ELSEVIER

Information Processing Letters 81 (2002) 93–96

Information Processing Letters

www.elsevier.com/locate/ipl

On computing a longest path in a tree

Eindhoven Tuesday Afternoon Club

R.W. Bulterman, F.W. van der Sommen, G. Zwaan, T. Verhoeff, A.J.M. van Gasteren, W.H.J. Feijen *

Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Received 27 November 2000; received in revised form 6 February 2001
Communicated by F.B. Schneider

Keywords: Program derivation; Calculational reasoning

1. Introduction

The primary purpose of this note is to present an exercise in proof design. For us, such a design consists in isolating the relevant concepts for the problem at hand, introducing special-purpose notation for them that is geared to manipulation and to crisp formal specification, and then solving the problem in a demand-driven way, while on-the-fly extracting from the calculation additional theory useful for solving the problem proper.

The problem chosen is demonstrating the correctness of an algorithm for computing the longest path in a tree.

Given a finite tree with all edges having positive length, we wish to compute a longest path. This can be done using a procedure invented by Edsger W. Dijkstra around 1960, which is as follows.

Build a physical model of the tree by connecting each pair of adjacent nodes by a piece of string of the given edge length. Now pick up the physical tree at an arbitrary node U, let the contraption hang down, and determine a deepest node X. Then pick up the tree at X and determine a deepest node Y. The claim is that the path between X and Y is a longest path in the tree.

We have never seen a formal proof of this claim, and the purpose of this note is to provide one.

2. Formal specification

A tree provides a unique connection for each pair of nodes; we refer to this connection as *the path* between these nodes. The length of the path between nodes v and w is denoted by vw: it is the sum of the lengths of the edges on the path. Thus for all nodes v, w we have $vw = wv$.

Now we can formally state the problem. For v, w, z ranging over arbitrary nodes, we can summarize the procedure as follows:

* Corresponding author.
E-mail address: wf@win.tue.nl (W.H.J. Feijen).

PII: S0020-0190(01)00198-3

(0) $\langle \forall z :: Uz \leqslant UX \rangle$

(1) $\langle \forall z :: Xz \leqslant XY \rangle$,

and then the claim is

$$\langle \forall v, w :: vw \leqslant XY \rangle.$$

Because the problem is entirely phrased in terms of path lengths, we will at least need a little "theory of trees" stating something about path lengths. The most elementary properties that spring to mind are the triangular properties

$$\Delta_{\leqslant} : \quad vw \leqslant vm + mw, \quad \text{and}$$

$$\Delta_{=} : \quad m \text{ on } vw \quad \equiv \quad vw = vm + mw$$

in which "m on vw" is a shorthand for "node m is on the path connecting nodes v and w, endpoints included". (In $\Delta_{\leqslant}$ we use that all edges have nonnegative length; for $\Delta_{=}$ we need that all lengths are positive.)

3. The formal proof

We now design a calculational proof, interspersed with some heuristic remarks. For any v and w we have

$vw \leqslant XY$

$\Leftarrow$ { by (1) and transitivity of $\leqslant$,
apart from which there is not much else we can do }

$\langle \exists z :: vw \leqslant Xz \rangle$

$\Leftarrow$ { looking for witnesses, we try to restrict ourselves to
the nodes identified so far, viz. v, w, U, X, and Y;
of these, X makes no sense and Y is absorbed
by the demonstrandum }

$vw \leqslant Xv \quad \vee \quad vw \leqslant Xw \quad \vee \quad vw \leqslant XU$

$\Leftarrow$ { it is unlikely that disjunct $vw \leqslant XU$ could
contribute to the validity of this expression:
node U is arbitrary and vw could be
the length of a longest path }

(∗) $vw \leqslant Xv \quad \vee \quad vw \leqslant Xw$.

And here we are left with an expression that is symmetric in v and w, so that we can afford to focus on one disjunct only.

