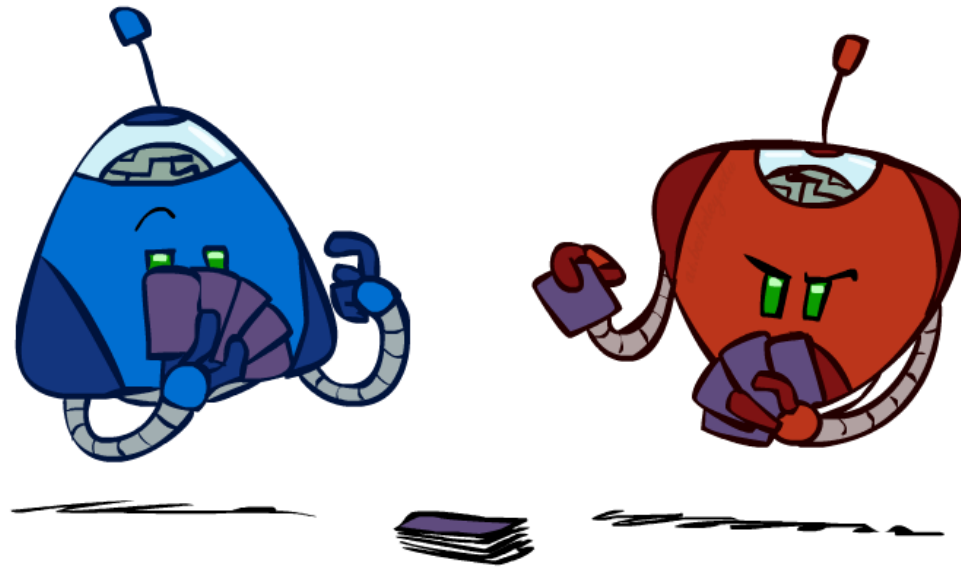
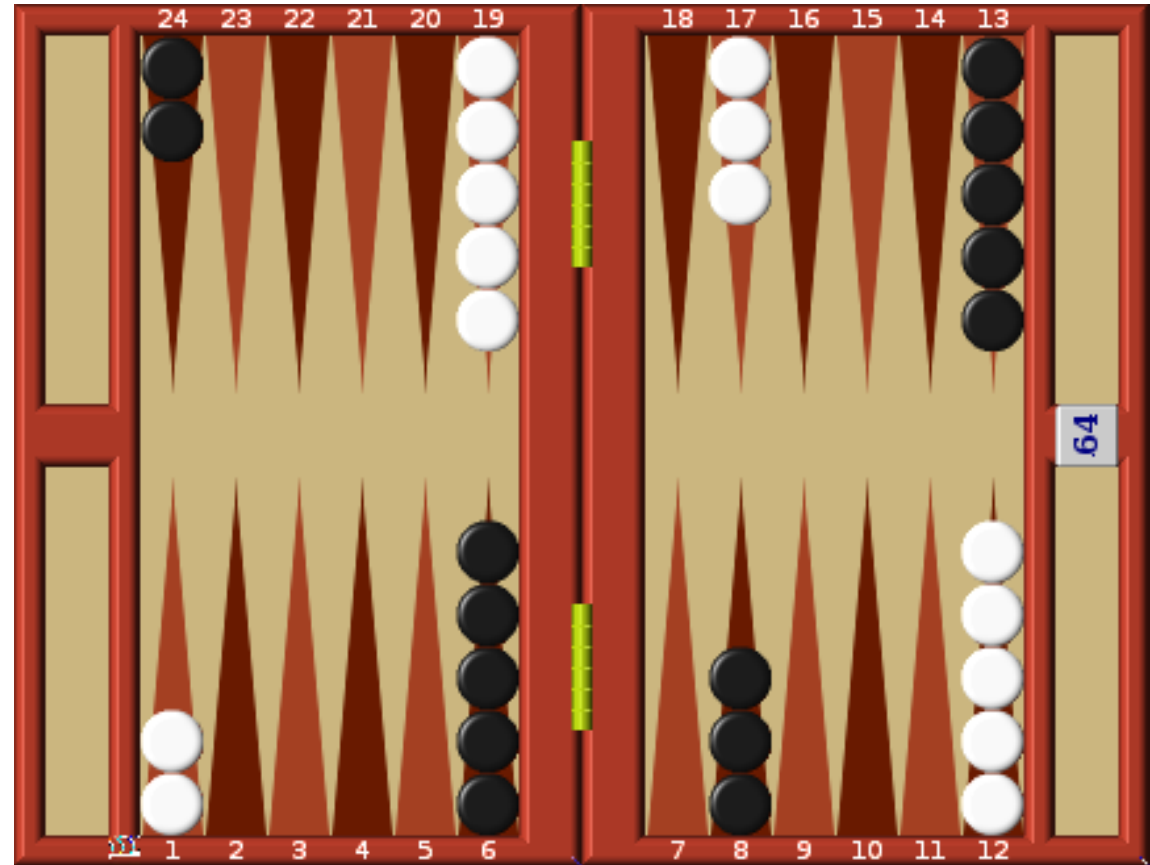


