Propositional Logic and First-order Logic

Yifeng Zeng

yfzeng@cs.aau.dk

Introduction

- Name: Yifeng Zeng
- PhD at National University of Singapore(2005)
- Course Teaching: Agent Programming(http://www.cs.aau.dk/ yfzeng/course/index.html)
  - BDI(Belief, Desire and Intention) Agent
  - Communicating, Coordination and Cooperation(Contract net, auction, negotiation, $\cdots$)
  - Decision Theory, Game Theory
  - Planning and Learning
- Research Areas
  - Multiagent Sequential Decision Making, Multiagent Systems
    - Opponent Modeling, Computer Games, and Robotics

Outline

- Contents
  - Propositional Logic
  - First-order Logic
  - Inference
  - Logic Programming: Prolog
- Literatures
  - http://www.cs.aau.dk/ yfzeng/course/AIP/
  - Artificial Intelligence: A Modern Approach (by Stuart J. Russell and Peter Norvig)
  - Prolog Programming for Artificial Intelligence (Second or Third versions, by Ivan Bratko)
    - Online materials
    - Prolog Programming: A First Course(By Paul Brna)
    - Logic, Programming and Prolog (2ed) (By Ulf Nilsson and Jan Maluszynski)

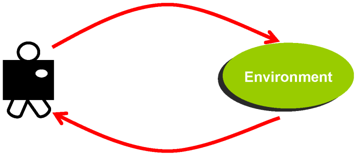
Outline

- Exercise Session
  - Reading and Problem Solving
  - TA(Kamal:MED) and Yifeng(CS)
- Examination
  - Mini-project submission before the examination
  - Questions from mini-project in the examination
- One Q&A session before the examination?

Today

- Knowledge and Reasoning
- Logic Types
- Propositional Logic
- Inference Rules
- First-order Logic

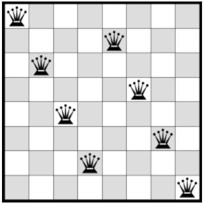
Knowledge-based Agent



- The agent must be able to:
    - Represent environmental states, actions and observations, etc.
    - Incorporate new percepts
    - Deduce hidden properties of the world
    - Deduce appropriate actions
    - Update internal representation

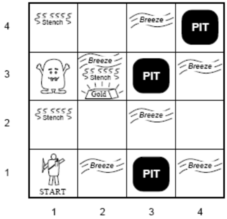
Example 1: Eight Queens Puzzle

- No two queens would be able to attack each other
- A solution requires that no two queens share the same row, column, or diagonal



Example 2: Wumpus World

- Grid
  - One wumpus, one gold, and pits
- Adjacent Grid
  - Adjacency: Left, right, top, or bottom
- Observations(Percepts)
  - Stench: Grids having or adjacent to wumpus
  - Breeze: Grids adjacent to a pit
- Agent
  - Goal: Grab the gold
  - Die if it enters into a pit or meets with the wumpus
  - Carry only one arrow and shoot the wumpus along a straight line
- Agent's optimal plans
  - Need reasoning!!!



Logic

- Logics are formal languages for representing information such that conclusions can be drawn
  - Syntax: define the sentences in the language(how to make sentence)
  - Semantic: define the meaning of sentences(the relation between the sentences and the states of affairs)
- Example
  - $x + 2 > y$ is a sentence; $x2 + y >$ is not a sentence
  - $x + 2 > y$ is true in a world where x=1, y=1

Proof Theory

- A set of rules for deducing the **entailments** of a set of sentences
- Entailment
  - Entailment means that one thing follows from another: $KB \models \alpha$
  - α is true in all worlds where KB is true
    - E.g., $x + y = 4$ entails $4 = x + y$
    - E.g., the KB containing *"A is correct"* and *"B is correct"* entails *"Either A or B is correct"*

Types of Logics

Language	*What exists*	*Beliefs of agent*
Propositional Logic	Facts	T/F/Unknown
First-order Logic	Facts, Objects, Relations	T/F/Unknown
Temporal Logic	Facts, Objects, Relations, Times	T/F/Unknown
Probability Theory	Facts	Degree of belief [0,1]
Fuzzy Logic	Facts+Degree of truth	Known interval value

Propositional Logic: Syntax

- Propositional logic is the simplest logic - illustrates basic ideas

Sentence	$\longrightarrow$	*Atomic Sentence* $\mid$ *ComplexSentence*
Atom	$\longrightarrow$	*True* $\mid$ *False* $\mid$ *A proposition*
ComplexSentence	$\longrightarrow$	$\neg$ *Sentence* $\mid$ *Sentence Connective Sentence*
Connective	$\longrightarrow$	$\wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow$

