CS 5/7320 Artificial Intelligence

Logical Agents

AIMA Chapter 7

Slides by Michael Hahsler
based on slides by Svetlana Lazepnik
with figures from the AIMA textbook

"Exercise Plays Vital Role Maintaining Brain Health" by A Health Blog

What is logic?

Logic is a formal system for manipulating facts so that true conclusions may be drawn

Syntax: rules for constructing valid sentences

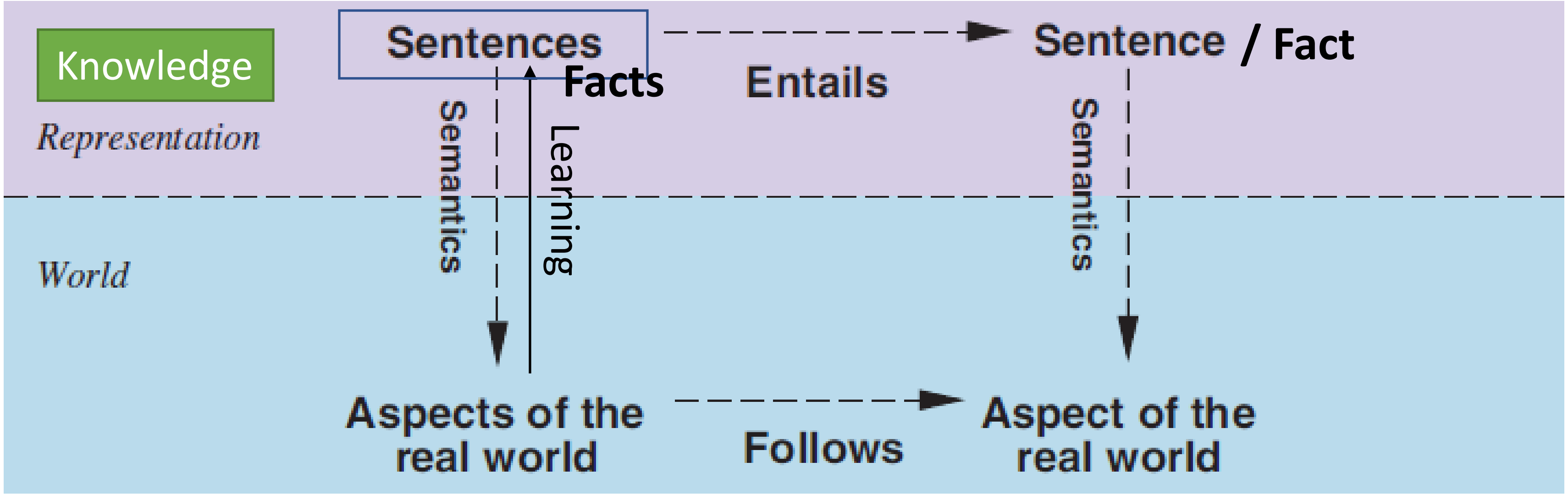
E.g., $x + 2 \geq y$ is a valid arithmetic sentence, $\geq x2y +$ is not

Semantics: “meaning” of sentences, or relationship between logical sentences and the real world

Specifically, semantics defines truth of sentences

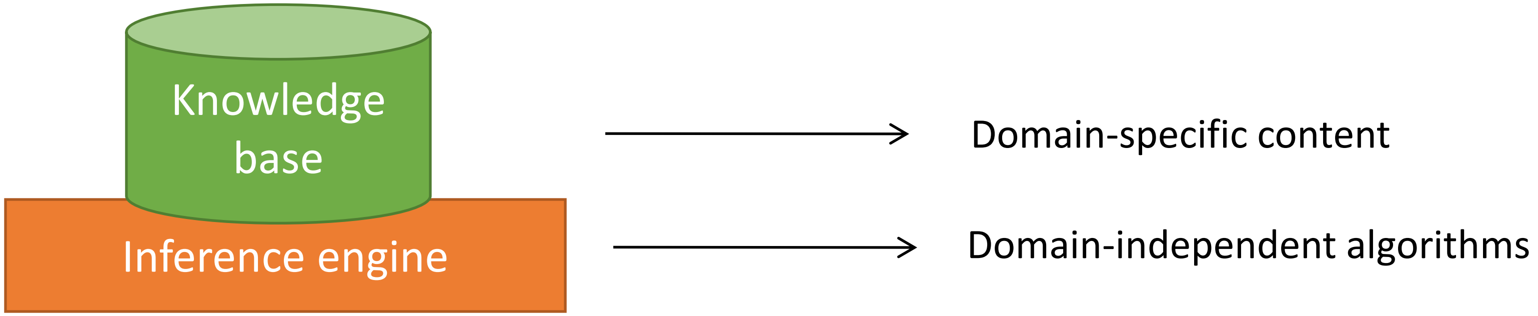
E.g., $x + 2 \geq y$ is true in a world where $x = 5$
and $y = 7$

