

Report

Works done before implementation:

1. Create the environment with the following packages:
`matplotlib, networkx, numpy`
2. Understand the Factor class, including the functions and the data structures, by reading and running the `factor_readme.py`.
3. Understand the code given in the function `factor_product()`, especially the code used to get the mapping from variables in one factor to those in another factor.

Functions implemented and effects:

`factor_product()`: Compute the product of two factors.

`factor_marginalize()`: Sums over a list of variables to obtain the marginal.

`observe_evidence()`: Modify a set of factors given some evidence.

`factor_sum()`: Compute the sum of two factors.

`factor_max_marginalize()`: Marginalize over a list of variables by taking the maximum value.

`compute_joint_distribution()`: Compute the joint distribution.

`compute_marginals_naive()`: Compute the marginal naively.

`compute_marginals_bp()`: Compute single node marginals for multiple variables using sum-product belief propagation algorithm.

`map_eliminate()`: Obtains the maximum a posteriori configuration for a tree graph given optional evidence.