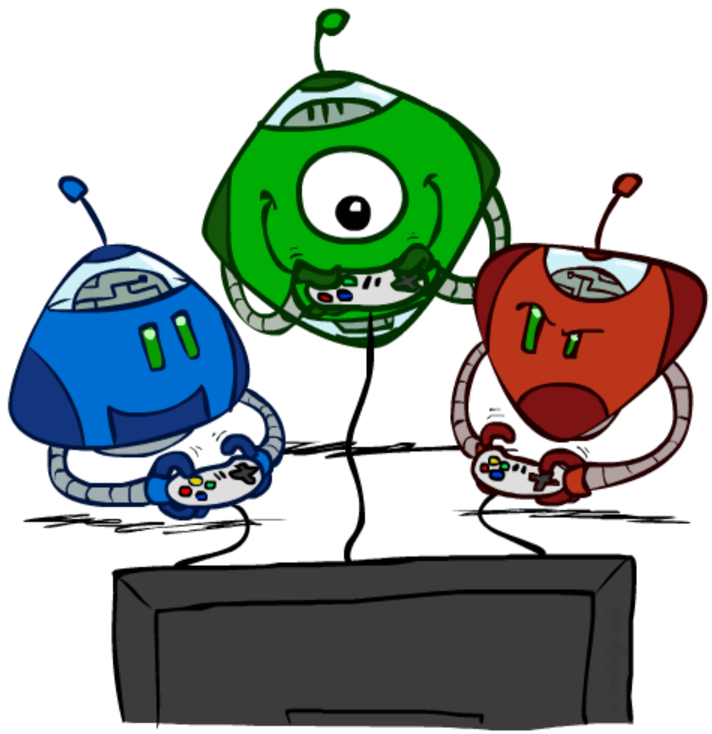
Games: Conclusions

With slides from
Dan Klein, Percy Liang, Luke Zettlemoyer

Project 2: Due Tomorrow Wed 2/15

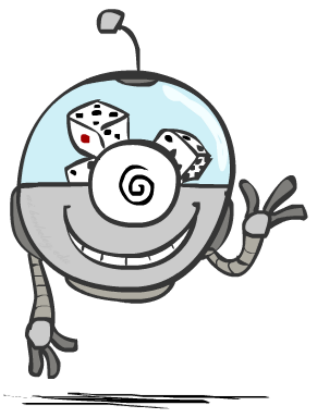
- http://www.mathcs.emory.edu/~eugene/cs325/p2/
  - Use Piazza for discussions
  - Questions?

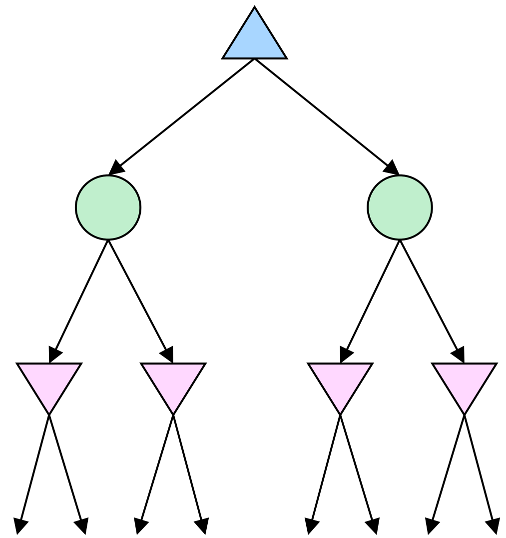
Other Game Types



Mixed Layer Games: Backgammon

- Expectiminimax
  - Environment is an extra “random agent” player that moves after each min/max agent
  - Each node computes the appropriate combination of its children





Example: Backgammon

- Dice rolls increase *b*: 21 possible rolls with 2 dice
  - Backgammon ≈ 20 legal moves
  - Depth 2 = $20 \times (21 \times 20)^3 = 1.2 \times 10^9$
- As depth increases, probability of reaching a given search node shrinks
  - So usefulness of search is diminished
  - So limiting depth is less damaging
  - But pruning is trickier…
- Historic AI: TDGammon uses depth-2 search + very good evaluation function + reinforcement learning: world-champion level play
- 1st AI world champion in any game!

