p7q6
MJCG

Specification and Verification I 2003

SV1: Solution Notes

This question pertains to the "Program verification" part of the syllabus.

(a) Explain the difference between a *variant* and an *invariant*. Briefly describe what they are used for.

A variant is a non-negative expression that strictly decreases each time around a loop. [1 mark] It is used to show termination. [1 mark] An invariant is a formula whost truth is preserved by the body of a loop. [1 mark] It is used to establish a postcondition. [1 mark]

(b) State and justify the verification conditions for the total correctness of `WHILE` commands.

A correctly annotated total correctness specification of a `WHILE`-command has the form `[`P`] WHILE` S `DO` $\{R\}$`[`E`]` C `[`Q`]`, where R is an invariant and E a variant. The verifications conditions are:

1 $P \Rightarrow R$ [1 mark]

2 $R \wedge \neg S \Rightarrow Q$ [1 mark]

3 $R \wedge S \Rightarrow E \geq 0$ [1 mark]

4 the (recursively generated) verification conditions for $[R \wedge S \wedge (E = n)]\ C\ [R \wedge (E < n)]$, where n is an auxiliary variable not occuring elsewhere. [1 mark]

If 4 holds, inductively $\vdash [R \wedge S \wedge (E = n)]\ C\ [R \wedge (E < n)]$, hence by 3 and the `WHILE`-rule for total correctness $\vdash$ [R] `WHILE` S `DO` C [$R \wedge \neg S$], hence by 1 + Precondition Strengthening and 2 + Postcondition Weakening it follows that $\vdash$ [P] `WHILE` S `DO` C [Q]. [2 marks]

(c) Devise a precondition P that makes the following specification true.

```
[P]
 WHILE I≤N DO SUM := SUM+(2×I); I := I+1
[SUM = N×(N-1)]
```

It is sufficient to take P to be `SUM=0` $\wedge$ `I=1` $\wedge$ `N≥0`.
This is justified below.

Devise and justify annotations for this specifications that yield provable verification conditions.

Here is an annotated total correctness specification (incorporating P as defined above):

```
[SUM=0 ∧ I=1 ∧ N≥0]
 WHILE I≤N
  DO{SUM = I×(I-1) ∧ I ≤ N+1}[(N+1)-I]
    SUM := SUM+(2×I); I := I+1
[SUM = N×(N+1)]
```

The verifications conditions are:

1 `SUM=0 ∧ I=1 ∧ N≥0 ⇒ SUM=I×(I-1) ∧ I≤(N+1)`

2 `SUM=I×(I-1) ∧ I≤(N+1) ∧ ¬(I≤N) ⇒ SUM=N×(N+1)`

3 `SUM=I×(I-1) ∧ I≤(N+1) ∧ I≤N ⇒ ((N+1)-I)≥0`

4 the (recursively generated) verification conditions for

```
[SUM=I×(I-1) ∧ I≤(N+1) ∧ I≤N ∧ ((N+1)-I=n)]
  SUM := SUM+(2×I); I := I+1
[SUM=I×(I-1) ∧ I≤(N+1) ∧ ((N+1)-I<n)]
```

The verification conditions 1, 2 and 3 are clearly true. The verification condition for 4 is the verification condition for

```
[SUM=I×(I-1) ∧ I≤(N+1) ∧ I≤N ∧ ((N+1)-I=n)]
  SUM := SUM+(2×I)
[SUM=(I+1)×((I+1)-1) ∧ (I+1)≤(N+1) ∧ ((N+1)-(I+1)<n)]
```

which, after simplifying, is the verification condition for

```
[SUM=I×(I-1) ∧ I≤(N+1) ∧ I≤N ∧ ((N+1)-I=n)]
  SUM := SUM+(2×I)
[SUM=(I+1)×I ∧ I≤N ∧ (N-I)<n)]
```

which is

```
SUM=I×(I-1) ∧ I≤(N+1) ∧ I≤N ∧ ((N+1)-I=n)
⇒
SUM+(2×I)=(I+1)×I ∧ I≤N ∧ (N-I<n)
```

Which is true.