Introduction to

Artificial Intelligence

with Python

Knowledge

knowledge-based agents

agents that reason by operating on internal representations of knowledge

If it didn't rain, Harry visited Hagrid today.

Harry visited Hagrid or Dumbledore today, but not both.

Harry visited Dumbledore today.

Harry did not visit Hagrid today.

It rained today.

Logic

AI **sentence**

an assertion about the world
in a knowledge representation language

Propositional Logic

Proposition Symbols

$P \qquad Q \qquad R$

Logical Connectives

$\neg$	$\wedge$	$\vee$
not	and	or

$\rightarrow$	$\leftrightarrow$
implication	biconditional

Not ($\neg$)

P	$\neg P$
false	true
true	false

And (∧)

P	Q	$P \wedge Q$
false	false	false
false	true	false
true	false	false
true	true	true

Or (∨)

at least one is true
XOR: exclusive or means exactly 1 is true

P	Q	$P \vee Q$
false	false	false
false	true	true
true	false	true
true	true	true

Implication ($\rightarrow$)

if P is false, nothing is implied of Q
the implication is vacuously true,
since the antecedent P cannot be satisfied
do NOT confuse this with Q being True

P	Q	$P \rightarrow Q$
false	false	true
false	true	true
true	false	false
true	true	true

Biconditional ($\leftrightarrow$)

P	Q	$P \leftrightarrow Q$
false	false	true
false	true	false
true	false	false
true	true	true

model

assignment of a truth value to every propositional symbol (a "possible world")

model

P: It is raining.

Q: It is a Tuesday.

$$\{P = \texttt{true}, Q = \texttt{false}\}$$

knowledge base

a set of sentences known by a knowledge-based agent

Entailment

"alpha entails beta"

$$\alpha \vDash \beta$$

In every model in which sentence **α** is true, sentence β is also true.

If it didn't rain, Harry visited Hagrid today.

Harry visited Hagrid or Dumbledore today, but not both.

Harry visited Dumbledore today.

Harry did not visit Hagrid today.

It rained today.

inference

the process of deriving new sentences from old ones

P: It is a Tuesday.

Q: It is raining.

R: Harry will go for a run.

KB: $(P \land \neg Q) \rightarrow R$ $\quad P \quad \neg Q$

Inference: R

Inference Algorithms

Does

$$\mathrm{KB} \models \alpha$$

?

Model Checking

Model Checking

- To determine if $\mathrm{KB} \vDash \alpha$:
  - Enumerate all possible models.
  - If in every model where KB is true, α is true, then KB entails α.
  - Otherwise, KB does not entail α.

P: It is a Tuesday. Q: It is raining. R: Harry will go for a run.

KB: $(P \land \neg Q) \rightarrow R$ $\boxed{P}$ $\boxed{\neg Q}$

Query: $\boxed{R}$

P	Q	R	KB
false	false	false	
false	false	true	
false	true	false	
false	true	true	
true	false	false	
true	false	true	true
true	true	false	
true	true	true	

P: It is a Tuesday. Q: It is raining. R: Harry will go for a run.

KB: $(P \land \neg Q) \rightarrow R$ P $\neg Q$

Query: R

P	Q	R	KB
false	false	false	false
false	false	true	false
false	true	false	false
false	true	true	false
true	false	false	false
true	false	true	true
true	true	false	false
true	true	true	false

P: It is a Tuesday. Q: It is raining. R: Harry will go for a run.

