12 grudnia 2006

Druga komputerówka

Należy skonstruować predykat szeregujący zadania. Zależności między zadaniami reprezentujemy w formie grafu skierowanego: krawędź $a \to b$ oznacza, że zadanie a musi być zrealizowane przed zadaniem b. Należy obliczyć kolejność realizacji zadań zgodną z zadanymi zależnościami.
Zapytanie `szereguj(G,L)` powinno dla grafu `G` reprezentowanego za pomocą list sąsiedztwa zwrócić co najmniej jedną listę zadań `L` w kolejności zgodnej z wymaganiami zawartymi w `G`. Jeśli zadań nie da się uszeregować zgodnie z wymaganiami, zapytanie powinno ponieść porażkę.
Nie ma obowiązku zwracania wszystkich poprawnych szeregowań. W szczególności, dla pewnych *ustalonych* poprawnych szeregowań `L` zapytanie może ponosić porażkę.
Przyjmujemy, że każdy wierzchołek ma swoją listę sąsiedztwa, być może pustą.

Przykłady:

```
:-szereguj([para(a,[b]), para(b,[c]), para(c,[b])], L).
No
:-szereguj([para(a,[b]), para(b,[c]), para(c,[])], L).
L=[a,b,c] n
No
:-szereguj([para(a,[]), para(b,[c]), para(c,[])], L).
L=[a,b,c] n
L=[b,a,c] n
L=[b,c,a] n
No
```

Efektywność rozwiązania będzie miała wpływ na punktację, jednak najważniejsze jest uzyskanie *poprawnego* rozwiązania.