Academic Year	2023
Semester	1
Module Number	CM2602
Module Title	Artificial Intelligence
Assessment Method	CourseWork
Deadline (time and date)	20th December 2023
Word Limit	10000
Module Co-Ordinator	Mr. Nipuna Senanayake

Name	Seth Rajarathne
IIT Student ID	20211344
RGU ID	2237948

Table of Contents

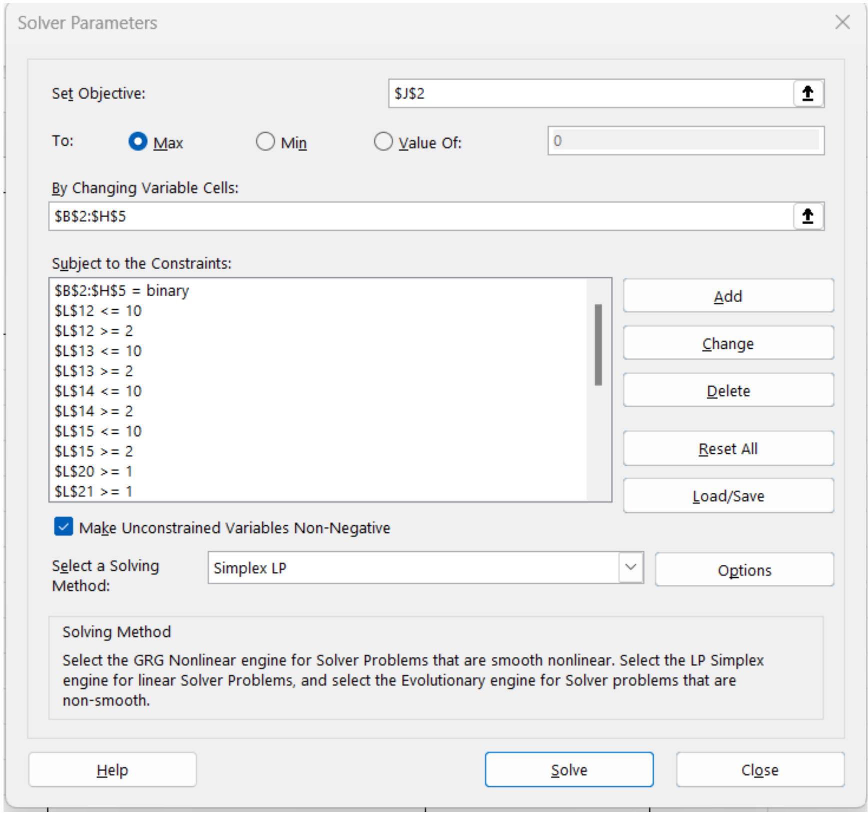
Question 1

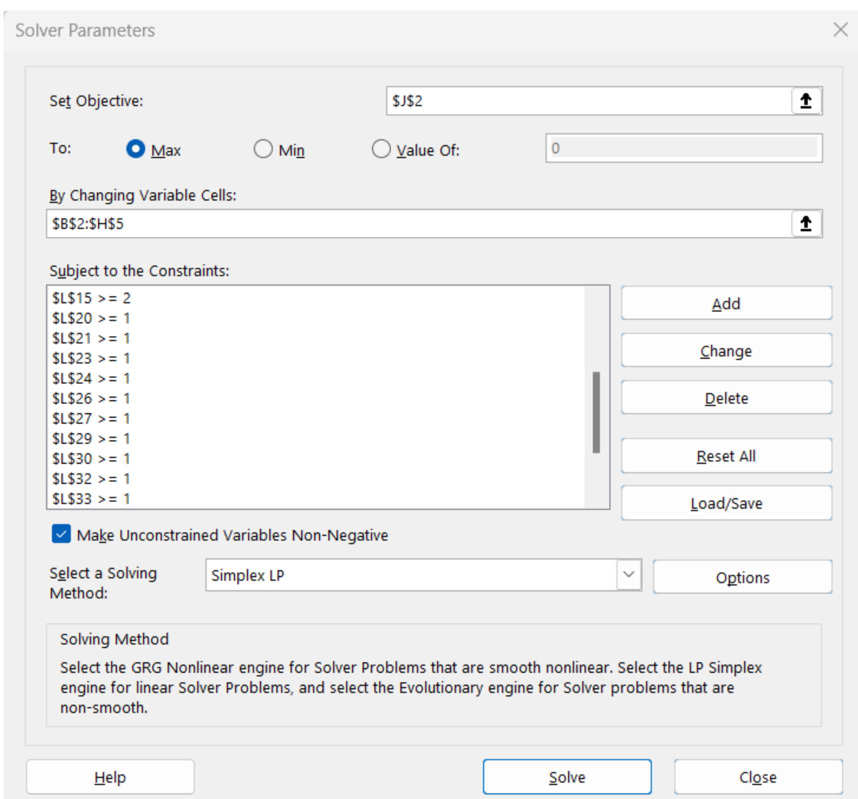
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday		Total Shifts
Employee 1	1	1	1	1	1	1	1	Morning	
Employee 2	1	1	1	1	1	1	1	Morning	
Employee 3	1	1	1	1	1	1	1	Evening	
Employee 4	1	1	1	1	1	1	1	Evening	28

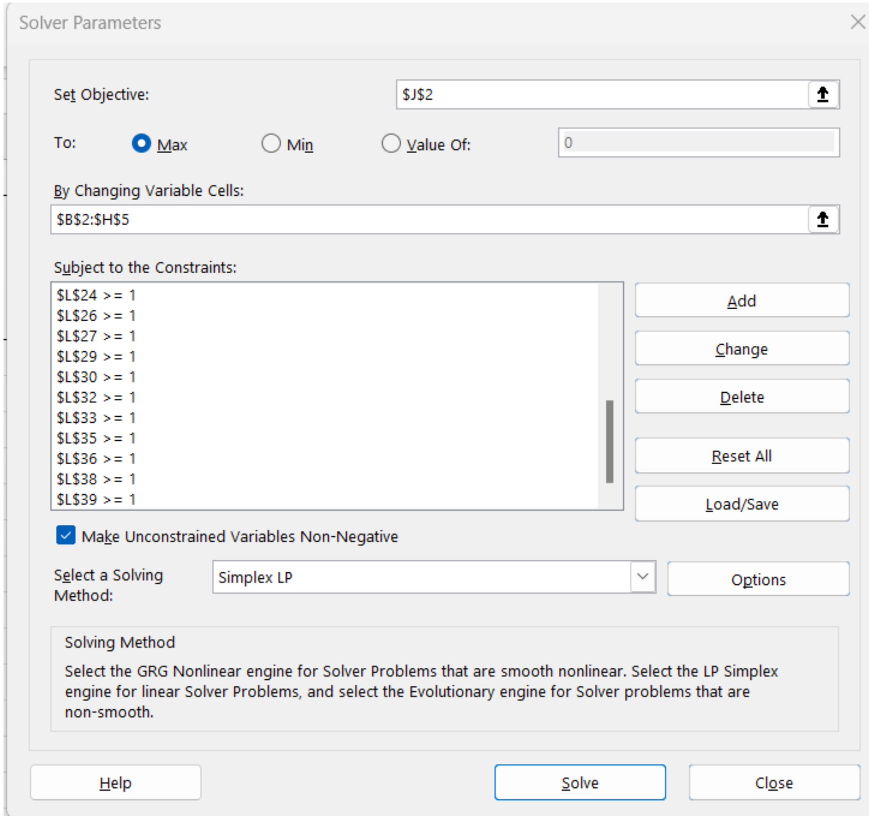
Total shifts for each employee	
Employee 1	7
Employee 2	7
Employee 3	7
Employee 4	7