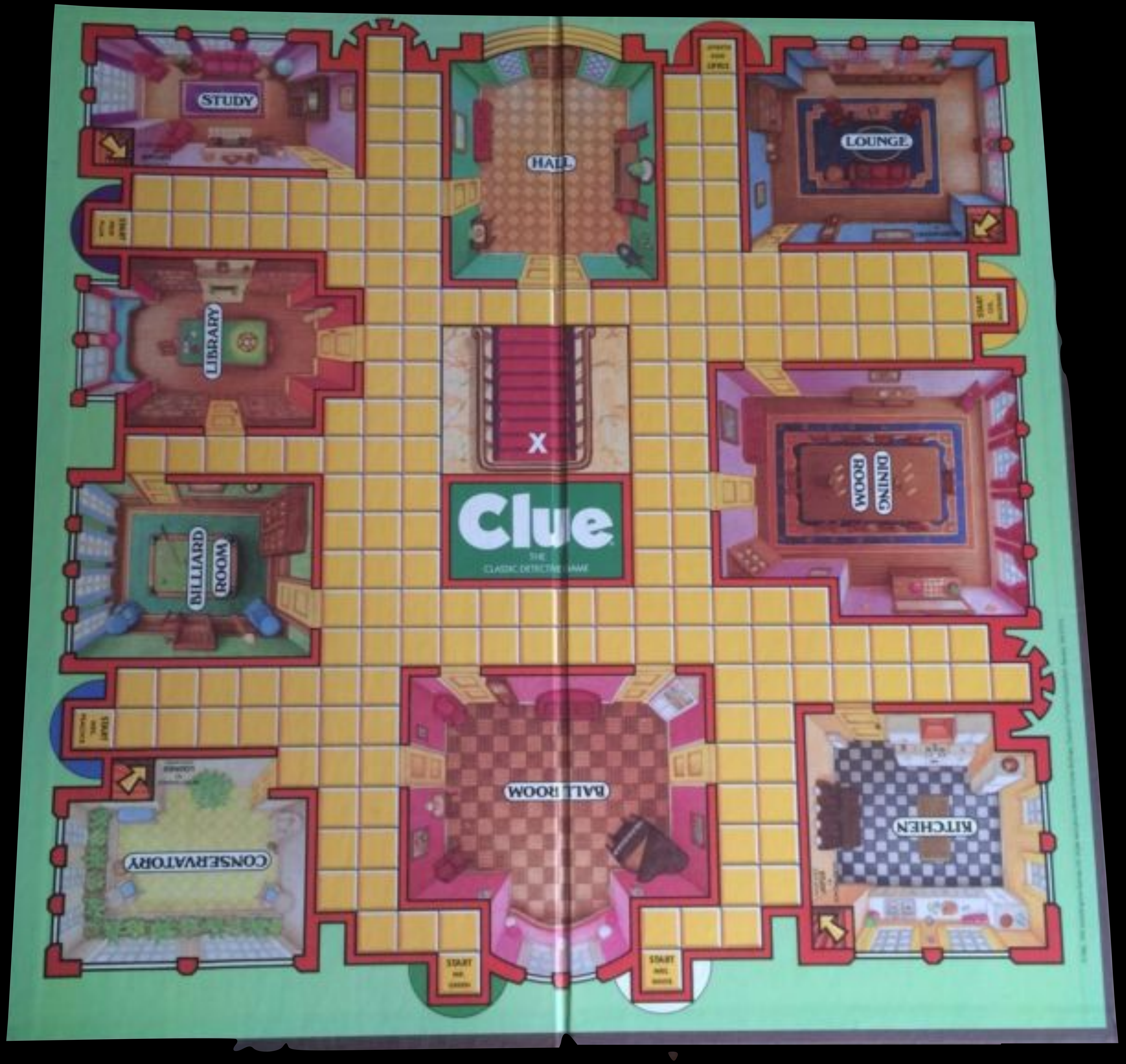
KB: $(P \land \neg Q) \rightarrow R$ P $\neg Q$

Query: R

P	Q	R	KB
false	false	false	false
false	false	true	false
false	true	false	false
false	true	true	false
true	false	false	false
true	false	true	true
true	true	false	false
true	true	true	false

Knowledge Engineering

Clue


STUDY
HALL
LOUNGE
LIBRARY
DINING ROOM
BILLIARD ROOM
Clue
CONSERVATORY
BALLROOM
KITCHEN

Clue

People	Rooms	Weapons
Col. Mustard	Ballroom	Knife
Prof. Plum	Kitchen	Revolver
Ms. Scarlet	Library	Wrench

