# Lecture 8: First-order Logic

Artificial Intelligence

CS-UY-4613-A / CS-GY-6613-I

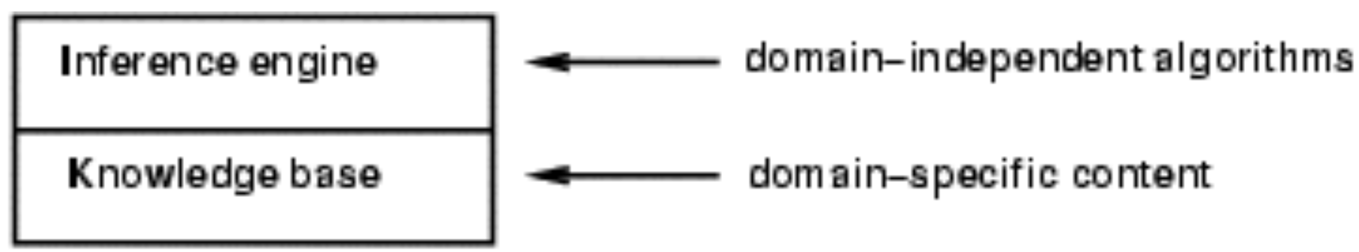
Julian Togelius

julian.togelius@nyu.edu

# Outline

- Recap: propositional logic
- First-order logic: definitions
- Midterm results and discussion

# Knowledge bases




- Knowledge base = set of sentences in a formal language
- Declarative approach; the procedural part is in the domain-general inference engine
- "Tell": add sentences to the KB
- "Ask": get answers needed for action

