
CSE 598
Algorithms for Synthesis and Optimization of Digital Systems

Programming Project 3
Kernel Computation

Programming Assignment 3: Kernel Computation

In this project, you are to design and implement an algorithm in Python that computes the kernels of a set of logic expressions (Algebraic) using a cube-literal matrix.

The input to the program should be a file with a logic expression of the form $F1 = \text{exp1}$

❖ A collection of logic expressions each representing a Boolean function, e.g.

1. $F1 = \text{exp1};$

2. $F2 = \text{exp2};$

Etc.

The output of the program should be a list of kernels of each function.

1. $F1 = (K11), (K12), \dots$

2. $F2 = (K21), (K22), \dots$

In addition, output the binary matrix M where the rows represent the *co-kernels*, and the columns represent the set of all cubes among the functions.

A suggestion: Functions that process M associated with a function F should not create copies of M , but instead pass three arguments (M , Rows, Cols), where Rows and Cols are binary vectors that represent which rows and columns are blocked.

Another Suggestion: Make your program recursive, to compute cubes at all levels, and show how to avoid repeated computation of the same kernel.