System)**.** A computer algebra system is a program, described by an entry in the SD-Table `CAS` (see Definition 1.7). It provides algorithms to solve different computation problems (see Definition 1.3).

Example 1.10 (Computer Algebra System)**.** SINGULAR (`http://www.singular.uni-kl.de/`) is for example a computer algebra system.

Definition 1.11 (Taskfolder)**.** A taskfolder is always associated to exactly one task (see Definition 1.1). It contains

- The task itself
- For every problem instance (see Definition 1.5) and for every computer algebra system (see Definition 1.9) the taskfolder contains one executable file, that contains the calculation steps for the calculation of the computation problem (see Definition 1.3).
- An executable file `runTasks.py` and all needed modules (see Definition 1.13) of this file.
- The machine settings (see Definition 1.15) for the target machine (see Definition 1.17) of the task

It serves the purpose to be exported as a single folder to a target machine, where the calculations shall be run at.

Example 1.12 (Taskfolder)**.** In order to have an example of an taskfolder, just run `create_tasks[_gui].py`, and the folder that is created in the end, that is an taskfolder.

Definition 1.13 (Module)**.** A module is a collection of executable code of a certain programming/script-language.

Example 1.14 (Module)**.** In the SDEval project, there is for example a module called

`classes.probleminstances`.

It contains representations as classes of different types of problem instances.

Definition 1.15 (Machine Settings)**.** Machine settings is a collection of machine-specific constants of the target machine (see Definition 1.17); Those are:

- For every computer algebra system the command to execute it on the target machine.

- The command for the time measurement (time command, see Definition 1.18) on the target machine with the desired options of the user.
- Optional, it can contain further entries.

Example 1.16 (Machine Settings)**.** The command for time measurement is on every `UNIX`-like machine simply `time`. The command to run e.g. version 15 of MAPLE on a machine known to the author is `maple15`.

Definition 1.17 (Target Machine)**.** A target machine is the computer on which in the end the computations for a specific task (see Definition 1.1) will take place. In our choices of words we assume in general, that it has a `UNIX`-like operating system installed. There must be a time command available on that machine (see Definition 1.18).

Definition 1.18 (Time Command)**.** A time command is a tool for time measurement. On `UNIX`-like operating systems it is e.g. the command `time`.

In the context of SDEval we need that this time command is able to produce its output in the IEEE Std 1003.2-1992 ("POSIX.2") described way. On `UNIX`-machines, it is achieved by adding the option `-p` to the command.

Example 1.19 (Time Command)**.** For example, on a mac, this is how the time command works:

```
$ time -p echo "Hello"
Hello
real 0.00
user 0.00
sys 0.00
```

Definition 1.20 (Proceedings)**.** Proceedings contain information about the status of the execution of the files in the task folder (see Definition 1.11). For the status, there are the following options:

- RUNNING – The file is running
- WAITING – The file is waiting for its execution
- COMPLETED – The execution of the file is finished
- ERROR – An Error occured

During the execution, there will always be an XML and an HTML file written visualizing the current proceedings.

Example 1.21 (Proceedings)**.** Let us take the entry `Amrhein` of the `INTPS` table and let the computer algebra systems SINGULAR and MAPLE be chosen for the computation of a Gröbner basis of those systems.

In the beginning, both execution files (i.e. for SINGULAR and MAPLE) are WAITING. If one of the files is executed on its computer algebra system, it has the status RUNNING, and when the computation is completed, the status will change to COMPLETED.

Definition 1.22 (Resulting File)**.** During the execution of the files in an task folder (see Definition 1.11), there are always resulting files containing the output of the computer algebra system (see Definition 1.9) and the time command (see Definition 1.18).

Example 1.23 (Resulting File)**.** A resulting file after executing some singular commands has for example the following form:

```
                     SINGULAR                                 /
 A Computer Algebra System for Polynomial Computations       /   version 3-1-5
                                                            0<
 by: W. Decker, G.-M. Greuel, G. Pfister, H. Schoenemann     \   Jul 2012
FB Mathematik der Universitaet, D-67653 Kaiserslautern        \
a2+a+2bf+2ce+d2,
2ab+b+2cf+2de,
2ac+b2+c+2df+e2,
2ad+2bc+d+2ef,
2ae+2bd+c2+e+f2,
2af+2be+2cd+f

$Bye.
real 0.53
user 0.01
sys 0.01
```

Index