Employees assigned for each shift	
Monday morning	2
Monday evening	2
Tuesday Morning	2
Tuesday Evening	2
Wednesday Morning	2
Wednesday evening	2
Thursday Morning	2
Thursday evening	2
Friday Morning	2
Friday Evening	2
Saturday Morning	2
Saturday Evening	2
Sunday Morning	2
Sunday Evening	2

Constrains


Solver Parameters
Set Objective:
$J$2
To:
Max
Min
Value Of:
0
By Changing Variable Cells:
$B$2:$H$5
Subject to the Constraints:
$B$2:$H$5 = binary
$L$12 <= 10
$L$12 >= 2
$L$13 <= 10
$L$13 >= 2
$L$14 <= 10
$L$14 >= 2
$L$15 <= 10
$L$15 >= 2
$L$20 >= 1
$L$21 >= 1
Add
Change
Delete
Reset All
Load/Save
Make Unconstrained Variables Non-Negative
Select a Solving Method:
Simplex LP
Options
Solving Method
Select the GRG Nonlinear engine for Solver Problems that are smooth nonlinear. Select the LP Simplex engine for linear Solver Problems, and select the Evolutionary engine for Solver problems that are non-smooth.
Help
Solve
Close


Solver Parameters
Set Objective:
$J$2
To:
Max
Min
Value Of:
0
By Changing Variable Cells:
$B$2:$H$5
Subject to the Constraints:
$L$15 >= 2
$L$20 >= 1
$L$21 >= 1
$L$23 >= 1
$L$24 >= 1
$L$26 >= 1
$L$27 >= 1
$L$29 >= 1
$L$30 >= 1
$L$32 >= 1
$L$33 >= 1
Add
Change
Delete
Reset All
Load/Save
Make Unconstrained Variables Non-Negative
Select a Solving Method:
Simplex LP
Options
Solving Method
Select the GRG Nonlinear engine for Solver Problems that are smooth nonlinear. Select the LP Simplex engine for linear Solver Problems, and select the Evolutionary engine for Solver problems that are non-smooth.
Help
Solve
Close


Solver Parameters
Set Objective:
$J$2
To:
Max
Min
Value Of:
0
By Changing Variable Cells:
$B$2:$H$5
Subject to the Constraints:
$L$24 >= 1
$L$26 >= 1
$L$27 >= 1
$L$29 >= 1
$L$30 >= 1
$L$32 >= 1
$L$33 >= 1
$L$35 >= 1
$L$36 >= 1
$L$38 >= 1
$L$39 >= 1
Add
Change
Delete
Reset All
Load/Save
Make Unconstrained Variables Non-Negative
Select a Solving Method:
Simplex LP
Options
Solving Method
Select the GRG Nonlinear engine for Solver Problems that are smooth nonlinear. Select the LP Simplex engine for linear Solver Problems, and select the Evolutionary engine for Solver problems that are non-smooth.
Help
Solve
Close

Question 2

Part A

Engineering Design:

- **Scope:** The ontology will assist engineers with mechanical system design and simulation, product efficiency optimization, and consistency compliance.

- **Competency Questions:**
  1. Can you explain the role of ontology in supporting simulation activities for mechanical systems?
  2. How does the ontology help ensure consistency in compliance with regulations and design standards?
  3. How can the ontology be adapted for various mechanical engineering projects, considering their unique requirements?

Knowledge Discovery:

- **Scope:** The ontology will be applied by researchers to assist with knowledge discovery, conceptual search, and data integration in mechanical engineering, enabling improvements in both product design and manufacturing.

- **Competency Questions:**
  1. How would you apply ontology to support researchers in the field of mechanical engineering for knowledge discovery?
  2. In what ways does ontology facilitate data integration for improved insights in mechanical engineering research?
  3. Discuss the role of ontology in enabling better manufacturing processes and optimization in the field of mechanical engineering.

Quality Control:

- **Scope:** To further enhance the quality and safety of their products, manufacturers will employ the ontology for monitoring quality parameters, inspection procedures, and adherence to industry regulations.
  .
- **Competency Questions:**
  1. How would you integrate the ontology into manufacturing processes to monitor and enhance product quality?
  2. In what ways does ontology support manufacturers in ensuring adherence to industry regulations for quality and safety?
  3. Discuss the potential impact of employing ontology on enhancing both the quality and safety aspects of manufactured products.

Education and Training:

- **Scope:** The ontology will assist experts, instructors, and students of mechanical engineering better understand mechanical engineering principles and best practices.

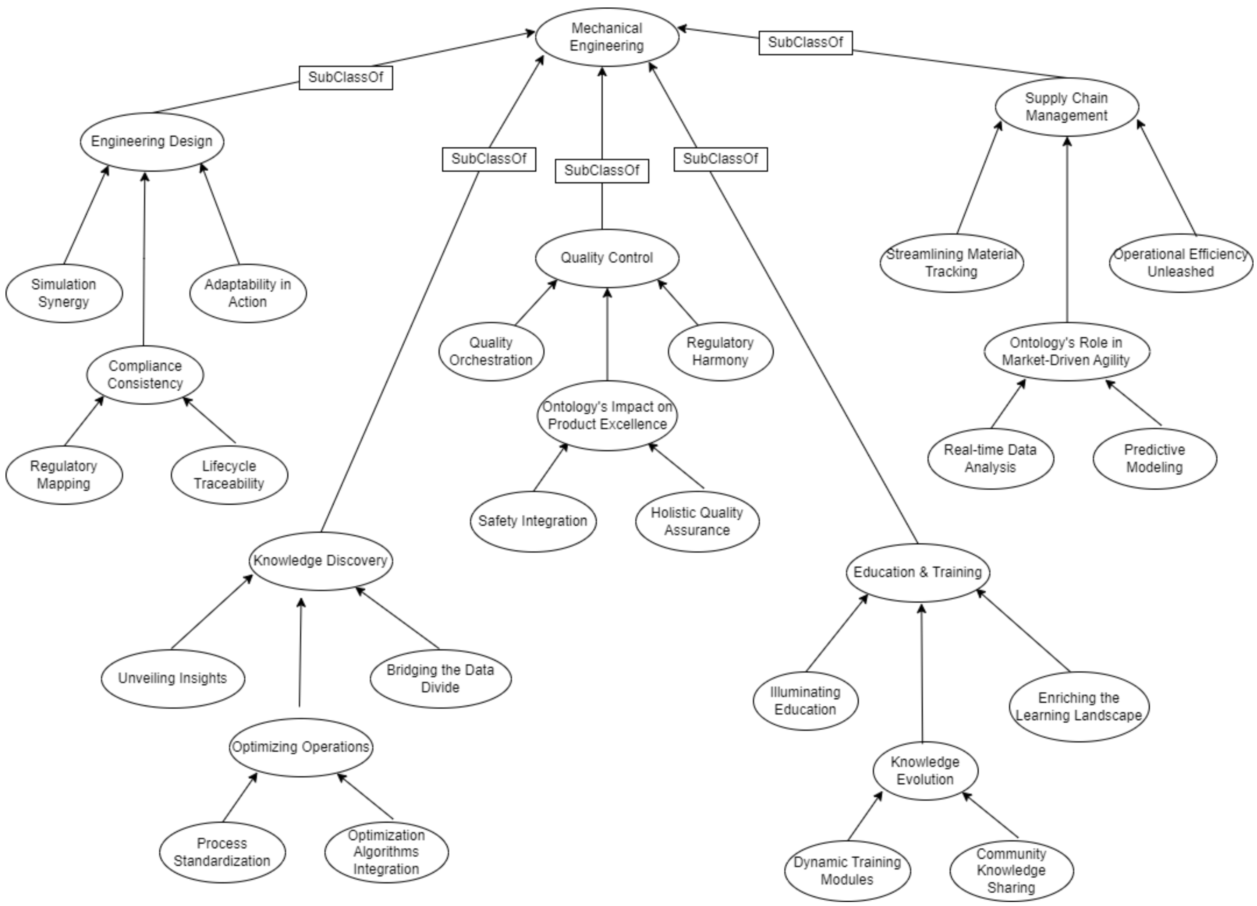
- **Competency Questions:**
  1. How would you utilize ontology to improve the understanding of mechanical engineering principles among students and instructors?
  2. Explain how ontology contributes to enhancing the learning experience and comprehension for students studying mechanical engineering.
  3. How can ontology be integrated into training programs to help experts and professionals stay updated on mechanical engineering principles and best practices?

