

# Lecture #4: Regular Operations and Regular Expressions

## Proofs of Closure Properties

### Introduction

This document provides a proof of the following result — which was stated, but not proved, in the notes for Lecture #4.

**Theorem 1.** *Let  $\Sigma$  be an alphabet, and let  $A, B \subseteq \Sigma^*$ .*

- (a) *If  $A$  and  $B$  are regular languages then  $A \cup B$  is a regular language, as well.*
- (b) *If  $A$  and  $B$  are regular languages, then  $A \circ B$  is a regular language, as well.*
- (c) *If  $A$  is a regular language then  $A^*$  is a regular language as well.*

### A Useful Minor Result

The following minor result will be repeatedly of use when developing a proof of the above claim.

**Lemma 2.** *Let  $\Sigma$  be an alphabet, and let  $L \subseteq \Sigma^*$ . Then  $L$  is a regular language if and only if  $L$  is the language  $L(M)$  of some nondeterministic finite automaton  $M = (Q, \Sigma, \delta, q_0, F)$  which satisfies the following properties.*

- (a) *There are no transitions into  $q_0$ , at all. That is,  $q_0 \notin \delta(q, \sigma)$  for any state  $q \in Q$  or any symbol  $\sigma \in \Sigma_\lambda$ , so that the only string  $\omega \in \Sigma^*$  such that  $q_0 \in \delta^*(q_0, \omega)$  is the empty string,  $\omega = \lambda$ .*
- (b)  *$M$  has exactly one accepting state,  $q_F$ , and there are no transitions out of this state. That is,  $F = \{q_F\}$  and  $\delta(q_F, \sigma) = \emptyset$  for every symbol  $\sigma \in \Sigma_\lambda$ .*

*Sketch of Proof.* Suppose, first, that  $L$  is the language  $L(M)$  of some nondeterministic finite automaton  $M = (Q, \Sigma, \delta, q_0, F)$  which satisfies properties (i) and (ii), above. Then, since  $M$  is a nondeterministic finite automaton, it follows by the results in established in Lecture #3

that  $L$  is the language of some deterministic finite automaton as well — that is,  $L$  is a regular language.

Suppose, next, that  $L$  is a regular language. Then — once again, by the results established in Lecture #3 —  $L = L(\widehat{M})$  for some nondeterministic finite automaton

$$\widehat{M} = (\widehat{Q}, \Sigma, \widehat{\delta}, \widehat{q}_0, \widehat{F}).$$

Renaming the states in  $\widehat{Q}$  if necessary, we may assume without loss of generality that  $\widehat{Q}$  does not include states called either  $q_0$  or  $q_F$ .

Consider an NFA  $M = (Q, \Sigma, \delta, q_0, F)$  such that the following properties are satisfied.

- $Q = \widehat{Q} \cup \{q_0, q_F\}$  — that is, we have added states  $q_0$  and  $q_F$  to the set of states of  $\widehat{M}$ .
- The only transition out of the new start state,  $q_0$ , is a  $\lambda$ -transition to the old start state  $\widehat{q}_0$  of  $\widehat{M}$ . That is,  $\delta(q_0, \lambda) = \{\widehat{q}_0\}$  and  $\delta(q_0, \sigma) = \emptyset$  for every symbol  $\sigma \in \Sigma$ .
- Transitions for the states in  $\widehat{Q}$  are unchanged — except that a  $\lambda$ -transition is added from each state in  $\widehat{F}$  to the new state  $q_F$ . That is,  $\delta(q, \sigma) = \widehat{\delta}(q, \sigma)$  for every state  $q \in \widehat{Q}$  and symbol  $\sigma \in \Sigma$ , while if  $q \in \widehat{Q}$  then

$$\delta(q, \lambda) = \begin{cases} \widehat{\delta}(q, \lambda) \cup \{q_F\} & \text{if } q \in \widehat{F}, \\ \widehat{\delta}(q, \lambda) & \text{if } q \notin \widehat{F}. \end{cases}$$

- $q_F$  is the only accepting state of  $M$  — that is,  $F = \{q_F\}$  — and there are no transitions out of  $q_F$ . That is,  $\delta(q_F, \sigma) = \emptyset$  for all  $\sigma \in \Sigma_\lambda$ .

