Check-In C7

Review: ASTs

Show the AST for the expression

```
(1 + x) * 2
```

Announcements

Housekeeping

Review Session:

Learned 3151 is reserved Wednesday, February 1, 6-8:30pm

Zoom: https://compilers.cool/zoom

Recording: https://compilers.cool/tests

Announcements

Housekeeping

Nice job on Project 1!

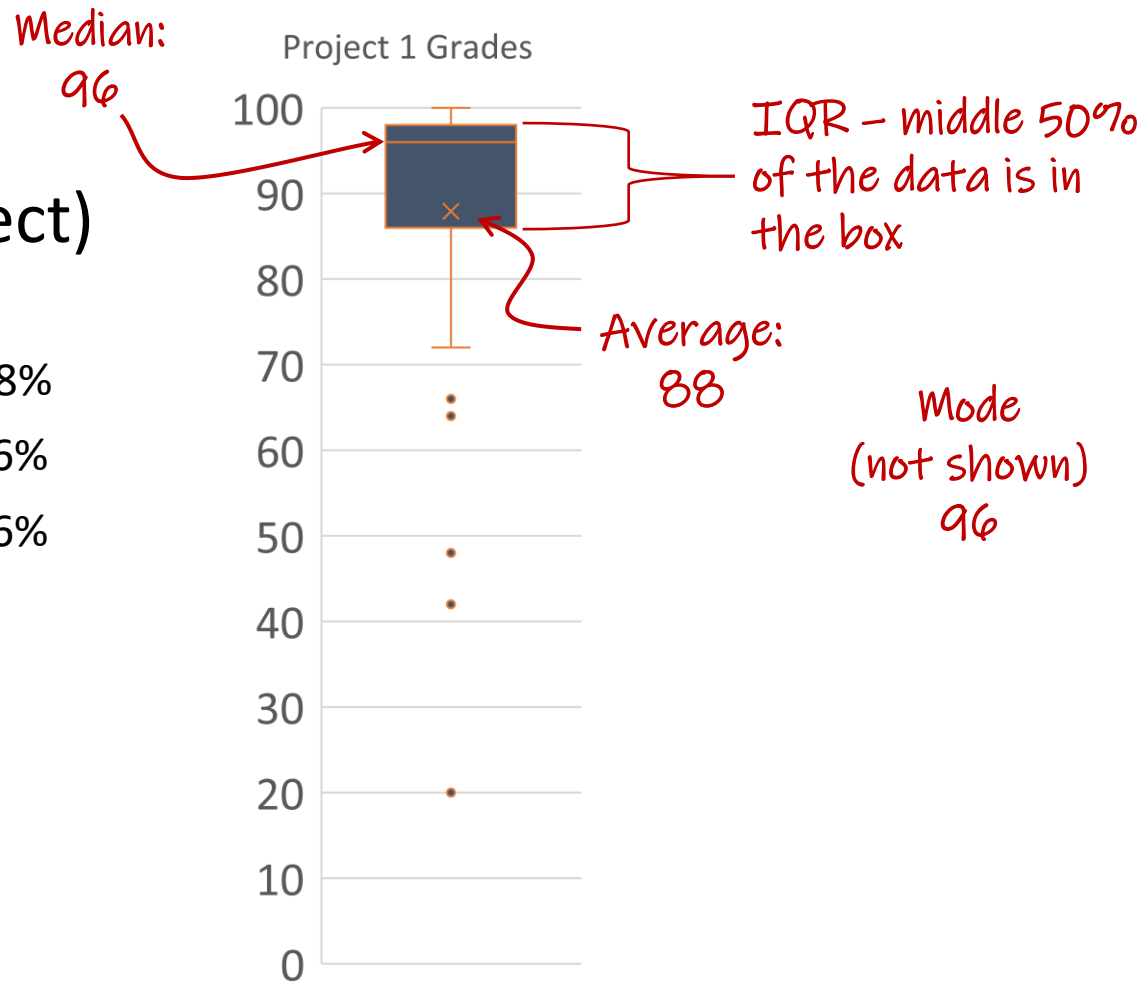
70%	A
11%	B
7%	C
7%	D
5%	F

11% (perfect)