Image: Wikipedia

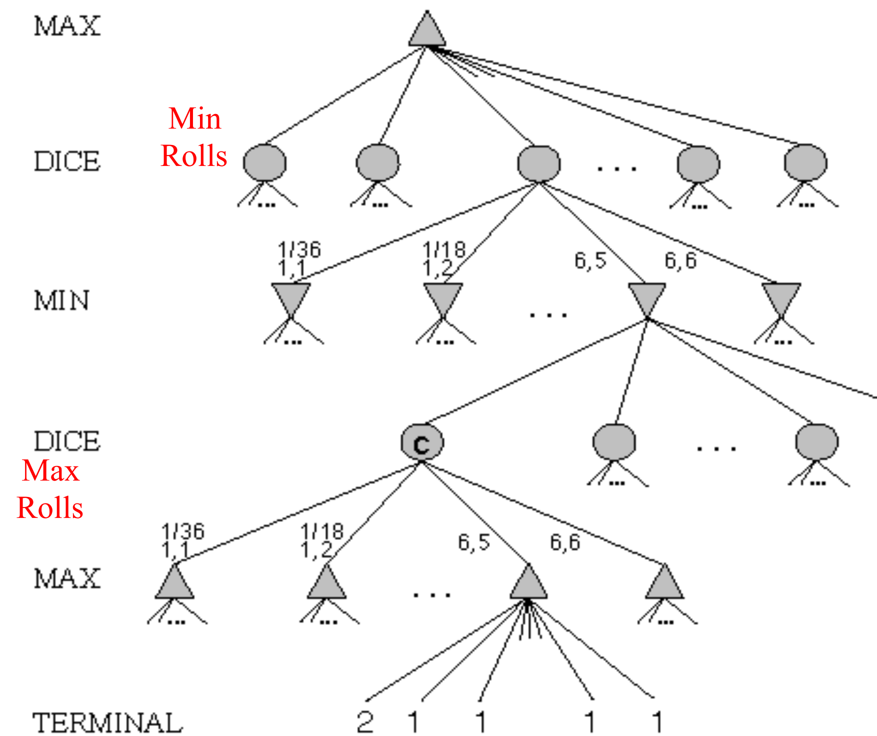
Game trees with chance nodes

- **Chance nodes** (shown as circles) represent random events
- For a random event with N outcomes, a chance node has N children; a probability is associated with each
- 2 dice: 21 distinct outcomes
- Use minimax to compute values for MAX and MIN nodes
- Use **expected values** for chance nodes
- For chance nodes over a max node:

$$\text{expectimax}(C) = \textstyle\sum_i (P(d_i) * \text{maxvalue}(i))$$

- For chance nodes over a min node:

$$\text{expectimin}(C) = \textstyle\sum_i (P(d_i) * \text{minvalue}(i))$$



Impact on Lookahead

- Dice rolls increase branching factor
  - 21 possible rolls with 2 dice
- Backgammon has ~20 legal moves for a given roll

  ~6K with 1-1 roll
- At depth 4 there are 20 * (21 * 20)**3 ≈ 1.2B boards
- As depth increases, probability of reaching a given node shrinks
  - value of lookahead is diminished
  - alpha-beta pruning is much less effective
- TDGammon used depth-2 search + very good static evaluator to achieve world-champion level

Games with **imperfect** information

- Example: card games, where opponent's initial cards are unknown
  - We can calculate a probability for each possible deal
  - Like having one big dice roll at the beginning of the game
- Possible approach: compute minimax value of each action in each deal, then choose the action with highest expected value over all deals
- Special case: if action is optimal for all deals, it's optimal
- GIB, a top bridge program, approximates this idea by

