**KBAI Exam 2 - Unofficial Study Guide**

Daniel Torres

## References:

* Lecture Slides
* AI - Patrick Henry Winston (3rd edition)
* <https://quizlet.com/414373298/kbai13-planning-flash-cards/>
* <https://quizlet.com/414381556/kbai15-commonsense-reasoning-flash-cards/>
* <https://quizlet.com/414445959/kbai16-scripts-flash-cards/>
* <https://quizlet.com/414451228/kbai17-explanation-based-learning-flash-cards/>
* <https://quizlet.com/416359769/version-spaces-flash-cards/>
* <https://quizlet.com/322380942/kbai-ch-10-17-flash-cards/>
* <https://quizlet.com/415582442/kbai-18-analogical-reasoning-flash-cards/>
* <https://quizlet.com/416362327/20-contraint-propagation-and-configuration-flash-cards/>
* <https://quizlet.com/416368623/22-diagnosis-and-flash-cards/>
* <https://quizlet.com/416371973/23-learning-by-correcting-mistakes-flash-cards/>
* [Exam two questions Flashcards](https://quizlet.com/573921641/exam-two-questions-flash-cards/)

## Content

[References:](#_8h9gwi5vpibp)

[Content](#_h37gxzkls4pj)

[13: Planning (75)](#_lrymsnuls6fb)

[14: Understanding (30)](#_tnwqs7klgmfr)

[15: Commonsense Reasoning (60)](#_19jtxfbee6gb)

[16: Scripts (30)](#_gnodhimynif4)

[17: Explanation-Based Learning (45)](#_9f4ly2evjb1v)

[18: Analogical Reasoning (60)](#_kbeddpzjbcc)

[19: Version Spaces (60)](#_fyj78lvbh2co)

[20: Constraint Propagation (45)](#_9oetd3nke9mt)

[21: Configuration (45)](#_dnl7k52058ol)

[22: Diagnosis (45)](#_36m03hhxvix1)

[23: Learning by Correcting Mistakes (45)](#_bbr14acje0l3)

[24: Meta-Reasoning (30)](#_h061ck9lzorj)

[25: Advanced Topics (60)](#_ixi15c65oin1)

[26: Wrap-Up (30)](#_bj2luwrm1zcv)

# 13: Planning (75)

* States, goals and operators
  + Current state vs goal state
  + Weak methods: means end analysis
  + Ladder/Ceiling - paint ceiling first
  + Propositional logic
    - Second goal state: Painted(Ladder)
    - Full goal state: Painted(Ceiling) AND painted(ladder)
  + Initial State
  + Intermediate State: On(Robot, Ladder) ∧ Painted(Ceiling)
  + Operators
    - Preconditions/Postconditions
      * Climb Ladder
      * Descend Ladder
      * Paint Ceiling
      * Preconditions MUST match state in order to be applied
  + Logic provides clarity and precision
  + Only have positive literals in PRE condition
    - It’s okay to have negative literals in POST condition
* Comparing Planning to Search
  + Layer by layer explosion - combinatorial
  + Goals provide “Control Knowledge”
    - Reduce difference between current and goal state
    - How to do operator selection /action selection
  + Mean Ends Analysis
    - Might paint the ladder first - bad! (you have to do the ceiling first.)
    - Means ends goes about choosing actions/operators as if they can be done in any order, this is a problem in some situations
* Partial-order planning
  + Nonlinear planning
  + Use goal knowledge as controlled knowledge
    - Postulates:
      * Controlled knowledge helps us select between operators
      * Goals provide controlled knowledge and can be used to decide between different operators
        + We select an operator that helps us move closer to the goal
      * Partial-order planning viewed as an interaction between several different kinds of agents or abilities:
        + Each agent represents a micro ability; i.e. one agent for:

Generating plans for each of the goals independently

Detecting conflicts between them

Resolving the conflicts

* + - We choose operators whose post conditions match the GOAL and the preconditions match the CURRENT STATE
    - Subgoal
    - View two goals as if they are independent
  + 3 Agents
    - Plans independent for each goal
    - Conflict detection
    - Conflict resolution (reorder plans so goals in conflict are resolved)
* Conflicts in planning: Algorithm for partial order planning
  + **For each** precondition in the current plan:
    - **If** precondition for operator in current plan is ruined by state in other plan:
      * Promote current plan’s goal above the other plan
* Hierarchical task networks (HTN)
  + Complex problems require a large number of operations–HTN abstracts some of these operations at a higher level
  + Macro operations - do preconditions and postconditions
    - Unstack
    - Stack- ascending (i.e. stack blocks in a particular ascending order)