Relation between Knowledge Representation and the World



- **Facts:** Sentences we know to be true.
- **Possible worlds**: all worlds/models which are consistent with the facts we know (compare with belief state).
- Learning new facts reduces the number of possible worlds.
- **Entailment:** A sentence logically follows from what we already know.

Knowledge-based agents



- Knowledge base (KB) = set of **sentences** in a **formal** language (knowledge representation) that are known to be true = **set of facts**
- Declarative approach to building an agent: Define what it needs to know
- Distinction between data (knowledge) and program (inference)
- Fullest realization of this philosophy was in the field of expert systems or knowledge-based systems in the 1970s and 1980s

Generic Knowledge-based Agent

function KB-AGENT(*percept*) **returns** an *action*
  persistent: *KB*, a knowledge base
      t, a counter, initially 0, indicating time

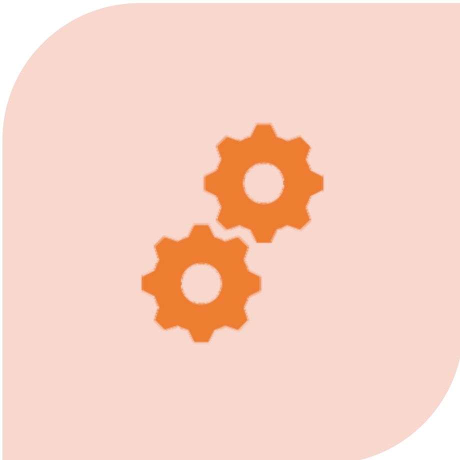
  TELL(*KB*, MAKE-PERCEPT-SENTENCE(*percept*, *t*))
  action $\leftarrow$ ASK(*KB*, MAKE-ACTION-QUERY(*t*))
  TELL(*KB*, MAKE-ACTION-SENTENCE(*action*, *t*))
  $t \leftarrow t + 1$
  return *action*

