AI-Based Crime Solver Using Prolog

Md. Mahir Uddin
United International University
ID: 0122430022

1. Introduction

In this project, an AI based murder crime investigation model was built to find the killer using Prolog. The scenario revolves around a murder that occurred in an IT company. The Project Manager (Mushfiq) was found dead in his office shortly after office hours. Several employees were present in the office during the time of the crime, and preliminary investigation reveals conflicting testimonies, hidden motives, and tampered CCTV footage. However, there is no direct evidence pointing to a specific individual.

The project aims to logically infer the killer by encoding all relevant facts such as exit times, testimonies, contradictory statements, motives, finger print at crime zones. The model filters out innocent employees step by step and finally infers whether the crime was committed by an individual or pair or group of people based on logical consistency and contradictions.

2. Case Information

The murder took place in an IT company, where the Project Manager (PM) was found dead in his cabin between 5:10 PM and 5:40 PM, shortly after office hours. Only few employees were present in the office at that time. So, Initially, only people who left office after 5:10 PM are kept as initial suspects and investigated. The CCTV footage was erased exactly at 5:40 PM, making it impossible to visually track any activity during the crime window. The crime scene was later examined by the police, and various pieces of information and evidences were collected.

To simulate this case in Prolog, a knowledge base was built that contains all known facts, observations, and testimonies. The model will infer the killer from these facts by deductive reasoning. The knowledge base includes:

- **Suspect data:** Suspect names, their position, departments.
- **Victim details:** Identity of the victim, their position, department and location of death.
- **Exit logs:** Exit time for each employee. The time was collected from biometric attendance systems to know who stayed till the crime.
- **Locations and testimonies:** Each employee was asked where they were between 5:10 PM and 5:40 PM, and with whom. These testimonies are used to detect contradictions and inconsistencies.
- **Motive evidence:** Digital and physical documents (e.g., accounting mismanagement papers, mobile messages) linking certain employees to clear motives like financial fraud, blackmail, or affair disputes.
- **Fingerprint evidence:** Collected from the crime scene and server room from where CCTV footage was deleted.
- **Cover-up links:** Some employees have vouched for others, which could mean either genuine support or a coordinated attempt to hide joint involvement in the crime.

The Prolog system is fed all this structured information as facts. From there, a set of rules and logical inference mechanisms are applied to uncover contradictions, evaluate motives, detect false

alibis, and ultimately deduce who committed the murder — whether it was an individual or a coordinated pair.

This structured approach enables the model to mimic a real-world detective process using only pure logic.

3. Rules and Logic

The Prolog model used a set of structured rules to infer guilt based on the information gathered from the case. These rules perform filtering, contradiction detection, evidence evaluation, and finally, deduction of the killer(s). The rules are organized into three logical groups:

3.1 Suspect Evaluation Rules

These rules identify individuals who are likely to have committed the crime based on presence, motive, testimonies, and evidence.

- **has_motive(X)**
  An employee is considered to have a motive if physical or digital evidence or testimonies link them to reasons such as fraud, blackmail, financial mismanagement or enemity.
- **stayed_till_crime(X)**
  Returns the name of employees who left office after CCTV footage erasing from server room.
- **strong_suspect(X)**
  Identifies suspects who both had a motive and stayed till crime.
- **contradiction(X, Y)**
  Detects contradictions between testimonies — if two people claim to be in the same place but disagree on each other's presence. Or suspects who have claimed to be together but their said place doesn't match.
- **contradictors(X)**
  Gives list of suspects whose testimony is contradictory with others.
- **evidence(X)**
  Gives list of suspects whose workplace is not in the crime zone but their fingerprint was found in those places.
- **possible_killer(X)**
  Strong suspects with either contradictory statements or with fingerprint in crime zones.
- **killer(X)**
  If only one possible killer remains after applying all filters (motive, evidence, contradiction, etc.), this rule declares that person as the killer.

3.2 Collaborative Crime Detection

These rules deal with the situation where multiple people might be involved together in the murder. It's expected that if it is a collaborative crime, the killers will cover up each other.

- **possible_killer_pair(X, Y)**
  If two people are both possible killers *and* they're covering for each other, then they might be involved together in planning or executing the murder.
- **killer_pair(X, Y)**
  If there's only one possible killer pair remaining who meet all the criteria, we identify them as the killer pair. Otherwise, we will have list of possible candidates but can not certainly declare someone as killer pair.
- **killer_group_chain(X, Z)**
  This rule uses recursion to track longer chains of cover-ups or contradictions — helpful if there's a whole group involved rather than just two people.

3.4 Information Extraction Rules

These rules aren't for solving the case directly, but to help us understand more details about the suspects or killers after they've been identified. The rules are grouped in 3 parts. Possible killer information, confirmed killer information and killer pair information.

Possible Killer Information

These tell us more about the suspects the system thinks *might* be the killer.

- **possible_killers_motive(Killer, Motive)**
  Shows what motive each possible killer had.
- **possible_killers_evidence(Killer, Fingerprint)**
  Tells us in which suspicious locations their fingerprints or evidence were found.
- **possible_killer_contradiction(Killer, Contradiction_with)**
- Shows who the possible killer gave a conflicting statement about.
- **possible_killer_coverup(Killer, Coverup)**
  Shows who they tried to protect — maybe they were involved together.

Confirmed Killer Information

If the possible killer domain gets reduced to one then the model will declare that one suspect as the killer.

- **killer_motive(Killer, Motive)**
  Shows the exact motive of the confirmed killer.
- **killer_evidence(Killer, Fingerprint)**
  Lists the fingerprint evidence that links the killer to the crime zones.
- **killer_contradiction(Killer, Contradiction_with)**
  Tells us whose statement the killer's statement conflicted with.
- **killer_coverup(Killer, Coverup)**
  Shows if the killer tried to protect or cover for someone else.

Killer Pair Info

These give extra details if the model identifies a killer pair.

- **killer_pair_motives(X, Motive)**
  Lists the motives of each person in the killer pair.
- **killer_pair_evidence(X, Fingerprint)**
  Tells us what physical evidence (fingerprints) links each of them to the scene.
- **killer_pair_coverup(X, Coverup)**
  Shows how the killer pair supported each other — possibly trying to fool investigators.

All these rules can be used to query about killer or possible killer's various information.

4. Future Work

Right now, all the information — like statements and evidence — has to be added manually in Prolog. In future, we could build a simple front-end or use natural language processing so that the model can take inputs directly from users or witness statements and turn them into Prolog facts automatically.

5. Results and Discussion

Based on these logical deductions, the model successfully identified Mehedi and Nanziba as the killer pair. We queried the Prolog model to extract various pieces of information about the killers, their motives, contradictions, and evidence.

The model inferred this from multiple clues:

- Both Mehedi and Nanziba had a strong motive — the project manager had discovered their involvement in a financial scandal.
- They both stayed in the office after 5:40 PM, which made them part of the prime suspect pool since the CCTV footage was erased around that time.
- Their statements contradicted each other. Mehedi claimed he was alone in the coffee room, while Nanziba said she was in the accounts department with Mehedi — indicating deception.
- They both covered for each other.
- Fingerprint evidence placed Mehedi in both the manager's room and the server room — the latter being a location where he had no professional reason to be. Nanziba's fingerprints were also found in the manager's room.

After querying the system, all this evidence was logically linked, and the model confidently concluded that Mehedi and Nanziba were the killer pair.

6. Conclusion

This project shows how a logic-based system like Prolog can be used to solve a complex murder case using pure reasoning. We added all the known facts, motives, timings, fingerprints, and testimonies, and the model handled everything step by step to figure out who the killers were.