AI Researcher Project: Constraint-Solving Mystery Game

Generating Logic Puzzles

Fall 2025

Contents

List of Figures

List of Tables

1 Introduction and Motivation

This project introduces an **AI-driven logic mystery game** that constitutes the first stage in developing an interactive environment where **humans and AI jointly engage in rational problem solving**. The central premise is to design a setting where reasoning, rather than chance, determines success. In each instance of the game, exactly one suspect is the murderer, and the system procedurally generates a consistent microcosm of facts, where each is logically related, internally coherent, and verifiable through deduction. No clue ever lies: information is complete but distributed, and the player (human or AI) must infer the consistent equilibrium of facts that uniquely identifies the murderer. The game thereby functions as a stylized market of logical propositions, in which agents update their beliefs and eliminate inconsistent hypotheses until convergence on the truth.

From an analytical standpoint, the project operationalizes key concepts in AI that parallel logical reasoning under constraints (most notably, *Constraint Satisfaction Problems (CSPs)*, *search algorithms*, and *knowledge representation*). Each mystery instance can be viewed as a small, well-defined logical puzzle in which characters, locations, weapons, times, and motives play the role of agents and resources, while the logical clues define the feasible set of allocations. The solution corresponds to the equilibrium state in which all constraints are simultaneously satisfied, and no further reallocation (logical reassignment) is possible without violating consistency. This approach formalizes reasoning as a process of constrained optimization, where the "utility" of a solution corresponds to its internal coherence and explanatory power.

Methodologically, this project extends the standard CSP framework in three significant directions. First, each character is endowed with a complete multi-attribute specification, analogous to a logical agent's endowment vector, across attributes such as location, weapon, time, and motive. Second, the framework relaxes the classical "all-different" constraint: multiple agents may share the same attribute values, as might occur in real-world settings with overlapping preferences or endowments, while the equilibrium (i.e., the identity of the murderer) remains unique. Third, a *hardness controller* is introduced to govern the cognitive complexity of the problem by stochastically selecting between *direct* clues (transparent information) and *complex* clues (indirect or negated information). Examples include:

```
The person who was in the Kitchen had a Rope.
The person who was in the Kitchen did not have a Rope.
```

The hardness controller functions analogously to informational frictions in real-world settings (introducing uncertainty, partial observability, and the need for inference under incomplete information). It allows the designer to tune the game's difficulty.

The intellectual motivation behind this project is twofold. First, it seeks to formalize reasoning and inference as computational processes that can be shared between humans and AI. Second, it establishes a scalable framework for generating and solving logically consistent worlds, forming the foundation for future AI-human collaboration games. In this sense, the project can be viewed as an experimental platform for studying joint cognition, belief updating, and strategic reasoning. By embedding rational inference within a game environment, we aim to illuminate how artificial and human agents can cooperate to achieve logically efficient equilibria in the space of information and belief.

2 Objectives

The overarching objective of this project is to design and implement an **AI-driven logic mystery game** that provides a rigorous yet engaging framework for human-AI collaborative reasoning. The project seeks to advance both the theoretical understanding of AI reasoning systems and their practical application in interactive, information-rich environments. Specifically, the objectives are threefold:

(1) To formalize the mystery game as a Constraint Satisfaction Problem (CSP). The first goal is to construct a precise mathematical formulation of the logic mystery as a CSP, where each game instance represents a finite set of variables (characters, locations, weapons, times, motives) and a corresponding set of logical constraints. The system must guarantee the existence and uniqueness of the solution with guarantees that, despite potential overlaps in attributes, only one consistent configuration satisfies all clues. This step provides the formal backbone of the game and guarantees that each generated instance is solvable, consistent, and logically self-contained.

(2) To develop an adaptive clue-generation mechanism governed by a hardness controller. The second goal is to create a stochastic mechanism that modulates the cognitive and computational difficulty of the game. The hardness controller will probabilistically determine whether each clue is direct (explicitly revealing a fact) or complex (involving negation, relational reasoning, or indirect implication). This feature introduces a continuum of difficulty levels and allows for the systematic study of how both human and AI agents adjust their inference strategies under differing information structures.

(3) To establish a framework for human-AI collaborative reasoning. The ultimate objective is to develop a cooperative reasoning environment where human intuition and AI inference complement one another. The game will serve as a controlled laboratory for examining how human and machine agents can jointly navigate an information space, resolve inconsistencies, and converge on a shared, logically valid conclusion. This human-AI teaming framework lays the groundwork for broader research into collective intelligence, distributed reasoning, and adaptive trust calibration in human-autonomy systems.