Using the above rules, the following properties about  **$\lambda$ -closures of states** are easily established.

- If  $\lambda \notin L$  then the  $\lambda$ -closure of the new start state  $q_0$  in  $M$  is the union of  $\{q_0\}$  and the  $\lambda$ -closure of the old start state,  $\widehat{q}_0$ , in  $\widehat{M}$ .
- On the other hand, if  $\lambda \in L$  then the  $\lambda$ -closure of the new start state  $q_0$  in  $M$  is the union of  $\{q_0, q_F\}$  and the  $\lambda$ -closure of the old start state,  $\widehat{q}_0$ , in  $\widehat{M}$ .
- For every state  $q \in \widehat{Q}$ , if the  $\lambda$ -closure of  $q$  in  $\widehat{M}$  does not include any accepting states (that is, states in  $\widehat{F}$ ), then the  $\lambda$ -closure of  $q$  in  $M$  is the same set as the  $\lambda$ -closure of  $q$  in  $\widehat{M}$ .
- For every state  $q \in \widehat{Q}$ , if the  $\lambda$ -closure of  $q$  in  $\widehat{M}$  *does* include at least one accepting state, then the  $\lambda$ -closure of  $q$  in  $M$  is the union of the  $\lambda$ -closure of  $q$  in  $\widehat{M}$  and the set  $\{q_F\}$ .
- The  $\lambda$ -closure of the new accepting state  $q_F$  in  $M$  is the set  $\{q_F\}$

It follows by the above that

$$\delta^*(q_0, \lambda) = \begin{cases} \{q_0, q_F\} \cup \widehat{\delta}^*(\widehat{q}_0, \lambda) & \text{if } \lambda \in L, \\ \{q_0\} \cup \widehat{\delta}^*(\widehat{q}_0, \lambda) & \text{if } \lambda \notin L, \end{cases}$$

so that  $\lambda \in L(M)$  if and only if  $\lambda \in L(\widehat{M})$ . Furthermore, it can also be proved (by induction<sup>1</sup> on the length of the string  $\omega$ ) that if  $\omega \in \Sigma$  is a *non-empty* string then

$$\delta^*(q_0, \omega) = \begin{cases} \widehat{\delta}^*(\widehat{q}_0, \omega) \cup \{q_F\} & \text{if } \omega \in L, \\ \widehat{\delta}^*(\widehat{q}_0, \omega) & \text{if } \omega \notin L. \end{cases}$$

Thus  $\omega \in L(M)$  if and only if  $\omega \in L(\widehat{M})$  as well.

It follows that  $L(M) = L(\widehat{M}) = L$  and, since  $M$  is a nondeterministic finite automaton that satisfies properties (a) and (b), above, this establishes the claim.  $\square$

## Establishing Closure Under Union

**Lemma 3.** *Let  $\Sigma$  be an alphabet and let  $L_1, L_2 \subseteq \Sigma^*$ . If  $L_1$  and  $L_2$  are both regular languages then  $L_1 \cup L_2$  is a regular language as well.*

*Sketch of Proof.* Let  $\Sigma$  be an alphabet, let  $L_1, L_2 \subseteq \Sigma^*$ , and suppose that the languages  $L_1$  and  $L_2$  are both regular. Then there exist nondeterministic finite automata

$$M_1 = \{Q_1, \Sigma, \delta_1, q_{1,0}, F_1\} \quad \text{and} \quad M_2 = \{Q_2, \Sigma, \delta_2, q_{2,0}, F_2\}$$

such that  $L(M_1) = L_1$ ,  $L(M_2) = L_2$ , and these nondeterministic finite automata have all the properties described in Lemma 2 — so that, in particular,  $F_1 = \{q_{1,F}\}$  for some state  $q_{1,F} \in Q_1$  and  $F_2 = \{q_{2,F}\}$  for some state  $q_{2,F} \in Q_2$ . Renaming states as needed we may assume that  $Q_1 \cap Q_2 = \emptyset$  and that  $q_0 \notin Q_1$  and  $q_0 \notin Q_2$ .

