

Monadic Constraint Programming with Gecode

Pieter Wuille Tom Schrijvers

ModRef'09

Lisbon, September 20, 2009



Modelling languages

Zinc approach

Separate language:

- Very declarative
- Transferring data/results
- Not all language features

Gecode approach

Library for existing language:

- Boilerplate/function calls
- Directly callable/usable
- All host-language features

Modelling languages

Zinc approach

Separate language:

- Very declarative
- Transferring data/results
- Not all language features

Gecode approach

Library for existing language:

- Boilerplate/function calls
- Directly callable/usable
- All host-language features



MCP approach

Best of both worlds:

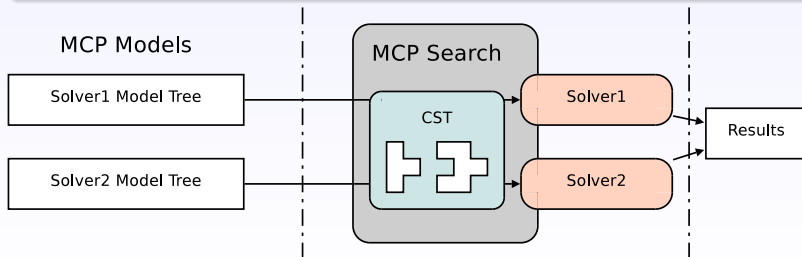
- High-level abstractions
- Very declarative
- Directly usable

+ Compositional search transformers

MCP Framework

MCP Framework

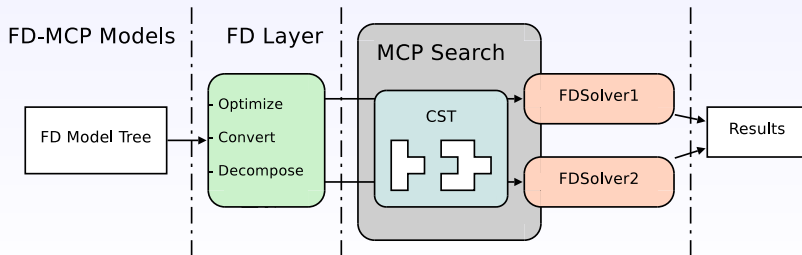
- Constraint Solving framework for Haskell
- Multiple solvers, variables, domains, search strategies, ...
- Fully functional, no side-effects
- Solver states and search trees are first-class data
- Defines common syntactic sugar



FD-MCP: FD solving for MCP

FD-MCP: Common FD Infrastructure

- FD syntax for term and constraint language
- Model optimizations and decompositions
- Multiple solver backends





FD-MCP: FD solving for MCP: example

```
model :: FModel
model =
  exist 4 $ \list@[a,b,c,d] ->
    list 'allin' (0,711)           ^
    sorted list                   ^
    a + b + c + d @= 711          ^
    a * b * c * d @= 711*100*100*100 ^
    return list

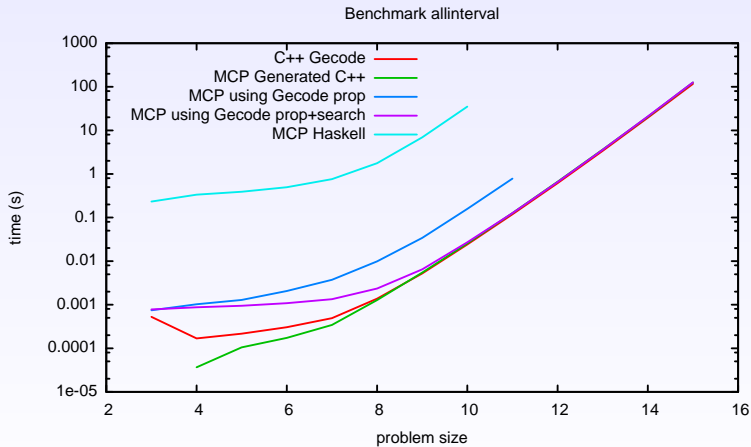
main = example_main_void model
```

FD-MCP: Solvers

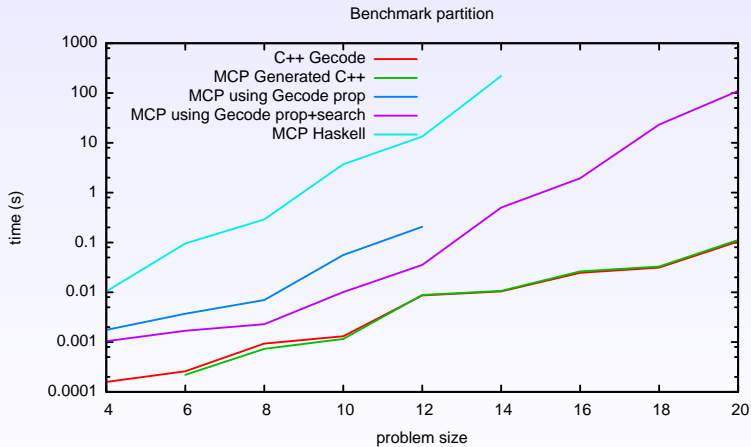
FD Solvers

- Haskell-only solver
- Pseudo-solver that generates C++ code for Gecode
- **new** Real solver using Gecode propagation
- **new** Real solver using Gecode search+propagation

Benchmark: Allinterval



Benchmark: Partition



Future Work

Future work

- More benchmarks
- Improve performance of MCP using Gecode propagation
- Parametrized models, loop constructs, reification, ...
- More backends, constraints, variable types, ...
- Optimizations, generalizations
- Concurrency

Thank you!

Questions?

Further reading:

Monadic Constraint Programming,
T. Schrijvers, P. Stuckey, P. Wadler.
Journal of Functional Programming – to appear.

More information:

Source code, papers, ...

<http://www.cs.kuleuven.be/~pieterw/site.php/Topics/FDMCP>