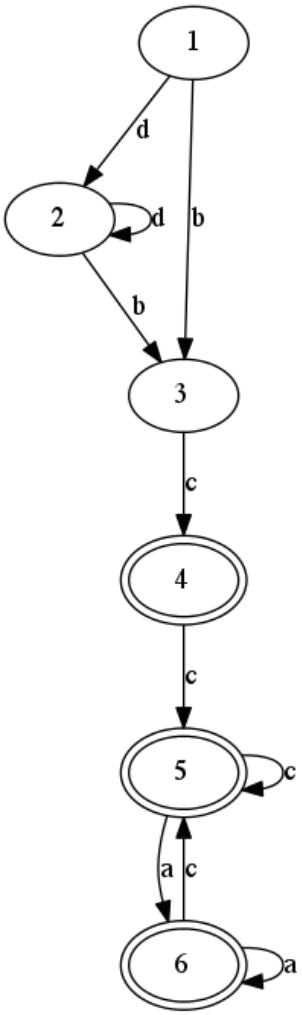
CS 655: Analyzing Sequences
Homework 1

Part 1: FSA to Regular Expression

1. /[ab]+cdcd(cd)?e+fg?/
2. /([abj]c|[dev]f)[0-9]+/

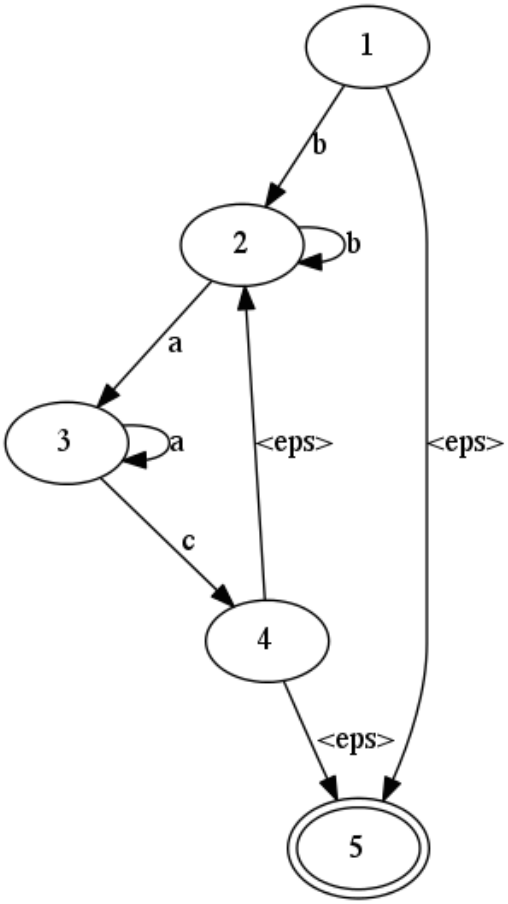
Part 2: Regular Expression to FSA

1. d*bc(ca*)*




```
digraph{
        1 -> 2[label="d"];
        2 -> 2[label="d"];
        2 -> 3[label="b"];
        1 -> 3[label="b"];
        3 -> 4[label="c"];
        4 -> 5[label="c"];
        5 -> 5[label="c"];
        5 -> 6[label="a"];
        6 -> 5[label="c"];
        6 -> 6[label="a"];
        4 [peripheries=2];
        5 [peripheries=2];
        6 [peripheries=2];
    }
```

2. (b+a+c)*




```
digraph{
        1 -> 2[label="b"];
        2 -> 2[label="b"];
        2 -> 3[label="a"];
        3 -> 3[label="a"];
        3 -> 4[label="c"];
        4 -> 5[label="<eps>"];
        4 -> 2[label="<eps>"];
        1 -> 5[label="<eps>"];
        5 [peripheries=2];
    }
```

Part 3: Counting Sonnets

There are 154 sonnets in the file. Each sonnet in the file is numbered using roman numerals. The line containing the roman numerals does not have any other character in the line other than whitespaces. I used a regular expression (`'^\s*[A-Z]+\s*$'`) to identify the roman numerals and kept a count of their occurrences, which is the same as the number of sonnets in the file.

Another interesting formatting style in the file is that, last two lines of every sonnet have 4 whitespaces followed by words with first letter in the first word being capital or starting with single quote. Using the regular expression (`'^\s{4}[A-Z\'].*'`), the count would have been 154, if sonnet XXV also followed the same formatting style. As the last two lines in that sonnet do not have 4 leading spaces, a count of 153 is obtained.