# 

# 14: Understanding (30)

* Analysis Strategies: “Ashok made pancakes for David with a griddle”
  + Lexical Analysis
    - Ashok = noun
    - Made = verb
    - Pancakes = noun
  + Syntactic Analysis
    - Ashok = noun phrase
    - Made pancakes for David with a griddle - verb phrase with sub phrases
  + Semantic Analysis (lexical and syntactic analyses will serve semantic analysis)
    - Ashok = agent
    - Made = action
    - Pancakes = thematic object, thing getting made
    - David = beneficiary
    - Griddle = instrument
* Thematic Roles Systems: a type of frame system where the frame represents an action or event identified by a verb; a verb that causes a number of expectations about the roles that are connected with that particular event
  + Story understanding
    - Earthquake vs proposals
    - We can generate expectations with the general action of a verb
      * I.e. “throw”: expect someone to do the throwing, something to be thrown, target thrown at or to
  + Semantic Categories allow us to draw inferences
    - Agent
    - Verb
    - Thematic object
    - Beneficiary
    - Instrument
* Resolving Ambiguity
  + Certain words or frames can constrain the possible meanings of a sentence and help us figure out those ambiguous meanings
  + Verbs are important - they imply what's happening and what the frame should be
    - Verbs with multiple meanings require background knowledge and context to resolve ambiguity and determine the correct interpretation of the verb
  + Limitations: as we look at increasingly large number of forms of a sentence, the number of rules that we need to disambiguate starts exploding
  + Ontology: conceptualization of the world; the categories and terms of which I specify the world of the AI
    - Vocab of knowledge representation
      * High level are thematic roles
      * Low level are instances of these roles
  + IMAGE GOES HERE
* Leveraging Constraints (grammar)
  + Semantic categories help resolve problems
    - AI can make use of the structure of a sentence to make sense of the story
  + Prepositions: by, for, from, to, with
  + Thematic roles: agent, conveyance, location, beneficiary, duration, source, destination, coagent, instrument
* The cognitive connection (Understanding is about making sense of the world):
  + How do we make sense of the world? There are 3 sources of power:
    - 1st, we exploit the constraints of the world
    - 2nd, we have structured knowledge representations
    - 3rd, low level bottom up processing helps us activate these knowledge structures from memory. Once activated, these knowledge structures generate expectations that make the processing top down.

# 15: Commonsense Reasoning (60)

* Primitive Actions
  + 14 primitive actions - organized into actions and sub actions that can result in changes in the state
  + The structure of the sentence, background knowledge, and context helps determine the real meaning the action
  + Single equivalent representation of many given words; forms an ontology of actions; each primitive action is a frame
    - Move-body-part: move hand
    - Expel: exhale
    - Propel: kick a ball
    - See
    - Smell
    - Move-possession: John took the book from Mary
    - Think-about: forming an idea inside one’s mind
    - Move-object: move car to office, I walked across the room
    - Ingest
    - Speak
    - Hear
    - Feel
    - Move-concept: where Ashok gave the idea to Suzie about going to a movie
    - Conclude: decision-making
  + Common sense reasoning: Implied Actions that capture the meaning of the sentence
    - John fertilized the field
      * Actual verb: fertilized
      * Implied action: put fertilizer (on the field)
    - Bill shot Bob
      * Actual verb: shot
      * Implied action: propelled a bullet (into Bob)
      * Propel is the primitive action here
* Actions/subactions
  + Inferences
  + Microstory
  + Move-object implies move-body-part actions
    - Ashok put the wedge on the block. (move-object)
      * Implies Ashok moved his hand/fingers
  + Computational tractability difficult because number of sub stories can explode
* State changes
  + Result = state change
    - Fields: object and destination
  + State change can be represented in an additional frame connected to the original frame
  + Ashok enjoyed eating the frog.
    - Enjoyed → state change of object (Ashok’s mood) to destination (happy)
* Implied actions
  + DO = unknown primitive, use for generic primitive action
  + Susan comforted Jing.
    - Unclear what Susan actually did or what the primitive action is
    - AI can still use common sense reasoning with generic frame for “do”
* Wrap Up
  + Commonsense reasoning gives us a formal structure to interpret the world around us. This lets us build agents to do the same.
  + Actions and Sub-actions are composed of Primitive Actions
  + These Primitive Actions can cause State Changes (Which let's us predict the effect or cause of certina events)

# 

# 16: Scripts (30)

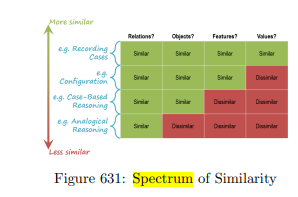
* Defining Scripts
  + Scripts are another Structured Knowledge Representation
  + Stories enable common sense reasoning
  + Scripts help us make sense of complex, repeated events with relative ease and generate expectations about the world around us
  + Scripts are the culmination of frames and understanding and common-sense reasoning
  + Scripts capture 3rd dimension of creativity (unexpectedness)
  + What do you do when visiting a coffee house?
    - Script generates expectations about scenes in the world
  + Definition:
    - A script is/captures causally coherent set of events
      * Causally: Each event sets off or causes the next event.
      * Coherent: The causal connections between events make sense
      * Events: The parts are actions or scenes in the world
* 6 Parts of a Script
  + Entry Conditions
    - Conditions necessary to execute the script
      * Customer is hungry and has some money
  + Result
    - Conditions that will be true after the script has taken place
      * Restaurant owner has more money, customer is pleased, no longer hungry, and has less money
  + Props
    - Objects involved in the execution of the script
      * Tables, menus, food items
  + Roles
    - Agents involved in the execution of the script
      * Customer, restaurant owner, waiter/waitress
  + Track
    - Variations or subclasses of the particular script
      * Tracks for going to a coffee house, fast food restaurant, or fine dining house
  + Scenes
    - Sequence of events that occurs during the execution of the script (kind of like nodes)
      * Scene for entering a restaurant, scene for ordering food, scene for accepting the food, etc.
* Form vs. Content
  + A script is like a class that we can instantiate
  + Primitive actions compose the script
    - Provides the fundamental units of frames that, composed together in some casually coherent sequence, make a script
  + Knowledge structures are composed out of other knowledge structures
    - Frames are composed of primitive actions
    - Scripts are composed of these framelike knowledge structures
  + Help generate expectations without knowing the details
* Generating Expectations
  + Scripts not only allow us to make sense of the world, they also tell us what to expect and what not to expect
* Hierarchies of Scripts
  + Tracks generate a hierarchy of scripts
    - Different dining tracks under restaurant script
* Topics that relate to LEARNING A SCRIPT
  + Semantic networks
  + Frames
  + Incremental Concept Learning
  + Planning
  + Commonsense Reasoning
* Topics that relate to USING a script
  + Problem Reduction
  + Classification
  + Logic
  + Understanding
* Scripts are ***not*** related to problem-solving methods like Generate & Test or Means-Ends Analysis because:
  + Scripts are often used when we already have a solution and we simply need to execute it
* Scripts serve as an **abstraction** over a number of instances
* Script-based reasoning, and cased-based reasoning, are both very memory intensive (memory often supplies most of the answer)
* Scripts are ways of making sense of complex events in the world (by understanding stories in the natural world).

