Assessment Task No. 4			
Topic:	**Monotonic vs. Non-Monotonic Reasoning**	**Week No.**	9
Course Code:	**CSST101**	**Term:**	1st Semester
Course Title:	**Advance Knowledge Representation and Reasoning**	**Academic Year:**	2025-2026
Student Name	Mendoza, Nick Narry S.	**Section**	BSCS 3A
Due date		**Points**	

Learning Outcomes Assessed

After completing this assessment, students should be able to:

1. Explain the key features of **non-monotonic reasoning.**
2. Apply logical reasoning that adapts when new information is added.
3. Construct examples of **argumentation frameworks** showing conflicting knowledge.
4. Demonstrate understanding of belief revision through code or written explanation.

Assessment Title:

"When Logic Changes: Exploring Non-Monotonic Reasoning and Argumentation"

Part I. Conceptual Understanding (20 points)

Instruction:

Answer the following questions briefly but clearly. Each question is worth 4 points.

1. ***Define non-monotonic reasoning in your own words.***

   Non-monotonic reasoning is a type of logic where conclusions can change when new information appears. In normal or classical logic, once something is proven true, it always stays true. But in non-monotonic reasoning, what we believe can be updated or corrected when new facts are discovered. This is similar to how people think in real life, because we often make decisions based on limited or changing information. For example, an AI system might first assume that "birds can fly," but when it learns that penguins are birds that cannot fly, it changes its conclusion. In short, non-monotonic reasoning shows how intelligence can adapt and change beliefs when new evidence is found.

2. ***How does non-monotonic reasoning differ from monotonic reasoning?***

The main difference between monotonic and non-monotonic reasoning is how they deal with new information. In monotonic reasoning, adding new facts does not change what was already concluded — the list of conclusions can only stay the same or increase. For example, in mathematics or classical logic, once something is proven true, it will always remain true no matter what else we learn.

On the other hand, non-monotonic reasoning allows conclusions to change or be taken back when new evidence disagrees with them. This flexible way of thinking is more useful in real-life situations and artificial intelligence, where information can be incomplete or uncertain. It also reflects how people think and change their minds when they discover new facts, helping AI systems act in a more natural and realistic way.

3. ***Give a real-life situation where a conclusion must change after new information is added.***

Imagine you see dark clouds in the sky and think it is going to rain. That conclusion makes sense based on what you see. But then you check the weather report and find out the dark sky is actually smoke from a nearby factory, not rain clouds. You then change your mind and decide it might not rain after all.

This example shows how our reasoning can change when new information appears. It represents non-monotonic reasoning because your first belief (“it will rain”) was updated after learning new facts (“it’s just smoke”). This process of changing beliefs shows how both humans and AI systems adjust their knowledge when new or better information becomes available.

4. ***What is a default rule? Provide one example.***

A default rule is an assumption we make when we don’t have complete information. It helps both people and reasoning systems fill in missing details based on what is usually true, unless there is proof that says otherwise. Default rules make decision-making easier when not all facts are known.

For example, we normally assume that “birds can fly.” This is true for most birds, but if we later learn that the bird is a penguin or an ostrich, we change our conclusion because those are exceptions to the rule.

5. How do argumentation frameworks help AI systems decide between conflicting rules?

   Argumentation frameworks help AI systems deal with conflicting information in a logical way. They show each argument as a point or "node" and show conflicts between them as connections or "attacks." The system then compares the arguments to see which one is stronger, more specific, or better supported by facts, and chooses that as the correct conclusion.
   For example, one rule might say "birds can fly," but another rule says "penguins are birds that cannot fly." The AI uses the argumentation framework to see that the second rule is more specific and therefore stronger. As a result, it concludes that a penguin cannot fly.

Rubric:

- 4 pts – clear, accurate, example-supported answer
- 3 pts – mostly correct, minor errors
- 2 pts – incomplete or unclear
- 1 pt – incorrect or irrelevant

Part II. Laboratory Application (40 points)

Task 1: Belief Revision Simulation (20 points)

Objective: Implement a simple reasoning program in Python or R that revises conclusions when new information is added.

