Student id: C1981034

Task 1

Code Syntax and Execution

Syntactically correct and executes.

Validity of Solutions and Optimisation

Representation of the expertise is somewhat unusual (better to have a list of triples as input, [("REVIEWER","PATENT","EXPERTISE"),...]), but the particular approach for 4 reviewers creates this.

Constraints are correctly checked with check, but inefficient as the expertise count is generated multiple times, and similarly for the counts (could be done in one go, taking head and last).

Maximum experts constraint is not checked.

Note, solutions generates all possible assignments, but also adds these to a list, so that the list you construct does not only contain the final valid assignment, but all previously partial assignments that pass the partial test. This is much more than what really needs to be generated and tested. Moreover, it makes checking the constraints actually quite complex and you do recheck all constraints each time instead of only checking them for the extension.

Documentation

Each function and usage is explained. There is a good attempt to explain the strategy.

Result

Merit : Code is valid without syntax errors and creates a valid output for every valid input (or produces a suitable error message for valid cases it cannot process). A suitable attempt to solve the problem has been made, that will find all solutions (if there are any). The attempt has been well documented. [30 - 34]

Marks: 33 / 50

Student Id: C1981034

Task 2

Two Arguments for and against Logic Programming

Valid arguments presented and a reasonable attempt has been made to link these to the problem. The link to the problem could have been made stronger, though, for some arguments.

Two Arguments for and against Functional Programming

Valid arguments presented. The absence of side effects is mainly useful for parallelisation and correctness proofs. The point on robustness/crashes is very weak. There are variables in Haskell, but they are immutable, so the first argument against should be considered more carefully. The link of the arguments to the problem remains quite generic.

Justification for Choice of Programming Paradigm

Valid justification, that considers arguments for and against. It could have been more concretely represented in the problem context.

Result

Distinction : The required number of valid arguments has been submitted. They show a clear understanding of the logic and functional programming paradigms and the underlying theoretical concepts and/or realisations on programmable machines and how these relate to the problem. The choice of paradigm to solve the problem is well justified based on these arguments and shows an understanding of related performance issues. [35 - 50]

Marks: 35 / 50

done

Total

Total Marks: 68 / 100