Big Idea #2: Representation & Reasoning

Key Insights	Explanation
Representations are data structures; reasoning methods are algorithms.	Artificial intelligence uses the tools of computer science: data structures and algorithms.
Representations support reasoning; reasoning methods operate on representations.	Representation and reasoning are mutually dependent.
The two major types of knowledge representations are symbolic and numerical representations.	Reasoning with symbolic representations is performed using logical inference rules, while reasoning with numerical representations utilizes complex mathematical functions such as neural networks.
"Knowing" something means the ability to both represent it and reason with it.	Books and videos can represent knowledge but they don't "know" things because they can't make use of that knowledge.
Agents are considered intelligent if they employ a non-trivial sense-deliberate-act cycle to make progress toward achieving their goals.	To be considered intelligent, at least one of the sense, deliberate (reason), or act components must require computational sophistication or signifcant computing power. Garage door openers are not intelligent agents because their sensing, reasoning, and action are all trivial.

Big Idea #2: Representation and Reasoning	*Computers maintain representations of the world and use them for reasoning.*	LO = Learning Objective: What students should be able to do. EU = Enduring Understanding: What students should know. Unpacked descriptions are included when necessary to illustrate the LO or EU		
Concept	**K-2**	**3-5**	**6-8**	**9-12**
Representation *(Abstraction)* **2-A-i**	**LO:** Construct a map of a home, school, or neighborhood. **EU:** The map is not the territory. A map is a *representation* of a territory.	**LO:** Demonstrate how different styles of map capture different aspects of the world. **EU:** Maps can be geometrically accurate or just show which places are reachable from which other places (topology). They can contain many types of information, such as the names and sizes of cities and towns, or the types of roads connecting them. **Unpacked:** Self-driving cars rely on road maps to plan their routes. Subway, train, and bus route maps may be either geometric or topological.	**LO:** Show how a game board (e.g., tic-tac-toe,Chutes and Ladders, Monopoly, chess) can be represented by a description in plain language. **EU:** The essential information in a game board is the set of squares that make up the board, the relationships among those squares, and the positions of the pieces. **Unpacked:** Game playing programs operate on abstract representations of boards. A chess program knows the ranks and files of all 64 squares, but does not represent properties of a physical board such as its dimensions or the material it is made of.	**LO:** Describe how to represent a concept as a schema. **EU:** a schema describes a concept by listing its superconcepts and defining its properties, some of which may be inherited from superconcepts. Examples can be found at schema.org. **Unpacked:** millions of web sites use schema representations to make information intelligible to AI programs that utilize knowledge graphs. such as search engines and recommender systems. Examples include schemas for businesses such as restaurants, and schemas that describe creative works such as books, movies, and television series.
Representation *(Symbolic representations)* **2-A-ii**	**LO:** Give examples of symbols you encounter in daily life. **EU:** Concepts can be represented using symbols in place of words. **Unpacked:** Examples of symbols include male and female signs on restrooms, emojis, the clubs/diamonds/hearts/spades suits in a deck of cards, the dollar and cents signs for currency, red octagons for stop signs, and the "Mr. Yuk" poison symbol. AI programs use symbols just as people do.	**LO:** Give examples of *tree structures* commonly used by people and explain the relationships among the components. **EU:** Many types of information have a hierarchical structure, which can be visualized as a tree. **Unpacked:** Ancestry trees, taxonomies, tables of contents, organizational trees, and simple forms of mind map all have the same hierarchical structure: every node save the root has a single link to a parent node that appears higher in the hierarchy. Taxonomic information is an important part of the knowledge graphs used by AI systems, and typically forms a tree structure within the larger graph.	**LO:** Illustrate translation of a structure such as a game board. road map, or mind map into a *labeled graph* and explain the contributions of the components. **EU:** The nodes of a graph represent things, and the links represent relationships between those things. Labels supply additional information about what the nodes and links represent. **Examples:** In the game Chutes and Ladders (originally Snakes and Ladders), the squares are represented as nodes in a graph. Nodes representing adjacent squares are connected by "successor" lnks. Nodes whose squares are reachable by chutes or ladders are connected by "chute" or "ladder" links. In a road map, cities are nodes and roads are links. The links might be labeled with the road name or a mileage value. Mind maps are a kind of informal knowledge graph used by people, not by computers.	**LO:** Translate the premises of a syllogism expressed in English into *logical notation* and complete the syllogism correctly. **EU:** Syllogisms represent statements about categories and instances in a way that allows computers to derive new knowledge from existing knowledge by following simple inference rules. The components of a syllogism are the major premise, the minor premise, and the conclusion. **Unpacked:** Human reasoning is rich and complex, and we don't have a complete description of how it works. Syllogistic reasoning is a special case that is simple enough that we can write precise rules for how to do it. Syllogistic reasoning is a type of logical deduction that is common in AI reasoners that use taxonomy information. Example: all humans are mammals, all mammals are living things, therefore all humans are living things. A variety of notations can be used for syllogisms. Here is one based on predicate logic: **∀x [Human(x) → Mammal(x)]** **∀x [Mammal(x) → LivingThing(x)]** **∀x [Human(x) → LivingThing(x)]**