3 World Model

Let $\mathcal{S}$ be the set of suspects (e.g., $\{$Ava, Blake, Casey, Drew$\}$). Let $\mathcal{A}$ be a fixed set of attributes of size $K = 5$:

$$\mathcal{A} = \{\text{Room, Weapon, Time, Entry, Motive}\}.$$

For each attribute $A \in \mathcal{A}$, let $\mathcal{V}_A$ denote its finite domain (e.g., Room = {Kitchen, Library, Study, Conservatory}, etc.).

Full Specification. Each suspect $s \in \mathcal{S}$ is assigned a concrete value *for every attribute*:

$$X_{s,A} \in \mathcal{V}_A \quad \text{for all } s \in \mathcal{S},\ A \in \mathcal{A}.$$

It is permissible that $X_{s,A} = X_{s',A}$ for $s \neq s'$, i.e., two suspects can have the same value for an attribute A.

Murderer and Crime Tuple. One suspect $M \in \mathcal{S}$ is designated the murderer. The *crime tuple* is

$$(\text{CRoom}, \text{CWeapon}, \text{CTime}, \text{CEntry}, \text{CMotive}) = (X_{M,\text{Room}}, X_{M,\text{Weapon}}, X_{M,\text{Time}}, X_{M,\text{Entry}}, X_{M,\text{Motive}}).$$

4 Clue Language

All clues are *truthful*, i.e., satisfied by the hidden world (X, M).

4.1 Direct Clues

- **Unary positive:** $X_{s,A} = v$.

- **Unary negative:** $X_{s,A} \neq v$.
- **Crime attribute positive:** C$A = v$ (e.g., "The murder weapon was Rope").
- **Crime attribute negative:** C$A \neq v$.
- **Time-ordering (if $A =$ Time):** CTime $\leq X_{s,\text{Time}}$ or $X_{s,\text{Time}} \geq$ CTime.

4.2 Complex Clues

Complex clues are either **negated** statements or **indirect relational** statements linking two attributes. Typical patterns:

- **Indirect positive (attribute-to-attribute):**

$$\exists s \in \mathcal{S} : \ X_{s,A} = v \ \wedge \ X_{s,B} = w,$$

  e.g., "*The person(s) who was in the Kitchen had a Rope.*"
- **Indirect negative (attribute-to-attribute):**

$$\forall s \in \mathcal{S} : \ X_{s,A} = v \Rightarrow X_{s,B} \neq w,$$

  e.g., "*The person(s) who was in the Kitchen did not have a Rope.*"
- **Negated unary:** $X_{s,A} \neq v$ (already allowed as direct, but under "complex" mode these will be more prevalent).

These increase reasoning difficulty by requiring the solver to reason over *linkages* instead of single-attribute facts.

5 Hardness Controller

We expose two knobs:

Direct vs. Complex selector ($\alpha \in [0, 1]$). Draw $Z \sim$ Bernoulli(α). If $Z = 1$ generate a *direct* clue; if $Z = 0$ generate a *complex* clue. Lower $\alpha \Rightarrow$ harder on average.

Complex subtype selector ($\beta \in [0, 1]$). Conditioned on complex mode ($Z = 0$), draw $Y \sim$ Bernoulli(β). If $Y = 1$ generate an *indirect positive* clue; if $Y = 0$ generate an *indirect negative* clue. Lower β yields more negative (often harder) relational clues.

Optionally use a schedule or mapping $\alpha = \alpha$(target_difficulty) and $\beta = \beta$(target_difficulty) so instructors can request an overall difficulty level.

6 Solver and Uniqueness

Because values can be shared across suspects, the CSP is less constrained; nonetheless, we require *uniqueness of the murderer*:

$$|\{ M \in \mathcal{S} \ : \ \exists X \text{ s.t. } (X, M) \text{ satisfies all clues} \}| = 1.$$

6.1 Feasibility Check for a Candidate Murderer m

Formulate binary decision variables

$$\delta_{s,A,v} = \begin{cases} 1 & \text{if } X_{s,A} = v, \\ 0 & \text{otherwise,} \end{cases} \quad \forall s \in \mathcal{S}, \ A \in \mathcal{A}, \ v \in \mathcal{V}_A.$$

Crime-linking to murderer M:

$$\gamma_{A,v} = \delta_{M,A,v} \quad \forall A, v, \qquad \text{where } \gamma_{A,v} = 1 \iff \text{C}A = v.$$

Encoding clues. Each clue translates to linear/logical constraints, e.g.,

- $X_{s,A} = v \Rightarrow \delta_{s,A,v} = 1$.
- $X_{s,A} \neq v \Rightarrow \delta_{s,A,v} = 0$.
- C$A = v \Rightarrow \delta_{M,A,v} = 1$.
- Indirect positive ($\exists s : X_{s,A} = v \wedge X_{s,B} = w$):

$$\sum_{s \in \mathcal{S}} \Big(\text{min}(\delta_{s,A,v}, \delta_{s,B,w}) \Big) \geq 1 \quad \text{(encode via standard linearization or SAT).}$$

- Indirect negative ($\forall s : X_{s,A} = v \Rightarrow X_{s,B} \neq w$):

$$\delta_{s,A,v} \leq 1 - \delta_{s,B,w} \quad \forall s.$$

- Time order $X_{s,\text{Time}} < X_{t,\text{Time}}$: model times as ordered labels and enforce via table constraints or enumeration of consistent assignments.

