rec.1 Trees

cmp:rec:tre:sec

Sometimes it is useful to represent trees as natural numbers, just like we can represent sequences by numbers and properties of and operations on them by primitive recursive relations and functions on their codes. We'll use sequences and their codes to do this. A tree can be either a single node (possibly with a label) or else a node (possibly with a label) connected to a number of subtrees. The node is called the *root* of the tree, and the subtrees it is connected to its *immediate subtrees*.

We code trees recursively as a sequence $\langle k, d_1, \ldots, d_k \rangle$, where k is the number of immediate subtrees and d_1, ..., d_k the codes of the immediate subtrees. If the nodes have labels, they can be included after the immediate subtrees. So a tree consisting just of a single node with label l would be coded by $\langle 0, l \rangle$, and a tree consisting of a root (labelled l_1) connected to two single nodes (labelled l_2, l_3) would be coded by $\langle 2, \langle 0, l_2 \rangle, \langle 0, l_3 \rangle, l_1 \rangle$.

cmp:rec:tre:prop:subtreeseq

Proposition rec.1. *The function* SubtreeSeq(t)*, which returns the code of a sequence the elements of which are the codes of all subtrees of the tree with code t, is primitive recursive.*

Proof. First note that $\text{ISubtrees}(t) = \text{subseq}(t, 1, (t)_0)$ is primitive recursive and returns the codes of the immediate subtrees of a tree t. Now we can define a helper function $\text{hSubtreeSeq}(t, n)$ which computes the sequence of all subtrees which are n nodes remove from the root. The sequence of subtrees of t which is 0 nodes removed from the root—in other words, begins at the root of t—is the sequence consisting just of t. To obtain a sequence of all level $n+1$ subtrees of t, we concatenate the level n subtrees with a sequence consisting of all immediate subtrees of the level n subtrees. To get a list of all these, note that if $f(x)$ is a primitive recursive function returning codes of sequences, then $g_f(s, k) = f((s)_0) \frown \ldots \frown f((s)_k)$ is also primivive recursive:

$$\begin{aligned} g(s, 0) &= f((s)_0) \\ g(s, k+1) &= g(s, k) \frown f((s)_{k+1}) \end{aligned}$$

For instance, if s is a sequence of trees, then $h(s) = g_{\text{ISubtrees}}(s, \text{len}(s))$ gives the sequence of the immediate subtrees of the elements of s. We can use it to define hSubtreeSeq by

$$\begin{aligned} \text{hSubtreeSeq}(t, 0) &= \langle t \rangle \\ \text{hSubtreeSeq}(t, n+1) &= \text{hSubtreeSeq}(t, n) \frown h(\text{hSubtree}(t, n)). \end{aligned}$$

The maximum level of subtrees in a tree coded by t, i.e., the maximum distance between the root and a leaf node, is bounded by the code t. So a sequence of codes of all subtrees of the tree coded by t is given by $\text{hSubtreeSeq}(t, t)$. □

Problem rec.1. The definition of hSubtreeSeq in the proof of Proposition rec.1 in general includes repetitions. Give an alternative definition which guarantees that the code of a subtree occurs only once in the resulting list.

Photo Credits

Bibliography