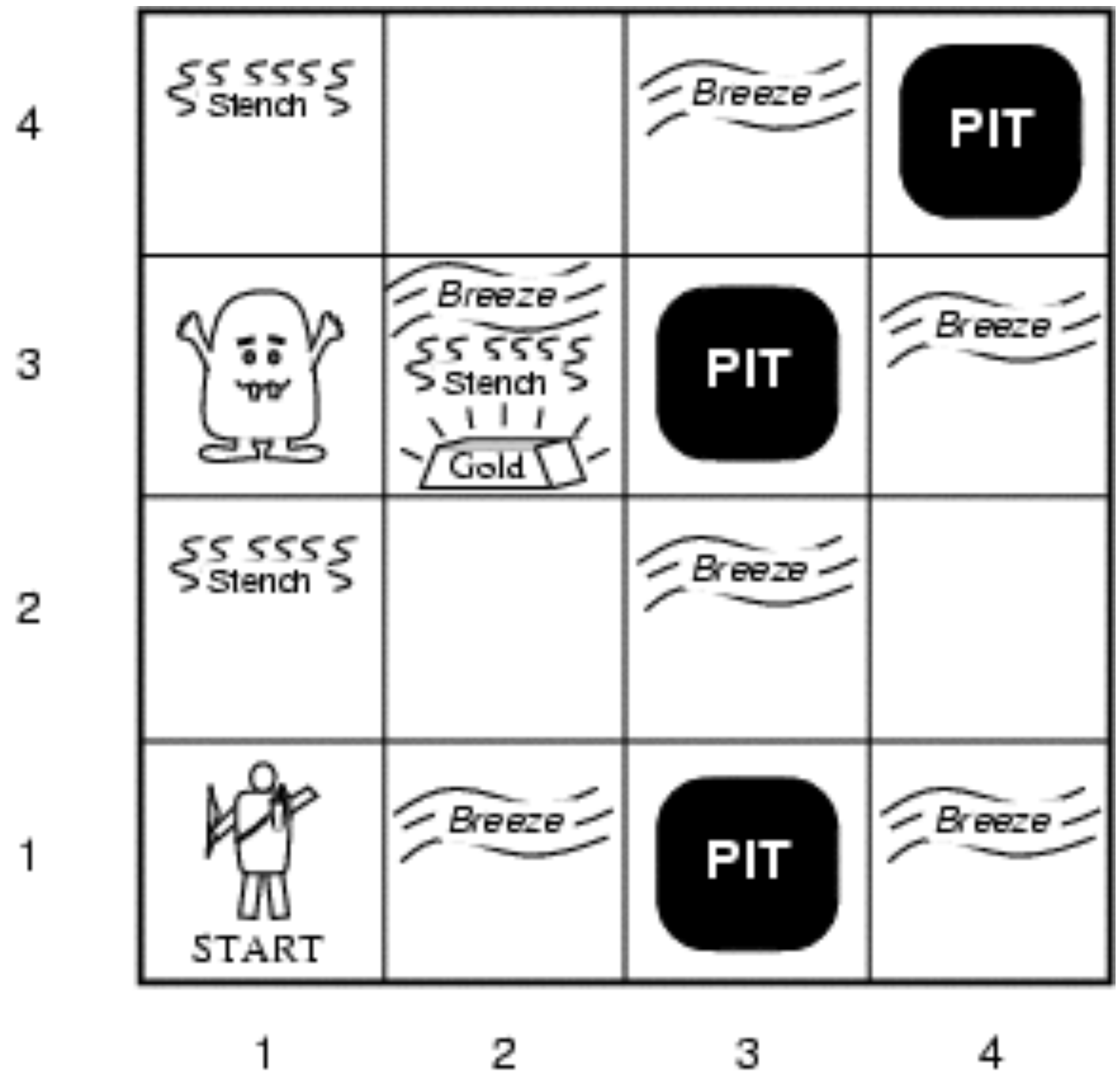
# Logic

- *Logics* are formal languages for representing information such that conclusions can be drawn
- *Syntax* defines the sentences in the language
- *Semantics* define the "meaning" of sentences; i.e., define truth of a sentence in a world

# Propositional logic: Syntax

- Propositional logic is the simplest logic – illustrates basic ideas
- The proposition symbols P1, P2 etc are sentences
  - If S is a sentence, $\neg S$ is a sentence (negation)
  - If S1 and S2 are sentences, $S1 \wedge S2$ is a sentence (conjunction)
  - If S1 and S2 are sentences, $S1 \vee S2$ is a sentence (disjunction)
  - If S1 and S2 are sentences, $S1 \Rightarrow S2$ is a sentence (implication)
  - If S1 and S2 are sentences, $S1 \Leftrightarrow S2$ is a sentence (biconditional)

# Hunt the Wumpus



# Models

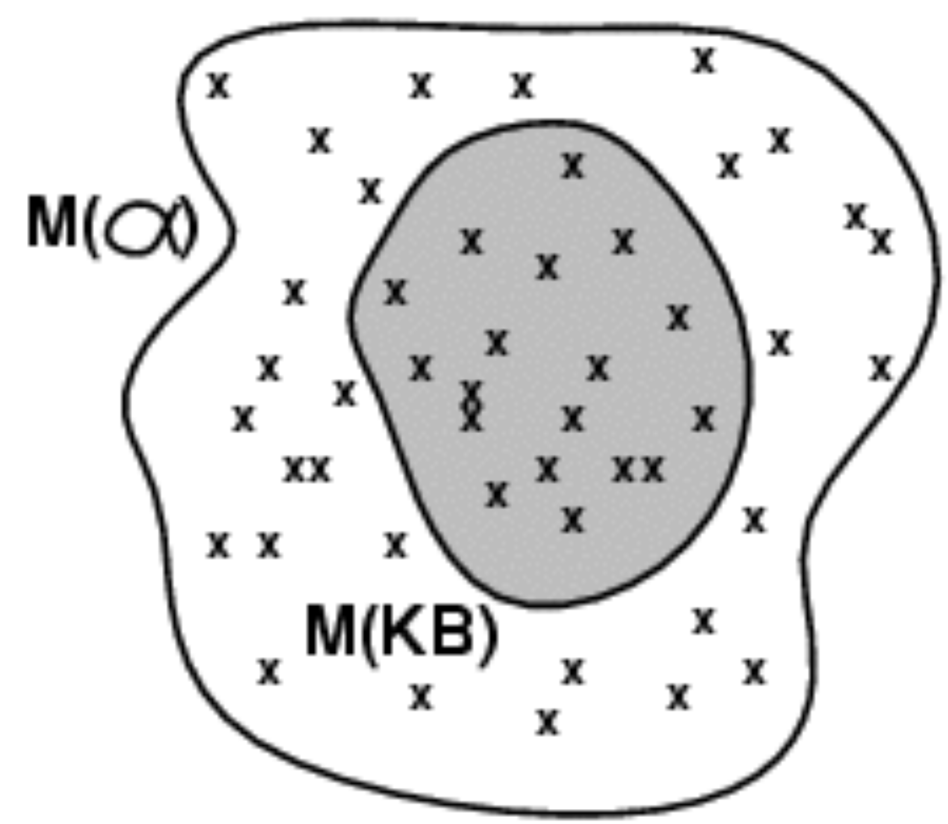
- Logicians typically think in terms of models, which are formally structured worlds with respect to which truth can be evaluated

