

Introduction to MiniCP

Frej Knutar Lewander

Combinatorial Optimisation and Constraint Programming (course 1DL441)
at
Uppsala University, Sweden

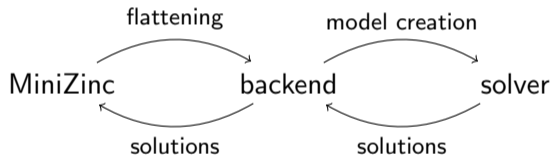
1st November 2023

Help Sessions and Assignments

- Scheduled time for working on an assignment and asking questions.
- Help sessions are *not* labs: *you have to work on the current assignment before and after each help session!*
- The assignments are challenging (no exam!) and will require much time.
- There are three assignments.
- You upload your PDF report (singular) and Java code

Introduction

In the modelling part of this course (the MiniZinc part), we had a "helicopter view": we agnostically modelled our problems and MiniZinc and the backends "translated" and ran the model for us.



In this part of the course, we will "zoom in" on the solver part by creating our own solver.

MiniCP

- It is a minimal Constraint Programming solver written in Java.
- You will extend the solver, adding missing functionality.
- The missing functionality has unit tests: if you pass the unit tests, then there is a high chance that your implementation is correct.

MiniCP

Navigating the code base and understanding how the different parts of the solver are connected will take some time, but the assignments will help guide you through the code base and reveal the different parts of the solver when needed.

MiniCP

Navigating the code base and understanding how the different parts of the solver are connected will take some time, but the assignments will help guide you through the code base and reveal the different parts of the solver when needed.

MiniCP relies (somewhat) heavily on lambda expressions (called lambda functions in Python and anonymous functions in both Haskell and JavaScript).

MiniCP – Lambda Expressions

- A lambda expression is a block of code that takes zero or more parameters and that either returns no value, or returns a value: just like a function.
- Lambda expressions are objects that can be stored in data structures and be parameters of methods.
- A lambda expression does not have its own variable scope, instead it has the same variable scope as its enclosing scope.
- MiniCP implements the Callback pattern using lambda expressions. When certain events occur (such as the domain of a variable becoming fixed), the corresponding lambda expression(s) will be called, if any.

MiniCP – Lambda Expressions

Here is a lambda expression that takes an integer parameter and that returns an integer:

```
Function<Integer, Integer> square = n -> n*n;
```


MiniCP – Lambda Expressions

Here is a lambda expression that takes an integer parameter and that returns an integer:

```
Function<Integer, Integer> square = n -> n*n;
```

and in Python:

```
square = lambda n : n*n
```

and in Haskell:

```
square = \n -> n*n
```

MiniCP – Lambda Expressions

To get you well started with lambda expressions, there are some warm-up exercises (no hand-in!) at <https://user.it.uu.se/~frekn832/courses/COCP/assignments/warmup/LambdaExpressionTest.html>.

MiniCP – TinyCSP

- TinyCSP is even more minimal than MiniCP and can only solve Constraint Satisfaction Problems (CSPs).
- TinyCSP is only used in the first assignment, and serves as an introduction to MiniCP.

Bugs

The official repo may turn out to contain bugs: please tell us if you suspect having found one!

INGInious

For this course, we are using the online teaching assistant INGInious, maintained and developed by UCLouvain.

You need to create an account on INGInious and enrol to the course. The password for the enrolling will be given by me and on Studium.

INGInious is similar to Studium, but allows for automatic testing of the Java code of your repository. This functionality is used for your assignment submissions.

For each assignment, besides verifying that your code is correct, you have to correctly *and individually* answer *all* multiple-choice questionnaires of that assignment.

INGInious (cont'd)

The first thing after successfully creating an account and enrolling in the course, you and your teammate need to place yourself in the same team and you *need to inform us via the help-desk that you have enrolled*, in order for us to facilitate creating a GitHub repository for you.

If you don't already have a GitHub account, then you need to create one.

You then need to use INGInious to fork the MiniCP repository for your team by performing the “Create GitHub repository” task in INGInious.

Locate Everything

Take a few minutes to locate everything:

- Creating a GitHub repository of MiniCP for your team

<https://INGInious.org/course/minicp-s/repo-create>

- Setting up the project

<https://github.com/minicp/minicp#readme>

- The assignments are at

<https://user.it.uu.se/~pierref/courses/COCP/assignments>

- The exercises on lambda expressions are at

[https://user.it.uu.se/~frekn832/courses/COCP/assignments/warmup/
LambdaExpressionTest.html](https://user.it.uu.se/~frekn832/courses/COCP/assignments/warmup/LambdaExpressionTest.html)

Next Steps

What to do now:

- Enrol at <https://www.inginius.org/>
- Notify me of your INGINIOUS user name and your team number, for creation of the team's repository.
- Create your team's repository using INGINIOUS.
- Look at the exercises on lambda expressions.
- Look at Assignment 4.

Additionally, you can sign up and enrol also at the LouvainX: Constraint Programming course, in order to have access to the MiniCP lecture videos at

<https://www.edx.org/learn/computer-programming/universite-catholique-de-louvain-constraint-programming>

Questions?

