

COURSE TITLE: INNOVATION IN FINTECH

LECTURE: INNOVATIONS IN INVESTMENT
TECHNOLOGY: ARTIFICIAL INTELLIGENCE

Lecturer Radjabov Jamsher

AI DEFINITIONS

- ▶ The study of how to make programs/computers do things that people do better
- ▶ The study of how to make computers solve problems which require knowledge and intelligence
- ▶ The exciting new effort to make computers think ... machines with minds
- ▶ The automation of activities that we associate with human thinking (e.g., decision-making, learning...)
- ▶ The art of creating machines that perform functions that require intelligence when performed by people
- ▶ The study of mental faculties through the use of computational models
- ▶ A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes
- ▶ The branch of computer science that is concerned with the automation of intelligent behavior

Thinking
machines or
machine
intelligence

Studying
cognitive
faculties

Problem
Solving and
CS

SO WHAT IS AI?

- ▶ AI as a field of study
 - ▶ Computer Science
 - ▶ Cognitive Science
 - ▶ Psychology
 - ▶ Philosophy
 - ▶ Linguistics
 - ▶ Neuroscience
- ▶ AI is part science, part engineering
- ▶ AI often must study other domains in order to implement systems
 - ▶ e.g., medicine and medical practices for a medical diagnostic system, engineering and chemistry to monitor a chemical processing plant
- ▶ AI is a belief that the brain is a form of biological computer and that the mind is computational
- ▶ AI has had a concrete impact on society but unlike other areas of CS, the impact is often
 - ▶ felt only tangentially (that is, people are not aware that system X has AI)
 - ▶ felt years after the initial investment in the technology

WHAT IS INTELLIGENCE?

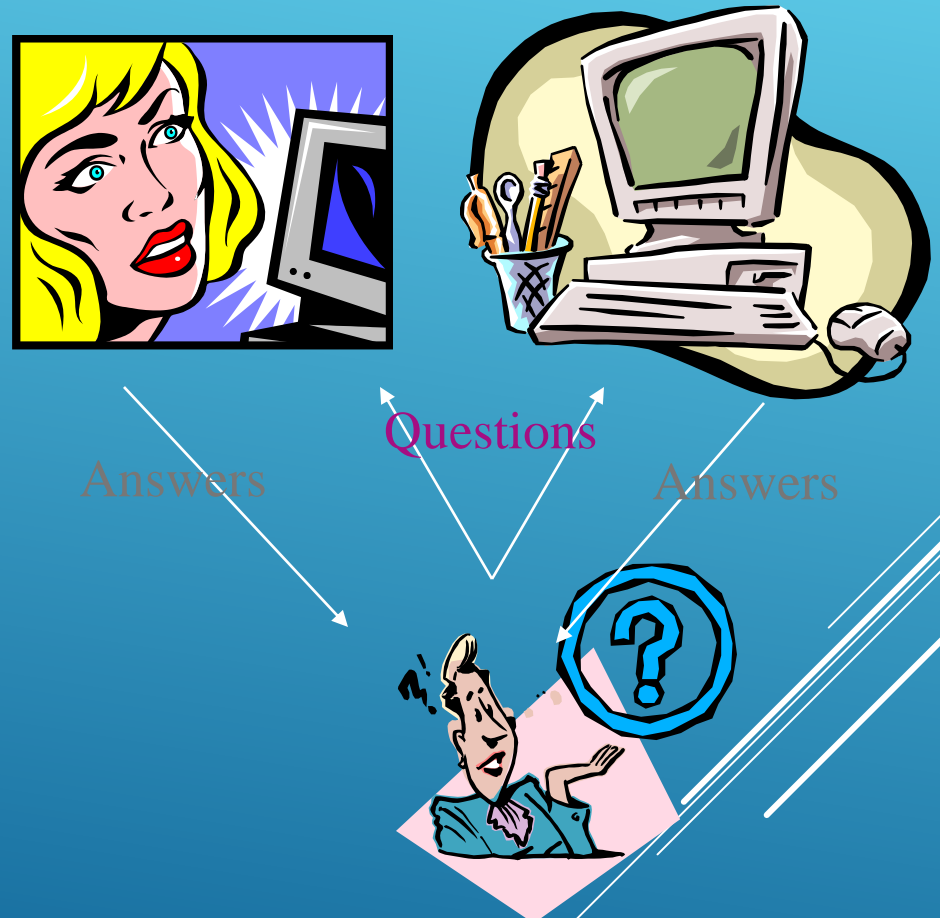
- ▶ Is there a “holistic” definition for intelligence?
- ▶ Here are some definitions:
 - ▶ *the ability to comprehend; to understand and profit from experience*
 - ▶ *a general mental capability that involves the ability to reason, plan, solve problems, think abstractly, comprehend ideas and language, and learn*
 - ▶ *is effectively perceiving, interpreting and responding to the environment*
- ▶ None of these tells us what intelligence is, so instead, maybe we can enumerate a list of elements that an intelligence must be able to perform:
 - ▶ *perceive, reason and infer, solve problems, learn and adapt, apply common sense, apply analogy, recall, apply intuition, reach emotional states, achieve self-awareness*
- ▶ Which of these are necessary for intelligence? Which are sufficient?
- ▶ Artificial Intelligence – should we define this in terms of human intelligence?
 - ▶ does AI have to really be intelligent?
 - ▶ what is the difference between being intelligent and demonstrating intelligent behavior?

PHYSICAL SYMBOL SYSTEM HYPOTHESIS

- ▶ A physical symbol system (PSS)
 - ▶ consists of symbols (patterns) which are combinable into expressions
 - ▶ there are processes which operate on these symbols to create new symbols and expressions
 - ▶ consider for instance English as a physical symbol system
 - ▶ symbols are the alphabet
 - ▶ expressions are words and sentences
 - ▶ the processes are the English grammar and parsers and dictionaries
- ▶ The PSS Hypothesis states that a PSS has the *necessary* and *sufficient* means for intelligent action
 - ▶ a computer is a PSS
 - ▶ if the PSS Hypothesis is true, then it should be possible to program a computer to produce intelligent actions
 - ▶ this is the (or a) goal of AI
 - ▶ if the PSS Hypothesis is not true, what are the consequences for AI?

THE TURING TEST

- ▶ 1950 – Alan Turing devised a test for intelligence called the Imitation Game
 - ▶ Ask questions of two entities, receive answers from both
 - ▶ If you can't tell which of the entities is human and which is a computer program, then you are fooled and we should therefore consider the computer to be intelligent



Which is the person?
Which is the computer?

