Säännöllisten lausekkeiden tulkitseminen.

1. Lyhyesti säännöllisistä lausekkeista(RegEx).

- Säännöllisille lausekkeille on monia hyöty käyttötarkoituksia, kuten esim. syötteen oikeellisuuden tarkastaminen ja oikeanlaisten merkkijonojen etsiminen suuremmista merkkijonoista, tai muista merkkijonoja sisältävistä tietorakenteista.
- Säännöllisen lausekkeen käyttö voi usein helpottaa ohjelmointi työtä tekemällä halutun merkkijonon ominaisuuksien määrittelystä vaivattomampaa.

Esimerkki säännöllisestä lausekkeesta: a*(b|cd?)+

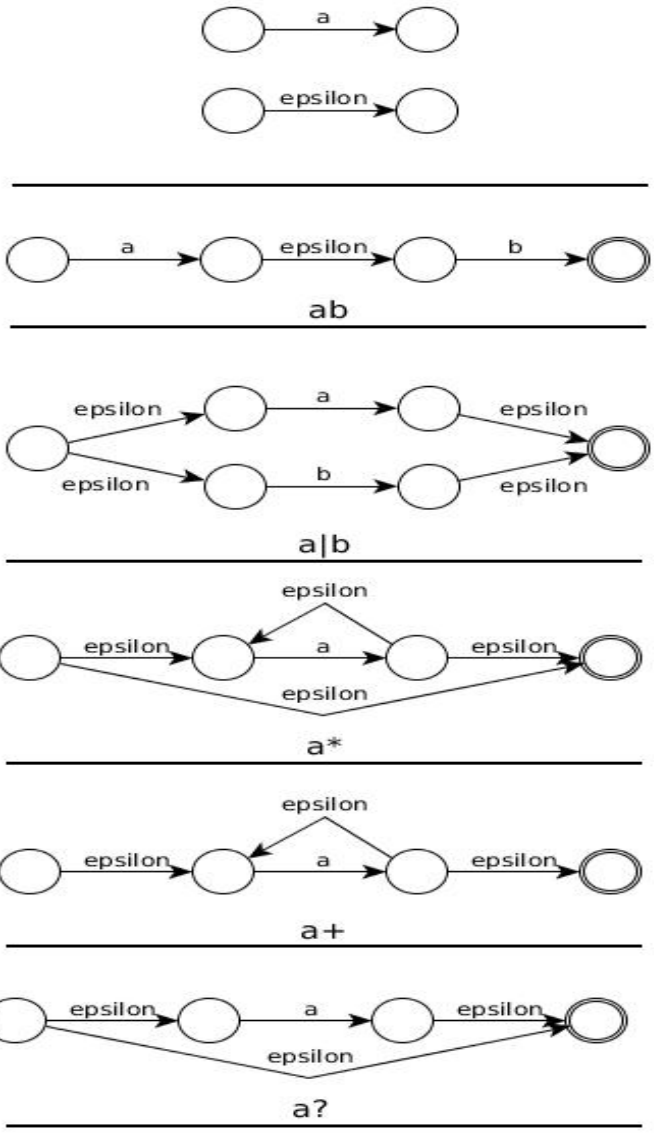
2. Säännöllisen lausekkeen muuntaminen merkkijonosta tietorakenteeksi.

* Ensin merkkijonon pohjalta rakennetaan Epädeterministinen äärellinen automaatti(NFA).

- NFA koostuu tiloista ja tilojen muutoksista, jotka kuvaavat eri syöte symboleja.

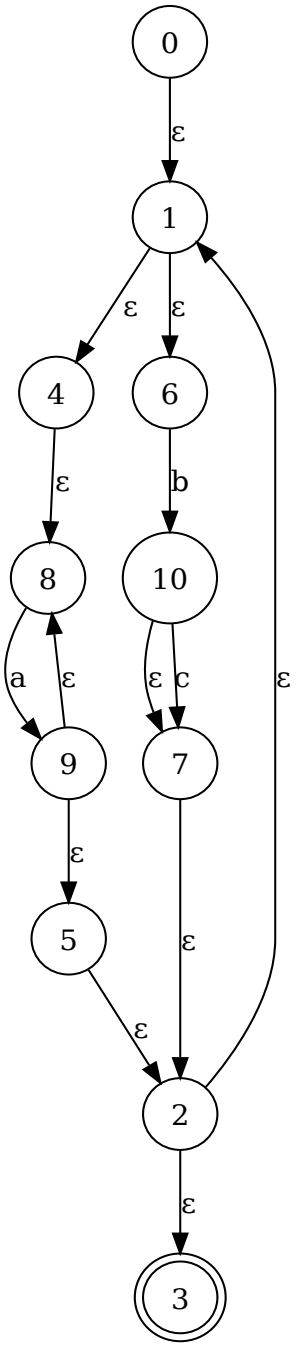
Säännöllisten lausekkeiden komponentit epädeterministisessä äärellisessä automaatissa.

Syöte symboleja kuvaavien tilanmuutosten ja epsilon tilanmuutosten avulla voidaan rakentaa kaikki säännöllisiä lausekkeita kuvaavat operaatiot.

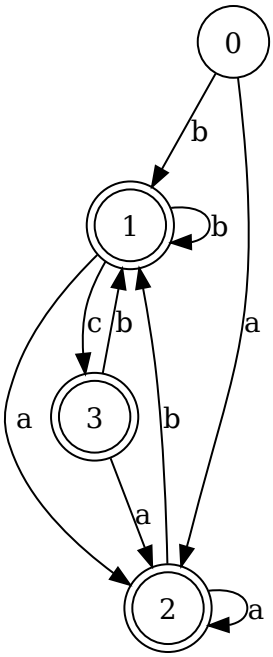


* Seuraavaksi NFA muutetaan deterministiseksi äärelliseksi automaatiksi(DFA).
    - DFA eroaa NFA:sta siten, että siinä ei ole epsilon muunnoksia.
* DFA:ta voidaan vielä optimoida poistamalla siitä turhia
*Esim RegEx: (a+|bc?)+ NFA ja DFA muodoissaan.

NFA: (a+|bc?)+



DFA: (a+|bc?)+



3. Tietorakenteen käyttö.

- Tulkille annetaan merkkijono, josta etsitään säännöllisen lausekkeen ehdot täyttävät merkkijonot ja palautetaan ne käyttäjälle.
- Etsintä toimii niin, että merkkijonoa vertaillaan optimoituun versioon DFA:sta

4. Oma toteutukseni

- Sisältää operaatiot konkatenaatiolle, ja symboleille: *, |, +, ?, \
- Käyttää myös sulkuja oikein.

- Näillä ominaisuuksilla voi perjaatteessa johtaa kaikki muutkin operaatiot ankkureita lukuunottamatta tosin se voi olla joissain tilanteissa hyvin vaivalloinen tehtävä.