Memorize percept at time t

Ask for logical action

Record action taken at time t

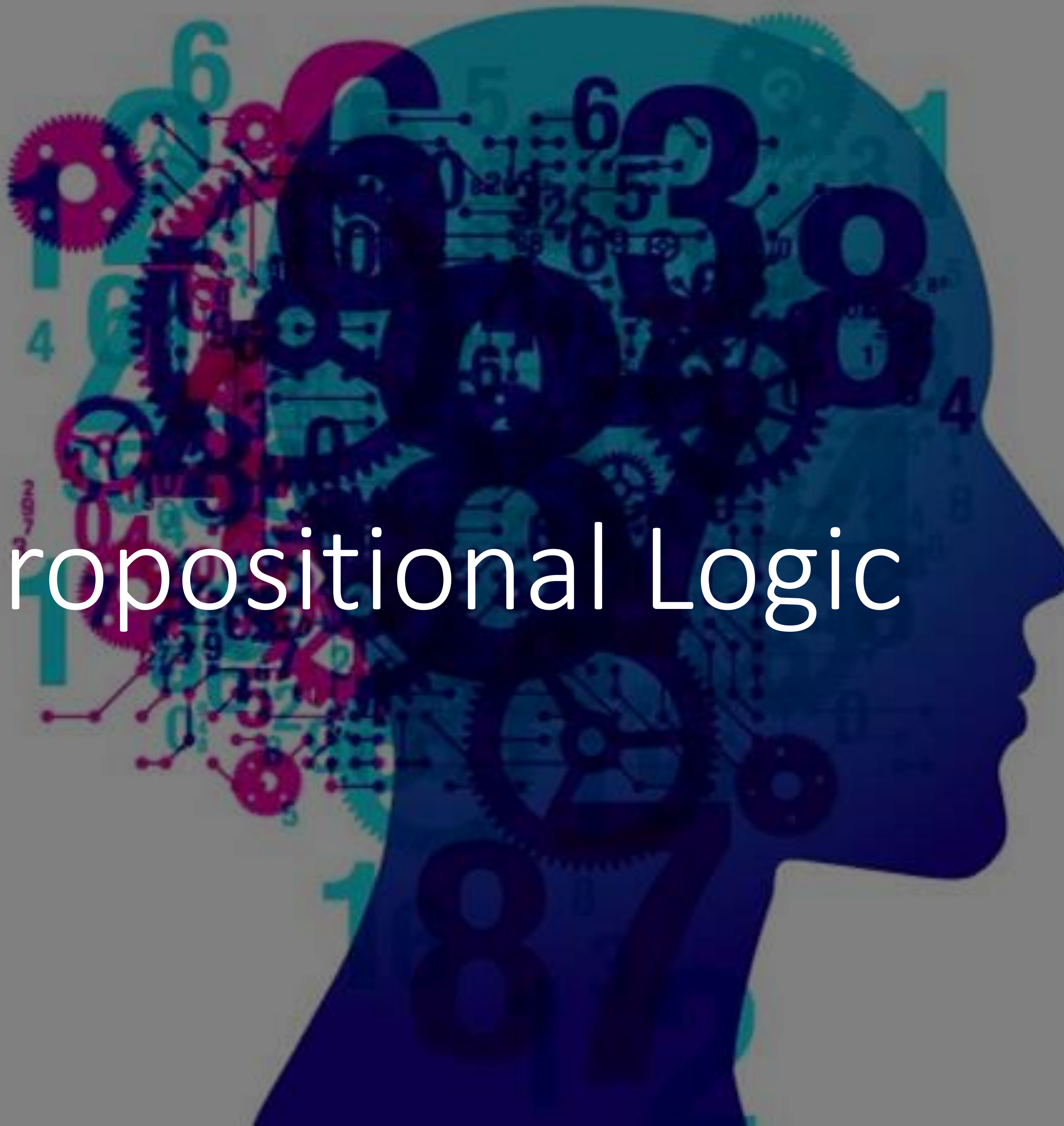
Overview



PROPOSITIONAL LOGIC

FIRST ORDER LOGIC

Propositional Logic



Propositional logic: Syntax in BN-Form

$$
\begin{array}{rcll}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} & \\
\mathit{AtomicSentence} & \rightarrow & \mathit{True} \mid \mathit{False} \mid P \mid Q \mid R \mid \ldots & \text{= Symbols} \\
\mathit{ComplexSentence} & \rightarrow & (\ \mathit{Sentence}\ ) & \\
 & \mid & \neg\ \mathit{Sentence} & \text{Negation} \\
 & \mid & \mathit{Sentence} \land \mathit{Sentence} & \text{Conjunction} \\
 & \mid & \mathit{Sentence} \lor \mathit{Sentence} & \text{Disjunction} \\
 & \mid & \mathit{Sentence} \ \Rightarrow\ \mathit{Sentence} & \text{Implication} \\
 & \mid & \mathit{Sentence} \ \Leftrightarrow\ \mathit{Sentence} & \text{Biconditional} \\
 & & & \\
\textsc{Operator Precedence} & : & \neg, \land, \lor, \Rightarrow, \Leftrightarrow &
\end{array}
$$

Validity and Satisfiability

A sentence is **valid** if it is true in **all** models (called a tautology)	e.g., *True*, $A \vee \neg A$, $A \Rightarrow A$, $(A \wedge (A \Rightarrow B)) \Rightarrow B$ useful to deduct new sentences.
A sentence is **satisfiable** if it is true in **some** model	e.g., $A \vee B$, C useful to find new facts that satisfy all possible worlds.
A sentence is **unsatisfiable** if it is true in no models	e.g., $A \wedge \neg A$

Possible Worlds, Models and Truth Tables

A **model** specifies a "possible world" with the true/false status of each proposition symbol in the knowledge base

- E.g., **P** is true and **Q** is true
- With two symbols, there are $2^2 = 4$ possible worlds/models, and they can be enumerated exhaustively using:

A **truth table** specifies the truth value of a composite sentence for each possible assignments of truth values to its atoms. Each row is a model.

