9.7 Accepting the empty string is needed for some automata if more than one state must act on the same input symbol. Turing machine does not require A transitions because it does not have to consume the input - it can move back onto the same symbol.

9.11 a) A generic TM can erase (write blanks) and can halt anywhere on the tape. Therefore without information about the particular machine $T$, a second machine $T_0$ has no way of knowing where the end of the data is - if it moves right and finds a blank, it has to keep searching for non-blanks to the right of the blank. So when it hits the actual end of data, it can not recognize it and continues advancing right forever.

9.11 b) Any construction that allows $R_T$ to emulate $T$ and keep track of the rightmost square written will work. One example: Count the number of times $T$ moves right. This number is the rightmost cell that could have data on it, so move to this cell and then move left until a non-blank is found.

10.3 $L$ is r.e. but not $r$. and $M$ accepts $L$. Assume there is a single $w \notin L$ that causes $M$ to loop infinitely. Then there is a language $L_1 = L + w$, and a machine $M_1$ that recognizes $L_1$. Given $M_1$, we can recognize $L$ by building a machine $M_2$ which runs $M_1$ except that if $M_1$ accepts $w$, $M_2$ rejects. Apply same argument inductively for any fixed number of words $w$.

10.11 If: Let $f : \Sigma^* \rightarrow \Sigma^*$ be a computable partial function with range $L$. Then a TM $M_f$ exists, and computes $f$. We construct a machine that enumerates $L$: generate $\Sigma^*_C \equiv \Sigma^*$ in canonical order. Construct a machine $M_N$ that emulates $M_f$ for $n$ steps. Then: $M_L = [\text{for } n = 1 \rightarrow \infty (\text{first } n \text{ terms of } \Sigma^*_C)_n \rightarrow M_L]$ enumerates $L$. Any equivalent construction is acceptable.

Only If: Given $M_L$: construct $f$ by enumerating $L$ and looking up $w$ in the enumeration. For any particular $w \in L$ it will show up in finite steps.

10.16 Construct a 2D array, $R \times C$. Let the horizontal axis (columns) be denominator, the vertical (rows) be numerator. Then $\frac{R}{C}$ (R=row number, N=column number) is entry $(R,N)$ in the table, and a rational number. Since the axes contain all the integers, the table contains all the rationals. List the numbers in the table as follows: for $R = k$, list $(N,k)$ for $N = 1$ to $k$; then list the entries $(k,N)$ from $N = k - 1$ to 1. Start at $k = 1$ and increment $k$. There are many other constructions that also work, the key is that both numerator and denominator have to be generated to some finite limit at each step.