**Batch: A1 Roll No.: 1711008**

**Experiment/ assignment/ tutorial No. 8**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **Title:** Solution of a logic based agent for Wumpus World |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective:** Solution of a logic-based agent for an exploration problem

**Expected Outcome of Experiment:**

|  |  |
| --- | --- |
| **Course Outcome** | **After successful completion of the course students should be able to** |
| **CO3** | Represent and formulate the knowledge to solve the problems using various reasoning techniques |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. **“Artificial Intelligence: a Modern Approach” by Russel and Norvig, Pearson education Publications**
2. **“Artificial Intelligence” By Rich and knight, Tata Mcgraw Hill Publications**
3. **http://www.kr.tuwien.ac.at/students/prak\_wumpusjava/simulator/Rules.html**
4. **http://www.cis.temple.edu/~giorgio/cis587/readings/wumpus.shtml**
5. **www.cs.laurentian.ca/dgoforth/cosc4117/ppt/63**wumpus**FOL.ppt**
6. [**http://www.cs.uku.fi/~mnykanen/TEK/teklectures6.pdf**](http://www.cs.uku.fi/~mnykanen/TEK/teklectures6.pdf)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pre Lab/ Prior Concepts:** Problem formulation, Basic Logic concepts (AND, OR etc.)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Historical Profile:**

Sometimes the problem is so unknown that the state space tree cannot be generated for the same. Such problems interleave actions of search and execution and are called Exploration Problems. Wumpus World is an example exploration problem.

The agent should be able to work also in the unknown environment. This program gives an idea about how it should explore and memorize the unknown world.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**New Concepts to be learned:** Solving the problems and partial knowledge of the task environment. Knowledge representation method.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Example Wumpus World configuration:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **8** |  |  |  |  |  |  |  |  |
| **7** |  |  |  |  | **Breeze** |  |  |  |
| **6** |  | **Stench** |  | **Breeze** | **Pit** | **Breeze** |  | **Breeze** |
| **5** | **Stench** | **Wump** | **G/S/B** | **Breeze** | **Breeze** | **Pit** | **Breeze** | **Pit** |
| **4** |  | **Stench** |  | **Breeze** | **Pit** | **Breeze** |  | **Breeze** |
| **3** |  |  |  |  | **Breeze** |  |  |  |
| **2** |  |  |  |  |  |  |  |  |
| **1** | **Agent** |  |  |  |  |  |  |  |
| **X** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |

**PEAS:**

**Performance Measure:**

* Gold +100
* Death -100
* -1 per action
* -10 for using the arrow.
* The game ends either when the agent dies or when the agent climbs out of the cave.

**Environment:**

* 4 X 4 grid of rooms
* Agent always starts in square [1,1], facing to the right
* Squares adjacent to wumpus are smelly
* Squares adjacent to pit are breezy
* Glitter iff gold is in the same square
* Shooting kills wumpus if you are facing it
* Shooting uses up the only arrow
* Grabbing picks up gold if in the same square
* Releasing drops the gold in the same square

**Actuators**:

* Left Turn, Right Turn, Forward
* Grab
* Shoot

**Sensors**:

* Squares adjacent to wumpus (not diagonally), the agent will perceive a **Stench**.
* Squares directly adjacent to a pit (not diagonally), the agent will perceive a **Breeze**.
* In the square where the gold is (not diagonally), the agent will perceive a **Glitter**.
* When an agent walks into a wall, it will perceive a **Bump**.

**Properties of task environment:**

* Observable: No, only local perception
* Deterministic: Yes outcome is specified
* Episodic: No, Sequential at the level of actions
* Static: Yes, wumpus and pits don’t move around
* Discrete: Yes
* Single-agent: Yes (Wumpus is considered as a feature of nature)

**Base knowledge for the agent:**

* We take a grid of 64 squares labelled from (1,1) to (8,8).
* Wumpus (W) lives in one of the grids (2,5) & eats all those who venture there
* Pit (P) exists in one or more grids and the agent dies once it enters that grid.
* All adjacent grids to P have breeze (B)
* One of the grids have gold (G) in it