SOLVING THE TURING TEST: ELIZA

- ▶ Created by Joseph Weizenbaum
- ▶ This system would act like a Rogerian psychoanalyst by holding a dialog with a person
 - ▶ the dialog would be consist of the doctor (Eliza) asking questions, the human responding, and the doctor using the response to ask another question
- ▶ Weizenbaum wrote the program so that it would generate an English response/question based on a group of patterns
 - ▶ If the user sentence matched a pattern, this pattern would be used to generate the next sentence/question
 - ▶ A more recent, and much superior version is called A.L.I.C.E, see <http://alicebot.blogspot.com/>
 - ▶ click on "chat with A.L.I.C.E."

- H: Men are all alike
- E: In what way?
- H: They're always bugging us about something or another.
- E: Can you be more specific?
- H: Well, my boyfriend made me come here.
- E: Your boyfriend made you come here?
- H: He says I'm depressed much of the time.
- E: I am sorry to hear you are depressed.
- H: It's true. I am unhappy.
- E: Do you think coming here will help you not to be unhappy?
- H: I need some help, that much seems certain.
- E: What would it mean to you if you got some help?
- H: Perhaps I could learn to get along with my mother.
- E: Tell me more about your family

EXAMPLE ELIZA DIALOG



(?* ?x) hello (?* ?y) →

(How do you do. Please state your problem.)

(?* ?x) I want (?* ?y) →

(What would it mean if you got ?y)

(Why do you want ?y)

(Suppose you got ?y soon)

(?* ?x) if (?* ?y) →

(Do you really think its likely that ?y)

(Do you wish that ?y)

(What do you think about ?y) (Really-- if ?y)

(?* ?x) no (?* ?y) → (Why not?)

(You are being a bit negative)

(Are you saying "NO" just to be negative?)

(?* ?x) I was (?* ?y) → (Were you really?)

(Perhaps I already knew you were ?y)

(Why do you tell me you were ?y now?)

(?* ?x) I feel (?* ?y) → (Do you often feel ?y ?)

(?* ?x) I felt (?* ?y) →

(What other feelings do you have?)

ELIZA RULES

Eliza also has a rule to swap 'I' and 'my' to 'you' and 'your'

ELIZA PATTERN SYNTAX

pat →	var	match any one expression to a variable
	constant	or to a constant (see below)
	segment-pat	match against a sequence
	single-pat	match against one expression
	(pat . pat)	match the first and the rest of a list
single-pat →	(?is var predicate)	test predicate on one expression
	(?or pat1 pat2 ...)	match on any of the patterns
	(?and pat1 pat2 ...)	match on every of the expressions
	(?not pat)	match if expression does not match
segment-pat →	((?* var) ...)	match on zero or more expressions
	((?+ var) ...)	match on one or more expressions
	((?? var) ...)	match zero or one expression
	((?if expr) ...)	test if expression is true
var →	?chars	variables of the form ?name
constant →	atom	symbols, numbers, chars

ALICE DEMO

Jerry: my name is jerry.

ALICE: Hey jerry.

Jerry: do you know Turing?

ALICE: A brilliant and tragic figure in the history of computer science.

Jerry: who else do you know?

ALICE: Hugh Loebner, Robby Garner, and Steven Spielberg

Jerry: which film do you like?

ALICE: My favorite movie is Teknolust. Have you seen it?

Jerry: no.

ALICE: I would recommend you check it out.

Jerry: what is it about?

ALICE: Science fiction.

Jerry: do you remember my name?

ALICE: Your name is jerry, seeker.

Notice unlike Eliza, ALICE has a memory, can handle references
(have you seen it? no. I would recommend it.)

ALICE VS. ELIZA

ALICE talking to **ELIZA**

ELIZA: I am ELIZA.

ALICE: I am very pleased to meet you ELIZA.

ELIZA: How can I help you?

ALICE: Just keep talking to me as long as you can.

ELIZA: Do you wish to as long as me?

ALICE: OK, I will as long as you.

ELIZA: Do you wish to will as long as me?

ALICE: OK, I will will as long as you.

ELIZA: Do you wish to will will as long as me?

ALICE: OK, I will will will as long as you.

ELIZA: Do you wish to will will will as long as me?

ALICE: OK, I will will will will as long as you.

ELIZA: Do you wish to will will will will as long as me?

ALICE: OK, I will will will will will as long as you.

ELIZA: Do you wish to will will will will will as long as me?

...

Eliza gets stuck on the phrase “I will” and then ALICE gets stuck on the same phrase

HOW USEFUL IS THE TURING TEST?

- ▶ With Eliza or Alice like rules, we can eventually solve the Turing Test – it just takes writing enough rules
- ▶ Does the system understand what it is responding to?
 - ▶ No, neither Eliza nor Alice understand the text, its just that Alice has better, more in depth and wider ranging rules
- ▶ However, we could build a representation that models some real-world domain and knowledge base
 - ▶ The system can fill in information from the conversation
 - ▶ this is sort of like a database, or an object with data attributes to be filled in
 - ▶ we can use a variety of AI representations like scripts, frames, semantic networks
 - ▶ Questions can be responded to by looking up the stored data
 - ▶ In this way, the system is responding, not based merely on “canned” knowledge, but on knowledge that it has “learned”
- ▶ Does this imply that the system *knows* what it is discussing?
 - ▶ What does it mean to *know* something?

