

Q&A Session for Programming Languages Lecture 23

Session Number: 1206826447

Date: 2020-12-1

Starting time: 15:11

ANON - 14:46

Q: For questions 3 and 4, should we type the output as 7, 7 OR 7 7

Priority: N/A

Konstantin Kuzmin - 14:46

A: Either way is fine.

ANON - 14:59

Q: I can't type today sorry. for question 1: is it [row,col,depth]?

Priority: N/A

Konstantin Kuzmin - 15:07

A: This is something you will need to decide based on the examples given in the lecture. Think about how elements are stored in memory.

ANON - 14:58

Q: for question 1

Priority: N/A

ANON - 14:58

Q: oh whoops. for question. is it (row,col,depth)?

Priority: N/A

ANON - 15:21

Q: Hi, is there a maximum number of people in one team in hw7?

Priority: N/A

Ana L. Milanova - 15:21

A: Minimum number is 1, and maximum number is 3. Setup is same as in HW5.

Konstantin Kuzmin - 15:22

A: Yes. Please see: https://www.cs.rpi.edu/~milanova/csci4430/Lec22_QA.pdf

ANON - 15:21

Q: So the Primary Disadvantage is the Run-Time Cost due to Dynamic Checking?

Priority: N/A

Ana L. Milanova - 15:23

A: Yes, I think there is a general agreement that runtime cost is the primary disadvantage.

ANON - 15:23

Q: How is protected different in C++ vs. Java?

Priority: N/A

Ana L. Milanova - 15:28

A: One key difference is that in Java, which has package structure, protected allows classes from the same package to see the field/method. In C++, protected allows access from subclasses and I believe "friend" classes as well.

Konstantin Kuzmin - 15:33

A: In Java, protected gives access to class, package, and subclasses. Here's the full access modifier table: <https://docs.oracle.com/javase/tutorial/java/java00/accesscontrol.html>

Konstantin Kuzmin - 15:36

A: There is also a table that compares access modifiers in several languages here: https://en.wikipedia.org/wiki/Access_modifiers

ANON - 15:23

Q: What does package mean in Java?

Priority: N/A

Konstantin Kuzmin - 15:26

A: Package is defined as all classes that have been assigned to this package using the "package" directive. (Or the default package if there is no package directive.) Packages in Java are loosely analogous to namespaces in C++.

ANON - 15:24

Q: Does implicit creation just mean creating an object with a copy constructor?

Priority: N/A

Steven Haussmann - 15:32

A: Two separate concepts -- implicit creation just means that, when you declare the variable, the object has to be constructed, even if you don't explicitly invoke a constructor. The copy constructor is one way that this implicit construction could occur.

ANON - 15:25

Q: does subtype polymorphism always result in dynamic method binding?

Priority: N/A

Steven Haussmann - 15:30

A: Nominally, you could avoid it via static analysis -- e.g. by concluding that a variable will always contain a specific subclass. But, yes, dynamic dispatch is necessary in general.

ANON – 15:27

Q: If languages like C don't follow the open/closed principle but are still widely used, does that mean the design principles are just suggestions and it's fine if they are not followed?

Priority: N/A

Konstantin Kuzmin – 15:28

A: Following design principles makes your code better (safer, easier to maintain, etc.)

ANON – 15:35

Q: should we understand polymorphism as to how it applies to certain features in C++/Java or are we focusing more on the concept?

Priority: N/A

Ana L. Milanova – 15:37

A: We will be focused on the general concept, but we will discuss mechanisms in specific programming languages, including C++ and Java.

ANON – 15:37

Q: is method overloading considered ad-hoc polymorphism?

Priority: N/A

Ana L. Milanova – 15:37

A: Yes.

ANON – 15:38

Q: What are some examples of parametric polymorphism in imperative languages?

Priority: N/A

Konstantin Kuzmin – 15:38

A: Generics in Java and templates in C++ would be such examples.

Ana L. Milanova – 15:39

A: We give a lot of examples in lecture from now on.

ANON – 15:42

Q: if explicit parametric polymorphism is implemented by creating multiple copies, how does it work if we can also use custom types in a template/generic?

Priority: N/A

Ana L. Milanova – 15:45

A: What do you mean by "cusome types"?

ANON – 15:47

Q: By custom types, I mean creating our own objects/classes.

Priority: N/A

Ana L. Milanova - 15:50

A: The compiler will replace the type parameter with your class and then type check. The problem is, at least with older versions of C++, that if your class does not provide the operations the generic code expects, you get tons of errors, pointing to STL code

ANON - 15:48

Q: Does implicit parametric polymorphism only occur in dynamically typed languages?