Average grade: 88%

Median grade: 96%

Mode grade: 96%



Announcements

Housekeeping

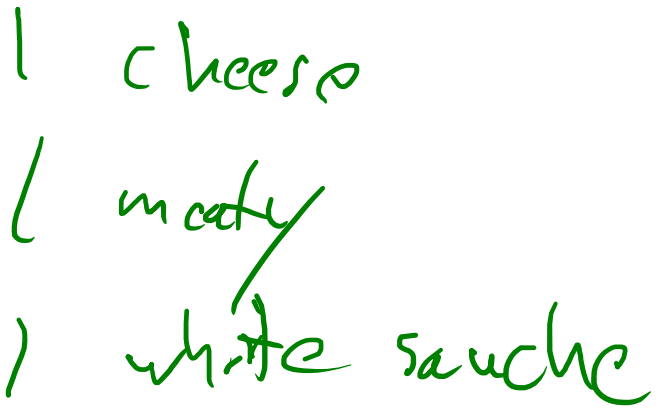
A contingency that *probably* won't come up…

F L I P P E D
W E D N E S D A Y

Written Work #2

Topics:

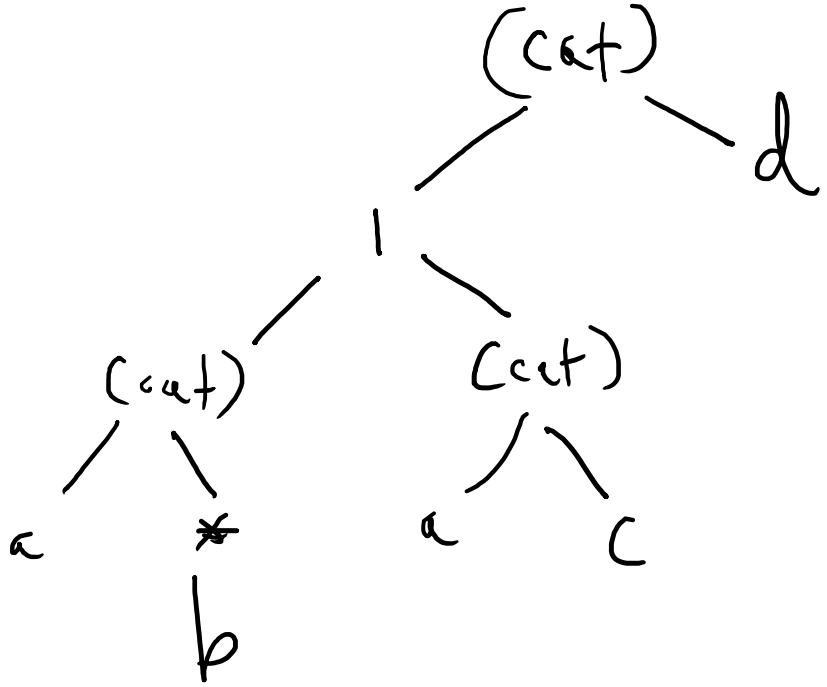
- Scanning
- Defining Syntax
- Ambiguous Syntax



