Proof. \((\Leftarrow)\) Let \(A\) and \(B\) be recursive sets. Define the following function:

\[
f(x) = \begin{cases} 
1 & \text{if } x/2 \in A \text{ or } x/2 - 1 \in B \\
0 & \text{otherwise}
\end{cases}
\]

\(f(x)\) is effectively computable by the following algorithm: For input \(x\), compute \(x/2\) and check if it is in \(A\) (this is decidable since \(A\) is recursive by assumption); if yes, output 1, otherwise compute \((x - 1)/2\) and check if it is in \(B\) (this is also decidable since \(B\) is recursive); if yes, output 1, otherwise output 0. Therefore, by the Church-Turing thesis, \(f(x)\) is computable.

Since \(f(x)\) is the characteristic function of \(A \oplus B\), by the definition of recursive sets, \(A \oplus B\) is recursive.

\((\Rightarrow)\) Let \(A \oplus B\) be recursive. We capitalize on the fact that if an even number \(2x\) is in \(A \oplus B\), \(x \in A\), and if an odd number \(2x - 1\) is in \(A \oplus B\), \(x \in B\).

Define the following functions:

\[
c_A(x) = \begin{cases} 
1 & \text{if } 2x \in A \oplus B \\
0 & \text{if } 2x \notin A \oplus B
\end{cases}
\]

\[
c_B(x) = \begin{cases} 
1 & \text{if } 2x - 1 \in A \oplus B \\
0 & \text{if } 2x - 1 \notin A \oplus B
\end{cases}
\]

\(c_A\) is effectively computable by the following algorithm: For input \(x\), compute \(2x\) and check if \(2x \in A \oplus B\) (This is decidable since \(A \oplus B\) is recursive by assumption): If yes, output 1, if not, output 0. By the Church-Turing thesis, \(c_A\) is computable.

Similarly \(c_B\) is effectively computable by the following algorithm: For input \(x\), compute \(2x - 1\) and check if \(2x - 1 \in A \oplus B\) (This is decidable for the same reason as above); If yes, output 1, if not, output 0. By the Church-Turing thesis, \(c_B\) is computable.

\(c_A\) and \(c_B\) are the characteristic functions of \(A\) and \(B\) respectively. Therefore, \(A\) and \(B\) are recursive.

(b) Prove that if \(A, B \neq \emptyset\), then \(A \otimes B\) is recursive iff \(A\) and \(B\) are both recursive.

Proof. Let \(A, B \neq \emptyset\).

\((\Leftarrow)\) Let \(A\) and \(B\) be recursive sets. Define the following function:

\[
f(x) = \begin{cases} 
1 & \text{if } \pi_1(x) \in A \text{ and } \pi_2(x) \in B \\
0 & \text{otherwise}
\end{cases}
\]

\(f(x)\) is effectively computable by the following algorithm: For input \(x\), compute \(\pi_1(x)\) and \(\pi_2(x)\); Check \(\pi_1(x) \in A\) and \(\pi_2(x) \in B\), which is decidable since both \(A\) and \(B\) are both recursive by assumption; If yes, output 1, if no, output 0. By the Church-Turing thesis, \(f(x)\) is computable.

Since \(f(x)\) is the characteristic function of \(A \otimes B\), \(A \otimes B\) is recursive.

\((\Rightarrow)\) Let \(A \otimes B\) be recursive. Pick any \(n \in A \otimes B\) and let \(\pi_1(n) = a\) and \(\pi_2(n) = b\). It is clear that \(a \in A\) and \(b \in B\). We will use \(a\) and \(b\) immediately below.

\(x \in A\) is effectively decidable by the following algorithm: For any \(x\), compute \(\pi(x, b)\); Check if \(\pi(x, b) \in A \otimes B\) (which is decidable since \(A \otimes B\) is recursive); If yes, \(x \in A\), if no, \(x \notin A\). This is because \(x \in A\) if \(\pi(x, y) \in A \otimes B\) for any \(y \in B\).

By a similar algorithm, \(y \in B\) is effectively decidable. That is, for any \(y\), compute \(\pi(a, y)\); Check if \(\pi(a, y) \in A \otimes B\); If yes, \(y \in B\), if no, \(y \notin B\).
Since both \( x \in A \) and \( y \in B \) are effectively decidable, the characteristic functions of \( A \) and \( B \) are effectively computable. By the Church Turing thesis, they are computable. Therefore, \( A \) and \( B \) are recursive.

(c) Show why the nonemptiness restriction is required in (b).

**Proof.** Suppose \( A = \emptyset \). Then, \( A \) is recursive since its characteristic function \( f_A \) is computable. Also, \( A \otimes B = \emptyset \) for any \( B \). Therefore, the 'if' direction of (b) holds. However, the 'only if' direction is false, since \( A \otimes K = \emptyset \) is recursive. The same argument applies to the case where \( B = \emptyset \).

