Regular Expression
Finite State Machine
Pattern Matching

02/10/2023

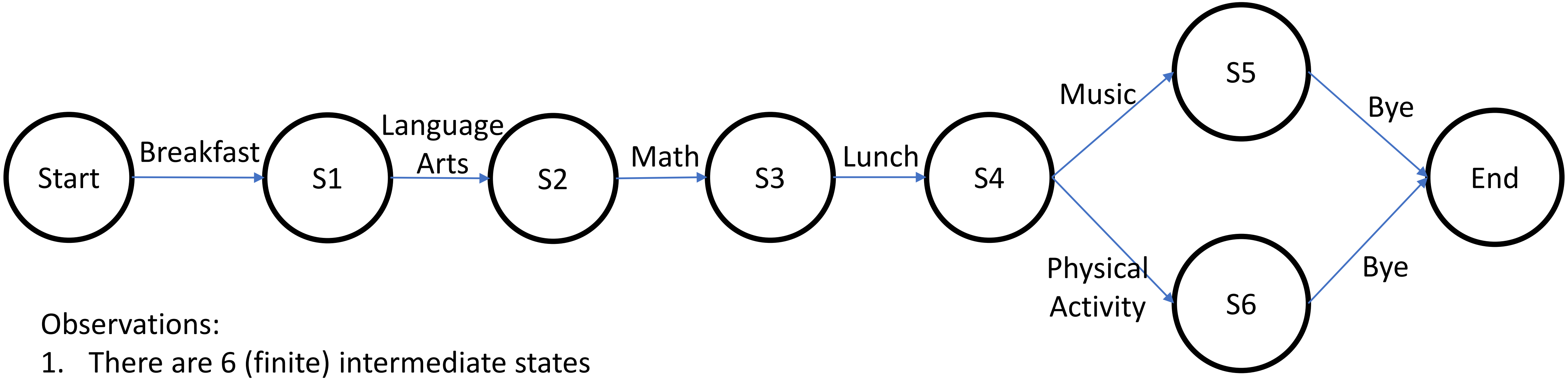
DSCI 510 Lab Session

Yuzhong Huang

Regular Expression

- Regular is a concept in language theory, which is out of scope.
- Each regular expression corresponding to a finite state machine
- Essentially, it’s doing pattern matching

Schedule in an Elementary School



Observations:

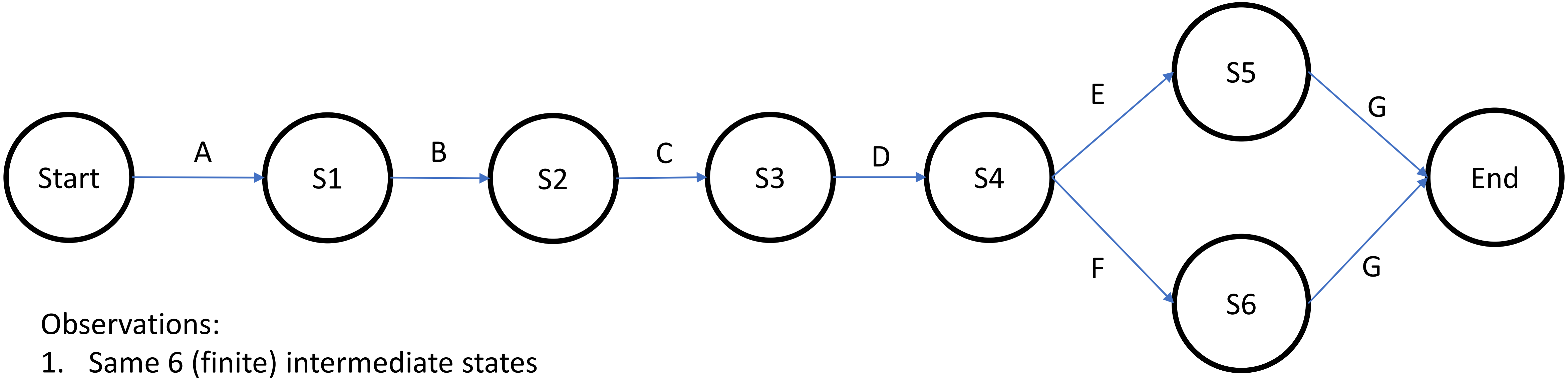
1. There are 6 (finite) intermediate states
2. There are some rules between states
3. Transition are directional, and it may have multiple transition from a state

Finite State Machine is a useful and straightforward tool to describe some pattern
It could be represented with:

- start state, end state, finite intermediate states
- Possible transitions between states, and requirements for transition

