The following exercises are designed to test your understanding of recursion. The functions are defined using a variant of LISP known as meta-LISP. In order to aid your understanding, the function defined in problem 1 is identical to the one below:

\[
\text{drop}(x) = \begin{cases} 
\text{null } x & \text{then nil} \\
\text{else (car } x\text{) cons drop(cdr } x) 
\end{cases}
\]

The idea is that
\[
\begin{align*}
a & = \text{car } x \\
d & = \text{cdr } x \\
n & = \text{null } x \\
at & = \text{atom } x \\
a.b & = a \text{ cons } b \\
\langle a \rangle & = a \text{ cons nil = a list whose single element is } a \\
a*b & = \text{concatenate lists } a \text{ and } b \text{ (i.e. append list } b \text{ to list } a) \\
\text{reverse}[x] & = \text{reverses the top level list } x. \text{ For example } \text{reverse}[(A \ B \ C)] = (C \ B \ A). \text{ But } \\
\text{reverse}[(A \ B \ C)(D \ E)] & = ((D \ E)(A \ B \ C)).
\end{align*}
\]

1. Consider the function \(\text{drop}\) defined by
\[
\text{drop}[x] \leftarrow \begin{cases} 
\text{null } x & \text{then nil} \\
\text{else [a x].drop[d x].} 
\end{cases}
\]
Compute (by hand) \(\text{drop}[(A \ B \ C)].\) What does \(\text{drop}\) do to lists in general?

2. What does the function \(\text{r2}[x]\) do to lists of lists? How about
\[
\begin{align*}
\text{r2}[x] & \leftarrow \begin{cases} 
\text{null } x & \text{then nil} \\
\text{else reverse[a x].r2[d x]} 
\end{cases} \\
\text{r3}[x] & \leftarrow \begin{cases} 
\text{atom } x & \text{then x} \\
\text{else reverse[r4[x]]} 
\end{cases} \\
\text{r4}[x] & \leftarrow \begin{cases} 
\text{null } x & \text{then nil} \\
\text{else r3[a x].r4[d x]} 
\end{cases}
\end{align*}
\]

3. Compare the following function with the function \(\text{r3}\) of the preceding example:
\[
\text{r3'}[x] \leftarrow \begin{cases} 
\text{atom } x & \text{then x} \\
\text{else r3'[d x]^r3'[a x]} 
\end{cases}
\]

4. Consider \(\text{r5}\) defined by
\[
\begin{align*}
\text{r5}[x] & \leftarrow \begin{cases} 
\text{null } x \lor \text{null } d x & \text{then x} \\
\text{else [a r5[d x]] . r5[a . r5[d r5[d x]]].} 
\end{cases}
\end{align*}
\]
Compute \(\text{r5}[(A \ B \ C \ D)].\) What does \(\text{r5}\) do in general. Needless to say, this is not a good way of computing this function even though it involves no auxiliary functions.