**Knowledge Representation:** *(Both in Propositional and Predicate Logic)*

Propositional Logic:

Bxy = Breeze in grid(x,y)

Wxy = Wumpus in grid(x,y)

Pxy = Pit in grid(x,y)

Sxy = Stench in grid(x,y)

Gxy = Gold in grid(x,y)

First Order Logic:

Smelly(x,y,t) = (x,y) has stench at time instance t

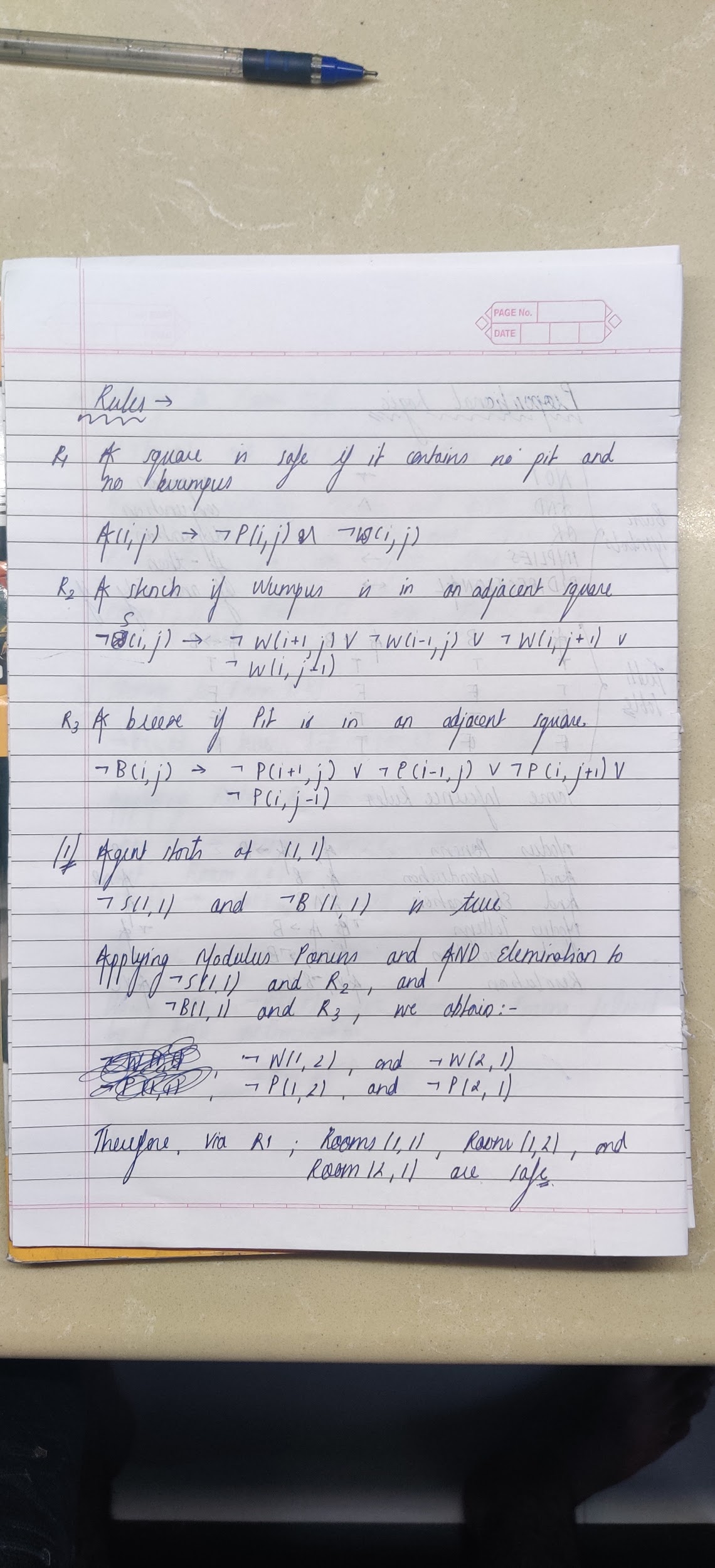
Breezy(x,y,t) = (x,y) has breeze at time instance t

