

Automata and Formal Languages (4)

Email: christian.urban at kcl.ac.uk

Office: SI.27 (1st floor Strand Building)

Slides: KEATS (also home work is there)

Regexps and Automata

Thompson's construction subset construction

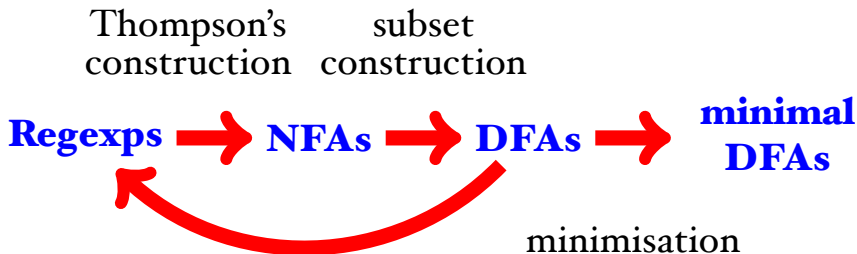
Regexps  **NFAs**  **DFAs**

Regexps and Automata

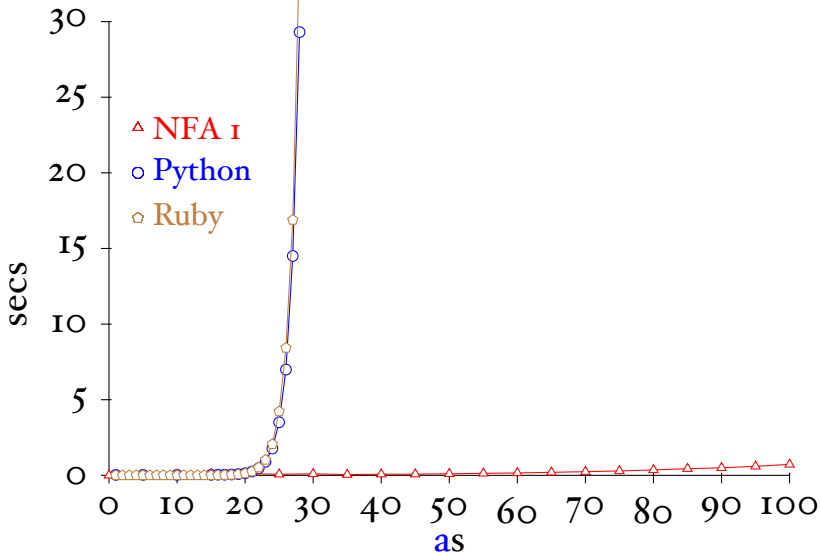
Thompson's construction subset construction