Supply Chain Management:

- **Scope:** The ontology optimizes the mechanical engineering supply chain by tracking materials, improving collaboration, and reducing costs while enhancing operational efficiency and responsiveness to market demands.

- **Competency Questions:**
  1. How does ontology aid in making the mechanical engineering supply chain more responsive to market demands?
  2. How would you use the ontology to optimize material tracking within the mechanical engineering supply chain?
  3. Discuss how ontology can enhance operational efficiency in the mechanical engineering supply chain.

Part B



Part C

```xml
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <!-- OWL Header -->
  <owl:Ontology rdf:about="http://www.linkeddatatools.com/MechanicalEngineering">
    <dc:title>Mechanical Engineering</dc:title>
    <dc:description>All mechanical engineering parts</dc:description>
  </owl:Ontology>

  <!-- Defining the subclasses under Thing -->
```

```xml
  <!-- OWL Subclass definition - Engineering Design -->
                                                                      <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#EngineeringDesign
">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:label>Engineering Design</rdfs:label>
    <rdfs:comment>Assist engineers with mechanical system design</rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Compliance Consistency -->
                                                                      <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#ComplianceConsist
ency">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
                                                                 <rdfs:subClassOf
rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#EngineeringDes
ign"/>
    <rdfs:label>Compliance Consistency</rdfs:label>
    <rdfs:comment>Ensure consistency in compliance</rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Regulatory Mapping -->
                                                                      <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#RegulatoryMapping
">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
                                                                 <rdfs:subClassOf
rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#ComplianceCons
istency"/>
    <rdfs:label>Regulatory Mapping</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Lifecycle Traceability -->
                                                                      <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#LifecycleTraceabi
lity">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
                                                                 <rdfs:subClassOf
rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#ComplianceCons
istency"/>
    <rdfs:label>Lifecycle Traceability</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>
```

```xml
  <!-- OWL Subclass definition - Adaptability in Action -->
  <owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#AdaptabilityInAction">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#EngineeringDesign"/>
    <rdfs:label>Adaptability In Action</rdfs:label>
    <rdfs:comment>Adapting for various projects</rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Simulation Synergy -->
  <owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#SimulationSynergy">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#EngineeringDesign"/>
    <rdfs:label>Simulation Synergy</rdfs:label>
    <rdfs:comment>supporting mechanical system</rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Knowledge Discovery -->
  <owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#KnowledgeDiscovery">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:label>Knowledge Discovery</rdfs:label>
    <rdfs:comment>Assist researchers to discover knowledge</rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Optimizing Operations -->
  <owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#OptimizingOperations">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#KnowledgeDiscovery"/>
    <rdfs:label>Optimizing Operations</rdfs:label>
    <rdfs:comment>Enable better manufacturing process</rdfs:comment>
```

```
  </owl:Class>

  <!-- OWL Subclass definition - Process Standardization -->
  <owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#ProcessStandardization">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#OptimizingOperations"/>
    <rdfs:label>Process Standardization</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Optimization Algorithms Integration -->
  <owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#OptimizationAlgorithmsIntegration">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#OptimizingOperations"/>
    <rdfs:label>Optimization Algorithms Integration</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Unveiling Insights -->
  <owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#UnveilingInsights">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#KnowledgeDiscovery"/>
    <rdfs:label>Unveiling Insights</rdfs:label>
    <rdfs:comment>provide insights for mechanical engineering insight</rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Bridging the Data Divide -->
  <owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#BridgingTheDataDivide">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
```

```
                                                                  <rdfs:subClassOf
rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#KnowledgeDisco
very"/>
    <rdfs:label>Bridging the Data Divide</rdfs:label>
    <rdfs:comment>Support engineers in discovering knowledge</rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Quality Control -->
                                                                        <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#QualityControl">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:label>Quality Control</rdfs:label>
    <rdfs:comment>Enhancing the quality and saftey of the product</rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Quality Orchestration -->
                                                                        <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#QualityOrchestrat
ion">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
                                                                  <rdfs:subClassOf
rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#QualityControl
"/>
    <rdfs:label>Quality Orchestration</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Ontology's Impact on Product Excellence -->
                                                                        <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#OntologysImpactOn
ProductExcellence">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
                                                                  <rdfs:subClassOf
rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#QualityControl
"/>
    <rdfs:label>Ontology's Impact on Product Excellence</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Holistic Quality Assurance -->
                                                                        <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#HolisticQualityAs
surance">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
```

```xml
                                                                            <rdfs:subClassOf
rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#OntologysImpac
tOnProductExcellence"/>
    <rdfs:label>Holistic Quality Assurance</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

 <!-- OWL Subclass definition - Safty Intrgration -->
                                                                                  <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#SaftyIntrgration"
>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
                                                                            <rdfs:subClassOf
rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#OntologysImpac
tOnProductExcellence"/>
    <rdfs:label>Safty Intrgration</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Regulatory Harmony -->
                                                                                  <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#RegulatoryHarmony
">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
                                                                            <rdfs:subClassOf
rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#QualityControl
"/>
    <rdfs:label>Regulatory Harmony</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Education and Training -->
                                                                                  <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#EducationAndTrain
ing">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:label>Education and Training</rdfs:label>
     <rdfs:comment>Support users to understand mechanical engineeering principals
better</rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Illuminating Education -->
```