Priority: N/A

Ana L. Milanova - 15:54

A: The examples I am familiar with, yes, they all come from dynamically typed languages.

Ana L. Milanova - 15:55

A: Statically typed languages typically make use of a type system that follows standard type theory and explicit parametric polymorphism.

ANON - 15:49

Q: Wait so without (Num a) Haskell would throw an Error since it uses Static Type Checking?

Priority: N/A

Ana L. Milanova - 15:51

A: Yes, if you write the type signature without the (Num a) predicate, that will be a type error. If you omit the signature altogether, Haskell will infer it correctly, I am pretty sure.

ANON - 15:50

Q: Are there no generics in Haskell because of the types? I don't completely understand what slide 28 is saying

Priority: N/A

Ana L. Milanova - 15:53

A: There are generics in Haskell. sum is a "generic function" because it can work on multiple types, as long as those types are instances of Num.

ANON - 15:51

Q: From slide 26, what is a bounded type in Java?

Priority: N/A

Konstantin Kuzmin - 15:53

A: The bound introduced with the "extends" keyword puts a restriction on what the type parameter can be. E.g., "E extends Number" means that when the type is instantiated, E has to be a Number or a subtype of Number.

Steven Haussmann - 15:54

A: They let you constrain the types you can parameterize something with -- e.g. "this type must extend List"

ANON - 15:54

Q: in scheme, does implicit constraint mean it takes in a function?

Priority: N/A

Ana L. Milanova - 15:57

A: By "implicit constraint" I meant that the function code imposes certain constraints on the types of the parameters.

Steven Haussmann - 15:58

A: contrast this with an explicit constraint, where we deliberately declare a limitation. Implicit constraints come from how the parameters are used.

ANON - 15:56

Q: So this Syntax is ML Style?

Priority: N/A

Steven Haussmann - 15:57

A: the syntax being shown here is from Haskell; the polymorphism itself is what is "ML-style"

ANON - 15:59

Q: Oh that's what I thought. Just forgot that $\lambda x \rightarrow$ Notation is used in Haskell

Priority: N/A

ANON - 16:00

Q: So (f True) does not apply f but rather instantiate it as the appropriately typed function?

Priority: N/A

Ana L. Milanova - 16:01

A: The type system instantiates the type. Haskell is able to type it then, and the checker passes. At runtime, (f True) does entail an application of f on argument True.

ANON - 16:01

Q: I'm confused what was meant by the fact that we cannot do this with a function argument?

Priority: N/A

Ana L. Milanova - 16:02

A: We cannot infer types. If we have let-bound polymorphic function, we can infer types. If we have lambda-bound polymorphic function, we cannot infer types.

ANON - 16:01

Q: what does functions defined at let bindings? What makes let polymorphism more specific than explicit parametric polymorphism?

Priority: N/A

Ana L. Milanova - 16:03

A: It is special in the sense that it's expressive and in the same time type inference is practical. General polymorphism is more expressive, but type inference is undecidable.

ANON - 16:08

Q: Since the final exam is in 2 weeks, what is the format of the exam? Will there be practice problems posted soon?

Priority: N/A

Ana L. Milanova - 16:18

A: The exam is cumulative. Possibly greater emphasis on last third of material. Format will be similar to format of previous two tests. Practice problems will be coming by the end of the week.

ANON - 16:10

Q: So when we say that Lambda Bound Polymorphic Functions are Undecidable for their Type Inference, this is because we do not know the Intended Type Signature for these Functions unlike if we defined them locally in a Let Binding?

Priority: N/A

Ana L. Milanova - 16:17

A: Yes, essentially.

ANON - 16:17

Q: is the final exam going to be stricly from 6:30 to 9:30pm? or is there going to be a range of times we can start while taking 3 hours?

Priority: N/A

Ana L. Milanova - 16:20

A: We are going to make this decision soon and we will post more details on the Forum once we make the decisions.

ANON - 16:20

Q: How long will we have to take the final?

Priority: N/A

Ana L. Milanova - 16:21

A: I expect it will be 3 hours, as scheduled. Look for announcements on the forum. There are still decisions to be made, and we'll make announcements as soon as we make the decisions.

ANON - 16:22

Q: On slide 37, what are the differences between the two?

Priority: N/A

Ana L. Milanova – 16:24

A: In terms of the final result of evaluation, they are the same. Both yield 1. In terms of the static type system (of Haskell), they are different. The type system accepts (1) but rejects (2).