Defn: For any $X, Y \subseteq \Sigma^*$, define $X \cdot Y = \{ xy | x \in X \land y \in Y \}$

Ex:
$X = \{ a, ab \}$
$Y = \{ \varepsilon, b, bb \}$
$X \cdot Y = \{ a, ab, abb, abbb \}$
$Y \cdot X = \{ a, ab, ba, bab, bba, bbab \}$

note $|X \cdot Y| \leq |X| \cdot |Y|$
\( X, Y \subseteq \Sigma^* \)

\[ X \cdot x^* = \exists x \cdot x \cdot x^* \text{ if } x \in X \land y \in Y \]

Examples

\[ \text{Odd parity} \cdot \text{Odd parity} = \text{Even} \]

\[ \text{Odd parity} \cdot \text{Even} = \text{Odd} \]

A \cdot B \quad \text{possible?}

\[ \text{Fruits} \cdot \text{Fruits} \quad \text{Fruits} \]

\[ \text{Fruits} \cdot \text{Fruits} = \text{Fruits} \quad \text{always?} \]
Q:

- Is the class of regular languages closed under concatenation?
- Again, for Java programs, say, it’s not too hard to prove this.
- What about finite automata? Inability to back up the input tape is one issue...

An idea for closure under concatenation, but not clear how to do it – may need to stay in M₁ for several visits to F before jumping to M₂.

E.g.:

{even parity} • {exactly 5 1’s}
which 1 is 5th from end?
L = \{ w \in \{a,b\}^* \mid 3\text{rd letter from the right end of } w \text{ is } "a" \}
L = \{ w \in \{a,b\}^* \mid \text{3rd letter from the right end of } w \text{ is } "a" \} 

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure131.png}
\caption{Example “guess & check”: \( L = \{ a^n \mid n \text{ is a multiple of } 2, 3, 5 \text{ or } 7 \} \)}
\end{figure}

Example “guess & check”:
L = \{ a^n \mid n \text{ is a multiple of } 2, 3, 5 \text{ or } 7 \}

Note: equiv DFA has \( 2^3 \cdot 5 \cdot 7 \) states and messy set of final states
L = \{ w \in \{a,b\}^* | \text{3rd letter from the right end of } w \text{ is } "a" \}

(Non-)Example

L = \{ a^p | p \text{ is prime} \}

M:

Q: is M deterministic?
Q: Does M accept a^p for every prime p?
Q: does L(M) = L?
Q: but, doesn’t it always guess right?

Nondeterminism: How

• View it as a *generator* of a language
• View it as a *recognizer* of a language
  - “build the tree”
  - explore all paths
  - guess-and-check

Nondeterminism: Why

• Specifications: say, clearly & concisely, what, not *how*
• Precise, and often *concise* specification
  - “do A or B, but I don’t yet know/don’t want clutter of saying which”
  - Sometimes *exponentially* more concise - “3rd letter from end”
• Natural model of incompletely specified/partially known systems
  - if correct wrt a partial spec, then correct wrt *any* implementation consistent with that spec
  - “is state ‘reactor boiling / control rods out’ unreachable, even allowing for unknown behavior of subsystem X?”
Kleene Star

* Defn: \( L^* = \bigcup_{n \geq 0} L^n \)

* Examples
  i) \( \Sigma^* \): a simple special case
  ii) \( L = \{ a^p b \mid p \text{ is prime} \} \)
      \( L^* = \{ \varepsilon \} \cup \{ a^{p_1} b a^{p_2} b \ldots b a^{p_k} b \mid k \geq 1, \)
      and each \( p_i \) is prime\}

Closure under union

Given NFA \( M \), can build one for \( L(M)^* \)?
Given NFA $M$, can build one for $L(M)^*$?

No
(may reject $\varepsilon$)

Given NFA $M$, can build one for $L(M)^*$?

or

No, may accept extra stuff (if $M$ can loop back to start before reaching $F$)
Closure under \(*\)

For the correctness proof, there are usually 2 directions, namely:

1) \((L(N_1))^* \subseteq L(N)\) and \(L(N) \subseteq (L(N_1))^*\)