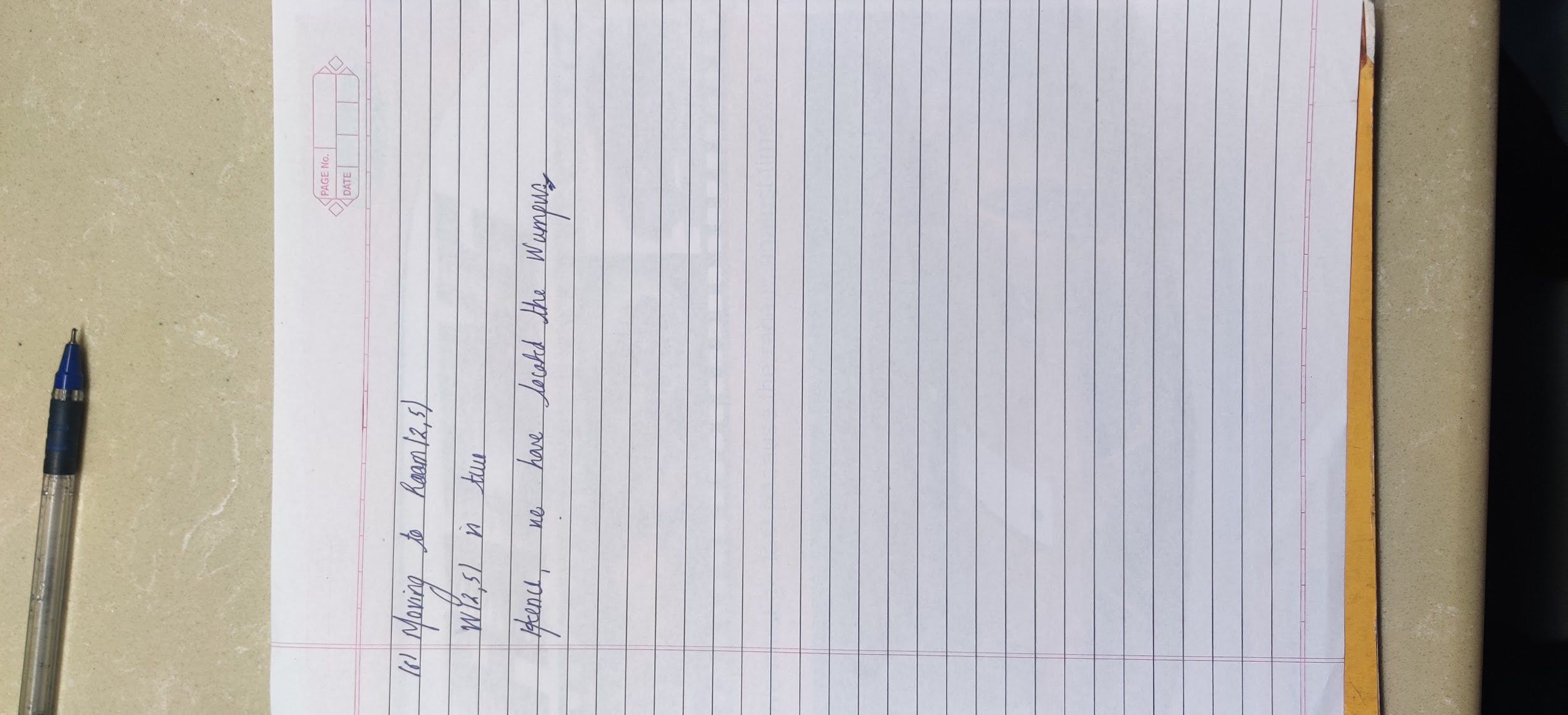
