A generalization of the functions head and tail.

by Edsger W. Dijkstra and C. S. Scholten

We consider sequences defined as structures on the natural coordinate \( x \). Let \( S \) be such a sequence. The functions \( h(=\text{head}) \) and \( t(=\text{tail}) \) are defined by

\[
   h.S = S_0^x \quad \text{and} \quad t.S = S_{1+x}^x
\]

Note that \( h.S \) is an "element" — viz. the "leading" one — and \( t.S \) is again a sequence — viz. "the rest". (With the : for concatenation — as, for instance in SASL — we have the identity \( S = h.S : t.S \).)

There are several ways of expressing \( S'_{\text{sub}}S \), i.e. that \( S' \) is a postfix of \( S \):

\[
   S'_{\text{sub}}S = (\text{En: } n \geq 0: \ S' = t^n.S) \quad \text{or} \quad S'_{\text{sub}}S = (\text{En: } n \geq 0: \ S' = S_{n+x}^x)
\]

We prefer the latter one. Representing the natural number \( n \) by a string of \( n \) zeros, and hence addition by juxtaposition, we would get

\[
   S'_{\text{sub}}S = (\text{En: } n \in \mathbb{N}^*: \ S' = S_{n+x}^x)
\]

The above is extended to tuples of sequences. Illustrating it for two we thus define

\[
   (S',T')_{\text{sub}}(S,T) \equiv (\text{En: } n \in \mathbb{N}^*: \ (S',T') = (S,T)_{n+x})
\]

Substitution being defined to distribute over pair forming.
the quantified expression may be rewritten as
\[(S', T') = (S^x_{nx}, T^x_{nx})\]

with element-wise application of equality this yields
\[S' = S^x_{nx} \land T' = T^x_{nx}\]

To complete the understanding of the above we define for sequences \(S\) and \(T\) equality by
\[S = T \equiv h.S = h.T \land t.S = t.T\]

We mention without proof
\[S = T \equiv (A(S', T'): (S', T')_{sub}(S, T): h.S' = h.T')\]
(The proof is left as an exercise for the authors.)

A sequence is a special instance of a rooted tree with constant fan-out, viz. with \(\text{fan-out} = 1\), in exactly the same way as \(\text{f0}_3\) is a special case of a finite alphabet. In the following, \(C\) stands for an alphabet of \(m\) characters, our tuples will be \(m\)-tuples and our trees trees with constant fan-out = \(m\).

We now consider a tree as a structure defined on a coordinate \(x\) ranging over \(C^*\). Let \(S\) be such a tree. The function head has its obvious analogue: it is known under the name root, and we shall denote it by \(r\) and define it by
In which $\epsilon$ denotes the empty string.

The corresponding notion $\text{sub}$, however, poses a problem. Do we define

$$S' \text{ sub } S \equiv (\text{En: } n \in C^*: S' = S_{n\times}^x)$$

or

$$S' \text{ sub } S \equiv (\text{En: } n \in C^*: S' = S_{\times n}^x)$$

Note In either case we have the theorem—mentioned without proof—that for trees $S$ and $T$

$$S = T \equiv (\forall S', T': (S', T') \text{ sub } (S, T): r. S' = r.T')$$

(End of Note.)

In this stage we have no grounds for preferring the one sub over the other. (For a single character alphabet, the two definitions coincide.)

For $m \geq 2$, we have two different ways of defining under control of a parameter $c$, $c \in C^*$, a new tree in terms of a given one:

$$c \text{ ex } S = S_{c\times}^x$$

$$S \text{ ex } c = S_{\times c}^x$$

Here we have used the same operator $\text{ex}$ as an asymmetric infix operator between a tree and an element of $C^*$. 
With $b \in C^*$ and $c \in C^*$ we then have
\[
c \text{ ex } (b \text{ ex } S) = (bc) \text{ ex } S
\]
\[
(S \text{ ex } b) \text{ ex } c = S \text{ ex } (cb)
\]
note that on account of the types of $b, c,$ and $S,$ the parentheses in the left-hand sides of the above could have been omitted.

Furthermore, we have
\[
b \text{ ex } (S \text{ ex } c) = (b \text{ ex } S) \text{ ex } c
\]
both sides being equal to $S_{bx}^*$. Consequently, also here the parentheses may be omitted. We conclude that the "continued" $\text{ ex }$ of which 1 operand is a tree while the others are from $C^*$ needs no parentheses.

Finally, note that $\langle \rangle \text{ ex }$ and $\text{ ex } \langle \rangle$ are identity operators. Note also
\[
r.(c \text{ ex } S) = r.(S \text{ ex } c)
\]
So much for the general $\text{ ex }$.

*       *
*       *

Of special interest is the use of $\text{ ex }$ with the string operand of length 1. Let $e$ be a parameter ranging over the strings of length 1 in $C^*$ or - if we don't distinguish between one-element strings and elements - ranging over $C$; $e$ has $m$ distinct
possible values. For such e ,

\[ e \in S \text{ is called "son tree node of } S\] \text{ and } S e \in e \] \text{ is called "daughter tree node of } S\].

These are the closest analogue of the function tail: firstly it has an additional parameter ranging over \( C \), secondly there is the distinction between sons and daughters. The latter distinction gives us two alternative recursive definitions for the equality of two trees \( S \) and \( T \):

\[
S = T \equiv r. S = r. T \land (\forall e : e \in C : e \in S = e \in T) \\
S = T \equiv r. S = r. T \land (\forall e : e \in C : S e \in e = T e \in e) 
\]

After \( \in \) we turn our attention to a number of unary operators, to begin with some that form a new tree from a given one.

Consider the function \( \text{rev} \) on strings, with \( b \in C^* \), \( c \in C^* \), and \( e \in C \) given by

\[
\text{rev} . <> = <> \\
\text{rev} . e = e \\
\text{rev} . (bc) = (\text{rev} . c)(\text{rev} . b)
\]

In terms of \( \text{rev} \) we now define the "transpose"

\[ S^T = S_{\text{rev} . x} \]

Since \( \text{rev} . (\text{rev} . x) = x \), \( (S^T)^T = S \). The connection between the transpose and \( \in \) is given by
\((b \in S)^T = S^T \in (\text{rev}.b)\), and in particular
\((e \in S)^T = S^T \in e\).

For our purposes, the transpose is not a very important operator; it has been mentioned because it illustrates an underlying duality so nicely.

For the sake of completeness we also mention \(\text{ROT}\) defined by
\[ \text{ROT}.S = S_{\text{rot}^x}^x \]
where \(\text{rot}.<> = <>\)
\(\text{rot}.(e \cdot b) = be\).

This is a function in which we are even less interested than in the transpose. This is because our interest in such infinite trees, i.e. functions on \(C^*\), stems from considerations about recursion, which relate elements of \(C^*\) with, say, a common prefix, a relation which is completely destroyed by \(\text{rot}\). (So we hardly take the trouble to observe
\[ e \in (\text{ROT}.S) = S \in e \) .

Now we return to \(e \in S\); it is again a tree, comprising, so to speak, \(1/m\)-th of the elements of \(S\) minus its root. Let now \(e\) range over \(C\); the combined elements of the resulting \(m\) trees comprise all the elements of \(S\) except the root: it can be viewed, therefore, as a function on \(C^*\), i.e. all non-empty finite strings of \(C^*\). We can denote
the aggregate of the son trees of $S$ by $s.S$ and