- We say m is a model of a sentence α if α is true in m

- M(α) is the set of all models of α

- Then KB $\models$ α iff M(KB) ⊆ M(α)

- E.g. KB = Blue won and Red won
  α = Blue won

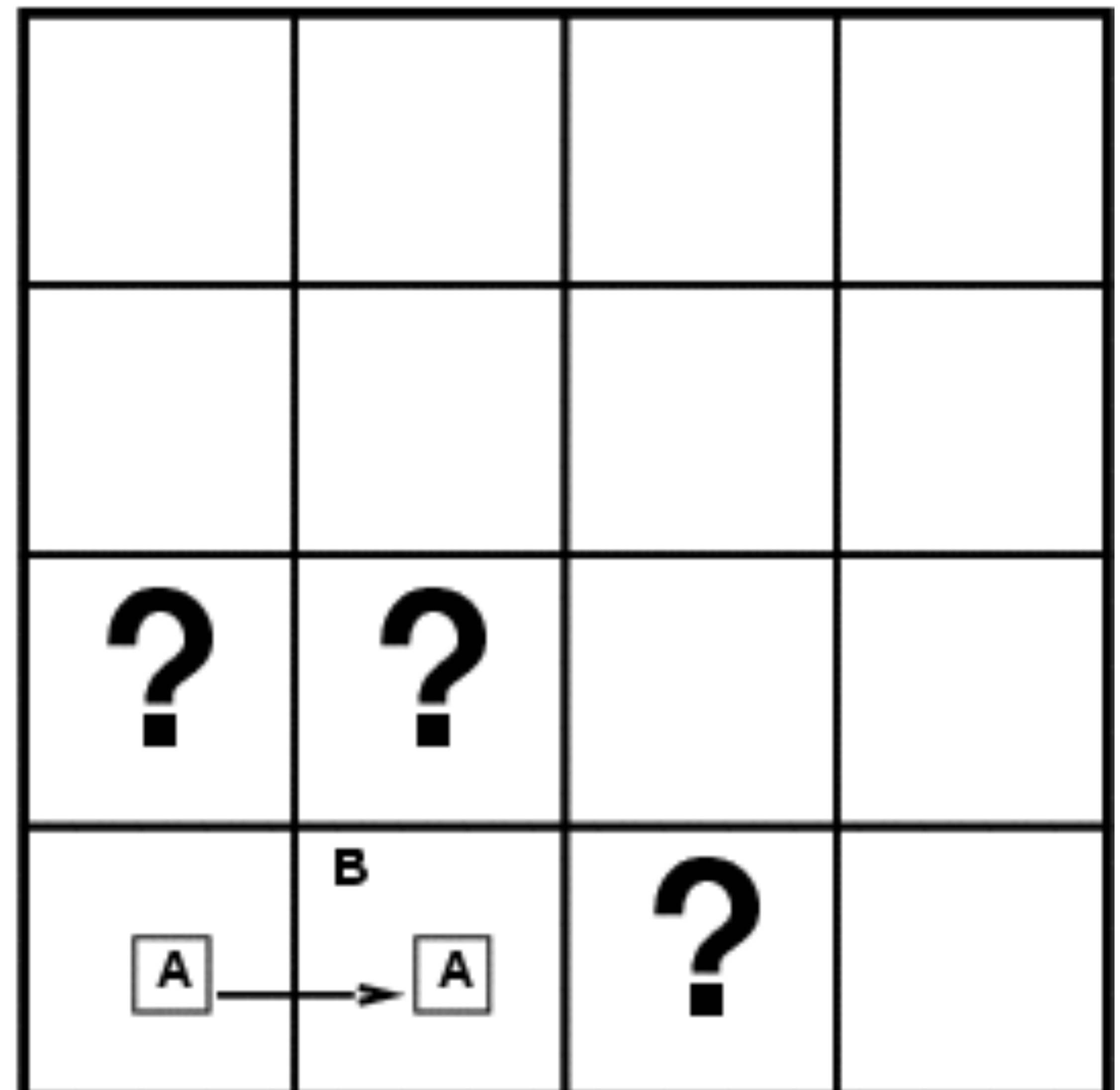


# Entailment in Wumpus

Situation after detecting nothing in [1,1], moving right, breeze in [2,1]

Consider possible models for *KB* assuming only pits

3 Boolean choices ⇒ 8 possible models



# Inference

- KB $\vdash_i \alpha$ = sentence $\alpha$ can be derived from KB by procedure i
- *Soundness*: i is sound if whenever KB $\vdash_i \alpha$, it is also true that KB $\models \alpha$
- *Completeness*: i is complete if whenever KB $\models \alpha$, it is also true that KB $\vdash_i \alpha$
- A sound and complete inference procedure: will answer any question whose answer follows from what is known by the KB.

