**1/10/18**

In AI, proper formulation supports:

* automatic solving
* computer-assisted manual solving
* problem-space analysis
* visualization
* use of heuristics
* reasoning about the problem

**Steps in Problem Formulation**

* Describing a need
* Identifying resources
* Restriction and simplification
* Designing a state representation
* Designing a set of operators
* Listing constraints and desiderata
* Specifying in code the state representation, operators, constraints, evaluation criteria, and goal criterion.
* Specifying in code a state visualization method.
* If appropriate, providing for multiple roles within teams of solvers.

essentially:

* Preformulation
  + describing need and identifying resources
* Posing
  + constraints, representations, restriction, and simplification
* Coding the Formulation
  + solving problem using code and building off of previous steps

**Eight Puzzle & Fifteen Puzzle**

* 3x3 tray with 8 tiles or 4x4 tray with 15 tiles
* can only move adjacent tiles to an empty slot to the empty slot
  + put the puzzle in order
  + mac game (tile game) (the one with the pictures)

**Definition of Problem (revisited 02-State-Spaces)**

* A problem is a triple (sigma\_0, phi, gamma)
  + sigma\_0 is an initial state
  + phi is a set of operators
  + gamma is a set of goal states
* Each phi\_i \in phi has a precondition, state-transformation function, and optional parameter list
* This implicitly defines \Sigma , the set of all states reachable from sigma\_0 by applying phi\_i \in phi >= 0 more times.

**Eight Puzzle Formulation**

* State: 3x3 array containing 8 tiles
  + tiles represented by 1...8
  + blank represented by 0
* Initial state: a random state, except it must represent an even permutation
  + if it starts with an odd permutation it cannot go to an even permutation (solved)
* Goal State: [[0,1,2][3,4,5][6,7,8]]
* Operators: N,E,W,S
* not all of these choices are definite, specification of problem states , info content, structures, and operators are **design decisions.**

**Coding a Formulation**

* METADATA
  + data about problem
* COMMON\_DATA
  + data that should be available to rest of code
* COMMON\_CODE
  + most of the formulation resides
  + State class
  + Supporting methods
* OPERATORS
  + correspond to components of problem formulation
* INITIAL\_STATE
  + correspond to components of problem formulation
* GOAL\_TEST
  + correspond to components of problem formulation
* STATE\_VIS (not required)
  + not needed (no need for visualizations)

Classes with \_\_eq\_\_ can redefine ==

* \_\_str\_\_ redefines string.
* \_\_hash\_\_ redefines hash function (hash table)
* copy method: makes sure to deep copy a state otherwise there will be bugs