

Ο γράφος προβαδίσματος είναι ένα πολύ καλό εργαλείο για τη γραφική αναπαράσταση ταυτόχρονων προγραμμάτων. Οι συνηθισμένες όμως γλώσσες προγραμματισμού δεν είναι φτιαγμένες έτσι ώστε να μπορούν να περιγράψουν ταυτόχρονα προγράμματα και επομένως ούτε και γράφους προβαδίσματος. Για να συμβεί αυτό, θα πρέπει οι γλώσσες προγραμματισμού να εμπλουτιστούν κατάλληλα με νέα στοιχεία, όπως θα δούμε στη συνέχεια.

Ο συμβολισμός `parbegin ... parend`

Οι εντολές `parbegin` (**parallel begin**) και `parend` (**parallel end**) είναι εντολές υψηλού επιπέδου για την περιγραφή ταυτοχρονισμού που προτάθηκαν από τον Dijkstra το 1965.

Σε ένα πρόγραμμα π.χ. Pascal τα `begin ... end` περικλείουν μια ομάδα εντολών που πρέπει να εκτελεστεί σειριακά. Οι εντολές της ομάδας χωρίζονται μεταξύ τους με ερωτηματικό («;»). Αντίστοιχα, μια ομάδα εντολών που μπορεί να εκτελεστεί παράλληλα περικλείεται από τα `parbegin ... parend` και οι εντολές χωρίζονται μεταξύ τους με το σύμβολο «||». Οι εντολές αυτές μπορούν να είναι απλές εντολές ή σύνθετες, να είναι δηλαδή με τη σειρά τους μια ομάδα εντολών που περικλείονται από `begin...end`.

Η εκτέλεση μιας ομάδας παραλλήλων εντολών που περικλείονται από τα `parbegin ... parend` γίνεται ταυτόχρονα: ξεκινούν όλες μαζί και η ομάδα εντολών τελειώνει όταν όλες τους τερματίσουν.

Η παράλληλη υλοποίηση του προγράμματος του Σχήματος Α φαίνεται στο Σχήμα Β.

Οι εντολές `parbegin ... parend` δεν είναι σε θέση να περιγράψουν όλους τους δυνατούς γράφους προβαδίσματος· έχουν όμως το πλεονέκτημα ότι κάνουν το παράλληλο πρόγραμμα πολύ ευανάγνωστο και κατανοητό.

```
begin
  m1 := x1 * x2;
  m2 := x3 - x4;
  m3 := m2 / x5;
  r := m1 - m3
end
```

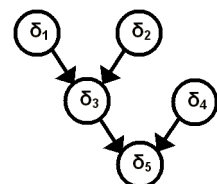
Σχήμα Α

```
begin
  parbegin
    m1 := x1 * x2 ||
    begin
      m2 := x3 - x4;
      m3 := m2 / x5;
    end;
  parend;
  r := m1 - m3
end
```

Σχήμα Β

Γράφοι προβαδίσματος και προγράμματα

Πολλοί γράφοι προβαδίσματος μπορούν να αντιστοιχιστούν με περισσότερα από ένα προγράμματα γραμμένα με τις εντολές `parbegin-parend`. Στο γράφο του σχήματος, η εντολή δ_4 μπορεί να εκτελεστεί ταυτόχρονα με οποιαδήποτε από τις δ_1 , δ_2 και δ_3 , γιατί δεν υπάρχει μονοπάτι μέσα στο γράφο που να τις συνδέει. Έτσι μπορούν να γραφούν τουλάχιστον τρία διαφορετικά παράλληλα προγράμματα τα οποία υπακούουν στους περιορισμούς του γράφου.



Οι τρεις διαφορετικές λύσεις είναι:

- 1) Η δ_4 εκτελείται παράλληλα με τις δ_1 και δ_2 (οι οποίες μπορούν να εκτελεστούν παράλληλα), πριν από την δ_3 . Το πρόγραμμα είναι:

```
parbegin 1 || 2 || 4 parend;
3; 5
```

- 2) Πρώτα εκτελούνται παράλληλα οι δ_1 και δ_2 , και στη συνέχεια η δ_3 εκτελείται παράλληλα με την δ_4 .

```
parbegin 1 || 2 parend;
parbegin 3 || 4 parend;
5
```

- 3) Η δ_1 και η δ_2 εκτελούνται παράλληλα, και μόλις τελειώσουν και οι δύο εκτελείται η δ_3 . Η δ_4 εκτελείται παράλληλα με ολόκληρη την ομάδα των δ_1 , δ_2 , δ_3 : ξεκινά μαζί με τις δ_1 , δ_2 , αλλά η δ_3 ξεκινά ανεξάρτητα από αυτήν. Η δ_5 περιμένει τις δ_3 και δ_4 να τερματίσουν για να ξεκινήσει.

```
parbegin
  begin
    parbegin 1 || 2 parend;
    3;
  end
  ||
  4
parend;
5
```