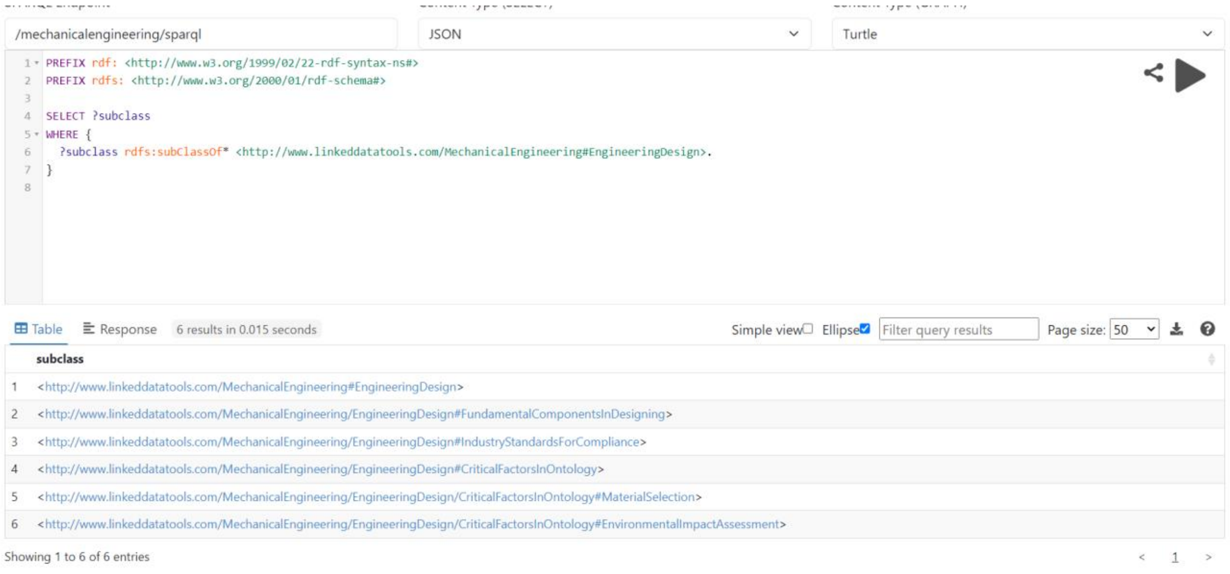
```xml
<owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#IlluminatingEducation">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#EducationAndTraining"/>
    <rdfs:label>Illuminating Education</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Knowledge Evolution -->
<owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#KnowledgeEvolution">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#EducationAndTraining"/>
    <rdfs:label>Knowledge Evolution</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Dynamic Training Modules -->
<owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#DynamicTrainingModules">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#KnowledgeEvolution"/>
    <rdfs:label>Dynamic Training Modules</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Community Knowledge Sharing -->
<owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#CommunityKnowledgeSharing">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#KnowledgeEvolution"/>
    <rdfs:label>Community Knowledge Sharing</rdfs:label>
    <rdfs:comment></rdfs:comment>
```

```
  </owl:Class>

  <!-- OWL Subclass definition - Enriching the Learning Landscape -->
  <owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#EnrichingTheLearningLandscape">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#KnowledgeEvolution"/>
    <rdfs:label>Enriching the Learning Landscape</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Supply Chain Management -->
  <owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#SupplyChainManagement">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:label>Supply Chain Management</rdfs:label>
    <rdfs:comment>Optimize suply chain by tracking materials</rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Streamlining Material Tracking -->
  <owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#StreamliningMaterialTracking">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#SupplyChainManagement"/>
    <rdfs:label>Streamlining Material Tracking</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Ontology's Role in Market-Driven Agility -->
  <owl:Class rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#OntologysRoleInMarketDrivenAgility">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#SupplyChainManagement"/>
    <rdfs:label>Ontology's Role in Market-Driven Agility</rdfs:label>
```

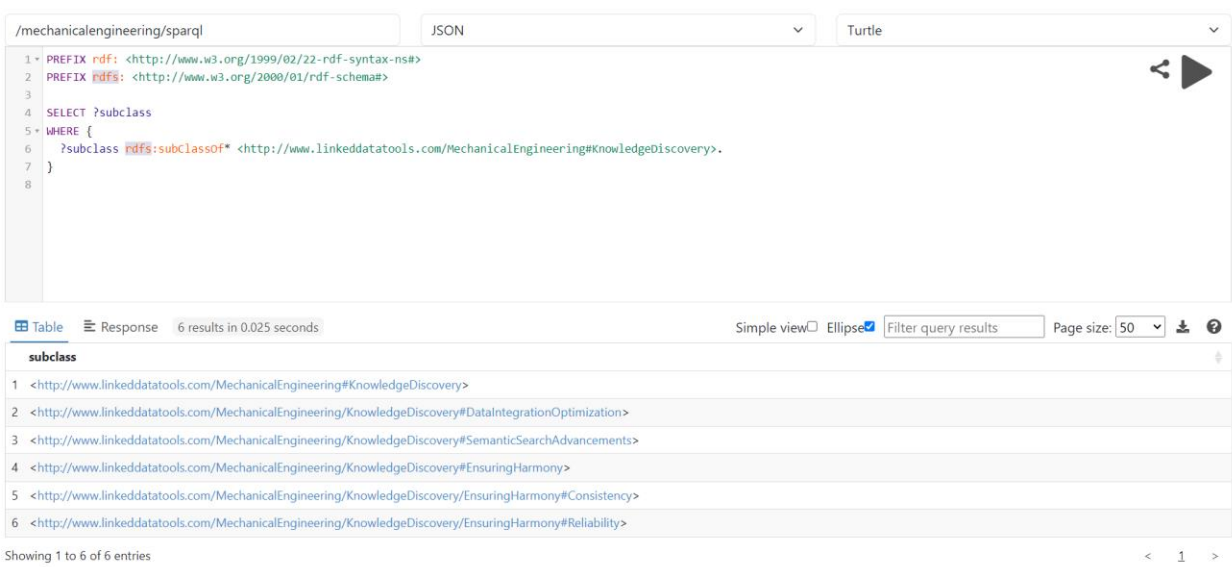
```
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Real-time Data Analysis -->
                                                                      <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#RealTimeDataAnaly
sis">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
                                                                <rdfs:subClassOf
rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#OntologysRoleI
nMarketDrivenAgility"/>
    <rdfs:label>Real-time Data Analysis</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Predictive Modeling -->
                                                                      <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#PredictiveModelin
g">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
                                                                <rdfs:subClassOf
rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#OntologysRoleI
nMarketDrivenAgility"/>
    <rdfs:label>Predictive Modeling</rdfs:label>
    <rdfs:comment></rdfs:comment>
  </owl:Class>

  <!-- OWL Subclass definition - Operational Efficiency Unleashed -->
                                                                      <owl:Class
rdf:about="http://www.linkeddatatools.com/MechanicalEngineering#OperationalEffici
encyUnleashed">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
                                                                <rdfs:subClassOf
rdf:resource="http://www.linkeddatatools.com/MechanicalEngineering#SupplyChainMan
agement"/>
    <rdfs:label>Operational Efficiency Unleashed</rdfs:label>
    <rdfs:comment>Enhance suply chain efficiency</rdfs:comment>
  </owl:Class>

</rdf:RDF>
```

Part D

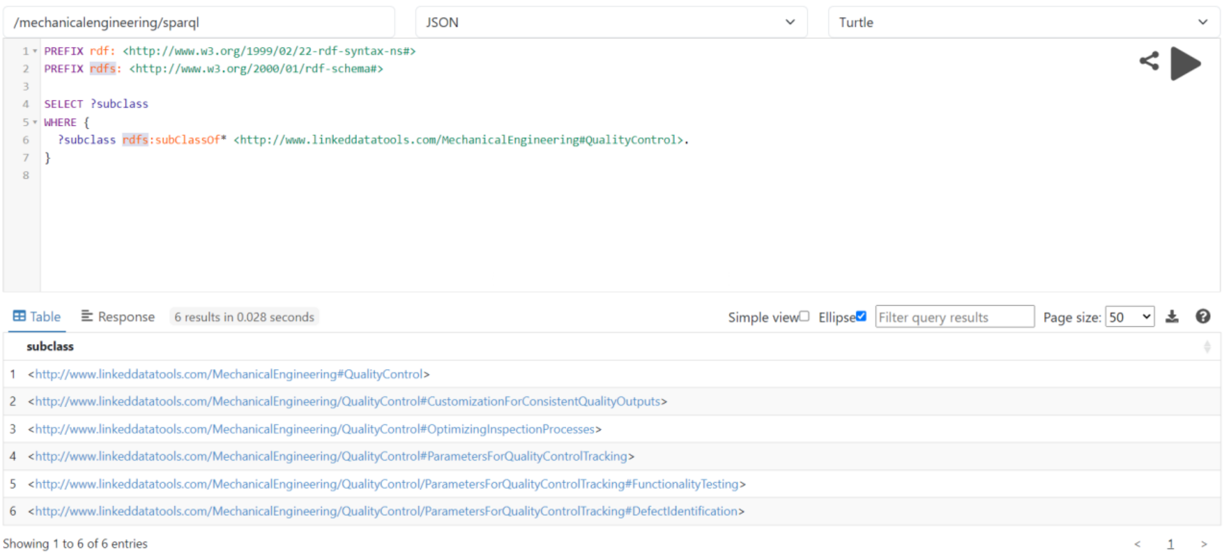
1. What specific areas fall under 'Engineering Design' in the 'Mechanical Engineering' ontology?