Now consider a nondeterministic finite automaton

$$M = (Q, \Sigma, \delta, q_0, F)$$

that has  $M_1$  and  $M_2$  as components and whose structure is as shown in Figure 1 on page 4. That is,

$$Q = \{q_0\} \cup Q_1 \cup Q_2,$$

the alphabet  $\Sigma$  is the same as for  $M_1$  and  $M_2$ , the new state,  $q_0$ , is the start state,

$$F = F_1 \cup F_2 = \{q_{1,F}, q_{2,F}\},$$

and the transition function  $\delta : Q \times \Sigma_\lambda \rightarrow \mathcal{P}(Q)$  is defined as follows.

---

<sup>1</sup>This proof, and other proofs by induction mentioned in this document, are left as exercises.

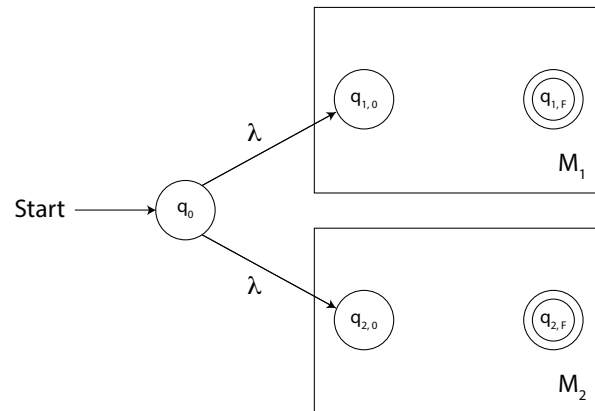


Figure 1: A Nondeterministic Finite Automaton with Language  $L_1 \cup L_2$

- It is only possible to move from the new start state to one of the old start states, and no symbols are processed when doing this — so that

$$\delta(q_0, \lambda) = \{q_{1,0}, q_{2,0}\}$$

and

$$\delta(q_0, \sigma) = \emptyset \quad \text{for every symbol } \sigma \in \Sigma.$$

- All transitions for states in  $Q_1$  are the same in  $M$  as they were in  $M_1$ . That is,

$$\delta(q, \sigma) = \delta_1(q, \sigma) \quad \text{for every state } q \in Q_1 \text{ and for all } \sigma \in \Sigma_\lambda.$$

- All transitions for states in  $Q_2$  are the same in  $M$  as they were in  $M_2$ . That is,

$$\delta(q, \sigma) = \delta_2(q, \sigma) \quad \text{for every state } q \in Q_2 \text{ and for all } \sigma \in \Sigma_\lambda.$$

This can be used to confirm that  **$\lambda$ -closures** in these automata are related as follows.

- The  $\lambda$ -closure of  $q$  in  $M$  is the union of  $\{q_0\}$ , the  $\lambda$ -closure of  $q_{1,0}$  in  $M_1$ , and the  $\lambda$ -closure of  $q_{2,0}$  in  $M_2$ .
- If  $q \in Q_1$  (so that  $q$  is a state in the automaton  $M_1$ ) then the  $\lambda$ -closure of  $q$  in  $M$  is the same set as the  $\lambda$ -closure of  $q$  in  $M_1$ .
- If  $q \in Q_2$  (so that  $q$  is a state in the automaton  $M_2$ ) then the  $\lambda$ -closure of  $q$  in  $M$  is the same set as the  $\lambda$ -closure of  $q$  in  $M_2$ .