  1) generating 100 deals consistent with bidding information

  2) picking the action that wins most tricks on average

Poker

- Recognized challenge problem in AI
  - Hidden information: other players' cards
  - Uncertainty about future events
  - Deceptive strategies needed in a good player
- Very large game trees
- Texas Hold'em: most popular variant

Texas Hold'em poker

Nature deals 2 cards to each player

Round of betting

Nature deals 3 shared cards

Round of betting

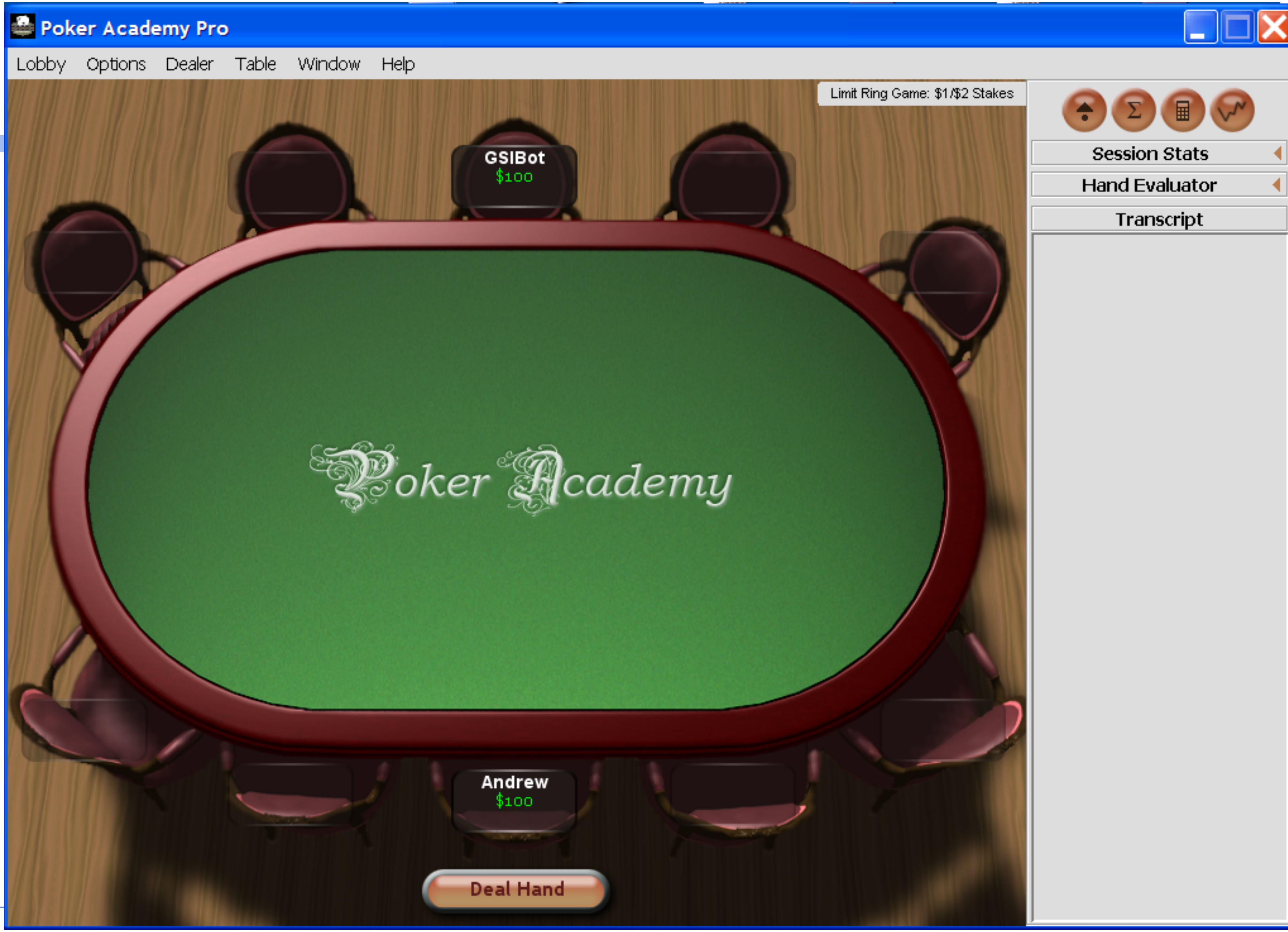
Nature deals 1 shared card

Round of betting

Nature deals 1 shared card

Round of betting

- 2-player Limit Texas Hold'em has ~10^{18} leaves in game tree
- Losslessly abstracted game too big to solve => abstract more => lossy



Poker Academy Pro
Lobby
Options
Dealer
Table
Window
Help
Limit Ring Game: $1/$2 Stakes
Session Stats
Hand Evaluator
Transcript
GSIBot
$100
Poker Academy
Andrew
$100
Deal Hand


Poker Academy Pro
Lobby
Options
Dealer
Table
Window
Help
Limit Ring Game: $1/$2 Stakes
Session Stats
Hand Evaluator
Transcript
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
GSIBot
$99
Call $0.50
Pot: $2
Poker Academy
Andrew
$99
Fold
Check
Bet $1


Poker Academy Pro
Lobby Options Dealer Table Window Help
Limit Ring Game: $1/$2 Stakes
Session Stats
Hand Evaluator
Transcript
GSIBot
$99
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
Andrew checks
FLOP: Qd 7s 4c
Pot: $2
Andrew
$99
Fold