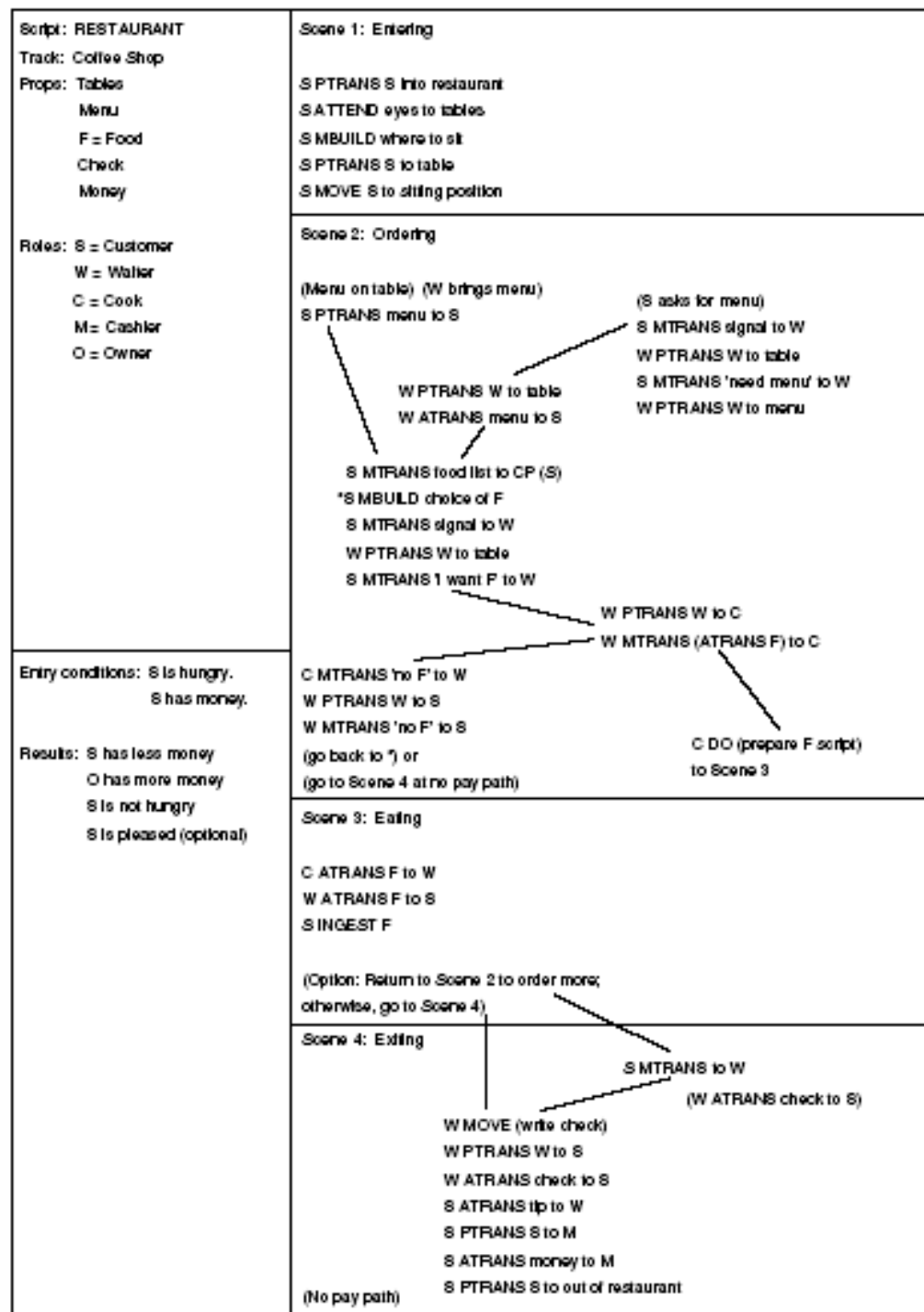
TABLE-LOOKUP VS. REASONING

- ▶ Consider two approaches to programming a Tic-Tac-Toe player
 - ▶ Solution 1: a pre-enumerated list of best moves given the board configuration
 - ▶ Solution 2: rules (or a heuristic function) that evaluate a board configuration, and using these to select the next best move
- ▶ Solution 1 is similar to how Eliza works
 - ▶ This is not practical for most types of problems
 - ▶ Consider solving the game of chess in this way, or trying to come up with all of the responses that a Turing Test program might face
- ▶ Solution 2 will reason out the best move
 - ▶ Given the board configuration, it will analyze each available move and determine which is the best
 - ▶ Such a player might even be able to “explain” why it chose the move it did
- ▶ We can (potentially) build a program that can pass the Turing Test using table-lookup even though it would be a large undertaking
- ▶ Could we build a program that can pass the Turing Test using reasoning?
 - ▶ Even if we can, does this necessarily mean that the system is intelligent?

SLOT FILLING

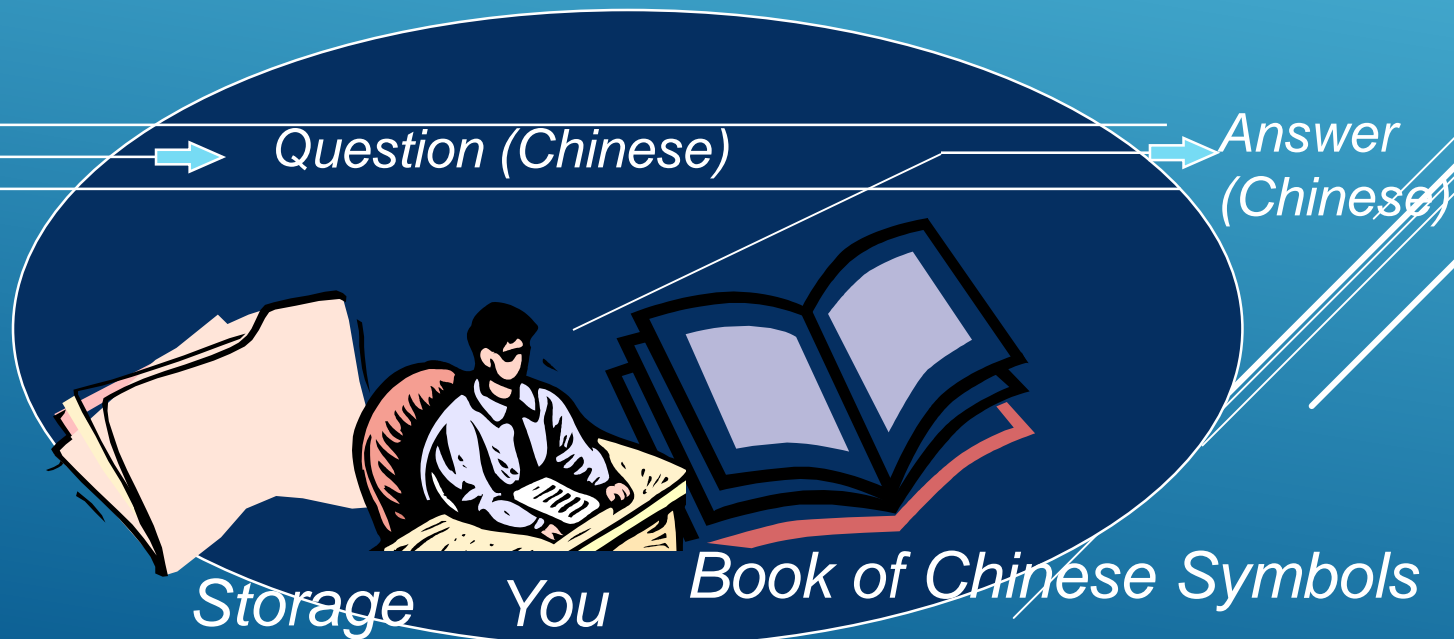
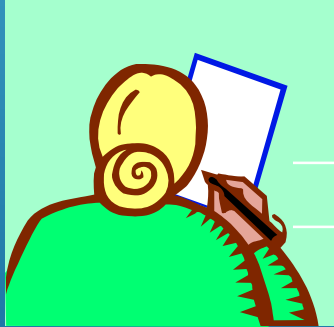
- ▶ Roger Schank created the Script representation
 - ▶ the script describes typical sequences of actions and actors for some real-world situation
 - ▶ a story (newspaper article) is parsed and slots are filled in
 - ▶ SAM (Script Applier Mechanism) uses the filled in script to answer questions
- ▶ The Script provides the knowledge needed to respond like a human and thus solve the Turing Test