It follows from the above that

$$\delta^*(q_0, \lambda) = \{q_0\} \cup \delta_1^*(q_{1,0}, \lambda) \cup \delta_2^*(q_{2,0}, \lambda).$$

On the other hand, if  $\omega$  is a non-empty string in  $\Sigma^*$  then it can be proved, by induction on the length of  $\omega$ , that

$$\delta^*(q_0, \omega) = \delta_1^*(q_{1,0}, \omega) \cup \delta_2^*(q_{2,0}, \omega).$$

Now, since  $F = F_1 \cup F_2$  (and  $q_0 \notin F$ ) it immediately follows that if  $\omega \in \Sigma^*$  then  $\omega \in L(M)$  if and only if either  $\omega \in L(M_1)$  or  $\omega \in L(M_2)$  (or both). That is — since  $L_1 = L(M_1)$  and  $L_2 = L(M_2)$  —

$$L(M) = L_1 \cup L_2.$$

Since  $L_1 \cup L_2$  is the language of a nondeterministic finite automaton it follows, by the results established in Lecture #3, that  $L_1 \cup L_2$  is also the language of a *deterministic* finite automaton. That is,  $L_1 \cup L_2$  is a regular language, as needed to establish the lemma.  $\square$

## Establishing Closure Under Concatenation

**Lemma 4.** *Let  $\Sigma$  be an alphabet and let  $L_1, L_2 \subseteq \Sigma^*$ . If  $L_1$  and  $L_2$  are both regular languages then  $L_1 \circ L_2$  is a regular language as well.*

*Sketch of Proof.* Let  $\Sigma$  be an alphabet, let  $L_1, L_2 \subseteq \Sigma^*$ , and suppose that the languages  $L_1$  and  $L_2$  are both regular. Then there exist nondeterministic finite automata

$$M_1 = \{Q_1, \Sigma, \delta_1, q_{1,0}, F_1\} \quad \text{and} \quad M_2 = \{Q_2, \Sigma, \delta_2, q_{2,0}, F_2\}$$

such that  $L(M_1) = L_1$ ,  $L(M_2) = L_2$ , and these nondeterministic finite automata have all the properties described in Lemma 2 — so that, in particular,  $F_1 = \{q_{1,F}\}$  for some state  $q_{1,F} \in Q_1$  and  $F_2 = \{q_{2,F}\}$  for some state  $q_{2,F} \in Q_2$ . Renaming states as needed we may assume that  $Q_1 \cap Q_2 = \emptyset$  and that  $q_0 \notin Q_1$  and  $q_0 \notin Q_2$ .

Now consider a nondeterministic finite automaton

$$M = (Q, \Sigma, \delta, q_0, F)$$

that has  $M_1$  and  $M_2$  as components and whose structure is as shown in Figure 2 on page 6. That is,

$$Q = \{q_0\} \cup Q_1 \cup Q_2,$$

the alphabet  $\Sigma$  is the same as for  $M_1$  and  $M_2$ , the new state,  $q_0$ , is the start state,

$$F = F_1 = \{q_{2,F}\},$$

and the transition function  $\delta : Q \times \Sigma_\lambda \rightarrow \mathcal{P}(Q)$  is defined as follows.

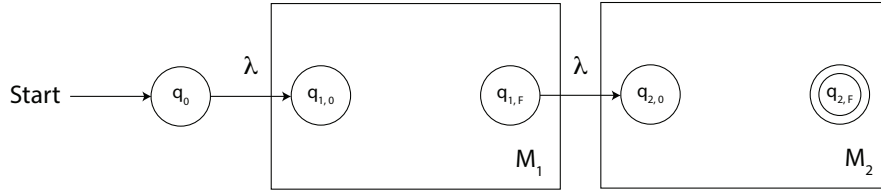


Figure 2: A Nondeterministic Finite Automaton with Language  $L_1 \circ L_2$

- It is only possible to move from the new start state to the start state for  $M_1$ , and no symbols are processed when doing this — so that

$$\delta(q_0, \lambda) = \{q_{1,0}\}$$

and

$$\delta(q_0, \sigma) = \emptyset \quad \text{for every symbol } \sigma \in \Sigma.$$

