Combinatorial Optimisation and Constraint Programming (1DL442) Uppsala University – Autumn 2023 Assignment 4: Filtering, Consistency, Fixpoint, Domains, Variables, Constraints, Memory Management, and Search

Prepared by Pierre Flener and Frej Knutar Lewander

— Deadline: 13:00 on Friday 17 November 2023 —

It is strongly recommended to read the *Grading Rules* below and the *Submission Instructions* at the end of this document even ***before*** attempting to tackle its tasks. It is also strongly recommended to prepare and attend the help sessions, as huge time savings may ensue.

Questions and Grading Rules

Assignment 4 is pass/fail and covers **Modules 1 to 2** of the MiniCP teaching materials [1]. The tasks are as follows, with ***no*** report to be written:

A. Individually: Pass *all* the **Theoretical Questions** at INGInious (find the UU-specific link in an announcement at Studium) of *all* those modules.

B. As a team:

- Pass *all* the unit tests at INGInious (ditto) for `GraphColoringTinyCSP` (Module 1), `IntVarImpl` (Module 2), `StateSparseSet` (Module 2), and `Maximum` (Module 2).
- Upload *also* at Studium *all* `*.java` (except the `*Test.java`) mentioned in the questions, for a local archive at UU.

The solution session will be questions & answers on the **Theoretical Questions** of Modules 1 to 2.

References

[1] Laurent Michel, Pierre Schaus, and Pascal Van Hentenryck. MiniCP: A lightweight solver for constraint programming. *Mathematical Programming Computation*, 13(1):133–184, 2021. The source code is available at `http://minicp.org` and the teaching materials are available at `https://www.edx.org/course/constraint-programming`.

Submission Instructions

In order to protect yourself against an unnecessary loss of points, use the following to-do list before submitting:

- There is no demo report, but remember best practice on comments for code and on experimental evaluation from Sections C and D of the MiniZinc demo report (`https://user.it.uu.se/~pierref/courses/COCP/demoReport`) and use your best judgement.
- Write in the report (if you write one) a paragraph, which will not be graded, describing your experience with this assignment: Which aspects were too difficult or too easy? Which aspects were interesting or boring? This will help us improve the course in the coming years.
- ***Thoroughly*** proofread, spellcheck, and grammar-check the report, at least once per teammate, including the comments in ***all*** source code. In case you are curious about technical writing: the *English Style Guide* of UU at `https://mp.uu.se/en/web/info/stod/kommunikation-riktlinjer/sprak/eng-skrivregler` and the technical-writing *Checklist & Style Manual* of the Optimisation group at `https://optimisation.research.it.uu.se/checkList.pdf` offer many pieces of advice; common errors in English usage are discussed at `https://brians.wsu.edu/common-errors`; in particular, common errors in English usage by native Swedish speakers are listed at `https://www.crisluengo.net/english-language`.
- Produce the report as a ***single*** file in ***PDF*** format; all other formats will be rejected.
- Remember that when submitting you implicitly certify (a) that your files were produced solely by your team, except where explicitly stated otherwise and clearly referenced, (b) that each teammate can individually explain any part starting from the moment of submitting your files, and (c) that your files are not freely accessible on a public repository.
- Submit (by only ***one*** of the teammates) the files (all `*.java` mentioned in the questions, except the `*Test.java`, and possibly a report) ***without*** folder structure and ***without*** compression via *Studium*, whose clock may differ from yours, by the given ***hard*** deadline.