# 17: Explanation-Based Learning (45)

* Intro
  + **Does not learn about new** concepts, rather learns about **connections between existing concepts**
  + Transfer of knowledge from an old situation to a new situation: transporting soup example
    - The pitcher, not the backpack, car, or box
  + Example of incremental learning, learning 1 step at a time
    - Similar to chunking
  + Related to creativity - novel situation
  + No more cups example: what can a robot use if cups are all dirty?
    - Find something that it can explain to be an instance of a cup
* Concept Space
  + Is a space of information that enables us to draw inferences and connections about existing concepts
  + Prior Knowledge is mapped onto this concept space for new reasoning
    - Having several definitions of concepts semantically
    - Causal relationships help us determine what's relevant
      * Abstract from prior knowledge only the things that are causally related
    - We connect structural features to functional features (yellow line in diagram denotes casual connections) to describe causal relationships
* Abstraction
  + Involves how we may abstract over our prior knowledge to discern transferable "nuggets"/parts
  + The explanation here is a causal explanation (simple features that have no causal relationships with other things are dropped).
  + Causal relationships are used to create abstractions of prior knowledge
    - Object CARRIES liquid
      * From object HAS concavity
    - Object IS liftable
      * From object IS light
    - Therefore object enables drinking
  + Object is a cup if STABLE and ENABLES DRINKING
* Analogical Transfer
  + Here is where the AI builds an explanation for the new concept using prior object determined properties to determine if the object is an instance of said concept
  + The "how" in how we transfer those "transferrable nuggets/parts" during the Abstraction part onto the new problem
  + Causal chains are connected together into a more complex causal explanation i.e. for why an object is a cup
  + Agent works backwards → Top down to lowest level causal conditions
  + In order for an object to be a cup, it must be stable and enable drinking. Store those goals into memory and try to prove them for the object.
  + Related to previous lessons: Problem Reduction and Planning
* Improvisation
  + We use explanation based learning to improvise
    - Using cup as paperweight or using an eraser/chair as a doorstop
  + Speed Up Learning
    - Humans constantly come up with creative solutions to novel situations
      * Using existing concepts in new ways
    - Find new connections between existing concepts by building explanations for them
* Generating explanations can lead to much deeper and richer understanding and learning because it exposes the casual connections
* Comes up in Learning by Correcting MIstakes and Diagnosis

# 18: Analogical Reasoning (60)

1. Preview
   1. Understanding new problems from known problems
   2. Analogical Reasoning is inherently transfer-based
   3. Lesson Preview
      1. Similarity and case-based reasoning
      2. Process of analogical reasoning
      3. Design by analogy
2. Exercise: Similarity Ratings
   1. A woman is climbing a ladder (example)
      1. A woman climbing a step ladder
      2. A woman climbing a set of stairs
      3. An an walking up the wall
      4. A woman climbing the corporate ladder
      5. A plane taking off into the sky
      6. A woman painting a ladder
      7. A water bottle sitting on a desk
   2. Similarities can be ranked based on background knowledge, around:
      1. Dimension of relationships
      2. Dimensions of objects
      3. Dimensions of features of objects
      4. Dimensions of values of features of objects that are participating in relationships
3. Cases Revisited: note that new problem and source case were in the same domain
   1. K-Nearest Neighbor (from Learning by Recording Cases)
   2. Array method of organizing cases (from Case-Based Reasoning)
   3. Discrimination Tree (from Case-Based Reasoning)
4. Need for Cross-Domain Analogy (where the new problem and source case are from different domains)
   1. Patient laser gun: tumor will be killed but so will the patient
   2. Fort surrounded by mines
      1. Army split into smaller units to avoid mines
   3. In both cases, a resource was decomposed into smaller resources and sent to the goal at the same time
      1. In cross-domain analogy, the objects and the features/values of the objects can be different
      2. It is the relationship that gets transferred from the source case to the target problem
      3. Recording Cases and Case-Based Reasoning are examples of Analogical Reasoning, except that the target and source cases occur in the same domain.
5. Spectrum of Similarity
   1. Relations
   2. Objects
   3. Features
   4. Values
   5. Get Picture of the chart!



