MidTerm

- Q1) Under what circumstances would it make sense to go “down hill” (i.e. explore a child whose heuristic value is worse than the parent’s) when executing the hill-climbing search algorithm? When would it *not* make sense?

Question 2

- Explain how does the Genetic Algorithm work? Why is it considered as Local Search Algorithm
- And Explain!!!
- local search algorithms are

useful for solving pure **optimization**

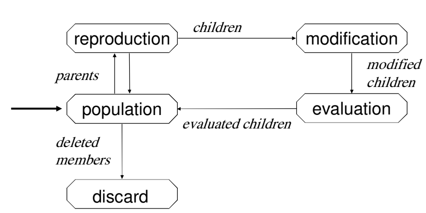
problems, in which the aim is to find th
objective function.

In such cases, can use iterative

improvement algorithms; keep a

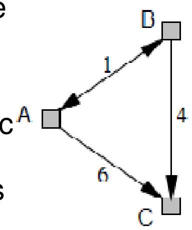
single “current" state, try to improve it

The GA Cycle of Reproduction



Question 3

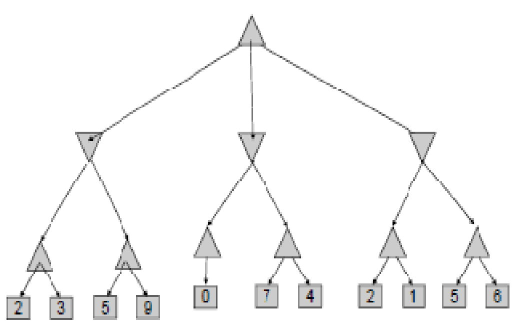
- The following diagram represents the state space of a deterministic problem, with each arrow denoting a possible operator (labeled with the step cost). Assume that the successors of a state are generated in alphabetical order, and that there is no repeated state checking.
- Show the search tree generated by breadth-first searc applied to the problem of staring in A, where C is the goal. Circle the tree node that the search identifies as the solution.



- What is the branching factor here?
- Which of the following algorithms will find solution in this case breadth first, depth first?

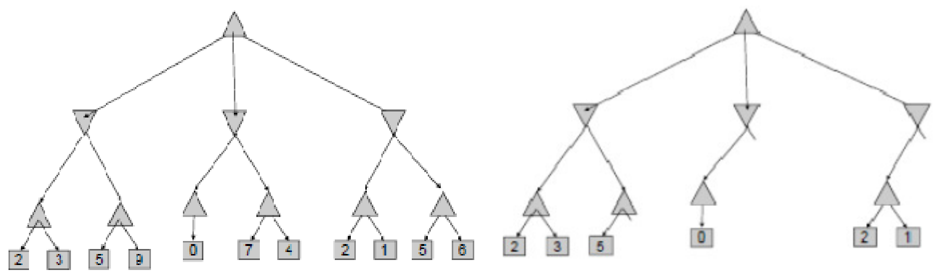
Question 4

- Apply the minimax algorithm to the game tree below, where it is the maximizer's turn to play. The values of the evaluation function of the leaves are listed.
- Write the values of the intermediate nodes

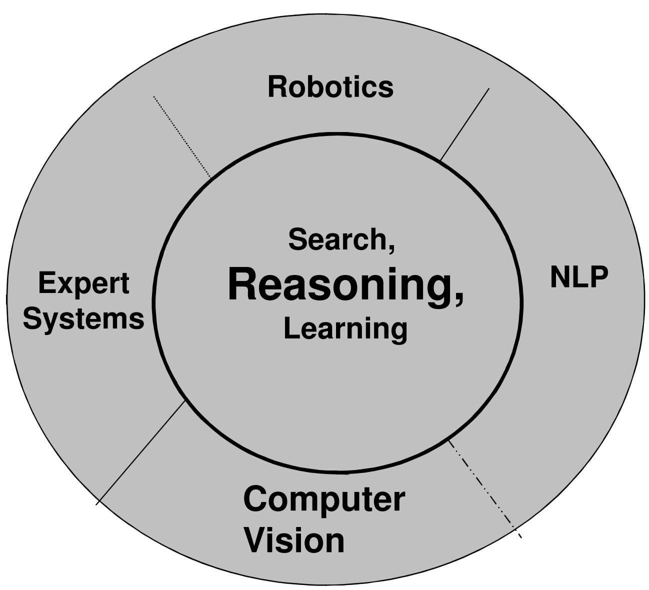


Question 4

- Indicate (mark the edge) the proper move of the maximizer by marking one of the root's outgoing arcs.

