1. (a) Rice’s theorem cannot be used to show that $A$ is decidable since the property involves the operation of the machine and not the language recognized by the machine.

(b) Rice’s theorem does not apply, since the property “$L(M)$ is recognizable” is trivial, since every Turing machine recognizes a recognizable language.

(c) Rice’s theorem can be used to show that $C$ is undecidable, since it is a non-trivial semantic property.

(d) Rice’s theorem does not apply since $A_{TM} \leq_M L(M)$ implies that $L(M)$ is unrecognizable. This is a trivial property since no Turing machine accepts an unrecognizable language.

2. We show that $FIN(\Sigma)$ has a correspondence with the set of binary words $\{0, 1\}^*$, which we know to be countable. We also know that the set of words over $\Sigma$ is countable and can be enumerated in lexicographic order $s_1, s_2, s_3, \ldots$. We define the characteristic sequence of a language $L \in FIN(\Sigma)$ to be a binary string $b = b_1 b_2 \cdots b_n$ with

$$b_i = \begin{cases} 0 & \text{if } s_i \not\in L, \\ 1 & \text{if } s_i \in L. \end{cases}$$

If $s_n$ is the lexicographically greatest string in $L$, then we define $s_j = \varepsilon$ for all $j > n$. The string $s_n$ must exist since $L$ is finite. Then every finite language $L$ has a finite characteristic binary sequence and every finite binary string corresponds to a language over $\Sigma$. Thus, $FIN(\Sigma)$ is countable.

3. We define our problem as the following language

$$L = \{ \langle M \rangle \mid \text{TM } M \text{ has a useless state} \}.$$ 

Then we show that if we can decide $L$, then we can decide $A_{TM}$. Suppose there exists a Turing machine $R$ that decides $L$. Then we can construct the following machine to decide $A_{TM}$:

1. On input $\langle M, w \rangle$, construct the Turing machine $M'$, which operates as follows:
   1. On input $x$, if $x \neq w$, then skip to the next step. Otherwise, simulate $M$ on $w$.
   2. If $M$ rejects $w$ or $x \neq w$, then visit every state except $q_A$ or $q_R$. We indicate that we are doing this by writing a special symbol $\zeta$ to the tape. After we have visited every state, enter $q_R$ and reject.
3. If \( M \) accepts \( w \), then accept.

2. Run \( R \) on \( \langle M' \rangle \).

3. If \( R \) accepts, then reject; otherwise, accept.

If \( M \) does not accept \( w \), then every state of \( M' \) is visited except for \( q_A \). In this case, \( q_A \) is a useless state and \( R \) accepts. If \( M \) accepts \( w \), then \( M' \) will enter the accepting state on input \( w \) and every other state is visited on input \( x \neq w \). In this case \( R \) will reject. Thus, \( M' \) has a useless state iff \( w \notin L(M) \).

Thus, if \( L \) is decidable, we can decide \( A_{TM} \). Therefore \( L \) is undecidable.

4. If the alphabet is unary (\( \Sigma = \{a\} \)), then the strings only differ by length. Then the following algorithm decides Unary PCP:

Given an instance of PCP \((u_1, v_1), \ldots, (u_k, v_k)\) over \( \Sigma = \{a\} \),

1. If there is a pair \((u_i, v_i)\) with \( u_i = v_i \), then this is a trivial match, so accept.

2. If for every pair \((u_i, v_i)\), we have \( |u_i| > |v_i| \), then reject. If \( |u_i| < |v_i| \) for all \((u_i, v_i)\), then reject. In both cases, either the \( u_i \)'s or \( v_i \)'s will be larger and there will never be a match.

3. Otherwise, there is a pair \((u_i, v_i)\) with \( |u_i| > |v_i| \) and a pair \((u_j, v_j)\) with \( |u_j| < |v_j| \). Let \( m = |u_i| - |v_i| \) and \( n = |v_j| - |u_j| \). Then a solution is the sequence of \( m + n \) integers

\[
i_1 = i_2 = \cdots = i_n = i, \quad i_{n+1} = \cdots = i_{m+n} = j
\]

and thus we can accept.

5. (a) False. If \( K \leq_M L \) is decidable and \( L \) is decidable, then \( K \) is decidable. However, \( A_{TM} \) is not decidable, so we cannot conclude anything about the decidability of \( B \). (Note that this does not imply that \( B \) is undecidable; if \( B \) is decidable, there is a mapping reduction from \( B \) to \( A_{TM} \).)

(b) True. If \( K \leq_M L \) and \( L \) is recognizable, then \( K \) is recognizable. Since \( A_{TM} \) is recognizable, \( B \) is recognizable.

6. Let \( M \) be a Turing machine and \( w \) be an input word such that \( \langle M, w \rangle \) is encoded over an alphabet \( \Sigma \) and let \( \zeta \) be a symbol not in \( \Sigma \). Consider the language

\[
L = \{ \langle M, w, \zeta^i \rangle \mid M \text{ accepts } w \text{ within } i \text{ steps} \}.
\]

\( L \) is decidable since it is guaranteed to halt. We can construct a machine that simulates \( M \) on \( w \) for up to \( i \) steps. Either \( M \) accepts \( w \) within \( i \) steps and accepts or \( M \) fails to accept within \( i \) steps and we halt the simulation after \( i \) steps and reject.

Now, we define the following homomorphism \( \varphi \) by \( \varphi(a) = a \) for all \( a \in \Sigma \) and \( \varphi(\zeta) = \varepsilon \), where \( \varepsilon \) is the empty string. Then \( \varphi(L) = A_{TM} \). Since \( A_{TM} \) is undecidable, \( \varphi(L) \) is undecidable and thus decidable languages are not closed under homomorphism.