Introduction to constraint logic programming with ECLiPSe

Sergii Dymchenko

http://sdymchenko.com

May 22, 2015

CLP

Constraint Logic Programming (CLP) is an augmentation of logic programming paradigm where relations between variables are specified with constraints.

- $X \leq 10$
- $X > Y$
- $Y = 9$

CLP: Pros

- Declarative
- Compact
- Understandable
- Easy to modify
- Often fast enough

CLP: Cons

- Running time extremely depends on the instance
- Running time extremely depends on heuristics
- Optimization (vs any feasible solution) is slow
- Based on non-mainstream logic programming paradigm

ECLiPSe

ECLiPSe CLP (`http://eclipseclp.org/`) is an open-source Prolog-based system which aims to serve as a platform for integrating various logic programming extensions, in particular constraint logic programming.

ECLiPSe: libraries

- `ic`: interval arithmetic constraint library
- `gfd`: interface to Gecode (`http://www.gecode.org/`)
- Other CLP libraries (constraints on graphs, sets, ...)
- Interfaces to linear programming solvers
- A lot of them: `http://eclipseclp.org/doc/bips/index.html`

Demo some Prolog in ECLiPSe

Demo some Prolog in ECLiPSe (incl. `append`).

TPK

TPK is a simple algorithm proposed by D. E. Knuth and L. T. Pardo in "The Early Development of Programming Languages". It is used to demonstrate some basic syntax of a language beyond the "Hello, World!".

- Prompt for 11 real numbers $(a_0 \dots a_{10})$
- For each a_i compute $b_i = f(a_i)$, where $f(t) = \sqrt{|t|} + 5t^3$
- For $i = 10 \dots 0$ (in that order) output a pair (i, b_i) if $b_i \leq 400$, or $(i,$ TOO LARGE) otherwise

Demo `tpk.ecl`.

Arithmetic in Prolog

- `is`-based arithmetic in Prolog is not relational
- CLP can be seen as extension that brings relational arithmetic to Prolog
- Demo $c = a/b$

Exercise: relational factorial

- Complete code in `factorial.ecl`
- `factorial` should work "both ways": each argument can be input or output
- Demo `factorial.ecl`

SEND + MORE = MONEY

- Cryptarithmetic puzzle (`http://en.wikipedia.org/wiki/Verbal_arithmetic`)
- Demo `money.ecl`

Constraint propagation and search

- Just constraint propagation is not enough in general case
- Search is needed
- Parameters:
  `http://eclipseclp.org/doc/bips/lib/ic/search-6.html`
- Demo `alldifferent.ecl`

N-queens puzzle

- Place N non-attacking chess queens on an $N \times N$ board
- `http://en.wikipedia.org/wiki/Eight_queens_puzzle`
- Different formulations are possible: N^2 0/1 vars, $2 \times N$ 1..N vars, N 1..N vars
- Demo `queens.ecl`

Exercise: playing with N-queens

Try to introduce different changes:

- Different variable ordering heuristic
- Different value ordering heuristic
- Arrays instead of lists
- `gfd` instead of `ic`
- Maybe something else

Observe changes in running time for larger problem instances.

Optimization

- Optimal vs any feasible solution
- branch-and-bound: `http://eclipseclp.org/doc/bips/lib/branch_and_bound/index.html`
- Demo `alldifferent-bb.ecl`

Real world

- http://eclipseclp.org/reports/index.html#Applications
- Almost any combinatorial problem
- Interfaces to Java, C++, Python, ...
- Development tools: testing, debugging, profiling, ...

More info

- ECLiPSe official site – `http://www.eclipseclp.org/`
- Book "Constraint Logic Programming using ECLiPSe" by Krzysztof Apt and Mark Wallace
- ELearning course by Helmut Simonis – `http://4c.ucc.ie/~hsimonis/ELearning/index.htm`
- Examples by Hakan Kjellerstrand – `http://www.hakank.org/eclipse/`