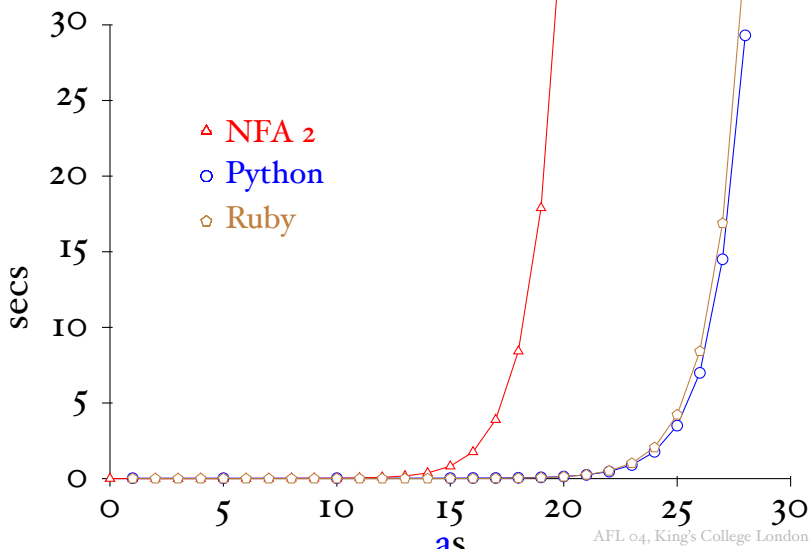
Regexps and Automata



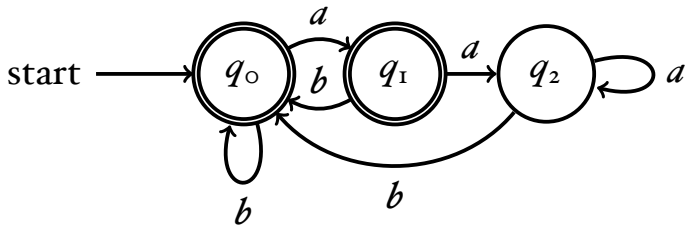
$$(a^{\{n\}}) \cdot a^{\{n\}}$$

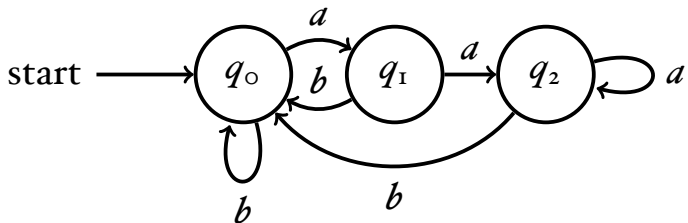


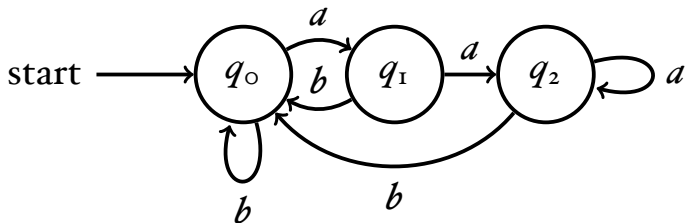
$$(a?\{n\}) \cdot a\{n\}$$



DFA to Rexp



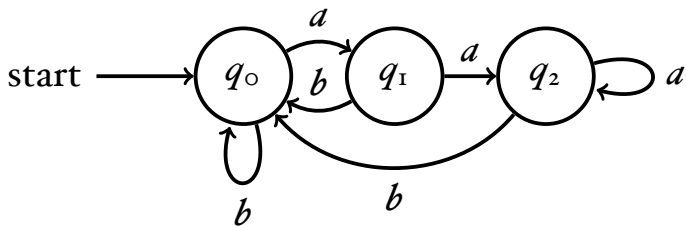


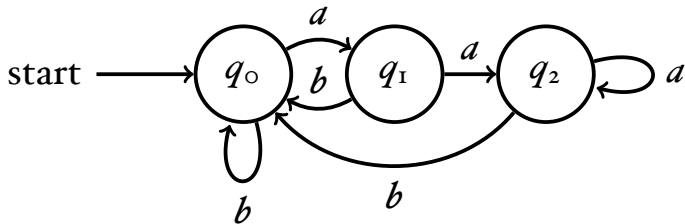


$$q_0 = 2q_0 + 3q_1 + 4q_2$$

$$q_1 = 2q_0 + 3q_1 + 1q_2$$

$$q_2 = 1q_0 + 5q_1 + 2q_2$$

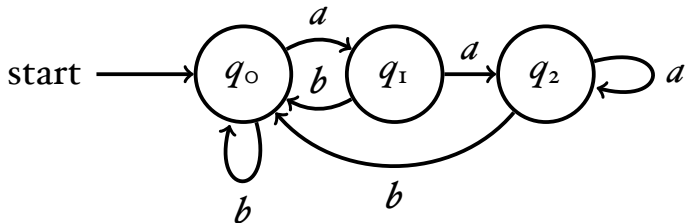




$$q_0 = \epsilon + q_0 b + q_1 b + q_2 b$$

$$q_1 = q_0 a$$

$$q_2 = q_1 a + q_2 a$$



$$q_0 = \epsilon + q_0 b + q_1 b + q_2 b$$

$$q_1 = q_0 a$$

$$q_2 = q_1 a + q_2 a$$

Arden's Lemma:

