saylor.org ACADEMY

Saylor Academy awards

DAN RIMNICEANU

this certificate for the prescribed program of study for the

COMPUTER SCIENCE CURRICULUM

Issue Date: 30 mai 2018

Certificate ID: 11589059

Sean Connor
Director of Student Affairs
Saylor Academy

saylor.org ACADEMY

Saylor Academy awards

Dan Rimniceanu

this certificate of achievement for

CS408: Advanced Artificial Intelligence

23 iunie 2018

Issue Date

11658368

Certificate ID

CS408: Advanced Artificial Intelligence

Course Introduction
This course will present advanced topics in Artificial Intelligence (AI). We will begin by defining the term "software agent" and discussing how software agents differ from programs in general. We will then take a look at those problems in the field of AI that tend to receive the most attention. Different researchers approach these problems differently. In this course, we will focus on how to build and search graph data structures needed to create software agents, an approach that you will find useful for solving many problems in AI. We will also learn to "break down" larger problems into a number of more specific, manageable sub-problems.

In the latter portion of this course, we will review the study of logic and conceptualize the differences between propositional logic, first-order logic, fuzzy logic, and default logic. After learning about statistical tools commonly used in AI and about the basic symbol system used to represent knowledge, we will focus on artificial neural network and machine learning, which are essential components of computational and statistical methods, and theoretical computer science. The course will then conclude with a study of the Turing machine and a discussion of the questionable claims that human thinking is a symbol manipulation.

Unit 1: Intelligent Agents and Problems of AI
AI is often seen through the autonomous, rational intelligent agents paradigm, which we will emphasize in this unit. This unit will begin by discussing what software agents are and how agents differ from programs in general. The unit will then provide a natural taxonomy of autonomous agents and discusses possibilities for further classification before presenting those problems in AI that seem to received the most attention. The problem of creating intelligence is then broken down into a number of specific sub-problems, which consist of particular traits that should be found in an intelligent system. Note that different researchers approach the problems of AI from different perspectives, depending on their respective training, fields of expertise, and favored tools.

1.1: Is It an Agent, or Just a Program?

- The University of Memphis: Stan Franklin and Art Graesser's "Is It an Agent, or Just a Program?"URL

  Read this page to learn about the advent of software agents. Memorize the definitions of the AIMA, Maes, KidSim, Hayes-Roth, IBM, SodaBot, Foner, and Brustoloni Agents. Make sure you know how to define "agency" and work to memorize Franklin's definition of an agent. Read through the examples of the different taxonomies and classifications of agents.

- **1.1.1: John Lloyd on Intelligent Agents**

- [John Lloyd's"Intelligent Agents"URL]

  Watch the first part of this three-part video series by John Lloyd. As he lectures you may wish to work through the slides included on the page. Throughout the lecture, Professor Lloyd talks about AIMA agents and presents some pertinent examples. Compare his thoughts with yours and Franklin's from the previous sections.

- **1.1.2: Stan Franklin - A Cognitive Theory of Everything**
  - Stan Franklin's "A Cognitive Theory of Everything"URL

    Watch this video, which explains how intelligent agents fit into "the big picture." Ask yourself whether Franklin's thoughts make sense to you.

1.2: Overview of AI General Problems

- Wikipedia: "Artificial Intelligence"URL

  Read this entry on AI. After you read, you should know the meaning of terms such as knowledge representation, planning, learning, natural language processing, motion and manipulation, perception, social intelligence, creativity, and general intelligence.

- **1.2.1: Knowledge Representation**
  - Maurice Pagnucco's "Knowledge Representation and Reasoning"URL

    Watch the first part of this three-part video series by Maurice Pagnucco. After you watch, you should be able to define the terms knowledge, representation, and reasoning; realize the advantages of this approach; and define the forms of knowledge representation.

- **1.2.2: Planning**
  - Jussi Rintanen's "Planning"URL

    Watch this lecture. You may wish to work through the slides provided on the right-hand side of the screen as Rintanen lectures. After viewing the lecture, you should understand why planning can be difficult and be able to define the term "transition systems."