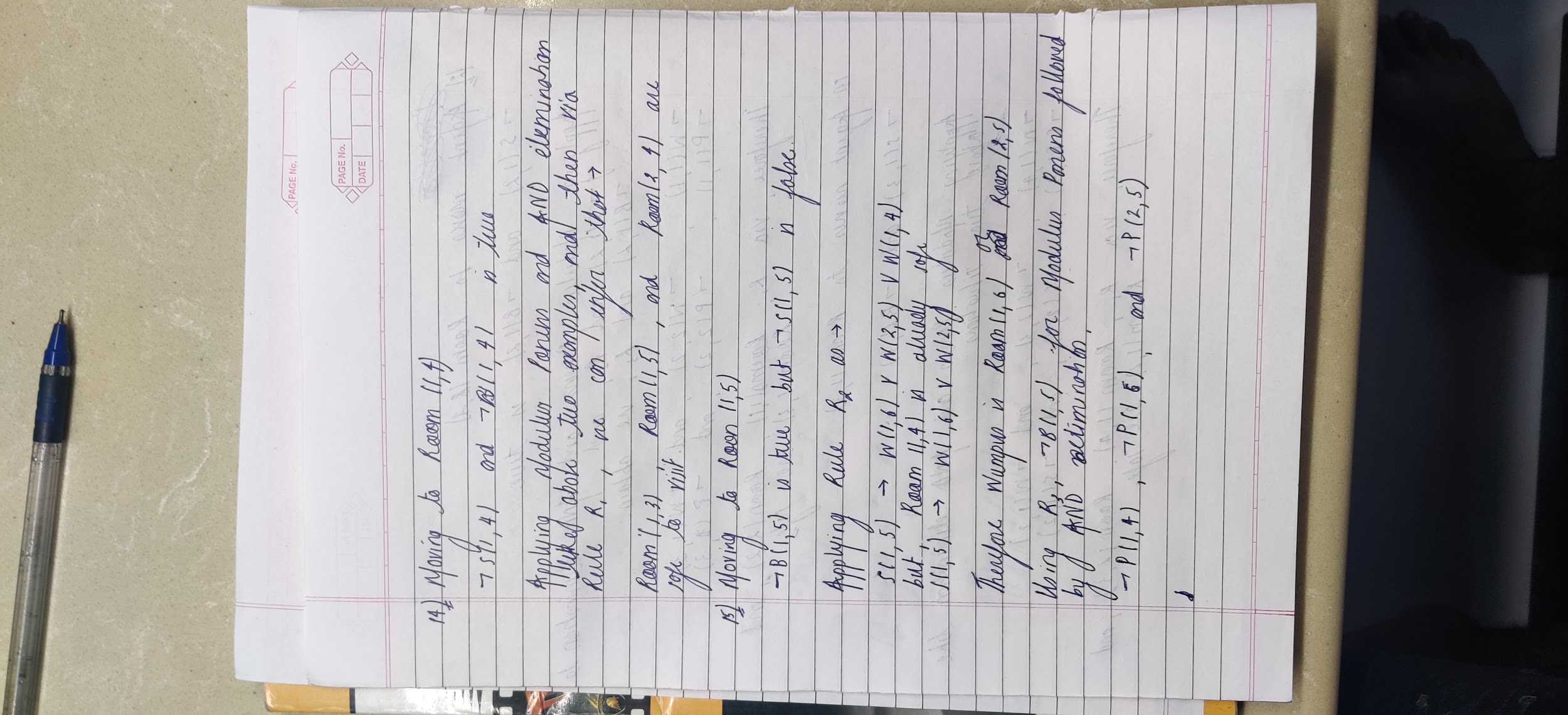