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

We have 3 possible worlds for $P \Rightarrow Q = true$

Propositional logic: Semantics

Rules for evaluating truth with respect to a model:

- $\neg$**P** is true iff **P** is false
- **P** $\wedge$ **Q** is true iff **P** is true and **Q** is true
- **P** $\vee$ **Q** is true iff **P** is true or **Q** is true
- **P** $\Rightarrow$ **Q** is true iff **P** is false or **Q** is true

Sentence

Model

Logical equivalence

Two sentences are logically equivalent iff (read if, and only if) they are true in same models

$$
\begin{aligned}
(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) && \text{commutativity of } \wedge \\
(\alpha \vee \beta) &\equiv (\beta \vee \alpha) && \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) && \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma)) && \text{associativity of } \vee \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \vee \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta) && \text{de Morgan} \\
\neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta) && \text{de Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) && \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) && \text{distributivity of } \vee \text{ over } \wedge
\end{aligned}
$$

Entailment

- **Entailment** means that a sentence follows from the premises contained in the knowledge base:

$$\boldsymbol{KB \models \alpha}$$

- The knowledge base *KB* entails sentence α *iff* α is true in all models where *KB* is true
  - E.g., KB with x = 0 entails sentence x * y = 0
- Tests for entailment
  - $\boldsymbol{KB \models \alpha}$ iff $(\boldsymbol{KB \Rightarrow \alpha})$ is *valid*
  - $\boldsymbol{KB \models \alpha}$ iff $(\boldsymbol{KB \wedge \neg \alpha})$ is *unsatisfiable*

Inference

- **Logical inference:** a procedure for generating sentences that follow from a knowledge base KB

- An inference procedure is **sound** if it derives a sentence α iff KB $\models \alpha$. I.e, it only derives **true sentences**.

- An inference procedure is **complete** if it can derive **all** α for which $KB \models \alpha$.

Inference

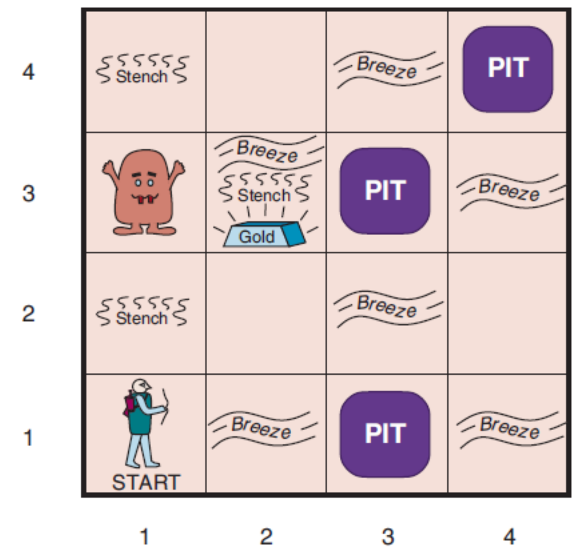
- How can we check whether a sentence α is entailed by KB?
- How about we **enumerate all possible models of the KB** (truth assignments of all its symbols), and check that α is true in every model in which KB is true?
  - Is this sound (if an answer is produced, it is correct)?
  - Is this complete (guaranteed to produce the correct answer)?
  - Problem: if KB contains *n* symbols, the truth table will be of size 2^n
- Better idea: use ***inference rules***, or sound procedures to generate new sentences or *conclusions* given the *premises* in the KB