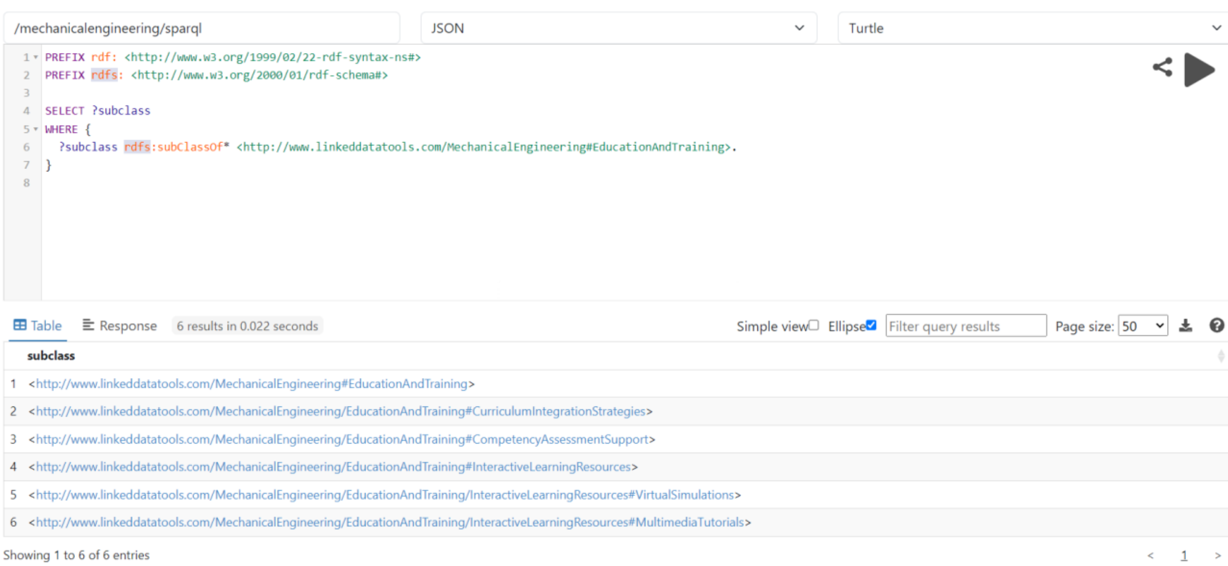
2. Which specialized domains or subjects are encompassed within the scope of 'Knowledge Discovery' in the context of the 'Mechanical Engineering' ontology?



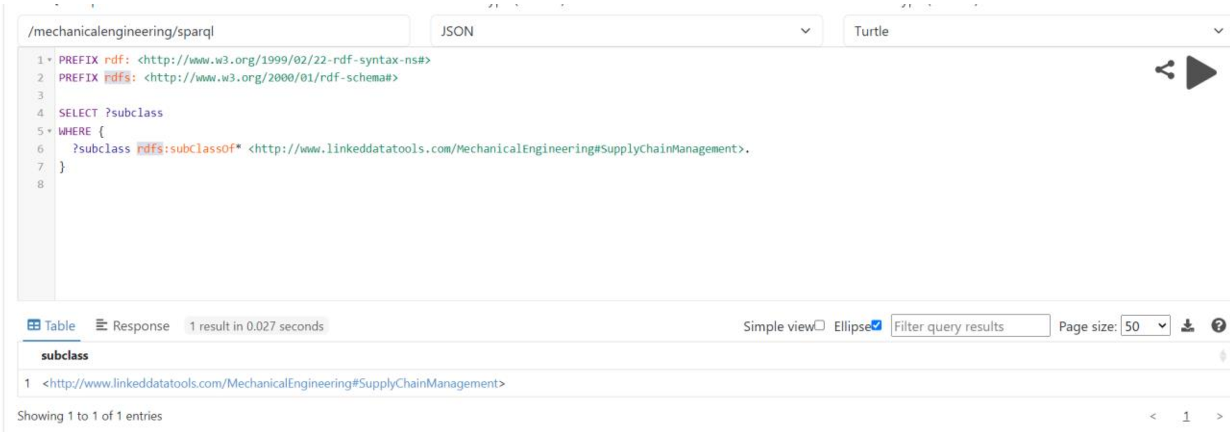
3. What specific aspects or categories fall under 'Quality Control' within the 'Mechanical Engineering' ontology?



4. What are the various subdomains or specialized topics covered within 'Education and Training' in the 'Mechanical Engineering' ontology?



5. Which specific areas or subcategories are included within the domain of 'Supply Chain Management' in the 'Mechanical Engineering' ontology?



Question 3

Main Code

```python
import random
import statistics

class Maze:
    def __init__(self, maze):
        self.maze = maze
        self.rows = len(maze)
        self.cols = len(maze[0])

    def is_valid_move(self, row, col):
        return 0 <= row < self.rows and 0 <= col < self.cols and self.maze[row][col] != 3

    def get_neighbors(self, row, col):
        neighbors = []
        directions = [
            (0, 1), (1, 0), (0, -1), (-1, 0),  # right, down, left, up
            (1, 1), (1, -1), (-1, 1), (-1, -1)  # diagonals
        ]
        for dr, dc in directions:
            new_row, new_col = row + dr, col + dc
            if self.is_valid_move(new_row, new_col):
                neighbors.append((new_row, new_col))
        return neighbors

class DFSResult:
    def __init__(self):
        self.visited = set()
        self.path = []
        self.time_to_goal = 0
        self.path_length = 0

def dfs_recursive(maze, current, goal, result):
    result.visited.add(current)
    result.path.append(current)

    if current == goal:
        result.time_to_goal = len(result.visited)
        return True

    neighbors = sorted(maze.get_neighbors(current[0], current[1]))
    for neighbor in neighbors:
        if neighbor not in result.visited and dfs_recursive(maze, neighbor, goal, result):
            return True

    result.path.pop()
    return False
```

```
def perform_dfs(maze, start, goal):
    result = DFSResult()
    dfs_recursive(maze, start, goal, result)
    result.path_length = len(result.path)
    return result

def manhattan_heuristic(node, goal):
    return abs(node[0] - goal[0]) + abs(node[1] - goal[1])

class AStarResult:
    def __init__(self):
        self.visited = set()
        self.path = []
        self.time_to_goal = 0
        self.path_length = 0

def astar_search(maze, start, goal):
    result = AStarResult()
    open_set = [(start, 0)]  # Priority queue with initial cost

    while open_set:
        current, cost_so_far = open_set.pop(0)
        result.visited.add(current)
        result.path.append(current)

        if current == goal:
            result.time_to_goal = cost_so_far
            result.path_length = len(result.path)
            return result

        neighbors = maze.get_neighbors(current[0], current[1])
        for neighbor in neighbors:
            if neighbor not in result.visited:
                heuristic_cost = manhattan_heuristic(neighbor, goal)
                total_cost = cost_so_far + 1 + heuristic_cost  # 1 is the
cost to move to a neighbor
                open_set.append((neighbor, total_cost))
                open_set.sort(key=lambda x: x[1])  # Sort by total cost

    return result

def generate_random_maze():
    maze_data = [[0] * 6 for _ in range(6)]

    start_row = random.randint(0, 3)
    start_col = random.randint(0, 1)
    start_point = (start_row, start_col)

    goal_row = random.randint(2, 5)
    goal_col = random.randint(4, 5)
    goal_point = (goal_row, goal_col)

    available_nodes = set(range(36)) - {start_point[0] * 6 + start_point[1],
goal_point[0] * 6 + goal_point[1]}
    barrier_nodes = random.sample(list(available_nodes), 4)

    maze_data[start_row][start_col] = 1
```

```python
    maze_data[goal_row][goal_col] = 2
    for barrier_node in barrier_nodes:
        barrier_row, barrier_col = divmod(barrier_node, 6)
        maze_data[barrier_row][barrier_col] = 3

    return maze_data

def analyze_results(results, algorithm_name):
    completeness = all(result.time_to_goal > 0 for result in results)
    optimality = all(result.time_to_goal == result.path_length for result in results)
    solution_times = [result.time_to_goal for result in results]
    path_lengths = [result.path_length for result in results]

    print(f"\nAnalysis for {algorithm_name}:")
    print(f"Completeness: {completeness}")
    print(f"Optimality: {optimality}")
    print(f"Mean Solution Time: {statistics.mean(solution_times)} minutes")
    print(f"Variance in Solution Time: {statistics.variance(solution_times)}")


# Repeat the process for three random mazes
dfs_results = []
astar_results = []

for _ in range(3):
    maze_data = generate_random_maze()
    maze = Maze(maze_data)

    start_point = (random.randint(0, 3), random.randint(0, 1))
    goal_point = (random.randint(2, 5), random.randint(4, 5))

    # Perform DFS
    dfs_result = perform_dfs(maze, start_point, goal_point)
    dfs_results.append(dfs_result)

    # Perform A*
    astar_result = astar_search(maze, start_point, goal_point)
    astar_results.append(astar_result)

    # Analyze and print results for each algorithm
    print("\nMaze Data:")
    for row in maze_data:
        print(row)

    print("\nDFS Results:")
    print("Time to Find Goal:", dfs_result.time_to_goal, "minutes")
    print("Final Path:", dfs_result.path)

    print("\nA* Search Results:")
    print("Time to Find Goal:", astar_result.time_to_goal, "minutes")
    print("Final Path:", astar_result.path)

# Analyze and report overall results
analyze_results(dfs_results, "DFS")
analyze_results(astar_results, "A*")
```

