Search

Dr. Melanie Martin
CS 4480
September 12, 2012

Some previous projects

- Twenty Questions
- Library Search Assistant
- Euclid's Game
- FedEx on the Go
- Battleship Game
- Line-Following Robot
- Connect Four
- "Reen"

Some previous projects

- Learning Checkers
- Agent Fred
- Agent Using Genetic Algorithm
- Guess Who
- Color Memory Game
- TicTac Chat
- Eight Queens
- Super Mario Bros. AI

Single-state problem formulation

A problem is defined by four items:

1. initial state e.g., "at Arad"
2. actions or successor function *S(x)* = set of action–state pairs
   - e.g., *S(Arad)* = {<*Arad* → *Zerind, Zerind*>, … }
3. goal test, can be
   - explicit, e.g., *x* = "at Bucharest"
   - implicit, e.g., *Checkmate(x)*
4. path cost (additive)
   - e.g., sum of distances, number of actions executed, etc.
   - *c(x,a,y)* is the step cost, assumed to be ≥ 0

- A solution is a sequence of actions leading from the initial state to a goal state

Selecting a state space

- Real world is absurdly complex
  → state space must be abstracted for problem solving
- (Abstract) state = set of real states
- (Abstract) action = complex combination of real actions
  - e.g., "Arad → Zerind" represents a complex set of possible routes, detours, rest stops, etc.
- For guaranteed realizability, any real state "in Arad" must get to some real state "in Zerind"
- (Abstract) solution =
  - set of real paths that are solutions in the real world
- Each abstract action should be "easier" than the original problem