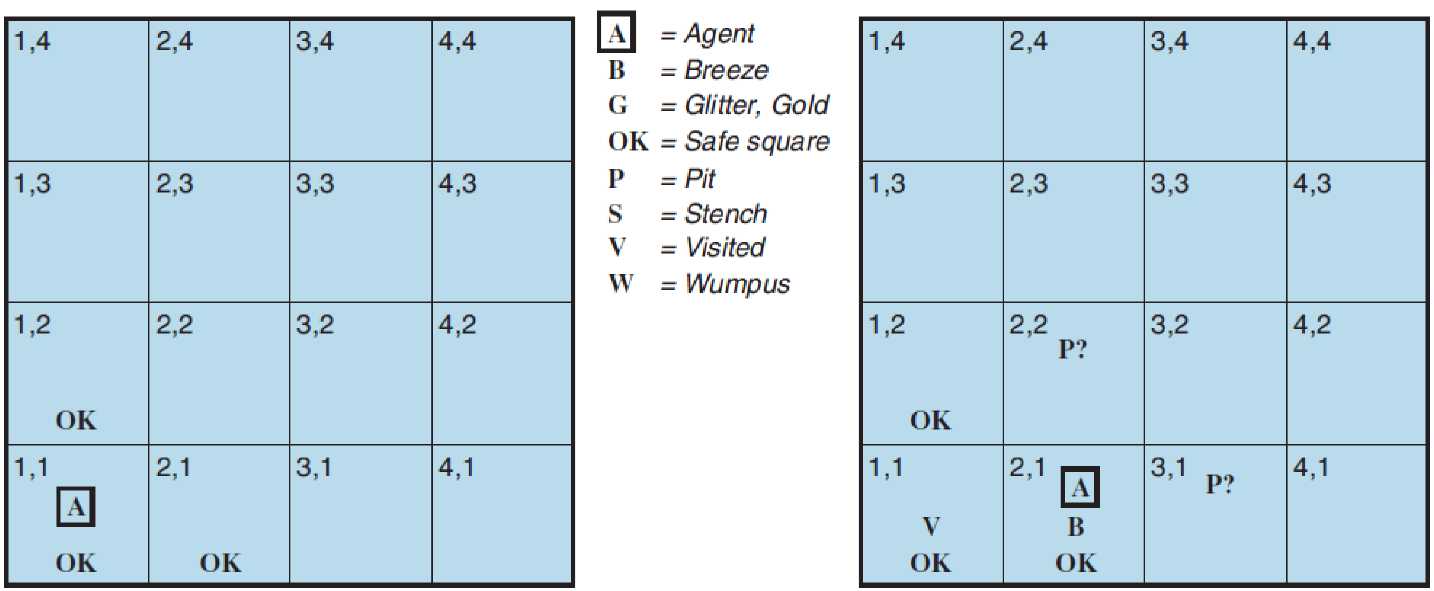
Complexity of inference

- Propositional inference is ***co-NP-complete***
  - *Complement* of the SAT problem: $\alpha \models \beta$ if and only if the sentence $\alpha \land \neg \beta$ is *unsatisfiable*
  - Every known inference algorithm has worst-case exponential running time

- Efficient inference possible for restricted cases
  - e.g., Horn clauses are disjunctions of literals with at most one positive literal.

Example: Wumpus World





(a)

(b)

Example: Wumpus World

Initial KB needs to contain rules like these for each square:
$Breeze(1,1) \Leftrightarrow Pit(1,2) \lor Pit(2,1)$
$Breeze(1,2) \Leftrightarrow Pit(1,1) \lor Pit(1,3) \lor Pit(2,2)$
$\text{Stench}(1,1) \Leftrightarrow W(1,2) \lor W(2,1)$
…

Percepts at (1,1) are no breeze or stench. Add the following facts to the KB:
$\neg Breeze(1,1)$
$\neg Stench(1,1)$

Inference will tell us that the following facts are entailed:
$\neg Pit(1,2), \neg Pit(2,1), \neg W(1,2), \neg W(2,1)$

This means that (1,2) and (2,1) are safe.

Initial KB needs to contain rules like these for each square:
$Breeze(1,1) \Leftrightarrow Pit(1,2) \lor Pit(2,1)$
$Breeze(1,2) \Leftrightarrow Pit(1,1) \lor Pit(1,3) \lor Pit(2,2)$
$\text{Stench}(1,1) \Leftrightarrow W(1,2) \lor W(2,1)$
…

Percepts at (1,1) are no breeze or stench. Add the following facts to the KB:
$\neg Breeze(1,1)$
$\neg Stench(1,1)$

Inference will tell us that the following facts are entailed:
$\neg Pit(1,2), \neg Pit(2,1), \neg W(1,2), \neg W(2,1)$

This means that (1,2) and (2,1) are safe.

Summary

- Logical agents apply inference to a knowledge base to derive new information and make decisions
- Basic concepts of logic:
  - syntax: formal structure of sentences
  - semantics: truth of sentences in models
  - entailment: necessary truth of one sentence given another
  - inference: deriving sentences from other sentences
  - soundness: derivations produce only entailed sentences
  - completeness: derivations can produce all entailed sentences
- Resolution is complete for propositional logic
- Algorithms use forward, backward chaining, are linear in time, and complete for special clauses (definite clauses).

First-Order Logic

Limitations of propositional logic

- Suppose you want to say “All humans are mortal”
  - In propositional logic, you would need ~6.7 billion statements of the form:

    MichaelIsHuman and MichaelIsMortal,

    SarahIsHuman and SarahIsMortal, …

- Suppose you want to say “Some people can run a marathon”
  - You would need a disjunction of ~6.7 billion statements:

