I. Testcase 목록

1) (ABA)*(BB)*
   - 입력: ABAABABBB
   - 정답: false
2) (A + B)*BA
   - 입력: AABBBAB
   - 정답: false
3) ((BA)* + (AB)*)
   - 입력: ABABA
   - 정답: false
4) (AB + B)*
   - 입력: BBBBAB
   - 정답: true
5) (B + ε)(AAA)*B
   - 입력: BAAAAAAB
   - 정답: true
6) (Bε)*(εA)*
   - 입력: BBBAA
   - 정답: true
7) ((ABε)* + (BB)*)*
   - 입력: ABABBB
   - 정답: true
8) ((εA)* + B)
   - 입력: AB
   - 정답: false
9) (AA + εBAB)*
   - 입력:
   - 정답: true
10) (B + ε)*(ε + A)*ε
    - 입력: BAB
    - 정답: false
11) (B*E + (AB)*(BA)*)
    - 입력: ABABBABAA
    - 정답: false
12) (B*E + (AB)*(BA)*)
    - 입력: ABABBABAAB
    - 정답: false
13) (B*E + (AB)*(BA)*)
    - 입력: ABABBABA
    - 정답: true
14) (AA(AB)* + (((ε + Bε)(ε + Bε))* + (BA)*))(A*B + B*A)
    - 입력: AABAAAAAAAAAAA
    - 정답: false

15) (AA(AB)* + (((ε + Bε)(ε + Bε))* + (BA)*))(A*B + B*A)
   - 입력: BBAAAAAAAAAABA
   - 정답: false
16) (AA(AB)* + (((ε + Bε)(ε + Bε))* + (BA)*))(A*B + B*A)
   - 입력: BBAAAAAAAAAAB
   - 정답: true
17) (AA(AB)* + (((ε + Bε)(ε + Bε))* + (BA)*))(A*B + B*A)
   - 입력: AAABAAAAAAAAAAAB
   - 정답: true
18) (ABAAAB + AB((Aε)* + (BΕ)*)(AB)*)
   - 입력: ABAAAB
   - 정답: true
19) (ABAAAB + AB((Aε)* + (BΕ)*)(AB)*)
   - 입력: ABAAABAB
   - 정답: true
20) (ABAAAB + AB((Aε)* + (BΕ)*)(AB)*)
   - 입력: ABAAABA
   - 정답: false
21) (AΕ(A + ε) + (B + ε)AAA)*
   - 입력: AAABAA
   - 정답: false
22) (AΕ(A + ε) + (B + ε)AAA)*
   - 입력: AAAAAA
   - 정답: true
23) (AΕ(A + ε) + (B + ε)AAA)*
   - 입력: AAAAAAA
   - 정답: false
24) (AΕ(A + ε) + (B + ε)AAA)*
   - 입력: BAAABAAA
   - 정답: true
25) (AΕ(A + ε) + (B + ε)AAA)*
   - 입력: BAAAAAAA
   - 정답: false