Big Idea #2: Representation and Reasoning		*Computers maintain representations of the world and use them for reasoning.*	LO = Learning Objective: What students should be able to do. EU = Enduring Understanding: What students should know. Unpacked descriptions are included when necessary to illustrate the LO or EU	
Concept	**K-2**	**3-5**	**6-8**	**9-12**
Representation *(Data structures)* **2-A-iii**	**LO** Draw a tree by repeatedly splitting each branch into sub-branches multiple times, and putting a piece of data at each branch. **EU:** A tree is a way of organizing information. **Unpacked:** Many types of data could be placed on a branch. For a decision tree, the data would be questions and answers. For an ancestry tree, there would be "mother of" branches and "father of" branches. For a taxonomic tree there would be taxa of increasing specificity, e.g., animals and plants, mammals and reptiles, primates and rodents. **Guided activity resource:** tree drawing exercise.	**LO:** Describe the parts of a tree and how those parts are related. **EU:** A tree is a collection of labeled **nodes**, each of which (save the **root**) has a **link** to a **parent** node above it in the hierarchy. Childless nodes are called **leaves** or **terminal** nodes; those with **children** are **non-terminal** nodes.	**LO**: Describe the parts of a graph and how those parts are related. **EU:** A graph is a collection of labeled **nodes** connected by labeled **links**. Every link has a **source** node and a **target** node; every node hast a set of **incoming** links and a set of **outgoing** links. Trees are a special case of graphs. **Unpacked:** Graphs in AI are used to represent things such as road maps (where the cities are nodes and the roads are links), mazes, and knowledge graphs. A knowledge graph can represent people, places, and things, and the relationships between them. Mind maps are mostly trees but it is possible to have multiple paths to a node, making the map a graph. Each node is a concept and the links can be labeled with relations between concepts, or they may be unlabeled.	**LO:** Describe how schemas are used to structure information about people, places, or things in knowledge graphs. **EU:** A schema specifies the attributes of the concept being described, and its relationships to other concepts, e.g., the *Restaurant* schema inherits properties from the *FoodEstablishment* schema. **Unpacked:** A knowledge graph encodes information about things and the relationships between them. Search engines like Google rely on knowledge graphs to generate knowledge panels in search results. The *Restaurant* and *FoodEstablishment* schemas are defined at schema.org.
Representation *(Feature vectors)* **2-A-iv**	**LO:** Identify the features that make each object in a collection unique. and create a table of features to organize the objects. **EU:** Objects can be described in terms of the features they possess. **Unpacked:** This could be as simple as Legos bricks of different shapes, sizes, and colors, or features that distinguish different types of animals: cats, dogs, chickens, goldfish, penguins, etc., e.g., does it have fur, does it fly, etc. Another option is features that describe face emojis indicating different emotional states.	**LO:** Construct a feature vector representation for a set of objects and show how similar objects are close together in feature space. **EU:** Recommender systems represent things like movies, books, consumer products. or social media posts using feature vectors. **Unpacked:** Feature vectors represent concepts as sequences of numbers. The distance between two feature vectors can be measured by counting the number of positions at which they disagree, so similar objects lie closer together in feature space. Feature vectors can be constructed by hand, but they can also be constructed automatically using machine learning. **Example:** in the Pasta Land exercise students develop a discrimination tree for recognizing different types of pasta. The questions that make up the nodes of the tree can provide the features for a binary feature vector representation of the pasta types.	**LO:** Explain how word embeddings (which are feature vectors) represent words as sequences of numbers. **EU:** Word embeddings are a key part of neural natural language processing, including machine translation (e.g., Google Translate) and text generation systems (BERT, GPT3, etc.) **Unpacked:** Each word is a point in a feature space with many dimensions, organized so that words with similar meanings are close to each other in the feature space. See this Word2VecDemo.	**LO:** Describe how a transformer network operates. **EU:** Transformer networks map sequences of input words to sequences of output words, where words are represented as feature vectors. **Unpacked:** Neural network natural language processing applications such as machine translation or question answering are driven by word embedding representations, which are feature vectors. Words are fed in one vector at a time, and the network delivers its output one vector at a time. **Activity:** https://app.inferkit.com/demo

Big Idea #2: Representation and Reasoning	*Computers maintain representations of the world and use them for reasoning.*	LO = Learning Objective: What students should be able to do. EU = Enduring Understanding: What students should know. Unpacked descriptions are included when necessary to illustrate the LO or EU	

Concept	K-2	3-5	6-8	9-12
