2.2.7 (a) See figure 1
(b) See figure 2
(c) There is no DFA accepting this language with fewer states. (We’ll prove this when we finish with state minimization.)

2.2.9 (a) The NFA accepts any string with a $b$. There’s a 2-state DFA that accepts this language: the initial state and one final state. From the initial state input $a$ keeps the DFA in the initial state, and $b$ drives it to the final state. From the final state any input keeps it in the final state.

The algorithm for building a DFA from an NFA builds a 3-state DFA with an initial state and two final states. The state minimization algorithm (which we’ve started discussing) will combine the two final states into a single final state.

(b) The only difference between the given NFA and the DFA built by the algorithm is that there is an additional state in the DFA corresponding to the empty set. From each state of the NFA from which there is no transition for one of the symbols of the alphabet $\{(q_1,a), (q_2,b), (q_3,b), (q_4,a)\}$, there is a transition labelled with this symbol from the corresponding state of the DFA to the empty set state of the DFA.

2.4.1 (a) There are several possible finite automata that accept $\{a^{p+q^n} : n \in \mathbb{N}\}$. A particularly simple one consists of $p+q$ states arranged in a row with $a$-transitions joining successive

Figure 1: NFA accepting $(a \cup b)^* aabab$
states as you move from left to right. The $p + 1$st state is the only final state, and there’s an $a$-transition from the rightmost state back to the $p + 1$st. (All transitions are $a$-transitions.) Figure 3 illustrates the case $p = 5$ and $q = 3$.

(b) By using the scheme outlined in part (a) we can build a finite automaton accepting any arithmetic progression. Using the construction of theorem 2.3.1(a), we can build an NFA accepting any finite union of arithmetic progressions.

Additional: Use the proof of the pumping lemma to find a decomposition of $w = xyz$, so that $x, y, \text{ and } z$ satisfy the conclusions of the pumping lemma.

(a) $L$ is the language over $\{a, b\}^*$ consisting of all strings with an odd number of $a$’s, and $M$ is a two-state deterministic finite automaton accepting $L$.

A 2-state DFA accepting $L$ is shown in figure 4

---

Figure 2: DFA accepting $(a \cup b ^ * aabab$)

Figure 3: DFA accepting $a^{5+3n}$

Figure 4: DFA accepting strings with an odd number of $a$’s
i. \( w = abaa \). Since \( M \) has two states, \( n = 2 \). So \( |xy| \leq 2 \) and \( y = ab, a, \) or \( b \). Recollect that \( y \) is chosen so that it drives \( M \) from some state \( q \) back to \( q \). The only one of the candidates for \( y \) that does this is \( b \). Thus \( x = a, y = b, \) and \( z = aa \).

Checking the conclusions of the pumping lemma, we see that \( w = xyz \), and

1. \( y = b \neq e \),
2. \( |xy| = |ab| = 2 \leq n = 2 \), and
3. \( xy^iz = ab^iaa \in L, \) for all \( i \in \mathbb{N} \).

ii. \( w = aaba \). Once again \( n = 2 \). This time, the candidates for \( y \) are \( a \) and \( aa \). Since \( a \) takes \( M \) from state 1 to state 2, while \( aa \) takes \( M \) from state 1 to state 1, we must have \( y = aa \). So \( x = e \) and \( z = ba \).

Checking the conclusions of the pumping lemma, we see that \( w = xyz \), and

1. \( y = aa \neq e \),
2. \( |xy| = |aa| = 2 \leq n = 2 \), and
3. \( xy^iz = (aa)^iba \in L, \) for all \( i \in \mathbb{N} \).

(b) \( L \) is the language over \( \{a, b\}^* \) consisting of all strings with no more than two consecutive \( b \)'s, and \( M \) is the DFA on page 60 of the text that accepts \( L \).

i. \( w = abbaab \). In this case \( M \) has four states. So \( n = 4 \), and there are several possibilities for \( y \). It must be a nonempty substring of \( abba \), and it must drive \( M \) from some state back to that state. So it could be the first \( a \), it could be the substring \( abba \), or it could be \( bba \). Any of these choices will work. Let’s look at the case in which \( y \) is \( abba \). Then \( x = e \), and \( z = ab \).

Checking the conclusions of the pumping lemma, we see that \( w = xyz \), and

1. \( y = abba \neq e \),
2. \( |xy| = |abba| = 4 \leq n = 4 \), and
3. \( xy^iz = (abba)^iba \in L, \) for all \( i \in \mathbb{N} \).

ii. \( w = ababb \). Once again, \( n = 4 \), and there are several choices for \( y : \) it can be the first \( a \), it can be \( aba \), or it can be \( ba \). Let’s look at the case in which \( y = ba \). In this case \( x = a \), and \( z = ab \).

Checking the conclusions of the pumping lemma, we see that \( w = xyz \), and

1. \( y = ba \neq e \),
2. \( |xy| = |aba| = 3 \leq n = 4 \),
3. \( xy^iz = a(aba)^iabb \in L, \) for all \( i \in \mathbb{N} \).