# GMDL, HW #4

Oren Freifeld and Ron Shapira Weber
The Department of Computer Science, Ben-Gurion University

Release Date: 18/5/2022
Submission Deadline: 15/6/2022, 23:59

**Abstract**

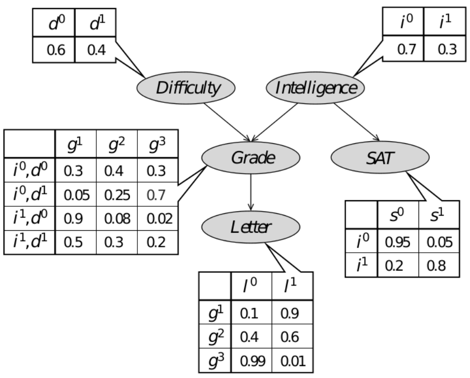
This assignment focuses on Belief Propagation in a Bayesian network.

## Version Log

- 1.00, 18/5/2022. Initial release.

**Remark 1** *When you draw a graphical model (*e.g.*, a factor graph), you are free to use any appropriate software tool to do that (*e.g.*, PowerPoint, LaTeX + TikZ, Inkscape,* etc.*). Our recommendation, however, for the most painless way to do it well, is that you use Or Dinari's fork of* **daft**: `https://github.com/dinarior/daft`. ◊

Recall that we learned a variant of the Belief Propagation algorithm that uses a factor graph. While we showed it in the context of MRFs (with an undirected tree structure), the exact same algorithm is applicable to Bayesian networks as long as their structure is a directed tree (not necessarily a directed rooted tree). That it, such a directed graphical model may be converted to an undirected factor graph and the latter will be a tree.




**Computer Exercise 1** *In the Student example we saw from Koller and Friedman (which also appears in the figure above), draw the associated undirected factor graph, implement the aforementioned Belief Propagation algorithm for* *that particular example (*i.e.*, your code doesn't have to be able to handle other cases – though you can make it more general if you choose to), and report the results you obtain, using that implementation of yours, for each of the marginals:* $p(i)$; $p(d)$; $p(g)$; $p(s)$; $p(l)$. *Note well: it is a requirement that your code will never compute the same message more than once. So use some bookkeeping to indicate which messages were already computed and reuse the associated results whenever possible.*