Check
Bet $1

Poker Academy Pro
Lobby Options Dealer Table Window Help
Limit Ring Game: $1/$2 Stakes
Session Stats
Hand Evaluator
Transcript
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
Andrew checks
FLOP: Qd 7s 4c
Andrew checks
GSIBot bets $1
GSIBot
$98
Bet $1
Pot: $3
Andrew
$99
Fold
Call $1
Raise $1

Poker Academy Pro
Lobby Options Dealer Table Window Help
Limit Ring Game: $1/$2 Stakes
GSIBot
$98
Session Stats
Hand Evaluator
Transcript
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
Andrew checks
FLOP: Qd 7s 4c
Andrew checks
GSIBot bets $1
Andrew calls $1
TURN: Qd 7s 4c 3s
Pot: $4
Andrew
$98
Fold
Check
Bet $2


Poker Academy Pro
Lobby Options Dealer Table Window Help
Limit Ring Game: $1/$2 Stakes
Session Stats
Hand Evaluator
Transcript
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
Andrew checks
FLOP: Qd 7s 4c
Andrew checks
GSIBot bets $1
Andrew calls $1
TURN: Qd 7s 4c 3s
Andrew bets $2
GSIBot calls $2
RIVER: Qd 7s 4c 3s Qs
GSIBot
$96
Pot: $8
Andrew
$96
Fold
Check
Bet $2


Poker Academy Pro
Lobby Options Dealer Table Window Help
Limit Ring Game: $1/$2 Stakes
GSIBot
$94
Bet $2
Pot: $10
Andrew
$96
Fold
Call $2
Raise $2
Session Stats
Hand Evaluator
Transcript
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
Andrew checks
FLOP: Qd 7s 4c
Andrew checks
GSIBot bets $1
Andrew calls $1
TURN: Qd 7s 4c 3s
Andrew bets $2
GSIBot calls $2
RIVER: Qd 7s 4c 3s Qs
Andrew checks
GSIBot bets $2


