Loop Termination

Suppose we have a **while** loop of the form

```
while cond do
        S
endwhile
```

Our concern is proving that the loop terminates. Obviously we could do that if we can show that eventually the loop condition cond is eventually falsified. At a high level, everything discussed here is to that end.

Let's begin with a very simple example:

```
while i ≤ 10 do
        sum := sum+v[i]
        i := i+1
endwhile
```

Notice that I have no preconditions stated at all. In particular, I do not care about the initial value of *i* nor its original relation to *10*. What I informally notice is that

- the expression *10-i* is going to be decreasing
- the loop terminate if the expression *10-i* is negative.

Next, I actually prove the first claim.

```
while i ≤ 10 do
        sum := sum+v[I]
        i := i+1_____________________________ i = i'+1
                 _____________________________ 10 − i < 10 − i'
endwhile
```

Notice that this was done without needing any assertions at all. The postcondition $i = i'+1$ follows from the previous assignment, and $10-i<10-i'$ follows from that. If I recognize that the loop condition $i \le 10$ is equivalent to $0 \le 10-i$, then I can state:

> The quantity $10-i$ is an integer and strictly decreases with each iteration of the loop. Therefore, eventually $10-i$ is negative, thus $i>10$, and the loop terminates.

Lastly, a few more words on irrelevancies. My termination proof says nothing about the number of steps through the loop, the final value of i, or anything about a postcondition to the loop. It speaks only to one issue: termination. Also, the key to the proof was identifying the critical expression $10-i$. To be sure you understand, alter the loop condition to $i<10$ and the assignment to i := i+2, and see what I have said needs to be changed.

Many termination proofs will have exactly this character: some expression decreases and the loop condition is falsified when the expression passes some critical value.

As a slightly trickier example, let's try to prove termination of:

$0 < t$

while $even(t) \wedge (0 < t)$ **do**
    t := t/2
endwhile

First, I will verify a loop invariant $0 < t$:

$0 < t$

while $even(t) \wedge (0 < t)$ **do** $even(t) \wedge (0 < t)$

$2 \le t$

t := t/2 $(t = t'/2) \wedge (2 \le t')$

$0 < t$

endwhile

Next, I assert that

$0 < t$

t := t/2 $t < t'$

is correct, so that the sequence of values of t , is positive and strictly decreasing. Any such sequence must be finite. Again we argue that the loop condition is eventually falsified because it there does not exist an infinite sequence of positive, decreasing integers.

Lastly, consider the termination of:

$\mathcal{A}$ *is finite*

while $nonempty(\mathcal{A})$ **do**
    A := remove(A, min(A))
endwhile

We will assume we have already verified that

$(\mathcal{A} \neq \varnothing) \wedge (\mathcal{A}$ *is finite*$)$ {b := min(A)} $b = \min(\mathcal{A})$

and

$(\mathcal{A} \neq \varnothing) \wedge (b \in \mathcal{A})$ {A := remove(A, b)} $\mathcal{A} \underset{\neq}{\subset} \mathcal{A}'$.

First, I will verify a loop invariant $A \; is \; finite$:

$A \; is \; finite$

while $nonempty(A)$ **do** $(A \neq \varnothing) \wedge (A \; is \; finite)$

A := remove(A, min(A)) $A \underset{\neq}{\subset} A' \wedge (A' \; is \; finite)$

$A \; is \; finite$

endwhile

Notice, that concurrently I showed that $A \underset{\neq}{\subset} A'$. I conclude that the sequence of finite sets A is strictly decreasing in cardinality. We argue that the loop condition is eventually falsified because there does not exist an infinite sequence of finite sets with decreasing cardinality.