Section 1:

Lexical Analysis, Regex, & FSAs

CS 164 @ UC Berkeley, Spring 2024

http://tinyurl.com/164section1

Sora Kanosue (he/him)

Master's Student advised by Sarah Chasins

Research interests: Programming languages, and how the power of programming can be made accessible to non-technical users

Just graduated from Berkeley in May

Went to high school in New York, originally from Japan

Reminders

WA 1 will be released later tonight, and will be due on Feb 7.

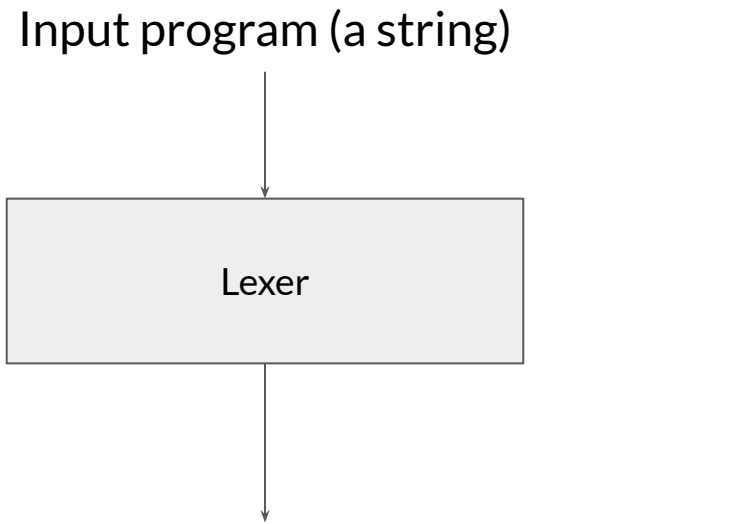
(No slip days for written assignments)

Reminder to take care of yourselves, and to prioritize your health! Feel free to submit extension requests if you need them!

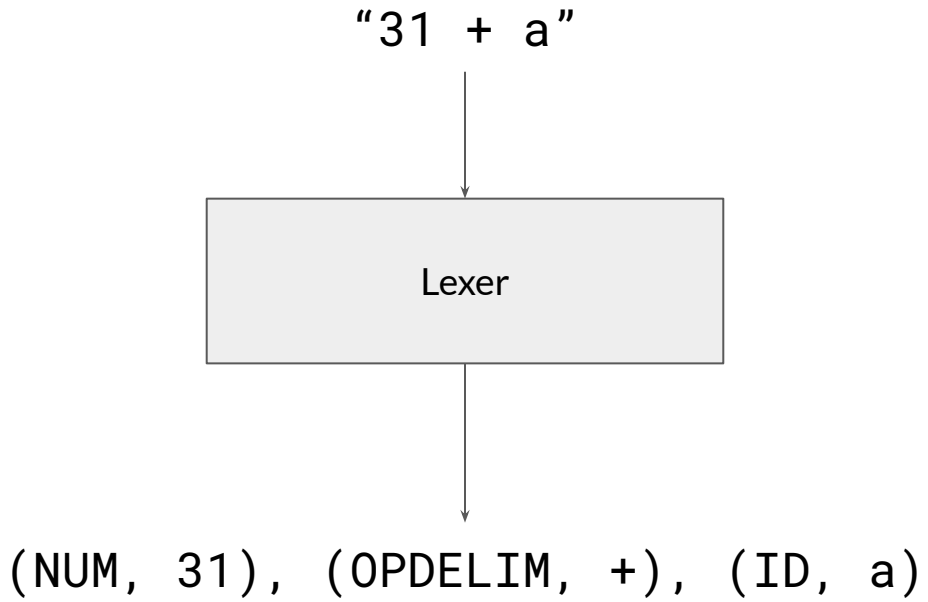
Lexing

```
for (int i=0; i<=10; i++) {
arr[i] += i;
}
```

More formally:



Lexing Example



Regex

$L(\text{‘a’}) = \{\text{‘a’}\}$

Concatenation: $L(AB) = \{ab | a \in L(A) \text{ and } b \in L(b)\}$

E.g. $L(\text{‘ab’}) = \{\text{‘ab’}\}$

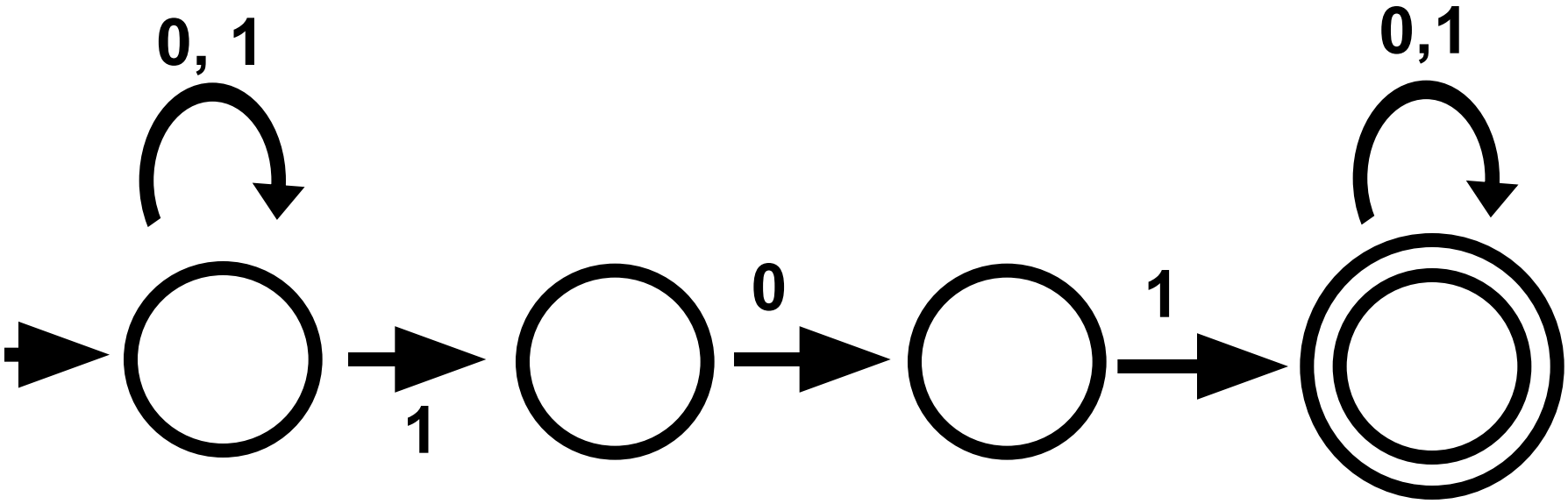
Union: $L(A|B) = L(A) \cup L(B)$

E.g. $L(\text{‘a’}|\text{’b’}) = \{\text{‘a’}, \text{’b’}\}$

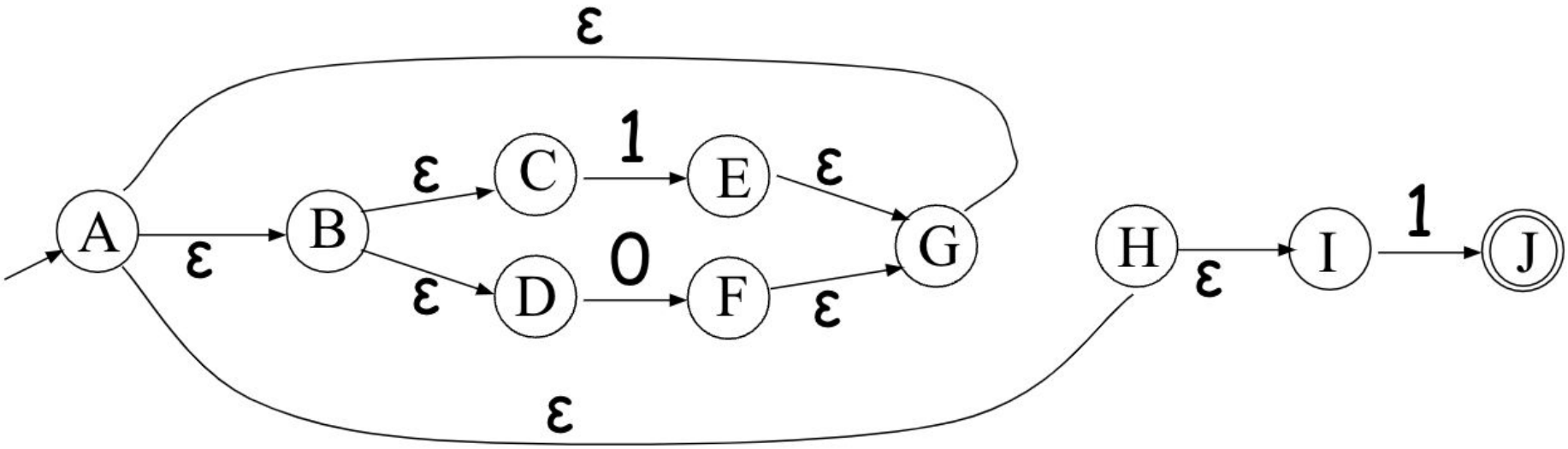
Iteration/Kleene closure: $L(A^*) = \{\text{‘’}\} \cup L(A) \cup L(AA) \cup L(AAA) \cup$ …

E.g. $L(\text{‘a’}^*) = \{\text{‘’}, \text{‘a’}, \text{‘aa’}, \text{‘aaa’}, \ldots\}$

Deterministic Finite Automata



Non-deterministic Finite Automata



Anonymous feedback form:
http://tinyurl.com/SoraDisFeedback