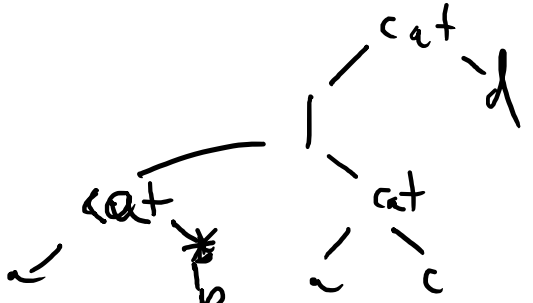
Written Work #2: Question 1

Draw out the expression tree representation of the following regular expression

(ab*|ac)d

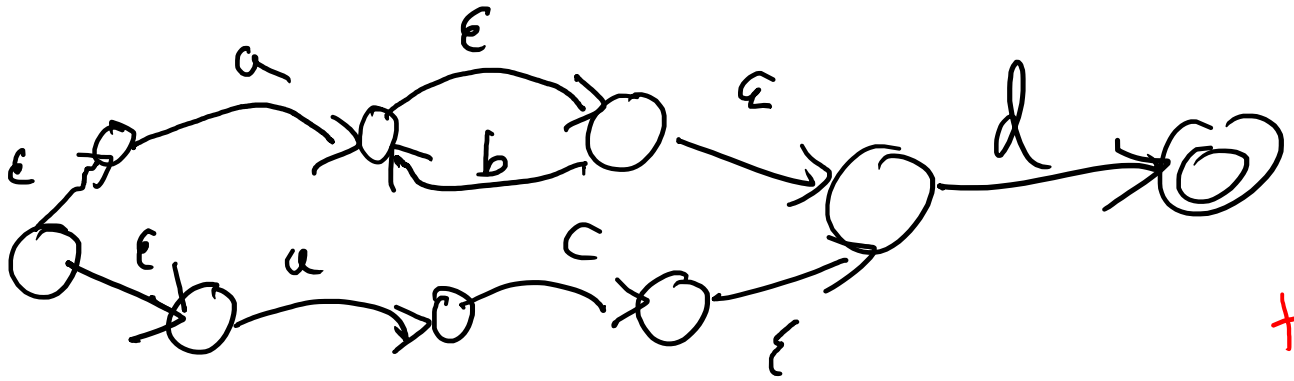


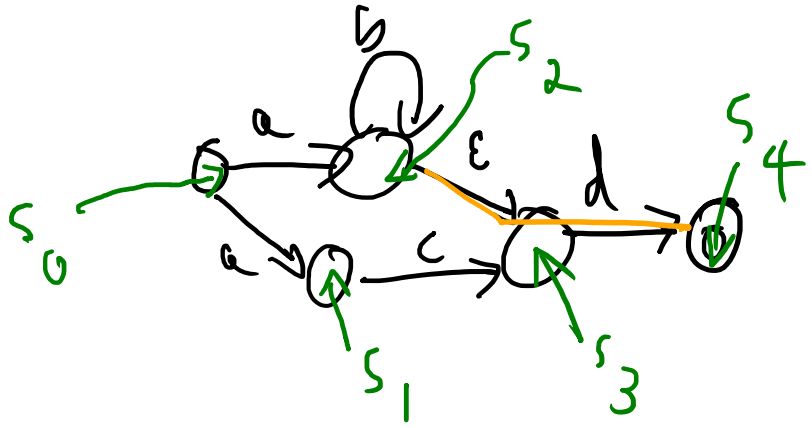
Written Work #2: Question 2

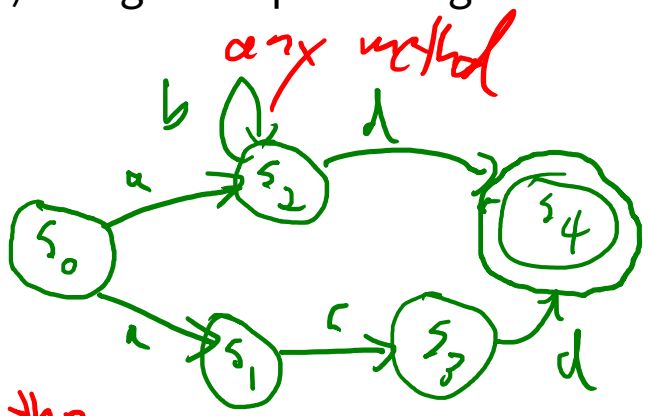


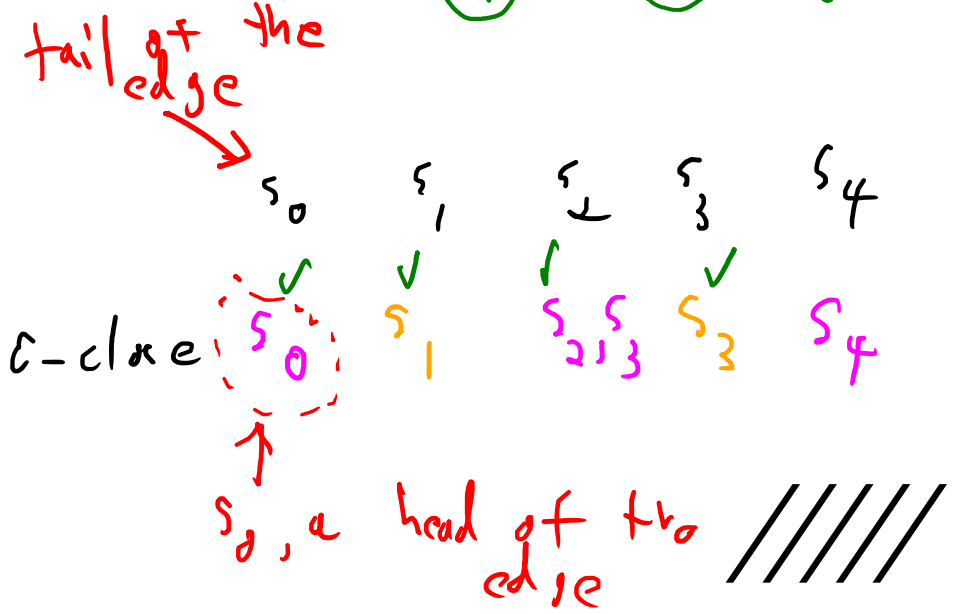
Convert the regular expression from above into an ε-NFA (i.e. an NFA with ε-edges) using ~~Thompson's algorithm~~