Poker Academy Pro
Lobby Options Dealer Table Window Help
Limit Ring Game: $1/$2 Stakes
GSIBot
$106
GSIBot wins $12 with Two Pair, Queens and Sevens
Andrew
$94
Deal Hand
Session Stats
Hand Evaluator
Transcript
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
Andrew checks
FLOP: Qd 7s 4c
Andrew checks
GSIBot bets $1
Andrew calls $1
TURN: Qd 7s 4c 3s
Andrew bets $2
GSIBot calls $2
RIVER: Qd 7s 4c 3s Qs
Andrew checks
GSIBot bets $2
Andrew calls $2
GSIBot shows 2c 7c
Andrew shows 2s Kh
GSIBot wins $12 with Two Pair, Queens and Sevens

Sequential imperfect information games

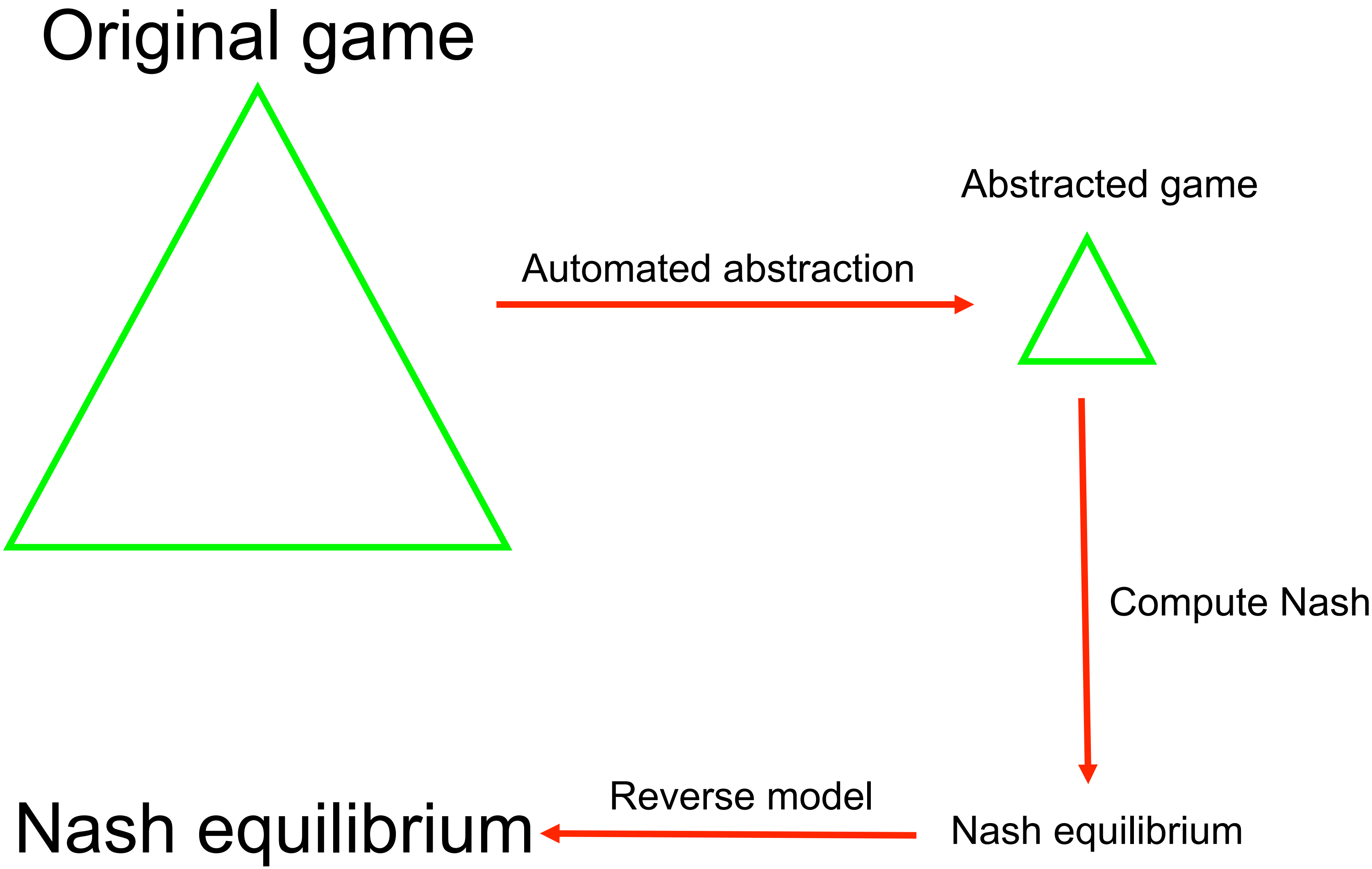
- Players face uncertainty about the state of the world
- Most real-world games are like this
  - A robot facing adversaries in an uncertain, stochastic environment
  - Almost any card game in which the other players' cards are hidden
  - Almost any economic situation in which the other participants possess private information (*e.g.* valuations, quality information)
    - Negotiation
    - Multi-stage auctions (e.g., English)
    - Sequential auctions of multiple items
  - …
- This class of games presents several challenges for AI
  - Imperfect information
  - Risk assessment and management
  - Speculation and counter-speculation
- Techniques for solving <u>sequential complete-information</u> games (like chess) don't directly apply