II. 별첨

```
let testcases : (Regex.t * alphabet list) list =
  [
      (* 1 *)   (CONCAT (STAR (CONCAT (Alpha A, CONCAT (Alpha B, Alpha A))), STAR (CONCAT (Alpha B, Alpha
B))), [A;B;A;A;B;A;B;B;B]);
      (* 2 *)   (CONCAT (STAR (OR (Alpha A, Alpha B)), CONCAT (Alpha B, Alpha A)), [A;A;B;B;B;A;B]);
      (* 3 *)   (OR (STAR (CONCAT (Alpha B, Alpha A)), STAR (CONCAT (Alpha A, Alpha B))), [A;B;A;B;A]);
      (* 4 *)   (STAR (OR (CONCAT (Alpha A, Alpha B), Alpha B)), [B;B;B;B;A;B]);
      (* 5 *)   (CONCAT (OR (Alpha B, Epsilon), CONCAT (STAR (CONCAT (Alpha A, CONCAT (Alpha A, Alpha
A))), Alpha B)), [B;A;A;A;A;A;A;B]);
      (* 6 *)   (CONCAT (STAR (CONCAT (Alpha B, Epsilon)), STAR (CONCAT (Epsilon, Alpha A))), [B;B;B;A;A]);
      (* 7 *)   (STAR (OR (STAR (CONCAT (Alpha A, CONCAT (Alpha B, Epsilon))), STAR (CONCAT (Alpha B, Alpha
B)))), [A;B;A;B;B;B]);
      (* 8 *)   (OR (STAR (CONCAT (Epsilon, Alpha A)), Alpha B), [A;B]);
      (* 9 *)   (STAR (OR (CONCAT (Alpha A, Alpha A), CONCAT (CONCAT (Epsilon, Alpha B), CONCAT (Alpha
A, Alpha B)))), []);
      (* 10 *)   (CONCAT (CONCAT (STAR (OR (Alpha B, Epsilon)), STAR (OR (Epsilon, Alpha A))), Epsilon),
[B;A;B]);
      (* 11 *)   (OR (CONCAT (STAR (Alpha B), Empty), CONCAT (STAR (CONCAT (Alpha A, Alpha B)), STAR
(CONCAT (Alpha B, Alpha A)))), [A;B;A;B;B;A;B;A;A]);
      (* 12 *)   (OR (CONCAT (STAR (Alpha B), Empty), CONCAT (STAR (CONCAT (Alpha A, Alpha B)), STAR
(CONCAT (Alpha B, Alpha A)))), [A;B;A;B;B;A;B;A;A;B]);
      (* 13 *)   (OR (CONCAT (STAR (Alpha B), Empty), CONCAT (STAR (CONCAT (Alpha A, Alpha B)), STAR
(CONCAT (Alpha B, Alpha A)))), [A;B;A;B;B;A;B;A]);
      (* 14 *)   (CONCAT (OR (CONCAT (CONCAT (Alpha A, Alpha A), STAR (CONCAT (Alpha A, Alpha B))), OR
(STAR (CONCAT (OR (Epsilon, CONCAT (Alpha B, Epsilon)), OR(Epsilon, CONCAT (Alpha B, Epsilon)))), STAR
(CONCAT (Alpha B, Alpha A)))), OR (CONCAT (STAR (Alpha A), Alpha B), CONCAT (STAR (Alpha B), Alpha A))),
[A;A;B;A;A;A;A;A;A;A;A;A;A;A]);
      (* 15 *)   (CONCAT (OR (CONCAT (CONCAT (Alpha A, Alpha A), STAR (CONCAT (Alpha A, Alpha B))), OR
(STAR (CONCAT (OR (Epsilon, CONCAT (Alpha B, Epsilon)), OR(Epsilon, CONCAT (Alpha B, Epsilon)))), STAR
(CONCAT (Alpha B, Alpha A)))), OR (CONCAT (STAR (Alpha A), Alpha B), CONCAT (STAR (Alpha B), Alpha A))),
[B;B;A;A;A;A;A;A;A;A;A;B;A]);
      (* 16 *)   (CONCAT (OR (CONCAT (CONCAT (Alpha A, Alpha A), STAR (CONCAT (Alpha A, Alpha B))), OR
(STAR (CONCAT (OR (Epsilon, CONCAT (Alpha B, Epsilon)), OR(Epsilon, CONCAT (Alpha B, Epsilon)))), STAR
(CONCAT (Alpha B, Alpha A)))), OR (CONCAT (STAR (Alpha A), Alpha B), CONCAT (STAR (Alpha B), Alpha A))),
[B;B;A;A;A;A;A;A;A;A;A;B]);
      (* 17 *)   (CONCAT (OR (CONCAT (CONCAT (Alpha A, Alpha A), STAR (CONCAT (Alpha A, Alpha B))), OR
(STAR (CONCAT (OR (Epsilon, CONCAT (Alpha B, Epsilon)), OR(Epsilon, CONCAT (Alpha B, Epsilon)))), STAR
(CONCAT (Alpha B, Alpha A)))), OR (CONCAT (STAR (Alpha A), Alpha B), CONCAT (STAR (Alpha B), Alpha A))),
[A;A;A;B;A;A;A;A;A;A;A;A;A;A;B]);
      (* 18 *)   (OR (CONCAT (CONCAT (Alpha A, CONCAT (Alpha B, Alpha A)), CONCAT (CONCAT (Alpha A,
Alpha A), Alpha B)), CONCAT (CONCAT (CONCAT (Alpha A, Alpha B), OR (STAR (CONCAT (Alpha A, Epsilon)),
STAR (CONCAT (Alpha B, Empty)))), STAR (CONCAT (Alpha A, Alpha B)))), [A;B;A;A;A;B]);
      (* 19 *)   (OR (CONCAT (CONCAT (Alpha A, CONCAT (Alpha B, Alpha A)), CONCAT (CONCAT (Alpha A,
Alpha A), Alpha B)), CONCAT (CONCAT (CONCAT (Alpha A, Alpha B), OR (STAR (CONCAT (Alpha A, Epsilon)),
STAR (CONCAT (Alpha B, Empty)))), STAR (CONCAT (Alpha A, Alpha B)))), [A;B;A;A;A;B;A;B]);
      (* 20 *)   (OR (CONCAT (CONCAT (Alpha A, CONCAT (Alpha B, Alpha A)), CONCAT (CONCAT (Alpha A,
Alpha A), Alpha B)), CONCAT (CONCAT (CONCAT (Alpha A, Alpha B), OR (STAR (CONCAT (Alpha A, Epsilon)),
STAR (CONCAT (Alpha B, Empty)))), STAR (CONCAT (Alpha A, Alpha B)))), [A;B;A;A;A;B;A]);
      (* 21 *)   (STAR (OR (CONCAT (Alpha A, CONCAT (Empty, OR (Alpha A, Epsilon))), CONCAT (OR (Alpha B,
Epsilon), CONCAT (Alpha A, CONCAT (Alpha A, Alpha A))))), [A;A;A;B;A;A;]);
      (* 22 *)   (STAR (OR (CONCAT (Alpha A, CONCAT (Empty, OR (Alpha A, Epsilon))), CONCAT (OR (Alpha B,
Epsilon), CONCAT (Alpha A, CONCAT (Alpha A, Alpha A))))), [A;A;A;A;A;A;]);
```

```
        (* 23 *)    (STAR (OR (CONCAT (Alpha A, CONCAT (Empty, OR (Alpha A, Epsilon))), CONCAT (OR (Alpha B,
Epsilon), CONCAT (Alpha A, CONCAT (Alpha A, Alpha A))))), [A;A;A;A;A;A;A;]);
        (* 24 *)    (STAR (OR (CONCAT (Alpha A, CONCAT (Empty, OR (Alpha A, Epsilon))), CONCAT (OR (Alpha B,
Epsilon), CONCAT (Alpha A, CONCAT (Alpha A, Alpha A))))), [B;A;A;B;A;A;A;]);
        (* 25 *)    (STAR (OR (CONCAT (Alpha A, CONCAT (Empty, OR (Alpha A, Epsilon))), CONCAT (OR (Alpha B,
Epsilon), CONCAT (Alpha A, CONCAT (Alpha A, Alpha A))))), [B;A;A;A;A;A;A;]);
    ]
```