Poker Hand

by ToF

November 10, 2011

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

1 Solving the problem

Did we solve our problem?

Not yet.

What do we need to do, then?

Mark the lines from the input with the ranking of the hand, and suffix the bast one with "(winner").

What test should I write?

Write the simplest test you can think of.

What is the trivial case for a function that should mark lines?

No hand at all.

Ok

```
markResults [Nothing] ~?= [""]
```

If we don't have a hand, then there is no mark.

■ I see. Here's the function:

```
markResults :: [Maybe Ranking] → [String]
markResults _ = [""]
```

■ Done.

Here's my next case.

```
markResults [Nothing, Just Pair]
            ~?= ["","Pair  (winner)"]
```

■ Ok. I'll just add a pattern:

```
markResults :: [Maybe Ranking] → [String]
markResults [Nothing]   = [""]
markResults [Nothing, Just Pair] = ["", "Pair  (winner)"]
```

■ It's a *fake*, as usual.

Do you see a possible refactoring here?

I see a map:

```
markResults :: [Maybe Ranking] → [String]
markResults = map mark
    where mark Nothing = ""
          mark (Just Pair) = "Pair  (winner)"
```

■ Refactoring done.

Here's a new case:

```
markResults [Nothing, Just Pair, Just HighCard] ~?=
                    ["","Pair (winner)","High Card"]
```

■ We can have several hands. The best one is the winner. There's *non-exaustive patterns* error in our code, now.

■ Sure. Here's a fix:

```
markResults :: [Maybe Ranking] → [String]
markResults = map mark
    where mark Nothing = ""
          mark (Just Pair) = "Pair (winner)"
          mark (Just r) = show r
```

■ It's still a *fake*.

How can we remove the *fake*?

By comparing each value in the list with the maximum value in the list.

```
markResults :: [Maybe Ranking] → [String]
markResults rs = map mark rs
    where mark Nothing = ""
          mark v@(Just r)
              | v == maximum rs = show r ++ " (winner)"
          mark (Just r) = show r
```

■ It works!

Can you remove duplication?

Yes.

```
markResults :: [Maybe Ranking] → [String]
markResults rs = map mark rs
    where mark Nothing = ""
          mark v@(Just r) = (show r) ++ if (v ==
                maximum rs) then " (winner)" else ""
```

■ Done.

Could we have pattern in lieu of the if then else?

Yes.

```
markResults :: [Maybe Ranking] → [String]
markResults rs = map mark rs
    where mark Nothing = ""
          mark (Just r) = (show r) ++ winner (Just r)
          winner v | v == maximum rs = " (winner)"
          winner _ = ""
```

■ Done.

And we could avoid computing the maximum at each line.

You are right.

```
markResults :: [Maybe Ranking] → [String]
markResults rs = map mark rs
    where mark Nothing = ""
          mark (Just r) = (show r) ++ winner (Just r)
          winner v | v == m = " (winner)"
          winner _ = ""
          m = maximum rs
```