

# Języki i paradygmaty programowania

## Lista 5 (elementy języka Prolog)

Przemysław Kobyłański

Zaprogramuj w języku Prolog rozwiązania poniższych zadań. Na ocenę dostateczną trzeba rozwiązać wszystkie zadania bez gwiazdek. Na ocenę dobrą trzeba dodatkowo rozwiązać wszystkie zadania z jedną gwiazdką. Na ocenę bardzo dobrą trzeba rozwiązać wszystkie zadania.

### Zadanie 1

Graf skierowany można reprezentować w postaci faktów `arc(X, Y)` wyrażających istnienie łuku od wierzchołka `X` do wierzchołka `Y`.

Zapisz graf z rys. 1 w postaci faktów `arc/2`. Następnie napisz predykat `path(X, Y)` w postaci reguł, które wnioskują czy istnieje ścieżka od wierzchołka `X` do wierzchołka `Y`.

Czy Twój program będzie działać poprawnie dla dowolnych grafów, również takich, które zawierają cykl.

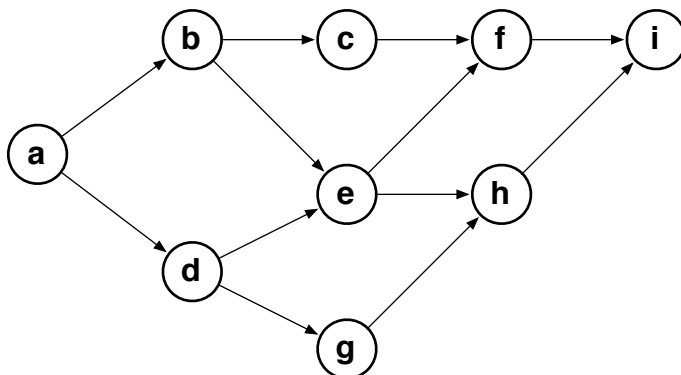
### Uwaga

Jeśli Twój program będzie udzielał tylu odpowiedzi ile jest ścieżek to również zostanie uznany za poprawny.

### Zadanie 2\*

Zdefiniuj następujące predykaty:

**jednokrotnie(X, L)** spełniony gdy wartość `X` występuje dokładnie jeden raz na liście `L`,



Rysunek 1: Przykład grafu skierowanego.

**dwukrotnie(X, L)** spełniony gdy wartość X występuje dokładnie dwa razy na liście L.

## Przykład

Zwróć uwagę na przepływ danych w poniższym przykładzie oraz na liczbę odpowiedzi, szczególnie w przypadku predykatu `!tt dwukrotnie/2`.

```
?- jednokrotnie(1, [1, 2, 3, 2, 3, 4]).  
true ;  
false.
```

```
?- jednokrotnie(2, [1, 2, 3, 2, 3, 4]).  
false.
```

```
?- jednokrotnie(3, [1, 2, 3, 2, 3, 4]).  
false.
```

```
?- jednokrotnie(4, [1, 2, 3, 2, 3, 4]).  
true ;  
false.
```

```
?- jednokrotnie(X, [1, 2, 3, 2, 3, 4]).  
X = 1 ;  
X = 4 ;  
false.
```

```
?- dwukrotnie(1, [1, 2, 3, 2, 3, 4]).  
false.
```

```
?- dwukrotnie(2, [1, 2, 3, 2, 3, 4]).  
true ;  
false.
```

```
?- dwukrotnie(3, [1, 2, 3, 2, 3, 4]).  
true ;  
false.
```

```
?- dwukrotnie(4, [1, 2, 3, 2, 3, 4]).  
false.
```

```
?- dwukrotnie(X, [1, 2, 3, 2, 3, 4]).  
X = 2 ;  
X = 3 ;  
false.
```

```
?- jednokrotnie(X, [1, 2, 1, 2, 1, 2]).  
false.
```

```
?- dwukrotnie(X, [1, 2, 1, 2, 1, 2]).  
false.
```

## Wskazówka

Do rozwiązania zadania wystarczy znajomość predykatów `member/2`, `append/3` oraz `\+/1` (negacja).

## Zadanie 3\*\*

Korzystając z zamrażania celów (predykaty `freeze/2` i `when/2`) napisz, w postaci predykatu `sieve(N, X)`, sito Eratostenesa dostarczające w strumieniu (na liście) `X` kolejne liczby pierwsze z zakresu od 2 do `N`.

## Literatura

- [1] W. Clocksin, C. Mellish. Prolog. Programowanie. Helion, 2003.