1. Process of Analogical Reasoning
   1. Analogical Process
      1. Retrieval
      2. Mapping: map conceptual relationship to target problem with source case
      3. Transfer: abstract those relationships in the source case and transfer them to the target problem
      4. Evaluation
      5. Storage
   2. Case-based
      1. Retrieval
      2. Adaptation
      3. Evaluation
      4. Storage
2. Analogical Retrieval
   1. Different domains
      1. Superficial Similarity (Unary Relationship)
         1. Features
         2. Counts
         3. Objects
      2. Deep Similarity
         1. Relationshipsbetween objects (Binary Relationship)
            1. Relationship between the objects is similar
         2. Relationships between relationships (Tertiary or Higher Order Relationship)
            1. Relationship between the relationship between objects is similar
3. Three Types of Similarity
   1. Semantic
      1. Conceptual similarity between the target and the source problem
   2. Pragmatic
      1. Similarity of external factors, such as similarity of goals
   3. Structural
      1. Similarity between representational structures
      2. Captures relational similarity
4. Analogical Mapping
   1. Correspondence Problem
      1. What from the target problem corresponds to what in the source case?
      2. If we can answer this question, we can align the target problem with the source case so it makes the deep similarities between relationships stand out
5. Analogical Transfer
   1. Transfer strategy by using correct mapping and pattern of relationships between problems
6. Evaluation and Storage in Analogical Reasoning
   1. Very similar to evaluation and storage steps in Case-Based Reasoning
   2. A solution needs to be evaluated by doing a simulation to see if it is successful
      1. If successful, encapsulate the target problem and proposed solution as a case and store in memory
      2. Could potentially be a source case for a new target problem later on
         1. AI agent learns incrementally
7. Design by Analogy
   1. Bullet Train example: inspired by the shape of the beak of the Kingfisher
   2. Biologically inspired design
      1. There’s a target problem, source case, and cross-domain analogical transfer
8. Design by Retrieval
   1. Basilisk Lizard informs design of robot that walks on water
   2. Structure Behavior Function Model
      1. Represented as a series of states and transitions between those states
9. Design by Analogy: Mapping & Transfer
   1. Align structural models: based on the similarity between relationships
   2. Compositional Analogy
      1. Structural model first needs to be in place, then transport the:
      2. Behavioral model
      3. Functional model
   3. Might have loops
      1. For example map->transfer; transfer leads to more mapping and so on
   4. Biologically inspired design
      1. Robot swimming underwater stealthily example
         1. Part of the design comes from a copepod, other part comes from a squid
      2. Compound analogy
         1. Borrowing parts of the design from multiple source cases rather than just one source case
         2. Non-linear process: can iterate back and forth between Transfer and Retrieval
10. Design by Analogy: Evaluation & Storage
    1. Evaluation can go back to Transfer, Mapping, or Retrieval
       1. Evaluations of simulated solutions can fail
11. Advanced & Open Issues in Analogy
    1. Common Vocabulary
       1. I.e. Different words to describe the same relationship can cause alignment problems
    2. Abstraction and Transformation
    3. Compound and Compositional Analogies
    4. Visuospatial Analogies
    5. Conceptual Combination

# 19: Version Spaces (60)

* Intro
  + Generalization and Learning
    - Background knowledge may not be available as it was in Incremental Concept Learning.
    - May have no control over order in which examples arrive
* Definition
  + Version Spaces is the technique of learning concepts incrementally, learning from a small set of examples with one example at a time.
  + Incremental Concept Learning Revisited
    - Current Concept
    - Positive/Negative example
      * Negative → specialization to exclude it
      * Positive → generalization to include it
    - New concept is a result
    - Ordering was very important
      * We want the examples to come in the right order
      * Want 1 important difference from current concept to arrive at new definition
      * Good teacher is important because they give us the right order
        + What happens if we tell how far to generalize??
        + Or what if the examples come in the wrong order
      * **Version Spaces is a technique that under certain conditions allows the agent to converge on the right level of generalization**
* Abstract Version Spaces
  + Start with two models: Specific model and General Model
    - As positive examples come we **generalize** the specific model
    - As negative examples come we **specialize** the general model
  + We will prune the models, as the examples come, if we get the right ordering of examples
  + The two SEPARATE models (general and specific) MAY merge (under the right conditions) and we have a solution
  + Concepts vs Models
    - Prototypical concepts are models
      * Model is a 1-to-1 correspondence to what is being represented in the world and the representation itself
  + Examples
    - Allergic to cheap meals at Sam’s
* Algorithm for Version Spaces
  + Guarantees Convergence with enough examples
    - Doesn’t have to be in the right order
    - Doesn’t need background knowledge
    - Examples can be different in many ways
  + FOR EACH example:
    - If example is positive:
      * Generalize all specific models to include it
      * Prune general models that can’t include it
    - If example is negative:
      * Specialize all general models to exclude it
      * Prune specific models that cannot include it
    - Prune any models subsumed (closer to root) by other models
      * If there is a node that lies on a pathway from the most general concept that is subsumed by a node from another pathway from the same general concept, then prune that particular node because that node is subsumed by this node.
* Identification Trees (Decision Trees)
  + Discrimination Trees (map example) - similar to incremental discrimination tree in case based reasoning
    - Provides no guarantee of OPTIMALITY of paths
  + New Identification tree is great for optimality
    - Cost of new approach is you need to give all examples at the beginning
    - Branching happens on decisive features (restaurant, meal, day, cost, etc)
      * One category contains either only false or true instances
    - If the number of examples or number of features were very large, it’s hard to decide on the feature we should use to discriminate on
    - OPTIMAL trees split on features that result in each branch consisting of ONLY positive or ONLY negative instances
  + Decision Tree Learning leads to more optimal classification trees but you need all examples up front
  + Discrimination Tree (from Case-Based Reasoning) may lead to suboptimal (bushier) trees, but you can learn one example at a time