A candidate M is *feasible* iff the constraints admit a solution. The puzzle is *valid* when exactly one candidate is feasible. Each generated puzzle needs to be valid by construction.

7 Generator with Hardness Control

```
// Inputs: alpha in [0,1], beta in [0,1], max_clues, target_uniqueness=true
function GENERATE_PUZZLE(alpha, beta, max_clues):
  // 1) Sample a full world
  for each suspect s and attribute A:
      X[s,A] <- Uniform choice from V_A         // values can repeat
  M <- Uniform(suspects)                        // murderer
  // 2) Build truthful clue pool
  P <- []
  P += DIRECT_POOL(X, M)                        // unary +/- and crime +/- and time
  P += COMPLEX_POOL(X, M)                       // indirect +/- (A->B), negations
  shuffle(P)

  // 3) Greedy selection under hardness controller
  C <- []                                       // selected clues
  while |C| < max_clues:
      // Draw clue type by hardness
      Z ~ Bernoulli(alpha)
      if Z == 1:
          candidates <- {c in P that are "direct" and keep truth feasible}
      else:
          // complex mode
          Y ~ Bernoulli(beta)
          if Y == 1: candidates <- {indirect positive}
          else:      candidates <- {indirect negative or negated unary}
      // Choose the candidate that minimizes #feasible murderers, tie-break:
      //  - keeps true murderer feasible
      //  - maximizes information gain (e.g., drops most spurious assignments)
      best <- argmin_c   NumFeasibleMurderers(C U {c})
      C <- C U {best}; remove best from P
       if UniqueMurderer(C): break
  return Puzzle { suspects, attributes, clues=C, truth=(X,M) }
```

Notes.

- **Uniqueness oracle** `UniqueMurderer(C)` calls the solver for each candidate m.
- To prevent trivial puzzles when α is large, cap the number of direct crime-attribute reveals (e.g., allow at most one of $\{\text{CRoom}, \text{CWeapon}, \dots\}$ to be direct).
- To increase difficulty when α is small, bias complex clues toward *indirect negatives* by lowering β.

8 Difficulty Metrics

Beyond (α, β), report:

- **Search hardness**: node expansions or SAT solve time.
- **Information gain**: reduction in feasible assignments per added clue.
- **Player difficulty proxy**: fraction of clues that are indirect or negative.

9 Example Complex Clues

- Indirect positive: "The person who was in the Kitchen had a Rope."
- Indirect negative: "The person who was in the Kitchen did not have a Rope."
- Mixed with crime: "The person whose time was 8:00 did not have the murder weapon."

10 Validation Protocol

1. **Truthfulness check**: the sampled world (X, M) satisfies every generated clue.
2. **Feasibility set size**: compute $\{m : \exists X' \text{ s.t. } (X', m) \text{ satisfies clues}\}$.
3. **Uniqueness**: accept puzzle iff the set size is exactly 1.

11 Deliverables

1. Game generator with hardness control (α, β) and uniqueness guarantee.
2. Solver supporting direct + complex clues and repeated values across suspects.
3. JSON export with suspects, attributes, clues, and hidden truth (for instructors).
4. Short presentation analyzing hardness vs. (α, β) and runtimes.

Mini Illustration (Sketch)

Suppose the hidden world has two suspects in the same Room and two with the same Weapon. With $(\alpha{=}0.3, \beta{=}0.4)$ the generator draws mostly complex clues, e.g.:

- "The person who was in the Kitchen had a Rope." (indirect positive)

- “The person who was in the Kitchen did not have a Rope.” (indirect negative; cannot both hold unless Kitchen is empty or different times; the solver ensures only the true configuration remains consistent together with other clues.)
- “The murder time was earlier than Casey’s time.” (direct time-order)

The final selected set yields exactly one feasible murderer even though attributes repeat across suspects.

Next Steps

Advanced solutions may enable cross-attribute clues and experiment with more suspects or larger domains, or integration of LLMs into storytelling. Later LLMs will be integrated for story-telling as the cues are revealed. We will also implement an AI co-Investigator to help during the investigation and find the solution.