Advanced Type Theory

For the masses

Subtyping

When a set of values in one type, T, is a subset of the set of values in another type, U

we say that T is a **subtype** of U

Subtyping

So $[1\ to\ 10]$ is a **subtype** of int

But also int is a **subtype** of num

$int \in num$

functions written to operate on elements of int also operate on elements of num

Variance

$$A : nat \in int \in num$$

Given A, what types are acceptable subtypes of the following function

$int \rightarrow int$

Variance

$$int \to num$$

Being **less specific** about the return type does not require us to change the internals

Variance

$$nat \rightarrow int$$

Being **more restrictive** of the function input, does not require changes to the function internals

Variance and Generics

What happens to the subtyping relation of generics?

Consider `Source<int>`, that we can `get` the value from.

Variance and Generics

`Source<int>` can be treated as `Source<num>`

Variance and Generics

Now consider `Sink<int>` that we can put the value in.

Variance and Generics

`Sink<int>` can be treated as `Sink<nat>`

Variance and Generics

Thus `Source<int>` can be treated as `Source<num>`

While `Sink<int>` can be treated as `Sink<nat>`

Co-variance

Since

$$nat \in num$$

and

`Sink<nat>` $\in$ `Sink<int>`

the relation is the same (or co-)

Contra-variance

Though

$$int \in num$$

and

`Source<num>` $\in$ `Source<int>`

the relation is opposite (or contra-)

Open Recursion

Open recursion is the ability for one method body to invoke another method of the same object via a special variable. The special behavior of this variable is that it is late-bound, allowing a method defined in one class to invoke another method that is defined later, in some subclass of the first.

Dynamic dispatch

The process of selecting which implementation of a polymorphic operation (method or function) to call at run time.

Subtyping vs Inheritance

$A \in B$, if every function that can be invoked on an object of type A can also be invoked on an object of type B.

$A\ extends\ B$, type B inherits from another type A if some functions for B are written in terms of functions of A.

Nominal subtyping

in which only types declared in a certain way may be subtypes of each other

Structural subtyping

in which the structure of two types determines whether or not one is a subtype of the other.

Widening

A conversion from a subtype to a supertype is called a widening conversion. It is called a widening conversion because it goes from a smaller type(the subtype) to a bigger type

Narrowing

A conversion from a supertype to a subtype is called a narrowing conversion. It is called a narrowing conversion because it goes from a bigger type (supertype) to a smaller type (subtype).