## Project sheet 5: Competition

Prof. Dr. Bernhard Nebel, Dr. Stefan Wölfl, Dr. Julien Hué, and Matthias Westphal

July 16, 2012

If you want to participate, please hand in your code by July, 22 in the git repository. Working on this project sheet is not mandatory; there are no points.

## **1** The Competition

We will evaluate your solvers in the form of a competition. The authors of the top three ranked solvers will get a small prize.

This evaluation will be done by us starting next week, thus any changes you want to make should be done by Sunday, July 22. The evaluation will be presented by us on Wednesday, July 25.

## 2 What You Should Prepare

We will restrict instances to *binary* constraint networks in order to avoid additional programming work.

You already have implemented a selection of different algorithms and heuristics. For the competition we only want to run each solver once. Thus, you have to decide on a reasonable default selection for backtracking algorithms and heuristics. To this end, implement a --auto command line switch that sets the heuristics and algorithms as you want; your selection of preprocessing, maintaining arc consistency/backtracking algorithm, heuristics, is up to you. We will invoke your solver with:

./solver --auto problem\_instance.csp

Further, we require you to print the found solution in case of satisfiable problems and otherwise the information that it is not satisfiable. The output format for satisfiable problems must be:

SAT variable\_name: value, variable\_name: value, ...

E.g., for V = (var1, var2, var3) and a solution (1, 2, 3) you must output:

SAT var1: 1, var2: 2, var3: 3

For unsatisfiable problems output

UNSAT

Further, make sure not to print anything else to the output.

## **3** Competition Rules

The evaluation will be done on 64 instances *without* giving you access to these (blind evaluation). We will use a time-out of 5 minutes per solver and instance. For each correctly solved instance, the solver will get *points*. Each solver accumulates the points over the 64 instances. Solvers will be ranked according to the number of points they have accumulated.

To calculate the number of points for a solver that successfully solves an instances we will use a *purse-based method*:

- "solved-purse": there are 100 points for each problem which are distributed *equally* among all solvers that correctly solve this instance.
- "speed-purse": there are 100 points for each problem which are distributed relative to the speed (i.e., runtime) of each solver that correctly solves this instance:

$$F_{solver} := \frac{1}{1 + time(solver)}$$
$$points(solver) := \frac{100 \cdot F_{solver}}{\sum_{s \in solvers} F_s}$$

Note, the "solved-purse" gives points for solving instances, but also rewards solvers that tackle problems that could not be solved by the competition (thus rewarding success on hard problems). The "speed-purse" rewards fast solvers.

**Example 1.** *If 4 solvers correctly finish on instance 1 with running times* 20.5, 30.4, 80.8,140.0 *seconds:* 

- each of them will be given  $\frac{100}{4} = 25$  points for the solution.
- the "speed-purse" will give 47.6 points to the first, 32.6 to the second, 12.5 to the third, and 7.3 to the forth.

For this instance we have the following points (and preliminary ranking):

- 1. solver-1 72.6
- 2. solver-2 57.6
- *3. solver-3* 37.5
- 4. solver-4 32.3