- **1.2.3: Learning**
  - Olivier Bousquet's "Introduction to Learning Theory"URL

    Watch this lecture, working through the provided on the right-hand side of the screen as you listen. After you watch, you should have a general understanding of "learning theory," be able to differentiate between deduction and induction, and describe, in general terms, the concept of probability and Bayes' rule.

1.3: Approaches to AI

- Wikipedia: "Artificial Inelligence Approaches"URL

  Review this entry on the different paradigms that guide AI research and make sure you know the differences between them.

- **1.3.1: Systems with General Intelligence**
  - Michael Thielscher's "Systems with General Intelligence"URL

    Watch this lecture about general problems in AI, working through slides provided on the right-hand side of the screen. After you watch, you should be familiar with the chess-as-an-intelligent-system example, understand what general game playing is about, and identify the major questions with which general AI is concerned. Do not let yourself get bogged down by the details; work for a general understanding of AI.

1.4: Agents in Code

- National Taiwan Normal University:Tsung-Che Chiang's "Vacuum Cleaner World"URL

  Complete this activity.

Unit 2: Solving Problems by Searching

This unit will teach you how to build and search data structures needed to create software agents. We will focus on graph structures and a few classical graph search algorithms because their understanding is important for solving many problems that arise in AI. Graphs enable logical description of the problems. A graph search, then, represents the search for the solutions. We will begin this unit with some basic graph theory definitions and then learn how to solve some problems with a graph. The last section of this unit has a video that will expand the understanding of the graph structures.

2.1: Graphs

- **2.1.1: Graph Definition**
  - Cameron McLeman's "Graph"URL

    Study the definition of a graph from this section and draw some examples of your own.

- **2.1.2: Binary Tree**
  - Thomas Niemann's "Binary Tree"URL

Make sure you know how a binary tree differs from a regular tree after the reading this section.

- **2.1.3: Example Problem: Minimum Spanning Tree**
  - Cameron McLeman's "Minimum Spanning Tree"URL

    Read about minimum spanning trees and try to figure out how Prim's algorithm works; the solution can be found here. Before you check the solution, try to solve problem yourself. After you have solved the problem (or if you have spent a couple of hours working on it, and are stumped!), study the solution.

2.2: Tree Search Algorithms

- **2.2.1: Binary Search Trees**
  - Thomas Niemann's "Binary Search Tree"URL

    Read the article to learn how to build and search binary trees.

- **2.2.2: Red-Black Trees**
  - Thomas Niemann's "Red-Black Trees"URL

    Read this article. After you read, you should know how a binary tree differs from a red-black tree and understand the basics of building and searching red-black trees.

- **2.2.3: Skip List**
  - Thomas Niemann's "Skip List"URL

    Read this article to learn how to build and search a skip list.

2.3: Common Search Techniques with Graphs

- **2.3.1: Depth-first Search**
  - Wikipedia: "Depth-First Search"URL

    Read this article to learn how depth-first search works. Study the included example.

- **2.3.2: Breadth-First Search**
  - Wikipedia: "Breadth-First Search"URL

    Read this article and make sure you know the differences between depth-first and breadth-first search algorithms.

- **2.3.3: Dijkstra's Algorithm**
  - Wikipedia: "Dijkstra's Algorithm"URL

Read this article to learn how Dijkstra's algorithm works. Work through the example in the article.

2.4: Search Algorithms in General

- Wikipedia: "Search Algorithm"URL

  Read this article for an overview of search techniques.

2.5: Basic Notions in Graph Theory

- Zoubin Ghahramani's "Graphical Models: Parts 1-3"URL

  Watch this three-part lecture on graph theory to develop a better understanding of how to use graphs in AI. After viewing the first part, you should know about directed, undirected, and factor graphs, conditional independence, d-separation, and plate notation. The second part will teach you about inference in graphical models, key ideas in belief propagation, and the junction tree algorithm. Watch the third part for fun, trying to follow along as much as possible.

2.6: Graph Examples in Code

- *Artificial Intelligence: A Modern Approach*: "Route Finding Agent"URL

  Create a route-finding agent given the environment in the form of a graph. One possible solution can be under the "Route Finding Agent" section. Study the solution code after you have already solved the problem, or if you have spent a substantial amount of time and are stuck (this problem could take you up to 12 hours to solve!).

