COMP102P. Model checking coursework

Robin Hirsch

January 13, 2016

This coursework is compulsory and assessed, the deadline for submitting your program is 15th January 2016.

Put all your code in a single file, based on the skeleton program 'graph_skeleton.c'. Do not use any C libraries except those already included in graph_skeleton.c. To submit your coursework, get on one of the department's managed unix machines (e.g. newgate.cs.ucl.ac.uk). Type

```
cp CODEFILE /cs/docs/academic/rhirsch/1styr/LOGINNAME
```

where LOGINNAME is your CS username (e.g. for me it would be ucacrdh). This will place a copy of your code in a directory. Only you have write permission for the file you create. Only a small group of administrators will have read access to your file. You should be able to see your file — just type

```
more /cs/docs/academic/rhirsch/1styr/LOGINNAME
```

to see the contents of the file you uploaded.

Have a look at the file graph_skeleton.c This is a skeleton program for you to complete. The program reads its input from a file called 'input.txt'. It sends its output to a file called 'output.txt'. I have posted a sample input.txt file and the corresponding output.txt file. Try your program on this input.txt file before you submit your code. You are advised to try your program on one of the department's managed unix machines. No manual checks will be done by us if your program fails to compile.

The program expects 6 examples. (const int cases=6). The first six lines of the input file specify 6 different examples. Each example consists of a string (the formula), an integer (no_nodes in graph), an integer (no_edges), then 2*no_edges more integers (specifying the edges of the graph) and finally three integers (specifying the nodes assigned to variables x, y and z). The entries are separated by a blank character.

The critical method is called "eval". Its first parameter is a string (the formula to be evaluated), the next two parameters are the list of edges and the number of nodes (in other words a description of the graph). The final parameter is a variable assignment for our three variables. The method is supposed to work out whether the formula is true in the graph under the given variable assignment or not. I suggest you start by parsing the formula and then evaluate it according to its type.

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.

Here are some sample formulas and graphs with the expected output.

fmla	nodes	edges	var_assig	true?
$AxEyX[xy]$	3	$\begin{pmatrix} 0 & 0 & 1 & 2 \\ 1 & 2 & 2 & 2 \end{pmatrix}$	$(1,1,1)$	yes
$AxEyX[yx]$	3	$\begin{pmatrix} 0 & 0 & 1 & 2 \\ 1 & 2 & 2 & 2 \end{pmatrix}$	$(1,1,1)$	no
$ExAyX[yx]$	3	$\begin{pmatrix} 0 & 0 & 1 & 2 \\ 1 & 2 & 2 & 2 \end{pmatrix}$	$(1,1,1)$	yes
$(X[xy] \to -X[yx])$	3	$\begin{pmatrix} 0 & 0 & 1 & 2 \\ 1 & 2 & 2 & 2 \end{pmatrix}$	$(1,1,1)$	yes
$(X[xy] \to -X[yx])$	3	$\begin{pmatrix} 0 & 0 & 1 & 2 \\ 1 & 2 & 2 & 2 \end{pmatrix}$	$(0,2,1)$	yes
$(X[xy] \to -X[yx])$	3	$\begin{pmatrix} 0 & 0 & 1 & 2 \\ 1 & 2 & 2 & 2 \end{pmatrix}$	$(2,2,1)$	no