- For every state  $q \in Q_1$  such that  $q \neq q_{1,F}$  (so that  $q$  is not  $M_1$ 's accepting state)

$$\delta(q, \sigma) = \delta_1(q, \sigma) \quad \text{for all } \sigma \in \Sigma_\lambda.$$

- It is possible to move from  $M_1$ 's accepting state to  $M_2$ 's start state, and no symbols are processed when doing so, so that

$$\delta(q_{1,F}, \lambda) = \{q_{2,0}\}$$

and

$$\delta(q_{1,F}, \sigma) = \emptyset \quad \text{for every symbol } \sigma \in \Sigma.$$

- For every state  $q \in Q_2$ ,

$$\delta(q, \sigma) = \delta_2(q, \sigma) \quad \text{for all } \sigma \in \Sigma_\lambda.$$

This can be used to confirm that  **$\lambda$ -closures** in these automata are related as follows.

- If  $\lambda \in L_1$  (so that  $q_{1,F}$  is in the  $\lambda$ -closure of  $q_{1,0}$  in  $M_1$ ) then the  $\lambda$ -closure of  $q_0$  in  $M$  is the union of  $\{q_0\}$ , the  $\lambda$ -closure of  $q_{1,0}$  in  $M_1$ , and the  $\lambda$ -closure of  $q_{2,0}$  in  $M_2$ .

On the other hand, if  $\lambda \notin L_1$ , then the  $\lambda$ -closure of  $q_0$  in  $M$  is the union of  $\{q_0\}$  and the  $\lambda$ -closure of  $q_{1,0}$  in  $M_1$ .

- For every state  $q \in Q_1$ , if  $q_{1,F}$  is in the  $\lambda$ -closure of  $q$  in  $M_1$ , then the  $\lambda$ -closure of  $q$  in  $M$  is the union of the  $\lambda$ -closure of  $q$  in  $M_1$  and the  $\lambda$ -closure of  $q_{2,0}$  in  $M_2$ .

On the other hand, if  $q_{1,F}$  is *not* in the  $\lambda$ -closure of  $q$  in  $M_1$ , then the  $\lambda$ -closure of  $q$  in  $M$  is the same set as the  $\lambda$ -closure of  $q$  in  $M_1$ .

- For every state  $q \in Q_2$ , the  $\lambda$ -closure of  $q$  in  $M$  is the same set as the  $\lambda$ -closure of  $q$  in  $M_2$ .

It follows from the above that

$$\delta^*(q_0, \lambda) = \begin{cases} \{q_0\} \cup \delta_1^*(q_{1,0}, \lambda) \cup \delta_2^*(q_{2,0}, \lambda) & \text{if } \lambda \in L_1, \\ \{q_0\} \cup \delta_1^*(q_{1,0}, \lambda) & \text{if } \lambda \notin L_1. \end{cases}$$

The following properties can be established by induction on the length of the string,  $\omega$ :

- (a) For all states  $r_1, r_2 \in Q_1$  and for every string  $\omega \in \Sigma^*$ ,

$$r_2 \in \delta^*(r_1, \omega) \quad \text{if and only if} \quad r_2 \in \delta_1^*(r_1, \omega).$$

- (b) For every state  $r_2 \in Q_1$  and for every string  $\omega \in \Sigma^*$ ,

$$r_2 \in \delta^*(q_0, \omega) \quad \text{if and only if} \quad r_2 \in \delta_1^*(q_{1,0}, \omega).$$

- (c) For all states  $r_1 \in Q_1$  and  $r_2 \in Q_2$ ,  $r_2 \in \delta^*(r_1, \omega)$  if and only if there exist strings  $\mu, \nu \in \Sigma^*$  such that the following properties are satisfied.

- i.  $\omega = \mu \cdot \nu$ .
- ii.  $q_{1,F} \in \delta_1^*(r_1, \mu)$ .
- iii.  $r_2 \in \delta_2^*(q_{2,0}, \nu)$ .

- (d) For every state  $r_2 \in Q_2$ ,  $r_2 \in \delta^*(q_0, \omega)$  if and only if there exist strings  $\mu, \nu \in \Sigma^*$  such that the following properties are satisfied.

