Regex
from the ground up

403410033 資工四 曾俊宏

Specification

- 目前：
  - 使用 NFA
  - 支援 ? * | + . (match any)
  - 支援 escape sequences, e.g. \(, etc.
  - 支援 unicode
  - ~~用 DFA, 支援 character class~~
    - ~~夢想遠大，能力不足QAQ~~

Implementation

- Infix to postfix
- Postfix to NFA
- Matching!

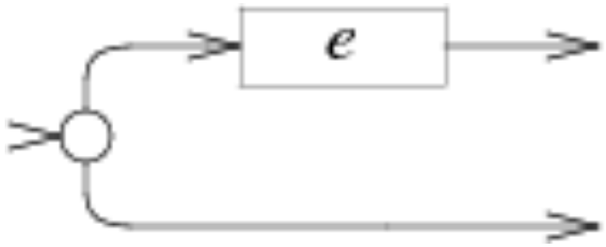
Infix to Postfix

- 對於 + ? * | ( ) . \ 做特殊處理
  - 遇到 \ 就看下一個，如果也是 operator 就 escape，不然就補回 \
  - 過程中會一直兩兩間補上 concat 的符號 (帶個特殊 control value)
    - 遇到 ( ): 現有狀態上 stack
    - +?* 遇到直接輸出
- 一般的 rune 直接輸出
  - rune 是 golang 裡的類似 C char 的資料型態
  - unicode!

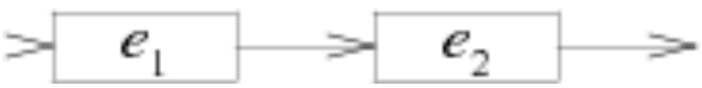
Postfix to DFA

- 分成 6 種 cases 來建立
  - Exact match 的 operator . 用一個特殊 control code 紀錄，之後當一般的字母建立 NFA state

a

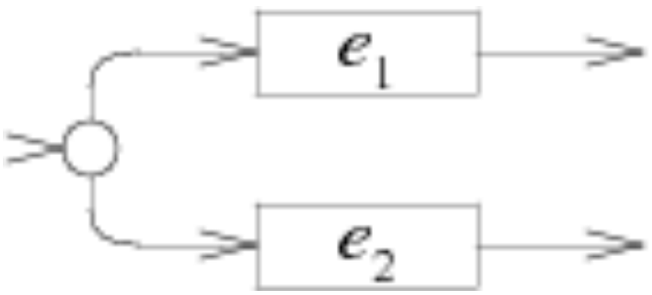


e?

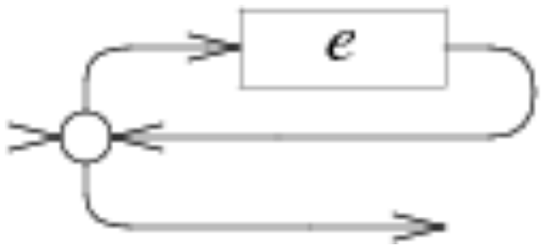


e_1e_2

e+



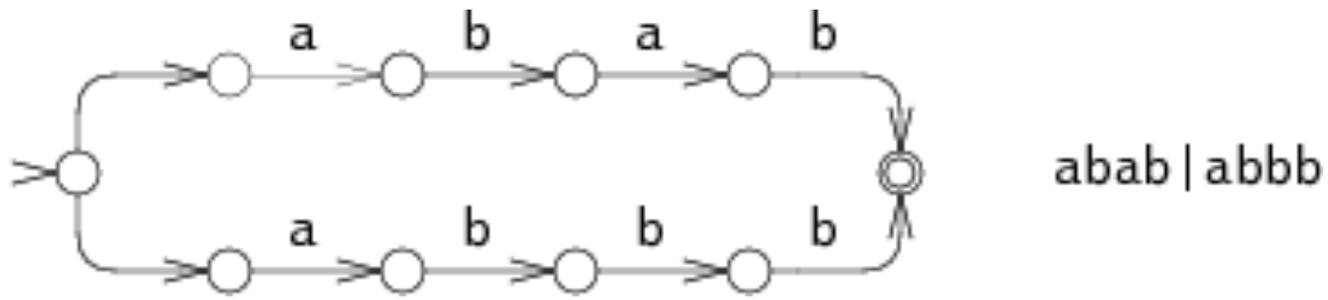
$e_1 | e_2$



e*

Matching

- 對於起點
  - 記錄透過 epsilon 邊可以到達的狀態
- 對於 input string 的每個 rune
  - 走 NFA



- input string 邊走邊看有無現有狀態可以到達 final state
  - 有的話更新 index (longest match)
    - index == strlen(input string): exact match
    - otherwise, partial match

Correctness

- regexgen (string to regex)

Example

The simplest use is to simply pass an array of strings to `regexgen`:

```
const regexgen = require('regexgen');

regexgen(['foobar', 'foobaz', 'foozap', 'fooza']); // => /foo(?:zap?|ba[rz])/
```

- exrex (regex to string)

```
>>> list(exrex.generate('((hai){2}|world!)'))
['haihai', 'world!']

>>> exrex.getone('\d{4}-\d{4}-\d{4}-[0-9]{4}')
'3096-7886-2834-5671'
```

Demo (?)

- Issue: 我沒有實作 range operator，所以實用價值有點低
  - Match floating point numbers
    - ./nfa -pat "(-?(0|1|2|3|4|5|6|7|8|9)+)(\.(0|1|2|3|4|5|6|7|8|9)+)?" -str "-0.12465" 2>/dev/null
  - Match url
    - ./nfa -pat "(q|w|e|r|t|y|u|i|o|p|a|s|d|f|g|h|j|k|l|z|x|c|v|b|n|m)+://.*\.htm" -str "https://deerchao.net/tutorials/regex/common.htm" 2>/dev/null

- 但是之前的 a?a?…aaa 去配對 aaaaaa 的話速度很可以
    - time ./nfa -pat "a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?a?aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" -str "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" 2>/dev/null

- Unicode :)
  - time ./nfa -pat "別噢 (沒事)*" -str "別噢 沒事沒事" 2>/dev/null
  - time ./nfa -pat "哇哈* (A|a)mos loves (to say)?喂*.?" -str "哇哈哈 Amos loves 喂喂喂喂." 2>/dev/null

- Escape
    - ./nfa -pat '\a?\' -str '\\' 2>/dev/null
- Shows partial match (longest match)
    - time ./nfa -pat 'aabb' -str 'aabbbb' 2>/dev/null