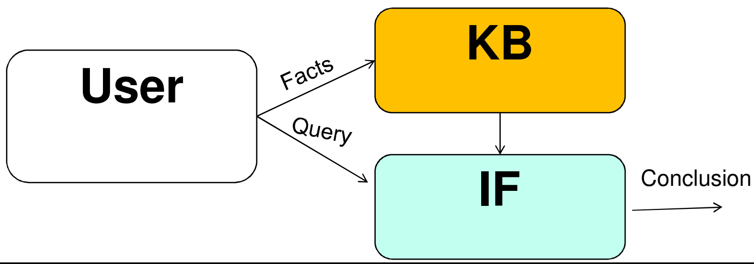


Reasoning



First-order logic Chapter 8-Russel
Representation and Reasoning

- In order to determine appropriate actions to take, an intelligent system needs to represent information about the world and draw conclusions based on general world knowledge and specific facts.
- Knowledge is represented by sentences in some language stored in a **knowledge base (KB).**
- A system draws conclusions from the KB to answer questions, take actions using **Inference Engine (IF)**.



Knowledge Representation

- 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
- E.g., the language of arithmetic
  - x+2 ≥ y is a sentence; x2+y > {} is not a sentence syntax
  - x+2 ≥ y is true in a world where x = 7, y = 1
  - x+2 ≥ y is false in a world where x = 0, y = 6

  } semantics

Inference

- Logical Inference (deduction) derives new sentences in the language from existing ones,.

  Socrates is a man.

  All men are mortal.

  Socrates is mortal.

- Proper inference should only derive sound conclusions

Logics

- 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 true of a sentence in a world

Examples of Types of Logics

Language	What exist	Degree of belief of an Agent
Propositional Logic	Facts	{o,1} T or F
First Order Logic	Facts, Objects, Relations	{o,1} T or F
Temporal Logic	Facts, Objects, Relations, Time	{o,1} T or F
Probability Theory	Facts	Chances of belief [0,1]
Fuzzy Logic	Degree of truth about Facts	Degree of belief [0,1]

Propositional calculus & First-order logic

- Propositional logic assumes world contains facts.
- First-order logic (like natural language) assumes the world contains
  - Objects: people, houses, numbers, …
  - Relations: red, round, prime,…
  - Functions: fatherof, friend, in,…
- Propositional calculus
  $A \wedge B \Rightarrow C$
- First-order predicate calculus
  $(\forall x)(\exists y)$ Mother(y,x)

Syntax of PC Chapter 7-Russel

- Connectives: ¬, ∧, ∨, ⇒
- Propositional symbols, e.g., P, Q, R, …
  - *True, False*
  - **Syntax of PC**
- sentence → atomic sentence | complex sentence
- atomic sentence → Propositional symbol, *True, False*
- Complex sentence → ¬sentence
  | (sentence ∧ sentence)
  | (sentence ∨ sentence)
  | (sentence ⇒ sentence)
  - **Rules of Inference:**
  - Ex: Modus ponens

Sentence in PC

A **sentence (also called a formula or well-formed formula or wff)** is defined as**:**

- A symbol (S, P, …etc)
- If S is a sentence, then ¬S is a sentence, where "¬" is the "not" logical operator
- If S and T are sentences, then (S v T), (S ^ T), (S => T), and (S <=> T) are sentences, where the four logical connectives correspond to "or," "and," "implies," and "if and only if," respectively

Example

P means "It is hot"
Q means "It is humid"
R means "It is raining“

Examples of PL sentences:
(P ^ Q) => R (here meaning "If it is hot and humid, then it is raining")
Q => P (here meaning "If it is humid, then it is hot")
¬ Q (here meaning "It is not humid.")

Semantics of PC

A	B	¬ A	A ∧ B	A ∨ B	A ⇒ B
True	*True*	*False*	*True*	*True*	*True*
True	*False*	*False*	*False*	*True*	*False*
False	*False*	*True*	*False*	*False*	*True*
False	*True*	*True*	*False*	*True*	*True*

Semantics of PC
Validity and Inference

• **interpretation of the sentence:** Given the truth values of all of the constituent symbols in a sentence, that sentence can be "evaluated" to determine its truth value (True or False).