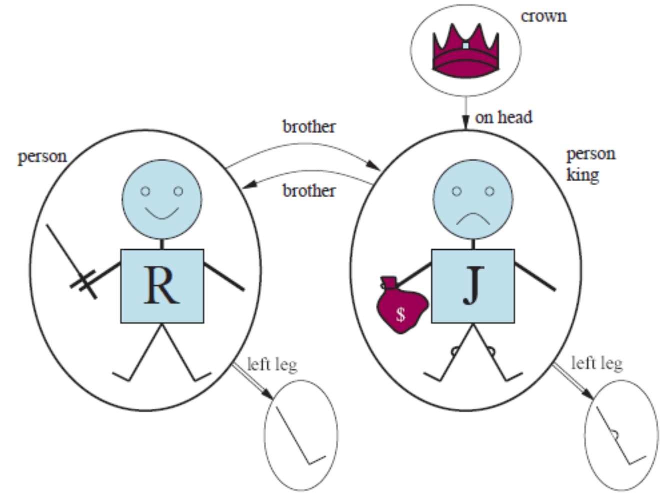
    MichaelcanRunAMarathon or … or SarahCanRunAMarathon

Other Languages to represent Knowledge

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $\in [0, 1]$
Fuzzy logic	facts with degree of truth $\in [0, 1]$	known interval value

- First-order Logic adds **objects** and **relations.**

Syntax of FOL



$$
\begin{array}{rcl}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} & \rightarrow & \mathit{Predicate} \mid \mathit{Predicate}(\mathit{Term},\ldots) \mid \mathit{Term} = \mathit{Term} \\
\mathit{ComplexSentence} & \rightarrow & (\ \mathit{Sentence}\ ) \\
 & \mid & \neg\ \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \land \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \lor \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \ \Rightarrow\ \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \ \Leftrightarrow\ \mathit{Sentence} \\
 & \mid & \mathit{Quantifier}\ \mathit{Variable},\ldots\ \mathit{Sentence} \\
 & & \\
\mathit{Term} & \rightarrow & \mathit{Function}(\mathit{Term},\ldots) \\
 & \mid & \mathit{Constant} \\
 & \mid & \mathit{Variable} \\
 & & \\
\mathit{Quantifier} & \rightarrow & \forall \mid \exists \\
\mathit{Constant} & \rightarrow & A \mid X_1 \mid \mathit{John} \mid \cdots \\
\mathit{Variable} & \rightarrow & a \mid x \mid s \mid \cdots \\
\mathit{Predicate} & \rightarrow & \mathit{True} \mid \mathit{False} \mid \mathit{After} \mid \mathit{Loves} \mid \mathit{Raining} \mid \cdots \\
\mathit{Function} & \rightarrow & \mathit{Mother} \mid \mathit{LeftLeg} \mid \cdots \\
\textsc{Operator Precedence} & : & \neg, =, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

Objects

Relations. Predicate is/returns True or False

Function returns an object

Universal quantification

- $\forall$**x P(x)**

- Example: "Everyone at SMU is smart"
  $\forall$**x At(x,SMU)** $\Rightarrow$ **Smart(x)**
  Why not $\forall$**x At(x,SMU)** $\wedge$ **Smart(x)**?

- Roughly speaking, equivalent to the conjunction of all possible instantiations of the variable:
  [At(John, SMU) $\Rightarrow$ **Smart(John)]** $\wedge$ **...**
  [At(Richard, SMU) $\Rightarrow$ **Smart(Richard)]** $\wedge$ **...**

- $\forall$**x P(x)** is true in a model m iff **P(x)** is true with **x** being each possible object in the model

Existential quantification

- **$\exists$x P(x)**

- Example: “Someone at SMU is smart”
  $\exists$x At(x,SMU) $\wedge$ Smart(x)
  Why not **$\exists$x At(x,SMU) $\Rightarrow$ Smart(x)**?

- Roughly speaking, equivalent to the disjunction of all possible instantiations:
  [At(John,SMU) $\wedge$ Smart(John)] $\vee$
  [At(Richard,SMU) $\wedge$ Smart(Richard)] $\vee$ …

- **$\exists$x P(x)** is true in a model m iff **P(x)** is true with **x** being some possible object in the model

Inference in FOL

- Inference is complicated.

1. **Reduction to propositional logic** and then use propositional logic inference.

2. **Directly do inference on FOL (or a subset like definite clauses)**
   - Unification: Combine two sentences into one.
   - Forward Chaining for FOL
   - Backward Chaining for FOL
   - Logical programming (e.g., Prolog)

Limitations of Logic

- What if we cannot collect enough knowledge to make a decision
  e.g., the environment if only partially observable
- What about facts about the future?
  grade(Michael, AI, this_semester)

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $\in [0, 1]$
Fuzzy logic	facts with degree of truth $\in [0, 1]$	known interval value