Mini Project: Rule-Based Expert System for Sugarcane Disease Diagnosis

1. Introduction

This mini project demonstrates the application of **Knowledge Representation and Reasoning** using a **rule-based expert system**. The system is designed to diagnose common sugarcane diseases based on observed symptoms and recommend appropriate treatments. Knowledge is represented using **predicate logic**, **facts**, and **production rules**, together with **forward chaining**, **backward chaining**, and **conflict resolution**.

---

2. Problem Description

Sugarcane farmers often experience difficulty in identifying diseases early due to overlapping symptoms. This expert system reasons over observed symptoms represented as predicates and infers the most likely sugarcane disease together with a recommended treatment.

3. Knowledge Representation

The knowledge base consists of **facts (predicates)** and **production rules**.

3.1 Disease Facts

The following predicates represent known sugarcane diseases:

```
disease(red_rot).
disease(smut).
disease(grassy_shoot).
disease(mosaic).
disease(orange_rust).
disease(yellow_leaf_disease).
```

3.2 Symptom Facts

Observed symptoms are represented as unary predicates:

```
symptom(reddened_stem).
symptom(white_patches).
symptom(hollow_stem).
symptom(drying_stool).
symptom(black_whip).
symptom(excessive_tillering).
symptom(stunted_growth).
symptom(narrow_leaves).
symptom(yellow_leaves）.
```

symptom(mosaic_pattern).
symptom(orange_pustules).
symptom(premature_drying).
symptom(leaf_yellowing).
symptom(reduced_tillering).
symptom(weak_canes).

4. Production Rules

Rules are expressed in **IF–THEN** form using predicates.

R1: IF symptom(reddened_stem) AND symptom(white_patches)
THEN disease(red_rot)

R2: IF symptom(hollow_stem) AND symptom(drying_stool)
THEN disease(red_rot)

R3: IF symptom(black_whip) AND symptom(excessive_tillering)
THEN disease(smut)

R4: IF symptom(stunted_growth) AND symptom(narrow_leaves)
THEN disease(grassy_shoot)

R5: IF symptom(yellow_leaves) AND symptom(mosaic_pattern)
THEN disease(mosaic)

R6: IF symptom(orange_pustules) AND symptom(premature_drying)
THEN disease(orange_rust)

R7: IF symptom(leaf_yellowing) AND symptom(reduced_tillering)
THEN disease(yellow_leaf_disease)

Treatment Rules

R8: IF disease(red_rot)
THEN treatment(use_fungicide)

R9: IF disease(smut)
THEN treatment(remove_infected_plants)

R10: IF disease(mosaic)
THEN treatment(use_virus_free_seed)

R11: IF disease(grassy_shoot)
THEN treatment(rogue_and_destroy)

R12: IF disease(orange_rust)
THEN treatment(spray_triazole)

5. Inference Mechanisms

5.1 Forward Chaining

Forward chaining is a data-driven inference technique where reasoning begins with known facts and applies rules to derive new facts.

Initial Facts:

```
symptom(reddened_stem)
symptom(white_patches)
symptom(hollow_stem)
symptom(drying_stool)
```

Inference Steps:

- R1 fires → disease(red_rot)
- R2 fires → disease(red_rot)
- R8 fires → treatment(use_fungicide)

Final Conclusion:
The system diagnoses **red_rot** and recommends **use_fungicide**.

5.2 Backward Chaining

Backward chaining is a goal-driven reasoning process that starts from a hypothesis and works backward to confirm it.

Goal: treatment(use_fungicide)

Reasoning:

- treatment(use_fungicide) is concluded if disease(red_rot) is true (R8)
- disease(red_rot) is concluded if symptom(reddened_stem) AND symptom(white_patches) are true (R1)
- Both symptoms exist as facts in the knowledge base

Goal Proven: The treatment recommendation is logically valid.

6. Conflict Resolution Strategy

When multiple rules are applicable, the system resolves conflicts using:

- **Specificity:** rules with more conditions have higher priority

- **Rule support:** diseases supported by multiple rules are preferred
- **Tie handling:** multiple diseases may be reported if ambiguity remains

7. Test Cases

Test Case 1

Input Facts: symptom(reddened_stem), symptom(white_patches)

Expected Output: disease(red_rot)

Actual Output: disease(red_rot)

Status: Pass

Test Case 2

Input Facts: symptom(black_whip), symptom(excessive_tillering)

Expected Output: disease(smut)

Actual Output: disease(smut)

Status: Pass

Test Case 3

Input Facts: symptom(yellow_leaves), symptom(mosaic_pattern)

Expected Output: disease(mosaic)

Actual Output: disease(mosaic)

Status: Pass

Test Case 4

Input Facts: symptom(orange_pustules), symptom(premature_drying)

Expected Output: disease(orange_rust)

Actual Output: disease(orange_rust)

Status: Pass

8. Documentation

This expert system was developed using predicate logic to represent knowledge in a structured and interpretable form. Facts are encoded as predicates, and reasoning is performed using production rules. Forward and backward chaining provide transparent explanations for conclusions. The system is modular and can be extended with additional diseases, symptoms, and treatments.

9. Conclusion

The project successfully demonstrates the use of predicate logic and rule-based reasoning for sugarcane disease diagnosis. By combining facts, rules, inference mechanisms, and conflict resolution, the system provides accurate and explainable recommendations, fulfilling the objectives of the mini project.

Members

Bakari Juma Bakari	C026-01-0805/2024
Bridgett Wairimu	C026-01-0735/2024
Alisha Cherono	C026-01-2642/2024
Micah Kiprotich	C026-01-2795/2021
Mary Teresa Nyambura	C026-01-2743/2024