• A **model** is an interpretation (i.e., an assignment of truth values to symbols) of a set of sentences such that each sentence is True. A model is just a formal mathematical structure for the world.

• A **valid sentence (also called a tautology)** is a sentence that is True under ***all interpretations. Hence, no matter what the world is actually like or what the semantics is, the sentence is True.***
For example "It's raining or it's not raining."
Remark: Validity can be checked by the truth table

Semantics of PC
Validity and Inference

• An **inconsistent sentence (also called unsatisfiable or a contradiction)** is a sentence that is False under ***all interpretations.***
For example, "It's raining and it's not raining."

• Sentence P **entails sentence Q, written P |= Q,** means that whenever P is True, so is Q. In other words, all models of P are also models of Q

Satisfiability

- A sentence is **satisfiable** if it is true under some interpretation (i.e. it has a model), otherwise the sentence is **unsatisfiable.**
- A sentence is **valid** if and only if its negation is unsatisfiable.
- •Therefore, algorithms for either validity or satisfiability
- checking are useful for logical inference.
- If there are *n propositional symbols in a sentence, then we* must check 2^n *rows for validity*
- **Satisfiability is** NP-complete, i.e. there is no polynomial-time algorithm to solve.
- Yet, many problems can be solved very quickly.

Rules of Inference

- A sequence of inference rule applications that leads to a desired conclusion is called a **logical proof.**
- A |- B , denotes that B can be derived by some inference procedure from the set of sentences A.
- Inference rules can be verified by the truth-table
- The truth table method of inference is **complete for PL**
- Then used to construct **sound proofs.**
- Finding a proof is simply a **search problem** with the inference rules as operators and the conclusion as the goal

Rules of Inference

- **Modus Ponens:** $\{\alpha \Rightarrow \beta, \alpha\} \vdash \beta$
- **And Elimination:** $\{\alpha \wedge \beta\} \vdash \alpha$; $\{\alpha \wedge \beta\} \vdash \beta$
- **Double negation Elimination:** $\{\neg\neg\alpha\} \vdash \alpha$
- **Implication Elimination** $\{\alpha \Rightarrow \beta\} \vdash \neg\alpha \vee \beta$
- **Unit resolution:** $\{\alpha \vee \beta, \neg\beta\} \vdash \alpha$
- **Resolution:** $\{\alpha \vee \beta, \neg\beta \vee \gamma\} \vdash \alpha \vee \gamma$

Famous logical equivalences

• $(a \vee b) \equiv (b \vee a)$	*commutatitvity*
• $(a \wedge b) \equiv (b \wedge a)$	*commutatitvity*
• $((a \wedge b) \wedge c) \equiv (a \wedge (b \wedge c))$	*associativity*
• $((a \vee b) \vee c) \equiv (a \vee (b \vee c))$	*associativity*
• $\neg(\neg(a)) \equiv a$	*double-negation elimination*
• $(a \Rightarrow b) \equiv (\neg(b) \Rightarrow \neg(a))$	*contraposition*
• $(a \Rightarrow b) \equiv (\neg(a) \vee b)$	*implication elimination*
• $\neg(a \wedge b) \equiv (\neg(a) \vee \neg(b))$	*De Morgan*
• $\neg(a \vee b) \equiv (\neg(a) \wedge \neg(b))$	*De Morgan*
• $(a \wedge (b \vee c)) \equiv ((a \wedge b) \vee (a \wedge c))$	*distributitivity*
• $(a \vee (b \wedge c)) \equiv ((a \vee b) \wedge (a \vee c))$	*distributitivity*

Pros and cons of propositional logic

- ✓ Propositional logic is declarative:

  pieces of syntax correspond to facts
- ✓ Propositional logic is compositional:

  meaning of A ^ B is derived from meaning of A and B
- ✓ 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)

Propositional logic is a weak language

- Hard to identify "individuals." Ex. Mary, 3
- Can't directly talk about properties of individuals or relations between individuals. Ex. "Bill is tall"
- Generalizations, patterns, regularities can't easily be represented. Ex. all triangles have 3 sides
- **First-Order Logic** (abbreviated FOL or FOPC) is expressive enough to concisely represent this kind of situation.
  - FOL adds relations, variables, and quantifiers, e.g.,
    - "Every elephant is gray": $\forall$ x (elephant(x) $\rightarrow$ gray(x))
    - "There is a white elephant": $\exists$ x (elephant(x) ^ white(x))