Mastery Check Questions 1

|  |  |  |
| --- | --- | --- |
| **Topic** | **Level** | **Question** |
| Scope | Fundamental | What are the different types of declarations in Source? |
| Scope | Fundamental | What are the differences between name declarations and name occurrences, function declarations with function applications? |
| Scope | Fundamental | In sub model, how do you replace the name occurrences (aka how do you determine what each occurrence refers to) |
| Scope | Intermediate | Match these naming occurrences (From RA 21) |
| Scope | Intermediate | What are the differences between function declaration and const declaration?  Hint: nested function (maybe there are others too) |
| Scope | Intermediate | Which scope does math\_e, constant belong to? And is it possible to redefine math\_e in Source Language by using some of the languages construct (such as {}, const, recursion, etc)? If it is, how? |
| Scope | Advanced | In your substitution model, you always replace the function application with the function body (or the code within {}). If you think about it further, very rarely if not none there is any programming language that uses sub model as the mental evaluation model, why? Are there any gaps in sub model? Or can you justify that the model is perfect |
| Scope | Advanced | Does this work? State your reasoning!  What if I moved the fun (x + y) above const z = 10; Will this work? |
| Scope | Advanced | Modified from RA 21, will this work?    What kind of error would it give if it doesn’t work or prove me that it works! |
| HOF | Fundamental | How do you differentiate which function are higher-order functions? |
| HOF | Fundamental | How can you transform a function declaration into a constant declaration? And is it possible for **all** functions to be declared as constant declaration? |
| HOF | Fundamental | What are the differences between these two expressions, g(f) and g(f(2))? |
| HOF | Intermediate | How do you do recursion, using only lambda construct without any function declaration? To demonstrate your understanding, try to write a factorial function using const declaration only! |
| HOF | Advanced | Consider this lambda function    What is the result of the evaluation of this programme? |
| HOF | Advanced | HOF can be used to implement data structures such as pair, consider  ***const p = (x, y) => flag => flag < 1? x: y***; This is a very simple implementation of pair purely using flag (and not using other constructs). With this implementation of pair, how then will you implement the head, tail and a list using this pair (aka list (…)) |
| Sub model / Processes | Fundamental | Explain the differences between iterative and recursive processes! What are their implications programmes such as space and time complexity, code length, code quality, etc |
| Sub model / Processes | Fundamental | Explain the differences between applicative and normal order reductions then explains what the implications on the evaluation of the programme are! |
| Sub model / Processes | Fundamental | Consider this,    Does this program perform any addition? |
| Sub model / Processes | Intermediate | Can switching the order of operands reduce the number of deferred operations? (Source evaluates from left to right for operations that have equal precedence) |
| Sub model / Processes | Intermediate | If a program gives rise to a recursive process, can you transform it into a program that gives rise to an iterative process by switching the operands of additions / subtractions / multiplications? |
| Sub model / Processes | Advanced | Does this give programme give rise to a recursive or iterative process? |
| Sub model / Processes | Advanced | Taken from RA 21, Does this process give rise to iterative or recursive process?    And if it’s recursive, what are the deferred operations here? |