Search *(State spaces and operators)* **2-B-i**	**LO:** Illustrate a next possible state in the game of tic-tac-toe given a starting state. **EU:** A game such as tic-tac-toe can be described as a sequence of states, where each move transitions from a state to a successor state. **Unpacked:** Each state should be drawn as a separate tic-tac-toe board. Answers may vary depending on which move the student chooses to make. **Resource:** online tic-tac-toe games: https://playtictactoe.org/ or https://www.coolmathgames.com/0-tic-tac-toe	**LO:** Illustrate how a computer can represent the playing of a game such as tic-tac-toe or nim by drawing the linear sequence of board positions produced by the players' moves. **EU:** Computers play games and solve puzzles by creating a sequence of states (board positions) connected by legal moves, using an algorithm to choose their next move at each step. **Unpacked:** The state space (or search space) of a game is the set of all board states reachable from the start state (illustration), and the operators are the set of possible moves a player can make that adhere to the rules of the game. A particular game (linear sequence of board positions: illustration) is one path through this state space.	**LO:** Illustrate how a computer can solve a maze, find a route on a map, or reason about concepts in a knowledge graph by drawing a search tree. **EU:** Computers solve mazes, find driving routes, and reason about concepts in knowledge graphs using graph search algorithms, which construct search trees. **Unpacked:** The search space of a graph search problem is the set of all paths originating from the designated start node of the graph. The operators used for solving a maze move one node north, south, east, or west. In the more general case of graph search the operators extend a path by adding a new node at the end. Legal moves add a node that is reachable by a direct link in the graph. Legal states are those reachable by a sequence of legal moves. **Example:** to determine whether a kangaroo is a mammal, we search for a path in the knowledge graph from the "kangaroo" node to the "mammal" node that is comprised of "is-a" links. Quick intro article.	**LO:** Identify types of real-world problems that are search problems and describe their states and operators. **EU:** Computers can solve many types of problems using search techniques if the problem can be described in terms of finding a path from a start state to a goal state. **Unpacked:** Examples include task planning problems, scheduling problems, and resource allocation problems. A search algorithm determines which operators to apply, in which order. Finding a sequence of legal moves (operators) to reach a goal state can be used even with problems whose solution is not a sequence. For example, if the problem is to pack a collection of objects of various sizes into a set of containers with various capacities, a solution is an assignment of objects to containers such that no container is overfilled and no object is left out. This can be formulated as a search problem where an operator places one object in a container that can hold it, and a goal state has all objects placed. For this type of problem, the sequence in which the operators are applied does not matter.
Search *(Combinatorial search)* **2-B-ii**	**LO:** Draw a simple search tree to illustrate two possible choices of moves by drawing two different successor states to a given tic-tac-toe-game board state, forming a tree. **EU:** A computer must make choices when playing games, in the same way that humans do. **Unpacked:** Students will draw a three-node tree, with the starting game state as the root, and two branches descending from the root showing alternative moves. The starting state need not be the empty board. Illustration.	**LO:** Given a state of a game such as tic-tac-toe or nim, draw a search tree showing all possible next moves and their resulting states, and pick the best next move. **EU:** Computer game playing programs may construct search trees to decide on their next move. **Unpacked:** Search trees are a way of systematically exploring all possible moves in a game. (Illustration.) The state space of a game is the set of all board states reachable from the start state, and the operators are the set of possible moves a player can make that adhere to the rules of the game.	**LO:** Model the process of solving a graph search problem using breadth-first search to draw a search tree. **EU:** Breadth-first search can be used to solve problems that involve reasoning about graphs. **Unpacked:** In breadth-first search, all the nodes at the current level of the search tree are expanded, one at a time, to generate the next level. Then the next level is expanded, and the process continues until the goal is reached. Given a graph of friendship relations, determining whether two people have a friend in common is a graph search problem. Another graph search problem is determining the degree of separation of two people, as in "six degrees of Kevin Bacon".	**LO:** Illustrate breadth-first, depth-first, and best-first search algorithms to grow a search tree for a graph search problem. **EU:** There are multiple algorithms for generating a search tree, each with its own advantages. **Unpacked:** Illustration is done by drawing the tree; writing code is appropriate for advanced students (AP CSA). Breadth-first search finds shallow solutions quickly if they exist, but requires lots of memory. Depth-first search is better than breadth-first for cases where the number of nodes at each level grows exponentially, because it uses less memory. Best-first search can find optimal (least cost) solutions given an accurate measure of solution "cost", such as total distance on a road map, while the solutions found by breadth-first and depth-first search are not guaranteed to be optimal because they don't pay attention to cost.