Clue

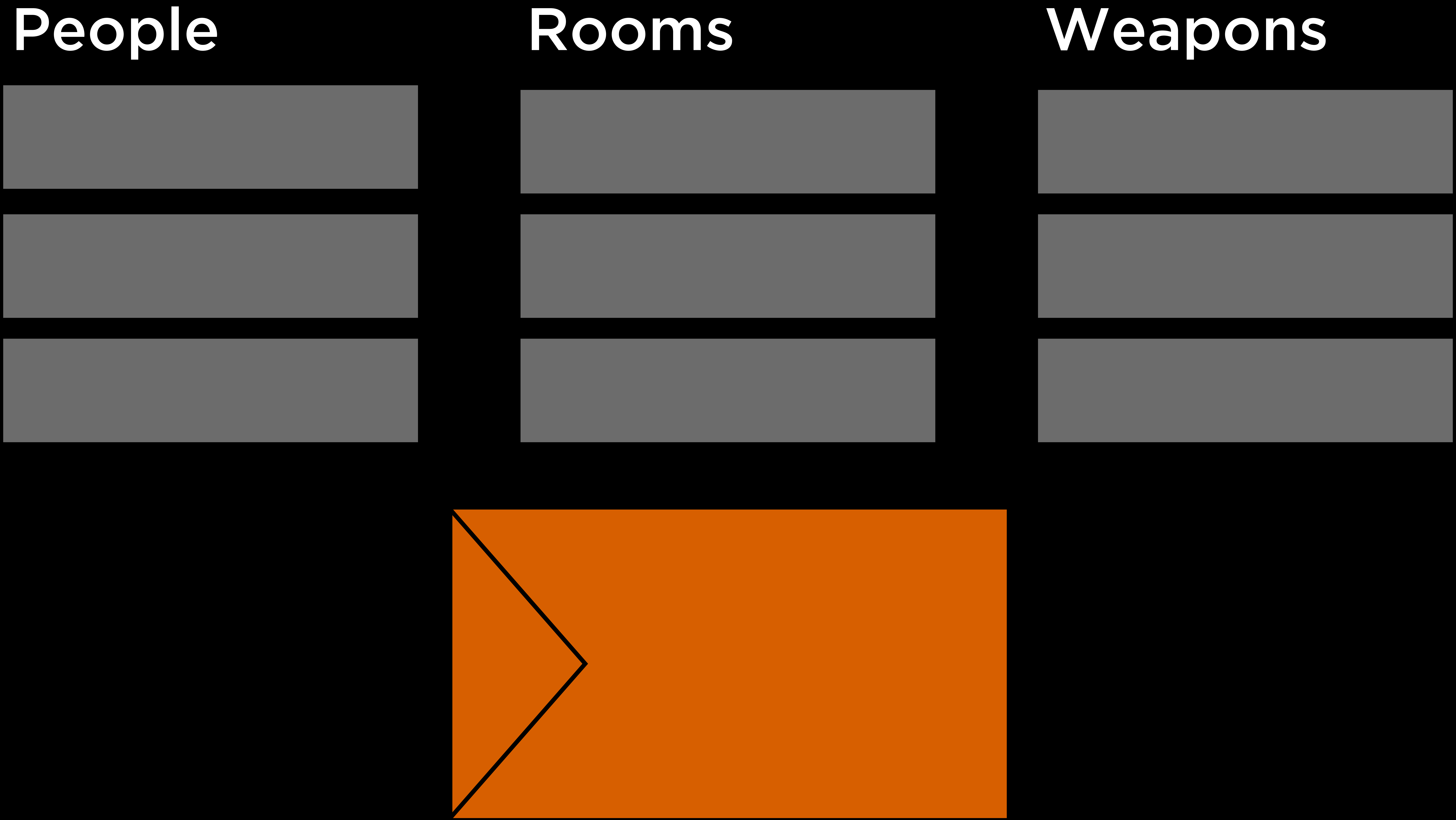
People	Rooms	Weapons

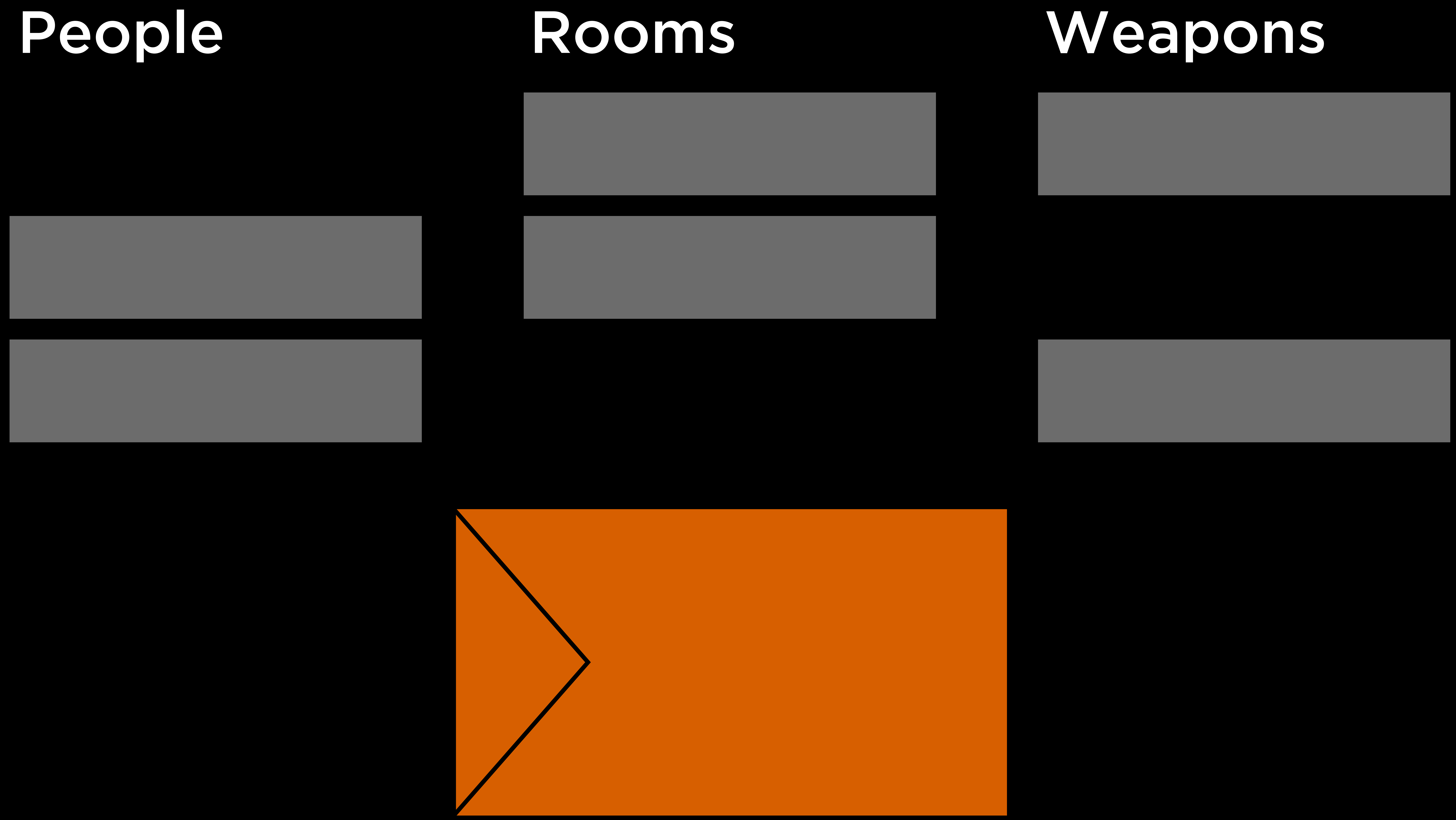
Clue

People

Rooms

Weapons


People
Rooms
Weapons


People
Rooms
Weapons

Clue

Propositional Symbols

mustard	*ballroom*	*knife*
plum	*kitchen*	*revolver*
scarlet	*library*	*wrench*

Clue

$$(mustard \lor plum \lor scarlet)$$

$$(ballroom \lor kitchen \lor library)$$

$$(knife \lor revolver \lor wrench)$$

$$\neg plum$$

$$\neg mustard \lor \neg library \lor \neg revolver$$

Logic Puzzles

- Gilderoy, Minerva, Pomona and Horace each belong to a different one of the four houses: Gryffindor, Hufflepuff, Ravenclaw, and Slytherin House.
- Gilderoy belongs to Gryffindor or Ravenclaw.
- Pomona does not belong in Slytherin.
- Minerva belongs to Gryffindor.

