Unequality constraints to propositional logic

Matouš Mařík

```
Suppose we are given a CSP over some unified domain D : [d1 ... d2] where all
constraints are of the form vi ≤ vj, vi − vj ≤ c or vi = vj + c for some constant c.
For example
    ((v2 ≤ v3) ∨ (v4 ≤ v1)) ∧ (v2 = v1 + 4) ∧ v4 = v3 + 3
for v1, v2, v3, v4 ∈ [0 ... 7] is a formula belonging to this fragment. This
formula is satisfied by one of two solutions: (v1 ↦ 0, v2 ↦ 4, v3 ↦ 4, v4 ↦ 7),
or (v3 ↦ 0, v1 ↦ 3, v4 ↦ 3, v2 ↦ 7). Show a reduction of such formulas to
propositional logic.

Hint: an encoding which requires |V|·|D| propositional variables, where |V| is
the number of variables and |D| is the size of the domain, is given by
introducing a propositional variable bij for each variable vi ∈ V and j ∈ D, which
indicates that vi ≤ j is true.
```

Pozn.: Výsledná formule bude konjunkce jednotlivých omezení.

Nechť $b_{i,j}$ jsou proměnné zakódované podle nápovědy (s indexováním V a D od 1). Je třeba zajistit konzistenci výsledného ohodnocení přidáním podmínky:

$$\forall i \in 1..|V| : \bigwedge_{d_1, d_2 \in D \; d_1 \leq d_2} (\neg b_{i,d_1} \vee b_{i,d_2})$$

- díky této podmínce se v kódování nemůže stát, že by hodnota proměnné nebyla menší, nebo rovna nějakému j a zároveň by hodnota proměnné byla menší, nebo rovna nějakému k, kde $j \leq k$, tedy zajišťuje tranzitivitu $\leq$.

Kódování podmínek

Nechť je doména D uspořádaná podle "$\leq$", tedy

$$\forall i, j \in 1..|D|, i \leq j : d_i \leq d_j$$

jinak pracujeme s D', která obsahuje všechny prvky D a navíc zachovává uspořádání.

Podmínka $v_i \leq v_j$

Pro každou možnou hodnotu proměnné v_i je vytvořena podmínka na hodnotu proměnné v_j, tak aby splňovala "$\leq$":

$$\bigwedge_{d \in D} (\neg b_{j,d} \vee b_{i,d})$$

- to odpovídá $\forall d \in D : v_j \leq d \implies v_i \leq d$

Podmínka $v_i - v_j \leq c$

- podmínka je nejdříve převedena do tvaru $v_i \leq v_j + c$ (plus definováno, díky výskytu další podmínky)

$$\forall d_1 \in D : \bigwedge_{d_2 \in D \; d_1 \leq d_2 + c} (\neg b_{j,d_2} \vee b_{i,d_1})$$

Podmínka $v_i = v_j + c$

$$\forall d_1 \in D : \bigwedge_{d_2 \in D \; d_1 \leq d_2 + c \; d_2 + c \leq d_1} (\neg b_{j,d_2} \vee b_{i,d_1})$$

Pozn.: pokud neexistuje žádné d_2 splňující podmínku, pak je přidána prázdná klauzule, která značí spor.

Pozn2.: výsledná formule by šla zkrátit pouze na omezování "sousedních" hodnot d_1, d_2, kde d_2 by byly nejmenší prvky z množiny $d | d_1 < d$.

Bonus

> In the previous example, try to describe an encoding with only a $O(\log2\ |D|)$ propositional variable for each variable vi.

Celá myšlenka je založena na indexování domény pomocí binární reprezentace indexu.

Každé hodnotě z D je přiřazen unikátní index (počínající od 0). Výsledná formule je potom vytvořena stejně jako v předchozím případě, akorát každá proměnná $b_{i,j}$ je nahrazena klauzulí s $\lceil \log_2 |D| \rceil$ proměnnými tak, že pro hodnotu j je vybrán odpovídající binární zápis B, který lze reprezentovat jako $b_{\lceil \log_2 |D| \rceil}, b_{\lceil \log_2 |D| \rceil - 1}, \ldots, b_1, b_0$, kde $b_i \in 0, 1$, a výsledná kombinace pro $b_{i,j}$ z předchozího řešení vypadá takto:

$$\bigwedge_{k \in 0..\lceil \log_2 |D|) \rceil} \text{bin\_repr}(i, j, k)$$

kde $\text{bin\_repr}(i, j, k)$ je buď "$\neg b_{i,k}$" pokud b_k v binárním zápisu j je 0, nebo "$b_{i,k}$", pokud je totéž rovno 1.

- Například pro 3. hodnotu z domény D - d_3(počítáno od 1). Je odpovídající index 2, jehož zápis je 0..010, tedy výsledná klauzule odpovídající proměnné b_{i,d_3} z předchozího řešení bude $(\ldots \wedge \neg b_{i,2} \wedge b_{i,1} \wedge \neg b_{i,0})$.

Zároveň je třeba zakázat všechny kombinace které neodpovídají validním indexům.

$$\forall v_i \in V, \forall d \in |D| \ldots \lceil \log_2 |D| \rceil : \neg \bigwedge_{k \in 0..\lceil \log_2 |D|) \rceil} \text{bin\_repr}(i, d, k)$$