

# PROGRAMOWANIE W LOGICE

## Ograniczenia arytmetyczne

### (Lista 8)

Przemysław Kobyłański

## Wstęp

brak

## Zadania

### Zadanie 1 (5 pkt)

Dla mapy podanej na rysunku 1 i trzech kolorów zapisz problem pokolorowania mapy (żadne dwa obszary graniczące ze sobą nie mogą być pokolorowane tym samym kolorem) w postaci prologowych zmiennych z dziedzinami, na które narzucono odpowiednie ograniczenia.

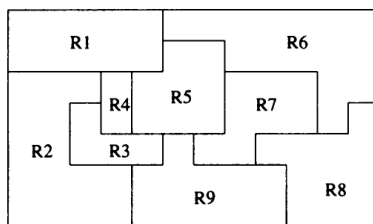
Znajdź w SWI-Prologu poprawne kolorowanie mapy.

Zadanie 3.1 z książki [1].

### Przykład

```
?- kolorowanie(X).
X = [_7942, _7948, _7954, _7960, _7966, _7972, _7978, _7984, _7990],
% tutaj pojawia się informacja o dziedzinach i ograniczeniach

?- kolorowanie(X), label(X).
X = [1, 2, 1, 3, 2, 3, 1, 2, 3] ;
X = [1, 3, 1, 2, 3, 2, 1, 3, 2] ;
X = [2, 1, 2, 3, 1, 3, 2, 1, 3] ;
X = [2, 3, 2, 1, 3, 1, 2, 3, 1] ;
X = [3, 1, 3, 2, 1, 2, 3, 1, 2] ;
X = [3, 2, 3, 1, 2, 1, 3, 2, 1].
```



Rysunek 1: Mapka do pokolorowania (źródło [1]).

## Zadanie 2 (3 pkt)

Napisz w SWI-Prologu z wykorzystaniem programowania ograniczeń predykat `plecak(+Wartości, +Wielkości, +Pojemność, -Zmienne)`, który dla danej listy wartości przedmiotów, listy wielkości przedmiotów i pojemności plecaka, rozwiązuje dyskretny problem plecakowy.

[pl.wikipedia.org/wiki/Proble](http://pl.wikipedia.org/wiki/Proble)

### Przykład

```
?- plecak([10,7,1,3,2],[9,12,2,7,5], 15, X).
X = [1, 0, 0, 0, 1].
```

### Wskazówka

Aby Prolog znajdował tylko jedno najlepsze rozwiązanie użyj predykatu `once/1`:

```
once(labeling([max(Cel)], Zmienne))
```

## Zadanie 3 (2 pkt)

Napisz w SWI-Prologu z wykorzystaniem programowania ograniczeń predykat `odcinek(Lista)`, który oddaje listę złożoną z szesnastu elementów będących zmiennymi o dziedzinach `0..1`, na które narzucono ograniczenia (koniecznie arytmetyczne liniowe) tż. wartości spełniające te ograniczenia tworzą na liście zwarty odcinek złożony z ośmiu jedynek.

Od dra M. Gębali.

### Przykład

```
?- odcinek(X).
X = [_11942, _11948, _11954, _11960, _11966, _11972, _11978, _11984, _11990|...],
% tutaj pojawia się informacja o dziedzinach i ograniczeniach

?- odcinek(X), label(X), writeln(X), fail.
[0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1]
[0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,0]
[0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0]
```

```
[0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0]
[0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0]
[0,0,0,1,1,1,1,1,1,1,1,1,0,0,0,0]
[0,0,1,1,1,1,1,1,1,1,1,0,0,0,0,0]
[0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0]
[1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0]
false.
```

## Literatura

- [1] Kimbal Marriott, Peter Stuckey. *Programming with Constraints: An Introduction*. The MIT Press, 1998.