Unit 3: Logical Agents and Knowledge Representation

Intelligent agents are supposed to make rational decisions, which are not just logically reasoned, but optimal as well, given the available information. Accordingly, in this unit, we will review the study of logic and conceptualize the differences between propositional logic, first-order logic, fuzzy logic, and default logic. This unit will also present an overview of common statistical tools used in AI. In the last part of this unit, we will try to clarify our definition of knowledge representation and will discuss its roles based on research conducted at MIT.

3.1: Logic Programming

- Wikipedia: "Logic Programming"URL

  Read this article on logic programming. Make sure you understand the differences between abductive logic, metalogic, constraint logic, concurrent logic, and inductive logic, higher-order logic, and linear logic programming.

- Alwen Tiu's "Introduction to Logic: Parts 1-3"URL

  Watch the first lecture on logic and compare it to the reading above. In this lecture, you will learn about the syntax and semantics of propositional logic, boolean functions, satisfiability, and binary decision trees. You will need to know the difference between conjunctive and disjunctive normal forms. Then, watch the second lecture to learn about first-order logic. Finally, watch the third lecture, which presents modal logic. Make sure you know the differences between propositional, first-order, and modal logic.

3.2: Probabilistic Methods for Uncertain Reasoning

- **3.2.1: Bayesian Network**
  - Paulo C.G. Costa and Kathryn B. Laskey's "Bayesian Networks"URL

    Read this article on Bayesian networks. Focus on the definitions it provides and work through the example provided.

  - Christopher Bishop's "Introduction to Bayesian Inference"URL

    Watch this lecture, which discusses Bayesian Inference. You may wish to work through the slides provided on the right-hand side of the screen as Bishop lectures. Focus on learning the rules of probability and understanding the terms Bayes' theorem, Bayesian inference, probabilistic graphical models. Make sure you know how factor graphs are used.

- **3.2.2: Hidden Markov Model**
  - Wikipedia: "Hidden Markov Model"URL

    Read this article, which discusses the hidden Markov model.

- **3.2.3: Other Methods for Uncertain Reasoning**
  - **3.2.3.1: Kalman Filter**
    - Wikipedia: "Kalman Filter"URL

      Read this article on the Kalman Filter.

  - **3.2.3.2: Decision Theory**
    - Wikipedia: "Decision Theory"URL

      Read this article, making sure you understand the normative and descriptive decision theory and what kinds of decisions need a theory.

3.3: Knowledge Representation and Reasoning

- Maurice Pagnucco's "Knowledge Representation and Reasoning"URL

  Watch this lecture. Focus on learning how to represent what we know and how to use representation to make inferences about that knowledge. Work carefully through the examples included in the lecture.

- Massachusetts Institute of Technology: Randall Davis, Howard Shrobe, and Peter Szolovit's "What Is a Knowledge Representation?"URL

  Read this article, which presents different views on knowledge representation. Contrast these views with your own.

3.4: Coding Drills

- Marty Hall's "N-Queens Problem Demo"URL

  Follow the instructions and solve this problem.

Unit 4: Learning

This unit presents an artificial neural network (NN) as the most important learning tool for machine learning. Machine learning research tries to automatically extract information from data through computational and statistical methods. Machine learning is closely related to not only data mining and statistics, but also theoretical computer science. NN is a computational model based on biological neural networks. It consists of an interconnected group of artificial neurons and processes. Practically, neural networks are non-linear statistical data modeling tools used to model complex relationships between inputs and outputs. After being successfully trained, NNs are able to perform classification, estimation, prediction, or simulation with new data. The second part of this unit reviews the Gaussian and Bayesian processes used in machine learning.

4.1: Machine Learning

- Wikipedia: "Machine Learning"URL

  Read this article for an overview of machine learning. Be sure you understand the differences between the learning methods, which you can read about beneath the 'Algorithm types' section.

- **4.1.1: Reinforcement Learning**

  - John Lloyd's "Intelligent Agents"URL

    Watch this lecture and pay attention to the AIMA learning agent. Compare Lloyd's explanation of reinforced learning with others that you have seen.

- **4.1.2: Machine Learning, Probability, and Graphical Models**

  - Sam Roweis' "Machine Learning, Probability, and Graphical Models"URL

    Watch this lecture to review the applications of probabilistic learning, the concept of representation, and examples of training and graphical models. You may wish to work through the slides available on the left-hand side of this web-page as you listen to the lecture.