(ab*|ac)d





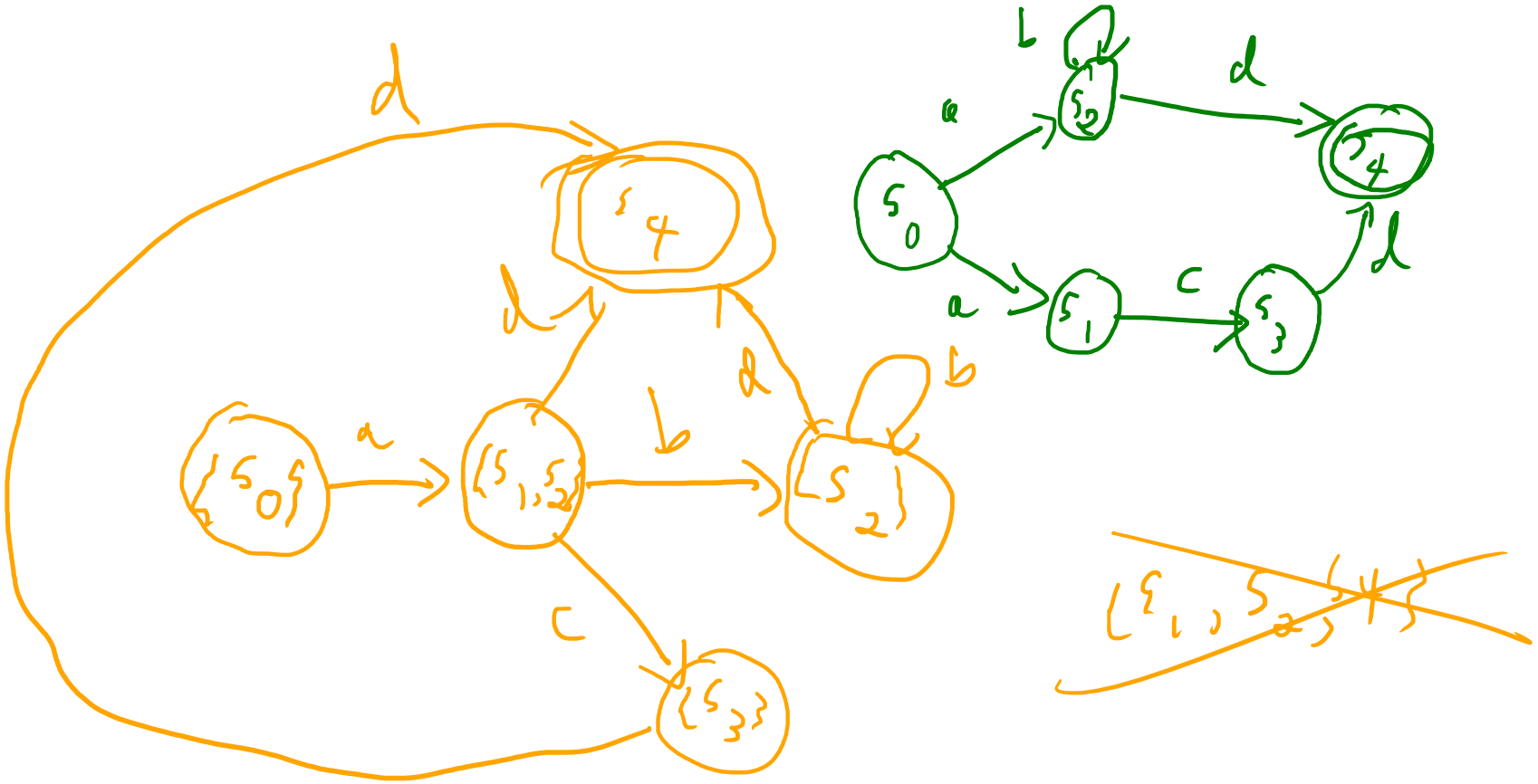




Written Work #2: Question 3

Use the ε-elimination technique to remove ε-edges from the previous ε-NFA

(ab*|ac)d



Written Work #2: Question 4

Let *DotList* be a language such that:

- The empty string is in the language
- The single terminal **dot** is in the language
- Sequences of more than 1 **dot** terminal separated by the **comma** terminal are in the language. e.g.:
  - **dot comma dot**
  - **dot comma dot comma dot**

No other strings are in the language

Write an unambiguous grammar that recognizes *DotList*