# 20: Constraint Propagation (45)

* Intro
  + Constraint propagation
  + Mechanism/method of inference- agent assigns values to variables to satisfy certain conditions (constraints)
  + Very common method in Knowledge-based AI like in Planning, Understanding, Natural Language Processing, and Visuospatial Reasoning
* Definition
  + A method of inference that assigns values to variables characterizing a problem in such a way that some conditions (called constraints) are satisfied
  + Given a problem, that problem will be characterized by a set of variables
    - Task: Assign specific values to each variable such that some global constraints are satisfied
* Constraint processing
  + Examine structure of sentences to see if they fulfill certain conditions or constraints expected from the rules of English language grammar
  + “Colorless green ideas sleep furiously.”
    - Variables = lexical categories like nouns, predicates
    - Constraints = rules of English language grammar
    - When assigning values to the variables, the assignment must satisfy the constraints before the sentence can be accepted as grammatically correct
* Constraint propagation doesn’t necessarily always succeed in ambiguating between different kinds of assignments
  + Multiple interpretations can simultaneously satisfy all the constraints
  + Also possible that no assignment of values with variables will satisfy all the constraints → interpretation becomes very difficult
* Image Processing
  + Pixels to 3D
    - Task Decomposition
      * Similar to a big task of Understanding got decomposed into a series of smaller tasks
      * Problem Reduction as a general purpose method for decomposing complex tasks into smaller tasks
  + Line Labelling
    - Cube with junctions - Constraints
      * Y, fold,fold fold
      * W, fold, blade, blade
      * L, Blade, blade
      * T, blade, fold, blade
    - Fold: a line where two surfaces meet
    - Blade: a line where we cannot infer that two surfaces are getting connected with each other
    - Simple ontologies can cause problems with certain examples of propagation
      * This comes with problems where some constraints are ambiguous
        + Resolve with abduction to disambiguate

Try to come up with the best explanation for data

* Natural Language Understanding
  + Parse tree: demarcate sentence lexically into the noun phrase and verb phrase
  + Constraint propagation is general - can be a basis for semantic understanding
  + Derive a parse tree for the sentence to support semantic analysis to build a semantic understanding of the sentence

# 21: Configuration (45)

1. Preview
   1. Routine Design task in which all the components of the design are already known
   2. Task is to assign values to the variables of those components so that they can be arranged according to some constraints
   3. First part of unit on design and creativity
   4. Configuration is a type of constraint propagation. It is connected to Case-based reasoning, Planning, and Classification.
2. Define Design
   1. Design in general takes as input some sort of needs or goals or functions
   2. It gives as output the specification of the structure of some artifact that satisfies those needs and goals and functions
   3. In design, the problem evolves as the solution evolves
3. Exercise: Designing a Basement
   1. Assign requirements by assigning values to variables
   2. Take the verbal requirements and map to variables
   3. Can uses different heuristics to order variables
4. Define Configuration
   1. **Configuration: A problem-solving activity that assigns values to variables to satisfy constraints.**
   2. In design in the beginning don’t know all the components or arrangements of the components vs. in configuration all the components are known, but **need to determine an arrangement that will satisfy some constraints**
      1. Assigning values to specific variables of those components
   3. Example: furniture in the current room.
      1. All components, the furniture are known
      2. The room is fixed
      3. Example constraint couch must be in front of TV and a certain number of feet away from it
      4. Example: Settings of phone camera
      5. Configuration is the most common kind of design
5. The Configuration Process
   1. Specification Space: specifications of all the constraints of the configuration problem
   2. Configuration Space
      1. Abstraction and Partial Solutions
         1. Design plan where each plan specifies a subset of all the variables and we assign values to those variables
      2. Expanded and Refined Solutions
         1. Iterate through specifying more variables and assigning values until arrangement model is complete
      3. Abstraction Hierarchy
         1. Diagrammatic representation of the plan’s arrangement from abstractions; starts with most abstract plan, refine and expand
      4. Component Hierarchy
         1. All components are already known
         2. Might also be able to select components
      5. Arrangement Model: output; a model of the arrangement of all the components with the components already known
         1. Once process has yielded arrangement model can assess the model and if necessary go back to the process
         2. TV is 12 feet from the couch but then assess that its too close, can go back and add constraint to make TV farther than 12 feet
      6. **In Problem Solving, the problem remains fixed while we come up with solution**
      7. **In Design, the problem evolves as the solution evolves** - problem/solution co-evolve
6. Chair Example: Applying a Constraint
   1. Configuration design is closely related to the method of constraint propagation
      1. Constraints are propagated downwards in the plan abstraction hierarchy
   2. Use of heuristics = rule of thumb
7. Connection to Classification
   1. Similarities
      1. Classification and configuration are both hierarchical
      2. Establish and Define in Classification is similar to Refine and Expand in Configuration
      3. Configuration leverages Classification’s notion of prototype concepts
   2. Differences
      1. Mapping
      2. Configuration = to act on the world by designing actions
      3. Classification = making sense of the world
      4. Configuration = creating the world
      5. Classification = perceive details of the world and decide what they are
      6. Configuration = given something to create and decide on individual variables
8. Contrast with Case-Based Reasoning
   1. Both methods are typically applied to routine design problems that we’ve encountered in the past
   2. Configuration
      1. Starts with a prototype concept and then assigns values to all the variables
      2. Assumes we already designed enough chairs so that we can abstract the plan hierarchy and start defining plans
      3. Calls upon the plan abstraction hierarchy and start defining plans
   3. Case-based Reasoning
      1. Starts from a specific chair we had designed earlier, looks at its variables, and tweaks it as needed to satisfy the constraint of the current problem
      2. Assumes we already designed other chairs and have stored examples of the designs in memory
      3. Goes into the case memory, retrieves the closest matching case, then starts tweaking the case
9. Connection to Planning
   1. In planning, a plan in the plan abstraction hierarchy drops the values of the variables and is converted into a skeletal plan that only has the variables specified
      1. Process of Configuration Planning instantiates these plans and refines and expands them
   2. The result of a planning task can lead to a prototype that can subsequently be configured for similar problems with differing constraints
   3. Describe what we need to learn before deciding on what is a good learning method
      1. Configuration Process tells us of the different kinds of knowledge that in turn become targets of learning

