8. For languages $A, B \subseteq \Sigma^*$, define $\textsc{Shuffle}(A, B)$ to be the set

$$\{w \mid w = a_1 b_1 a_2 b_2 \cdots a_k b_k \text{ where } a_1 \cdots a_k \in A \text{ and } b_1 \cdots b_k \in B, \text{ with each } a_i, b_i \in \Sigma^*\}.$$

Show that the regular languages are closed under shuffle. Give both a short, convincing, one paragraph "proof idea" similar to those in the text, and a formal proof. Hint: A variant of the "Cartesian product" construction in Theorem 1.25 may be useful. And, yes, "induction is your friend."

Note: Read the definition carefully. It says "$a_1 \cdots a_k \in A$," not "$a_1, \ldots, a_k \in A$"; the later specifies k strings, each individually in A; the former specifies k strings, perhaps none in A, whose concatenation (in order) is a single string in A.

$$\delta((q_1, q_2), c) = \{(\delta_1(q_1, c), q_2),\ (q_1, \delta_2(q_2, c))\}$$

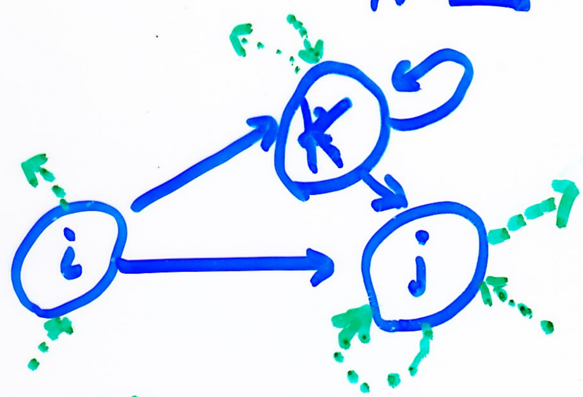
This works in part since, without loss of generality, every (ai,bi) pair has |aibi| = 1, i.e. one is epsilon, the other a single character.

Relating edges of G' to paths of G

A path in G : any sequence of states

A simple path in G : any sequence of ≥ 2 states s.t. 1^{st} & last are not k, and all intermediate ones (if any) are k.

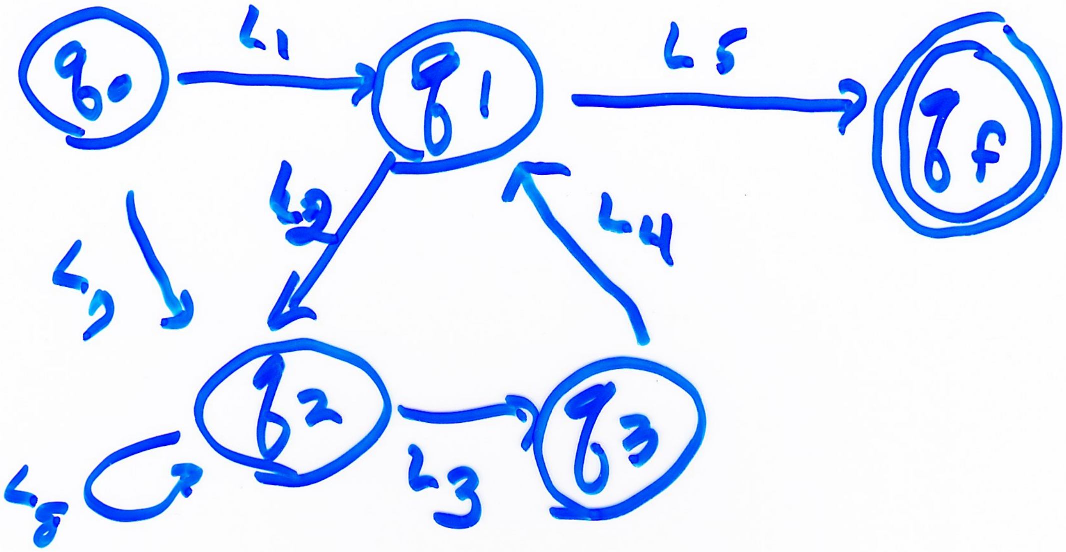
$i \to j$
$i \to k \to j$
$i \to k \to k \to j$
$\vdots$



The Point:

(a) every path in G can be decomposed into simple paths

(b) every edge in G', say $i \to j$, corresponds to the set of all simple paths in G with those endpoints



Q: What strings accepted by

$q_0 \to q_1 \to q_f$?

$\{ w \mid w = x_1 x_2 \text{ with } x_1 \in L_1 \text{ \& } x_2 \in L_5 \}$

$= L_1 \circ L_5$

$q_0 \to q_1 \to q_2 \to q_3 \to q_1 \to q_f$

$L_1 \circ L_2 \circ L_3 \circ L_4 \circ L_5$

$L = \bigcup_{\text{paths } p}$ concat of L_i's on path p

Claim 2

$L(r'_{ij}) = \{ w \mid G$ can move from i to j reading w and passing through no intermediate states except possibly k. $\}$

Equivalently:

$L(r'_{ij}) = \{ w \mid G$ can move from i to j reading w along a <u>simple</u> path $\}$

$\supseteq L(r_{ij} \cup r_{ik} \cdot r_{kk}^{*} \cdot r_{kj})$

Claim 4: $\forall$ NFA $\exists$ equiv. reg. expr.

