TDT4205 Compiler Construction
Assignment 1

1 Scanning Despacito

You are a creative writer trying to write a parody for the song Despacito (). You are writing a verse where each of the ending words have to rhyme with the word Despacito.

Sample ending words are *Mosquito Burrito DeVito Dorito Cheeto Finito*. Since you have been writing some verses endlessly on a computer while being heavily sleep deprived you notice that you have written a lot of them in one particular text file. Rather than read through all of them to select suitable candidates for the parody song, you decide to use a a state machine to scan through the letters of the last word to see if you have any suitable candidates.

1.1 Drawing the DFA

For selecting very good candidates you decide to select the following endings/suffixes in each last word to ensure the quality of rhyming.

- 'eto' as in Cheetos or Magneto
- 'ento' as in momento or Magento
- 'rito' as in burrito
- 'vito' as in Danny DeVito
- 'quito' as in Mosquito
- 'cito' as in canoncito
- 'nito' as in incognito

Since you were sleepy while writing all this, you notice that in some cases, the vowels are repeated several times in the suffixes like burriiiiiiitooooo (Positive Kleene Closure for ending vowels in the suffixes).

Draw the state machine/DFA that can find out whether the input string satisfies the above mentioned requirements.

1.2 The regular expression

Write the regular expression that performs the function as in the above DFA. A Python file is given where you have to write the expression. The python code will output the number of rhyming words per verse in the file "rhymes_per_verse.txt" if the right regular expression is written by using the regular expression to evaluate the last word of the verses in the textfile "parody_verse.txt".

2 Alien Invasion

After you are done with the previous exercise suddenly an alien appears infront of you and claims that you have accidentally discovered a secret of the universe unconsciously while you were writing all that (sleep deprived) . The alien provides you with a compiler that takes the output from the python script and draws equilateral triangles according to the following commands:

Table 1: Commands and Actions

Command	*Action*
One	Triangle with tip upward
Two	Triangle with tip downward
Three	Move cursor by a length of a triangle's side
Four	Move vertically down to another line
Stop	End the program

Unfortunately the toy compiler has a problem. It's not capable of recognizing the words even though the operations for each command are implemented as one() two() three() and four() (the stop command is just to be recognized). What is required of you from this task is to implement an automaton capable of recognizing the above commands from a text stream and call the appropriate function. The four commands must be case-insensitive and characters not part of the command should be ignored (like whitespace and other additional text).

2.1 DFA

Draw the deterministic finite automation to recognize the commands and ignore other input until a word has been recognized

2.2 Toy Compiler

The provided archive *aliens.tgz* contains the code to translate the specified langauge to ghostscript commands except that it does not have a scanner. Implement the DFA from the previous exercise in scanner.c to complete the program. Here pencil.c will output the ghostscript commands. To obtain the final image one can issue a command to take in the input from aliens.txt generated from the previous exercise and then put the output of the executable into a postscript file and then ghostscript can be used to open this file. Ex:./pencil <aliens.txt >mystery.ps ; ghostscript mystery.ps