AMMM Final Exam

June 4th, 2020

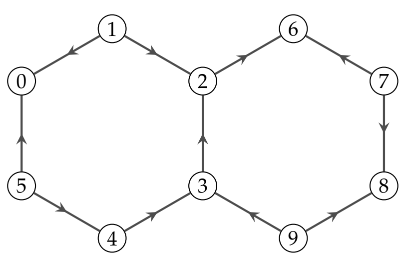
- *This part contains a single exercise and 2 pages*
- *Handwrite your solutions on a white paper, scan with app, and upload at Racó as* ***one*** *pdf file*
- *You are* ***not*** *allowed to use any support material or communicate with anyone during the exam*

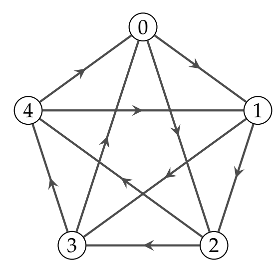
Part 2: Integer Programming **(3 pts)**

Exercise 1 (3 pts). A directed graph is called *oriented graph* if it does not have self-loops and none of its pairs of vertices is linked by two symmetric arcs. Formally, $D = (V, A)$ is *an oriented graph* if V is a finite set (the set of *vertices*) and $A \subseteq V \times V$ (the set of *arcs*) satisfies for every $u, v \in V$:

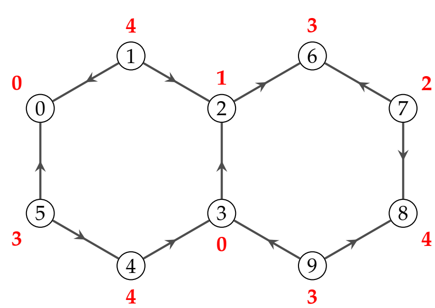
1. $(u, u) \notin A$
2. $(u, v) \in A \Rightarrow (v, u) \notin A$

The following are examples of oriented graphs, where every arc (u, v) is represented by an arrow from u to v. We call H_2 to the oriented graph on the left and T_{12} to the one on the right.





Let $D = (V, A)$ and $D' = (V', A')$ be two oriented graphs. A *homomorphism* from D to D' is a mapping $h : V \to V'$ that preserves the arcs, that is, such that if $(x, y) \in A$, then $(h(x), h(y)) \in A'$. For example, we can consider the following homomorphism h from H_2 to T_{12}, where the attached red numbers are the images of h. For instance, vertex 1 in H_2 is mapped to vertex 4 in T_{12}, that is, $h(1) = 4$. Note that for every arc (x, y) in H_2, the arc $(h(x), h(y))$ belongs to T_{12}.



We want to decide whether a given oriented graph has a homomorphism into T_{12}. Specifically, we want to find an integer program to decide, given any oriented graph D, whether there exists a homomorphism from D to T_{12}.

Suppose that D has `N` vertices and `M` arcs, and that the set of arcs is represented as a list `A` with indices ranging from `1` up to `M`. In the case of the oriented graph H_2, we would have:

```
N = 10;
M = 11;
A = [[1 0] [1 2] [2 6] [3 2] [4 3] [5 0] [5 4] [7 6] [7 8] [9 3] [9 8]];
```

In this notation, `A[1]` represents arc `[1 0]` and `A[1][2] = 0` stands for the second component of arc `[1 0]`. Note that our integer program only needs a solution since there is no function to optimize. Therefore, we can set

```
minimize 1
```

as our objective function.

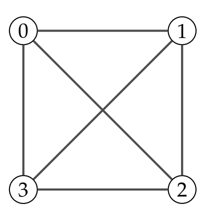
In relation to the previous problem, do the following:

1. **(2 pts)** Give an integer program for it, including the constraints and any other variables and data that may be needed. You can use either mathematical notation or OPL code.

2. **(0.5 pts)** Homomorphism for nondirected graphs is defined analogously to the oriented case. If $G = (V, E)$ and $G' = (V', E')$ are two nondirected graphs (without self-loops nor multiple edges), then a *homomorphism* from G to G' is a mapping $h : V \to V'$ that preserves the edges, that is, such that if $\{x, y\} \in E$, then $\{h(x), h(y)\} \in E'$.

   What are the changes that need to be made in your integer program so that it can decide automorphism between nondirected graphs? In concrete, consider the complete graph of four vertices, $K_4 = \{V_4, E_4\}$, where

   - $V_4 = \{0, 1, 2, 3\}$, and
   - $E_4 = \{\{u, v\} \mid u, v \in V_4 \ \wedge u \ \neq v\}$.

   Propose a representation for K_4 and give an integer program that decides whether a given nondirected graph has a homomorphism to K_4.

3. **(0.5 pts)** Explain the main ideas in the solution provided in the previous points in an understandable way (one or two paragraphs should be enough).

Solutions

1. The model is:

```
int N = ...;
int M = ...;
range I = 1..10;
range v_range = 0..N-1; // range of vertices
range a_range = 1..M; // range of arcs

int T12[i in I] = ...;
int arc[a_range][1..2] = ...;

dvar int lab[j in v_range];
dvar boolean z[i in I,a in a_range];

minimize 1;

subject to {
    // the labeling is correct
    forall (a in a_range)
        lab[A[a][1]] + 5*lab[A[a][2]] == sum(i in I) z[i,a]*T12[i];

    // labels go from 0 to 4
    forall (j in v_range) (lab[j] >= 0);
    forall (j in v_range) (lab[j] <= 4);

    // just one of the z's equals 1 for each arc
    forall (a in a_range) sum(i in I) z[i,a] == 1;
}
```

   We represent an arc (u, v) in the target graph T_{12} as the number $u + 5v$. The data is:

```
N = 10;
M = 11;
A = [[1 0] [1 2] [2 6] [3 2] [4 3] [5 0] [5 4] [7 6] [7 8] [9 3] [9 8]];
T12 = [3 4 5 8 10 11 17 19 21 22];
```

2. One possibility is just changing the representation of the target graph: now each edge $\{u, v\}$ in K_4 is represented the numbers $u + 4v$ and $4u + v$. The only change is, then, in the data file, where we have:

```
K4 = [1 2 3 4 6 7 8 9 11 12 13 15]
```

   which will be used in the model instead of `T12`.

3. Arcs are represented by unique numbers. Since T_{12} has 5 vertices with integer values ranging from 0 to 4, an arc (u, v) is uniquely represented by the number $u + 5v$ as a number in base 5. The correctness of a labeling in the source graph is checked, then, when it corresponds to some arc in the target graph. The disjunction is solved in the usual way with the help of auxiliary variables `z[i,a]`.

   In the case of undirected graphs, we duplicate the representation of the edges in the target graph (in the way explained in the previous question) so that if one of the two "directions" is detected, the edge will be detected.