Proof: NFA $\to$ GNFA $\to$ 2-state GNFA $\to$ r.e.

by induction on k, using claim 1

11-2

EXAMPLE 1.68

In this example we begin with a three-state DFA. The steps in the conversion are shown in the following figure.

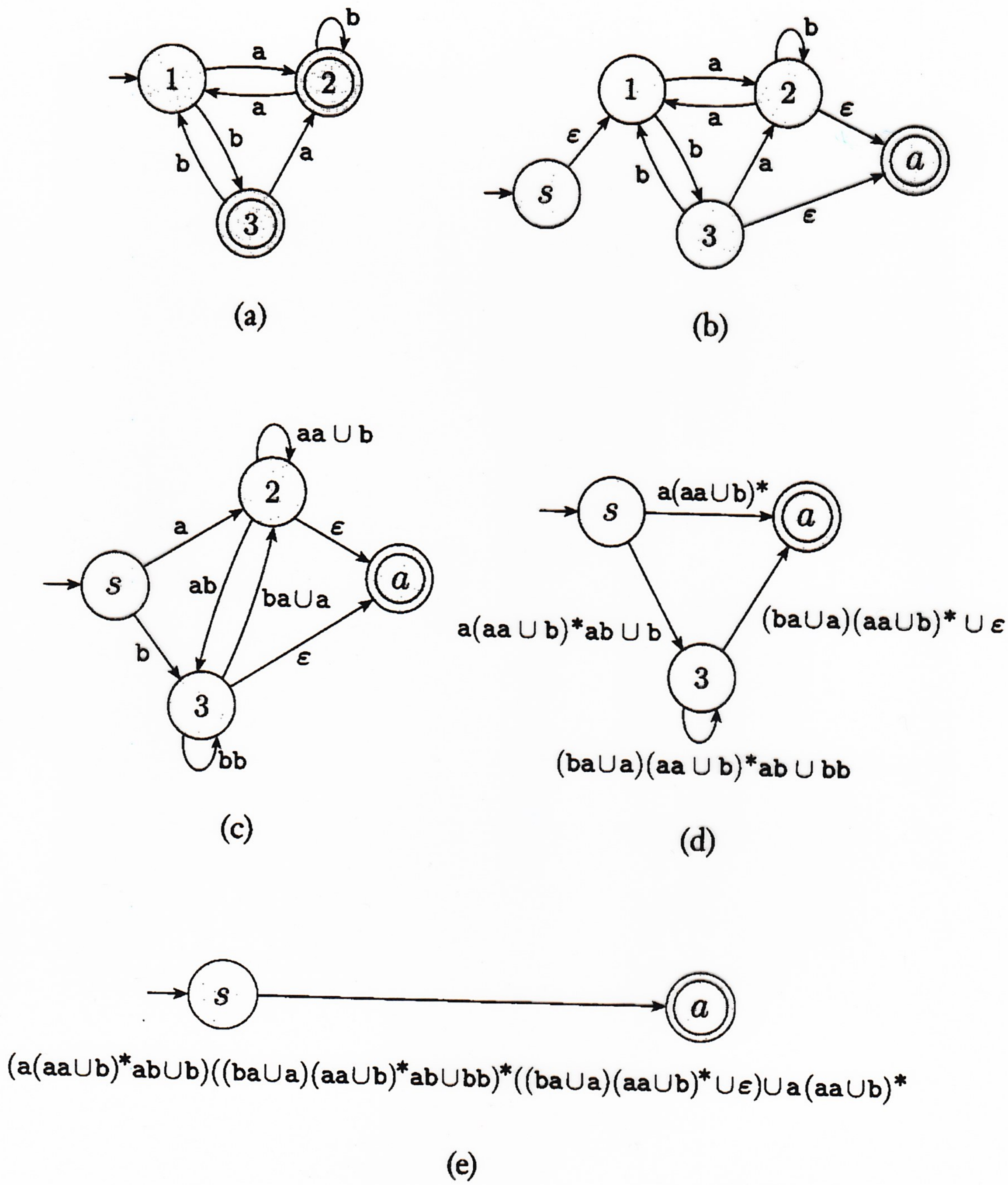


FIGURE 1.69
Converting a three-state DFA to an equivalent regular expression

Summary

L is regular $\iff$

$L = L(M)$ for some DFA M

$\iff$ $L = L(N)$ . . . . NFA N

$\iff$ $L = L(G)$ . . . . . . GNFA G

$\iff$ $L = L(R)$ . . . . . . reg. exp. R