COCP Part 2: Warm up

Based on Felix Morsbach, modified by Petr Binko

Uppsala University

Sweden

presentation of 27th October 2020

UPPSALA UNIVERSITET

Outline

1. General Information

2. Installation / Running

3. General Advice

4. Gecode Hands-on

General Information

- You don't have to be a C++ wizard (but it will help)
- Help?
  - **Gecode Manual** (MPG) [1]
  - Course notes from KTH Gecode course [2]
  - E-Mail us: it-COCP@lists.uu.se, not individually
    - However, **come to the help sessions**, as they are heavily prioritized !
- Follow **Submission Instructions** !
- **Use the demo report** template for this part of the course as you did for minizinc
- Check progress/feedback area, ask for more extensive feedback

[1] https://www.gecode.org/doc-latest/MPG.pdf

[2] http://user.it.uu.se/~pierref/courses/COCP/assignments/CourseNotes.pdf

Installing / Running Gecode

- Already installed on Linux servers
  - We will not be helping you for personal installations
- Installation Instructions: **MPG**
  - Precompiled binaries available for Windows, Mac, Debian and Ubuntu
    - Linux packages most likely will be outdated
    - Will need to compile from source
- Gecode programs are written in C++
  - Compiles to a binary, runs as a binary
  - Use provided Makefiles
- For running experiments:
  - Use the provided Python script[3]

[3] http://user.it.uu.se/~pierref/courses/COCP/assignments/gather_stats.py

Some advice from a former student

- **RTFM**, now !
  - At least chapters 1, 2, 3, 4, 7, 8, 11
  - Always refer back to it !
- For assignment 4, exercise 3 there is a table in the demo report, use it (both for you and for us) !
- Don’t be overwhelmed, use code-skeletons and take the tasks one at a time
- Model iteratively
  - Gecode error messages can be cryptic compared to MiniZinc, especially when mixed with C++ errors.

UPPSALA UNIVERSITET

Gecode Hands-on