Question 1: A Variation of the Language $L$

Let us consider a new language $L'$. This language is identical to $L$, except that instead of the instruction types

\begin{align*}
V &\leftarrow V \\
V &\leftarrow V + 1 \\
V &\leftarrow V - 1 \\
\text{IF } V \neq 0 \text{ GOTO } L
\end{align*}

we now have the instruction types

\begin{align*}
V &\leftarrow V \\
V &\leftarrow V + 2 \\
V &\leftarrow V - 1 \\
\text{IF } V = 0 \text{ GOTO } L
\end{align*}

(a) Show that the language $L'$ is at least as powerful as $L$, that is, can compute all functions that $L$ can compute. \textbf{Hint:} You need to show for each instruction type in $L$ how it can be simulated in $L'$, i.e., how we could translate it into $L'$ code that has the same effect.

(b) Write down the entire $L'$ code for the new universal programs $U'_n$ that can execute the code of any $L'$ program. Of course you can use macros, and you can reuse most of the code that we wrote for programming $U_n$ (see slides and textbook).
**Question 2: Some Set Operations**

Let A and B be sets. Prove or disprove:

(a) For all sets A and B, if A and B are both r.e., then \( A \cup B \) is also r.e.

(b) For all sets A and B, if A and B are both recursive, then \( A \cup B \) is also recursive.

(c) If \( A \subset B \) and B is r.e., then A is r.e.

(d) If \( A \cup B \) is recursive, then both A and B are recursive.

(e) If A is recursive, then \( \neg A \) is also recursive.

**Question 3: What About These Sets?**

For each of the following sets, determine whether it is recursive, r.e., or neither. Prove your answer. Of course, if you prove that a set is recursive, it is clear that it is also r.e., and you do not have to prove that.

(a) \( A = \{ x \in \mathbb{N} \mid x \text{ mod } 7 = 2 \} \)

(b) \( B = \{ x \in \mathbb{N} \mid x \text{ is the number of a program that computes the function } f(n) = 2n \} \)

(c) \( C = \{ x \in \mathbb{N} \mid x \text{ is the number of a program that terminates on input } 12 \text{ after at most 50 steps} \} \)

(d) \( D = \{ x \in \mathbb{N} \mid x \text{ is the number of a program whose output is defined for at least one input} \} \)