Combinatorial Problem Solving (CPS)

2017-18 Spring Term. Final Exam: 3 hours
Publication of grades: 28/06
Revision of exams: 02/07 (office 113, Ω building, under request by e-mail: `erodri@cs.upc.edu`)

---

1. *(4 pts.)* Let us consider variables $x_1, x_2, \ldots, x_n$ with respective domains $D_1, D_2, \ldots, D_n$ such that:

   i. $D_i \subseteq \{1, 2, \ldots, n\} - \{i\}$ for each $1 \leq i \leq n$.
   ii. $j \in D_i$ if and only if $i \in D_j$.

   The *symmetric alldifferent* constraint $\mathsf{SymmAllDiff}(x_1, x_2, \ldots, x_n)$ is defined as follows. A tuple $(v_1, v_2, \ldots, v_n)$ of values for $(x_1, x_2, \ldots, x_n)$ satisfies $\mathsf{SymmAllDiff}(x_1, x_2, \ldots, x_n)$ if and only if

   (1) $v_i \in D_i$ for all $1 \leq i \leq n$, and
   (2) $v_i \neq v_j$ for all $1 \leq i < j \leq n$, and
   (3) $v_i = j$ if and only $v_j = i$ for all $1 \leq i < j \leq n$.

   (a) *(1 pt.)* Consider the following problem. You are an activity leader in charge of 8 children. You have hired 4 paddle courts for a short time, so that the 8 children have to be grouped into 4 pairs to play a single match simultaneously, one pair for each of the courts.

   Moreover, the following compatibility relations have to be respected (to simplify the notation, we will identify the children with numbers from 1 to 8):

   - Child 1 can play with children 2 and 5.
   - Child 2 can play with children 1 and 5.
   - Child 3 can play with children 4 and 6.
   - Child 4 can play with children 3 and 6.
   - Child 5 can play with children 1, 2, 7 and 8.
   - Child 6 can play with children 3, 4, 7 and 8.
   - Child 7 can play with children 5, 6 and 8.
   - Child 8 can play with children 5, 6 and 7.

   Model this problem as a CSP that uses one SymmAllDiff constraint. Specify the variables, their meanings, their domains, and the constraints of the CSP.

   (b) *(1 pt.)* The *value graph* of a constraint $\mathsf{SymmAllDiff}(x_1, x_2, \ldots, x_n)$ is the graph $G = (V, E)$ where $V = \{1, 2, \ldots, n\}$ and $E = \{ \{i, j\} \mid i \in D_j,\ j \in D_i \}$.

   Draw the value graph that corresponds to the SymmAllDiff constraint of exercise (a).

   (c) *(1 pt.)* Prove that there is a bijection between solutions to $\mathsf{SymmAllDiff}(x_1, x_2, \ldots, x_n)$ and matchings of the value graph that cover $\{1, 2, \ldots, n\}$.

   (d) *(1 pt.)* Draw the value graph that corresponds to the SymmAllDiff constraint of exercise (a) after enforcing arc consistency.

2. *(3 pts.)* Consider a linear program of the following form:

$$\begin{aligned}&\min c^T x\\&Ax = b\\&0 \le x_i \le 1 \quad \text{for all } 1 \le i \le n,\end{aligned}$$

where $c \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{m\times n}, n \ge m$ and $\operatorname{rank}(A) = m$.

(a) *(1.5 pt.)* Let us assume we apply the bounded version of the simplex algorithm to solve the above linear program. Let us consider the basic solution determined by the basic variables $\mathcal{B}$, the non-basic variables $\mathcal{L}$ assigned to the lower bound, and the non-basic variables $\mathcal{U}$ assigned to the upper bound. Note that $\mathcal{B} \cup \mathcal{L} \cup \mathcal{U} = \{x_1, x_2, \ldots, x_n\}$.

Give the (simplest) formula that expresses the values of the basic variables in the basic solution in terms of the non-basic variables.

(b) *(1.5 pts.)* Let us assume that $b \in \mathbb{Z}^m$ and $A \in \mathbb{Z}^{m\times n}$. Moreover, let us assume that A is *totally unimodular*: the determinant of every square submatrix of A is 0 or ± 1. Prove that, under these assumptions, if the above linear program has a finite optimum, then there exists an optimal solution $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$ satisfying that $x_i^* \in \mathbb{Z}$ for all $1 \le i \le n$.

Hint: You can use Cramer's rule, which states that the inverse of an invertible matrix M is

$$M^{-1} = \frac{1}{\det(M)} C^T$$

where for all $1 \le i, j \le n$ the coefficient at row i and column j of C, the so-called cofactor matrix, *is $C_{ij} = (-1)^{i+j} D_{ij}$, and D_{ij} is the determinant of the $(n-1)\times(n-1)$ matrix that results from deleting row i and column j from M.*

3. *(3 pts.)* We define the problem **NEG-SAT** as follows: given a propositional formula F, to determine whether there exists I such that $I \models \neg F$.

(a) *(1.5 pts.)* Describe a linear-time algorithm for **NEG-SAT** when the input formula is in CNF. Justify its correctness and its cost.

Hint: you can use that, given a clause C, detecting if C contains contradictory literals, i.e., p and $\neg p$ for some variable p, can be done in linear time.

(b) *(1.5 pts.)* Let us call CNF-NEG-SAT the linear-time algorithm of the previous exercise for **NEG-SAT** when the input formula is in CNF :

Algorithm CNF-NEG-SAT

Input: propositional formula F in CNF

Output: YES if there exists I such that $I \models \neg F$, NO otherwise

Consider now the following algorithm for solving the **SAT** problem for arbitrary formulas:

Algorithm MY-SAT

Input: propositional formula F

Output: YES if there exists I such that $I \models F$, NO otherwise

Step 1. $G :=$ Tseitin-transformation-of($\neg F$)
Step 2. **return** CNF-NEG-SAT(G)

Is algorithm MY-SAT correct? If so, prove it. Otherwise, give a counterexample.