Regular Expressions

jiaao Ho
CST 62 Tsinghua Univ.

What is reg expr?

- A language to describe a string
- A way to do matching

Elements of reg expr

- '.' — any character
- '\w' — letters, numbers, underline
- '\s' — white space (including \t, \n, \r)
- '\d' — digits
- Capital letters of above chars — complement

Elements of reg expr

- '[]' — set of chars
  - '-' — range of chars
- '^' — beginning of the string
- '$' — ending of the string

Repeating suffixes

- '?' — appear once or not appear
- '+' — appear no less than once
- '*' — repeat as many times including zero

Specified repeating suffixes

- '{a}' — repeat exactly a times
- '{a,b}' — repeat between a and b times
- '{a,}' — repeat a or more times

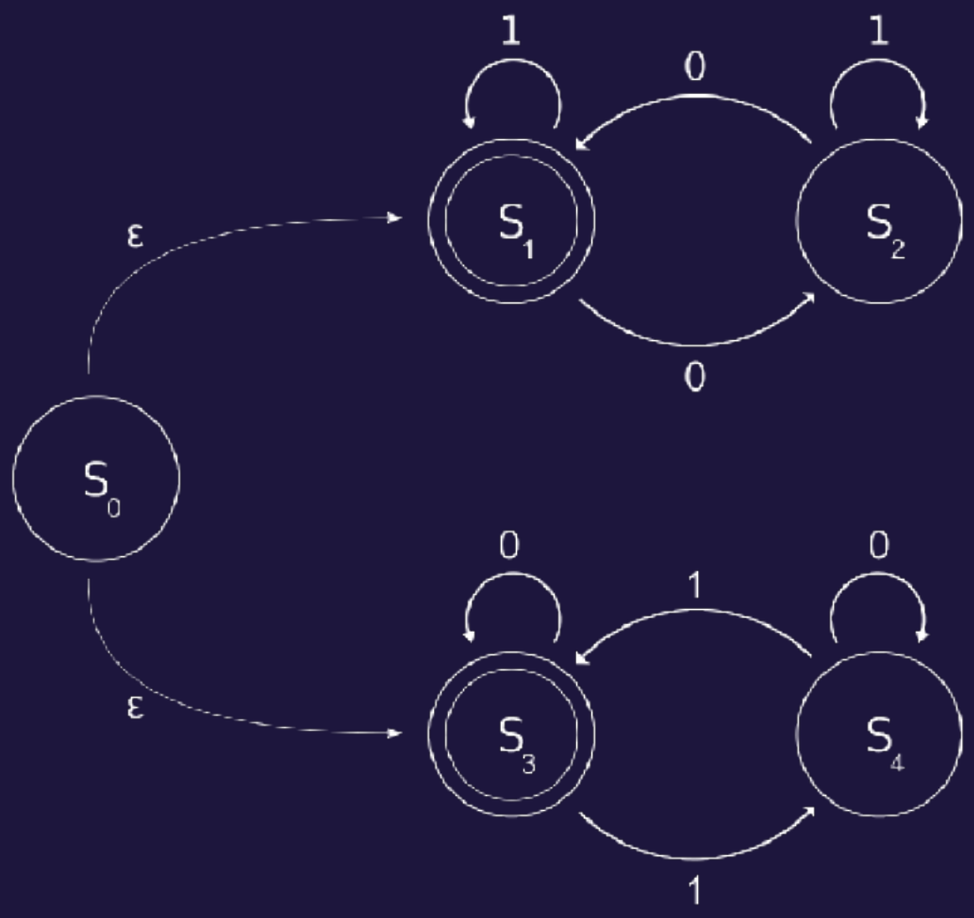
Combinations

- 'AB' — concat A and B
- 'A|B' — A or B
- '()' — calculating priority definations

And more excluded today …

Examples

- **Email address** — /^\w[a-zA-Z_.-]*\w@\w[a-zA-Z_.]*\w$/
- **Telephone** — /^(\+\d{1,3})?\d{8,11}$/
- **IP Addr** — /^((2[0-4]\d|25[0-5]|[01]\d\d|\d\d?)\.){3}((2[0-4]\d|25[0-5]|[01]\d\d|\d\d?))$/



Algorithm behind — NFA

A nondeterministic finite automaton

Usage

- **grep** —
  ```
  echo “hello world” | grep “\w{2}”
  ```

- **JavaScript** —
  ```
  (‘some string’).match(/\w{3}$/)
  ```

- **Python** —
  ```
  import re;
  re.search(‘expr’, ‘str’);
  ```

```cpp
#include <iostream>
#include <regex>

using namespace std;

int main() {
    if (regex_match("subject",
                regex("(sub)(.*)"))) {
        cout << "Match!\n";
    }
}
```

C++

<regex> standard library

References

- Nondeterministic finite automaton
  https://en.wikipedia.org/wiki/Nondeterministic_finite_automaton

- GNU Grep
  https://www.gnu.org/software/grep/manual/grep.html

- Deer Chao zhengze http://www.jb51.net/tools/zhengze.html