Problem 1

Derive the axiom schemata

\[ m \leq j \leq n \Rightarrow \left( \bigcap_{i=m}^{n} p(i) \Rightarrow (p(j) \ast \text{true}) \right) \]

from the axiom schemata for iterating separating conjunction given earlier on the first page of Chapter 5 of the class notes.

Problem 2

If \( \tau \) is an S-expression, then \(|\tau|\), called the flattening of \( \tau \), is the sequence defined by:

\[ |a| = [a] \quad \text{when } a \text{ is an atom} \]

\[ |(t_1 \cdot t_2)| = |\tau_1| \cdot |\tau_2|. \]

Here \([a]\) denotes the sequence whose only element is \(a\), and the “\(\cdot\)” on the right of the last equation denotes the concatenation of sequences.

Define and prove correct (by an annotated specification of its body) a recursive procedure \texttt{flatten} that mutates a tree denoting an S-expression \( \tau \) into a singly-linked list segment denoting the flattening of \( \tau \). This procedure should not do any allocation or disposal of heap storage. However, since a list segment representing \(|\tau|\) contains one more two-cell than a tree representing \( \tau \), the procedure should be given as input, in addition to the tree representing \( \tau \), a single two-cell, which will become the initial cell of the list segment that is constructed.

More precisely, the procedure should satisfy

\[
\{ \text{tree } \tau \text{ (i) } \ast j \rightarrow -, - \} \\
\texttt{flatten}(i, j, k) \\
\{ \text{lsig } |\tau| (j, k) \}.
\]

(Note that \texttt{flatten} must not assign to the variables i, j, or k.)