

PROGRAMOWANIE W LOGICE

Poszukiwanie rozwiązań

(Lista 4)

Przemysław Kobyłański

Wstęp

Naturalnym sposobem wyrażania poszukiwania rozwiązania jest *generate-and-test*. Cel taki ma postać koniunkcji dwóch warunków:

1. Pierwszy warunek pełni rolę generatora dostarczającego potencjalne rozwiązania.
2. Drugi warunek sprawdza czy skonstruowane rozwiązanie jest poprawne.

Oto przykład programu ustawiającego N hetmanów na szachownicy o N wierszach i N kolumnach, tak aby żadne dwa się nie biły:

```
hetmany(N, P) :-
    numlist(1, N, L),
    permutation(L, P),
    dobra(P).

dobra(P) :-
    \+ zla(P).

zla(P) :-
    append(_, [Wi | L1], P),
    append(L2, [Wj | _], L1),
    length(L2, K),
    abs(Wi - Wj) =:= K + 1.
```

Predykat `hetmany(N, P)` znajduje taką permutację P liczb od 1 do N , która jest dobra, tzn. żadne dwa hetmany ustawione zgodnie z tą permutacją nie biją się po ukosie.

Zadania

Zadanie 1 (10 pkt)

Napisz predykat wyrażenie(LISTA, WARTOŚĆ, WYRAŻENIE), który dla danej listy liczb LISTA i danej wartości WARTOŚĆ buduje wyrażenie WYRAŻENIE powstałe przez wstawienie między liczby znajdujące się na liście operatorów arytmetycznych +, -, * i / oraz, jeśli potrzeba, nawiasów.

?- wyrażenie([1, 2], 2, X).

X = 1*2 .

?- wyrażenie([1, 2, 3], 2, X).

X = 1-2+3 .

?- wyrażenie([1, 2, 3, 4], 2, X).

X = 1+ (2+ (3-4)) .

?- wyrażenie([1, 2, 3, 4, 5], 2, X).

X = 1+ ((2-3)*4+5) .

?- wyrażenie([1, 2, 3, 4, 5, 6], 2, X).

X = 1+ (2+ (3+4* (5-6))) .

?- wyrażenie([1, 2, 3, 4, 5, 6, 7], 2, X).

X = 1+ (2+ (3+ (4+ (5- (6+7))))) .

?- wyrażenie([1, 2, 3, 4, 5, 6, 7, 8], 2, X).

X = 1+ (2+ (3+4* (5+6* (7-8)))) .

?- wyrażenie([1, 2, 3, 4, 5, 6, 7, 8, 9], 2, X).

X = 1+ (2+ (3+ (4+ (5+ (6+7)* (8-9))))) .

?- wyrażenie([1, 9, 2, 8, 3, 7, 4, 6, 5], 1, X).

X = 1+ (9+ (2+ (8+ (3+ (7- (4*6+5))))) .

?- wyrażenie([1, 9, 2, 8, 3, 7, 4, 6, 5], 10, X).

X = 1+ (9+ (2+ (8+ (3- (7-4)*6+5))) .

?- wyrażenie([1, 9, 2, 8, 3, 7, 4, 6, 5], 100, X).

X = 1+ (9+ (2+ (8* (3-7)+4* (6*5))) .

?- wyrażenie([1, 9, 2, 8, 3, 7, 4, 6, 5], 1000, X).

X = 1+ (9+2* ((8+3)* ((7-4+6)*5))) .

?- wyrażenie([1, 9, 2, 8, 3, 7, 4, 6, 5], 10000, X).

X = 1+9* (((2+8*3)*7+4)*6-5) .

Zadanie 2 (6 pkt)

Na stronie https://pl.wikipedia.org/wiki/Zagadka_Einsteina znajdziesz treść zagadki.

Napisz predykat `rybki(Kto)`, który znajduje odpowiedź na pytanie „Kto hoduje rybki?”.

Uwaga

Pomyśl o domach jako o liście pięciu termów, przy czym każdy z termów opisuje jeden domek wyrażając:

- jakiego koloru jest domek,
- kto w nim mieszka,
- jakie zwierze jest w nim hodowane,
- co się w nim pije,
- co się z nim pali.

Przykład

```
?- time(rybki(Kto)).  
% 5,291 inferences, 0.001 CPU in 0.002 seconds (69% CPU, 3987189 Lips)  
Kto = niemiec .
```

Zadanie 3 (4 pkt)

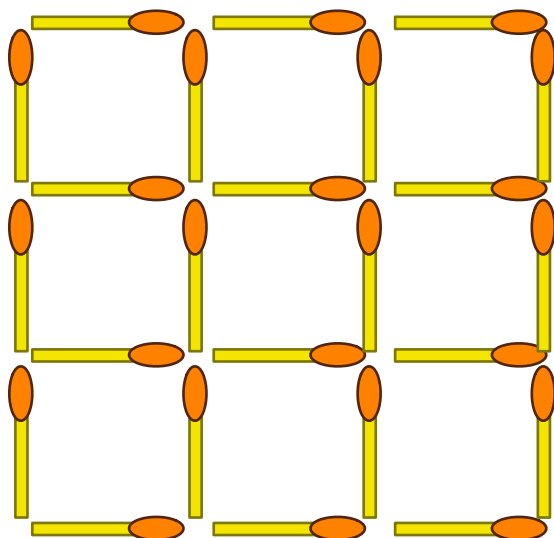
Dwadzieścia cztery zapalki ułożono w sposób przedstawiony na rysunku 1. Tworzą one jeden duży kwadrat 3×3 , cztery średnie kwadraty 2×2 oraz dziewięć małych kwadratów 1×1 .

