ToDo:

ADD: Read/Write sparse matrix encoding of M. (What’s a good format? Ideally something MATLAB can natively import.) (Elie will add.)

ADD: Write output of solver. This means outputting (raw) y, MonomialToRow encoding, the SymPy derived string. ALSO: Makes sure to print the NAMES of the random variables IN THE ORDER used by LearnParametersFrom graph! (Elie will add.)

GENERALIZE: Allow keyword arguments in functions to allow for file dumping or not.

GENERALIZE: Allow code to handle more than one expressible set. (Practice on triangle.)

Q: Why is function loading so slow? Anything we can do about it?

---Switch graph interface to NetworkX? (Elie will decide)

---Move code sections to specialized .py modules, to be loaded as needed. (Elie will start.)

---ADD DOCUMENTATION to code via Sphinx or ReadTheDocs or whatnot.

Conceptual homework:

DRAFT: Algorithm to determine all expressible sets

Algorithm to quickly identify marginal probabilities. (Preparing for more general expressible sets.)

Algorithm to compute products of marginal probabilities…

Things to teach:

TEACH Bora: Algorithm for finding all ai-expressible sets.

TEACH Bora: Algorithm to generate interesting incompatible distributions! (See if Bora has further insight.)

TEACH Bora: Convex hull algorithm to extract all inequalities. (Investigate Polytope/Polymake packages, think about interface to something more sophisticated such as PANDA/Sympol.)