Task 1

In task 1, programmer needs to set up a maze using suitable data structure and follow the rules they expect. Programmers should have to design Nodes as “Starting Node”, “Goal Node” and “Barrier Node” to implement the maze. Programmers should have to follow the rules that are given in the specification.

Task 2

In task two, programmers must implement a program to perform Depth-First Search on a randomly organized maze. The rules that must be followed are obstacles cannot be crossed, neighbors of a node must be examined in ascending order, exploration of a node takes one minute, and moves can be made in any direction.

Task 3

The goal of this exercise is to write a function that uses the Manhattan distance calculation to calculate the heuristic cost for each node in a maze. Based on the absolute differences in the x and y coordinates of a node (N) and the goal node, the formula h(N, Goal) determines the distance between them. Specifically, the work entails creating this heuristic function, which evaluates the expected cost from a node to the objective and can be used, for example, in algorithms such as A* search for maze pathfinding.

Task 4

The objective of this task is to apply the heuristic cost function to the A* search algorithm. In a maze, the A* search seeks to identify the best route from a beginning node to a goal node. The Manhattan distance calculation is used to calculate the heuristic cost, which helps prioritize nodes that are predicted to take a more efficient path. The A* search algorithm must be used to accomplish the objective.

Task 5

Finally, the task 5 requires us to make the maze run three times and analyze the two search results in terms of completeness, optimality and time complexity.

Code Output

```
Maze Data:
[0, 0, 0, 0, 0, 3]
[0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 2]
[1, 0, 0, 0, 0, 0]
[0, 3, 3, 0, 0, 0]
[0, 0, 0, 3, 0, 0]

DFS Results:
Time to Find Goal: 29 minutes
Final Path: [(1, 0), (0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (1, 3), (1, 2), (1, 1),

A* Search Results:
Time to Find Goal: 16 minutes
Final Path: [(1, 0), (2, 1), (1, 1), (2, 0), (0, 1), (0, 0), (3, 2), (2, 2), (3, 1),
```

```
Maze Data:
[0, 1, 0, 0, 3, 0]
[0, 0, 0, 0, 3, 0]
[0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0]
[3, 0, 0, 0, 2, 0]
[0, 0, 3, 0, 0, 0]

DFS Results:
Time to Find Goal: 16 minutes
Final Path: [(1, 1), (0, 0), (0, 1), (0, 2), (0, 3), (1, 2), (1, 3), (2, 2), (2, 1),

A* Search Results:
Time to Find Goal: 6 minutes
Final Path: [(1, 1), (2, 2), (1, 2), (2, 1), (2, 0), (0, 2), (2, 3), (1, 0), (0, 1),
```

```
Maze Data:
[1, 0, 0, 0, 0, 0]
[0, 3, 3, 0, 3, 0]
[0, 0, 0, 0, 0, 0]
[0, 3, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 2]

DFS Results:
Time to Find Goal: 22 minutes
Final Path: [(3, 1), (2, 0), (1, 0), (0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (0, 5),

A* Search Results:
Time to Find Goal: 10 minutes
Final Path: [(3, 1), (2, 2), (3, 2), (2, 1), (4, 2), (2, 0), (4, 1), (3, 0), (2, 3),
```

```
Analysis for DFS:
Completeness: True
Optimality: True
Mean Solution Time: 22.333333333333332 minutes
Variance in Solution Time: 42.333333333333336

Analysis for A*:
Completeness: True
Optimality: False
Mean Solution Time: 10.666666666666666 minutes
Variance in Solution Time: 25.333333333333332

Process finished with exit code 0
```

Question 4

```
import numpy as np

import skfuzzy.control as ctrl

import matplotlib.pyplot as plt


def setup_fuzzy_system():

    # Define fuzzy variables

    inputs = {

        'data_redundancy': ctrl.Antecedent(np.arange(0, 101, 1), 'data_redundancy'),

        'degradation_level': ctrl.Antecedent(np.arange(0, 101, 1), 'degradation_level'),

        'error_history': ctrl.Antecedent(np.arange(0, 101, 1), 'error_history')

    }


    outputs = {

        'error_severity': ctrl.Consequent(np.arange(0, 101, 1), 'error_severity',
defuzzify_method='centroid'),

        'error_mitigation': ctrl.Consequent(np.arange(0, 101, 1), 'error_mitigation',
defuzzify_method='centroid')

    }


    # Apply auto-membership function

    for var in inputs.values():

        var.automf(names=['low', 'medium', 'high'])


    for var in outputs.values():

        var.automf(names=['low', 'medium', 'high'])


    # Define fuzzy rules
```

```
    rules = [
        ctrl.Rule(inputs['data_redundancy']['high'] & inputs['degradation_level']['low'] & inputs['error_history']['low'], outputs['error_severity']['high']),
        ctrl.Rule(inputs['data_redundancy']['medium'] & inputs['degradation_level']['medium'] & inputs['error_history']['medium'], outputs['error_severity']['medium']),
        ctrl.Rule(inputs['data_redundancy']['low'] & inputs['degradation_level']['high'] & inputs['error_history']['high'], outputs['error_severity']['low']),
        ctrl.Rule(inputs['data_redundancy']['low'] | inputs['degradation_level']['low'], outputs['error_mitigation']['low']),
        ctrl.Rule(inputs['data_redundancy']['medium'] | inputs['degradation_level']['medium'], outputs['error_mitigation']['medium']),
        ctrl.Rule(inputs['data_redundancy']['high'] | inputs['degradation_level']['high'], outputs['error_mitigation']['high'])
    ]

    # Create control system and simulation object
    ctrl_system = ctrl.ControlSystem(rules)
    sim = ctrl.ControlSystemSimulation(ctrl_system)

    return inputs, outputs, sim

def compute_error_severity(inputs, outputs, sim, data_redundancy_val, degradation_level_val, error_history_val):
    # Set input values
    sim.input['data_redundancy'] = data_redundancy_val
    sim.input['degradation_level'] = degradation_level_val
    sim.input['error_history'] = error_history_val

    # Compute output
    sim.compute()
```

