VLSI Design Automation (EECE 6086C)

Ranga Vemuri

Home Work - 3

For this homework, you will work in teams (two students per team – select your own partner).

Each team will implement two programs – one for checking whether a given cover is a tautology (tc) and another for complementing a given cover (cc). One person should be primarily responsible for tc and the other for cc. Remaining work may be shared equally.

Both programs can be used as follows:

%tc < input_cover_file > output_cover_file

%cc < input_cover_file > output_cover_file

If the input cover is a tautology then tc generates a cover with the universal cube as the only cube in it. Otherwise it should generate a 'counter example' cover which should contain one or more minterms (or cubes) which are not covered by the input cover.

cc generates the complement of the input cover.

All covers files have the following format:

First Line: number of variables, n
Second Line: number of cubes, c
Next c lines contain the cubed, one per line in the usual cover table notation using 0's, 1's and –'s.

Example-1:
4
5
0-10-
--101
11-1-
01011
1---1

Example-2 (universal):
2
1
--

Example-3 (empty):
4
0

Benchmarks will be provided via blackboard in the next few days (Presently, a couple of covers to drive your development are provided. You may generate and use additional random covers if you wish.)

For each benchmark b, the following results must be reported:

1. Output of %tc < b.
2. Output of %cc < b.
3. Let c be the complement of b, generated using cc. Let u be the union of b and c (concatenate the cubes in b and c). Generate a u for each b and submit the output of %tc < u. (Note that u is expected to be a tautology.)
4. A table showing the execution time and memory for each run above of cc and tc.

In addition, your report must contain a description of the exact algorithms and data structures used for tc and cc. Provide any other information, results and discussion you may consider useful in evaluating your work. Include a clear description of how the work was divided. It is expected that the work would be divided approximately equally. Please explain if this is not so.

Include a retrospective describing what you think is the cause of the good/poor performance of your tools and how might you be able to improve the performance if you had more time?

Include the source code with appropriate comments and a make file in your submission.