Then we can say whether a specific day match this pattern

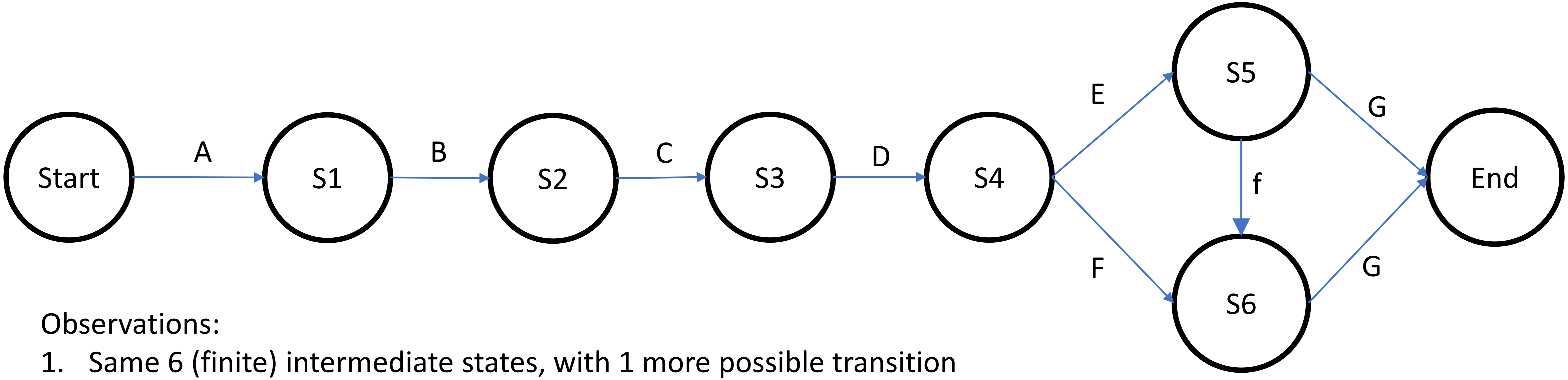
FSM for string



Observations:

1. Same 6 (finite) intermediate states
2. Match string:
    1. ABCDEG
    2. ABCDFG

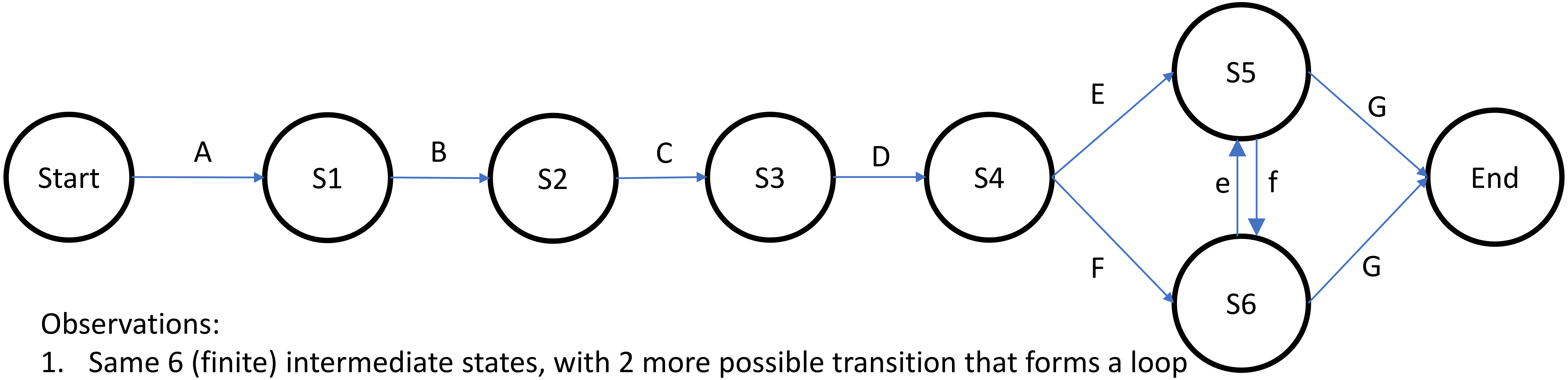
FSM for string 2



Observations:

1. Same 6 (finite) intermediate states, with 1 more possible transition
2. Match string:
    1. ABCDEG
    2. ABCDFG
    3. ABCDEfG

FSM for string 3



Observations:

1. Same 6 (finite) intermediate states, with 2 more possible transition that forms a loop
2. Match string
    1. ABCDEG
    2. ABCDFG
    3. ABCDEfG
    4. ABCDFeG
    5. ABCDEfeG
    6. ABCDEfefG
    7. ABCDEfefeG
    8. ……

More examples

- https://youtu.be/qGfPe2g8VOs?t=50

FAQ

1. What’s the relationship between regular expression and Python?
    - Regular expression is a small foreign language embedded in Python/C/C++/Java/JavaScript

2. Regular expression is a language on its own
    - Sequential execution
    - Loop execution
    - Branch execution

- 3. Why regular expression?
    - Simple split can’t handle complex states
    - Regex is more efficient (in terms of execution speed) to parse large document

- 4. It’s easier to write regex then read it.
  - Write is a process from easy-to-read diagram to compressed notation
  - Read is the inverse process, much harder
  - Try to learn regex by write, not read
  - If get lost reading a regex, try to make transition diagram from it.

- 5. How to debug a regex
  - Use online regex debugger, like regex101

How to debug general Python code?

1. Write your Python program incrementally.
2. Add a print statement where you are not sure.
3. Add try/except around where it get wrong and pdb.set_trace()
4. Set a breakpoint in VS Code.