# Pros and cons of propositional logic

- ☺️ Propositional logic is declarative
- ☺️ Propositional logic allows partial/disjunctive/negated information (unlike most data structures and databases)
- Propositional logic is compositional, meaning of $B_{1,1} \wedge P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$
- ☺️ Meaning in propositional logic is context-independent (unlike natural language, where meaning depends on context)
- ☹ Propositional logic has very limited expressive power (unlike natural language)
- E.g., cannot say "pits cause breezes in adjacent squares“, except by writing one sentence for each square

# First-order logic

- Whereas propositional logic assumes the world contains facts, first-order logic (like natural language) assumes the world contains
  - Objects: people, houses, numbers, colors, baseball games, wars…
  - Relations: red, round, prime, brother of, bigger than, part of, comes between…
  - Functions: father of, best friend, one more than, +…

# Syntax of FOL: elements

- Constants: KingJohn, 2, NYU,...
- Predicates: Brother, $>$,...
- Functions: Sqrt, LeftLegOf,...
- Variables: x, y, a, b,...
- Connectives: $\neg$, $\Rightarrow$, $\wedge$, $\vee$, $\Leftrightarrow$
- Equality: $=$
- Quantifiers: $\forall$, $\exists$

# Atomic sentences

- Atomic sentence = predicate $(term_1,...,term_n)$
  or $term_1 = term_2$

- Term = function $(term_1,...,term_n)$
  or constant or variable

- E.g., Brother(KingJohn,RichardTheLionheart)
  E.g., > ((Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))

