Travis Alpers
CSCI468
Quiz 1

Q1. As we know, the scanner module of a compiler is responsible for isolating the tokens of an input program and providing them along with various information about them to the parser.

Q1 1. (5) What is a token in the context of a programming language?

A token is a string to be replaced with another string during scanning and it defines lexeme types.

Q1 2.(5) List the items of information that must accompany a token when the parser requests a token from the scanner. Explain what each item of information is and its purpose.

Token – This allows the parser to work it's way through the LL(1) grammar.

Lexeme – This allows the parser to do necessary inserts/lookups on the symbol table.

Row – This allows the parser to print out user-friendly errors(not strictly necessary for compilation)

Col – This allows the parser to print out user-friendly errors(not strictly necessary for compilation)

Q2.Consider the process of designing and implementing a scanner for a programming language. We need a list of tokens and their definitions in order to construct a correctly functioning scanner.

Q2 1. (5) The definitions of the tokens are given as regular expressions. What is the purpose of using regular expressions as opposed to English prose to describe the tokens?

A regular expression can be translated into a Deterministic Finite Automaton which can be implemented as code to recognize specific strings.

Q2 2. (5) How do regular expressions help in the process of designing and implementing a scanner?

Regular expressions provide an easy blueprint for recognizing different token types and are naturally grouped based on their leading character, this provides much of the direction for implementing a scanner.

Travis Alpers
CSCI468
Quiz 1

Q3. Suppose that a new token named currency_literalis to be added to a programming language. This token is to stand for all strings that are currency values in dollars.

Valid examples of currency values include:

$1 $1.00 $0 $0.00 $0.05 $75 $75.44

Invalid examples include:

$01 $00 $1.5 $.05 $1.758

In words, a string that matches the currency pattern:

- starts with “$”
- which must be followed by a digit
- if that digit is 0
- else if that digit is not 0
  - the string must end, or
  - the string must be followed by a “.” and exactly two more digits
  - that digit must be followed only by zero or more digits and end when something other than a digit is encountered, OR
  - that digit must be followed by only by zero or more digits that are in turn followed by a “.” which in turn is followed by exactly two digits

As you answer the following sub parts of Q2 you may assume that you have the following auxiliary regular expressions to use: digit:= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 pos_digit:= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Q3 1. (10) Give a regular expression for currency. Use the digitand or pos_digit auxiliary expressions as appropriate to shorten the regular expression

currency_literal. Use the regular expression notation given below.

● concatenation is just two regular expressions listed in sequence with a space between, for example letter digit,which stands for the regular expression that matches a single letter followed by a single digit.

● alternation (OR) is two regular expressions listed in sequence with a | between, such as letter | digit, which stands for the regular expression that matches either a single letter or a single digit.

● "0 or more occurrences" of a particular regular expression is represented by enclosing that regular expression in curly braces, such letter {letter | digit}, which stands for the regular expression that matches strings formed as a single letter followed by 0 or more letters and digits in any combination.

● parentheses are used to enforce the evaluation order of operations in regular expressions where necessary, for example letter (letter | digit) {letter | digit} is a regular expression that matches strings that start with a letter that is followed by one or more letters and digits in any order. On the other hand, if the parentheses are omitted to give letter letter | digit {letter | digit}this is the regular expression that matches strings that consist of exactly two letters along with those that start with a digit and have any number of following letters and digits.

No other notation can be used.

currency_literal (0|(pos_digit(|{digit})))(|.digit digit)

Q3 2.(10) Give a deterministic finite state automaton (fsa) that recognizes currency literal values within the context of a program. Assume as specified in class and for your project that

● the dispatcher of the scanner calls this fsa when a $ is encountered and

● the $ is where the file pointer is pointing when the fsa starts processing the token.

