Last Name **First Name** **DNI/ID Num.**

Combinatorial Problem Solving (CPS)

2019-20 Spring Term. Final Exam: 2 hours
Publication of Grades: 19/06
Revision of Exams: 22/06
(remotely at Google Meet room `unq-mrij-ejz` under request by e-mail: `erodri@cs.upc.edu`)
Exam Code: 2345-57
(if you answer in a different sheet, please copy this code as well as all personal data of the top line)

---

1. (0.9 pts.) Let us define a binary CSP C with:
   - variable x_0 with domain $\{0, 1, 3\}$
   - variable x_2 with domain $\{0, 1, 2\}$
   - variable x_4 with domain $\{0, 2, 3\}$
   - constraint $x_0 \leq x_2$
   - constraint $x_4 \neq x_0$
   - constraint $x_4 = x_2$

   Write the domains of the variables after enforcing **arc-consistency** on C:

   (a) (0.3 pts.) variable x_0 has domain

   (b) (0.3 pts.) variable x_2 has domain

   (c) (0.3 pts.) variable x_4 has domain

   Use the algorithm for enforcing arc-consistency that you prefer.

2. (0.9 pts.) Let us consider the same binary CSP C as in exercise 1.

   Write the domains of the variables after enforcing **directional arc-consistency** on C with the ordering $x_0 \prec x_2 \prec x_4$:

   (a) (0.3 pts.) variable x_0 has domain

   (b) (0.3 pts.) variable x_2 has domain

   (c) (0.3 pts.) variable x_4 has domain

   Use the algorithm for enforcing directional arc-consistency that you prefer.

3. (0.9 pts.) Let us consider the same binary CSP C as in exercise 1.

   Write the domains of the variables after enforcing **bounded arc-consistency** on C:

   (a) (0.3 pts.) variable x_0 has domain

   (b) (0.3 pts.) variable x_2 has domain

(c) (0.3 pts.) variable x_4 has domain

Use the algorithm for enforcing bounded arc-consistency that you prefer.

4. (0.9 pts.) Let us consider the same binary CSP C as in exercise 1.

   Write the domains of the variables after enforcing **singleton arc-consistency** on C:

   (a) (0.3 pts.) variable x_0 has domain

   (b) (0.3 pts.) variable x_2 has domain

   (c) (0.3 pts.) variable x_4 has domain

   Use the algorithm for enforcing singleton arc-consistency that you prefer.

5. (2.1 pts.) Let us consider the following linear program:

$$
\begin{array}{lrrrrrrrrr}
\min & & x_0 & & & + & 2x_2 & - & x_3 & & \\
\text{such that} & & & & & & & & & & \\
& - & 2x_0 & - & x_1 & + & x_2 & + & x_3 & = & 0 \\
& & & & x_1 & - & 2x_2 & & & = & -2 \\
\end{array}
$$

$$
\begin{array}{rcl}
x_0 & \geq & 0, \quad x_0 \in \mathbb{R} \\
x_1 & \geq & 0, \quad x_1 \in \mathbb{R} \\
x_2 & \geq & 0, \quad x_2 \in \mathbb{R} \\
x_3 & \geq & 0, \quad x_3 \in \mathbb{R}
\end{array}
$$

   Variables (x_2, x_1) form a feasible basis for the simplex algorithm (you do not need to prove that).

   (a) The basic solution assigns

   - (0.3 pts.) basic variable x_2 to value
   - (0.3 pts.) basic variable x_1 to value

   (b) (0.3 pts.) The value of the objective function at the basic solution is

   (c) The reduced cost of

   - (0.3 pts.) non-basic variable x_0 is
   - (0.3 pts.) non-basic variable x_3 is

   (d) (0.3 pts.) Is non-basic variable x_0 satisfying the optimality conditions? (answer **yes**/**no**) If it does not, give the best value it can be assigned to according to the ratio test (answer $+\infty$ if no such value exists).

   (e) (0.3 pts.) Is non-basic variable x_3 satisfying the optimality conditions? (answer **yes**/**no**) If it does not, give the best value it can be assigned to according to the ratio test (answer $+\infty$ if no such value exists).

