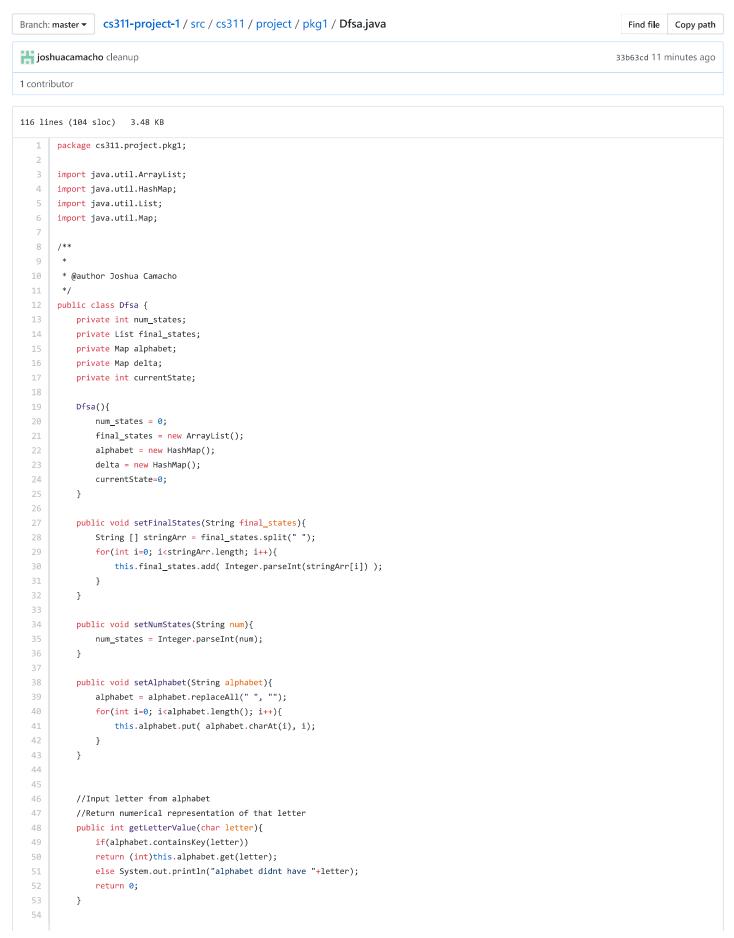
## joshuacamacho / cs311-project-1



```
// Gives a value for delta
 56
          // A sequence of transitions of the form (p a q)
          // p current state
 58
          // a input
          // q next state
          public void deltaPush(String set){
 61
              if(set.charAt(0)=='('){
 62
                   set = set.substring(1);
 63
 64
              if(set.charAt(set.length()-1)==')'){
                   set = set.substring(0, set.length()-1);
              }
              //allow for easier state transition definitions
              //a can be multiple symbols, setting multiple rules
 69
              String [] items = set.split(" ");
 70
              if(items[1].length()>1){
                  String state = items[0];
 72
                  int nextState = Integer.parseInt(items[2]);
                  String inputs = items[1];
 74
                   for(int i=0; i<inputs.length(); i++){</pre>
                      String temp = state + " "+inputs.charAt(i);
                      delta.put(temp, nextState);
                  }
 78
              }else{
                   String stateInput = items[0] + " " + items[1];
 80
                   int nextState = Integer.parseInt(items[2]);
 81
                  delta.put(stateInput, nextState);
 82
              }
 83
          }
 84
 85
          public String evalString(String input){
 86
              currentState=0;
 87
              for(int i=0; i<input.length(); i++){</pre>
                  String stateInput = Integer.toString(currentState)+" "+ input.charAt(i);
 88
 89
                  if(delta.containsKev(stateInput)){
                      currentState = (int)delta.get(stateInput);
                  }else{
                      //if a deadend was reached return false
 93
                       //this allows partial language definitions
 94
                      return "Reject";
 95
                  }
 96
              }
 97
              if(final_states.contains(currentState)){
                  return "Accept";
              } else {
                  return "Reject";
101
102
          }
103
104
          public void dump(){
105
              System.out.println("(1) number of states: "+num_states);
              System.out.print("(2) final states: ");
              for(int i=0; i<final_states.size(); i++){</pre>
                  System.out.print(final_states.get(i)+ " ");
              System.out.print("\n(3) alphabet: ");
110
              alphabet.forEach((k,v)->System.out.print(k+", "));
              System.out.println("\n(4) transitions");
              delta.forEach((k,v)->System.out.println("(" + k + " " + v+")"));
114
          }
      }
```