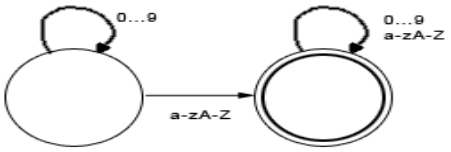
Matthew Kelly
Dr. George Rudolph
CSIS618
11 February 2009
Homework 3

Question 3.1

Integer as a right-regular grammar:
Integer → (0|…|9) Integer | (0|…|9)
Float as a right-regular grammar:
Float →*Integer*.*Integer*
Integer → (0|…|9) *Integer* | (0|…|9)

Question 3.3



Question 3.5

This cannot be done, as memory is needed is order to assure that the number of a's preceding the b's is the same in quantity as the number of b's. $\{a^n b^n\}$, however, is definable with a DFSA, as it has no such quantity matching requirement.

Question 3.8

Test 1 - tests whether carriage return is accepted as equivalent to \n and not an escaped string.

```
int main(){
  char acter = '\n';
}
```

Test 2 - tests for operator validity of Clite (*. is a valid operator in OCaml).

```
int main(){
  int eger1 = 9;
  int eger2 = 2;
  int eger3 = eger1 *. eger2;
}
```

Test3 - This test case tests for keyword restriction (char is listed as a keyword).

```
int main(){
 char char = 'c';
}
```

Test 4 - This test case tests identifier name restrictions (e.g. "o^o" is not 'a letter followed by a sequence of zero more more letters and digits' per the spec).

```
int main(){
```

```
 char o^o = ‘c’;
}
```

Test 5 – This test case tests identifier name restrictions (like test 4) but assures that a number cannot lead in the name of an identifier.

```
int main(){
 char 4cter = ‘c’;
}
```

Test 6 - This test case tests whether a floating point can be defined without a digit preceding the decimal character.

```
int main(){
 float intPointNumber = .2;
}
```

Test 7 - This test case tests whether a floating point can be defined without a digit following the decimal character.

```
int main(){
 float intPointNumber = 2.;
}
```

Test 8 – This test case tests whether curly braces can be used as equation separators (as opposed to just limiting scope of blocks of code.

```
int main(){
  int eger = 3 + {4 * 7};
}
```

Test 9 - This test case tests the robustness of the “any character” restriction.

```
int main(){
  char acter = ‘ﬃ’;
}
```

Question 3.15a

```
tokens = []
foreach i in inputChars{
   if(inputChars[i].isADigit()){tokens.push(inputChars[i])}
   else {throw invalidCharException}
}
return tokens
```

Question 3.15b

```
/* Assuming space delimiter */
tokens = []
tokenState = “”
```

```
pastDecimal = false
foreach i in inputChars{
  if(!(inputChars[i].isADigit()) && inputChars[i]!=".”){throw invalidCharException }
  if(tokenState == “”){ /* new token */
    if(inputChars[i] == “.”) {throw inappropriateDecimalException}
    elseif(inputChars[i].isADigit()) then { tokenState = inputChars[i]}
  }
  elseif(tokenState.lastChar() = “.”){
     if(!(inputChars.isADigit()){throw invalidCharException}
     tokensState += inputChars[i]
   }
   elseif(inputChars[i] == “.”){
      tokenState += inputChars[i]
     pastDecimal = true
   }
   elseif(inputChars.isADigit()) {tokenState += inputChars[i]}
   elseif(inputChars[i] == “ ” && tokenState != “” && pastDecimal) {
      tokens.push(tokenState)
      tokenState = “”
     pastDecimal = false

   }
} /* end foreach */

return tokens
```

Question 3.15c

```
tokens = []
currentToken = “”

foreach i in inputChars{
 if(i != 0){          /*Prevent negative array index access*/
 /*Current char same as prev */
   if(  inputChars[i].isADigit() && inputChars[i-1].isADigit()) ||
     !(inputChars[i].isADigit()) && !(inputChars[i-1].isADigit()))     {currentToken += inputChars[i]}
/*Current char diff from prev */
   elseif(  inputChars[i].isADigit() && !(inputChars[i-1].isADigit())
            !(inputChars[i].isADigit()) && inputChars[i-1].isADigit()){
               tokens.push(currentToken)
               currentToken = inputChars[i]}
 }else{currentToken = inputChars[i]} /*For first char, as i-1 might throw exception */
```

return tokens

Question 3.19

Yes, I believe a language can have no reserved words in the sense that every keyword can be defined. OCaml does this, even allowing the “+” operator to be redefined to different functionality. Doing so, however, could potentially break a language’s construct, e.g. overriding the class keyword might make defining a class impossible.