Logic Puzzles

Propositional Symbols

GilderoyGryffindor
GilderoyHufflepuff
GilderoyRavenclaw
GilderoySlytherin

PomonaGryffindor
PomonaHufflepuff
PomonaRavenclaw
PomonaSlytherin

MinervaGryffindor
MinervaHufflepuff
MinervaRavenclaw
MinervaSlytherin

HoraceGryffindor
HoraceHufflepuff
HoraceRavenclaw
HoraceSlytherin

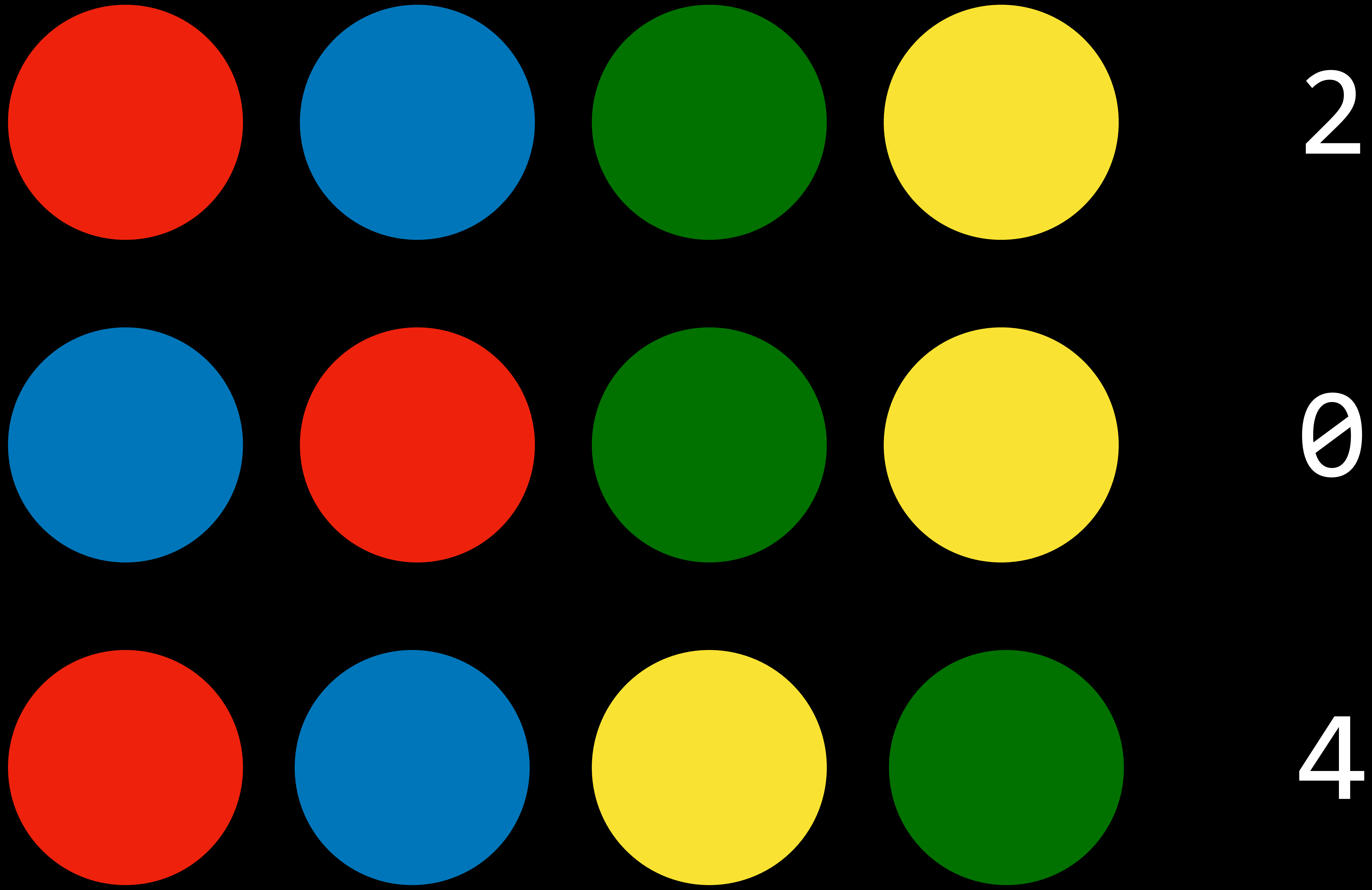
Logic Puzzles

$$(PomonaSlytherin \rightarrow \neg PomonaHufflepuff)$$

$$(MinervaRavenclaw \rightarrow \neg GilderoyRavenclaw)$$

$$(GilderoyGryffindor \lor GilderoyRavenclaw)$$

Mastermind



Inference Rules

Modus Ponens

If it is raining, then Harry is inside.

It is raining.

Harry is inside.

Modus Ponens

$$\alpha \rightarrow \beta$$

$$\alpha$$

$$\beta$$

And Elimination

Harry is friends with Ron and Hermione.

Harry is friends with Hermione.

And Elimination

$$\frac{\alpha \land \beta}{\alpha}$$

Double Negation Elimination

It is not true that Harry did not pass the test.

Harry passed the test.

Double Negation Elimination

$$\frac{\neg(\neg\alpha)}{\alpha}$$

Implication Elimination

If it is raining, then Harry is inside.

It is not raining or Harry is inside.

Implication Elimination

$$\alpha \rightarrow \beta$$

$$\neg\alpha \vee \beta$$

either:
α is not true
or
α and β is true

Biconditional Elimination

It is raining if and only if Harry is inside.

If it is raining, then Harry is inside,
and if Harry is inside, then it is raining.

Biconditional Elimination

$$\frac{\alpha \leftrightarrow \beta}{(\alpha \rightarrow \beta) \land (\beta \rightarrow \alpha)}$$

De Morgan's Law

It is not true that both
Harry and Ron passed the test.

