Declaratively solving tricky Google Code Jam problems with Prolog-based ECLiPSe CLP system

Sergii Dymchenko
Independent Researcher
sdymchenko@progopedia.com

Maria Mykhailova
Independent Researcher
michaylova@gmail.com

Introduction

Google Code Jam (GCJ) is one of the biggest programming competitions in the world. An individual GCJ round is usually 2-4 hours long and poses 3 or more problems. A solution is considered correct if it produces correct answers for all given test cases within a certain time limit (4 minutes for the "small" input and 8 minutes for the "large" one). GCJ competitors can use any freely available programming language or system.

ECLiPSe [1] is an open-source Prolog-based system that integrates various logic programming extensions, in particular constraint logic programming (CLP).

We claim that declarative programming with ECLiPSe is better suited for solving certain common kinds of problems offered in GCJ than imperative programming.


Problem: Star Wars

The essence of the problem is: you are given a set of N 4-tuples of integers \( (x_i, y_i, z_i, p_i) \). Find the minimal possible \( Y \) for which exists a triplet \( (x, y, z) \) such that for each original tuple:

\[
|x_i - x| + |y_i - y| + |z_i - z| \leq p_i Y
\]

This problem was one of the harder problems from round 2 of GCJ 2008, yet it can be almost trivially modeled and solved as a linear programming problem.

Direct translation of the problem statement to a model results in non-linear constraints, but these can be easily converted to linear constraints using the fact that:

\[
|X| \leq \text{Max} \equiv (X \leq \text{Max}) \land (-X \leq \text{Max})
\]

Initially all values from the initial domains are possible for all variables.

Problem: Triangle Areas

Given integer \( N, M \) and \( A \), find a triangle with vertices in integer points with coordinates \( X_i :: [0..N], Y_i :: [0..M] \), with an area \( S = \frac{1}{2}A \), or say that it does not exist.

The problem is almost perfect for solving with CLP. Variables are discrete, constraints are non-linear, and we are looking for any feasible solution. One vertex of the triangle can be chosen arbitrarily. To calculate the triangle area, place one vertex in \((0,0)\), then:

\[
2S = |x_2y_3 - x_3y_2|
\]

Running times for small (4 minutes time limit) and large (8 minutes) inputs

<table>
<thead>
<tr>
<th>Problem</th>
<th>Library</th>
<th>Small</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangle Areas</td>
<td>ic</td>
<td>0.2s</td>
<td>1.4s</td>
</tr>
<tr>
<td>Dancing With the Googlers</td>
<td>ic</td>
<td>timeout</td>
<td></td>
</tr>
<tr>
<td>Dancing With the Googlers</td>
<td>eplex</td>
<td>0.3s</td>
<td>1.7s</td>
</tr>
<tr>
<td>Star Wars</td>
<td>eplex</td>
<td>0.2s</td>
<td>0.4s</td>
</tr>
<tr>
<td>Mine Layer</td>
<td>eplex</td>
<td>0.2s</td>
<td>2.9s</td>
</tr>
</tbody>
</table>

Conclusions

Many GCJ problems that are hard to solve in time-restricted and stressful competition environment can be relatively easily modeled and solved in ECLiPSe.

Our declarative solutions for some problems require simpler and often fewer mental steps than possible imperative solutions in a language like C++ or Java.

Running times of our programs are several orders of magnitude smaller than the time limit imposed by GCJ rules.

1 https://code.google.com/codejam
2 http://www.eclipseclp.org/