Schank's Restaurant script



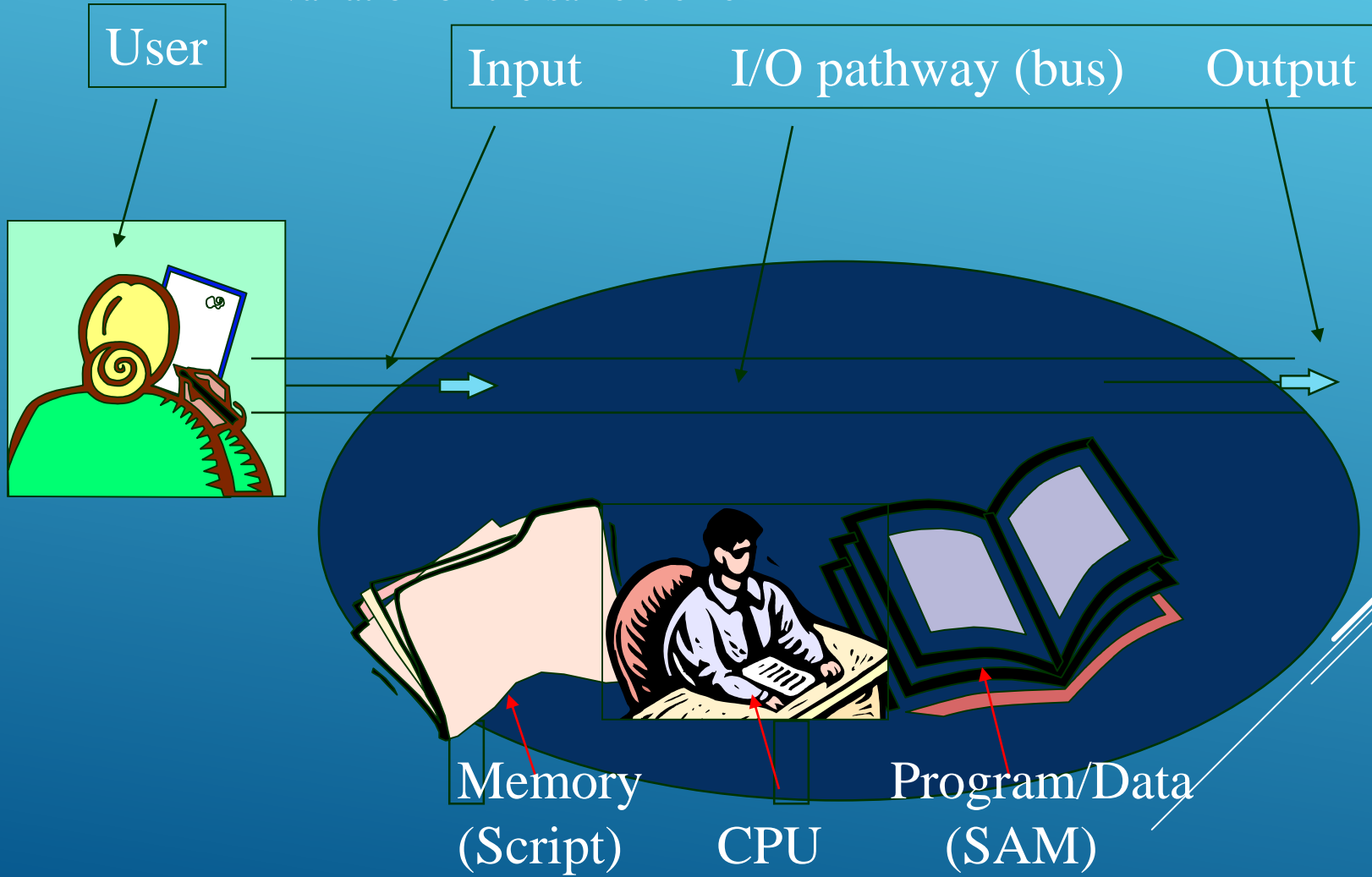
THE CHINESE ROOM PROBLEM

- ▶ From John Searle, Philosopher, in an attempt to demonstrate that computers cannot be intelligent
 - ▶ The room consists of you, a book, a storage area (optional), and a mechanism for moving information to and from the room to the outside
 - ▶ a Chinese speaking individual provides a question for you in writing
 - ▶ you are able to find a matching set of symbols in the book (and storage) and write a response, also in Chinese



CHINESE ROOM: AN ANALOGY FOR A COMPUTER

Note: Searle's original Chinese Room actually was based on a Script that was implemented in Chinese, our version is just a variation on the same theme



SEARLE'S QUESTION

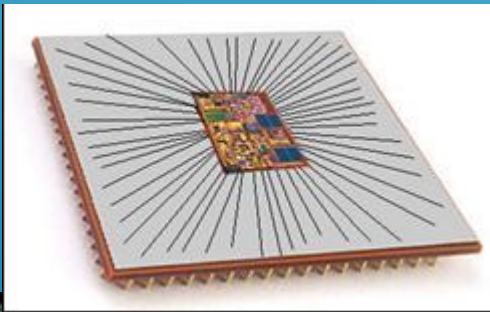
- ▶ You were able to solve the problem of communicating with the person/user and thus you/the room passes the Turing Test
- ▶ But did you understand the Chinese messages being communicated?
 - ▶ since you do not speak Chinese, you did not understand the symbols in the question, the answer, or the storage
 - ▶ can we say that you actually *used* any intelligence?
- ▶ By analogy, since you did not understand the symbols that you interacted with, neither does the computer understand the symbols that it interacts with (input, output, program code, data)
- ▶ Searle concludes that the computer is not intelligent, it has no “semantics,” but instead is merely a symbol manipulating device
 - ▶ the computer operates solely on syntax, not semantics
- ▶ He defines two categories of AI:
 - ▶ strong AI – the pursuit of machine intelligence
 - ▶ weak AI – the pursuit of machines solving problems in an intelligent way

BUT COMPUTERS SOLVE PROBLEMS

- ▶ We can clearly see that computers solve problems in a seemingly intelligent way
 - ▶ Where is the intelligence coming from?
- ▶ There are numerous responses to Searle's argument
 - ▶ The System's Response:
 - ▶ the hardware by itself is not intelligent, but a combination of the hardware, software and storage is intelligent
 - ▶ in a similar vein, we might say that a human brain that has had no opportunity to learn anything cannot be intelligent, it is just the hardware
 - ▶ The Robot Response:
 - ▶ a computer is void of senses and therefore symbols are meaningless to it, but a robot with sensors can tie its symbols to its senses and thus understand symbols
 - ▶ The Brain Simulator Response:
 - ▶ if we program a computer to mimic the brain (e.g., with a neural network) then the computer will have the same ability to understand as a human brain