(a) Suppose \( A \) is creative and \( B \neq \emptyset \). We first show that \( A \oplus B \) and \( A \otimes B \) are r.e. as well.

Indeed, since \( A \) is creative it is r.e. as well, and so "\( x \in A \)" and "\( x \in B \)" are both partially decidable predicates. We have shown that the statements "\( y \) is even" and "\( y \) is odd" are decidable. Since

\[
f(y) = \begin{cases} y/2 & \text{if } y \text{ is even} \\ (y-1)/2 & \text{if } y \text{ is odd} \end{cases}
\]

is computable, we have that

"\( (y \) is even AND \( f(y) \in A \) OR (\( y \) is odd AND \( f(y) \in B \))"

is a partially decidable predicate, and this is equivalent to the statement "\( y \in A \oplus B \)". Therefore, \( A \oplus B \) is r.e. Moreover, since each of the projection functions \( \pi_1 \) and \( \pi_2 \) are computable, we have that "\( \pi_1(z) \in A \) and \( \pi_2(z) \in B \)" is a partially decidable predicate, and it is equivalent to "\( z \in A \oplus B \)". Thus, \( A \otimes B \) is r.e. as well.

Now, we show that \( A \oplus B \) is productive. Notice that \( f(x) = 2x \) is a total computable function and that \( x \in A \) if and only if \( f(x) = 2x \in A \oplus B \). Thus, by Theorem 3.2 in Cutland, since \( A \) is productive, \( A \oplus B \) is productive as well.

Since \( B \neq \emptyset \), we can choose \( y_0 \in B \) and similarly use the total computable function \( g(x) = \pi_1(x,y_0) \) in Theorem 3.2 to see that \( A \otimes B \) is productive.

It follows that both \( A \oplus B \) and \( A \otimes B \) are creative. QED

(b) Suppose \( B \) is recursive.

Assume \( A \oplus B \) is creative. Then \( A \oplus B \) is r.e. and its complement is productive. Therefore, the predicate "\( x \in A \oplus B \)" is partially decidable. We have shown that \( 2x \) is a computable function, and so by substitution, "\( 2x \in A \oplus B \)" is partially decidable as well. But this statement is equivalent to "\( x \in A \)"; so \( A \) is r.e.

We now show that \( A \) is productive. To do so, we first show that \( A \) is not recursive. For, since \( A \oplus B \) is creative, its complement fails to be r.e. and so it is not recursive. By Problem 1(a), we have that \( A \) is not recursive. In particular, this means \( A \neq \emptyset \) and \( A \neq \mathbb{N} \), so we can choose fixed elements \( a_0 \in A \) and \( b_0 \in A \).

Now, define the function

\[
f(z) = \begin{cases} z/2 & \text{if } z \text{ is even} \\ a_0 & \text{if } z \text{ is odd and } (z - 1)/2 \in B \\ b_0 & \text{if } z \text{ is odd and } (z - 1)/2 \notin B \end{cases}
\]

First, note that since \( B \) is recursive and \( (z - 1)/2 \) is computable, \( (z - 1)/2 \in B \) and \( (z - 1)/2 \notin B \) are decidable predicates. Combining this with the fact that parity is decidable, we have that \( f \) is a total computable function.

If \( z \) is even, we have that \( z \in A \oplus B \) if and only if \( z/2 = f(z) \in A \), and if \( z \) is odd, we have that \( z \in A \oplus B \) if and only if \( (z - 1)/2 \in B \), which occurs if and only if \( f(z) \in A \). Thus, by Theorem 3.2, \( A \) is productive as well. It follows that if \( A \oplus B \) is creative, then so is \( A \).

Finally, suppose \( A \otimes B \) is creative. Since creative sets are nonempty, we have that both \( A \) and \( B \) are nonempty. By part 1(b), it follows that \( A \) is not recursive. Thus, again we can choose \( a_0 \in A \).

We now define

\[
g(z) = \begin{cases} \pi_1(z) & \pi_2(z) \in B \\ a_0 & \pi_2(z) \notin B \end{cases}
\]

Then using the fact that \( B \) is recursive, we have that \( g \) is a total computable function.

Moreover, \( z \in A \otimes B \) if and only if \( g(z) \in A \), and so by Theorem 3.2, \( A \) is productive. Therefore, if \( A \otimes B \) is creative, then so is \( A \).

QED
3. (a) In problem set 8 we showed that $A$ and $B$ are recursively inseparable iff whenever $A \subseteq W_a, B \subseteq W_b$, and $W_a \cap W_b = \emptyset$ then there is a number $x \notin W_a \cup W_b$.

If $A$ and $B$ are effectively recursively inseparable all these hypotheses are met. In particular, $f(a, b) = x$.