Propositional Logic: Semantics

- Specify how to compute the truth value of *any* sentence
- Some rules

$\neg S$	is true iff	S	is false		
$S_1 \wedge S_2$	is true iff	S_1	is true and	S_2	is true
$S_1 \vee S_2$	is true iff	S_1	is true or	S_2	is true
$S_1 \Rightarrow S_2$	is true iff	S_1	is false or	S_2	is true
i.e.,	is false iff	S_1	is true and	S_2	is false
$S_1 \Leftrightarrow S_2$	is true iff	$S_1 \Rightarrow S_2$	is true and	$S_2 \Rightarrow S_1$	is true

Assume S_1, S_2 and S_3 are *true*, *true* and *false*
Evaluate: $\neg S_1 \wedge (S_2 \vee S_3)$

Inference

- A sentence is valid if it is true in **all** models
  - E.g., *True*, $A \vee \neg A$, $A \Rightarrow A$
- Entailment($\alpha \models \beta$): derive one sentence from another
- Validity is connected to inference via the **Deduction Theorem**:
  - $\alpha \models \beta$ if and only if $\alpha \Rightarrow \beta$ is valid
- A sentence is satisfiable if it is true in some model
  - E.g., $A \vee B$, C
- A sentence is unsatisfiable if it is true in no model
  - E.g., $A \wedge \neg A$
- $\alpha \Rightarrow \beta$ is valid
  - $\neg\alpha \vee \beta$ is satisfiable
  - $\alpha \wedge \neg\beta$ is unsatisfiable

Inference Rules

- Deduce new sentences from knowledge bases
- Modus Ponens(MP) or Implication Elimination

$$\frac{\alpha \Rightarrow \beta, \quad \alpha}{\beta}$$

- Unit Resolution

$$\frac{\alpha \vee \beta, \quad \neg \beta}{\alpha}$$

Examples: Resolution

- Given $\alpha \vee \beta$ and $\neg\beta \vee \gamma$, deduce ???
- Given $\neg\alpha \Rightarrow \beta$ and $\beta \Rightarrow \gamma$, deduce ???

Automated Reasoning - 1

- Given a set of sentences, what could be deduced?
  - Formulate as propositional logic
  - Use inference rules to deduce the goal

Example: Computer V.S. Agent

- Knowledge Base
  - If computer is an agent, then it is intelligent; but if it is not an agent, then it is a stupid machine
  - If computer is either intelligent or a machine, then it is powerful
  - If computer is powerful, then it is useful
- Conclusion
  - Computer is Agent?Powerful?Useful?

Procedure 1: Formulation

- Formulate Propositions
    - CA: Computer is an agent
    - CS: Computer is stupid
    - CM: Computer is a machine
    - CU: Computer is useful
    - CP: Computer is powerful
- If computer is an agent, then it is intelligent; but if it is not an agent, then it is a stupid machine
    - P_1: $CA \Rightarrow \neg CS$
    - P_2:$\neg CA \Rightarrow CS \wedge CM$
- If computer is either intelligent or a machine, then it is powerful
    - P_3:$(CM \vee \neg CS) \Rightarrow CP$
- If computer is powerful, then it is useful
    - P_4:$CP \Rightarrow CU$

Procedure 2: Deduction

$$
\begin{array}{ll}
P_1: & CA \Rightarrow \neg CS \\
P_2: & \neg CA \Rightarrow CS \wedge CM \\
P_3: & (CM \vee \neg CS) \Rightarrow CP \\
P_4: & CP \Rightarrow CU
\end{array}
$$

- From P_1 and P_2, we get P_5: $\neg CS \vee (CS \wedge CM)$
- From P_5 and P_3, we get P_6: CP
- From P_6 and P_4, we get P_7: CU

Automated Reasoning - 2

- In general, the inference problem is NP complete
- If we restrict ourselves to Horn sentences, then repeated use of Modus Ponens gives us a polynomial time procedure.
  - $P_1 \wedge P_2 \wedge \cdots \wedge P_n \Rightarrow Q$

Automated Reasoning - 3

- Knowledge base contains: $P_1 \wedge P_2 \wedge \cdots \wedge P_n$, and we have a goal G
  - $P_1 \wedge P_2 \wedge \cdots \wedge P_n \Rightarrow G$
- Put $\neg G$ into *KB* and prove whether the whole set of propositions are (in)consistent
  - If we cannot derive *false*, it is satisfiable($P_1 \wedge P_2 \wedge \cdots \wedge P_n \wedge \neg G$)