Instructions:

1. Create a program that starts with the rule:
   "If an animal is a bird, assume it can fly."
2. Ask the user to input the animal name.
3. If the animal is a known exception (like penguin or ostrich), revise the conclusion.
4. Display the system's reasoning process step-by-step.

Sample Output:

```
Input: penguin
Reasoning: Penguins are birds.
However, penguins do not fly.
Conclusion: penguins cannot fly.
```

Republic of the Philippines

Laguna State Polytechnic University

Province of Laguna

Rubric:

Criteria	Excellent (5)	Good (4)	Fair (3)	Needs Improvement (2–1)
Correct logical flow	Complete & accurate	Mostly accurate	Minor issues	Incomplete
Program output clarity	Clear reasoning steps	Somewhat clear	Basic	Unclear
Code quality	Efficient & well-structured	Functional	Some redundancy	Many errors
Comments/documentation	Fully commented	Some comments	Few comments	None

```
def reasoning_process(animal):
    print("Reasoning Process:")
    print(f"1. You entered: {animal}")
    print("2. Rule: If an animal is a bird, assume it can fly.")

    birds = ["sparrow", "eagle", "parrot", "penguin", "ostrich"]
    exceptions = ["penguin", "ostrich"]

    if animal.lower() not in birds:
        print(f"3. {animal} is not a bird, so the rule doesn’t apply.")
        print(f"Conclusion: I don’t know if {animal} can fly.")
    elif animal.lower() in exceptions:
        print(f"3. However, {animal} is an exception.")
        print(f"Conclusion: {animal} cannot fly.")
    else:
        print(f"3. {animal} follows the default rule.")
        print(f"Conclusion: {animal} can fly.")

animal_input = input("Enter an animal: ")
reasoning_process(animal_input)
```

```
PS C:\Users\balli\Downloads\CSST101-3A-AY25-26> & C:/Users/balli/AppData/Local/Microsoft/WindowsApps/python3.12.
exe c:/Users/balli/Downloads/CSST101-3A-AY25-26/activity/Task.py
Enter an animal: parrot
Reasoning Process:
1. You entered: parrot
2. Rule: If an animal is a bird, assume it can fly.
3. parrot follows the default rule.
Conclusion: parrot can fly.
PS C:\Users\balli\Downloads\CSST101-3A-AY25-26> & C:/Users/balli/AppData/Local/Microsoft/WindowsApps/python3.12.
exe c:/Users/balli/Downloads/CSST101-3A-AY25-26/activity/Task.py
Enter an animal: penguin
Reasoning Process:
1. You entered: penguin
2. Rule: If an animal is a bird, assume it can fly.
3. However, penguin is an exception.
Conclusion: penguin cannot fly.
PS C:\Users\balli\Downloads\CSST101-3A-AY25-26>
```

Task 2: Argumentation Framework (20 points)

Objective: Create a simple argument diagram showing conflicting knowledge and how the stronger argument prevails.

