Report: HW 5 and 6 Clojure

Amin Babar

10/29/18

CPSCI – 220

Clojure was very different compared to the previous languages I had experience with in computer science. This was my first functional language.

Clojure uses outfix operators instead of using infix operators. So 2 +2 in clojure is (+ 2 2). So the way the code is typed is different compared to the languages I have had experience with in the past.

Another unusual thing about clojure that I found hard to get used to was the immutability of states in Clojure. Since we cannot change states of variables, there are no for loops because you cannot keep count of the loop you are in. Having had experience with C++, python and ruby, it was difficult to let go of the loop structures and start using the recursive functions. Eg in the first homework we were required to square the digits of a number. This task would have been really simple in any other language I know, but I could not index the same way in clojure.

```
Eg.     (map #(* (- (int %) (int \0))
                 (- (int %) (int \0)))
                  (str number))))
```

Recursion in clojure had a stack limit so any function that would result in multiple function calls was written down in a tail recursive form. Also, another new feature that I came across in clojure

that I had not come across before was the lazy sequences. I found lazy sequences really helpful and efficient.

Overall, I found clojure hard because I was not used to the different way of typing and thinking but I assume with more practice in functional languages, clojure would be a great tool to have to work on projects. The code in clojure looks more elegant and is more succinct compared to other scripting languages.