Problem set 1

Due Thursday, January 22, 2015

The lecture today was designed to supplement the reading, not stand on its own, especially with regard to regular expressions. For those of you who do not have the book yet, this leaves you without enough background to do problem 2.1. If you expect your book to arrive by early next week, you may want to wait before attempting problem 2.1. Otherwise, I suggest you find a basic regular expression tutorial online or in the library. The slides (now posted online) would probably be enough to see you through problem 2.4.

2.1 Write regular expressions for the following languages. You may use either Perl/Python notation or the minimal "algebraic" notation of Section 2.3, but make sure to say which one you are using. By "word", we mean an alphabetic string separated from other words by whitespace, any relevant punctuation, line breaks, and so forth.

2. the set of all lower case alphabetic strings ending in a b;

4. the set of all strings from the alphabet a, b such that each a is immediately preceded by and immediately followed by a b;

6. all strings that have both the word *grotto* and the word *raven* in them (but not, e.g., words like *grottos* that merely *contain* the word *grotto*);

2.4 Design an FSA that recognizes simple date expressions like *March 15, the 22nd of November, Christmas*. You should try to include all such "absolute" dates (e.g., not "deictic" ones relative to the current day, like *the day before yesterday*). Each edge of the graph should have a word or a set of words on it. You should use some sort of shorthand for classes of words to avoid drawing too many arcs (e.g., furniture → desk, chair, table).