4.2: Neural Network

- **4.2.1: Introduction to Neural Networks**
  - Wolfram: "Introduction to Neural Networks"URL

    Read this section to learn about general neural networks and how they are mathematically defined.

- **4.2.2: Feedforward Neural Networks**
  - Wolfram: "Feedforward Neural Networks"URL

    Read this section, which covers the Feedforward Neural Network. Make sure you understand this network's mathematical definition and that you study the figures in this article.

- **4.2.3: Radial Basis Function Networks**
  - Wolfram: "Radial Basis Function Networks"URL

    Read this section, which covers the Radial Basis Function Network. Make sure you understand the network's mathematical definition and be sure to study the figures in this article.

- **4.2.4: The Perceptron**
  - Wolfram: "The Perceptron"URL

    Read this section about Perceptron. Be sure to understand its mathematical definition, learn the training algorithm, and study the figures.

- **4.2.5: Vector Quantization (VQ) Networks**
  - Wolfram: "Vector Quantization Networks"URL

    Read this section about the Vector Quantization network.

- **4.2.6: Hopfield Network**
  - Wolfram: "Hopfield Network"URL

    Read this section, which presents the Hopfield network and the equations that describe it.

4.3: Other Classifiers and Statistical Learning Methods

- **4.3.1: Kernel Methods**
  - Wikipedia: "Kernel Methods"URL

Read this article to review Kernel methods.

- **4.3.2: k-nearest Neighbor Algorithm**
  - Wikipedia: "k-nearest Neighbor Algorithm"URL

    Make sure you know how the k-nearest neighbor algorithm works (in principle) after reading this entry.

- **4.3.3: Mixture Model**
  - Wikipedia: "Mixture Model"URL

    Read this article to learn about the different types of Mixture Models.

- **4.3.4: Naive Bayes Classifier**
  - Wikipedia: "Naive Bayes Classifier"URL

    Read this article and make sure you know the definition of the naive Bayes classifier.

- **4.3.5: Decision Tree**
  - Wikipedia: "Decision Tree"URL

    Read this article. You should be able to define the term "decision tree" when you are done.

- **4.3.6: Kernels and Gaussian Processes**
  - Mark Girolami's "Kernels and Gaussian Processes: Parts 1-3"URL

    Watch the first video about machine learning and compare it to what you have learned already. After watching this video, you should the basics of linear regression, loss function, prediction techniques. Study non-linear models, probabilistic regression, and uncertainty estimation. Then, watch the second video lecture to learn about Bayesian regression and classification. Finally, watch the last lecture to learn about Gaussian processes, regression, and classification.

4.4: Machine Learning Coding Drills

- "Tic-Tac-Toe Demo"URL

  Code an agent that plays the Tic-Tac-Toe game. You can choose to play the game yourself by selecting board positions or have the Agent propose moves. This link shows one possible solution. Work towards a solution for no more than 10 hours and then check your work against the solution code.

Unit 5: Philosophical Foundations of AI

In this unit, we will study the Turing machine as a definition of the intuitive notion of computability in the discrete domain. In the theory of computation, many major complexity classes can be characterized by an appropriately restricted Turing machine. We will also discuss the claim that human thinking is a kind of symbol manipulation. Note that a symbol system is necessary for intelligence and that machines can be intelligent. Finally, we will discuss the ongoing neuroscientific attempt to understand how the human brain works and examine the possible role of consciousness in the machines. Is it possible, in theory and then in practice, to create a machine that has all the capabilities of a human being?

5.1: Computing Machinery and Intelligence

- **5.1.1: Philosophical Issues and Turing Test**
  - John Lloyd's "Intelligent Agents"URL

    Watch the third part of this lecture series and compare his interpretation of the Turing test with what you learn later in this unit.
- **5.1.2: Computing Machinery and Intelligence**
  - A. M. Turing's "Computing Machinery and Intelligence"URL

    Read this paper by A. M. Turing, a cornerstone in the field of A.I. studies.
- **5.1.3: Turing Machine**
  - Paul M.B. Vitanyi's "Turing Machine"URL

    This article is fairly challenging; read through it to the best of your abilities for a detailed description of the Turing Machine. After you have read this article, you should know how to define Turing machine and summarize the Church-Turing theses. Make sure you know what the Halting problem is.
- **5.1.4: Computability and Incompleteness**

5.2: Important Propositions in the Philosophy of AI

- Wikipedia: "Artificial Brain"URL

  Read this article.
- Wikipedia: "Physical Symbol System"URL

  Read this article.

