CSCE 428/828 HW 2

due in class Monday 2/13/2017
1. (10 pts) Construct **regular expressions** for the following languages.

(a) \( \{ x \mid x \text{ starts with 0 and has an odd number of symbols, or starts with 1 and has an even number of symbols } \} \)

**SOLUTION:** \( 0(\Sigma \Sigma)^* \cup 1\Sigma(\Sigma \Sigma)^* \)

(b) \( \{ x \mid x \text{ is any string except 11 and 1111 } \} \)

**SOLUTION:** \( \varepsilon \cup \Sigma \cup 0\Sigma \cup 1\Sigma \cup \Sigma \Sigma \cup 0\Sigma \Sigma \cup 1\Sigma \Sigma \cup \Sigma 0\Sigma \cup \Sigma \Sigma 0\Sigma \cup \Sigma \Sigma \Sigma 0 \cup \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \Sigma \ Sigma
2. (10 pts) Use the method studied in class to convert the following DFA to a regular expression. To get partial credit in case of wrong results, show all your intermediate steps.

\[
\begin{align*}
\text{SOLUTION:} & \quad \text{One solution is } b^*a^*b(bb^*a^*b)^*a(a \cup b)^*. 
\end{align*}
\]
3. (15 pts) Find a **minimum state DFA** for regular expression \( b^*a(b \cup ab^*a)^* \). Hints: first convert this RE to NFA, then convert the NFA to DFA, and finally minimize the DFA. To get partial credit in case of wrong results, show all your intermediate steps.

**SOLUTION:** We first construct an equivalent NFA as shown in Figure 1.

![Figure 1: RE to NFA](image)

Next, we construct an equivalent DFA as shown in Figure 2.

![Figure 2: NFA to DFA](image)
Finally, we minimize the number of DFA states as shown in Figure 3. We can see that this DFA accepts all strings with an odd number of $a$’s.

Figure 3: Minimum DFA
4. (10 pts) In your own words, explain the pumping lemma. First give the intuition and provide details. Hint: explain the property that all regular languages have and then explain how to use that property to prove when a language is not regular.

**SOLUTION:**

The pumping lemma describes a common property that all regular languages have. If a language does not have that property, then it is not regular. Thus, the pumping lemma can be used to prove that some language is not regular.

**Pumping Lemma:** if $A$ is regular, then for any long enough string $s$ in $A$, some parts of its first $p$ symbols can be pumped. That is, if $A$ is regular, then there is a pumping length $p \geq 1$ such that every string $s \in A$ with length $\geq p$ can be written as $s = xyz$ where (i) $|y| \geq 1$, (ii) $|xy| \leq p$, and (iii) $\forall i \geq 0$. $xy^iz \in A$.

**Note:** if you want to be really formal, use quantifiers $\exists, \forall$ (e.g., see the Wikipedia page on pumping lemma).

To prove $L$ is not regular, we can use the Pumping Lemma and proof-by-contradiction.

(a) Assume $L$ is regular
(b) Let $p$ be the pumping length
(c) Pick a string $s \in L$ with $|s| \geq p$
(d) Enumerate all possible decompositions of $s$ into $xyz$ with $|xy| \geq p$ and $y \geq 0$.
(e) Show that for each decomposition, there exists an $i \geq 0$ such that $xy^iz \notin L$
(f) Conclude that the assumption is wrong, i.e., $L$ is not regular.
5. (10 pts) Use the **pumping lemma** to prove that the following language is not regular. The alphabet of the language is \{0, 1, +, =\}.

\[ L = \{ x = y + z \mid x, y, z \text{ are binary integers, and } x \text{ is the sum of } y \text{ and } z \} \]

For example, strings 1000 = 101 + 11, 0101 = 010 + 11, and 101 = 101 + 0 are in the language, but strings 100 = 101 + 10, 1 = 1 = 1, 1 + 1 = 10, and 1111 are not in the language.

**SOLUTION:** We’ll use proof by contradiction.

(a) Assume that \( L \) is regular.

(b) Let \( p \) be the pumping length.

(c) Consider string \( 1^p = 1^p + 0 \). It is in language \( L \) and its length is greater than \( p \). According to the pumping lemma, it can be divided into three parts \( xyz \) satisfying \( |y| > 0 \) and \( |xy| \leq p \).

(d) Since \( |xy| \leq p \), part \( y \) consists of only some 1’s before =. Suppose \( y \) consists of \( k \) 1’s, that is, \( y = 1^k \) where \( p \geq k > 0 \). Then according the pumping lemma, for any \( i \geq 0 \), the new string \( xy^iz \) is also in language \( L \). However, if \( i = 0 \), \( xy^0z \) is \( 1^{p-k} = 1^p + 0 \) which is not in language \( L \).

So there is a contradiction. Therefore, the assumption is wrong; that is, language \( L \) is not regular.
6. (10 pts) Use the **pumping lemma** to prove that language \( L \) is not regular.

\[
L = \{1^n x \mid n > 0, x \text{ is a string over alphabet } \{0, 1\} \text{ and } x \text{ contains at most } n \text{ 1s}\}
\]

A string in \( L \) can be split into two parts: the first part contains only 1s (let \( n \) denote the number of 1s in the first part), and the second part \( x \) contains at most \( n \) 1s. For example, string 1111 \( \in L \) because 1111 = \( 1^2 x \) with \( x = 11 \) (or 1111 = \( 1^3 x \) with \( x = 1 \), or 1111 = \( 1^4 x \) with \( x = \varepsilon \)). String 1101 \( \in L \) because 1101 = \( 1^2 x \) with \( x = 01 \). String 111101 \( \in L \) because 111101 = \( 1^4 x \) with \( x = 01 \) (or 111101 = \( 1^3 x \) with \( x = 101 \)).