BRAIN VS. COMPUTER

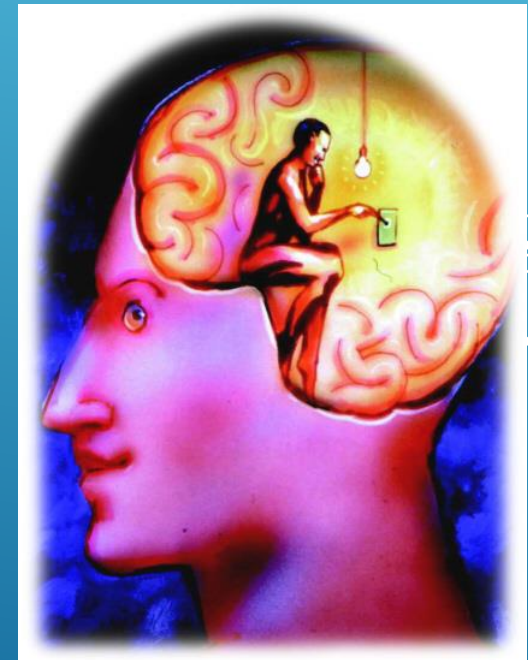
- ▶ In AI, we compare the brain (or the mind) and the computer
 - ▶ Our hope: the brain is a *form* of computer
 - ▶ Our goal: we can *create* computer intelligence through programming just as people become intelligent by learning



But we see that the computer is not like the brain

The computer performs tasks without understanding what its doing

Does the brain understand what its doing when it solves problems?



SYMBOL GROUNDING

- ▶ One problem with the computer is that it works strictly syntactically
 - ▶ Op code: 10011101 → translates into a set of microcode instructions such as: move $IR_{16..31}$ to MAR, signal memory read, move MBR to AC
 - ▶ There is no understanding
 - ▶ $x = y + z$; is meaningless to the computer
 - ▶ the computer doesn't understand addition, it just knows that a certain op code means to move values to the adder and move the result elsewhere
 - ▶ do you know what addition means?
 - ▶ if so, how do you proscribe meaning to +
 - ▶ how is this symbol grounded in your brain?
 - ▶ can computers similarly achieve this?
 - ▶ Recall Schank's Restaurant script
 - ▶ does the computer know what the symbols "waiter" or "PTRANS" represent? or does it merely have code that tells the computer what to do when it comes across certain words in the story, or how to respond to a given question?

TWO AI ASSUMPTIONS

- ▶ We can understand and model cognition without understanding the underlying mechanism
 - ▶ That is, it is the model of cognition that is important not the physical mechanism that implements it
 - ▶ If this is true, then we should be able to create cognition (mind) out of a computer or a brain or even other entities that can compute such as a mechanical device
 - ▶ This is the assumption made by symbolic AI researchers
- ▶ Cognition will emerge from the proper mechanism
 - ▶ That is, the right device, fed with the right inputs, can learn and perform the problem solving that we, as observers, call intelligence
 - ▶ Cognition will arise as the result (or side effect) of the hardware
 - ▶ This is the assumption made by connectionist AI researchers
- ▶ Notice that while the two assumptions differ, neither is necessarily mutually exclusive and both support the idea that cognition is *computational*

PROBLEMS WITH SYMBOLIC AI APPROACHES

▶ Scalability


- ▶ It can take dozens or more man-years to create a useful systems
- ▶ It is often the case that systems perform well up to a certain threshold of knowledge (approx. 10,000 rules), after which performance (accuracy and efficiency) degrade

▶ Brittleness


- ▶ Most symbolic AI systems are programmed to solve a specific problem, move away from that domain area and the system's accuracy drops rapidly rather than achieving a graceful degradation
 - ▶ this is often attributed to lack of common sense, but in truth, it is a lack of any knowledge outside of the domain area
- ▶ No or little capacity to learn, so performance (accuracy) is static

▶ Lack of real-time performance

PROBLEMS WITH CONNECTIONIST AI APPROACHES

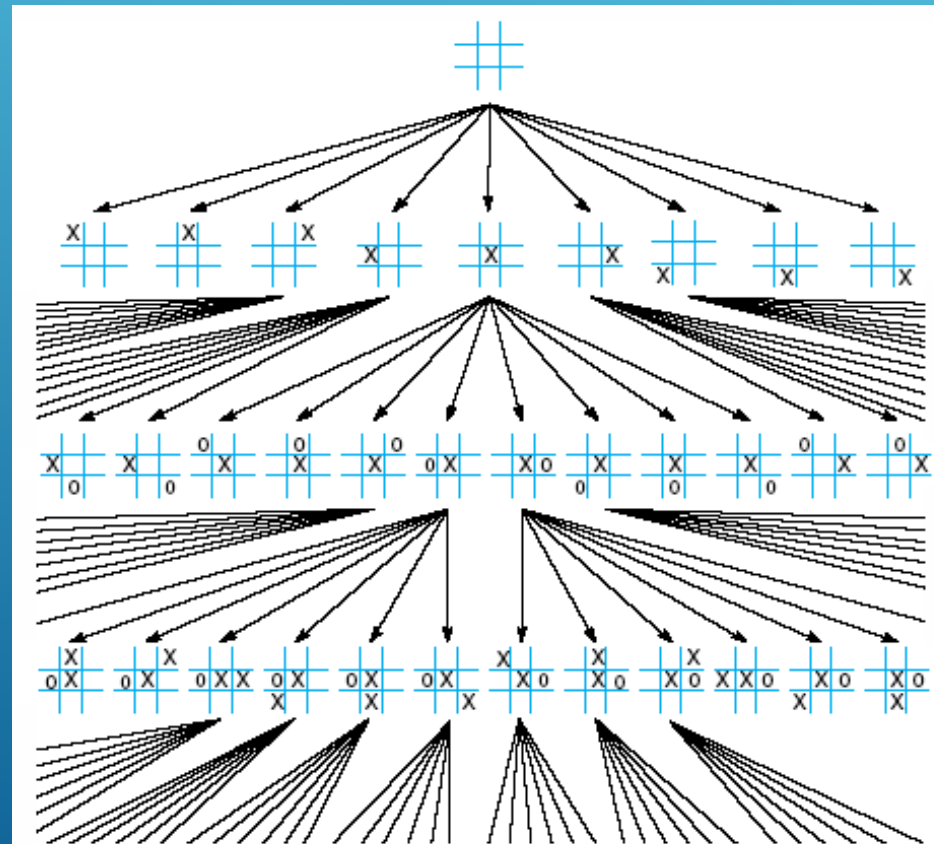
- ▶ No “memory” or sense of temporality
 - ▶ The first problem can be solved to some extent
 - ▶ The second problem arises because of a fixed sized input but leads to poor performance in areas like speech recognition
 - ▶ Learning is problematic
 - ▶ Learning times can greatly vary
 - ▶ Overtraining leads to a system that only performs well on the training set and undertraining leads to a system that has not generalized
 - ▶ No explicit knowledge-base
 - ▶ So there is no way to tell what a system truly knows or how it knows something
 - ▶ No capacity to explain its output
 - ▶ Explanation is often useful in an AI system so that the user can trust the system’s answer
- 

SO WHAT DOES AI DO?