- i.  $\omega = \mu \cdot \nu$ .
- ii.  $\mu \in L_1$  — so that  $q_{1,F} \in \delta_1^*(q_{1,0}, \mu)$ .
- iii.  $r_2 \in \delta_2^*(q_{2,0}, \nu)$ .

- (e) For all states  $r_1 \in Q_2$  and  $r_2 \in Q$ ,

$$r_2 \in \delta^*(r_1, \omega) \quad \text{if and only if} \quad r_2 \in Q_2 \text{ and } r_2 \in \delta_2^*(r_1, \omega).$$

Since  $F = \{q_{2,0}\}$  it now follows by part (d), above, that — for every string  $\omega \in \Sigma^*$  —  $\omega \in L(M)$  (that is,  $q_{2,F} \in \delta^*(q_0, \omega)$ ) if and only if there exist strings  $\mu, \nu \in \Sigma^*$  such that the following properties are satisfied

- i.  $\omega \in \mu \cdot \nu$ .
- ii.  $\mu \in L_1$  — so that  $q_{1,F} \in \delta_1^*(q_{1,0}, \mu)$ .
- iii.  $q_{2,F} \in \delta_2^*(q_2, \nu)$  — so that  $\nu \in L_2$ .

That is,  $L(M) = L_1 \circ L_2$ .

Since  $L_1 \circ L_2$  is the language of a nondeterministic finite automaton it follows, by the results established in Lecture #3, that  $L_1 \circ L_2$  is also the language of a *deterministic* finite automaton. That is,  $L_1 \circ L_2$  is a regular language, as needed to establish the lemma.  $\square$

## Establishing Closure Under Kleene Star

**Lemma 5.** *Let  $\Sigma$  be an alphabet and let  $L \subseteq \Sigma^*$ . If  $L$  is a regular language then  $L^*$  is a regular language as well.*

*Sketch of Proof.* Let  $\Sigma$  be an alphabet, let  $L \subseteq \Sigma^*$ , and suppose that the language  $L$  is regular. Then there exists a nondeterministic finite automaton

$$M_1 = \{Q_1, \Sigma, \delta_1, q_{1,0}, F_1\}$$

such that  $L(M_1) = L$ , and this nondeterministic finite automaton has all the properties described in Lemma 2 — so that, in particular,  $F_1 = \{q_{1,F}\}$  for some state  $q_{1,F} \in Q_1$ . Renaming states as needed we may assume that  $q_0 \notin Q_1$ .

Now consider a nondeterministic finite automaton

$$M = (Q, \Sigma, \delta, q_0, F)$$

that has  $M_1$  as a component and whose structure is as shown in Figure 3 on page 9. That is,

$$Q = \{q_0\} \cup Q_1,$$

the alphabet  $\Sigma$  is the same as for  $M_1$ , the new state,  $q_0$ , is the start state,

$$F = \{q_0\},$$

and the transition function  $\delta : Q \times \Sigma_\lambda \rightarrow \mathcal{P}(Q)$  is defined as follows.

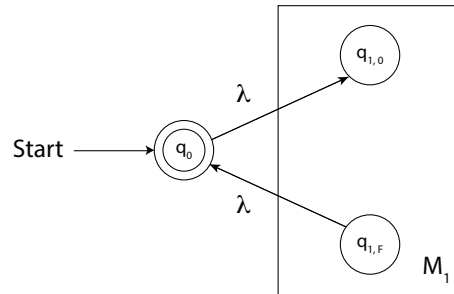


Figure 3: A Nondeterministic Finite Automaton with Language  $L^*$

- It is only possible to move from  $q_0$  to the start state,  $q_{1,0}$ , for  $M_1$ , and no symbols are processed when doing this — so that

$$\delta(q_0, \lambda) = \{q_{1,0}\}$$

and

$$\delta(q_0, \sigma) = \emptyset \quad \text{for every symbol } \sigma \in \Sigma.$$

- For every state  $q \in Q_1$  such that  $q \neq q_{1,F}$ ,

$$\delta(q, \sigma) = \delta_1(q, \sigma) \quad \text{for all } \sigma \in \Sigma_\lambda.$$

- It is only possible to move from  $q_{1,F}$  to  $q_0$ , and no symbols are processed when doing that, so that

$$\delta(q_{1,F}, \lambda) = \{q_0\}$$

and

$$\delta(q_{1,F}, \sigma) = \emptyset \quad \text{for every symbol } \sigma \in \Sigma.$$

This can be used to confirm that  **$\lambda$ -closures** in these automata are related as follows.

