

Φροντιστήριο 9, 13/11/19

Άσκηση 1

Να αποδείξετε την ορθότητα της προδιαγραφής $\models_{\text{par}} \{\text{true}\} \text{Mult } \{z = x \cdot y\}$ όπου ο κώδικας του Mult δίνεται πιο κάτω.

```
a := 0;
z := 0;
while (a != y) {
    z := z+x;
    a := a+1;
}
```

Ισχύει η προδιαγραφή και με την έννοια της ολικής ορθότητας;

Άσκηση 2

Να αποδείξετε την ορθότητα της προδιαγραφής $\models_{\text{tot}} \{x \geq 0\} \text{Copy1 } \{x=y\}$ όπου ο κώδικας του Copy1 δίνεται πιο κάτω.

```
a := x;
y := 0;
while (a != 0) {
    y := y+1;
    a := a-1;
}
```

Άσκηση 3

Να αποδείξετε την ορθότητα της προδιαγραφής $\models_{\text{tot}} \{x \geq 0\} \text{Copy2 } \{y=x\}$ όπου ο κώδικας του Copy2 δίνεται πιο κάτω.

```
y := 0;
while (y != x) {
    y := y+1;
}
```

Πως θα άλλαζε η απόδειξη αν είχαμε ως συνθήκη εκτέλεσης του while-loop την “ $y < x$ ”;

Άσκηση 4

Να αποδείξετε την ολική ορθότητα της προδιαγραφής $\models_{\text{tot}} \{n>0\} P \ \{ \forall j, 0 \leq j < n, A[k] \leq A[j] \}$ όπου ο κώδικας του P δίνεται πιο κάτω.

```
k := 0;
i := 1;
while (i < n) {
    if (A[i] < A[k])
        k := i;
    else
        k := k;
    i := i+1;
}
```

Άσκηση 5

Η εντολή `for` είναι μια χρήσιμη εντολή που παρουσιάζεται σε διάφορες διαδικαστικές γλώσσες προγραμματισμού. Στο πιο κάτω παράδειγμα την βλέπουμε να χρησιμοποιείται σε ένα πρόγραμμα που παράγει το άθροισμα των στοιχείων ενός πίνακα.

```
s = 0;
for (i := 0; i <= max; i := i+1) {
    s := s + a[i];
}
```

Να εξηγήσετε πως η εντολή `for(C1;B;C2){C3}` μπορεί να ορισθεί χρησιμοποιώντας εντολές της γλώσσας `WHILE`.

Χρησιμοποιώντας την πιο πάνω μετατροπή, να προτείνετε κανόνα για την ανάλυση προδιαγραφών της εντολής `for` υπό την έννοια της μερικής ορθότητας.