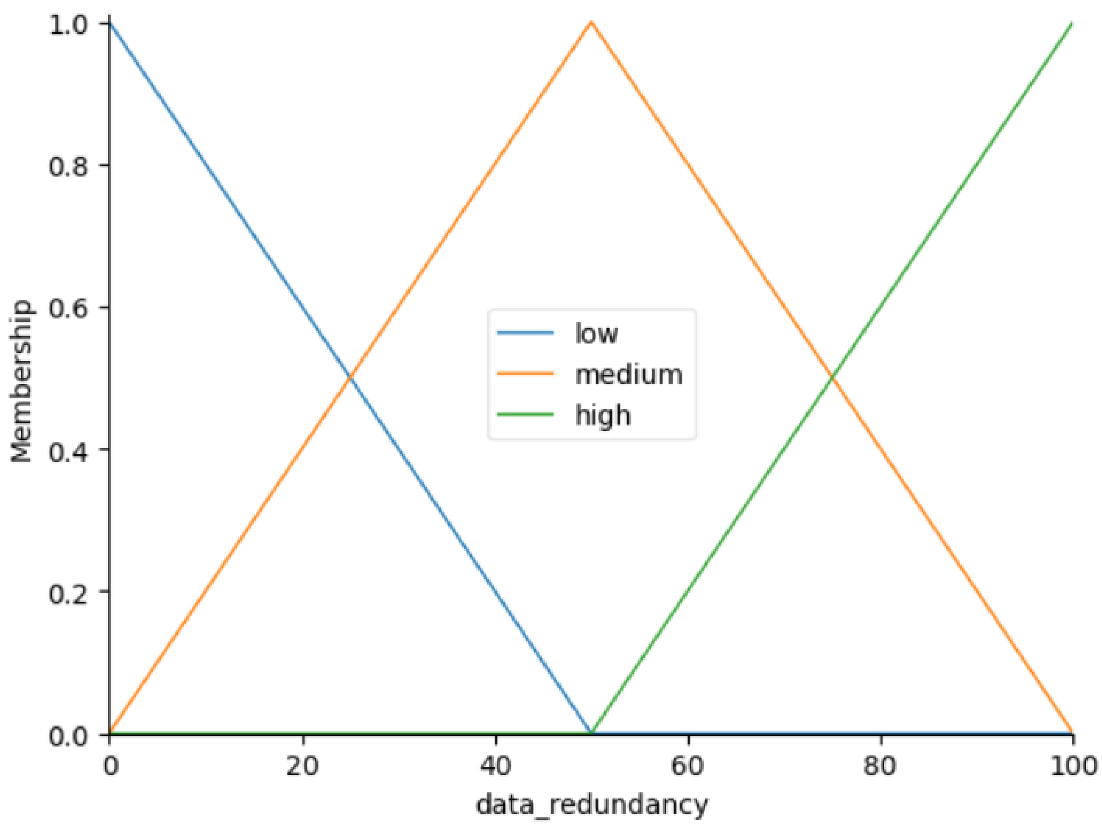
```
    # Return output values
    return sim.output['error_severity'], sim.output['error_mitigation']


def generate_sample_data(inputs, outputs, sim, num_samples=10):
    sample_data = []


    for _ in range(num_samples):
        # Simulate random input values
        data_redundancy_val = np.random.uniform(0, 100)
        degradation_level_val = np.random.uniform(0, 100)
        error_history_val = np.random.uniform(0, 100)


        # Compute error severity
        error_severity_val, error_mitigation_val = compute_error_severity(inputs, outputs, sim,
data_redundancy_val, degradation_level_val, error_history_val)


        # Store the sample data
        sample_data.append({
            'data_redundancy': data_redundancy_val,
            'degradation_level': degradation_level_val,
            'error_history': error_history_val,
            'error_severity': error_severity_val,
            'error_mitigation': error_mitigation_val
        })


    return sample_data


def plot_membership_functions(inputs):
    # Plot membership functions of the inputs
```

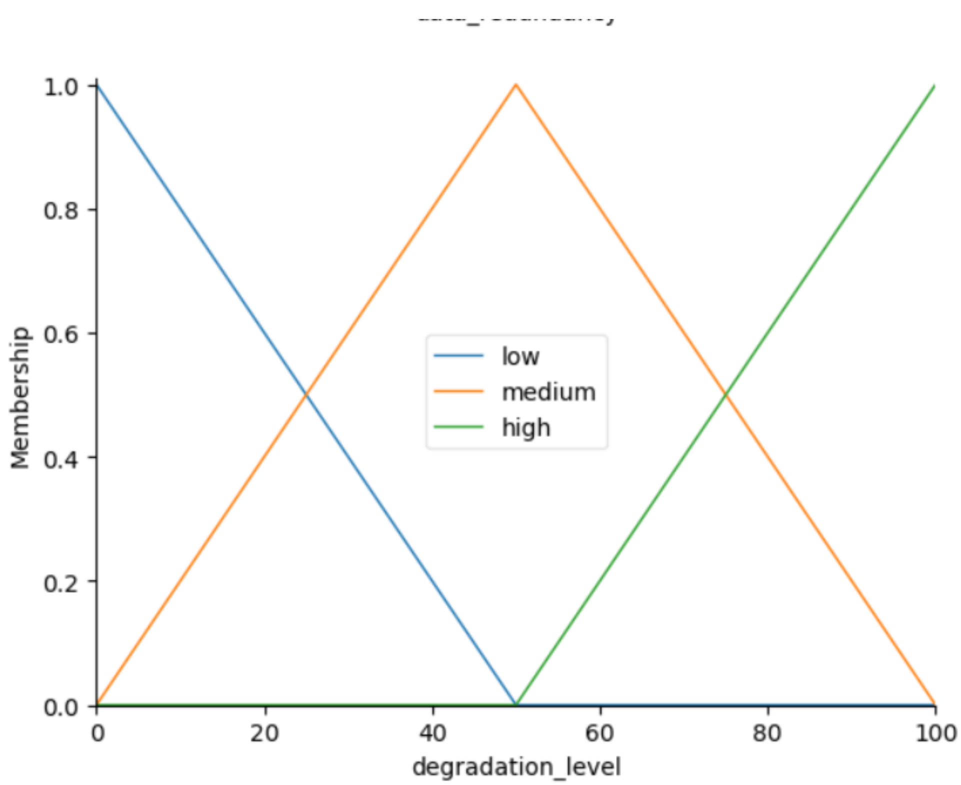
```
    for var in inputs.values():
        var.view()
        plt.show()

if __name__ == "__main__":
    # Setup fuzzy system
    fuzzy_inputs, fuzzy_outputs, fuzzy_sim = setup_fuzzy_system()

    # Run the CLI
    data_redundancy_val = float(input("Enter Data Redundancy (0-100): "))
    degradation_level_val = float(input("Enter Degradation Level (0-100): "))
    error_history_val = float(input("Enter Error History (0-100): "))

    error_severity_val, error_mitigation_val = compute_error_severity(fuzzy_inputs,
fuzzy_outputs, fuzzy_sim, data_redundancy_val, degradation_level_val, error_history_val)

    print(f"\nComputed Results:")
    print(f"Error Severity: {error_severity_val}")
    print(f"Error Mitigation: {error_mitigation_val}")

    # Generate and print a sample dataset with simulated errors
    sample_dataset = generate_sample_data(fuzzy_inputs, fuzzy_outputs, fuzzy_sim,
num_samples=5)
    print("\nGenerated Sample Dataset:")
    for sample in sample_dataset:
        print(sample)

    # Plot membership functions
    plot_membership_functions(fuzzy_inputs)
```