- The  $\lambda$ -closure of  $q_0$  in  $M$  is the union of  $\{q_0\}$  and the  $\lambda$ -closure of  $q_{1,0}$  in  $M_1$ .
- For every state  $q \in Q_1$ , if  $q_{1,F}$  belongs to the  $\lambda$ -closure of  $q$  in  $M_1$ , then the  $\lambda$ -closure of  $q$  in  $M$  is the union of the  $\lambda$ -closure of  $q$  in  $M_1$ , the set  $\{q_0\}$ , and the  $\lambda$ -closure of  $q_{1,0}$  in  $M_1$ .

On the other hand, if  $q_{1,F}$  does not belong to the  $\lambda$ -closure of  $q$  in  $M_1$ , then the  $\lambda$ -closure of  $q$  in  $M$  is the same set as the  $\lambda$ -closure of  $q$  in  $M_1$ .

It follows from the above that

$$\delta^*(q_0, \lambda) = \{q_0\} \cup \delta_1^*(q_{1,0}, \lambda)$$

— so that  $\lambda \in L(M)$ , since  $q_0 \in F$ .

The following properties are satisfied for every non-empty string  $\omega \in \Sigma^*$  — and can be proved by mathematical induction on the length of  $\omega$ :

- (a) For every state  $q \in Q_1$ ,  $q \in \delta^*(q_0, \omega)$  if and only if there exists an integer  $k$  such that  $k \geq 0$ , as well as strings  $\mu_1, \mu_2, \dots, \mu_k, \nu \in \Sigma^*$ , such that the following properties are satisfied.
- i.  $\mu_i$  is a non-empty string in  $L = L(M_1)$  for every integer  $i$  such that  $1 \leq i \leq k$ .<sup>2</sup>
  - ii.  $q \in \delta_1^*(q_{1,0}, \nu)$ .
  - iii.  $\omega = \mu_1 \cdot \mu_2 \dots \mu_k \cdot \nu$ .
- (b)  $q_0 \in \delta^*(q_0, \omega)$  — so that  $\omega \in L(M)$  — if and only if there exists a *positive* integer  $k$ , as well as strings  $\mu_1, \mu_2, \dots, \mu_k \in \Sigma^*$ , such that the following properties are satisfied.
- i.  $\mu_i$  is a non-empty string in  $L = L(M_1)$  for every integer  $i$  such that  $1 \leq i \leq k$ .
  - ii.  $\omega = \mu_1 \cdot \mu_2 \dots \mu_k$ .

It follows by the above that  $L(M) = (L(M_1))^* = L^*$ .

Since  $L^*$  is the language of a nondeterministic finite automaton it follows, by the results established in Lecture #3, that  $L^*$  is also the language of a *deterministic* finite automaton. That is,  $L^*$  is a regular language, as needed to establish the lemma.  $\square$

## Completion of the Proof

*Proof of Theorem 1.* Part (a) and (b) of the claim are implied by Lemmas 3 and 4, respectively, with languages  $L_1$  and  $L_2$  (in the lemmas) replaced by  $A$  and  $B$ , respectively. Part (c) of the claim is implied by Lemma 5, with language  $L$  (in the lemma) replaced by  $A$ .  $\square$

---

<sup>2</sup>Note that this part of the claim is trivially satisfied when  $k = 0$  because it is “vacuous” (that is, empty) — because there is no such integer  $i$  or string  $\mu_i$  in this case, at all.