Harry did not pass the test
or Ron did not pass the test.

or both Harry and Ron did not pass the test

De Morgan's Law

distribute NOT

flip AND $\Longleftrightarrow$ OR

$$\frac{\neg(\alpha \land \beta)}{\neg\alpha \lor \neg\beta}$$

De Morgan's Law

It is not true that
Harry or Ron passed the test.

Harry did not pass the test
and Ron did not pass the test.

De Morgan's Law

distribute NOT

flip AND $\Longleftrightarrow$ OR

$$\frac{\neg(\alpha \vee \beta)}{\neg\alpha \wedge \neg\beta}$$

Distributive Property

$$\frac{(\alpha \land (\beta \lor \gamma))}{(\alpha \land \beta) \lor (\alpha \land \gamma)}$$

Distributive Property

$$\frac{(\alpha \lor (\beta \land \gamma))}{(\alpha \lor \beta) \land (\alpha \lor \gamma)}$$

Search Problems

- initial state
- actions
- transition model
- goal test
- path cost function

Theorem Proving as a Search problem

- initial state: starting knowledge base set of all sentences known to be true
- actions: inference rules
- transition model: new knowledge base after inference prior set of true sentences + new inferences
- goal test: check statement we're trying to prove is inside KB
- path cost function: number of steps in proof minimize number of inference rules used

Resolution

(Ron is in the Great Hall) v (Hermione is in the library)

Ron is not in the Great Hall

Hermione is in the library

Unit Resolution Rule

$$P \vee Q$$

$$\neg P$$

$$Q$$

$$P \lor Q_1 \lor Q_2 \lor \ldots \lor Q_n$$

$$\neg P$$

$$Q_1 \lor Q_2 \lor \ldots \lor Q_n$$

By resolutions:
if (Ron is in the Great Hall), then (Harry is sleeping)
if (Ron is not in the Great Hall), then (Hermione is in the library)

