# AI Lab - Lesson 3

## Bucket Elimination

Davide Corsi
Alessandro Farinelli

University of Verona
Department of Computer Science

December $2^{nd}$ 2021


UNIVERSITÀ
di VERONA
Dipartimento
di INFORMATICA

# Start Your Working Environment

Start the previously installed (Session 1) conda environment *ai-lab*

## Listing 1: Update Environment

```
cd AI-Lab
git stash (NB: remember to backup the previous lessons before this step!)
git pull
git stash pop
conda activate ai-lab
pip install networkx
jupyter notebook
```

## Listing 2: Open Lesson

To open the tutorial navigate with your browser to:
lesson_3/lesson_3_problem.ipynb

# Bucket Elimination

## Constrained Optimization

Constrained optimization problems are problems for which an objective function $f(x)$ has to be minimized (or maximized), subject to soft and hard constraints.

## Dynamic Programming

- build the solution of a problem incrementally from those of smaller sub-problems
- convenient for constrained optimization as it exploits the underlying structure of the problem
- solve sub-problems locally and propagate only important information

## Bucket Elimination

The dynamic programming procedures to solve constrained optimization.

# Assignments

- Your assignments for this session are: *lesson_3/lesson_3_problem.ipynb*. You will be required to implement the following functions: *constraint_partitioning*, *main_bucket_elimination*, *get_max_table_size* and *evaluate_soft_constraints*
- In the following you can find the pseudo-code for the first function (*constraint_partitioning*) and a detailed description of the necessary step for the *main_bucket_elimination* process.

# constraint_partitioning

**Input:** *bucket_elimination*, *variable_order*, *soft_constraints*, *hard_constraints*
**Output:** *bucket_elimination*

```
1: Initialize the list of the already assigned constraints
2: for variable in (reverse) order do
3:     Initialize the list of the constraints for the current bucket
4:     for constraint in soft_constraints do
5:         Select the constraints that belong to the current bucket
6:     for constraint in hard_constraints do
7:         Select the constraints that belong to the current bucket
8:     bucket ← new Bucket(soft_constraints, hard_constraints)
9:     Add (bucket) to bucket_elimination
10: return bucket_elimination
```

# main_bucket_elimination

**Input:** *problem_name*, *problem_definition*
**Output:** *None*

1: Initialize the *bucket_elimination* object
2: CONSTRAINT_PARTITIONING(*bucket_elimination*, *variable_order*, *soft_constraints*, *hard_constraints*)
3: Process all the bucket *(Processing Step)*
4: Compute the tables
5: Propagate the value to obtain the assignment *(Propagation Step)*
6: Evaluate the soft constraints for the given assignment
7: Compute the max table size
8: Report the results