(b) We know from the last problem set that $K_a$ and $K_b$ are c.e. So we can use the programs for their partial characteristic function to make a new computable function $f(a, b, x)$ such that

$$f(a, b, x) = \begin{cases} 1 & : x \in W_a \\ 0 & : x \in W_b \\ \text{und.} & : \text{otherwise} \\ \end{cases}$$

for $W_a \cap W_b = K_a \subseteq W_a$ and $K_b \subseteq W_b$. We can use the srm-n theorem to get

$$\phi_{k(a, b)}(x) \equiv f(a, b, x).$$

Now if we consider $\phi_{k(a, b)}(k(a, b))$ it could be either 1, 0, or undefined.

If it is 1 then $k(a, b) \in W_a$. But by definition we know $k(a, b) \notin K_b \subseteq W_b$ because the output is 1. This is a contradiction since the output isn't 1.

If the output is 0 then we end up with an analogous contradiction, so we're forced to conclude $\phi_{k(a, b)}(k(a, b))$ is undefined.

This means $k(a, b) \notin W_a \cup W_b$, and thus $K_a$ and $K_b$ are effectively recursively inseparable.

(c) By symmetry we can just prove $A$ is creative and that $B$ is creative follows.

The big idea is to manipulate $f(a, b)$ so that it can serve as a production function.

We start by constructing on $f(a, b, x) = \phi_{k(a, b)}(x)$ like in (b). Since we want a witness $g(x) \in \overline{A} / W_a$, for any $W \subseteq A$, and we have a way to get a witness $f(a, b) \in \overline{A} / W_a$ we need to combine $W$ with $W_b \cap B \subseteq W$. We can do this by noting that by Cutland 5.3.1.3.

We can do this by noting that by Cutland 5.3.1.3.

$W_{\text{sec.} 3} = W_a \cup W_b$. By hypothesis $B \subseteq W_b$ and $W_a \cap W_b = \emptyset$ so $W_{\text{sec.} 3} \subseteq A$. Thus $f(a, s(x, b)) \in \overline{A} / W_{\text{sec.} 3}$, i.e. if $g(x) = f(a, s(x, b))$ then $g(x) \in \overline{A} / W_a$.

Thus $A$ is productive, $A$ is c.e. and therefore $A$ is creative.
We have \( A, B \) simple

\[
\begin{align*}
&\Rightarrow \text{ } A, B \text{ RE } \\
&\Rightarrow \text{ } \overline{A, B} \text{ infinite } \\
&\Rightarrow \text{ } A, B \text{ contain no infinite RE subset }
\end{align*}
\]

Since there are infinitely many elements not in \( A \), there are infinitely even numbers not in \( A \oplus B \) so there are infinitely many numbers not in \( A \oplus B \).

\( A \oplus B \) RE as follows:

Let \( x \) be partial then \( f(x) \) for \( A \oplus B \)

We have \( x_A, x_B \) as partial then \( f(x) \) for \( A, B \) since they're RE

Define:

\[
K(x) = \begin{cases} 
1 & \text{if } x \text{ even and } x_{A \oplus B} \\
0 & \text{if } x \text{ odd and } x_{A \oplus B} \\
\text{u.d.} & \text{otherwise}
\end{cases}
\]

Suppose \( \overline{A \oplus B} \) contains an infinite RE subset. Then \( \overline{2x : x \in A} \cup \overline{2x + 1 : x \in B} \) contains an infinite RE subset.

Then \( \overline{2x} : x \in A \cup \overline{2x + 1} : x \in B \) contains an infinite RE subset since we can partition the RE subset into sub-subsets from \( A, B \) this can't possible \( \Rightarrow A \oplus B \) simple.

We have satisfied all criteria, \( \Rightarrow A \oplus B \) is simple.
If $A$ is simple, $N$ is r.e. so $\overline{A \cap N}$ is r.e.

By $1B$, if $A$ and $N$ are non-empty, so $A \cap N$ is recursive if $A$ and $N$ are recursive, but $A$ is not recursive so $N$ is r.e.

By $2B$, $N$ is recursive, so if $A \cap N$ is recursive, so $A \cap N$ can be recursive.

If $A \cap N$ were simple, it would have no infinite r.e. subset. However, it is finite so it’s not possible.

Thus $A \cap N$ is not all of $N$, so there are $a, a' \in A$, such

$$\forall n, (2n+1)a' \in \overline{A \cap N}.$$ 

But $\exists (2n+1)a'$, $\{a' \mid (2n+1)a' \in \overline{A \cap N}\}$ is infinite r.e. subset of $\overline{A \cap N}$ because

$$\Phi(e)(x) = \begin{cases} 1 & \text{if } a' \mid x \text{ and } a \text{ odd} \\ \text{undefined otherwise} & \end{cases}$$

is computable because

the predicate $a' \mid x$ of odd is decided as decidable predicates.

This $\overline{A \cap N}$ has an infinite r.e. subset so $\overline{A \cap N}$ is not simple. Thus $A \cap N$ is not r.e. Hence, nor

$A \cap N$ is not simple.