If $q = qr + s$ then $q = sr^*$

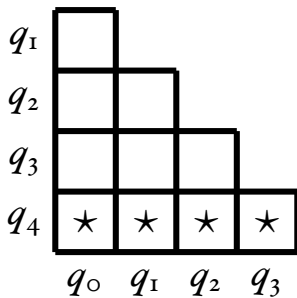
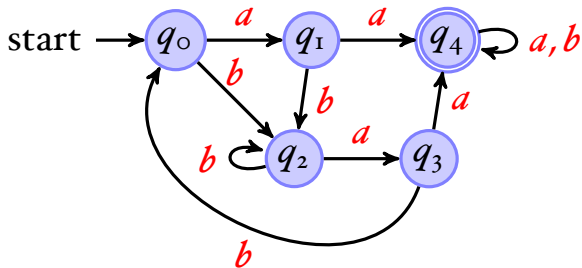
DFA Minimisation

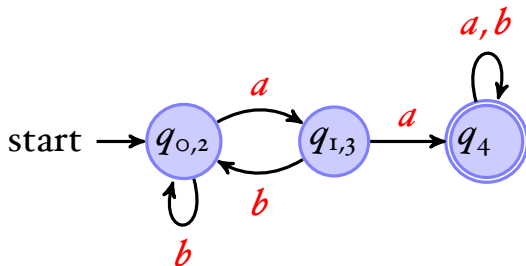
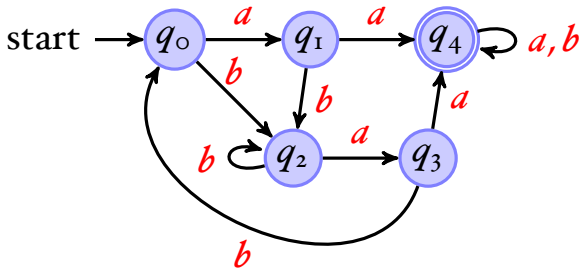
- 1 Take all pairs (q, p) with $q \neq p$
- 2 Mark all pairs that accepting and non-accepting states
- 3 For all unmarked pairs (q, p) and all characters c test whether

$$(\delta(q, c), \delta(p, c))$$

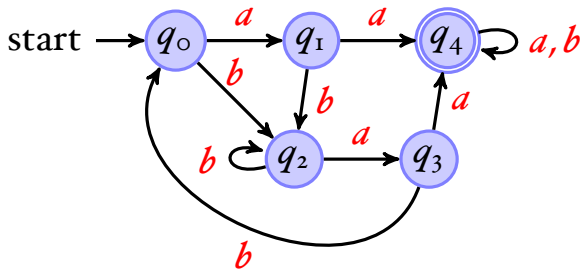
are marked. If yes, then also mark (q, p) .

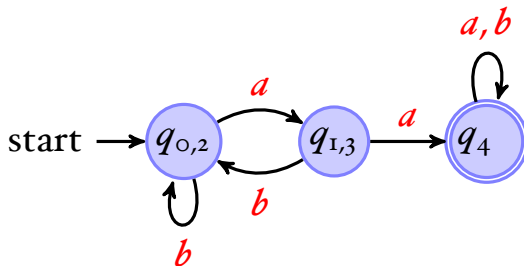
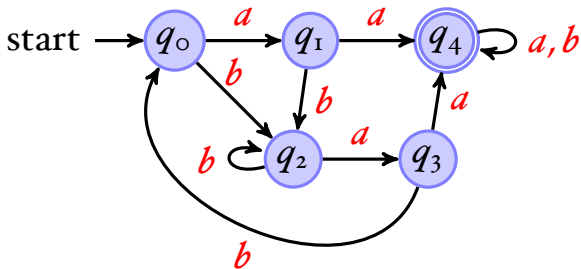
- 4 Repeat last step until no change.
- 5 All unmarked pairs can be merged.





minimal automaton





minimal automaton

- Assuming you have the alphabet $\{a, b, c\}$
- Give a regular expression that can recognise all strings that have at least one b .