2) \((L(N))^* \subseteq L(N_1)\), or equivalently, given any \(x\) in \(L(N)\), show that it can be broken into \(k \geq 0\) substrings \(x_1, x_2, ..., x_k\) (i.e., \(x = x_1 \ast x_2 \ast ... \ast x_k\)) so that each is in \(L(N_1)\). For this direction, suppose \(q_0 = r_0, r_1, r_2, ..., r_n \in F\) is an accepting trace in \(N\) for \(x\). Note that \(r_1 = q_1\), since the only transition leaving \(q_0\) goes to \(q_1\) (and is labeled \(\varepsilon\)). Let \(x_1\) be the concatenation of all edge labels up to (but excluding) the next green edge (i.e., an \(\varepsilon\)-move from a final state back to \(q_1\)). Note that \(x_1 \in L(N_1)\), since the included transitions are all present in \(N_1\) and run from its start state to a final state, so they are an accepting trace in \(N_1\). Similarly, let \(x_2\) be the concatenation of all edge labels up to the next green edge, ..., and \(x_k\) those after the last green edge. By the same reasoning, each \(x_i \in L(N_1)\), for each \(1 \leq i \leq k\). Finally, note that \(x = x_1 \ast x_2 \ast ... \ast x_k\) since the excluded transitions are all \(\varepsilon\)-moves. \(\therefore x \in (L(N_1))^*\)

QED
Closure under *, Leftovers

There are a few points in the proof above that I deliberately didn’t address. I strongly suggest that you think about them and see if you can fill in missing details and/or explain why they actually are covered, even if not explicitly mentioned. I suggest you write it (but no need to turn it in).

- Are \( x = \epsilon / k = 0 \) correctly handled, or do you need to say more?
- Is it a problem if \( N_1 \)'s start state is a final state?
- Is it a problem if \( N_1 \) includes \( \epsilon \)-moves from (some or all states in) \( F \) to \( q_i \)?
- Is there anything else I omitted?

NFA == DFA, or not?
**Definition**

Let $N_1$ and $N_2$ be NFA. They are equivalent if $L(N_1) = L(N_2)$.

**Theorem 1.39**

A nfa $N$ is equivalent to $M$ if:

1. $N = (Q, \Sigma, \delta, q_0, F)$
2. $M = (Q', \Sigma', \delta', q'_0, F')$

(Warm up: no $\varepsilon$-moves)

$Q' = 2^Q$

$\delta' = \{ (z, \varepsilon) \mid z \in Q \}$

$F' = \{ R \subseteq Q \mid R \cap F \neq \emptyset \}$

For all $a \in \Sigma$, $z \subseteq Q$, $z' \subseteq Q$:

$\delta'(z, a) = \bigcup_{z' \subseteq Q} \delta(z', a)$

**Exercise**: Apply the construction to the NFA below. Note: You will not get the DFA above (but it will be equivalent).
The text’s assertion that the construction given in the proof of Theorem 1.39 (1st ed: 1.19) is “obviously correct” is a little breezy. Here is an outline of a somewhat more formal correctness proof. I will only handle the case where the NFA has no ε-transitions. Notation is as in the book.

For any \( x \in \Sigma^* \), define
\[
Q_{N,x} = \{ r \in Q \mid N \text{ could be in state } r \text{ after reading } x \},
\]
and
\[
Q_{M,x} = \{ r \in Q' \mid M \text{ would be in state } r \text{ after reading } x \}.
\]

The key idea in the proof is that these two sets are identical, i.e., that the single state of the DFA faithfully reflects the complete range of possible states of the NFA. The proof is by induction on \(|x|\).

**Basis:** \(|x| = 0\). Obviously \( x = \epsilon \). Then
\[
Q_{N,x} = \{ \emptyset \} = Q_{M,x}.
\]

The first and third equalities follow from the definitions of “moves” for NFAs and DFAs, respectively, and the middle equality follows from the construction of \( M \).

**Induction:** \(|x| = n > 0\). Suppose \( Q_{N,y} = Q_{M,y} \) for all strings \( y \in \Sigma^* \) with \(|y| < n \), and let \( x \in \Sigma^* \) be an arbitrary string with \(|x| = n > 0\). Since \( x \) is not empty, there must be some \( y \in \Sigma^* \) and some \( a \in \Sigma \) such that \( x = ya \). For any \( r \in Q \),

\[
N \text{ could be in state } r \text{ after reading } x = ya
\]

\[
\Rightarrow \text{ there is some } r' \in Q \text{ such that } N \text{ could be in } r' \text{ after reading } y \text{ and } r \in \delta(r',a)
\]

\[
\Rightarrow r' \in Q_{N,y}.
\]

The equivalence of (1) and (2) follows from the definition of “moves” for NFAs; the last step must be a move that could be in state \( r \) after reading \( a \) if and only if it can reach a final state after reading \( y \), which will be true if and only if \( Q_{N,y} \) contains a final state, which happens if and only if \( Q_{M,y} = T^* \).

