1 Genetic Algorithms (?)

2 Representation

- Reading for this section is Chapter 2.
- In the second lecture, we talked about different ways to represent problems. Today we'll look at representation in a bit more detail.
- Different representations have different strengths/weaknesses.
- Simplest rep. is **atomic** – each state has a unique ID and that ID, in some sense, tells you everything you need to know about the state.
- The graphs you've seen in most of your previous classes are atomic. Each node in the graph is simply a little blob of memory, possibly storing some information. You usually talk about "graph $G = \langle V, E \rangle$" where V is just some arbitrary set of objects, and you are assumed to be able to iterate over V.
- Example: when you're doing Dijkstra's algorithm, the set of states is just the set of locations (i.e., cities) – ABQ, Boston, SF, etc. . . Each one gets its own label, and that's really all you need to know about it.
- So far in *this* class, we've mostly been using *implicit* graphs, where each node is represented in terms of a small *set* of variables, each with a restricted range.
- Example: in the 8-puzzle, you can represent each puzzle state with a vector of 9 elements, each over the range 0-8, with the restriction that all of the elements are different.
- Example 2: in the 8-queens problem, you can represent each state as a vector of 8 variables, one for each column, where the element in the i^{th} column gives the row for the queen (this is the framework that we used when talking about constraint satisfaction last time.)
- The key is that both of these are defined over *finite* domains – each state is described via a finite set of variables, each taking on a finite number of values. The whole state space is, therefore, finite (albeit exponentially large in the number of variables).
- These can be seen as examples of **propositional** representations.
- A bit more formally, the propositional calculus is made up of:
  - A set of **propositional symbols P, Q, R, S, . . .**
  - the values TRUE and FALSE, and the corresponding formal *symbols* for them, **T** and **F**
  - A set of connective symbols