**SOLUTION:** We’ll use proof by contradiction.

(a) Assume that \( L \) is regular.

(b) Let \( p \) be the pumping length.

(c) Consider string \( s = 1^p 01^p \). Since string \( 1^p 01^p = 1^p x \) with \( x = 01^p \), it is in the language. In addition, its length is greater than \( p \). According to the pumping lemma, it can be divided into three parts \( wyz \) satisfying \(|y| > 0\) and \(|wy| \leq p\).

(d) Since the first \( p \) symbols of string \( s \) contains only symbol 1, part \( y \) contains only symbol 1. Therefore, let \( y = 1^k \) with \( p \geq k \geq 1 \). If \( i = 0 \), we have \( wy^i z = wz = 1^{p-k} 01^p \), which is not in the language. A contradiction!

Therefore, the assumption is wrong; that is, \( L \) is not regular.

**NOTE:**

- \( i = 2 \) does not work. Because we have \( wy^i z = wyyz = 1^{p+k} 01^p \) which is still in the language.
- String \( s = 1^p 1^p \) does not work. Because for any \( i \), we have \( wy^i z = wz = 1^{p+(i-1)k} 1^p \) which is still in the language.
- String \( s = 10^p \) does not work.
7. (10 pts) In class we have learned the following closure properties for regular languages (RL)

(a) \( A \in RL \Rightarrow \overline{A} \in RL \)
(b) \( A \in RL \land B \in RL \Rightarrow (A \cup B) \in RL \)
(c) \( A \in RL \land B \in RL \Rightarrow (A \cap B) \in RL \)
(d) \( A \in RL \land B \in RL \Rightarrow (A \circ B) \in RL \)
(e) \( A \in RL \Rightarrow A^* \in RL \)

- (5pts) Use contrapositive to derive the closure properties for non-regular languages using the above closure properties.
- (5pts) Given \( L_0 = \{ w \mid w \) has the same number of 0’s and 1’s\} is not regular, can you use closure properties to reason about the following languages? (DO NOT use the pumping lemma). Explain your reasons. These languages have the same alphabet \{0, 1\}.
  (a) \( L_1 = \{ w \mid w \) has different number of 0’s and 1’s\}
  (b) \( L_2 = \{ 0^n1^n \mid n \geq 0 \}\)

**SOLUTION:**

- We use contrapositive to derive closure properties for non-regular languages. Recall the contrapositive of \( a \Rightarrow b \) is \( \overline{b} \Rightarrow \overline{a} \).
  (a) \( \overline{A} \notin RL \Rightarrow A \notin RL \)
  (b) \( (A \cup B) \notin RL \Rightarrow A \notin RL \lor B \notin RL \)
  (c) \( (A \cap B) \notin RL \Rightarrow A \notin RL \lor B \notin RL \)
  (d) \( (A \circ B) \notin RL \Rightarrow A \notin RL \lor B \notin RL \)
  (e) \( A^* \notin RL \Rightarrow A \notin RL \)

- Using closure properties
  (a) \( L_1 = \{ w \mid w \) has different number of 0’s and 1’s\}
      Observe that \( L_0 \) is \( L_1 \), and we already know that \( L_0 \) is not regular. Thus by \( \overline{A} \notin RL \Rightarrow A \notin RL \), \( L_1 \) is not regular.
  (b) \( L_2 = \{ 0^n1^n \mid n \geq 0 \}\). We cannot use \( L_0 \) begin non-regular to deduce that \( L_2 \) is non-regular. For example, you might try intersection. But the intersection of a non-regular language and an unknown one does not imply the unknown one is non-regular. You might notice that \( L_2 \) is a subset of \( L_0 \), but the subset of a non-regular language is not necessary non-regular.
8. (15 pts) Considering a language $L$, let $L'$ be the set of all first halves of strings in $L$ so that

$$L' = \{ x \mid \text{for some } y, |x| = |y| \text{ and } xy \in L \}$$

Prove that if $L$ is regular, then $L'$ is also regular by constructing a finite automaton for $L'$. Clearly describe how your machine works.

**SOLUTION:**

If $L$ is regular, there is an NFA $M = (Q, \Sigma, \delta, q_0, \{q_{\text{accept}}\})$ with only one final state recognizing $L$. Below, we construct an NFA $M' = (Q', \Sigma, \delta', q'_0, F')$ to recognize $L'$.

- $Q' = Q \times Q$
- For $p, q \in Q$, $a \in \Sigma$, $\delta'((p, q), a) = \{(u, v) \mid u \in \delta(p, a) \text{ and } q \in E(\delta(v, b)) \text{ for some } b \in \Sigma\}$
- $q'_0 = (q_0, q_{\text{accept}})$
- $F' = \{(p, p) \mid p \in Q\}$

where $E(\delta(v, b))$ is the set of all states in $\delta(v, b)$ and any additional states that can be reached from a state in $\delta(v, b)$ by following only $\varepsilon$ transitions of $M$.

Intuitively, $M'$ keeps track of two states in $M$ using two fingers. As it reads each input symbol, $M'$ uses one finger to simulate $M$ on that symbol. Simultaneously, $M'$ uses the other finger to run $M$ backwards from the accept state on a guessed symbol. $M'$ accepts whenever the forward simulation and the backward simulation are in the same state, that is, whenever the two fingers are together. At those points we are sure that $M'$ has found a string where another string of the same length can be appended to yield a member of $L$, precisely the definition of $L'$. 