6. (1.8 pts.) Let us consider the following linear program:

$$
\begin{array}{lrcrcrcrcr}
\min & & & x_1 & & & - & 2x_3 & & \\
\text{such that} & & & & & & & & & \\
& & - & x_1 & + & 2x_2 & + & x_3 & = & -2 \\
& x_0 & - & 2x_1 & - & 2x_2 & - & x_3 & = & 1 \\
& -\ x_0 & - & 2x_1 & + & x_2 & + & x_3 & = & 1
\end{array}
$$

$$
\begin{array}{rcccrl}
0 & \leq & x_0 & \leq & 2, & x_0 \in \mathbb{R} \\
1 & \leq & x_1 & \leq & 2, & x_1 \in \mathbb{R} \\
-6 & \leq & x_2 & \leq & -5, & x_2 \in \mathbb{R} \\
11 & \leq & x_3 & \leq & 12, & x_3 \in \mathbb{R}
\end{array}
$$

Variables (x_3, x_2, x_0) form a basis for the **bounded** simplex algorithm. If

- non-basic variable x_1 is assigned to value 1

the basic solution is feasible (you do not need to prove that).

(a) The basic solution assigns

- (0.3 pts.) basic variable x_3 to value
- (0.3 pts.) basic variable x_2 to value
- (0.3 pts.) basic variable x_0 to value

(b) (0.3 pts.) The value of the objective function at the basic solution is

(c) The reduced cost of

- (0.3 pts.) non-basic variable x_1 is

(d) (0.3 pts.) Is non-basic variable x_1 satisfying the optimality conditions? (answer **yes**/**no**) If it does not, give the best value it can be assigned to according to the ratio test.

7. (2.5 pts.) Let C be a constraint defined over Boolean variables $x_1, x_2, \ldots, x_n$, and let S be a CNF defined over $x_1, x_2, \ldots, x_n$ and possibly also over additional Boolean variables $y_1, y_2, \ldots, y_m$. We say that S is a *correct encoding* for C if:

(i) any assignment over $x_1, x_2, \ldots, x_n$ that satisfies C can be extended to an assignment over $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_m$ that satisfies S; and

(ii) any assignment over $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_m$ that satisfies S also satisfies C.

In this exercise we will focus on a particular kind of constraint: given n Boolean variables $x_1, x_2, \ldots, x_n$ (where $n \geq 3$), we define $AMT(x_1, x_2, \ldots, x_n)$ (At Most Two) as the constraint

$$x_1 + x_2 + \ldots + x_n \leq 2$$

(a) (0.5 pts.) Let us consider the CNF consisting of all clauses

$$\overline{x_i} \vee \overline{x_j} \vee \overline{x_k}$$

where $1 \leq i < j < k \leq n$. It can be shown (but you do not have to prove it) that this is a correct encoding for $AMT(x_1, x_2, \ldots, x_n)$, which we will refer to as the *naive encoding*.

What is the total number of clauses in terms of n? $\Theta($ ☐ $)$.

(b) (1 pt.) Let us introduce the *small-step* encoding for $AMT(x_1, x_2, \ldots, x_n)$. The CNF S is defined recursively as follows:

- If $n = 3$ then S consists of the clauses of the naive encoding for $AMT(x_1, x_2, \ldots, x_n)$.
- If $n > 3$ then S consists of:
  - the clauses of $AMT(x_1, x_2, \ldots, x_{n-1})$ using recursively the small-step encoding, and
  - a clause $\overline{x_n} \vee C$ for each clause C in Heule's encoding for $AMO(x_1, x_2, \ldots, x_{n-1})$.

What is the total number of additional variables in terms of n? $\Theta($ ☐ $)$

What is the total number of clauses in terms of n? $\Theta($ ☐ $)$.

(c) (1 pt.) Prove that the small-step encoding for $AMT(x_1, x_2, \ldots, x_n)$ is correct. You can use that Heule's encoding is correct without having to prove it.