# 22: Diagnosis (45)

1. Exercise: Diagnosing Illness
   1. Coverage - the diagnostic conclusion actually accounts for all input data
   2. Parsimony - want simple hypothesis for explaining all of the data
   3. Hypothesis can have various interactions between them
   4. Explanation - set a hypothesis that can explain the data
2. Defining Diagnosis
   1. Identification of the fault(s) responsible for a malfunctioning system
      1. To determine what is wrong with a malfunctioning device
   2. Diagnosis is an instance of abduction. It is also a task. We can use several methods to address it (like Case-based reasoning)
   3. Similar to Configuration in reverse
   4. Observed and expected behavior, discrepancy = malfunction
   5. Fault - responsible for malfunction
   6. Example of three methods for doing diagnosis:
      1. Car Engine: Rule based reasoning
         1. Use production system to isolate faults responsible for malfunctions
      2. Computer fan: Case based reasoning
      3. Writing a program: Model based reasoning
         1. Rubber duck debugging - explaining model of how program works to a rubber duck
3. Data Space and Hypothesis Space
   1. Diagnosis is mapping data space to hypothesis space. We want a hypothesis that is as refined as possible and explains all available data
      1. Mapping can be M to N
   2. Diagnosis:
      1. An example of a specific data point is that Ashok's temperature is 104 degrees fahrenheit
      2. An example of an abstraction of the Data Space is Ashok is running a fever
   3. Data Space
      1. Contains data about the malfunctioning system
      2. Data can be either very specific (Ashok has a temperature of 104 degrees) or very abstract (Ashok is running a fever)
   4. Hypothesis Space
      1. The Hypothesis space consists of all hypotheses that can explain parts of the observed data (Ex. a hypothesis space of diseases)
      2. A specific hypothesis in this space can explain some part of the data (In medicine, a hypothesis could refer to disease. Ex. Ashok is suffering from the flu, thus it explains his high fever (data point)
   5. Bottom Up Process of Classification
      1. Start with raw data then group and abstract them
      2. This abstract data can be mapped into an abstract hypothesis
   6. Top Down Process of Classification
      1. Start with high level class, then establish and refine them
      2. This abstract hypothesis can be refined to one that explains all the available data
   7. Heuristic Classification
      1. Similar to rule-based reasoning, case-based reasoning, and model-based reasoning with a diagnostic task
      2. Consists of the method of bottom up classification in the data space, mapping into the hypothesis space, and top down classification in the hypothesis space
4. Problems with Diagnosis as Classification
   1. One data point might be explained by multiple hypotheses
   2. One hypothesis may explain multiple sets of data (Diagnosis is also the mapping of hypothesis space to data space)
   3. Multiple hypotheses may explain multiple sets of data
      1. One hypothesis may explain multiple data items and multiple hypotheses may explain the same data item
      2. M to M mapping
   4. Mutually exclusive hypotheses: hypotheses can interact with each other
      1. If one hypothesis is present, another hypothesis cannot be true
   5. Interacting Data Points (Cancellation)
      1. When two hypotheses interact relative to a particular data item
5. Deduction-Induction-Abduction (Three of the fundamental forms of inference)
   1. Deduction: given rule (general principle) and the cause, deduce the effect (conclusion)
      1. Truth Preserving
         1. If rule is true and cause is true, then we can always guarantee that the effect is true as well
   2. Induction: Given a cause and an effect (observation), induce a rule (general principle)
      1. Not truth preserving
         1. Does not always guarantee correctness
   3. Abduction: Given a rule and an effect, abduce a cause
      1. Not truth preserving
         1. Can have multiple causes for the same effect or multiple hypotheses for the same data
         2. For example if you have fever, does not necessarily mean you have the flu
   4. Examples of Rules, Causes, and Effects
      1. Rule: If it’s cloudy then it rains
         1. Cause: It is cloudy
         2. Effect: It rains
      2. Rule: Bob hates Joe
         1. Cause: Joe walks in
         2. Effect: Bob leaves
6. Criteria for Choosing a Hypothesis
   1. Explanatory Coverage: Hypotheses must cover as much of the data as possible
   2. Principle of Parsimony: The smallest number of hypotheses ought to be used
   3. Some hypotheses may be more likely than others (higher confidence level)
7. Completing the Process
   1. Once we have some hypotheses for explaining the available data, then these hypotheses become indices to treatment plans
   2. Map the hypothesis space to the treatment space
   3. This last phase can be thought of as a type of Configuration

# 23: Learning by Correcting Mistakes (45)

* Learning by Correcting mistakes is an explanation-based reasoning process, one of the key concepts differentiating Knowledge-Based AI from different schools of AI.
* Identify three questions for learning from mistakes.
  + How can the agent isolate the error in its former model? (Identifying the fault)
  + How can the agent explain the problem that led to the error? (Explaining how the fault led to the failure)
  + How can the agent repair the model to prevent the error from recurring? (How can the AI agent repair the fault in order to prevent the error/failure from recurring)
* Credit/Blame Assignment
  + Identifying the error in one’s knowledge that led to a failure; what was the knowledge gap that was responsible for the failure?
  + In general, the error could be in one’s reasoning or architecture
  + Applies to all different kinds of errors; some consider it to be the central problem in learning
* It is also important to note that even if an agent was perfect, correcting mistakes is still needed because **agents exist in dynamic worlds**
  + The changes in the world should inform the agent. So the agent must be able to correct itself (self-diagnose, self-repair)
  + As the world changes, the agent will start failing and must have the ability of correcting its own knowledge/reasoning/architecture
* Error Detection
  + Example: Venn Diagram of a positive and negative example of a cup
    - False suspicious feature: characterizes only the negative example
      * These are the feature(s) that caused the agent to incorrectly classify the negative example as positive (failed to recognize the red pail was not a cup)
    - True suspicious feature: characterizes only the positive example
      * These are the feature(s) that caused the agent to incorrectly classify the positive example as negative (failed to recognize that the blue cup was in fact a cup)

