Write-up: Project 2

- Project Approach
    - In light of this project, it, therefore, came up with a Prolog program that is able to produce a valid work schedule given a number of constraints. The schedule covers morning, evening, and night shifts that are manned by employees on appropriate workstations.
    - Constraints involved ensuring each employee works exactly one shift, no forbidden shifts or workstations are assigned, and the number of employees per workstation stays within min/max limits. Idle workstations per shift were also excluded.
- Project Organization
    - The code can be described with logical components present in the schedule.pl file.
    - Data Gathering: Has predicates such as findall which is used to get employees and employees' workstations.
    - Active workstations: Excludes inactive workstations for all shifts.
    - Assignment Logic: Allocates employees to workstations while adhering to all conditions.
    - Verification: Controls the number of employees on each workstation to ensure that it complies with the minimum and maximum levels.
    - Output Formatting: Builds up the structure of the plan to be Morning, Evening and Night.
- Problems encountered
    - The most important challenge in this project was to impose various shifts determinism to obtain fixed determinants; Here, employee ordering affected the final output
    - Management of complexity: Handling of many constraints needs more careful design of the predicates than the management of the dimensions can take.
- Solutions Implemented
    - Deterministic Selection: The order of the employees was maintained as it was creating confusion. Sorting was entirely avoided. For employees the input order was maintained to ensure they correspond to debug.
- Lessons Learned
    - This project help reinforced the idea of controlling search space and order in Prolog.
    - I was able to gain a deeper understanding of Prolog's specific backtracking and its implications for solution reliability.
- Reversibility of Predicates
    - Certain predicates such as plan/1 and select employees/5 are hard to reverse. In other words, the predicates are more geared towards producing schedules from underlying facts than to drawing facts from a finished

schedule. For example, the execution of plan/1 is based on searching through possible employee assignments in order to finalize a valid plan. It is unrealistic to expect the clause to be used the other way around: A plan is given and the constraints are expected to be deduced. It is also the case with select employees/5 because under certain requirements employees are chosen and it is impossible to restore the list of candidates from the partial assignment. These predicates are more based on forward logic and impulsive backtracking and for that reason are one directional