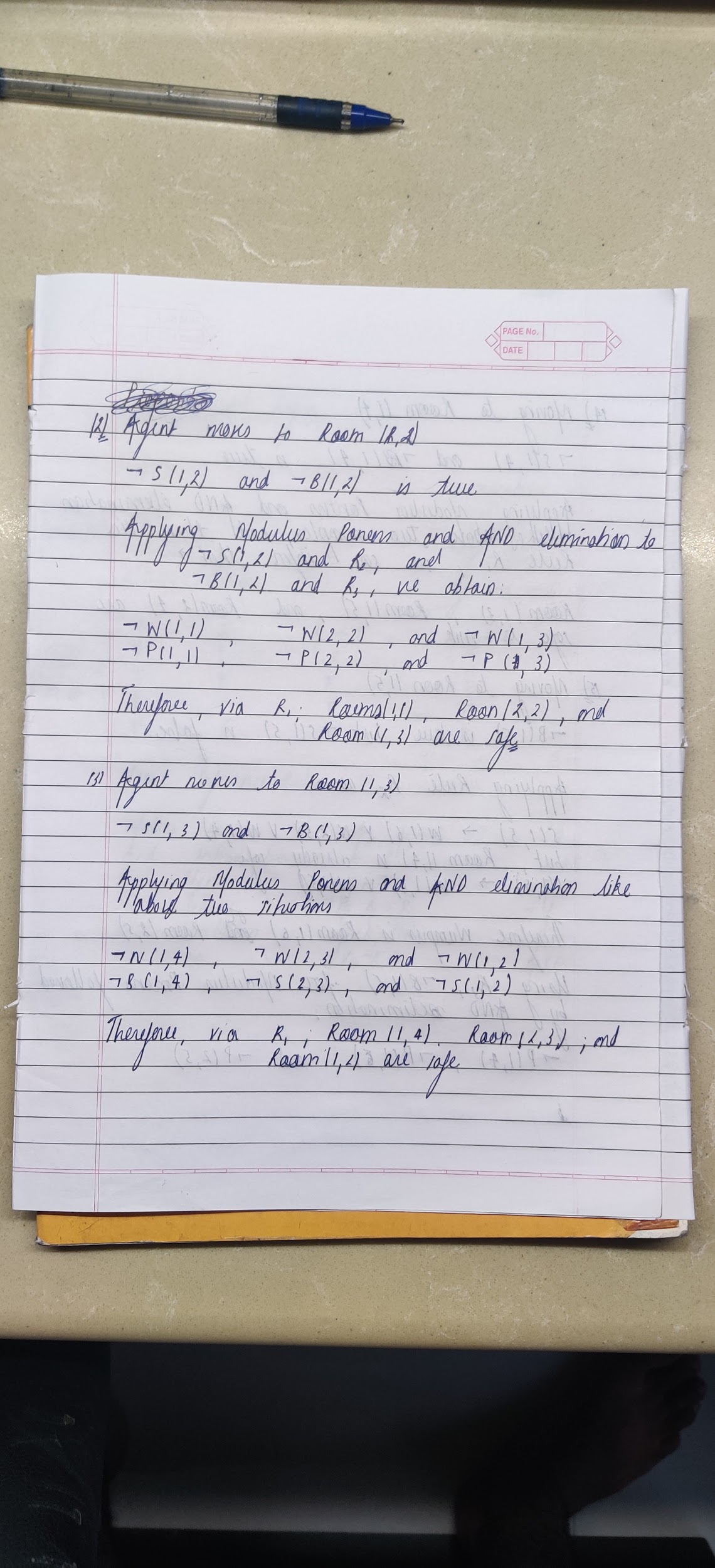
has Pit(x,y) = (x,y) has pit