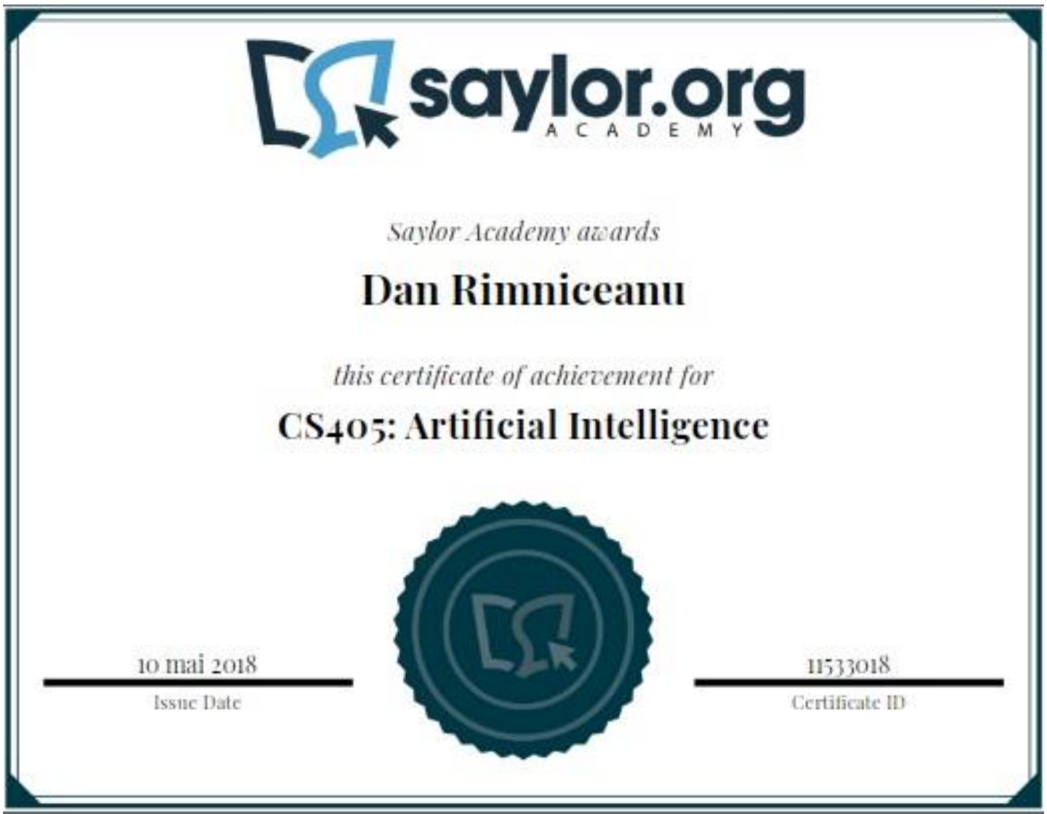
5.3: Machine Consciousness

- Igor Aleksander's "Machine Consciousness"URL

Read this article, which discusses machine consciousness. Focus on learning about early models of consciousness and neural models of consciousness.

- Massachusetts Institute of Technology: Marvin Minsky's "Emotion Machine"URL

Watch this lecture about emotional machines. Ask yourself whether you think one is possible and begin to think about how you would approach its creation.



Course Introduction
CS405 introduces the field of artificial intelligence (AI). Materials on AI programming, logic, search, game playing, machine learning, natural language understanding, and robotics introduce the student to AI methods, tools, and techniques, their application to computational problems, and their contribution to understanding intelligence. Because each of these topics could be a course unto itself, the material is introductory and not complete. Each unit presents the problem a topic addresses, current progress, and approaches to the problem. The readings include and cite more materials that are referenced in this course, and students are encouraged to use these resources to pursue topics of interest after this course.