Part 4: Collecting and Counting Surnames

I used a regular expression to find the words Mr., Mrs., Miss and Lady (`'[ML][r*ia*][s*d*\.?]s*\.*y?'`). The word that follows them is considered as a surname. All punctuation marks at the end of the word are removed. Other patters such as /'s/ as in Bennets's and /--.*/ as in Darcy--and or Lizzy--if or Darcy:--but are removed by matching with the regular expression (`'[\'-][s-].*'`).

Prefix at the end of line:
In case the pattern is found at the end of the line, the first word in the next line is considered as the surname. If this condition was not included Miss Grantley and Miss Webbs were missed out, as these names appear only once in the text and are interrupted by line breaks.

A total of 41 surnames were obtained. Out of which 5 are first names (Anne, Eliza, Elizabeth, Jane, Lydia), but are included in surnames as they are addressed as Miss Anne etc. There is the letter F in the result as there is a mention of Mrs. F. in the text. Bennets and Lucases are included in the list as the sisters are mentioned as Miss Bennets and Miss Lucases.

A bar plot showing frequency of the surnames found in Pride and Prejudice is shown below:

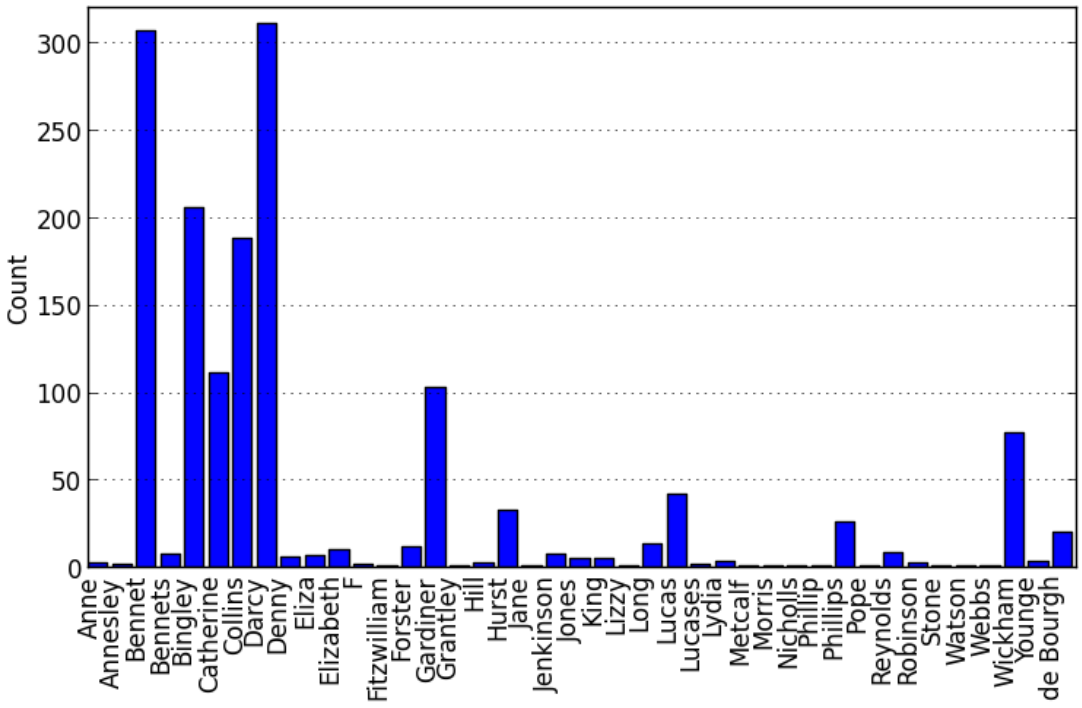


Figure 1. Histogram of surnames found in Pride and Prejudice sorted alphabetically

Frequency table of the surnames found in Pride and Prejudice sorted alphabetically:

Anne	3
Annesley	2
Bennet	307
Bennets	8
Bingley	206
Catherine	111
Collins	188
Darcy	311
Denny	6
Eliza	7
Elizabeth	10
F	2
Fitzwilliam	1
Forster	12
Gardiner	103
Grantley	1
Hill	3
Hurst	33
Jane	1
Jenkinson	8
Jones	5
King	5
Lizzy	1
Long	14
Lucas	42
Lucases	2
Lydia	4
Metcalf	1
Morris	1
Nicholls	1
Phillip	1
Phillips	26
Pope	1
Reynolds	9
Robinson	3
Stone	1
Watson	1
Webbs	1
Wickham	77
Younge	4
de Bourgh	20