High-Performance Game Programs

- Many game programs are based on alpha-beta + iterative deepening + extended/singular search + transposition tables + huge databases + ...
- For instance, Chinook searched all checkers configurations with 8 pieces or less and created an endgame database of 444 billion board configurations
- The methods are general, but their implementation is dramatically improved by many specifically tuned-up enhancements (e.g., the evaluation functions) like an F1 racing car

Approach [Gilpin & Sandholm EC'06, JACM'07…]

Now used by all competitive Texas Hold'em programs



Idea: Information filters

- Observation: We can make games smaller by filtering the information a player receives
- Instead of observing a specific signal exactly, a player instead observes a **filtered set** of signals
  - *E.g.* receiving signal {A♠,A♣,A♥,A♦} instead of A♥

GameShrink algorithm

- Bottom-up pass: Run DP to mark isomorphic pairs of nodes in signal tree
- Top-down pass: Starting from top of signal tree, perform the transformation where applicable

- **Theorem.** Conducts all these transformations
  - $\tilde{O}(n^2)$, where n is #nodes in *signal tree*
  - Usually highly *sublinear* in game tree size

- One approximation algorithm: instead of requiring perfect matching, require a matching with a penalty below threshold

Solving Rhode Island Hold'em poker

- AI challenge problem [Shi & Littman 01]
  - 3.1 billion nodes in game tree
- Without abstraction, LP has 91,224,226 rows and columns => unsolvable
- *GameShrink* runs in one second
- After that, LP has 1,237,238 rows and columns
- Solved the LP
  - CPLEX *barrier* method took 8 days & 25 GB RAM
- Exact Nash equilibrium
- Largest incomplete-info game solved to date by over 4 orders of magnitude

Science AAAS

SHARE

13

Researchers have developed a poker-playing computer program that can defeat even the best human players.

(Illustration) Peter and Maria Hoey/www.peterhoey.com

Texas Hold 'em poker solved by computer

By **Emily Conover** | Jan. 8, 2015 , 3:30 PM

http://www.sciencemag.org/news/2015/01/texas-hold-em-poker-solved-computer

Next: Sequential Decision Making

- Poker is an example of *sequential* game with *imperfect* information (uncertainty).
- Next, we tackle:
  – Uncertainty (in outcomes, in information): reasoning
  – Making a sequence of decisions, where agent has memory

Agent Cooperation vs. Competition

- https://deepmind.com/blog/understanding-agent-cooperation/
- Learned behavior: learn evaluation function using "deep" **reinforcement learning to optimize future rewards**
- Resource constraints affect behavior!
- Technical details: https://storage.googleapis.com/deepmind-media/papers/multi-agent-rl-in-ssd.pdf