15-312 Lecture on Sum Types

BREAKING NEWS: Starbucks™ Adopts Variant Types

The easiest way to understand variant types is to got to Starbucks for a coffee. Your order will be placed in a paper cups like the one to the right.

This paper cup is a variant type, a container where you can put any of the types of coffee listed on the back side. Using the concrete syntax,

$$\begin{array}{rll} \text{starbucks\_cup} = [& \text{Regular} & : \text{RegularCoffee} \\ & \text{Cappuccino} & : \text{CappuccinoCoffee} \\ & \text{Americano} & : \text{AmericanoCoffee} \\ & \text{Latte} & : \text{LatteCoffee} \\ & \text{Espresso} & : \text{EspressoCoffee} \\ & \text{Macchiato} & : \text{MacchiatoCoffee} \\ & \text{Mocha} & : \text{MochaCoffee} \quad] \end{array}$$

When a Starbucks employee processes your order for a short cappuccino, giving you a cup of the delicious beverage with a checkmark on the item "Cappuccino", he has just built an injection: he has placed a Cappuccino (to which we have ascribed the type "CappuccinoCoffee" above) into a cup, therefore obtaining an object of type "starbucks_cup":

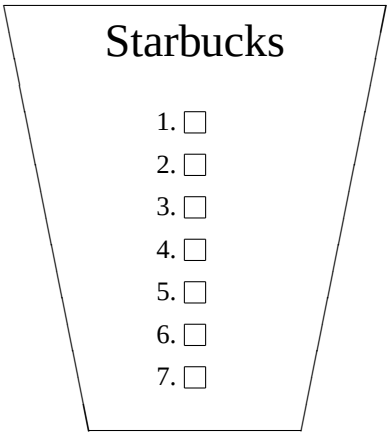
$$\text{Your\_Cup} = \mathsf{inj}[\text{starbucks\_cup}; \text{Cappuccino}](\text{short\_cap})$$

Now the cashier will use a case statement to figure out the price that you need to pay:

$$
\begin{array}{rlll}
\mathsf{case} & \text{Your\_Cup} & & \\
\mathsf{of} & \text{Regular} & => & \$1.95 \\
| & \text{Cappuccino} & => & \$3.25 \\
| & \text{Americano} & => & \$2.15 \\
| & \text{Latte} & => & \$3.25 \\
| & \text{Espresso} & => & \$2.50 \\
| & \text{Macchiato} & => & \$2.50 \\
| & \text{Mocha} & => & \$2.95
\end{array}
$$

N-Ary Sums

What would this same example been like if we had used n-ary sums? The cup would look like:



and the corresponding sum type would be:

$$
\begin{array}{rl}
\text{starbucks\_cup} = \mathsf{sum}(\text{RegularCoffee} & \\
& \text{CappuccinoCoffee} \\
& \text{AmericanoCoffee} \\
& \text{LatteCoffee} \\
& \text{EspressoCoffee} \\
& \text{MacchiatoCoffee} \\
& \text{MochaCoffee} \quad)
\end{array}
$$

It is your job to remember that item 2 is a cappuccino — not a great business model! When you order this cappuccino, the employee would perform the following injection:

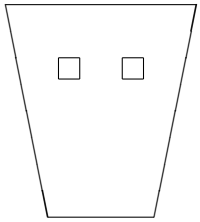
$$\text{Your\_Cup} = \mathsf{inj}[\text{starbucks\_cup}; 2](\text{short\_cap})$$

and the cashier would use the following table when it is time to pay:

```
case  Your_Cup
  of  1      =>  $1.95
   |  2      =>  $3.25
   |  3      =>  $2.15
   |  4      =>  $3.25
   |  5      =>  $2.50
   |  6      =>  $2.50
   |  7      =>  $2.95
```

Binary Sums

I don't know if it is true, but you can imagine that in the old days Starbucks was serving just two beverages: coffee and tea. It was a small company, and printing anything on a cup was expensive. Instead, it was buying its cups from a provider that put only two squares to mark on the cup:



(Cups were also smaller back then.) This starbucks start up had decided that if the customer ordered a coffee, the employee was to mark the left box, and for a tea, the right box would be checked. So, the type of this cup was

$$\text{early\_starbucks\_cup} = \mathsf{sum}(\text{Coffee}, \text{Tea})$$

You could not order a cappuccino then, but if you wanted a good old cup of Joe, the employee would perform a left injection:

$$\text{Your\_Cup} = \mathsf{inj}\text{l}[\text{earlystarbucks\_cup}](\text{joe})$$

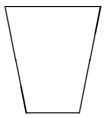
and the cashier would do a very very simple case analysis to decide how much to charge you:

```
case  Your_Cup
  of  inl(_)   =>  $0.95
   |  inr(_)   =>  $0.55
```

Back then, even Starbucks had inexpensive coffee!!!

Nullary Sums

Given all this, what is a nullary sum type? It is a cup where you cannot put any beverage, maybe because the lead is sealed:



We represent this with a sum with no alternatives, which is often called void:

$$\text{sealed\_cup} = \mathsf{void}$$

You can imagine this cup as a scam on the Internet, something that Starbucks does not carry.

Because Starbucks does not carry it, an employee does not know how to fill it, and there are no injections.

Similarly, if for some reason you show up with such a cup and try to pay, the cashier does not know what to charge you: there is no beverage that can go in it and so it's a case with zero branches.

$$\mathsf{case}\quad \text{Your\_Sealed\_Cup of } ?? => \{\}$$

This is an event that should never happen!