Artificial Intelligence

Tom: You want to do a lecture on AI?
Kevin: Yeah, I think artificial intelligence is great!
Tom: Yeah, I think your intelligence is artificial.
Two Types of AI

• Symbolic AI actually tries to encode knowledge as it is
  – I should play a lightning bolt when a creature with toughness 3 or less attacks

• Statistical AI
  – I should play a lightning bolt when my position is quantitatively better than it would be if I didn’t play it
Symbolic AI

- Most approaches rely upon logic
  - Usually, first-order logic
- As we’ve seen, magic can be stated largely as a series of logical statements
Logical Knowledge Representation

• Also known as KR
• Uses normal logical deduction to “learn” things
• $\forall c \forall p \forall t (\text{Attacking}(c,p) \& \sim\text{Blocked}(c) \rightarrow \text{Life}(p,t+1) = \text{Life}(p,t) - 3)$
  – Subtraction is actually a fn here, not a legal logical operator
How does this become valuable?

• KR allows us to understand situations more completely through inference

• Magic has many inferred concepts and aspects
  – Grizzly Bears that didn’t attack when it is the strongest is probably going to block

• Need to use a bunch of predicates to make our language more expressive
Predicates for our example

- attack(c, t) – creature, turn
- tapped(c, t, p) – creature, turn, phase
- after(x, y) – phase, phase
- For us, all lower-case letters are variables
- Things in quotes will be our contants, though not really
Example truth

• $\forall c \forall t \forall p ((\text{attack}(c, t) \land \text{after}(p, \text{“combat”})) \rightarrow \text{tapped}(c, t, p))$

• $\forall c \forall t \forall p (\neg)$
  • $(\text{attack}(c, t) \land \text{after}(p, \text{“combat”}))$
  • $\rightarrow \text{tapped}(c, t, p))$

• Generally true, and probably an axiom
• Why would this not be true?
• How would we fix that?
How does that help us?

• Assume attack("RagingGoblin","1")
• If we assert $\forall c \forall t \forall p ((\text{attack}(c,t) \land \text{after}(p, "combat")) \rightarrow \text{tapped}(c,t,p))$
• and after("end","combat")
• we know tapped("RagingGoblin","1","end")
The drawbacks to Logic

• Very large predicate and axiom requirements
  – Though not necessarily larger than human knowledge

• Quantitative knowledge is also hard
  – Had to cheat to get subtraction

• Frame problem: Hard to know if unrelated predicates are changing
  – Not so much in magic, but it exists
Why KR is still cool

• It can do a lot of stuff
  – There was a point in which mathematical facts were being generated by computers
• Can learn a lot from very small examples
• More powerful with inconsistent systems
  – Extensible into ML
• It actually makes sense
The next step

• KR is cool, but it doesn’t give us a Magic playing AI
• What more do we need for that?
Cognitive Architectures

• Basic environment or state
• KR or equivalent logical system to find truths in state
• Goals to shoot for
• Actions to achieve goals
Icarus

- One of many existing cognitive architectures
  - Soar, ACT-R, Prodigy
- Began by Pat Langley, now maintained by his lab
- Written entirely in Lisp
Basic flow

• Define environment and goal
• For each cycle
  – Recognize percepts from environment
  – Apply concepts to determine beliefs
  – If goal isn’t satisfied
    • Go to appropriate skill and execute action or subgoals
Cognitive Architecture Recap

- Often requires a lot of hand-coded domain knowledge
- Big goal is to demonstrate cross-domain knowledge
Let’s go statistical

• Look at an example of a statistical approach
• Built upon game theory, graph theory, and computation
Game Trees

• Game trees are directed graphs representing entire games

• Nodes are complete descriptions of positions
  – Includes creatures in play, cards in hand, library arrangement, etc

• Edges are various actions that change the position
  – Drawing a card, attacking
Example tree

