What really bugs me about modelling is... CCPCPPAPI: there is no Common CP C++ API

Tias Guns, KU Leuven, Belgium

September 3, 2021

Modeling languages like MiniZinc and Essence' and CPMpy and PyCSP3 and Numberjack are great.

A key part of what they do, is take a high-level CP specification, and translate into a lower level (flat) constraint specification.

But how to get that specification to the solver?

MiniZinc invented a text-based 'flat' language. PyCSP3 invented a text-based slightly-less-flat language. Essence' and Numberjack carefully map its expressions to different solver APIs. MiniZinc also has that for just a few solvers.

This really bugs me because...

- Solver developers need to write file parsers... to map text to their API, painstakingly like in the 90s
- Writing files and reading files is slow, especially when done repeatedly
- It stifles innovation, you can't just add another API call to a text file... for example for incremental solving, or nogoods

Why don't we come up with a Common CP C++ API (or **CCPCPPAPI** in short). Then, we can all do API-to-API communication, and no more tedious and limiting parsing.

There could even be 3 different function groups in such an API:

- base: creating variables, constraints, an objective and solving
- incremental: like 'base' but all functions can also be called after a 'solve()' [and in the best case, reuses information from previous solve in new solve]
- nogood: returning an unsat core if the solver returned unsat

But what would the API calls look like? Good question.