Question 1:

1.1 A

According to Theorem 4.5 in Sipser, we know that the problem of whether two DFA's recognize the same language is decidable. And according to Theorem 4.3 also in Sipser we know that a REG-EX that generates a string w is also decidable since it converts the REG-EX into its equivalent NFA. Furthermore we also know that each NFA has its equivalent DFA. Therefore, extrapolating from these known Theorems we can solve:

$$EQ_{REX} = \{\langle R_1, R_2 \rangle |\ R_1,\ R_2 \text{ are regular expressions and } L(R_1) = L(R_2)\}$$

Let $P =$ "On input $\langle A_R, B_R \rangle$ where A_R and B_R are regular expressions:
1. Convert regular expressions A_R and B_R to their equivalent NFA's A_N and B_N respectively using Theorem 1.54
2. Convert NFA's A_N and B_N into equivalent DFA's A_D and B_D
3. Run $TM\ F$ from Theorem 4.5 on input $\langle A_D, B_D \rangle$
4. If F accepts, *accept*. If F rejects, *reject*."

Therefore EQ_{REX} is decidable.

1.2 B

To show that the following is decidable:

$$REX_{101} = \{\langle R \rangle |\ R \text{ is a regular expression, and } L(R) \text{ is the set of all binary strings containing 101}\}$$

Construct $Y =$"On input $\langle R \rangle$ where R is a regular expression:
1. Let R' a regular expression that generates all binary strings containing 101.
2. Run $TM\ P$ from above on input $\langle R, R' \rangle$
3. If P accepts, *accept*. If P rejects, *reject*.

Therefore REX_{101} is decidable.

Question 2:

To prove that the following language is decidable:

$$ONE_{PDA} = \{\langle P \rangle |\ P \text{ is a } PDA \text{ with a single active accept state}\}$$

Construct $Q =$ "On input $\langle P \rangle$ where P is a PDA:

1. The equivalence of PDA's to CFG's states we can convert a PDA P to a CFG C
2. Then run $TM\ R$ from Theorem 4.8 on input $\langle C \rangle$
3. If R accepts, *reject*. If R rejects, *accept*."

Since, if there is only one accept state, this must be true so long as the PDA accepts something, we have shown that if the Grammer of the PDA is not empty, it generates some strings and therefore must accept something. And so we know that ONE_{PDA} is decidable.

Question 3:

3.1 A

We know that in-order for the composition of two functions, $g \circ f$ to be Turing-Simulable, it must satisfy both premises 1 and 2. Since we know both g and f are Turing-simulable. This means that both f and g satisfy premises 1 and 2. It then follows that since the TM T_f halts with output $b' \in B$, on its tape, where $b' = f(b)$, $g \circ f = g(f(b) = g(b') = b''$. And so the composition of two Turing-simulable functions must also be Turing-simulable.

3.2 B

To prove that the following language is decidable:

$$\{\langle m, b\rangle |\ \exists b' \in B, |b'| \leq m \text{ and } f(b') = b\}$$

Construct $A =$ "On input $\langle m, b\rangle$ where m is a non-negative integer and b is a binary string:

1. Define function $f^{-1}(b)$ and let that be the function of the TM T_f
2. Run T_f on input $\langle m, b\rangle$.
3. We know T_f now halts with output b' on the tape, if $|b'| \leq m$, *accept*. Else, *reject*."

Therefore the language is decidable.