Define \( \approx \) to give the deep contents of a string or set or list or function. Here are some examples.

\[
\approx(10; 11; 12; 13) = 10, 11, 12, 13 \\
\approx(10, \{11, 12\}, 13) = 10, 11, 12, 13 \\
\approx[10; [11; 12]; 13] = 10, 11, 12, 13 \\
\approx(10, [11; 12], 13) = 10, 11, 12, 13 \\
\approx[10; {11, 12}; 13] = 10, 11, 12, 13 \\
\approx(x: \text{nat} \to \langle y: \text{nat} \to 2 \times x \times y \rangle) = 2 \times \text{nat} \\
\approx(x: (0..4) \to \langle y: (0..x) \to x+y \rangle) = 1, 2, 3, 4, 5
\]

The contents operator \( \sim \) removes one level of structure from a set or list. The deep contents operator \( \approx \) removes all levels of structure.

§ It is convenient to start by defining \( \approx \) for base cases. If \( x \) is an element, and \( x \) is not a set, string, list, or function, then

\( \approx x = x \)

Now for bunches,

\( \approx \text{null} = \text{null} \)
\( \approx (A, B) = \approx A, \approx B \)

Now for sets,

\( \approx \{A\} = \approx A \)

For strings,

\( \approx \text{nil} = \text{null} \)
\( \approx (A; B) = \approx A, \approx B \)

For lists,

\( \approx [S] = \approx S \)

For functions,

\( \approx \langle v: D \to e \rangle = \langle v: D \to \approx e \rangle D \)

Or if you prefer,

\( \approx f = \langle v: \square f \to \approx f v \rangle (\square f) \)

For functions we could instead define \( \approx \) the way we usually define quantifiers.

\( \approx v: \text{null} \cdot e = \text{null} \)
\( \approx v: x \cdot e = \langle v: x \to \approx e \rangle x \)

for element \( x \)

\( \approx v: A \cdot B \cdot e = (\approx v: A \cdot e), (\approx v: B \cdot e) \)

\( \approx v: (\$v: D \cdot b) \cdot e = \langle v: D \to \text{if } b \text{ then } \approx e \text{ else } \text{null } \text{fi} \rangle D \)