CS 5/7320 Artificial Intelligence

Knowledge-Based Agents AIMA Chapters 7-9

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





Outline

Knowledge-Based Agents

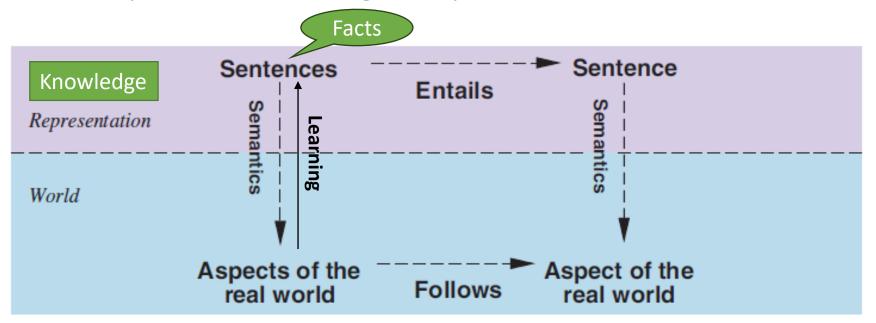
Logic

- Propositional
- First-Order

Large Language Models



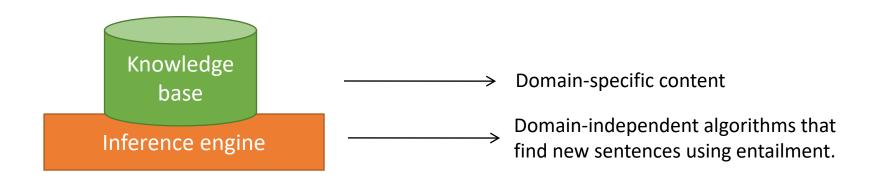
Reality vs. Knowledge Representation



- 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 new sentence logically follows from what we already know.



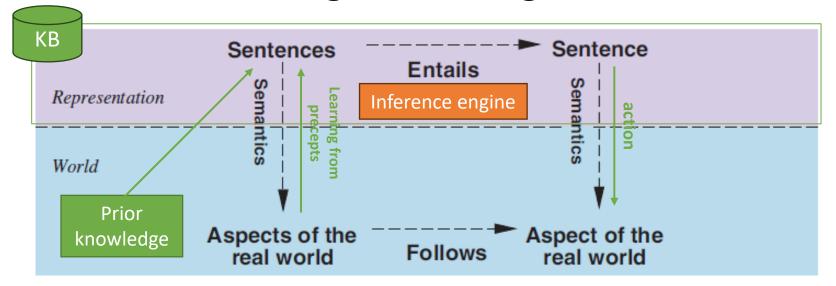
Knowledge-Based Agents



- 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 in its KB.
- Separation between data (knowledge) and program (inference).
- Actions are based on knowledge (sentences + inferred sentences) + an
 objective function. E.g., the agent knows the effects of 5 possible actions
 and chooses the action with the largest utility.



Generic Knowledge-based Agent





Outline

Knowledge-Based Agents

Logic

- Propositional
- First-Order

Large Language Models



Knowledge is Often Represented Using Logic



Logic is a formal system for representing and manipulating facts (i.e., knowledge) so that true conclusions may be drawn



Syntax: rules for constructing valid sentences

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



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

Specifically, semantics defines truth of sentences

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

Outline

Knowledge-Based Agents

Logic

- Propositional Logic
- First-Order

Large Language Models

Propositional logic: Syntax in Backus-Naur Form

OPERATOR PRECEDENCE : $\neg, \land, \lor, \Rightarrow, \Leftrightarrow$

```
Sentence \rightarrow AtomicSentence \mid ComplexSentence
AtomicSentence \rightarrow True \mid False \mid P \mid Q \mid R \mid \dots = Symbols
ComplexSentence \rightarrow (Sentence)
\mid \neg Sentence \mid Sentence \mid Conjunction
\mid Sentence \wedge Sentence \mid Disjunction
\mid Sentence \Rightarrow Sentence \mid Implication
\mid Sentence \Leftrightarrow Sentence \mid Biconditional
```

Validity and Satisfiability

A sentence is **valid** if it is true in **all** models/worlds

e.g., *True*, $A \lor \neg A$, $A \Rightarrow A$, $(A \land (A \Rightarrow B)) \Rightarrow B$ are called tautologies and are useful to deduct new sentences.

