1. The (regular) language $ab^* + b^*a(a + b)^*$.

2. $\delta(q_0, a) = (q_1, a, R)$ and $\delta(q_1, a) = (q_2, a, R)$ and $\delta(q_2, b) = (q_3, b, R)$ and $\delta(q_3, a) = (q_f, a, L)$.

3. The following TM accepts $\{a^n b^m \mid n \geq 1 \land n \neq m\}$.

The transition from $q_0$ to $q_6$ changes the first $a$ of the string to 1. Now in $q_6$, an arbitrary number of $a$’s are passed over. If the string consists of only $a$’s, then at the end of the string we encounter $\square$ and go to the final state. Else the first $b$ of the string is changed into 1 by going to $q_2$. Now we can repeatedly change one $a$ and one $b$ into 1: walk to the left passing over 1’s (in $q_2$), until an $a$ is encountered, which is changed into 1 (going to $q_1$); now walk to the right passing over 1’s (in $q_1$), until a $b$ is encountered, which is changed into 1 (going back to $q_2$).

There are two ways to leave this cycle. The first one is if there are more $a$’s than $b$’s: in $q_1$, while walking to the right passing over 1’s, $\square$ is encountered, meaning that the $b$’s have been exhausted. Then we go to the final state immediately. The second one is if there are more $b$’s than $a$’s: in $q_2$, while walking to the left passing over 1’s, $\square$ is encountered, meaning that the $a$’s have been exhausted. Then we go to $q_3$, where we walk to the right passing over 1’s, until a $b$ is encountered, which is changed into 1. This leads to $q_5$, where an arbitrary number of $b$’s are passed over, until $\square$ is encountered, after which we go to the final state.

4. Yes. If there are multiple final states, then select one of them, say $q_f$, as the only final state, and allow all states that used to be final to make a transition to $q_f$ for any input symbol.
5. Let $L_1$ and $L_2$ be recursively enumerable languages. Then there exist TMs $M_1$ and $M_2$ that accept $L_1 \setminus \{\lambda\}$ and $L_2 \setminus \{\lambda\}$, respectively.

We define a TM $N$ which nondeterministically chooses between executing $M_1$ or $M_2$, and accepts the input string if $M_1$ respectively $M_2$ accepts the input string. Clearly, $L(N) = (L_1 \cup L_2) \setminus \{\lambda\}$. So $L_1 \cup L_2$ is recursively enumerable.

We define a TM $N'$ which remembers the input string (for example on an additional tape). $N'$ first executes $M_1$ on the input string. If this leads to acceptance, then $N'$ executes $M_2$ on the input string (on the additional tape). If this also leads to acceptance, then $N'$ goes to a final state. Clearly, $L(N') = (L_1 \cap L_2) \setminus \{\lambda\}$. So $L_1 \cap L_2$ is recursively enumerable.

6. Let the TM $M$ accept $L \setminus \{\lambda\}$ and halt on every non-empty input string. We define a TM $N$ which behaves exactly as $M$. Only, each final state of $M$ becomes a non-final state of $N$ (from which no transitions are possible). And each non-final halt state of $M$ makes one extra transition in $N$, to a final state. Clearly, $L(N) = T \setminus \{\lambda\}$. So $T$ is recursively enumerable.

7. Take some nonempty $\Sigma$. For each finite $k$ there are only finitely many TMs that can be defined using $k$ symbols (also counting brackets, comma’s, equality signs, occurrences of $\delta$, etc.). On the other hand, each TM can be defined using finitely many symbols. So the collection of all TMs over $\Sigma$ can be counted: first count the TMs that are defined using one symbol, then the TMs that are defined using two symbols, etc.

This implies that the collection of languages that are accepted by a TM together with their complements is countable.

Take an $a \in \Sigma$. Suppose, toward a contradiction, that the collection of languages over $a$ is countable: $L_1, L_2, L_3, \ldots$. We define the language $L$ by

$$L = \{ a^i \mid a^i \notin L_i \}$$

Clearly, $L \neq L_i$ for each $i$, because they disagree on $a^i$. So $L$ is not in the sequence $L_1, L_2, L_3, \ldots$. Hence the languages over $a$ are apparently not countable.

Since the collection of recursively enumerable languages together with their complements is countable, while the collection of languages over $\{a\}$ aren’t, we conclude that there must be a language over $\{a\}$ such that neither the language itself nor its complement is recursively enumerable.