• Midterm Review
• New Eleusis

AI Class 13

Bookkeeping

- HW3
  - Due date: **10/17** @ 11:59pm
- Midterm is **10/25**, in class
  - Covers **through** multi-agent systems
  - Study material should include all readings!
- Project teams
  - Please fill out Google form (posted on schedule)
  - If you aren't part of a 2-4 person team or would like more members, you must solve it TODAY
  - Official description out today at (by) 11:59pm

Today's Class

- Quick midterm review
- Project overview
- New Eleusis practice
  - The ideal # for New Eleusis is 6 players, so we will combine teams
  - You should play with your team, however.

3

Midterm Review

10/25
10:00am – 11:15 am
ITE 227

What Will it Be Like?

- Broadly:
  - Turn a problem description into a solution
  - Work through a problem to reach a solution
  - Demonstrate a conceptual grasp of the material
- Concepts ←→ Algorithms and Implementations
- Basic idea: you need to **understand the ideas** behind the material we have covered, and be **able to apply it** to solving problems.

What Kind of Questions?

- Homeworks **and lectures** should be good practice
  - Work through an {algorithm | solution type | problem}
  - Draw something – search trees, states, paths through a map, …
  - Write a **short** answer to English questions
    - E.g.: What approach would you use to solve this problem?
    - E.g.: "We know these are independent. Why?"
- (Plus T/F, fill in the blank, etc.)
- Coding questions (not minor syntax mistakes, etc.)

Scoring

- Follow directions.
  - I will mark off for, e.g., short != long
- Start with a perfect score, mark down for mistakes
  - In practice: do not give extra information
  - If I ask for 2 examples, and you give 3, one of which is wrong: 50% on question (marked down for wrong)
- Read carefully.
  - My exams don't tend to be long. You have time.
  - ~~"I didn't see the part that said…"~~

Topics

- What is AI?
- Agents
  - What kinds are there?
  - What do they do?
- Constraint Satisfaction
  - What are constraints? What's a CSP problem?
  - What is a CSP problem? How would you express [*X*] as a CSP?
  - As a search?

Topics

- Search
  - What is it for?
  - What are the elements of a search problem?
  - Uninformed, informed, local
  - What are some high-level approaches to search, generally?
  - What are specific algorithms?
  - What are they good for?

Topics

- Game Playing
  - Why play games?
  - What games, and how?
  - How does it relate to (search, CSP, …)
  - How would you express a game as an [*X*] problem?
- Basic Probability
- Uncertainty
  - Sources of uncertainty
  - Reasoning under uncertainty

Topics

- Probabilistic Reasoning
  - Posteriors and Priors
  - Bayesian Reasoning
  - Induction, Deduction
  - Probabilities of Events
  - [In]dependence, conditionality, marginalization
- Bayesian Networks
  - What are they? How do you solve problems with them?
  - How can you express [*X*] as a network? What does network [*Y*] express?
  - **Inference**
- Multi-Agent Systems

Various Reminders

- **Everything in the readings** is fair game.
- Look at homeworks, sample problems in lectures.
- Look at lectures' "Why?" questions.
- Slides are a good source of **conceptual** understanding.
- Book goes into **detail** and explains more deeply.
- This does not lend itself to cramming.
  - You have an extra week – start!

Project Overview

New Eleusis

- Project: write a **player** for the game of New Eleusis.
- Goal: try to figure out a rule using **induction over data**
- It is a gameified version of the scientific method.
- You play cards in series, which either:
  - Obey the rule
  - Don't obey the rule
  - Either is informative!
- The goal is to figure out the rule first

Project Goals

- Write a player that
  - Takes inputs in a fixed format
  - Hypothesizes a rule describing that input
  - Makes plays intended to test hypotheses about rules
  - Announces the rule when it is successful
- This is an NP-complete problem

Deliverables

- Project Design
  - The names of your team members
  - Short (less than one page) description of your strategies
  - A Python design:
    - Main functions, inputs and outputs for each function, pseudocode/stubs for the behavior of the function
    - Helper functions and computations you will need to implement your planned strategies
    - Corresponding functions to do that work
  - The interface **will be informed by your intended designs**, so give this some thought
- You will inevitably make changes later

Deliverables 2

- An implemented player in 2 phases
- Did you correctly implement the solution that you described in your design?
- Design (generality, clarity, and elegance) and readability (indentation, comments, modularity, …)
- Score will be split among phases, 50/50
- Phases will have simplifications to relax.

Playing New Eleusis

- Martin Gardner wrote about this original version in his Mathematical Games column in the June, 1959 Scientific American.
  - It is a game of **inductive logic**.
  - You use cards to perform **experiments**.
- This is NOT the real thing – the real thing* is more complex, and the project version is simpler.
- The goal of this exercise is to understand what you are supposed to be implementing!

*http://matuszek.org/eleusis0.html

Setup

- Shuffle four(ish) decks together
- Pick a dealer ("Nature"/"God")
  - Nature: make up a rule and **write it down**.
  - The rules can only depend on things **visible on cards**
  - Suit, color, face value, face cards
- Deal 14 cards to everyone else

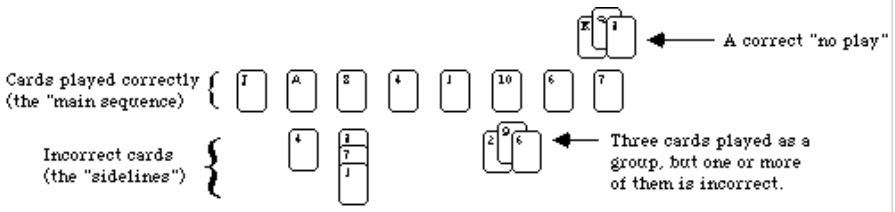
Play

- Nature: play 1 card to start.
- Go around clockwise to each player:
  - Play a card from your hand
  - Nature declares it right (follows the rule) or wrong
  - If it's right, proceed; otherwise, draw 2 cards

Induction

- When you think you have figured out the rule:
  - Declare yourself a "Prophet"
  - Take over calling right and wrong from Nature
    - But DO NOT announce what you think the rule is!
  - If you make a bad call, Nature takes back over; draw 5
  - After 10 successful calls, enter "Sudden death": anyone who plays a bad card is out
  - If everyone is out, Nature wins
  - After 20 successful calls, you win

Layout



Some of this refers to the "real" (full) game.