Then:

● Add commentary to your FSA at appropriate points to make it a useful design for

practical scanning (i.e., that will help in implementing the FSA in a program), including

○ actions to take when certain states are reached so that the line and column

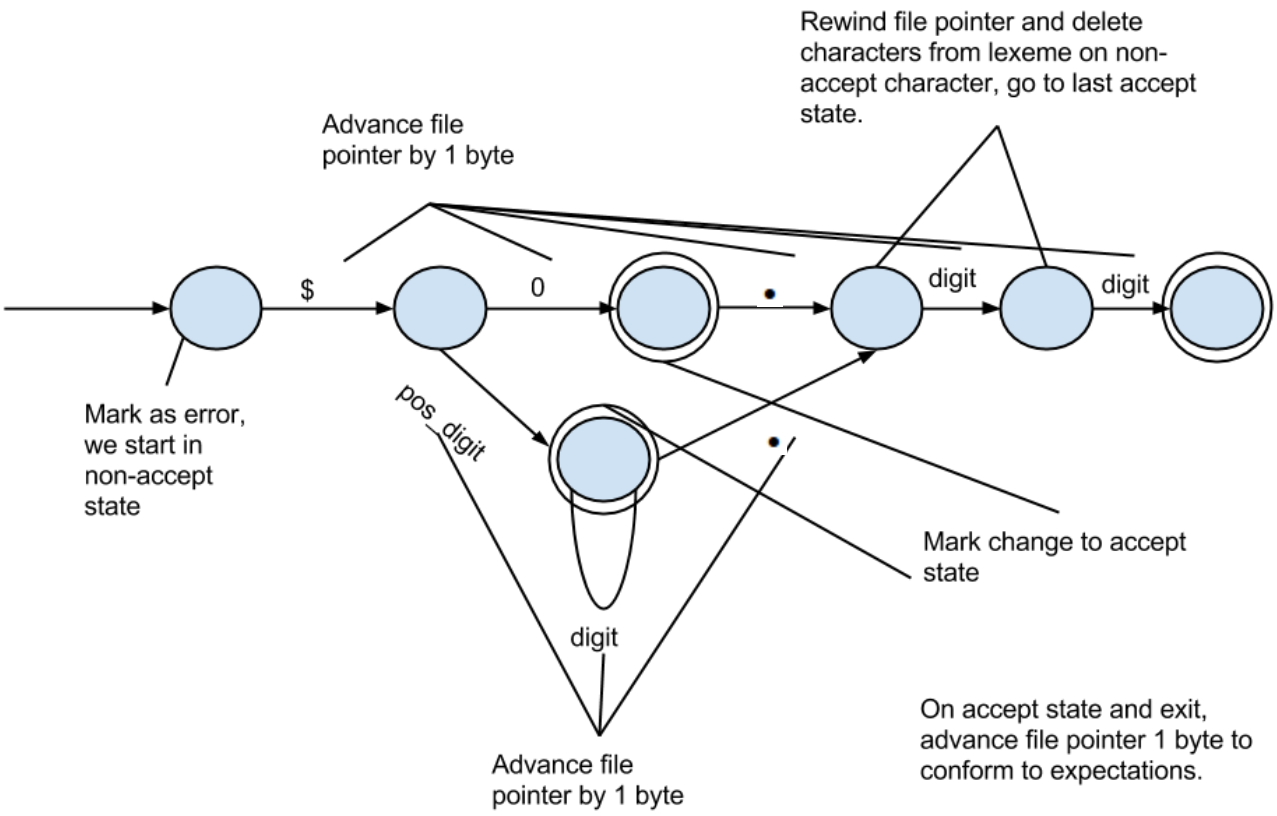
numbers are correctly registered, and the lexeme matched is properly

recorded.

○ ensuring that the file pointer is pointing to the first character after the lexeme

that is matched by the FSA

○ noting when if and when a scan error has occurred.



Travis Alpers
CSCI468
Quiz 1

Q4. Consider a scanner designed the way described in class.

Q4 1. (5) What is its big O time complexity given a program P as input?

Big O notation of the scanner discussed in class should be O(n).

Q4 2. (5) Explain what n in your stated time complexity represents.

This time complexity represents that we must only cycle through our code one time per token, with n being the number of tokens.

Q4 3. (5) Defend your stated time complexity.

Due to the way in which we have grouped and combined FSA's, each run through our parser will result in a token(MP_ERROR or otherwise) and advance the file pointer at least 1 byte until EOF.