Unit 1: Introduction to Artificial Intelligence (AI) and AI Programming
While AI applications can be developed in any number of different languages, certain language features make programming AI applications straightforward. Prolog is structured in such a way that AI program development is supported by Prolog language

features. Other languages, such as Java, support AI programming through code libraries. At this point in your career as a computer science major, you have already taken introductory programming courses; these should assist you in learning Prolog and using code libraries in other languages for AI program development.

This unit will provide you with an introduction to AI via programming features that support basic AI applications. By satisfying the goals of this unit, you will have a familiarity with AI programming and be able to use it in future models to implement various AI applications.

Completing this unit should take you approximately 10 hours.

- Unit 1 Learning OutcomesPage

- **1.1:Artificial Intelligence Introduction and Overview**
  - University of California, Irvine: Max Welling's "Introduction to AI"URL

    Read these slides, which summarize the types and scope of AI applications and the state of the art in the field, as well as the early history of AI.
  - Mark Watson's "Practical Artificial Intelligence Programming with Java"URL

    Read Chapter 1 on pages 1-4. AI concepts and techniques are learned in two steps: theory, and implementation of theory in programs. AI techniques are difficult to program and can be obscured by the programming details. To make these techniques explicit and to hide the programming details, AI languages, such as Lisp, Prolog, Scheme, and others, have been defined to have language features that directly support the implementation of AI techniques. The use of class libraries and source code libraries serve the same purpose for general-purpose languages, such as C++ and Java. The Watson text uses Java Classes for program examples of AI concepts and techniques. Java, being widely used, has the added advantage of making these techniques more widely available.

- **1.2: Logic Programming**
  - Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Logic Programming"URL

    Read these slides on logic as a programming language. In declarative programming, you write statements that specify how to manipulate the data; that is, you write specific algorithms. In logic programming, you write statements that specify what is true of the world represented and rely on general-purpose algorithms that are built into the logic (programming) system.

Unit 2: Search

Previous coursework has familiarized you with searching algorithms. In this unit, you will learn how to implement standard searching algorithms. We will first discuss the motivation behind exploring search from an AI perspective, learning new terminology as we go that will be used in this unit and beyond. We will then learn about basic search methods, as well as time and memory requirements, concluding with a discussion of the advantages and disadvantages of searching algorithms. By the end of this unit, you will be able to apply AI techniques when developing searching algorithms.

Completing this unit should take you approximately 12 hours.

- Unit 2 Learning OutcomesPage

- **2.1: Motivation**
  - Mark Watson's "Practical Artificial Intelligence Programming with Java"URL

    Read Chapter 2 on the search problem. One way to solve a problem is by searching for a solution in a set, called the search space. This approach assumes that a search can be done in an acceptable amount of time at an acceptable cost.

- **2.2: Types of Searches**
  - Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Search, Part 1"URL

    Read these slides.

  - Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Search, Part 2"URL

    Read these slides.

- **2.3: Time and Space Requirements, More on Heuristics, and Complexity**
  - Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Search, Part 3"URL

    Read these slides.

Unit 3: Constraint Satisfaction

AI applications are built upon the idea of a problem statement with constraints. In AI, we must work within those constraints in order to develop an optimal solution. In this unit, we will define "problem" in specific AI terms and discuss different approaches to constraint satisfaction. Constraint satisfaction is an important subject area within AI. The famous Map Coloring Problem has simple variables and simple constraints and is thus useful in illustrating the basics of constraint satisfaction. By the end of this unit, you will be able to solve basic problems.

Completing this unit should take you approximately 10 hours.

- Unit 3 Learning OutcomesPage

- **3.1: Problem Definition and Approaches**
  - Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Constraint Satisfactory Problems (CSP) and Games"URL

    Read these slides. CSP stands for Constraint Satisfaction Problems, which include, for example, problems in task scheduling, planning robot actions, solving puzzles (for example, the classic N-Queens Problem and the Four Color Problem), and interpreting sensory data.

Unit 4: Game Playing

Some of the earliest and most recognizable AI applications are games like chess and tic-tac-toe, the most famous being the chess match between Garry Kasparov and Deep Blue. In this unit, we will discuss the development of game-playing applications, as well as the relationship between game-playing and searching algorithms. The unit will also provide you with some best practices for building game programs.