Results

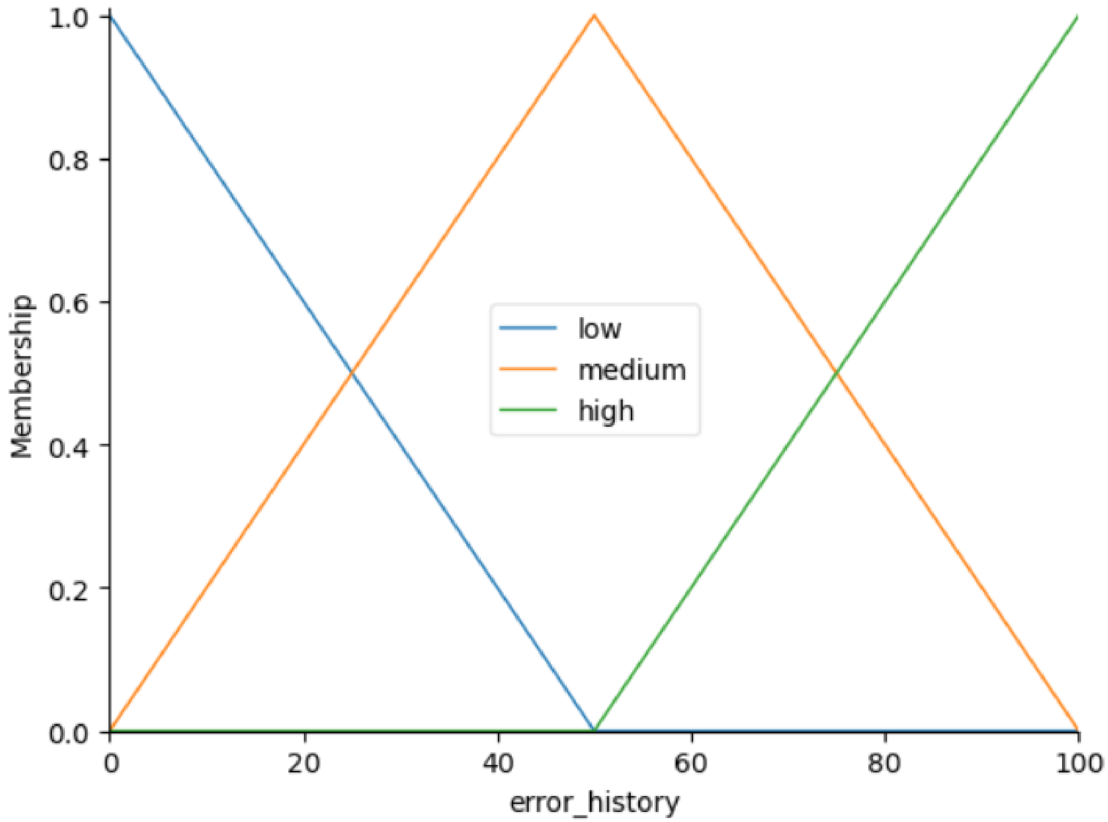
```
Enter Data Redundancy (0-100): 90
Enter Degradation Level (0-100): 10
Enter Error History (0-100): 6

Computed Results:
Error Severity: 71.82222222222221
Error Mitigation: 50.00000000000004
```

```
Generated Sample Dataset:
{'data_redundancy': 35.51627086980261, 'degradation_level': 86.66146513164801, 'error_history': 98.16117653356274, 'error_sever
ity': 28.371647688847418, 'error_mitigation': 55.88370789412628}
{'data_redundancy': 85.20118005167153, 'degradation_level': 49.68743766693078, 'error_history': 12.95081629667527, 'error_sever
ity': 50.00688413869726, 'error_mitigation': 56.99670209845287}
{'data_redundancy': 68.72871989468757, 'degradation_level': 62.394945970801594, 'error_history': 36.55268189898378, 'error_seve
rity': 50.00000000000005, 'error_mitigation': 53.04432999554489}
{'data_redundancy': 13.557423577196648, 'degradation_level': 27.715336401515767, 'error_history': 1.8407609143921366, 'error_se
verity': 49.99999999999998, 'error_mitigation': 41.38105238289559}
{'data_redundancy': 23.11972702941214, 'degradation_level': 82.40973876411809, 'error_history': 6.471556111795007, 'error_sever
ity': 50.000000000000114, 'error_mitigation': 51.71284124219386}
```





1.0
0.8
0.6
0.4
0.2
0.0
Membership
0
20
40
60
80
100
degradation_level
low
medium
high



1.0
0.8
0.6
0.4
0.2
0.0
Membership
low
medium
high
0
20
40
60
80
100
error_history

Explanations

FUZZY SETS:

- Variables and Membership Functions: The system defines three fuzzy variables: data redundancy, deterioration level, and error history. Each variable has three fuzzy sets: "low," "medium," and "high." Membership functions, which are used to represent these sets, determine the degree of membership of a given value inside each set.
- Auto-membership function: This code uses an automated function to automatically generate these membership functions, resulting in enhanced flexibility and adaptability.

RULES:

- The system defines a set of fuzzy rules that connect the input and output variables ("error_severity" and "error_mitigation"). These rules are founded on expert knowledge or empirical facts, and they employ logical operators such as AND and OR.

DECISION MAKING:

• Input values: The system uses three input values during operation: data redundancy, degradation level, and error history.

• Fuzzification: These values are "fuzzified" by using the corresponding membership functions to determine their degree of membership in each fuzzy set.

• Rule activation: The fuzzified input values are used to evaluate each rule. Each rule's degree of activation is determined by the minimum membership value of all its conditions.

• Aggregation: Based on their activation degrees, the activated rules contribute to the output fuzzy sets. The rule outputs are aggregated into a single fuzzy set for each output variable.

• Defuzzification: Finally, the fuzzy output sets are "defuzzified" into crisp (single-valued) integers for "error_severity" and "error_mitigation" using techniques such as centroid defuzzification.

Overall, this system shows how fuzzy logic may be utilized to generate inferences and conclusions based on imprecise or subjective information, simulating human reasoning and giving flexibility in dealing with real-world circumstances where clear boundaries and exact values are frequently unavailable.

References

1. Ekaputra, F., Sabou, M., Serral B, E. and Biffl, S. (n.d.). Ontology-Based Data Integration in Multi-Disciplinary Engineering Environments: A Review TYPE OF PAPER AND KEYWORDS. [online] Available at: https://lirias.kuleuven.be/retrieve/493363 [Accessed 14 Dec. 2023].

2. Szwed, D. (2023). *The Importance of Quality Control and Inspection in Industrial Manufacturing*. [online] Mechanical Power Inc. Available at: https://www.mechanicalpower.net/blog/manufacturing/the-importance-of-quality-control-and-inspection-in-industrial-manufacturing/ [Accessed 13 Dec. 2023].