Little Languages

Lecture 29:

Midterm Practice Questions

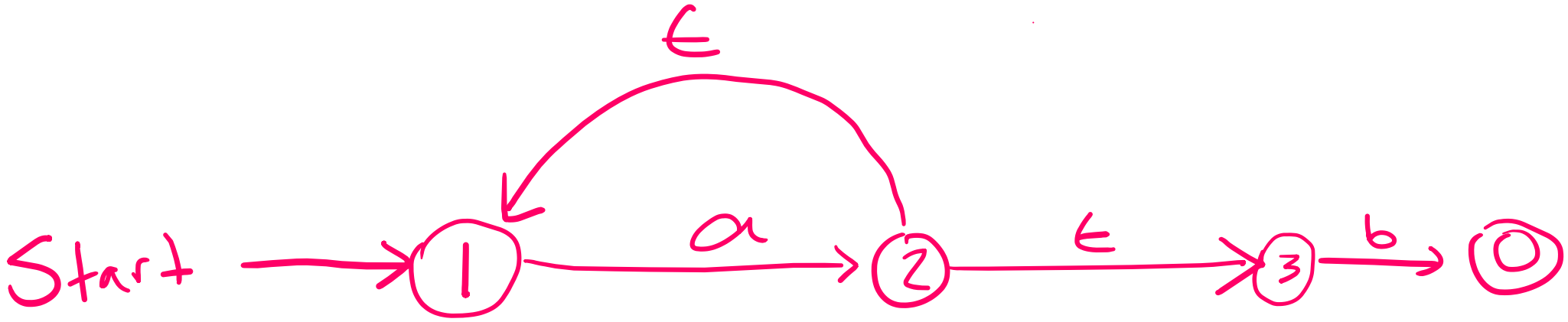
Midterm

- git
  - commit
  - staging
  - branch
  - merging
- processes & shell
  - stdin, stdout, stderr
  - redirection
  - pipelines
  - exit statuses
- files
  - general process for reading input from a file
- Tools (conceptually)
  - sed - capturing groups
  - xargs
  - make
- regular languages
  - determinist transition diagrams
  - ε-transition diagrams
  - regular expressions
  - RE to NFA conversion
- Arena Allocation
- Operator Overloading
  - Conceptually

Using the possible transition states of thegrep, draw the epsilon transition diagram for the regular expression: **a|bc|d**

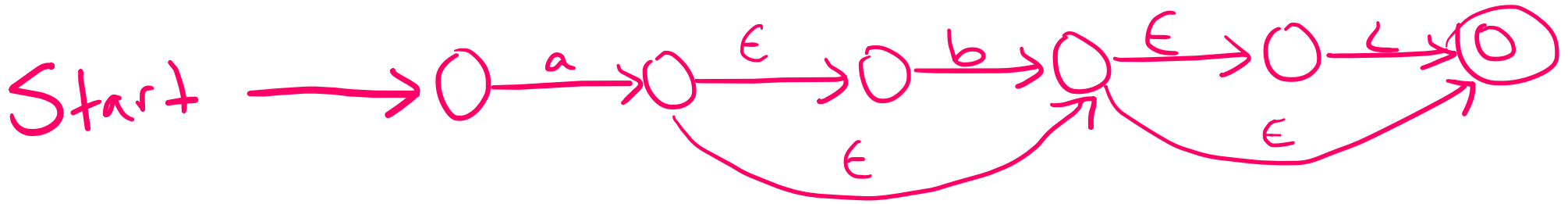
Using the possible transition states of thegrep, draw the epsilon transition diagram for the regular expression: **aa***

Transition Diagram Question



1. Give an input string of at least three characters that is accepted by this transition diagram.
2. Give an input string of at least three characters that is not accepted by this transition diagram but whose first two characters are valid. You must use the characters a and b in your string.

Transition Diagram Question



1. What is the shortest input string accepted by this automaton?
2. What is the longest input string accepted by this automaton?
3. How many different input strings are accepted by this automaton?

Discuss: Come up with an example of piping and redirection and discuss each in terms of what is happening with process(es) and their standard input/output/error.

Describe what is happening in this pipeline with as much detail as possible:

```
$ thecho "1 + 2" | thbc | thdc
```

Discuss: Describe git merging in terms of commits. What's scenarios lead to a fast-forward merge versus a merge commit? What's the difference?

Discuss:

Each State variant in `thegrep` had a fixed number of edges.

1. How would you model a State variant with an arbitrary numbers of out edges?

    Specifically, what would its field(s) be for edges?

2. Suppose you were trying to model a directed acyclic graph. Write some pseudocode to represent your strategy for detecting cycles.

Discuss: Using the arena allocation strategy as in thegrep, how could you modify the directed acyclic graph of the previous question such that your edges contained additional metadata besides simply the next node they were directed toward?