(Ron is in the Great Hall) v (Hermione is in the library)

(Ron is not in the Great Hall) v (Harry is sleeping)

Given the conflicting clause about Ron, we know one of these must be true

(Hermione is in the library) v (Harry is sleeping)

$$P \vee Q$$

$$\neg P \vee R$$

$$Q \vee R$$

$$\frac{\begin{array}{c} P \lor Q_1 \lor Q_2 \lor \ldots \lor Q_n \\ \neg P \lor R_1 \lor R_2 \lor \ldots \lor R_m \end{array}}{Q_1 \lor Q_2 \lor \ldots \lor Q_n \lor R_1 \lor R_2 \lor \ldots \lor R_m}$$

disjunctions are propositional connected with **or**
conjunctions are propositional connected with **and**

clause

a disjunction of literals

e.g. $P \vee Q \vee R$

inside the parenthesis are disjunctions
parenthesis are connected by and

conjunctive normal form

logical sentence that is a conjunction of clauses

e.g. $(A \vee B \vee C) \wedge (D \vee \neg E) \wedge (F \vee G)$

Conversion to CNF

- Eliminate biconditionals
  - turn $(\alpha \leftrightarrow \beta)$ into $(\alpha \rightarrow \beta) \land (\beta \rightarrow \alpha)$
- Eliminate implications
  - turn $(\alpha \rightarrow \beta)$ into $\neg\alpha \lor \beta$
- Move ¬ inwards using De Morgan's Laws
  - e.g. turn $\neg(\alpha \land \beta)$ into $\neg\alpha \lor \neg\beta$
- Use distributive law to distribute ∨ wherever possible

Conversion to CNF

$(P \vee Q) \rightarrow R$

$\neg(P \vee Q) \vee R$ eliminate implication

$(\neg P \wedge \neg Q) \vee R$ De Morgan's Law

$(\neg P \vee R) \wedge (\neg Q \vee R)$ distributive law

Inference by Resolution

$$P \vee Q$$

$$\neg P \vee R$$

$$(Q \vee R)$$

$$P \lor Q \lor S$$

$$\neg P \lor R \lor S$$

$$(Q \lor S \lor R \lor S)$$

remove duplicates

$$P \lor Q \lor S$$

$$\neg P \lor R \lor S$$

$$(Q \lor R \lor S)$$

resolution of this is the
empty clause
which is always false

$$\frac{\begin{array}{c} P \\ \neg P \end{array}}{(\,)}$$

Inference by Resolution

Knowledge Base entails query α

- To determine if $\mathrm{KB} \vDash \alpha$:

for sake of proof by contradiction
suppose KB and not(α)

  - Check if $(\mathrm{KB} \wedge \neg\alpha)$ is a contradiction?
    - If so, then $\mathrm{KB} \vDash \alpha$.

from Knowledge Base,
we can conclude α is true

    - Otherwise, no entailment.

Inference by Resolution

- To determine if $\mathrm{KB} \vDash \alpha$:
  - Convert $(\mathrm{KB} \wedge \neg\alpha)$ to Conjunctive Normal Form.
  - Keep checking to see if we can use resolution to produce a new clause. resolve literals that are complementary
    - If ever we produce the **empty** clause (equivalent to False), we have a contradiction, and $\mathrm{KB} \vDash \alpha$.
    - Otherwise, if we can't add new clauses, no entailment. proof exists somewhere

Inference by Resolution

Does $(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C)$ entail A?

$$(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C) \wedge (\neg A)$$

$(A \vee B) \quad (\neg B \vee C) \quad (\neg C) \quad (\neg A)$

Inference by Resolution

Does $(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C)$ entail A?

$$(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C) \wedge (\neg A)$$

$(A \vee B)$ $\quad$ $(\neg B \vee C)$ $\quad$ $(\neg C)$ $\quad$ $(\neg A)$

Inference by Resolution

Does $(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C)$ entail A?

$$(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C) \wedge (\neg A)$$

$(A \vee B)$ $\boxed{\underline{(\neg B \vee C)} \quad \underline{(\neg C)}}$ $(\neg A)$ $\boxed{(\neg B)}$