- ▶ Most AI research has fallen into one of two categories
 - ▶ Select a specific problem to solve
 - ▶ study the problem (perhaps how humans solve it)
 - ▶ come up with the proper representation for any knowledge needed to solve the problem
 - ▶ acquire and codify that knowledge
 - ▶ build a problem solving system
 - ▶ Select a category of problem or cognitive activity (e.g., learning, natural language understanding)
 - ▶ theorize a way to solve the given problem
 - ▶ build systems based on the model behind your theory as experiments
 - ▶ modify as needed
 - ▶ Both approaches require
 - ▶ one or more representational forms for the knowledge
 - ▶ some way to select proper knowledge, that is, search
- 
- A series of white diagonal lines of varying thicknesses and lengths, located in the bottom right corner of the slide, extending from the right edge towards the center.

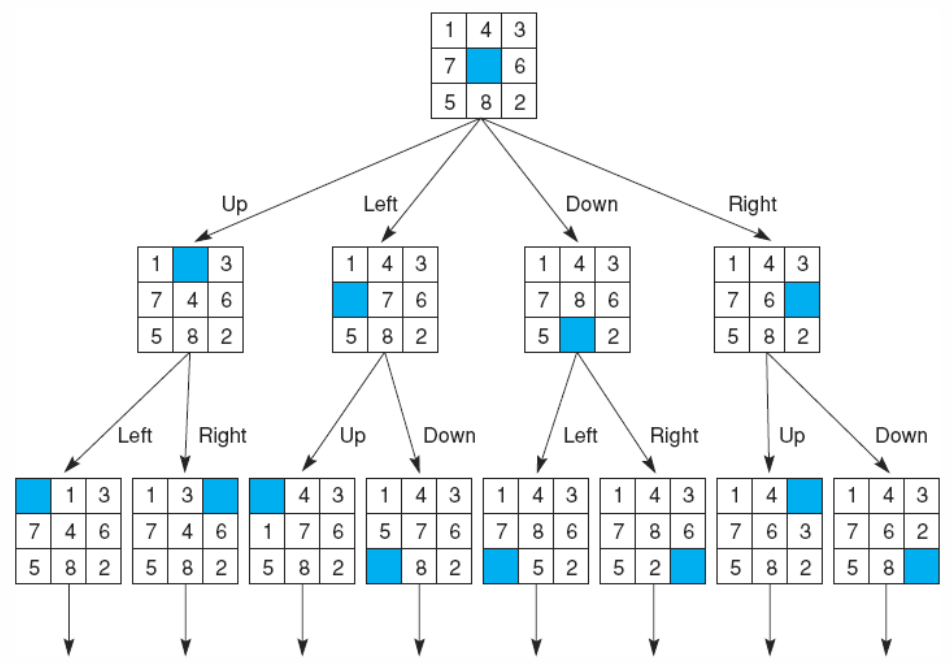
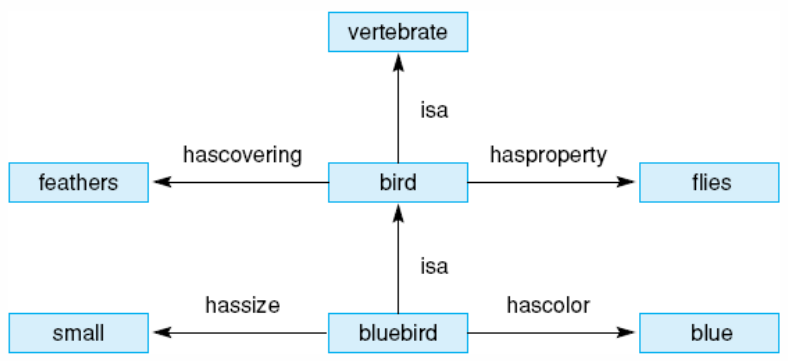
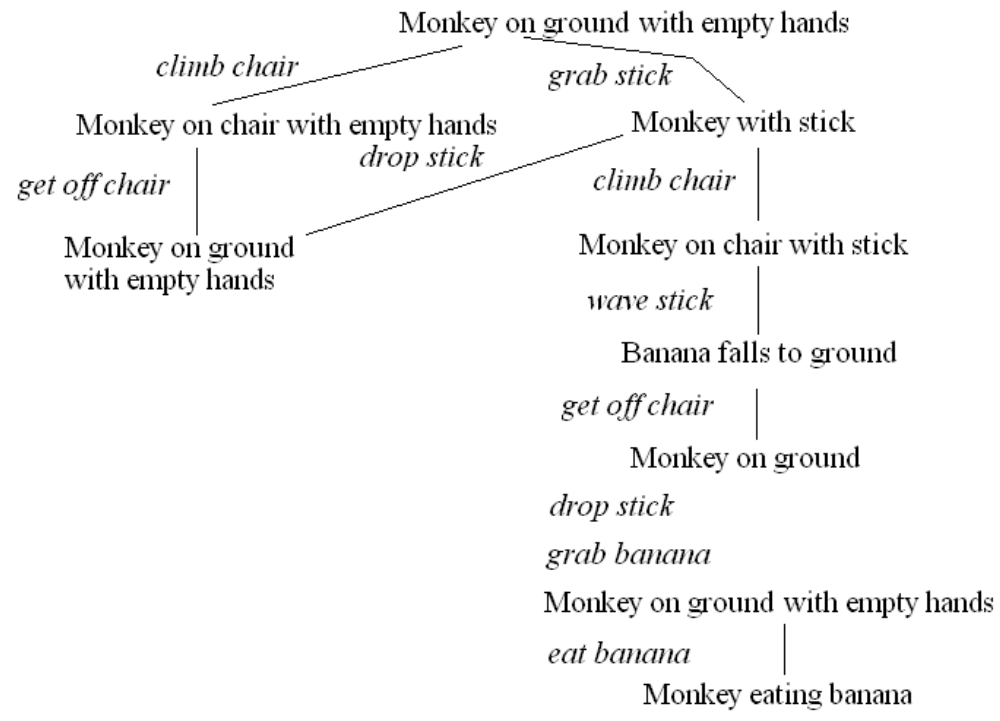
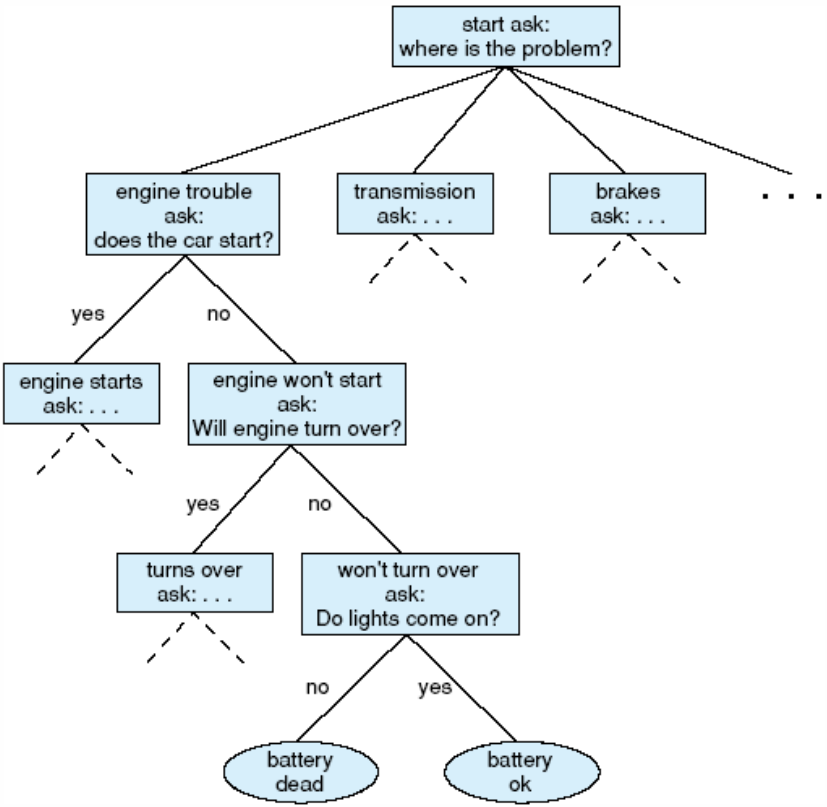
WHAT IS SEARCH?

