

Φροντιστήριο 1

Βασικές Έννοιες

Μοντελοποίηση Συστημάτων

Γράφος προγράμματος: πλειάδα (V, S, T, ρ) όπου

- V είναι κάποιο πεπερασμένο σύνολο μεταβλητών, $V = \{V_0, V_1, \dots, V_n\}$, $n \in \mathbb{N}$
- S είναι το σύνολο των καταστάσεων (ανάθεση τιμών στις μεταβλητές)
- T είναι κάποιο σύνολο μεταβάσεων της μορφής $e \rightarrow t$
όπου e είναι η συνθήκη και t το βήμα της μετάβασης.
(αν η κατάσταση s ικανοποιεί τη συνθήκη e τότε μπορεί να γίνει η μετάβαση
η οποία οδηγεί στην κατάσταση που επάγεται από την s μέσω του βήματος t)
- ρ είναι η αρχική συνθήκη

Σημασιολογικά μοντέλα

1. Σημασιολογία παρεμβαλλόμενης διάταξης
 - ολική διάταξη ανάμεσα στα γεγονότα
 - ο ταυτοχρονισμός μοντελοποιείται μέσω του μη-ντετερμινισμού
2. Σημασιολογία μερικής διάταξης
 - μερική διάταξη ανάμεσα στα γεγονότα
 - διαχωρίζεται ο ταυτοχρονισμός από τον μη-ντετερμινισμό

Άσκηση 1

Θεωρήστε το πιο κάτω πρωτόκολλο το οποίο επιχειρεί να πετύχει αμοιβαίο αποκλεισμό ανάμεσα σε δύο παράλληλες διεργασίες.

```
Bool T[2] = [0,0];

while True do {
    T[0] = 1;
    while T[1] = 1 do { /* nothing */; }
    CRITICAL;
    T[0] = 0;
}
||
while True do {
    T[1] = 1;
    while T[0] = 1 do { /* nothing */; }
    CRITICAL;
    T[1] = 0;
}
```

(α) Μοντελοποιήστε το πρωτόκολλο ως ένα γράφο προγράμματος.

(β) Αποδώστε τον γράφο προγράμματος του πρωτοκόλλου γραφικά ως ένα σύστημα μεταβάσεων.

(γ) Συζητήστε την καταλληλότητα του πρωτοκόλλου για διασφάλιση αμοιβαίου αποκλεισμού.

Άσκηση 2

Το πιο κάτω πρόγραμμα υπολογίζει την τιμή του

$$\binom{n}{k} = \frac{n \times (n-1) \times \dots \times (n-k+1)}{1 \times 2 \times \dots \times k}$$

χρησιμοποιώντας δύο παράλληλες διεργασίες. Να μοντελοποιήσετε το πρόγραμμα με σημασιολογία παρεμβαλλόμενης διάταξης. Μπορείτε να υποθέσετε ότι όλες οι πράξεις του προγράμματος είναι ατομικές.

```
initially
y1 := n; y2 := 0; y3 = 1;
```

```
left                                right
11: if (y1=n-k) then stop;          r1: if (y2=k) then stop;
12: y3 := y3*y1;                    r2: y2++;
13: y1--;                            r3: await y2 <= n-y1;
14: goto 11                          r4: y3 := y3/y2;
||                                    r5: goto r1;
```