Nicolas Melaerts
Kenza Khemar
Master in Computer Science
UMons

Subject: Cover Letter for Revised Final Report

Dear Professor,

Following your feedback on our final report, I have submitted a revised version on Moodle. Please find below a brief summary of the changes made:

- Corrected a grammatical mistake ("shows" instead of "show").
- Removed the tetracube IDs as suggested, since they are unnecessary and removing them simplifies the system.
- Added an explanation justifying why I did not use transformation matrices for rotations.

Modifications Summary

1. Changes in `PUZZLE.lp`
I rewrote the encoding to directly manipulate tetracube *types* instead of using numeric IDs.

Main changes:

- Removed all ID-based predicates like `tetracubeID(1..n)` and `assignType(ID, Type)`.
- Used the `Type` directly in predicates like `position`, `occupied`, etc.
- Adapted rules accordingly: e.g., `pieceGrid(P,G)` became `typeGrid(Type,G)` for grid type 3.
- Updated constraints to work with types (e.g., ensuring 4 pieces per grid).
- Updated display predicates: `fullPosition` and `hintPosition` now take `Type` as first argument.

Advantages:

- Simpler and more direct code.
- Better readability (clearer which piece type is used).
- Removed an unnecessary mapping layer (ID $\rightarrow$ Type).

2. Changes in report

- Change ASP Encoding section
- Change Interpreting Clingo Output section

3. Update GitHub repository
Added this document, the updated report, and the modifications to `PUZZLE.lp`.

Additional Note on Transformations

I would also like to clarify one design choice regarding the representation of rotations. In my implementation, I did not use transformation matrices to compute the rotations of the tetracubes, as suggested in the course. Instead, I explicitly listed the coordinates for each rotation manually, as shown in the code below:

```
% O tetracube
% Rotation 1
cube("O",1,0,0,0).cube("O",1,0,0,1).cube("O",1,0,1,0).cube("O",1,0,1,1).
% Rotation 2
cube("O",2,0,0,0).cube("O",2,0,0,1).cube("O",2,1,0,0).cube("O",2,1,0,1).
% Rotation 3
cube("O",3,0,0,0).cube("O",3,0,1,0).cube("O",3,1,0,0).cube("O",3,1,1,0).
```

While transformation matrices are a clean and reusable approach, during assignment 3 I opted for this explicit representation, and I found it to be very efficient in terms of runtime. When comparing my execution time with other students who used matrix-based rotations, my implementation appeared to be faster, potentially because it avoids the need for runtime transformations.

Of course, this hypothesis would need to be verified by benchmarking the three different approaches (explicit coordinates, transformation matrices, and procedural generation). However, in the context of this project, the manual approach worked well and allowed for faster development and testing.

Thank you for your time and your valuable feedback.

Sincerely,
Nicolas Melaerts
Kenza Khemar