Cvičení 5

Programování s omezujícími podmínkami

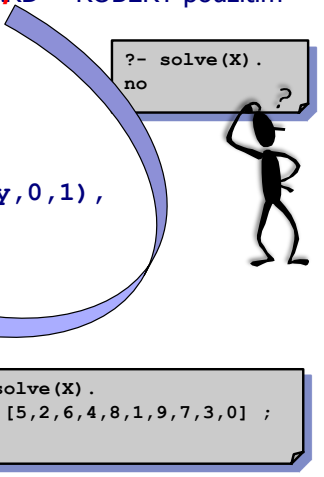
Roman Barták
Katedra teoretické informatiky a matematické logiky

roman.bartak@mff.cuni.cz
http://ktiml.mff.cuni.cz/~bartak

Algebrogramy

- Vyřešte algebrogram DONALD + GERARD = ROBERT použitím modelu s přenosovými bity.

```
:-use_module(library(clpfd)).
solve(Sol):-
  Sol=[D,O,N,A,L,G,E,R,B,T],
  Carry = [P1,P2,P3,P4,P5],
  domain(Sol,0,9), domain(Carry,0,1),
  D #\= 0, G #\= 0, R #\= 0,
  D+D    #= T + 10*P1,
  L+R+P1 #= R + 10*P2,
  A+A+P2 #= E + 10*P3,
  N+R+P3 #= B + 10*P4,
  O+E+P4 #= O + 10*P5,
  D+G+P5 #= R,
  all_different(Sol),
  append(Sol,Carry,Vars),
  labeling([],Vars).
```



Instanciace proměnných

- Podmínky většinou nezajistí ohodnocení proměnných a proto potřebujeme přiřadit hodnoty proměnným – prohledávání.
- **`indomain(X)`**
  - do proměnné přiřadí hodnotu (hodnoty zkouší v rostoucím pořadí)
- **`labeling(Params,Vars)`**
  - ohodnotí proměnné Vars
  - Algoritmus MAC – backtracking s udržováním hranové konzistence

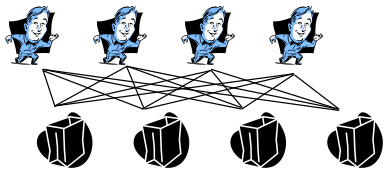
Specifikace podmínek
tabulkové binárně

- Podmínky popsané extenzionálně můžeme pomocí kódování se skrytou proměnnou převést na binární podmínky kódované podmínkou **`element`**.
- **`element(X,List,Y)`**
  - Y is X-tý prvek seznamu list
- **Postup**
  - jednotlivé n-tice očíslujeme
  - pro každou proměnnou uděláme element podmínku, kde seznam jsou hodnoty proměnné v n-ticích
- Příklad: [[1,2,3], [1,3,4], [2,4,4]]

```
?-element(I, [1,1,2], X), element(I, [2,3,4], Y),
  element(I, [3,4,4], Z)
I in 1..3,
X in 1..2,
Y in 2..4,
Z in 3..4
```

Modelování

Uvažujme čtyři pracovníky, kteří mohou pracovat na čtyřech produktech, kde efektivita pracovníka při výrobě produktu je dána tabulkou.



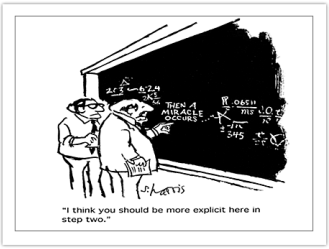
	P1	P2	P3	P4
W1	7	1	3	4
W2	8	2	5	1
W3	4	3	7	2
W4	3	1	6	3

Je-li Wi proměnná určující, na jakém produktu pracuje pracovník i, jak vypadá podmínka dávající jeho efektivitu Ei?

```
element(W1,[7,1,3,4],E1),
element(W2,[8,2,5,1],E2),
element(W3,[4,3,7,2],E3),
element(W4,[3,2,6,3],E4).
```


Modelování problémů

- Jaké **rozhodovací proměnné** potřebujeme?
  - proměnné, jejichž hodnoty určují řešení problému
  - také určují prohledávaný prostor
- Jakých **hodnot** mohou proměnné nabývat?
  - definice domény může ovlivnit výběr podmínek
- Jak formalizovat **podmínky**?
  - jaké podmínky jsou v systému dostupné?
  - možná budou potřeba pomocné proměnné



4-královny

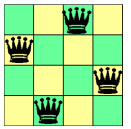
- Navrhněte model pro řešení problému 4-královen (rozmístěte královny na šachovnici 4x4 tak, aby se neohrožovaly).

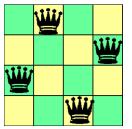
```
:-use_module(library(clpfd)).

queens([(X1,Y1),(X2,Y2),(X3,Y3),(X4,Y4)]):-
   Rows = [X1,X2,X3,X4], Columns = [Y1,Y2,Y3,Y4],
   domain(Rows,1,4),
   domain(Columns,1,4),
   all_different(Rows), all_different(Columns),
   abs(X1-X2) #\= abs(Y1-Y2),
   abs(X1-X3) #\= abs(Y1-Y3), abs(X1-X4) #\= abs(Y1-Y4),
   abs(X2-X3) #\= abs(Y2-Y3), abs(X2-X4) #\= abs(Y2-Y4),
   abs(X3-X4) #\= abs(Y3-Y4),
   append(Rows,Columns, Variables),
   labeling([], Variables).
```

4-královny
analýza

```
?- queens(L).
L = [(1,2),(2,4),(3,1),(4,3)] ? ;
L = [(1,3),(2,1),(3,4),(4,2)] ? ;
L = [(1,2),(2,4),(4,3),(3,1)] ? ;
L = [(1,3),(2,1),(4,2),(3,4)] ? ;
L = [(1,2),(3,1),(2,4),(4,3)] ? ;
L = [(1,3),(3,4),(2,1),(4,2)] ? ;
L = [(1,2),(3,1),(4,3),(2,4)] ? ;
L = [(1,3),(3,4),(4,2),(2,1)] ? ;
...
```





Kde je problém?

- různá ohodnocení odpovídají stejným pozicím!
- existují pouze dvě přípustné konfigurace (a i ty jsou „podobné“)
- prohledávaný prostor je zbytečně velký.

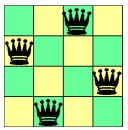
Řešení

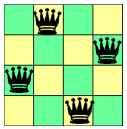
- zafixujeme řádky (nebo sloupce) královen

4-královny
lepší model

```
:-use_module(library(clpfd)).
queens4(Queens):-
   Queens = [X1,X2,X3,X4],
   domain(Queens,1,4),
   all_different(Queens),
   abs(X1-X2) #\= 1, abs(X1-X3) #\= 2, abs(X1-X4) #\= 3,
   abs(X2-X3) #\= 1, abs(X2-X4) #\= 2,
   abs(X3-X4) #\= 1,
   labeling([], Queens).
```

```
?- queens4(Q).
Q = [2,4,1,3] ? ;
Q = [3,1,4,2] ? ;
no
```





Vlastnosti modelu:

- méně proměnných (= menší prohledávaný prostor)
- méně podmínek (= rychlejší propagace)

DCV:

- program pro libovolný zadaný počet královen
- možná další vylepšení