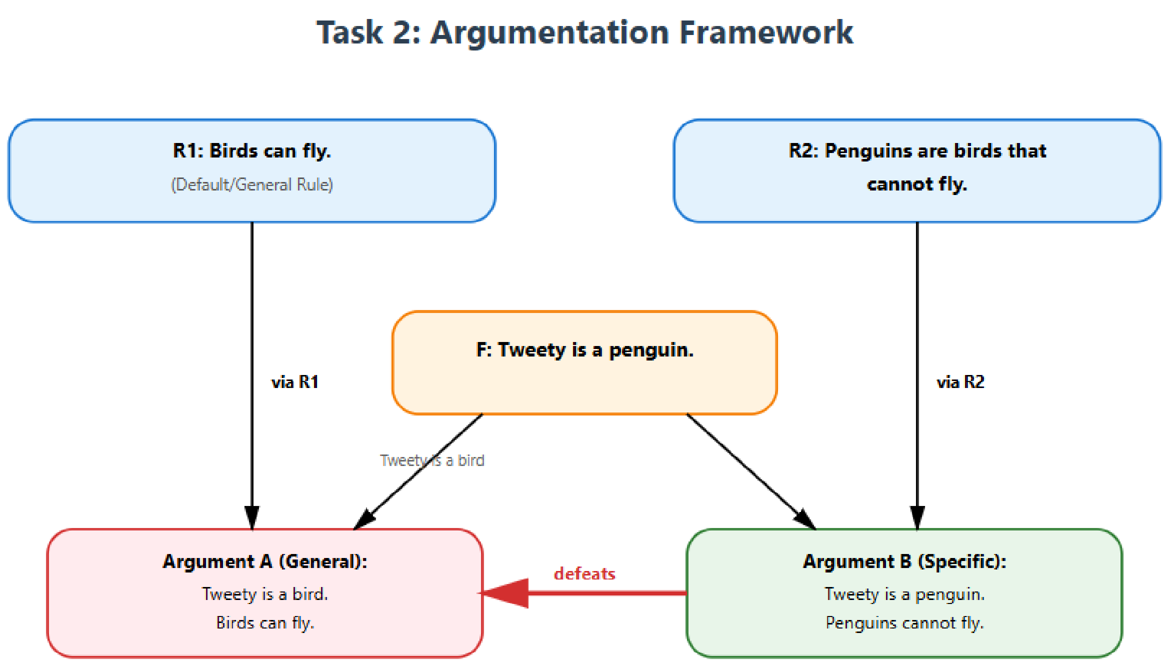
Example Scenario:
Rule 1: Birds can fly.
Rule 2: Penguins are birds that cannot fly.
Fact: Tweety is a penguin.

Expected Answer:
Argument A: Birds can fly.
Argument B: Penguins are birds that cannot fly.
→ **Argument B defeats A**, because it is more specific.

Rubric:

- 10 pts – complete diagram or description
- 5 pts – includes clear reasoning steps
- 5 pts – shows correct defeat or resolution

Students can draw this using **draw.io**, **Canva**, or on paper.



Part III. Reflection and Discussion (20 points)

Instruction: Write a short essay (150–200 words) answering the prompt below.

There was a time when I honestly believed that hard work was the only thing needed to be successful. I thought that if I just gave enough time, effort, and focus, I would always get the results I wanted. But as I grew older and faced more real situations, I learned that success doesn't depend on effort alone. It also needs planning, teamwork, flexibility, and sometimes even luck or timing. I've seen people who work really hard but still struggle because they lack direction or support. On the other hand, some achieve more because they plan smart and use their time wisely. That experience taught me to be more open-minded and to change what I believe when I learn something new.

This way of thinking is a lot like non-monotonic reasoning in artificial intelligence. In normal reasoning, once something is considered true, it stays true no matter what new facts come in. But in non-monotonic reasoning, beliefs can change when new information shows up. For example, an AI might first believe that all birds can fly, but once it learns that penguins and ostriches can't, it changes its conclusion. This makes the AI behave more realistically just like how people adjust their understanding when they get new evidence.

My view about hard work and success changed in the same way. I realized that holding on to one idea without questioning it can limit growth. Both humans and AI need the ability to update what they believe so they can make better choices. For me, non-monotonic reasoning is not just a computer concept—it's a reminder that learning never stops, and that being willing to change your mind is part of true intelligence.

Rubric:

Criteria	Excellent (5)	Good (4)	Fair (3)	Poor (2–1)
Relevance to topic	Strong connection to AI reasoning	Mostly relevant	Limited relation	Off-topic
Insight & reflection	Deep and thoughtful	Some insight	Simple restatement	Superficial
Clarity & grammar	Clear and polished	Minor errors	Understandable	Hard to read

Total Points: 80

Component	Points
Part I – Conceptual	20
Part II – Laboratory	40
Part III – Reflection	20
Total	**80 pts**