1. "Mistake-based" learning touches on Version Spaces and Incremental Concept Learning
2. False success
   1. An object that was identified as a success but was false
3. True success
   1. An object that the agent classified as a success and indeed was a success
4. **Error detection algorithms** are denoted by the following heuristics

* To find suspicious true-success relationships
  + Intersect all true successes(∩T)
  + Union all false successes (∪F)
  + Remove assertions in union from intersection (∩T - ∪F)
* To find suspicious false-success relations:
  + Intersect all false successes(∩F)
  + Union all true successes (∪T)
  + Remove assertions in union from intersection (∩F - ∪T)
* Correcting the definition of an object requires an explanation
  + Leads to deeper learning to be able to explain why a fault in its knowledge led to a failure
  + Classification is used in many schools of AI but Explanation is a key characteristic of knowledge-based AI

# 24: Meta-Reasoning (30)

1. Preview
   1. Meta Reasoning = How do you know that you don’t know?
2. Mistakes in Reasoning and Learning
   1. Block world problem agent tries to figure out error in reasoning
   2. Agent reflects on explanation based learning to find what it did wrong
   3. What was wrong in my process of explanation based learning that elected this incorrect explanation and how do I fix my process of explanation based learning so that I do not make the same error again
3. Beyond Mistakes: Knowledge Gaps
   1. Knowledge can be incomplete
   2. Once agent detects a knowledge gap, it can set a learning goal
   3. Spawning learning goals and finding ways of satisfying or achieving learning goals is another aspect of metacognition
4. The Blurred Line Between Cognition and Metacognition
   1. There is a lot of overlap here so it’s best to think of this as a cognition space, with sub-boxes for meta and deliberation spaces.
   2. The main takeaway is, what is the content of knowledge needed to carry out a process, and what is the process we need to carry out.
5. Strategy Selection
   1. Each method requires some knowledge of the world
   2. Depending on what knowledge is exactly available for addressing that specific input
   3. Agent can have multiple methods to choose from
   4. Computational efficiency is one factor to consider in selecting a strategy
6. Strategy Integration
   1. As problem evolves agent may decide to shift from one strategy to another
7. Process of Meta-Reasoning
   1. Cased based reasoning at a meta level
      1. Tries to see if problem has used a similar strategy before
   2. Planning at the meta level
      1. Use strategies as operators
   3. Metacognition can use same reasoning strategies as the deliberation level
8. Discussion: Meta-Meta-Reasoning
   1. Each level of metacognition is conceptually identical, so they are better represented as self-referential
   2. Metacognition reasons over itself recursively
9. Example of Meta-Reasoning: Goal-Based Autonomy
   1. Meta reasoning is circular
   2. Example: Robot assembles camera and given new goal to disassemble
10. Connections
    1. Basically whenever the agent is reflecting its metacognition
       1. Reflecting on mistakes
       2. Reflecting on learning process
       3. Reflecting on plans, resolving conflicts
       4. Production Systems: impasse -> new learning goal
       5. Version spaces
          1. Agent thinks about its own specific/general model and looking for opportunities to converge into one model
       6. Diagnosis
          1. If treatment doesn’t spawn desired results it produces information for meta layer
          2. Diagnose the problem with diagnosis process
11. Meta-Reasoning in CS 7637
    1. By building an agent to solve a problem we gain understanding of how we solve the problem (super meta)
12. The Cognitive Connection
    1. Learning how to learn ( YAAAY Metareasoning)

# 25: Advanced Topics (60)

Visuospatial reasoning (Constraint Propagation -> Visuospatial reasoning)

* Visuospatial reasoning is reasoning with visual spatial knowledge
  + Visual: The "what" part
  + Spatial: The "where" part
* Example: A picture of the sun in the top right
  + Visual: The Sun (the "what", the "object)
  + Spatial: The top right of the picture (the "where")
* Visuospatial Knowledge: Knowledge wherein causality is, at most, implicit. Causality is implicit when it enables inferences about causality.
* Two Views of Reasoning
  + Propositional Representations
    - Can work on image to produce new proposition representation
    - Amodal representation
  + Analogical Representations
    - Structural Correlation between representation and external figure
    - Modal representations. Close to perceptual modality
  + Most computers can only do proposition representations
* Symbol Grounding Problem
  + Visuospatial
    - Content
      * Appearance: What and where
  + Verbal
    - Content
      * Arbitrary: Driven by Inferential Needs
  + Analogical
    - Encoding: Structural Correspondence
  + Propositional
    - No correspondence
  + Reasoning and knowledge can be visuospatial and representations can be analogical
  + Have yet to build AI agents that can deal with visuospatial knowledge and analogical representations
* Visuospatial Reasoning: An Example
  + Galatea (AI Agent developed at GT)
  + Dunker Radiation Problem
    - First give story, then try to find answer to problem
    - Usually we try to extract casual pattern then apply it to the problem
    - Example of overthrowing a bad king in a castle surrounded by landmines. Army is split into small groups to not trigger the mine
    - Transfer the above story to solving the problem of surgery with a laser to remove a tumor. Laser is too strong and will kill the tumor as well as healthy tissue. Solve by splitting into smaller lasers
    - Galatea used visuospatial knowledge instead of casual pattern to solve the Dunker Radiation problem
* Visuospatial Reasoning: Another Example
  + Cylinder and Piston
  + Feed agent pictures and given new drawing it could assemble a causal functional model for the new drawing
    - Able to extract causal information from spatial presentations to analogical reasoning
* Raven’s Progressive Matrices
  + Used to only extract propositional representations
  + GT Students were able to use only analogical representations to solve
* Systems Thinking: Introduction
  + Reasoning about systems with numerous components and processes at multiple, potentially invisible, levels of abstraction
* Systems Thinking: Connections
  + Diagnosis is an example of systems thinking
  + Debugging code is hard because lots of lines of code interact at a bunch of complex levels (\*surprised pikachu\*)
  + Helps to understand the invisible levels of abstraction
* Structure-Behavior-Function
  + Capture invisible structure as well as invisible causal processes
  + Capture multiple levels of abstractions
* Design: Introduction
  + Reasoning about ill-defined, unconstrained, open problems that are situated in the world
* Agents Doing Design
  + Planned refinement
  + Flashlight Example
  + Every time you create a design every experience is an opportunity for learning
  + Ideal Program
    - Learned design patterns form simple design cases
    - Transferred electrical circuit knowledge to water pump problem (Example of different domain transfer)
* Creativity: Introduction
* Exercise: Defining Creativity I
  + What is creativity?
  + A non-obvious desirable product
* Exercise: Defining Creativity II
  + Novel
  + Useful or Valuable in some way
  + Unexpected or surprising
  + Novel is newness, unexpected is surprising
  + Some processes of creativity
    - Emergence
      * Draw 3 lines and a triangle emerges
    - Re-representation
      * Occurs when the original representation of the problem is not conducive to problem solving
    - Serendipity
      * Trying to find solution but can’t find one - Suspended goal - start doing something different- come back when relevant solution presents itself in different context
* Analogical reasoning is fundamental to creativity
* Explanation based learning can be relevant to creativity