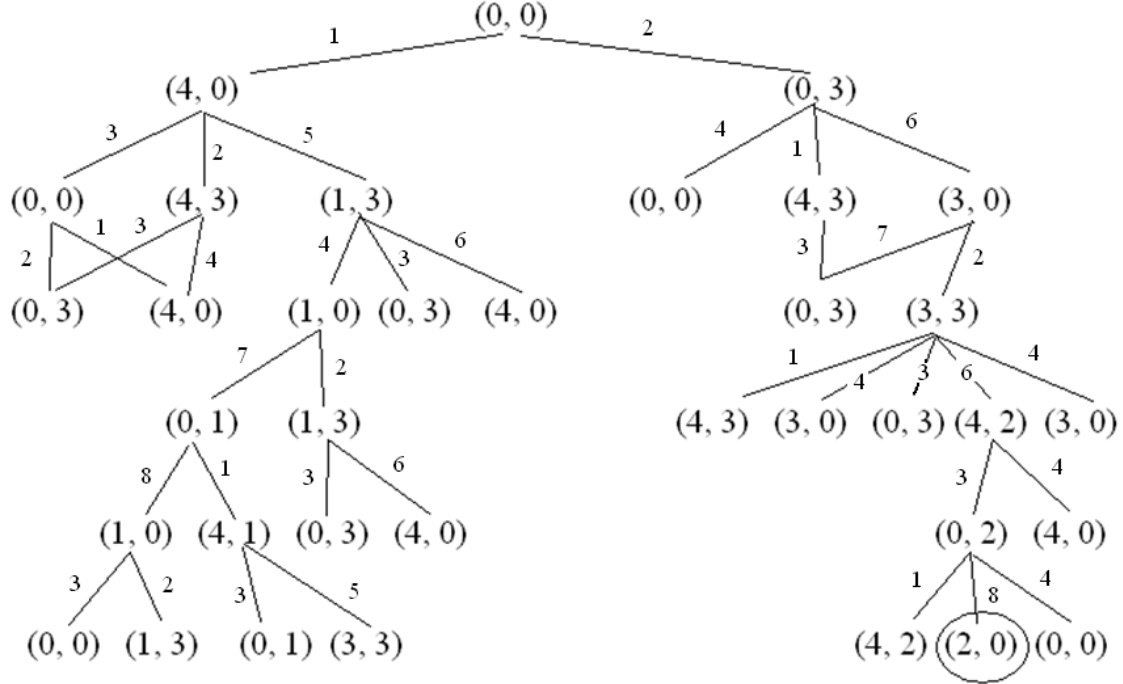
- ▶ We define the state of the problem being solved as the values of the active variables
 - ▶ this will include any partial solutions, previous conclusions, user answers to questions, etc
- ▶ while humans are often able to make intuitive leaps, or recall solutions with little thought, the computer must search through various combinations to find a solution
- ▶ To the right is a search space for a tic-tac-toe game



SEARCH SPACES AND TYPES OF SEARCH

- ▶ The search space consists of all possible states of the problem as it is being solved
 - ▶ A search space is often viewed as a tree and can very well consist of an exponential number of nodes making the search process intractable
 - ▶ Search spaces might be pre-enumerated or generated during the search process
 - ▶ Some search algorithms may search the entire space until a solution is found, others will only search parts of the space, possibly selecting where to search through a heuristic
 - ▶ Search spaces include
 - ▶ Game trees like the tic-tac-toe game
 - ▶ Decision trees (see next slides)
 - ▶ Combinations of rules to select in a production system
 - ▶ Networks of various forms (see next slides)
 - ▶ Other types of spaces
- 





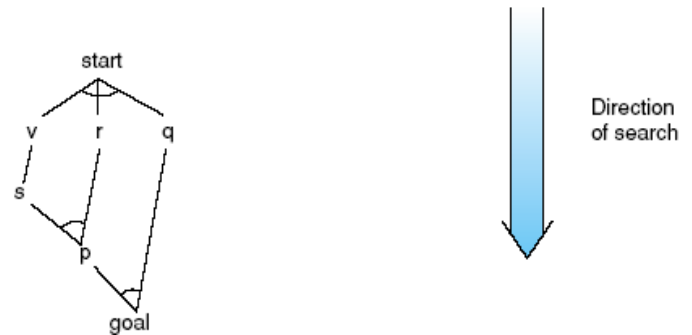
Production set:

1. $p \wedge q \rightarrow \text{goal}$
2. $r \wedge s \rightarrow p$
3. $w \wedge r \rightarrow q$
4. $t \wedge u \rightarrow q$
5. $v \rightarrow s$
6. $\text{start} \rightarrow v \wedge r \wedge q$

