Poker Hand

by ToF

July 29, 2014

$\lambda \quad \lambda \quad \lambda$

1 Types and Functions

If we described the program we need to write as one function, what would be the name and type of this function?

1 *main* :: IO()

What would that function do?

2 Compute the scores extracted from the data in the standard input, and print them on the standard output.

How would you write this?

3

```
main :: IO()
main = interact computeScores
```

How would you write that function: *computeScores* ?

4

```
computeScores :: String → String
computeScores = unlines . mystery . lines
```

Your function breaks the input data into separate lines, processes all those line through a mysterious function, then assemble the result into a single output block of text. Why the name *mystery* ?

5 That's because I don't know what function I should write yet.

Let's try to specify it, then. What is the information we have about the function so far?

6 It consumes entry lines and yields those same lines with some score info added. So both input and output type are lists of Strings.

```
mystery :: [String] → [String]
```

How would you describe a score?

[7] It's the combination of either the kind of hand the player made, or nothing if the player folded, and value denoting if that player is the winner of the game or not. Also there can be more than one winner.

How would write the type *Score*?

[8]

```
type Score = (Maybe Kind, Bool)
```

If it was given a list of entry lines, *and* a list of *Score*s, what would a function named *appendScores* produce?

[9] It would produce the correct presentation of scores, according to the rules of Poker. And the function type would be:

```
appendScores :: [String] → [Score] → [String]
```

Can we specify this function?

[10] Yes.