haswumpus(x,y) = (x,y) has wumpus

isSafe(x,y) = (x,y) is safe

**Solution for the problem world chosen:**

****

****

**Post Lab Objective Questions**

1. **Uncertainty arises in the wumpus world because the agent’s sensors give only**
   1. Full & Global information
   2. Partial & Global Information
   3. Partial & local Information
   4. Full & local information
   5. Global information only.

**Answer: C**

1. **A knowledgebase contains**:
   1. rules, facts, and relationships.
   2. only rules and relationships.
   3. simulation of human thinking.
   4. only facts.

**Answer: A**

3. **How do you represent “All dogs have tails”.**

1. ۷x: dog(x)🡪hastail(x)
2. ۷x: dog(x)🡪hastail(y)
3. ۷x: dog(y)🡪hastail(x)
4. ۷x: dog(x)🡪has🡪tail(x)
5. ۷x: dog(x)🡪has🡪tail(y)

**Answer: A**

**4. What are you predicting by the logic: ۷x: €y: loyalty(x, y).**

1. Everyone is loyal to someone
2. Everyone is loyal to all
3. Everyone is not loyal to someone
4. Everyone is loyal
5. Everyone is not loyal.

**Answer: A**

**Post Lab Subjective Questions**

1. **Differentiate between Propositional Logic and First Order Logic.**

Propositional logic consists of a set of atomic propositional symbols (e.g. Socrates, Father, etc), which are often referred to by letters p, q, r etc. (Note that these letters aren't variables as such, as propositional logic has no means of binding variables). These symbols are joined together by logical operators (or connectives) to form sentences. The basic logical operators are:

• Negation: ¬p ("it is not the case that ");

• Conjunction: p ∧ q ("p and q");

• Disjunction: p ∨ q ("p or q");

• Implication: p ⇒ q ("p implies q", or "q if p");

• Equivalence: p ⇔ q ("p if and only if q").

First-order Predicate Logic

First-order Predicate Logic is an extension of propositional logic, which allows quantification over variables. Whereas in propositional logic you can only talk about specifics (e.g. "Socrates is a man"), in predicate logic you can also talk more generally (e.g. "all men are mortal").

First-order logic

First-order predicate logic allows variables to range over atomic symbols in the domain. It doesn't allow variables to be bound to predicate symbols, however. A second-order logic (such as second-order predicate logic) does allow this, and you can write sentences such as:

∀p.p(Socrates).

1. **What are the issues for knowledge representation?**

**Important Attributes**

Are there any attributes that occur in many different types of problem?

There are two *instances* and each is important because each supports property inheritance.

**Relationships**

What about the relationship between the attributes of an object, such as, inverses, existence, techniques for reasoning about values and single-valued attributes. We can consider an example of an inverse in:

*band(John Zorn, Naked City)*

This can be treated as John Zorn plays in the band *Naked City* or John Zorn's band is *Naked City*.

Another representation is *band* = *Naked City*

*band-members* = *John Zorn, Bill Frissell, Fred Frith, Joey Barron,*

**Granularity**

At what level should the knowledge be represented and what are the primitives. Choosing the Granularity of Representation Primitives are fundamental concepts such as holding, seeing, playing and as English is a very rich language with over half a million words it is clear we will find difficulty in deciding upon which words to choose as our primitives in a series of situations.

1. **What is inferential knowledge and procedural knowledge?**

## Inferential Knowledge

Represent knowledge as *formal logic*:

*All dogs have tails* : *dog(x)* *hasatail(x)* Advantages:

* A set of strict rules.
  + Can be used to derive more facts.
  + Truths of new statements can be verified.
  + Guaranteed correctness.
* Many inference procedures available to in implement standard rules of logic.
* Popular in AI systems. *e.g* Automated theorem proving.

### Procedural Knowledge

Knowledge encoded in some procedures

* 1. small programs that know how to do specific things, how to proceed.
  2. *e.g* a parser in a natural language understander has the knowledge that a *noun phrase* may contain articles, adjectives and nouns. It is represented by calls to routines that know how to process articles, adjectives and nouns.

**Advantages:**

* *Heuristic* or domain-specific knowledge can be represented.
* *Extended logical inferences*, such as default reasoning facilitated.
* *Side effects* of actions may be modelled. Some rules may become false in time. Keeping track of this in large systems may be tricky.

**Disadvantages:**

* Completeness -- not all cases may be represented.
* Consistency -- not all deductions may be correct.

*e.g* If we know that *Fred is a bird* we might deduce that *Fred can fly*. Later we might discover that *Fred is an emu*.

* Modularity is sacrificed. Changes in a knowledge base might have far-reaching effects.

Cumbersome control information.

1. **Translate these sentences into formulae in predicate logic** 
   1. John likes all kinds of food

**∀x: food(x) -> likes(john,x)**

* 1. Apples are food

**Food(apple)**

* 1. Chicken is food

**Food(chicken)**

* 1. Anything anyone eats and isn’t killed by its food

**∀x,y: eat(x,y) ^ ~killedby(x,y) ->food(x)**

* 1. Bill eats peanuts and is still alive

**Eats (Bill, Peanuts)**

**Alive (Bill)**

* 1. Sue eats everything Bill eats

**∀x: eats(Bill, x) -> eats(Sue,x)**