Start → Untap → Untapped

Untapped → Draw Card

Untapped → Rishadan Port during upkeep

Tapped Land

Opponent at 10 → Lava Axe

8 Cards in hand → And so on
For some perspective...

• Rubik’s cube is a good way to think about the tree
• Each configuration of the cube is a different position
• Each spin, twist, pull, twirl, bop is an action
• **Total position size?**
So back to total positions

- According to Wikipedia, 3x3 cube has 43,252,003,274,489,856,000 positions.
- About 9000 magic cards in existence right now.
  - For 60 card decks, that's about \(10^{55}\) decks.
  - \(10^{80}\) atoms in the universe.
- Most of them suck, but possible.
- Magic is theoretically infinite, but not really.
So what’s next?

• Game tree is analogous to KR
  – Gives us a way to understand the game, but not a way to play
• How do we turn this into a Magic player?
• Search!
• Really looking for a path from current position to winning position
  – Use your favorite old-fashioned search algo with logic for following edges
Do you chump?

I’m not optimistic about this
What about in finite time?

- Need to use heuristics
- Can look a certain depth away instead of all the way down
- Use an evaluation function to compare the “goodness” of various positions
If we do block...

2 * 0 (for having no grizzly bears)
+ .1 * 20 (for having 20 life)
+ .01 * 44 (cards in library)
+ -.1 * 18 (opponent’s life)
+ 4 * 3 (cards in hand)
+ -3 * 2 (cards in opponent’s hand)
+ 1 * 1 (open forests)
+ .5 * 2 (open mountains)
+ .1 * 2 (cards in graveyard)
+ 1000 * 1 (because psychic battle isn’t in play)
+ -2 * 0 (for number of poison counters on you)
+ S * 3 (for the change in entropy)
+ avagadro's number * 4
+ number of episodes in Avatar: the Last Airbender

+ .... =

Something less than 5
If we don’t block...

2 * 1 (for having 1 grizzly bear)
+ .1 * 18 (for having 18 life)
+ .01 * 44 (cards in library)
+ -.1 * 18 (opponent’s life)
+ 4 * 3 (cards in hand)
+ -3 * 2 (cards in opponent’s hand)
+ 1 * 1 (open forests)
+ .5 * 2 (open mountains)
+ .1 * 1 (cards in graveyard)
+ 1000 * 1 (because psychic battle isn’t in play)
+ -2 * 0 (for number of poison counters on you)
+ S * 3 (for the change in entropy)
+ avagadro’s number * 4
+ number of episodes in Avatar: the Last Airbender

+ \ldots =

Something more than 5
Minimax

• So it’s helpful to do this to the deepest level possible
• Trickier because you need to alternate levels with your opponent
  – Assume opponent picks best move for them each time, and propagate that value upwards
• Can push a value up to your current position
• This is minimax
Minimax in action

• http://www.ocf.berkeley.edu/~yosenl/extras/alphabeta/alphabeta.html
Scenario

TOM
4 Life
0 Cards in hand

ME
2 Life
0 Cards in hand

What should I do?
Endless nonaction

Nothing

Nothing

Tom at 1

Nothi

pop fanatic

ish, if

nothing

nothing

you lose

you lose

Tom at 0

you win

you win
Minimax properties

• Designed for 2 player, 0 sum games
  – Your loss is my victory, vice versa
  – Alternating actions for 2 people

• Complete
  – ie slow as balls
  – What to do?
Alpha-Beta Pruning

- Reduces total search by ignoring unimportant subtrees
- Which subtrees are unimportant?
  - The ones that are worse than ones you’ve already seen
Example

TOM
6 Life
0 Cards in Hand
Piker on top

ME
1 Life
How are you a better player?

• Ignore situations that you know your opponent can play around
• Isolate important vars to determine if a position is better than another
• Think about Magic like a tree of actions
Want more?

• Gavin Verhey mentions that good players keep options open for later
  – Equivalent to picking paths that have multiple, high evaluation nodes

• Perhaps equivalent to constraint-satisfaction problems
  – Forward-checking and least-constraining vars

• General heuristics