This note deals with a problem posed to me by Apt when I visited him in Amsterdam. I mentioned the problem at the next session of the ETAC, where we did not solve it; Hoogerwoord, however, had suggested a structure of the argument and had provided a major building block. The next day, Voermans provided the missing ingredient and completed the proof.

*   *   *

We consider an expression built according to the following syntax:

\[ \text{<exp>} ::= \text{<atom>} \mid \neg \text{<exp>} \]

\[ \mid \vee \text{<exp>} \text{<exp>} \mid \wedge \text{<exp>} \text{<exp>} \]

We are given the following rewrite rules:

1. \( \neg x \rightarrow x \)
2. \( \neg \neg x \rightarrow x \)
3. \( \neg x \neg y \rightarrow x \wedge y \)
4. \( \wedge x \wedge y \rightarrow x \wedge y \)
5. \( \vee x \vee y \rightarrow \neg \neg x \vee y \)
6. \( x \vee y \vee z \rightarrow \neg \neg x \vee \neg \neg y \vee \neg \neg z \)
7. \( \neg x \neg y \neg z \rightarrow \neg \neg x \neg y \neg z \)

in which \( x, y, z \) are subexpressions of type \( \text{<exp>} \). Our algorithm consists of repeatedly replacing a subexpression matching the left-hand side of a rewrite rule by the corresponding
right-hand side of that rule. The challenge is to prove that the algorithm terminates because, sooner or later, none of the rewrite rules is applicable any more.

* * *

We try to define a function \( f \) from expressions to natural numbers — or, if the need arises, to a more general well-founded domain — such that the function value decreases when a rewrite rule is applied to the argument. Our first decision — taken for the sake of simplicity — is to define \( f \) recursively over the syntax, i.e. besides defining \( f.\langle \text{atom} \rangle = \text{const} \)
we define
\[
\begin{align*}
f.\langle \text{\( \neg \) x} \rangle &= \text{neg.}(f.x) \\
f.\langle \text{dis.} xy \rangle &= \text{dis.}(f.x, f.y) \\
f.\langle \text{con.} xy \rangle &= \text{con.}(f.x, f.y)
\end{align*}
\]

the challenge is now to define \( \text{const, neg, dis, and con} \) in such a way that application of rewrite rules leads to a decrease of \( f \). Since rewrite rules can be applied to replace subexpressions, whereas the \( f \)-value of the whole expression has to decrease, the functions \( \text{neg, dis, and con} \) have to be strongly monotonic in all their arguments, i.e.
(5) \( p > p' \land q > q' \Rightarrow \neg p > \neg p' \land \)
\[ \text{dis.}(p,q) > \text{dis.}(p',q) \land \text{dis.}(p,q) > \text{dis.}(p',q') \land \text{con.}(p,q) > \text{con.}(p',q) \land \text{con.}(p,q) > \text{con.}(p',q') \]

To begin with we focus our attention on (3) and (4), which describe how \( \land \) distributes over \( \lor \), viz. in the same way as \( \cdot \) (times) distributes over \( + \) (plus). This last observation suggests to choose for \( \text{dis} \) and \( \text{con} \) something like \( \text{dis.}(p,q) = p + q \) and \( \text{con.}(p,q) = p \cdot q \); I wrote "something like" because the above choice would leave the \( f \)-value under rewritings (3) and (4) unchanged. Let us investigate with the requirement that (3) leads to a decrease of \( f \):

\[
\begin{align*}
&f.(\land x y z) > f.(\lor x y z) \\
&= \{ \text{def. of con ; def. of dis} \} \\
&f.x \cdot f.(y z) - c > f.(\lor x y z) + f.(\land x z) \\
&= \{ \text{def. of dis ; def. of con} \} \\
&f.x \cdot (f.y + f.z) - c > f.x \cdot f.y - c + f.x \cdot f.z - c \\
&= \{ \text{algebra} \} \\
&c > 0 
\end{align*}
\]

The requirement that application of (4) leads to an \( f \)-decrease is equivalent to the same \( c > 0 \). In order to ensure that \( f \)-values are natural we choose a natural const satisfying
\[ \text{const}^2 - c \geq \text{const}. \]

These conditions can be satisfied, e.g. by \(c=1\) and \(\text{const} = 2\). We are left with the obligation of constructing a \(\text{neg}\), such that rewrites (0), (1) and (2) decrease \(p\).

From (0) we conclude the requirement
\[ (6) \quad \text{neg.} \,(\text{neg.} \,p) > p \]

From (1) we conclude the requirement
\[ (7) \quad \text{neg.} \,(p+q) > \text{neg.} \,p \times \text{neg.} \,q - c \]

From (2) we conclude the requirement
\[ (8) \quad \text{neg.} \,(p \times q - c) > \text{neg.} \,p + \text{neg.} \,q \]

These three requirements should be satisfied for \(p \geq \text{const}\) and \(q \geq \text{const}\), \(\text{const}\) being the minimum \(p\)-value.

(6) is satisfied if \(\text{neg.} \,p > p\); for monotonic \(\text{neg}\) satisfying \(\text{neg.} \,p > p\), (8) is unlikely to present problems for larger arguments; (7) imposes a clear constraint, but since \(c > 0\), \(\text{neg.} \,p = d^p\) satisfies (7). For \(c = 1\) and \(\text{const} = 2\), \(d=2\) is too small — since \(2^{2\times2-1} = 2^2 + 2^1\), (8) can be violated —; \(d=3\), however, does the job. In short
\[ f. (<\text{atom}> ) = 2 \]
\[ f. (\text{\texttt{1} x} ) = 3 f. x \]
\[ f. (\text{\texttt{v} x y} ) = f. x + f. y \]
\[ f. (\text{\texttt{a} x y} ) = f. x \times f. y - 1 \]
is a witness demonstrating the existence of a variant function. Another witness is given by $c, \text{const}, d = 1, 3, 2$.

*   *   *

At the ETAC, I got stuck. I started with (0), (1) & (2), mapping the latter two on each other by ignoring the difference between $\land$ and $\lor$; subsequently introducing the ignored difference by taking (3) & (4) into account gave serious problems. The advantage of starting with (3) & (4) is that then $T$ is ignored automatically.

The reader is asked to realize how much the derivation has been eased by the introduction of the named functions $\text{const}$, $\text{neg}$, $\text{dis}$, & $\text{con}$.

Nuenen, 29 August 1991

prof. dr. Edsger W. Dijkstra
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-1188
USA