Inference by Resolution

Does $(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C)$ entail A?

$$(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C) \wedge (\neg A)$$

$(A \vee B) \quad (\neg B \vee C) \quad (\neg C) \quad (\neg A) \quad (\neg B)$

Inference by Resolution

Does $(A \lor B) \land (\neg B \lor C) \land (\neg C)$ entail A?

$$(A \lor B) \land (\neg B \lor C) \land (\neg C) \land (\neg A)$$

$(A \lor B)$ $\quad$ $(\neg B \lor C)$ $\quad$ $(\neg C)$ $\quad$ $(\neg A)$ $\quad$ $(\neg B)$

Inference by Resolution

Does $(A \lor B) \land (\neg B \lor C) \land (\neg C)$ entail A?

$$(A \lor B) \land (\neg B \lor C) \land (\neg C) \land (\neg A)$$

$\boxed{(A \lor B)}$ $(\neg B \lor C)$ $(\neg C)$ $(\neg A)$ $\boxed{(\neg B) \quad (A)}$

Inference by Resolution

Does $(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C)$ entail A?

$$(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C) \wedge (\neg A)$$

$(A \vee B)$ $(\neg B \vee C)$ $(\neg C)$ $(\neg A)$ $(\neg B)$ (A)

Inference by Resolution

Does $(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C)$ entail A?

$$(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C) \wedge (\neg A)$$

$(A \vee B)$ $(\neg B \vee C)$ $(\neg C)$ $\underline{(\neg A)}$ $(\neg B)$ $\underline{(A)}$

Inference by Resolution

Does $(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C)$ entail A?

$$(A \vee B) \wedge (\neg B \vee C) \wedge (\neg C) \wedge (\neg A)$$

$(A \vee B)$ $(\neg B \vee C)$ $(\neg C)$ $(\neg A)$ $(\neg B)$ (A) $(\,)$

Inference by Resolution

Does $(A \lor B) \land (\neg B \lor C) \land (\neg C)$ entail A?

$$(A \lor B) \land (\neg B \lor C) \land (\neg C) \land (\neg A)$$

$$(A \lor B) \quad (\neg B \lor C) \quad (\neg C) \quad (\neg A) \quad (\neg B) \quad (A) \quad \boxed{(\,)}$$

AND False
is contradiction

First-Order Logic

Propositional Logic

<u>Propositional Symbols</u>

$MinervaGryffindor$

$MinervaHufflepuff$

$MinervaRavenclaw$

$MinervaSlytherin$

$\ldots$

First-Order Logic

represent objects

Constant Symbol

Minerva
Pomona
Horace
Gilderoy
Gryffindor
Hufflepuff
Ravenclaw
Slytherin

"functions" that take an input
and evaluate the input as True or False

Predicate Symbol

Person
House
BelongsTo

First-Order Logic

"functions" or "relations"

$Person(Minerva)$ Minerva is a person.

$House(Gryffindor)$ Gryffindor is a house.

$\neg House(Minerva)$ Minerva is not a house.

$BelongsTo(Minerva, Gryffindor)$

Minerva belongs to Gryffindor.

Universal Quantification

We know the following is True for *all* values of a variable ...

Universal Quantification

$$\forall x.\ BelongsTo(x, Gryffindor) \rightarrow \neg BelongsTo(x, Hufflepuff)$$

For all objects x, if x belongs to Gryffindor, then x does not belong to Hufflepuff.

Anyone in Gryffindor is not in Hufflepuff.

Existential Quantification

We know the following is True for *some* value of variable ...

Existential Quantification

$$\exists x.\ House(x) \land BelongsTo(Minerva, x)$$

There exists an object x such that
x is a house and Minerva belongs to x.

Minerva belongs to a house.

Existential Quantification

$$\forall x.\ Person(x) \rightarrow (\exists y.\ House(y) \wedge BelongsTo(x, y))$$

For all objects x, if x is a person, then
there exists an object y such that
y is a house and x belongs to y.

Every person belongs to a house.

Knowledge

Introduction to

Artificial Intelligence

with Python