Finite-State Automata and Regular Languages

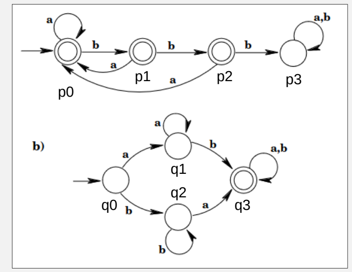
Theorem 3.5. *The intersection of two regular languages is a regular language.*

The union of two regular languages is a regular language.
The concatenation of two regular languages is a regular language.
The complement of a regular language is a regular language.
The Kleene closure of a regular language is a regular language.

- sketch of proof
  - build DFAs or NFAs accepting the languages using Theorem 3.3
  - build a DFA or NFA using those DFAs/NFAs which accepts the union/concatenation/complement/Kleene closure
    - for two DFAs M_1, M_2, M accepting $M_1 \cap M_2$ contains pairs of states from M_1 and M_2 – the idea is to move through M_1, M_2 simultaneously and only accept strings which end in final states in both machines
    - for a DFA M, M' accepting the complement of M's language is M with the final and non-final states switched
  - use Theorem 3.4 to conclude that there is a regular expression corresponding to that DFA/NFA

Finite-State Automata and Regular Languages

- for two DFAs M_1, M_2, M accepting $M_1 \cap M_2$ contains pairs of states from M_1 and M_2
  - the idea is to move through M_1, M_2 simultaneously and only accept strings which end in final states in both machines



Finite-State Automata and Regular Languages

- for a DFA M, M' accepting the complement of M's language is M with the final and non-final states switched
  - must have all transitions specified, no omit-trap-state shortcuts!