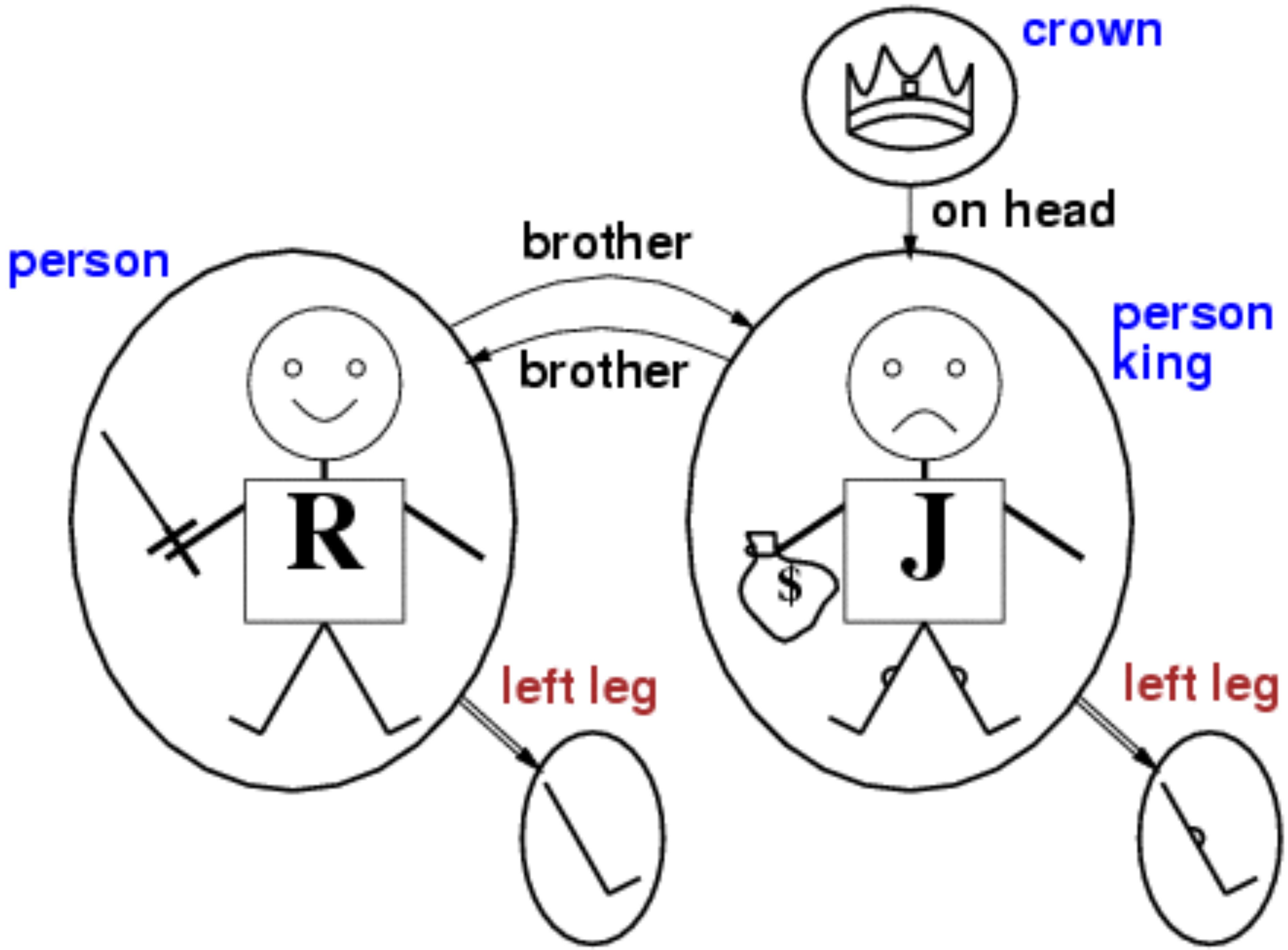
# Complex sentences

- Complex sentences are made from atomic sentences using connectives
- $\neg S$, $S_1 \wedge S_2$, $S_1 \vee S_2$, $S_1 \Rightarrow S_2$, $S_1 \Leftrightarrow S_2$,
- E.g. Sibling(KingJohn,Richard) $\Rightarrow$ Sibling(Richard,KingJohn)
- $>(1,2) \vee \leq (1,2)$
- $>(1,2) \wedge \neg >(1,2)$

# Truth in first-order logic

- Sentences are true with respect to a *model* and an *interpretation*
- Model contains objects (*domain elements*) and relations among them
- Interpretation specifies referents for
  - constant symbols → objects
  - predicate symbols → relations
  - function symbols → functional relations
- An atomic sentence predicate($term_1$,...,$term_n$) is true iff the objects referred to by $term_1$,…,$term_n$ are in the relation referred to by predicate

# Models for FOL: Example



# Universal quantification

- $\forall$<variables> <sentence>
- Everyone at NYU is smart:
  $\forall x\ At(x,NYU) \Rightarrow Smart(x)$
- $\forall x\ P$ is true in a model m iff P is true with x being each possible object in the model
- Roughly speaking, equivalent to the conjunction of instantiations of P

$$
\begin{aligned}
& At(KingJohn,NYU) \Rightarrow Smart(KingJohn) \\
\wedge\ & At(Richard,NYU) \Rightarrow Smart(Richard) \\
\wedge\ & At(NYU,NYU) \Rightarrow Smart(NYU) \\
\wedge\ & \ldots
\end{aligned}
$$

# Existential quantification

- ∃<variables> <sentence>
- Someone at NYU is smart:
- $\exists x\ At(x,NYU) \land Smart(x)$
- $\exists x\ P$ is true in a model $m$ iff $P$ is true with $x$ being some possible object in the model
- Roughly speaking, equivalent to the disjunction of instantiations of P
  $At(KingJohn,NYU) \land Smart(KingJohn)$
  $\lor At(Richard,NYU) \land Smart(Richard)$
  $\lor At(NYU,NYU) \land Smart(NYU)$
  $\lor \ldots$

# Properties of quantifiers

- $\forall x\ \forall y$ is the same as $\forall y\ \forall x$
- $\exists x\ \exists y$ is the same as $\exists y\ \exists x$
- $\exists x\ \forall y$ is not the same as $\forall y\ \exists x$
- $\exists x\ \forall y$ Loves(x,y)
  "There is a person who loves everyone in the world"
- $\forall y\ \exists x$ Loves(x,y)
  "Everyone in the world is loved by at least one person"
- Quantifier duality: each can be expressed using the other
  $\forall x$ Likes(x,IceCream) $\quad \neg\exists x\ \neg$Likes(x,IceCream)
  $\exists x$ Likes(x,Broccoli) $\quad \neg\forall x\ \neg$Likes(x,Broccoli)