# 

# 26: Wrap-Up (30)

* Major Topics
  + Architecture for cognitive system
    - Metacognition
    - Deliberation
      * Maps percepts into actions, with the mappings mediated by reasoning, learning and memory
    - Reaction
      * Maps precepts in the environment into actions in the environment
    - Overlapping systems that get percepts/inputs, agent may react directly, deliberate or metacognition can occur and act on the internal or external system
  + The world and Cognitive systems are input/outputs for each other
  + Cognitive systems are not just situated in the physical world, but also the social world - built out of other cognitive systems
* Course Topics
  + Fundamentals
    - Semantic Networks (Knowledge Representation)
    - Generate and Test, Means End Analysis, Problem Reduction (Problem Solving methods)
    - Production Systems (specific kind of Cognitive Architecture) combines reasoning, learning, and memory.
  + Planning
    - Logic (Formal/Propositional logic, syntax)
    - Planning (Origins of planning were proofs of why actions would lead to goals, Partial Order Planning, Hierarchical Planning)
  + Common Sense Reasoning
    - Frames
    - Understanding
    - Common sense reasoning → Scripts
  + Learning
    - Recording Cases
    - Incremental Concept learning
      * Classification
      * Version spaces
  + Analogical Reasoning
    - Recording Cases
    - Case based reasoning
    - Explanation Based Learning
  + Visuospatial Reasoning
    - Constraint Propagation
  + Design & Creativity
    - Configuration
    - Diagnosis
    - Design
    - Creativity
  + Metacognition
    - Learning by correcting mistakes
    - Meta-Reasoning
    - Ethics
* Principles
  + 1 - KBAI agents **represent** and organize knowledge into knowledge structures to guide and support **reasoning**
    - scripts
  + 2 - Learning in KBAI agents is often **incremental** 
    - Learning by recording cases
    - Concept learning
    - Version spaces
    - Correcting mistakes
  + 3 - Reasoning in KBAI agents is top-down AND bottom-up
    - Frames (top-down)
    - Scripts (top-down)
    - Constraint propagation (top-down)
  + 4 - KBAI agents match **methods** to **tasks**
    - Methods
      * Generate and test
      * Means ends analysis
      * Problem reduction
      * Production Systems
      * Case-based reasoning
      * Planning
      * Analogical Reasoning
    - Tasks
      * Configuration
      * Diagnosis
      * Design
      * Meta-reasoning
      * Creativity
      * Classification
      * Systems Thinking
  + 5 - KBAI agents use heuristics to find solutions that are **good enough (satisficing),** though not necessarily optimal
    - Plan to go from car to office is not optimal but is good enough
    - Exhaustive search is computationally costly
    - Concept learning - heuristics
    - Means end analysis - reduce cost to goal
  + 6 - KBAI Agents make use of **recurring patterns** in the problems they solve
    - Tying shoe laces
    - Case based reasoning
      * Sometimes we have to adapt old problems to new ones
    - Analogical reasoning
      * Domain transfer
    - Configuration
      * Specific chairs may differ
    - Knowledge based AI solve recurring problems based on recurring patterns, leverage recurring patterns to address novel problems
  + 7 - The architecture of KBAI agents enables reasoning, learning, and memory to support and constrain each other
    - Memory **stores** and **organizes** knowledge
    - Learning **acquires** knowledge
    - Reasoning **uses** knowledge
    - Production systems
      * When reasoning fails, memory provide episodic knowledge, chunking extracted rule from knowledge, rule broke the impasse and solved the problem
    - Logic
      * Begin with knowledge based axioms
    - Explanation based learning
      * Memory supplies us with previous instances
    - Learning by correcting mistakes
      * When a failure occurred, the agent reasoned and identified the fault responsible for the failure and corrected that particular fault
* Modern Research
  + Calo
    - Cognitive assistant
  + Cyc and OMCS
    - Common sense database
  + Wolfram Alpha
    - Answer engine
  + VITA
    - Visual thinking in autism
  + Dramatis
    - Computational model of suspense and drama in stories
  + DANE
    - Design by analogy to natural systems