LOGIC VS. FUNCTIONAL PROGRAMMING RELATED TO PATENT REQUEST ALLOCATED PROBLEM

CMT304 – Programming Paradigms

1. Introduction

In recent years it has become far more normal for developer choosing a non-Imperative programming language to deal with particularly for some lazy evaluation or pattern match problem, just like this patent requests allocating problem. Actually, for this assignment program, both logic and functional programming having both positive and negative influence in equal measure when we coding. So, to analysis which kind of programming should be used in this problem is very necessary.

2. Logic Programming:

:

Argument for using logic programming:

1. Base on the fact-rule-output model of logic programming, we need to focus on ''what is the problem'' rather than ''how to solve the problem''. It's easy for the program generates a suggestion or answer from a specific complex model. Therefore, in this problem we just need to enter the conditions which given in the question into the program and let the program give us the answer instead of thinking about it ourselves because in this way we can save time and costs significantly. So, in this viewpoint we choose a logical programming language for this problem is very necessary.

2. We can see coding with a logic programming language is easier as the syntax is very short and precise. We may find that this program only needs to be given a set of established rules to get the answer. Therefore, it is really not necessary to consider too many software architectures for this problem. In this way, who choose a logic programming language to solve this patent request allocated problem may become easier and clearer than those who code with an imperative language such as java, which saves time and costs

Argument against using logic programming:

1. The logic programming language is completely different from an imperative language, there are very few types of language statements, but they are very expressive, therefore we may need rich experience with ASP to handle this problem more flexible. The lack of logic programming experience can really give beginner developer a huge challenge.

2. For some types of problems logic programming can't efficiently solve which made execution can be very slow if we don't optimize it. In this problem, if all the cases of assigning a series of patent requests to n members are taken into account, the worst time complexity can reach O (2 ^ n), so the choice of logical language should still be carefully considered.

3. Functional Programming:

Argument for using functional programming:

1. Due to functional programming depends only on the arguments input to the function which does not suffer from side effects and provide a protected environment. Therefore, we only need to focus on the coding without worrying about some unexpected error, such as program crashes due to the release of space when assigning patent requests. In this way, the programs we write with functional programming are highly robust, and the highly cohesive code also makes these functions easy to reuse.

2. Functional programming is not completely different from imperative programming language programming, which is common exist in many popular languages like JavaScript, even developers who have not learned some pure functional programming language can also easily get started, in this assignment, 'do' and 'monad' in Haskell enables functional programming languages to write like an ordinary imperative language, so using Haskell to develop is bound to be easier and more familiar than ASP.

Argument against using functional programming:

1. There is no concept of variable in pure functional programming language like Haskell, so, all data cannot be changed, for example we cannot insert value to List or Map in Haskell, which made us so hard to think in a normal way to programming. What's more, all 'variables' always exist during the running of the program therefore it will also seriously occupy the operating resources and cause the running speed to be insufficient, which tend to require a large amount of time and memory.

2. Same to logic programming, it may be not very easily for new developer in some case, although functional programming has been in existence for many years, there are still many problems to be solved in engineering that want to use functional programming on a large scale. If we don't understand functional programming deeply enough, it will lead to an obscure situation to our assignment project.

4.Paradigm chosen for this problem

It is true that both logic programming and function programming can be used in this patent requests allocating problem. While I tend toward viewpoint that logic programming is better here by most of classmates, I would have to support my opinion that It makes more sense to use functional programming language (Haskell) here.

The main reason why people think we should choose logic programming is that It's easy for the program generates a suggestion or answer from a specific complex model, it's sounds correct because it fits the requirements of the question, which generating answers from a series of conditions. What's more the syntax of a logic programming is also very short and precise, from this theory, we seem can quickly solve the problem with logic programming.

However, the good syntax argument given above can be considered from the opposite angle. Firstly, rather than the positive feeling of the syntax and grammar, the logic programming language is completely different from an imperative language, we may need rich experience with it before we start handling this problem, but the function programming is more friendly to us. Secondly, from the programming language characteristics point of view, the program may crash due to the release of space when assigning patent requests, the rise in demand for security and code reuse is more likely for us to choose function programming rather than logic programming.

Based on the above discussion and analysis, we can see that functional programming is more suitable for this problem, even if there are many advantages in the logic programming.