# Equality

- term1 = term2 is true under a given interpretation if and only if term1 and term2 refer to the same object
- E.g., definition of Sibling in terms of Parent:
- $\forall x,y\ Sibling(x,y) \Leftrightarrow [\neg(x = y) \wedge \exists m,f\ \neg (m = f) \wedge Parent(m,x) \wedge Parent(f,x) \wedge Parent(m,y) \wedge Parent(f,y)]$

# Interacting with FOL Kbs

- Suppose a wumpus-world agent is using an FOL KB and perceives a smell and a breeze (but no glitter) at t=5:
  Tell(KB,Percept([Smell,Breeze,None],5))
  Ask(KB,∃a BestAction(a,5))

- I.e., does the KB entail some best action at t=5?
  Answer: Yes, {a/Shoot} ← substitution (binding list)

- Given a sentence S and a substitution σ,

- Sσ denotes the result of plugging σ into S; e.g.,
  S = Smarter(x,y)
  σ = {x/Hillary,y/Bill}
  Sσ = Smarter(Hillary,Bill)

- Ask(KB,S) returns some/all σ such that KB ⊨ σ

# KB for Wumpus World

- Perception
  $\forall t,s,b$ Percept([s,b,Glitter],t) $\Rightarrow$ Glitter(t)

- Reflex
  $\forall t$ Glitter(t) $\Rightarrow$ BestAction(Grab,t)

# Deducing hidden properties

- $\forall x,y,a,b\ Adjacent([x,y],[a,b]) \Leftrightarrow$

  $[a,b] \in \{[x+1,y], [x-1,y],[x,y+1],[x,y-1]\}$

- Properties of squares:
  $\forall s,t\ At(Agent,s,t) \wedge Breeze(t) \Rightarrow Breezy(s)$

- Squares are breezy near a pit:

  - Diagnostic rule---infer cause from effect
    $\forall s\ Breezy(s) \Rightarrow \exists r\ Adjacent(r,s) \wedge Pit(r)$

  - Causal rule---infer effect from cause
    $\forall r\ Pit(r) \Rightarrow \forall s\ Adjacent(r,s) \Rightarrow Breezy(s)$

# Summary

- First-order logic:
  - objects and relations are semantic primitives
  - syntax: constants, functions, predicates, equality, quantifiers
- Increased expressive power: sufficient to define wumpus world

# Midterm results

# Question 1

- Q: Will depth-first search eventually find a solution to any problem? Why/why not? [1p]

- A: No. Not if there is a loop in the search graph.

# Question 2

- Q: What is the difference between tree search and optimization? Name one algorithm of each kind. [2p]

- A:  Tree search searches forward from a start state, whereas optimization considers whole or final solutions. e.g. depth-first search and hill-climber.

# Question 3

- Q: What does it mean for a heuristic for the A* algorithm to be admissible? What happens if it is not? [2p]

- A: That the heuristic does not overestimate the distance to a solution. An inadmissible heuristic makesthe algorithm nonoptimal, as it might overlook paths to the solution.

# Question 4

- Q: Would a hill-climber or an evolution strategy be more likely to avoid getting stuck in a local optimum? Why? [1p]
- A: An evolution strategy, as it is population-based and typically not all members of the population would be stuck on the local optimum.

# Question 5

- Q: Why do we need to use a board evaluation function when using Minimax to play Chess? Name two features that could be used in such a function. [2p]

- A: Because the state space of Chess is so huge that we cannot search it exhaustively with Minimax. The function could include e.g. the difference between black and white pieces, and the number of moves available to the player.

# Question 6

- Q: Our knowledge base contains the following statements:
  $p \wedge q$
  $q \Rightarrow a$
  $a \vee b$
  Does this knowledge base entail b? Is there a model where a is false? [1p]

- A: No. No.

# Question 7

- Q: Which of the algorithms you have learnt about do you think would play Pac-Man best? Explain your answer briefly. [1p]

- A: (Any reasonable explanation that does not contain a falsehood.)

# Statistics

- Max: 10
- Average: 7.33
- Median: 8
- Min: 2

# Grade cutoffs

- A: 9
- A-: 8.5
- B+: 8
- B: 7
- B-: 6.5
- C+: 6
- C: 5
- D: 4