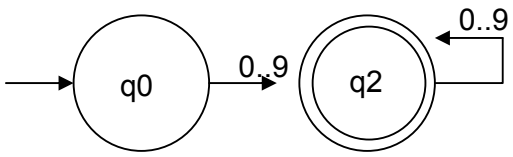
V. Ekspresi Regular (ER)

- Sebuah bahasa dinyatakan regular jika terdapat *finite state automata* yang dapat menerimanya.
- Bahasa-bahasa yang diterima oleh FSA bisa dinyatakan secara sederhana dengan ekspresi regular (regular expression).
- Ekspresi regular memberikan suatu pola (*pattern*) atau *template* untuk untai/*string* dari suatu bahasa.
- Banyak masalah pada perangkat lunak yang bisa disederhanakan dengan melakukan pengubahan notasi ekspresi regular ke dalam implementasi komputer dari FSA yang bersangkutan.
- Contoh : Finite State Automata untuk mengenal bilangan bulat /integer tidak bertanda



- Ekspresi Regularnya adalah : misal 0..9 disimbolkan sebagai digit, maka ERnya adalah : (digit)(digit)*

V.1. Notasi Ekspresi Regular

- Notasi yang digunakan untuk ER adalah :
  1. * : berarti bisa tidak muncul, bisa juga muncul berhingga kali (0-n)
  2. $^{+}$: berarti minimal muncul satu kali (1-n)
  3. + : berarti union atau bisa diganti dengan notasi U
  4. . : berarti konkatenasi, biasanya tanpa ditulis titiknya, misal ab sama dengan a.b

- Contoh ekspresi regular (ER) :

  1. ER : ab*cc

     Contoh *string* yang bisa dibangkitkan abcc, acc, abbcc, abbbcc, dst. (b bisa tidak muncul atau muncul sejumlah berhingga kali)

  2. ER : 010*

     Contoh *string* yang bisa dibangkitkan 01,010, 0100,01000, dst. (0 bisa tidak muncul atau muncul sejumlah berhingga kali)

  3. ER : $a^{+}d$

     Contoh *string* yang bisa dibangkitkan ad,aad, aaad,aaaad dst. (a minimal muncul satu kali)

  4. ER : a* U b*

     Contoh *string* yang bisa dibangkitkan a, b, aa, bb, dst.

  5. ER : 01*+0

     Contoh *string* yang bisa dibangkitkan 0, 01,011, dst.

V.2. Hubungan ER dengan FSA

- Untuk setiap ER ada satu NFA dengan transisi ε (NFA ε-move) yang ekivalen.
- Sementara untuk setiap DFA ada satu ER dari bahasa yang diterima oleh DFA.
- Hubungannya dapat digambarkan sebagai berikut :