First Names:

In order to find the first names, all lines containing surnames obtained previously are selected. These lines are then matched with a regular expression to see if there are two consecutive words with first letter as a capital letter. If such words are present, the first word is matched with the regular expression for (Mr., Mrs., Miss, Lady). If it is not one of those and if the word is not at the beginning of a sentence (i.e. not following a period), then this word is considered as a first name. Using this method, 11 first names were detected, out of which 9 are true first names. If the condition to omit words in the beginning of the sentence was removed, one more name Georgiana Darcy was also detected. But if that condition is removed, many starting sentences are included giving 26 results, out of which only 10 are true first names.

```
['William', 'Lucas']
['Elizabeth', 'Bennet']
['Charlotte', 'Lucas']
['London', 'Lydia']
['Anne', 'Darcy']
['George', 'Wickham']
['Maria', 'Lucas']
['Mary', 'King']
['Eliza', 'Bennet']
['But', 'Jane']
['Lydia', 'Bennet']

['Georgiana', 'Darcy']
```

Chapter-wise frequency:

The starting line of a chapter was found by using a regular expression (`'Chapter \d*'`) to match with the string stating the chapter number. 61 chapters were identified. Then the frequency of surnames in every chapter was calculated the same way as for the entire book, as mentioned in the beginning. The normalized frequency of occurrences of words among the chapters is shown in the fig. 2. To normalize, the occurrences of a surname in all chapters are divided by the maximum number of times it occurred in a single chapter. A 3D bar graph showing the frequency of occurrence of the first 8 surnames is also shown in fig. 3.

