Cvičení 6

Programování s omezujícími podmínkami

Roman Barták
Katedra teoretické informatiky a matematické logiky

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

all-different/all-distinct

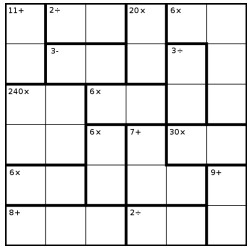
```
?- X in 1..2, Y in 1..2, Z in 1..3,
  all_different([X,Y,Z]).
X in 1..2,
Y in 1..2,
Z in 1..3
?- X in 1..2, Y in 1..2, Z in 1..3,
  all_distinct([X,Y,Z]).
Z = 3,
X in 1..2,
Y in 1..2
```

???

- **all-different** se chová jako množina binárních nerovností
- **all-distinct** je globální podmínka se silnější filtrací

Latinské čtverce

- **Latinský čtverec** řádu n je tabulka n x n obsahující čísla 1,…,n tak, že se žádné číslo neopakuje v řádku ani ve sloupci.
- Navrhněte model pro nalezení latinského čtverce obecného řádu, kde jsou některá čísla před-vyplněna.
- Navrhněte model pro řešení **Sudoku** problémů (latinský čtverec řádu 9, kde se číslo nesmí opakovat v „podoblastech 3x3).
- Navrhněte model pro řešení Sudoku problémů, kde mezi sousedními políčky může být aritmetická relace <.
- Navrhněte model pro řešení **KenKen** problémů (latinský čtverec, kde je u některých oblastí určena operace s čísly a její výsledek



Problém batohu

- Pašerák má batoh s kapacitou 9 jednotek. Může pašovat lahve whisky velikosti 4 jednotky, lahve parfémů velikosti 3 jednotky nebo kartony cigaret velikosti 2 jednotky. Zisk z whisky je 15 dolarů, z parfému 10 dolarů a z cigaret 7 dolarů. Pokud pašerák podniká cesty se ziskem minimálně 30 dolarů, co může vzít?

Problém batohu
řešení

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

smuggler(Goods):-
  Goods = [W,P,C],
  domain(Goods,0,4),
  4*W + 3*P + 2*C #=< 9,
  15*W + 10*P + 7*C #>= 30,
  labeling([],Goods).
```

```
?- smuggler(G).
G = [0,1,3] ? ;
G = [0,3,0] ? ;
G = [1,1,1] ? ;
G = [2,0,0] ? ;
no
```

A jaké je optimální složení kontrabandu?

- poslední dva řádky nahradíme následujícím kódem

```
15*W + 10*P + 7*C #= Cost,
labeling([maximize(Cost)],Goods)
```


Problém batohu
obecně

- Je dán batoh dané velikosti m. Máme n předmětů s velikostmi a cenami. Vyberte předměty tak, aby se vešly do batohu a jejich celková cena byla maximální.

```
knapsack(+Size,+ListOfSizes,
  +ListOfCosts,-Goods,-Cost)
```

- Bude se hodit podmínka:
  - `scalar_product(Coeffs, Variables, RelOp, Value)`
  - Coeffs*Variables RelOp Value

Stěhování

- Petra potřebuje vystěhovat svůj byt. Na pomoc má tři kamarády (celkem jsou tedy čtyři), ale k dispozici pouze 60 minut. Následující tabulka ukazuje, co se bude stěhovat, jak dlouho to trvá a kolik je potřeba lidí.

věc	čas (min.)	lidé
Piáno	30	3
Židle	10	1
Postel	15	3
Stůl	15	2

- Kdy se má jaký předmět stěhovat?

Stěhování
základní model

- Potřebujeme zjistit, kdy má začít stěhování každého předmětu tak, aby se v dané chvíli nestěhovalo více předmětů než je kapacita lidí.

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

furniture(Furniture):-
  Furniture = [P, Ch, B, T],
  domain(Furniture,0,60),
  P+30 #=< 60,          % piano
  Ch+10 #=< 60,         % chair
  B+15 #=< 60,          % bed
  T+15 #=< 60,          % table
  P+30 #=< B #\/ B+15 #=< P,
  P+30 #=< T #\/ T+15 #=< P,
  B+15 #=< T #\/ B+15 #=< T,

  labeling([],Furniture).
```

věc	čas	lidé
piano	30	3
chair	10	1
bed	15	3
table	15	2

stěhování piana a postele musí probíhat v různých časech

Stěhování

model s globální podmínkou

- Disjunkce (zpravidla) propagují málo. Je možné použít globální podmínku cumulative pro alokaci aktivit na zdroje.

Počet pokusů (voleb)	
furniture	19
furnitureG	3

```
furnitureG(Furniture):-
  Furniture = [P, Ch, B, T],
  domain(Furniture,0,60),
  P+30 #= EP, EP #=< 60,            % piano
  Ch+10 #= ECh, ECh #=< 60,         % chair
  B+15 #= EB, EB #=< 60,            % bed
  T+15 #= ET, ET #=< 60,            % table
  cumulative([task(P,30,EP,3,1),
              task(Ch,10,ECh,1,2),
              task(B,15,EB,3,3),
              task(T,15,ET,2,4)], [limit(4)]),

  labeling([],Furniture).
```

task(start,duration,end,capacity,id)

Toustovač

- Navrhněte CP model pro problém toustovače, kde je úkol minimalizovat počet opékaní. Do toustovače se vejdou dva tousty a každý toust je potřeba opéct z obou stran. Vstupem je počet toustů, které je potřeba opéct.

Domací úkol