This unit has been designed to teach you how to design algorithms for game-playing applications. For our purposes, you will find tic-tac-toe, which uses features of search and constraint satisfaction, the simplest. We suggest that as an informal exercise, you create a tic-tac-toe application and then play against it, noting the algorithm's success rates and determining which modifications will need to be implemented in order to improve its performance.

Completing this unit should take you approximately 10 hours.

  - Unit 4 Learning OutcomesPage

- **4.1: Background: Game Playing in AI**
  - Wikipedia: "History of Artificial Intelligence"URL

    Read this history of artificial intelligence.
  - University of California, Irvine: Max Welling's "Introduction to AI"URL

    Review these slides. Game playing provided numerous applications that motivated the development of AI techniques, such as search and problem-solving techniques. In addition, it served as a popular benchmark for demonstrating progress and improvements of AI research.

- **4.2: Applicability of Search**
  - Mark Watson's "Practical Artificial Intelligence Programming with Java"URL

    Read section 2.5 for a discussion of how to program searching and implement those ideas in Java programs for game playing. Min-max is a search strategy for two-person games whereby a move is selected by choosing the child node that has either the maximum (a player strives to maximize his/her advantage) or minimum (a player strives to minimize the other player's advantage). Alpha-Beta search is an improvement of min-max searching by eliminating, or pruning, branches from the search tree.

- Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Constraint Satisfactory Problems (CSP) and Games"URL

  Read these slides, which discuss the application of ideas presented so far on two-person games, like tic-tac-toe, checkers, and chess. The states of a board game are the board positions or configurations of the game pieces, such as pieces on the chessboard. The states are represented by the nodes of a tree. The game moves, or operators, are represented by the arcs of a tree. The goal state is a goal node, i.e., the board configuration that depicts the winning position of the game pieces. In many games, the size of the search tree can be very large and the complexity of the search can be high. Heuristics are used to guide the search, increase efficiency, and improve game-playing ability.

- University of California, Irvine: Max Welling's "Games"URL

  Read these slides. MinMax and Alpha-Beta are fundamental search algorithms used in game playing.

Unit 5: Logic

We have already briefly discussed logic, but this unit will provide you with a more formal definition. We will learn about two main types of logic--propositional and first-order. Prolog was designed for expressing logic. This unit gives you a strong foundation in logic so that you will be able to use or learn Prolog more easily to program logic applications. Similarly, you will be able to use or learn class libraries that support AI techniques in other languages, like C++ and Java.

Completing this unit should take you approximately 24 hours.

- Unit 5 Learning OutcomesPage

- **5.1: Definition and Types of Logic**
  - Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Logic"URL

    Read the following slides:

    - Logic I
    - Logic Ib
    - Logic II
    - Logic IIb
  - University of California, Irvine: Max Welling's "Logic"URL

    Read these slides.

- **5.2: Some Examples of Practical Applications**
    - Mark Watson's "Practical Artificial Intelligence Programming with Java"URL

      Read Chapters 3 through 5. Chapter 3 gives examples and tools for applying logic. Chapter 4 discusses the Semantic Web. Reasoning assumes a body of data from which inferences can be made, and this chapter discusses the Semantic Web as a source of data for use in programs, in particular for inference algorithms. Chapter 5 gives examples and tools for expert systems application in reasoning.

Unit 6: Machine Learning

Machine Learning refers to computer programs that are able to categorize data in order to maximize understanding of that information. Machine Learning is closely related to statistics and modeling and has a wide range of applications, from natural language processing, searching, robotics, and indexing, to other pattern recognition applications. This unit will begin by defining Machine Learning, its applications, and a number of other important terms that will be used in this unit. We will then go over the three main classes of Machine Learning: Supervised Learning, Semi-Supervised Learning, and Unsupervised Learning. You will also end up with an introductory foundation in Machine Learning that will be useful for further academic study in the field.

Completing this unit should take you approximately 26 hours.

- Unit 6 Learning OutcomesPage

- **6.1: Definition of Learning and Machine Learning**
    - Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Machine Learning Introduction"URL

      Read these slides. Machine learning is learning using methods that can be implemented in software.
    - Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Machine Learning I"URL

      Read these slides.
    - Mark Watson's "Practical Artificial Intelligence Programming with Java"URL

      Read Chapters 6 and 7. Recognition of patterns occurs in search, game playing, language recognition, expert systems and rule-based systems, vision, and learning.
    - Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Feature Spaces"URL

      Read these slides.

- Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Machine Learning IV"URL

  Read these slides, which deal with huge numbers of features by feature selection, ranking, and applications to clustering.

- **6.2: Types of Machine Learning**
  - Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Machine Learning III"URL

    Read these slides, which discuss training for artificial neural nets.

  - Wikipedia: "Types of Machine Learning"URL

    Read the following articles on types of machine learning.

    - Supervised Learning
    - Unsupervised Learning
    - Self-organizing Map
    - Adaptive Resonance Theory
    - Semi-supervised Learning
    - Co-training
    - Maximum Likelihood
    - Expectation Maximization

    There are many learning methods, each having strengths and weaknesses in particular applications, for particular data sets and situations. Issues that have to be contended with include: bias (a predicted value of a learning algorithm is systematically incorrect when trained on several different data sets) and variance (variation of a predicted value for a given input when trained on different data sets), complexity of functions to be predicted, complexity of data, noisy data, missing data, etc.

  - University of California, Irvine: Max Welling's "Machine Learning"URL

    Read these slides.

  - Stanford University: Andrew Ng's "Machine Learning"Page

    Watch these lectures.

  - Stanford University: Samuel Ieong's "Probability Theory Review"URL

    Read this section, which describes joint distributions. Joint distributions from probability theory are useful for studying semi-supervised learning. Two statistical techniques that are also helpful are maximum likelihood and expectation maximization, both of which are used to estimate the parameters of statistical models.

- **6.3: A Practical Tool for Machine Learning**
    - [Mark Watson's "Practical Artificial Intelligence Programming with Java"URL]

      Read Chapter 8 on pages 129-136, which describes tool support for machine learning.

Unit 7: Natural Language Understanding
This unit will provide you with a basic introduction to Natural Language Understanding (NLU) in AI. Syntax, semantics, and ambiguity of natural language are discussed. Simple examples are presented. Some of what we have seen, in search and in learning, is applied in NLU. Natural language processing and understanding is a large field of research and has entire courses devoted to it. So, in this introduction, our objective is simply to introduce the problems and approaches.

Completing this unit should take you approximately 20 hours.

- Unit 7 Learning OutcomesPage

- **7.1: Language Overview and Basics**
    - Mark Watson's "Practical Artificial Intelligence Programming with Java"URL

      Read Chapters 9 and 10. Chapter 9 gives practical examples of NLP from a programming perspective. Chapter 10 provides additional discussion on extracting semantic information from text and databases.

- **7.2: Syntax and Semantics**
    - Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Language Understanding I"URL

      Read these slides.
    - Massachusetts Institute of Technology: Leslie Kaelbling and Tomás Lozano-Pérez's "Language Understanding II"URL

      Read these slides.

Unit 8: Robotics
Robotics draws upon and integrates previous topics, as well as information and techniques from other disciplines, including many engineering fields, physics, controls, probability and statistics, differential equations, linguistics, and many applications, e.g., manufacturing, sensors, medical applications, etc. Some of the contributions of AI to robotics are search algorithms, representation and models for the robot world, inference, learning, and AI programming features and their integration.

Completing this unit should take you approximately 9 hours.

- Unit 8 Learning OutcomesPage

- **8.1: Introduction and Overview of Robotics**
  - Massachusetts Institute of Technology: Harry Asada and John Leonard's "Introduction to Robotics"URL

    Read this section, which gives a mechanical engineering perspective on robotics.
  - Massachusetts Institute of Technology: Daniela Rus' "Introduction to Robotics"URL

    Read this overview of the history of robots, definitions in robotics, and a less mechanically-oriented perspective on robotics.

- **8.2: Positioning, Vision, Planning, and Control**
  - Stanford University: Oussama Khatib's "Introduction to Robotics"Page

    Watch these lectures.

- **8.3: Practical Software and System Process for Robotic Applications**
  - Massachusetts Institute of Technology: Seth Teller's "System Engineering and Testing Strategies"URL

    Read these slides, which address systems issues in robotics applications.