Neither of our givens (0) and (1) apply to $vw \leqslant Xv$, and at this point we thus fall short of manipulative freedom. Additional freedom is usually obtained by parametrization, and it is here that $\Delta_{\leqslant}$ and $\Delta_{=}$ from our little "theory of trees" come in handy, since these can introduce new nodes. By applying $\Delta_{\leqslant}$ to vw and $\Delta_{=}$ to Xv —the least committing choice for strengthening $vw \leqslant Xv$!—, we obtain

$vw \leqslant Xv$

$\Leftarrow$ { introduction of nodes m and n,
• n on Xv }

$vm + mw \leqslant Xn + nv$,

where $\bullet$ is to be read as “on the premise that”. But if we have to proceed from here, we had better decide on $m = n$, because the terms vm and nv then cancel.

With these considerations in mind, our calculation continues as follows:

$$
\begin{array}{ll}
 & vw \leqslant Xv \\
\Leftarrow & \quad \{\ \Delta_{\leqslant} \text{ applied to } vw\text{, using transitivity of } \leqslant; \\
 & \quad\ \ \Delta_{=} \text{ applied to } Xv, \\
 & \quad\ \ \bullet\ m \text{ on } Xv \ \ \} \\
 & vm + mw \leqslant Xm + mv \\
\equiv & \quad \{\ \text{algebra, using } vm = mv \ \ \} \\
 & mw \leqslant Xm \\
\equiv & \quad \{\ \text{introduction of } U\text{, heading for (0),} \\
 & \quad\ \ \text{which has not been used yet } \} \\
 & Um + mw \leqslant Um + Xm \\
\Leftarrow & \quad \{\ \Delta_{\leqslant} \text{ applied to right-hand side;} \\
 & \quad\ \ \Delta_{=} \text{ applied to left-hand side,} \\
 & \quad\ \ \bullet\ m \text{ on } Uw \ \} \\
 & Uw \leqslant UX \\
\equiv & \quad \{\ \text{(0) with } z := w \ \} \\
 & \text{true .}
\end{array}
$$

In summary, we established

$$vw \leqslant Xv \quad \Leftarrow \quad \langle \exists m :: (m \text{ on } Xv) \wedge (m \text{ on } Uw) \rangle \,.$$

By the symmetry between v and w, we have for the other disjunct of $(*)$

$$vw \leqslant Xw \quad \Leftarrow \quad \langle \exists m :: (m \text{ on } Xw) \wedge (m \text{ on } Uv) \rangle \,.$$

So $(*)$, and hence the theorem, has been proved whenever we can rely on

$$
\begin{array}{l}
\langle \exists m :: (m \text{ on } Xv) \wedge (m \text{ on } Uw) \rangle \\
\vee \\
\langle \exists m :: (m \text{ on } Xw) \wedge (m \text{ on } Uv) \rangle \,,
\end{array}
$$

and it so happens that this is an instance of the general property of trees that

$$
\begin{array}{ll}
(\heartsuit) & \text{for all nodes } A,\ B,\ C, \text{ and } D \text{ of a tree,} \\
 & \langle \exists m :: (m \text{ on } AB) \wedge (m \text{ on } CD) \rangle \\
 & \vee \\
 & \langle \exists m :: (m \text{ on } AD) \wedge (m \text{ on } CB) \rangle \,,
\end{array}
$$

a property that we were unaware of when we started the exercise. (In contrast to the theorem of this paper, which has an arithmetic flavor, property $(\heartsuit)$ is a topological theorem. It can be proven along the following line: either AB and CD overlap —in which case any node in the overlap is a witness for the first disjunct—, or AB and CD are node-disjoint —in which case there is a unique path connecting AB and CD each node of which is a witness for the second disjunct.)

4. In conclusion

The calculation proper consists of eight small calculational steps and apart from the exploitation of the symmetry between v and w, there is no case analysis involved. Nevertheless, it took us a very long time to arrive at the above argument. It was only after we banished the use of pictures, both on the blackboard *and* in our minds, that our design started to converge to the above one. And only then it became clear, in a demand-driven way, what properties of trees were useful to solve the problem proper. Property ($\heartsuit$) in particular came as a surprise, and we could never have predicted its usefulness in advance, since we didn't even realize its existence.

We believe that without our calculational style we could never have arrived at the above design, in which the algorithm's two ingredients (0) and (1) are each used exactly once.

Acknowledgements

We are grateful for the constructive and pleasant comments provided by two anonymous referees.