Before the meeting:

- Drew diagrams representing the transformations done by the separableConvolution2D.scala application.
- Went through the steps in separableConvolution2Drewrite.scala.

Questions:

- In the base strategy, there seems to be a missing parameter. - SOLVED in meeting, there is an implicit parameter is given as argument.

During the meeting:

- Develop systematic way of turning one lift strategy into another. Have a bag of ‘rules’ (strategy[Lift]), a beginning and a goal expression, and do something like backtracking to search the applications of rules at different points in the beginning and subsequent expressions to see if we can transform beginning to goal.
- This will give an actual code implementation for the purpose of writing the dissertation.
- nbh = neighbourhood
- “<+” = Prefer the left choice, but if it doesn’t work out, try the choice on the right
- Oncetd = Once Top Down = Common way of traversing the Abstract Syntax Tree of a program in related work, attempts to apply the strategy to each node in the AST, if it fails, go down one level and recurse. To be clear, oncetd will traverse every element until it succeeds, or will traverse all nodes and fail.
- topdown(topdownall) = the same thing but will only succeed if it can apply the strategy at every node in a given level, not just one. Not really interested in this for Lift.

Next Steps:

- Fine if it is naive, but give a try writing such a system that at first will just take one start, goal, and single rule that should take one and turn it into the other, and try to get that to work.
- Meeting next week cancelled as everyone is busy with coursework.