Chapter 33 : Searching by Elimination.

In which we introduce a simple new search technique.

We are given a finite set W and a boolean function F defined on the elements of W. We are asked to construct a program to meet the following specification.

Pre : $\langle \exists\ w : w \in W : F.w \rangle$

Post : F.x

We begin by defining

* (0) P.V $=$ $\langle \exists\ v : v \in V : F.v \rangle$, $V \subseteq W$

We can now rewrite the specification as

Pre : P.W

Post : P.{x}

We introduce a set variable V and propose the following invariants

P0 : P.V
P1 : $V \subseteq W$

This is easily established by

V := W

We will choose the size of V as the variant which is written #.V

This leads us to the following program skeleton.

```
V := W {P0 ∧ P1}
;do #.V ≠ 1    →    {P0 ∧ P1 ∧ #.V ≠ 1}

                    "Decrease #.V but maintain P0 ∧ P1"

                    {P0 ∧ P1}
od
{P0 ∧ P1 ∧ #.V=1}
x := "the unique elements in V"
```

Constructing the loop body.

As W is not the empty set, $P1 \wedge \#.V \neq 1$ leads us to conclude that $2 \leq \#.V$. Given that there are at least 2 elements in V we should be able to remove one of them without violating P0.

We propose the following structure. Note that α and β are as yet unknown guards.

```
V := W
;do #.V ≠ 1→ {P0 ∧ P1 ∧ #.V ≠ 1}

          "choose a, b ε V, where a ≠ b"
          if α → V := V \ {a}
          [] β → V := V \ {b}
          fi

          {P0 ∧ P1}
od
x := "the unique element in V"
```

We must now determine α and β. Let us concentrate on α. The following must hold

$$\alpha \wedge (a \in V) \wedge (b \in V) \wedge (a \neq b) \wedge P.V \Rightarrow P.(V\backslash\{a\})$$

We calculate

$$
\begin{array}{ll}
 & P.V \Rightarrow P.(V\backslash\{a\}) \\
= & \qquad \{\text{definition of P, split off F.a, a } \varepsilon \text{ V}\} \\
 & F.a \vee P.(V\backslash\{a\}) \Rightarrow P.(V\backslash\{a\}) \\
= & \qquad \{P \vee Q \Rightarrow Q \equiv P \Rightarrow Q\} \\
 & F.a \Rightarrow P.(V\backslash\{a\}) \\
= & \qquad \{\text{split off b, b } \varepsilon \text{ } P.(V\backslash\{a\})\} \\
 & F.a \Rightarrow F.b \vee P.(V\backslash\{a,b\}) \\
\Leftarrow & \qquad \{ (P \Rightarrow Q) \Rightarrow (P \Rightarrow (Q \vee R))\} \\
 & F.a \Rightarrow F.b
\end{array}
$$

So a candidate for α is $F.a \Rightarrow F.b$

Symmetrically, a candidate for β is $F.b \Rightarrow F.a$

Thus we have arrived at the program

```
V := W
;do #.V ≠ 1→ {P0 ∧ P1 ∧ #.V ≠ 1}

            "choose a, b ε V, where a ≠ b"
            if F.a ⇒ F.b → V := V \ {a}
            [] F.b ⇒ F.a → V := V \ {b}
            fi

            {P0 ∧ P1}
od
x := "the unique element in V"
```

This is the generic Searching by Elimination algorithm.