Automated Reasoning - 4

- Forward Chaining
  - Start from known facts and rules, deduce new ones, add them into *KB*, $\cdots$
- Backward Chaining
  - Start from the goal, check whether the goal is matched in the right hand side, then replace the goal with a new goal, until facts are found in *KB*

Pros and Cons of Propositional Logic

- Propositional logic is declarative: pieces of syntax correspond to facts
- Propositional logic allows partial/disjunctive/negated information(unlike most data structures and databases)
- Propositional logic is compositional
- Meaning in propositional logic is context-independent(unlike natural language, where meaning depends on context)
- Propositional logic has very limited expressive power
  - E.g., cannot say ”pits cause breezes in adjacent grids” except by writing one sentence for each square

First-order Logic - 1

- Propositional logic assumes world contains *facts* while first-order logic(like natural language) assumes the world contains *Objects*, *Relations* and *Functions*
- Syntax

Constant	A,10,PC, $\cdots$
Variable	x,y,a,$\cdots$
Predicate	$After$,$>$,$Mother$,$\cdots$
Function	$LeftSideOf$,$Cubic$,$Consine$,$\cdots$

First-order Logic - 2

Sentence	$\longrightarrow$	Atomic Sentence
	$\mid$	Sentence Connective ComplexSentence
	$\mid$	Quantifier Variables, $\cdots$ Sentence
	$\mid$	$\neg$ Sentence
Atom	$\longrightarrow$	Predicate(Term,$\cdots$) $\mid$ Term=Term
Term	$\longrightarrow$	Function(Term,$\cdots$)$\mid$ Constant $\mid$ Variable
Connective	$\longrightarrow$	$\wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow$
Quantifier	$\longrightarrow$	$\forall \mid \exists$

Notes -1

- Predicate *V.S.* Function
  - Predicate returns **True** or **False**
  - Function returns any **value**
    - E.g., Everyone loves mother
- Two types of quantifies are NOT commutative
  - $\exists x \, \forall y \, \exists z \quad P(x, y, z)$
  - $\forall y \, \exists x \, \exists z \quad P(x, y, z)$

Notes -2

- Typically, $\Rightarrow$ is the main connective with $\forall$
  - Everyone at AAU is smart
    - $\forall x \quad At(x, AAU) \Rightarrow Smart(x)$
  - $\forall x \quad At(x, AAU) \wedge Smart(x)$
    - Everyone is at AAU and everyone is smart
- Typically, $\wedge$ is the main connective with $\exists$
  - Someone at DTU is smart
    - $\exists x \quad At(x, DTU) \wedge Smart(x)$
  - $\exists x \quad At(x, DTU) \Rightarrow Smart(x)$
    - If someone is at DTU then he/she is smart

Examples

- Not all students take both AIP and AP(Agent Programming) courses
  - $\neg[\forall x, Student(x) \Rightarrow Take(AIP, x) \wedge Take(AP, x)\ ]$
  - $\exists x, Student(x) \Rightarrow [\neg Take(AIP, x) \vee \neg Take(AP, x)]$
- Only one student failed AP
  - $\exists x, [Student(x) \wedge Fail(AP, x) \wedge \forall y, [\neg(x = y) \wedge Student(y) \Rightarrow \neg Fail(AP, y)]]$
- Only one student failed both AIP and AP
  - $\exists x, [Student(x) \wedge Fail(AP, x) \wedge Fail(AIP, x) \wedge \forall y, [\neg(x = y) \wedge Student(y) \Rightarrow \neg Fail(AP, y) \vee \neg Fail(AIP, y)]]$

More

- The best grade in AIP is better than the best grade in AP
  - $\forall x, [Student(x) \wedge Take(AIP, x) \Rightarrow [\exists y, Student(y) \wedge Take(AP, y) \wedge Greater(Score(AIP, x), Score(AP, y))]]$
- No one likes a game unless the game is funny
  - $\forall x, [Game(x) \wedge \neg Funny(x) \Rightarrow \forall y \neg Likes(x, y)]$
- Politicians can fool some of the people all of the time, and they can fool all of the people some of the time, but they cannot fool all of the people all of the time
  - $\forall x \;\; Politician(x) \Rightarrow (\exists y \; \forall t \;\; Person(y) \wedge Fools(x, y, t)) \wedge (\exists t \; \forall y \;\; Person(y) \Rightarrow Fools(x, y, t)) \wedge \neg(\forall t, y \;\; Person(y) \Rightarrow Fools(x, y, t))$