Given the equivalence established above, it’s easy to see that \( L(N) = L(M) \), since \( N \) accepts \( x \) if and only if it can reach a final state after reading \( x \), which will be true if and only if \( Q_{N,x} \) contains a final state, which happens if and only if \( Q_{M,x} = T^* \).

The equivalence of (3) and (4) follows from the definition of \( \delta' \). The equivalence of (4) and (5) follows from the induction hypothesis. The equivalence of (5) and (6) follows from the definition of “moves” for DFAs.

The key idea in the proof is that these two sets are identical, i.e., that the single state of the DFA faithfully reflects the complete range of possible states of the NFA. The proof is by induction on \(|x|\).

**Basis:** \(|x| = 0\). Obviously \( x = \epsilon \). Then
\[
Q_{N,x} = \{ \emptyset \} = Q_{M,x}.
\]

The first and third equalities follow from the definitions of “moves” for NFAs and DFAs, respectively, and the middle equality follows from the construction of \( M \).

**Induction:** \(|x| = n > 0\). Suppose \( Q_{N,y} = Q_{M,y} \) for all strings \( y \in \Sigma^* \) with \(|y| < n \), and let \( x \in \Sigma^* \) be an arbitrary string with \(|x| = n > 0\). Since \( x \) is not empty, there must be some \( y \in \Sigma^* \) and some \( a \in \Sigma \) such that \( x = ya \). For any \( r \in Q \),

\[
N \text{ could be in state } r \text{ after reading } x = ya
\]

\[
\Rightarrow \text{ there is some } r' \in Q \text{ such that } N \text{ could be in } r' \text{ after reading } y \text{ and } r \in \delta(r',a)
\]

\[
\Rightarrow r' \in Q_{N,y}.
\]

The equivalence of (1) and (2) follows from the definition of “moves” for NFAs; the last step must be a move that could be in state \( r \) after reading \( a \) if and only if it can reach a final state after reading \( y \), which will be true if and only if \( Q_{N,y} \) contains a final state, which happens if and only if \( Q_{M,y} = T^* \).

Given the equivalence established above, it’s easy to see that \( L(N) = L(M) \), since \( N \) accepts \( x \) if and only if it can reach a final state after reading \( x \), which will be true if and only if \( Q_{N,x} \) contains a final state, which happens if and only if \( Q_{M,x} = T^* \).

The key idea in the proof is that these two sets are identical, i.e., that the single state of the DFA faithfully reflects the complete range of possible states of the NFA. The proof is by induction on \(|x|\).

**Basis:** \(|x| = 0\). Obviously \( x = \epsilon \). Then
\[
Q_{N,x} = \{ \emptyset \} = Q_{M,x}.
\]

The first and third equalities follow from the definitions of “moves” for NFAs and DFAs, respectively, and the middle equality follows from the construction of \( M \).

**Induction:** \(|x| = n > 0\). Suppose \( Q_{N,y} = Q_{M,y} \) for all strings \( y \in \Sigma^* \) with \(|y| < n \), and let \( x \in \Sigma^* \) be an arbitrary string with \(|x| = n > 0\). Since \( x \) is not empty, there must be some \( y \in \Sigma^* \) and some \( a \in \Sigma \) such that \( x = ya \). For any \( r \in Q \),

\[
N \text{ could be in state } r \text{ after reading } x = ya
\]

\[
\Rightarrow \text{ there is some } r' \in Q \text{ such that } N \text{ could be in } r' \text{ after reading } y \text{ and } r \in \delta(r',a)
\]

\[
\Rightarrow r' \in Q_{N,y}.
\]

The equivalence of (1) and (2) follows from the definition of “moves” for NFAs; the last step must be a move that could be in state \( r \) after reading \( a \) if and only if it can reach a final state after reading \( y \), which will be true if and only if \( Q_{N,y} \) contains a final state, which happens if and only if \( Q_{M,y} = T^* \).

Given the equivalence established above, it’s easy to see that \( L(N) = L(M) \), since \( N \) accepts \( x \) if and only if it can reach a final state after reading \( x \), which will be true if and only if \( Q_{N,x} \) contains a final state, which happens if and only if \( Q_{M,x} = T^* \).
Notes on Subset Construction:
1) only the top 6 states are reachable from the start state, but all 16 are required by the construction.
2) ε moves come after Σ moves. E.g., $\delta(q_2,1) = \emptyset$, not $\{q_4\}$.