Solution sketch 4 - Computational Models

1. Use a two tape TM where the second tape will simulate the stack. $M = (Q', \Sigma, \Gamma, \delta', q_0, q_a, q_r)$ where:
   $Q' = Q \cup \{q_a, q_r\}, \Gamma = \Sigma \cup \{\sqcup\}$
   $\delta'(q, \sigma) = (\delta(q, \sigma), \sigma, R)$ for all $\sigma \in \Sigma$
   $\delta'(q, \sqcup) = (q_a, \sqcup, R)$ if $q \in F$
   $\delta'(q, \sqcup) = (q_r, \sqcup, R)$ if $q \notin F$

2. Let $L \in RE$. Let M_L be a TM that accepts L. That is $L(M_L) = L$. Now, a TM that accepts Prefix(L) will do the following: On input x it will use a monotone enumerator for Σ^* to concurrently simulate M_L on all words of the form xy for $y \in \Sigma^*$ (as was shown in the recitation for the TM that accepts $\overline{E_{TM}}$), and will accept if M_L accepts any of the inputs.

3. Let M_L be a TM that accepts $L_f \in RE$. A TM that computes f is the following: On input x, it simulates (simultaneously, using an enumerator for Γ^*) the computations of M_L on all inputs (x, y), and outputs y whenever M_L accepts (x, y). For the other direction, let M be a TM that computes f. A TM that accepts L_f is the following: on input (x, y) it simulates the computation of M on input x and accepts if the result of the computation equals y.
   A remark- a similar proof show that: $L_f \in \mathcal{R} \iff f$ is computable, which means that $L_f \in \mathcal{R} \iff L_f \in \mathcal{RE}$.

4. (a) True. If $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$, decides L then $M = (Q, \Sigma, \Gamma, \delta, q_0, q_r, q_a)$ decides $\overline{L}$.
   (b) False. We know A_{TM} in $R.E.$, but $\overline{A_{TM}}$ isn't.
   (c) True. Given M_1 which decides L_1 and M_2 which decides L_2, we can simply run M_1 on our input, then run M_2 on our input, and accept iff both accepted.
   (d) True. Use definitions, De-Morgan rule and the fact that $R.E$ is closed under union.

(e) True. Construct a non-deterministic TM that works as follows: it guesses how to split the input to an arbitrary number of parts, and then runs a TM that accepts M on all parts. It accepts iff M accepted all parts.

5. Let M be a NTM that accepts $\overline{L}$. On input x, M will:

   (a) Check if x is on the form $< M_1, M_2 >$. if not, reject.

   (b) choose non-deterministic a string w and run M_1 and M_2 on w in parallel (step by step). if they both accept, accept.

6. (a) For any alphabet Σ, the set of decidable languages over Σ is infinite (obviously it includes the infinite set $\{a^i | a \in \Sigma, i \in \mathbb{N}\}$). However, every TM decides exactly one language.

   (b) For any Σ, this computational model includes a finite set of TMs (why?). Now use the first part.

7. We can easily construct a TM M_1, that calculates the series x, f(x), f(f(x)),... for its input x, and accepts iff it hits 1.

   Construct a TM M_2 that works as follows:

   - ignores its input.
   - loops on all natural numbers $x \geq 0$:
     - if $H(M_1, x)$ rejects, accept.

   The required TM is M_3 that works as follows:

   - ignores its input.
   - if $H(M_2, \epsilon)$ accepts, print "the conjecture is wrong".
   - if $H(M_2, \epsilon)$ rejects, print "the conjecture is right".