rec.1  Bounded Minimization

It is often useful to define a function as the least number satisfying some property or relation \( P \). If \( P \) is decidable, we can compute this function simply by trying out all the possible numbers, 0, 1, 2, \ldots, until we find the least one satisfying \( P \). This kind of unbounded search takes us out of the realm of primitive recursive functions. However, if we’re only interested in the least number \textit{less than some independently given bound}, we stay primitive recursive. In other words, and a bit more generally, suppose we have a primitive recursive relation \( R(x, z) \). Consider the function that maps \( y \) and \( z \) to the least \( x < y \) such that \( R(x, z) \). It, too, can be computed, by testing whether \( R(0, z) \), \( R(1, z) \), \ldots, \( R(y - 1, z) \). But why is it primitive recursive?

**Proposition rec.1.** If \( R(x, z) \) is primitive recursive, so is the function \( m_R(y, z) \) which returns the least \( x < y \) such that \( R(x, z) \) holds, if there is one, and 0 otherwise. We will write the function \( m_R \) as

\[
\min_{x < y} R(x, z),
\]

**Proof.** Note than there can be no \( x < 0 \) such that \( R(x, z) \) since there is no \( x < 0 \) at all. So \( m_R(x, 0) = 0 \).

In case the bound is \( y + 1 \) we have three cases: (a) There is an \( x < y \) such that \( R(x, z) \), in which case \( m_R(y + 1, z) = m_R(y, z) \). (b) There is no such \( x \) but \( R(y, z) \) holds, then \( m_R(y + 1, z) = y \). (c) There is no \( x < y + 1 \) such that \( R(x, z) \), then \( m_R(y + 1, z) = 0 \). So,

\[
m_R(0, z) = 0
\]

\[
m_R(y + 1, z) = \begin{cases} m_R(y, z) & \text{if } \exists x < y \text{ such that } R(x, z) \\ y & \text{otherwise, provided } R(y, z) \\ 0 & \text{otherwise.} \end{cases}
\]

The choice of “0 otherwise” is somewhat arbitrary. It is in fact even easier to recursively define the function \( m'_R \) which returns the least \( x < y \) such that \( R(x, z) \) holds, and \( y + 1 \) otherwise. When we use \( \min \), however, we will always know that the least \( x \) such that \( R(x, z) \) exists and is less than \( y \). Thus, in practice, we will not have to worry about the possibility that if \( \min x < y R(x, z) = 0 \) we do not know if that value indicates that \( R(0, z) \) or that for no \( x < y \), \( R(x, z) \). As with bounded quantification, \( \min x \leq y \ldots \) can be understood as \( \min x < y + 1 \ldots \).

**Problem rec.1.** Suppose \( R(x, z) \) is primitive recursive. Define the function \( m'_R(y, z) \) which returns the least \( x < y \) such that \( R(x, z) \) holds, if there is one, and \( y + 1 \) otherwise, by primitive recursion from \( \chi_R \).