A sentence is satisfiable if it is true in some model

e.g., AVB, C useful to find new facts that satisfy all current possible worlds.

A sentence is unsatisfiable if it is true in no models

e.g., A∧¬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	$\neg P$	$P \wedge Q$	$P \lor Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
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:

```
iff P
                                   is false
       is true
• ¬P
• P \wedge Q is true
                     iff P
                                  is true and Q
                                                            is true
                     iff P
• P \lor Q is true
                                  is true
                                            or Q
                                                            is true
                    iff P
                                  is false or
• \mathbf{P} \Rightarrow \mathbf{Q} is true
                                                   Q
                                                           is true
```

Sentence Model

Logical equivalence

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

```
(\alpha \wedge \beta) \equiv (\beta \wedge \alpha) commutativity of \wedge
           (\alpha \vee \beta) \equiv (\beta \vee \alpha) commutativity of \vee
((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma)) associativity of \wedge
((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma)) associativity of \vee
             \neg(\neg\alpha) \equiv \alpha double-negation elimination
       (\alpha \Rightarrow \beta) \equiv (\neg \beta \Rightarrow \neg \alpha) contraposition
       (\alpha \Rightarrow \beta) \equiv (\neg \alpha \lor \beta) implication elimination
      (\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) biconditional elimination
       \neg(\alpha \land \beta) \equiv (\neg \alpha \lor \neg \beta) de Morgan
       \neg(\alpha \lor \beta) \equiv (\neg \alpha \land \neg \beta) de Morgan
(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) distributivity of \wedge over \vee
(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) distributivity of \vee over \wedge
```

Entailment

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

$$KB \models \alpha$$

- 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
 - KB $\models \alpha$ iff (KB $\Rightarrow \alpha$) is valid
 - $KB = \alpha$ iff $(KB \land \neg \alpha)$ is unsatisfiable

Inference

• Logical inference: a procedure for generating sentences that follow from (ar entailed by) 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?
 - This is sound: All produced answer are correct.
 - This is complete: It will produce all correct answers.
 - **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.
- Look at the textbook for inference rules and resolution.

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

Example: Wumpus World

4 \$\frac{\fr

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 A OK	2,1 OK	3,1	4,1

A	= Agent
В	= Breeze
\mathbf{G}	= Glitter, Gold
OK	= Safe square
P	= Pit
\mathbf{S}	= Stench
\mathbf{V}	= Visited
W	= Wumpus

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2 P?	3,2	4,2
1,1 V OK	2,1 A B OK	3,1 P?	4,1

(b)

(a)

Example: Wumpus World

Initial KB needs to contain rules like these for each square:

```
Breeze(1,1) \Leftrightarrow Pit(1,2) \vee Pit(2,1)

Breeze(1,2) \Leftrightarrow Pit(1,1) \vee Pit(1,3) \vee Pit(2,2)

Stench(1,1) \Leftrightarrow W(1,2) \vee 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.

We have to enumerate all possible scenarios in propositional logic! First-order logic can help.

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).

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:

MichaellsHuman and MichaellsMortal, SarahlsHuman and SarahlsMortal, ...

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

First-order logic can help with that by adding objects and relations to the facts represented by propositional logic.

Outline

Knowledge-Based Agents

Logic

- Propositional Logic
- First-Order Logic

Large Language Models

Different Languages to Represent Knowledge

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

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

Syntax of FOL

```
Sentence \rightarrow AtomicSentence \mid ComplexSentence
AtomicSentence \rightarrow Predicate \mid Predicate (Term, ...) \mid Term = Term
ComplexSentence \rightarrow (Sentence) \mid \neg Sentence \mid Sentence \land Sentence \mid Sentence \lor Sentence \mid Sentence \lor Sentence \mid Sentence \Rightarrow Sentence \mid Sentence \Rightarrow Sentence \mid Quantifier Variable, ... Sentence
Term \rightarrow Function (Term, ...)
```

person brother on head person king

R left leg left leg

| Constant | Objects

Quantifier $\rightarrow \forall \mid \exists$

 $Constant \rightarrow A \mid X_1 \mid John \mid \cdots$

 $Variable \rightarrow a \mid x \mid s \mid \cdots$

 $Predicate \rightarrow True \mid False \mid After \mid Loves \mid Raining \mid \cdots$

 $Function \rightarrow Mother \mid LeftLeg \mid \cdots \blacktriangleleft$

_ ..

Function returns an object

Relations. Predicate

is/returns True or False

OPERATOR PRECEDENCE : $\neg, =, \land, \lor, \Rightarrow, \Leftrightarrow$

Universal quantification

- ∀x P(x)
- Example: "Everyone at SMU is smart"
 ∀x At(x,SMU) ⇒ Smart(x)
 Why not ∀x At(x,SMU) ∧ 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

- ∃x P(x)
- Example: "Someone at SMU is smart"
 ∃x At(x,SMU) ∧ Smart(x)
 Why not ∃x At(x,SMU) ⇒ Smart(x)?
- Roughly speaking, equivalent to the disjunction of all possible instantiations:

```
[At(John,SMU) ∧ Smart(John)] ∨ [At(Richard,SMU) ∧ Smart(Richard)] ∨ ...
```

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

Inference in FOL

Inference in FOL 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 randomness?
- 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 First-order logic Temporal logic Probability theory Fuzzy logic	facts facts, objects, relations facts, objects, relations, times facts facts with degree of truth $\in [0,1]$	true/false/unknown true/false/unknown true/false/unknown degree of belief $\in [0,1]$ known interval value



+ Natural Language

facts, objects, relations, ...

555



Outline

Knowledge-Based Agents

Logic

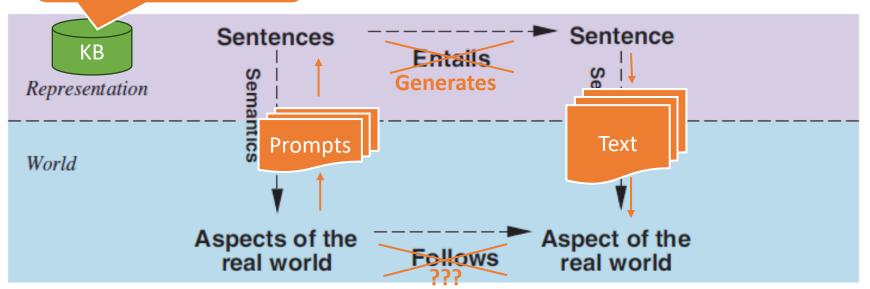
- Propositional
- First-Order

Large Language Models



Using Natural Language for Knowledge Representation

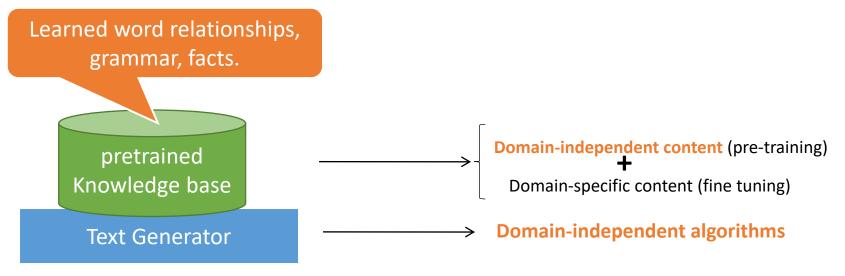
Pretrained model knows words relationship, grammar, facts.



- The user formulates a question about the real world as a natural language prompt (a sequence of tokens).
- The LLM generates text using a model representing its knowledge base.
- The text (hopefully) is useful in the real world. The **objective function** is not clear. Maybe it is implied in the prompt?



LLM as a Knowledge-Based Agents



Current text generators are:

- Pretrained decoder-only transformer models (e.g., GPT stands for Generative Pre-trained Transformer). The knowledge base is not updated during interactions.
- Tokens are created autoregressively. One token is generated at a time based on all the previous tokens using the transformer attention mechanism.



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 generating text (generate new tokens using autoregression)

Record action taken at time t

Conclusion

- The clear separation between knowledge and inference engine is very useful.
- **Pure logic** is often not flexible enough. The fullest realization of knowledge-based agents using logic was in the field of expert systems or knowledge-based systems in the 1970s and 1980s.
- Pretrained Large Language Models are an interesting new application of knowledge-based agents based on natural language.
- Next, we will talk about probability theory which is the standard language to reason under uncertainty and forms the basis of machine learning.