Big Idea #2: Representation and Reasoning		*Computers maintain representations of the world and use them for reasoning.*	LO = Learning Objective: What students should be able to do. EU = Enduring Understanding: What students should know. Unpacked descriptions are included when necessary to illustrate the LO or EU	
Concept	**K-2**	**3-5**	**6-8**	**9-12**
Reasoning *(Types of reasoning problems)* **2-C-i**	**LO:** Identify problems as either classification problems or search problems. **EU:** In classification problems we decide what kind of thing we have based on its features. In search problems we find a path from a start to a goal, such as finding a route on a map or exploring possible moves in a game.	**LO:** Categorize problems as either classification problems or search problems. **EU:** Classification problems assign each input to one of a predetermined set of classes. Search problems construct answers by applying operators to states to generate new states. **Unpacked:** Labeling images as dog photos or cat photos (as in Teachable Machine) is a classification problem. Finding the board positions that can be reached in one move from a given starting position is an example of a search problem.	**LO:** Categorize problems as classification, prediction, combinatorial search, or sequential decision problems. **EU:** Prediction problems are similar to classification problems except they estimate a continuous value, such as height or daily temperature. Sequential decision problems choose the next move for any given state in order to maximize overall reward. **Unpacked:** Sequential decision problems are covered in Big Idea 3; they are addressed using reinforcement learning. **Examples:** spam vs. not-spam (classification), tomorrow's high temperature (prediction), solving puzzles such as the wolf, goat, and cabbage problem (combinatorial search), and playing a video game such as Super Mario (sequential decision problem).	**LO:** Categorize real-world problems as classification, prediction, sequential decision problems, combinatorial search, heuristic search, adversarial search, logical deduction, or statistical inference. **EU:** Reasoning problems can be categorized based on the types of inputs supplied, the types of outputs to be produced, and the characteristics of the search space, if applicable. **Unpacked:** Heuristic search is needed when the state space is too large to examine all possible states. uses a rule of thumb (heuristic) to limit the search by focusing on the most promising states. In adversarial search, used in game playing, the algorithm alternates between finding the best move for the player and finding the best response for the opponent, which would be the worst move from the player's perspective. Adversarial search may require heuristics if the game is complex, such as chess or go. In logical deduction, the reasoner starts with a set of facts and derives new facts by applying inference rules. Logical deduction can be done using formal logic such as propositional or predicate logic, or ad hoc inference rules used with semantic networks or the IF-THEN rules found in expert systems. Statistical inference involves reasoning with probabilities.
Reasoning *(Reasoning algorithms)* **2-C-ii**	**LO:** Model the use of a classification or search algorithm to solve a problem. **EU:** Reasoning algorithms are ways to solve reasoning problems. **Unpacked:** Students could solve a classification problem using the table of features they developed in 2-A-iv, or the decision tree they developed in 3-A-v. They could solve a map search problem using the map they developed in 2-A-i.	**LO:** Describe the differences between two algorithms for classifiying things: decision trees, or neural networks like Teachable Machine. **EU:** Multiple algorithms can be used to solve a reasoning problem. **Unpacked:** Classification of pasta could be done either by constructing a decision tree (see the Pasta Land activity) or by training a visual classifier like Teachable Machine on example images of each pasta type. Pasta Land description: https://www.aimyway.com/blank-page-1	**LO:** Compare several algorithms that could be used to solve a specific type of reasoning problem. **EU:** The choice of reasoning algorithm depends on the characteristics of the input data and the types of decisions to be made. **Unpacked:** For classifying dogs vs. cats, if all we have are raw images, we cannot use a decision tree because it would require an astronomical number of nodes; we must use a neural network because it can efficiently perform feature extraction. But if we already have a description of an animal in terms of features such as snout length and ear shape, then a decision tree could be used. In general, classification problems can be solved using decision trees, nearest-neighbor algorithms, or neural networks. Prediction problems can be solved using linear regression or neural networks. Combinatorial search can make use of several different algorithms for growing the search tree. Sequential decision problems can use either Q tables or neural networks to choose the best action.	**LO:** For each of these types of reasoning problems (classification, prediction, sequential decision making, combinatorial search, heuristic search, adversarial search, logical deduction, and statistical inference), list an algorithm that could be used to solve that problem. **EU:** AI includes a wide variety of reasoning algorithms for solving different types of reasoning problems. Some use symbolic representations (e.g., search trees) while others are numerical in nature (e.g., neural nets operating on feature vectors). **Unpacked:** Reasoning problems are discussed in 2-C-i. Selection of an algorithm depends on characteristics of the input data and the complexity of the decisions to be made. Heuristic search algorithms include best-first search and A* search. Adversarial search can be done with variants of combinatorial search, or heuristically with alpha-beta pruning. Logical deduction can be done using resolution theorem proving or by applying ad hoc IF-THEN rules. Statistical inference involves reasoning with probabilities or distributions, such as Bayesian networks.