Napisz predykat `zapałki(K, D, S, M)`, który zabiera K zapalek tak aby pozostałe tworzyły: D dużych kwadratów, S średnich kwadratów i M małych kwadratów.

Predykat powinien rysować rozwiązanie prostymi znakami ASCII (prezentować jak leżą pozostawione zapalki).

Uwaga

Przy pisaniu programu mogą być przydatne predykaty wyliczające przecięcie i sumę mnogościową zbiorów (poczytaj o predykcach `intersection/3` i `union/3` w rozdziale <http://www.swi-prolog.org/pldoc/man?section=lists>).



Rysunek 1: Ułożenie dwudziestu czterech zapalek

Przykłady

Zabierz cztery zapalki aby pozostał jeden duży kwadrat, dwa średnie i trzy małe:

?- zapalki(4, 1, 2, 3).

Rozwiązanie:

```
+---+---+---+
|       |   |
+   +---+---+
|   |   |   |
+---+---+   +
|   |       |
+---+---+---+
```

true ;

Rozwiązanie:

```
+---+---+---+
|   |       |
+---+---+   +
|   |   |   |
+   +---+---+
|       |   |
+---+---+---+
```

true ;

false.

Zabierz cztery zapalki aby pozostaly trzy srednie i cztery male kwadraty:

?- zapalki(4, 0, 3, 4).

Rozwiazanie:

```
+---+---+---+
|   |   |   |
+---+---+   +
|   |   |   |
+---+---+---+
|           |
+---+---+   +
```

true ;

Rozwiazanie:

```
+---+---+---+
|   |   |   |
+   +---+---+
|   |   |   |
+---+---+---+
|           |
+   +---+---+
```

true ;

Rozwiazanie:

```
+---+---+   +
|           |
+---+---+---+
|   |   |   |
+---+---+   +
|   |   |   |
+---+---+---+
```

true ;

Rozwiazanie:

```
+   +---+---+
|           |
+---+---+---+
|   |   |   |
+   +---+---+
|   |   |   |
+---+---+---+
```

true ;

false.

Zabierz siedem zapalek aby pozostaly dwa srednie i dwa male kwadraty.

?- zapalki(7, 0, 2, 2).

Rozwiazanie:

```
+---+---+ +
|   |   |
+ +---+---+
|   |   |   |
+---+---+ +
      |   |
+ +---+---+
```

true ;

Rozwiazanie:

```
+---+---+ +
|           |
+---+---+---+
|   |   |   |
+---+---+ +
      |   |
+ +---+---+
```

true ;

Rozwiazanie:

```
+---+---+ +
|           |
+ +---+---+
|   |   |   |
+---+---+---+
      |   |
+ +---+---+
```

true ;

Rozwiazanie:

```
+---+---+ +
|           |
+ +---+---+
|   |   |   |
+---+---+ +
      |   |   |
+ +---+---+
```

true ;

Rozwiazanie:

```
+ +---+---+
      |   |   |
+---+---+ +
|   |   |   |
+ +---+---+
|           |
+---+---+ +
```

true ;

Rozwiązanie:

```
+ +---+---+
  |         |
+---+---+ +
| | | |
+---+---+---+
|         |
+---+---+ +
true ;
```

Rozwiązanie:

```
+ +---+---+
  |         |
+---+---+---+
| | | |
+ +---+---+
|         |
+---+---+ +
true ;
```

Rozwiązanie:

```
+ +---+---+
  |         |
+---+---+ +
| | | |
+ +---+---+
| | |
+---+---+ +
true ;
false.
```

Ile zabrać zapalek aby pozostał jeden duży i jeden mały kwadrat?

?- zapalki(K, 1, 0, 1).

Rozwiązanie:

```
+---+---+---+
|         |
+ +---+ +
| | | |
+ +---+ +
|         |
+---+---+---+
K = 8 ;
```

Rozwiązanie:

```
+---+---+---+
|   |   |   |
+   +---+   +
|   |       |
+   +   +   +
|   |       |
+---+---+---+
```

K = 9 ;

Rozwiązanie:

```
+---+---+---+
|       |   |
+---+   +   +
|   |       |
+---+   +   +
|       |   |
+---+---+---+
```

K = 9 ;

Rozwiązanie:

```
+---+---+---+
|       |   |
+   +   +---+
|       |   |
+   +   +---+
|       |   |
+---+---+---+
```

K = 9 ;

Rozwiązanie:

```
+---+---+---+
|       |   |
+   +   +   +
|       |   |
+   +---+   +
|   |   |   |
+---+---+---+
```

K = 9 ;

Rozwiązanie:

```
+---+---+---+
|   |       |
+---+   +   +
|       |   |
+   +   +   +
|       |   |
+---+---+---+
```

K = 10 ;

Rozwiazanie:

```
+---+---+---+
|           | |
+   +   +---+
|           |
+   +   +   +
|           |
+---+---+---+
```

K = 10 ;

Rozwiazanie:

```
+---+---+---+
|           |
+   +   +   +
|           |
+---+   +   +
|   |           |
+---+---+---+
```

K = 10 ;

Rozwiazanie:

```
+---+---+---+
|           |
+   +   +   +
|           |
+   +   +---+
|           | |
+---+---+---+
```

K = 10 ;

false.