COMP102P. Model checking coursework

Robin Hirsch

December 11, 2014

This coursework is compulsory and assessed, the deadline for submitting your program is 16th January 2015. To submit your coursework, use the link “Graph model checker” on the moodle page and submit a text file with your program.

In this coursework you have to parse first order formulas in a language with no function symbols and one binary predicate symbol ’X’ denoting the edge relation in a graph, so $X[xy]$ means that there is an edge from x to y. A binary connective is a character ’v’, $'\hat{}'$ or ’>’ (denoting or, and, implies). A variable is a character ’x’, ’y’ or ’z’ (three variables should be enough for this coursework). There are no constants and no function symbols, so a term is just a variable. A formula is defined by

$$\phi ::= X[ts] \mid -\phi \mid (\phi \circ \phi) \mid Ez\ \phi \mid Az\ \phi$$

where $\circ$ is a binary connective, t, s, z are variables. We will not include any spaces in our formulas.

Have a look at the program called “graph.c”. You can see in line 4 a statement

# include “yourfile.h”

Your job is to provide a textfile with your program. Your file should be called “firstnamefamilyname” (e.g. for me it would be “robinhirsch”). We will rename your submitted file as “yourfile.c” At a unix prompt, on one of the CS department’s servers, I will have a directory including the following files: (i) graph.c (ii) yourfile.h (iii) yourfile.c. Your marker will type

```
gcc -Wall graph.c yourfile.c -o output
```

If this does not compile you will get no marks. Even if you do not implement all parts of this coursework, make sure you have at least a dummy function for all functions declared in the header file.
If this compiles without errors we will then type

```
output
```

When prompted for input we will provide it. The main method calls a function called “parse(char *g)” from yourfile.c. If it correctly returns 0 for a non-formula, 1 for an atomic formula, 2 for a negated formula, 3 for a binary formula, 4 for an existential formula and 5 for a universal formula, then you will get one third of the marks. After this, the user is prompted to input data specifying the graph (how many nodes, which edges?) and after that the user is asked to specify a variable assignment V. The main method then calls a function “eval(name, edges, no_nodes, V)”. If this correctly calculates whether the formula “name” is true in the graph under the variable assignment V, then you will get full marks.