Trace of execution:

Iteration #	Working memory	Conflict set	Rule fired
0	start	6	6
1	start, v, r, q	6, 5	5
2	start, v, r, q, s	6, 5, 2	2
3	start, v, r, q, s, p	6, 5, 2, 1	1
4	start, v, r, q, s, p, goal	6, 5, 2, 1	halt

Space searched by execution:



SEARCH ALGORITHMS AND REPRESENTATIONS

- ▶ Breadth-first
- ▶ Depth-first
- ▶ Best-first (Heuristic Search)
- ▶ A*
- ▶ Hill Climbing
- ▶ Limiting the number of Plies
- ▶ Minimax
- ▶ Alpha-Beta Pruning
- ▶ Adding Constraints
- ▶ Genetic Algorithms
- ▶ Forward vs Backward Chaining
- ▶ We will study various forms of representation and uncertainty handling in the next class period
- ▶ Knowledge needs to be represented
 - ▶ Production systems of some form are very common
 - ▶ If-then rules
 - ▶ Predicate calculus rules
 - ▶ Operators
 - ▶ Other general forms include semantic networks, frames, scripts
 - ▶ Knowledge groups
 - ▶ Models, cases
 - ▶ Agents
 - ▶ Ontologies


A BRIEF HISTORY OF AI: 1950S

- ▶ Computers were thought of as an electronic brains
- ▶ Term “Artificial Intelligence” coined by John McCarthy
 - ▶ John McCarthy also created Lisp in the late 1950s
- ▶ Alan Turing defines intelligence as passing the Imitation Game (Turing Test)
- ▶ AI research largely revolves around toy domains
 - ▶ Computers of the era didn’t have enough power or memory to solve useful problems
 - ▶ Problems being researched include
 - ▶ games (e.g., checkers)
 - ▶ primitive machine translation
 - ▶ blocks world (planning and natural language understanding within the toy domain)
 - ▶ early neural networks researched: the perceptron
 - ▶ automated theorem proving and mathematics problem solving

THE 1960S

- ▶ AI attempts to move beyond toy domains
- ▶ Syntactic knowledge alone does not work, domain knowledge required
 - ▶ Early machine translation could translate English to Russian (“the spirit is willing but the flesh is weak” becomes “the vodka is good but the meat is spoiled”)
- ▶ Earliest expert system created: Dendral
- ▶ Perceptron research comes to a grinding halt when it is proved that a perceptron cannot learn the XOR operator
- ▶ US sponsored research into AI targets specific areas – not including machine translation
- ▶ Weizenbaum creates Eliza to demonstrate the futility of AI

1970S

- ▶ AI researchers address real-world problems and solutions through expert (knowledge-based) systems
 - ▶ Medical diagnosis
 - ▶ Speech recognition
 - ▶ Planning
 - ▶ Design
 - ▶ Uncertainty handling implemented
 - ▶ Fuzzy logic
 - ▶ Certainty factors
 - ▶ Bayesian probabilities
 - ▶ AI begins to get noticed due to these successes
 - ▶ AI research increased
 - ▶ AI labs sprouting up everywhere
 - ▶ AI shells (tools) created
 - ▶ AI machines available for Lisp programming
 - ▶ Criticism: AI systems are too brittle, AI systems take too much time and effort to create, AI systems do not learn
- 


1980S: AI WINTER

- ▶ Funding dries up leading to the AI Winter
 - ▶ Too many expectations were not met
 - ▶ Expert systems took too long to develop, too much money to invest, the results did not pay off
- ▶ Neural Networks to the rescue!
 - ▶ Expert systems took programming, and took dozens of man-years of efforts to develop, but if we could get the computer to learn how to solve the problem...
 - ▶ Multi-layered back-propagation networks got around the problems of perceptrons
 - ▶ Neural network research heavily funded because it promised to solve the problems that symbolic AI could not
- ▶ By 1990, funding for neural network research was slowly disappearing as well
 - ▶ Neural networks had their own problems and largely could not solve a majority of the AI problems being investigated
 - ▶ Panic! How can AI continue without funding?

1990S: ALIFE

- ▶ The dumbest smart thing you can do is staying alive
 - ▶ We start over – lets not create intelligence, lets just create “life” and slowly build towards intelligence
 - ▶ Alife is the lower bound of AI
 - ▶ Alife includes
 - ▶ evolutionary learning techniques (genetic algorithms)
 - ▶ artificial neural networks for additional forms of learning
 - ▶ perception and motor control
 - ▶ adaptive systems
 - ▶ modeling the environment
- ▶ Let's disguise AI as something new, maybe we'll get some funding that way!
 - ▶ Problems: genetic algorithms are useful in solving some optimization problems and some search-based problems, but not very useful for expert problems
 - ▶ perceptual problems are among the most difficult being solved, very slow progress

TODAY: THE NEW (OLD) AI

- ▶ Look around, who is doing AI research?
 - ▶ By their own admission, AI researchers are not doing “AI”, they are doing
 - ▶ Intelligent agents, multi-agent systems/collaboration
 - ▶ Ontologies
 - ▶ Machine learning and data mining
 - ▶ Adaptive and perceptual systems
 - ▶ Robotics, path planning
 - ▶ Search engines, filtering, recommendation systems
 - ▶ Areas of current research interest:
 - ▶ NLU/Information Retrieval, Speech Recognition
 - ▶ Planning/Design, Diagnosis/Interpretation
 - ▶ Sensor Interpretation, Perception, Visual Understanding
 - ▶ Robotics
 - ▶ Approaches
 - ▶ Knowledge-based
 - ▶ Ontologies
 - ▶ Probabilistic (HMM, Bayesian Nets)
 - ▶ Neural Networks, Fuzzy Logic, Genetic Algorithms
- 

Reference

1. The Innovation Ultimatum: Six strategic technologies that will reshape every business in the 2020s By Steve Brown.
2. Bank 4.0: Banking Everywhere, Never at a Bank By Brett King.
3. The World of Digital Payments: Practical Course By Pavlo Sidelov.
4. The PayTech Book Edited by a team of Susanne Chishti.
5. The Future Is Faster Than You Think: How Converging Technologies Are Disrupting Business, Industries, and Our Lives By Dr. Peter H. Diamandis.
6. Advances in Financial Machine Learning By Marcos Lopez de Prado
7. Financial Services Revolution: How Blockchain is Transforming Money, Markets, and Banking By Alex Tapscott.
8. The STO Financial Revolution: How Security Tokens Change Businesses Forever By Alex Nascimento.
9. FinTech Founders: Inspiring Tales from the Entrepreneurs that are Changing Finance By Agustín Rubini
10. The Innovation Ultimatum: Six strategic technologies that will reshape every business in the 2020s By Steve Brown