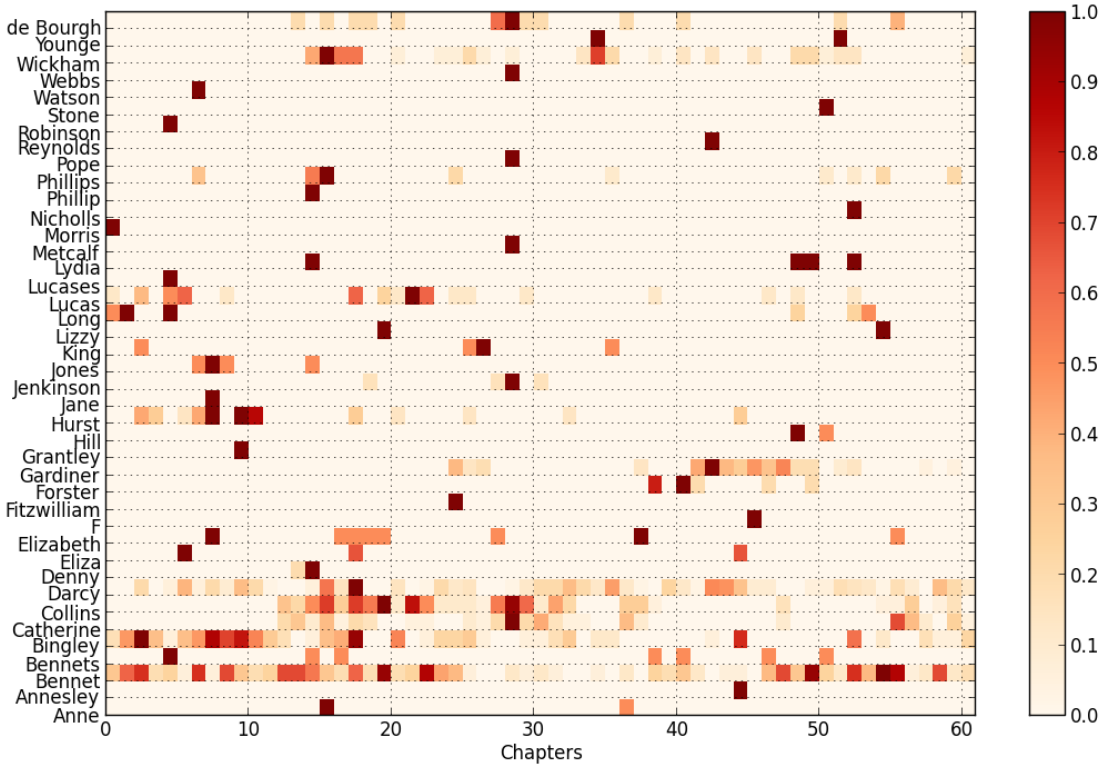


Figure 2. Chapter-wise distribution of surnames found in Pride and Prejudice

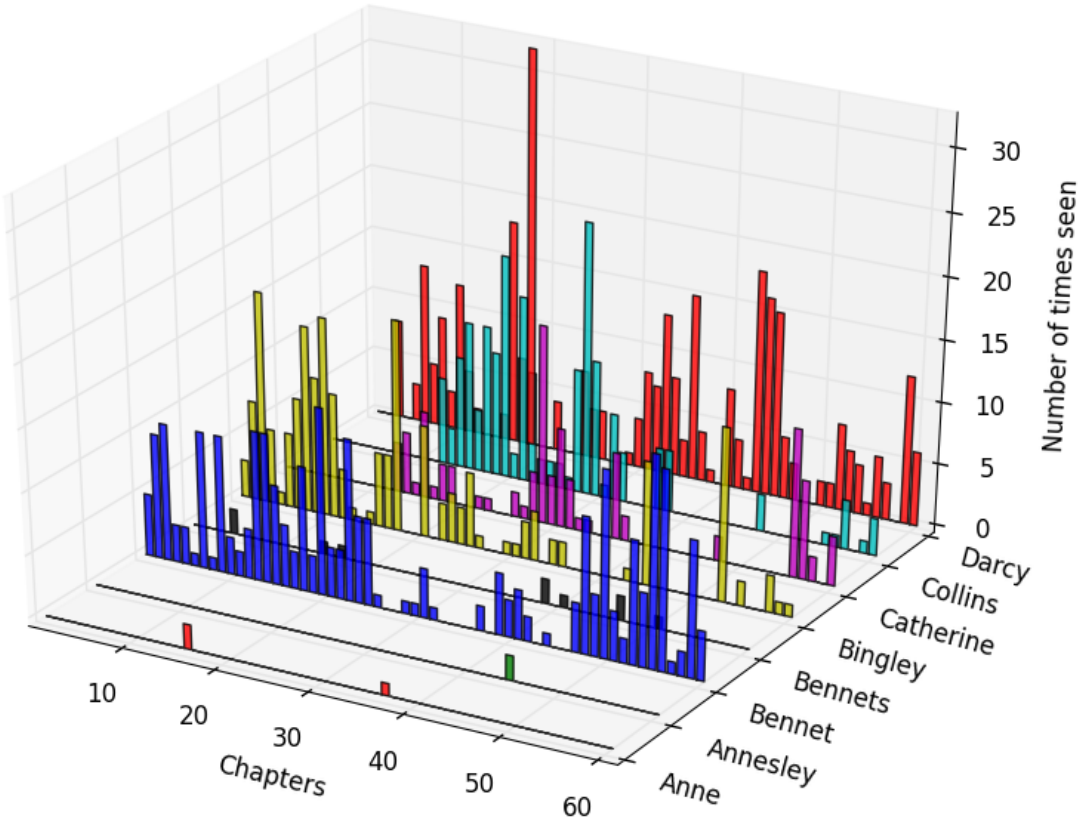


Figure 3. Chapter-wise histogram of eight surnames found in Pride and Prejudice

Part 5: Regex Golf

S.No.	Level		Regular Expression
1	Warmup	:	foo
2	Anchors	:	ick\b
3	Ranges	:	[a-f]{4}
4	Backrefs	:	(\w{3}).*\1
5	Abba	:	^(?!.*(.)(.)\2\1)
6	A man, a plan	:	(.)(.)(.)?.?\3\2\1$
7	Prime	:	^(?!(xx+)\1+$)
8	Four	:	(.)(.\1){3}
9	Order	:	^[a-m][b-o].*[^ed]$

This was the part of the assignment I first jumped to doing. First few levels were easy. Then it took a while to find a pattern in the words. But once the pattern was found it was easy to write the regular expression. The Prime level was frustrating as I took a long time to figure that out. Initially I was able to include all except 2,3 and 5 x cases. Till that point I was only using '*' operator, and only then I understood the importance of '+' operator. I hope I'll be able to complete all levels soon.