# Stochastic Search



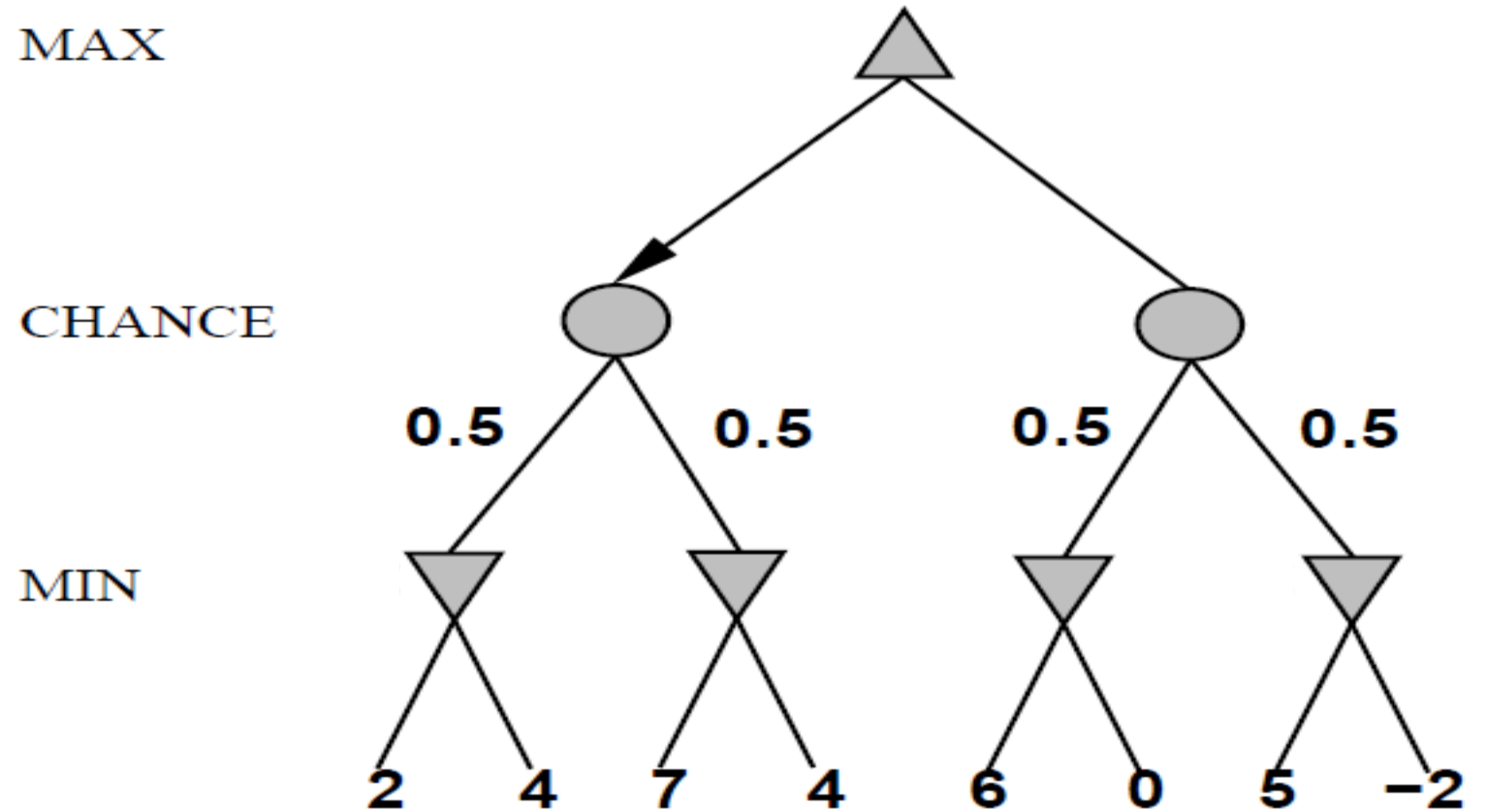
# Nondeterministic Games

- In nondeterministic games, chance introduced by dice, card-shuffling



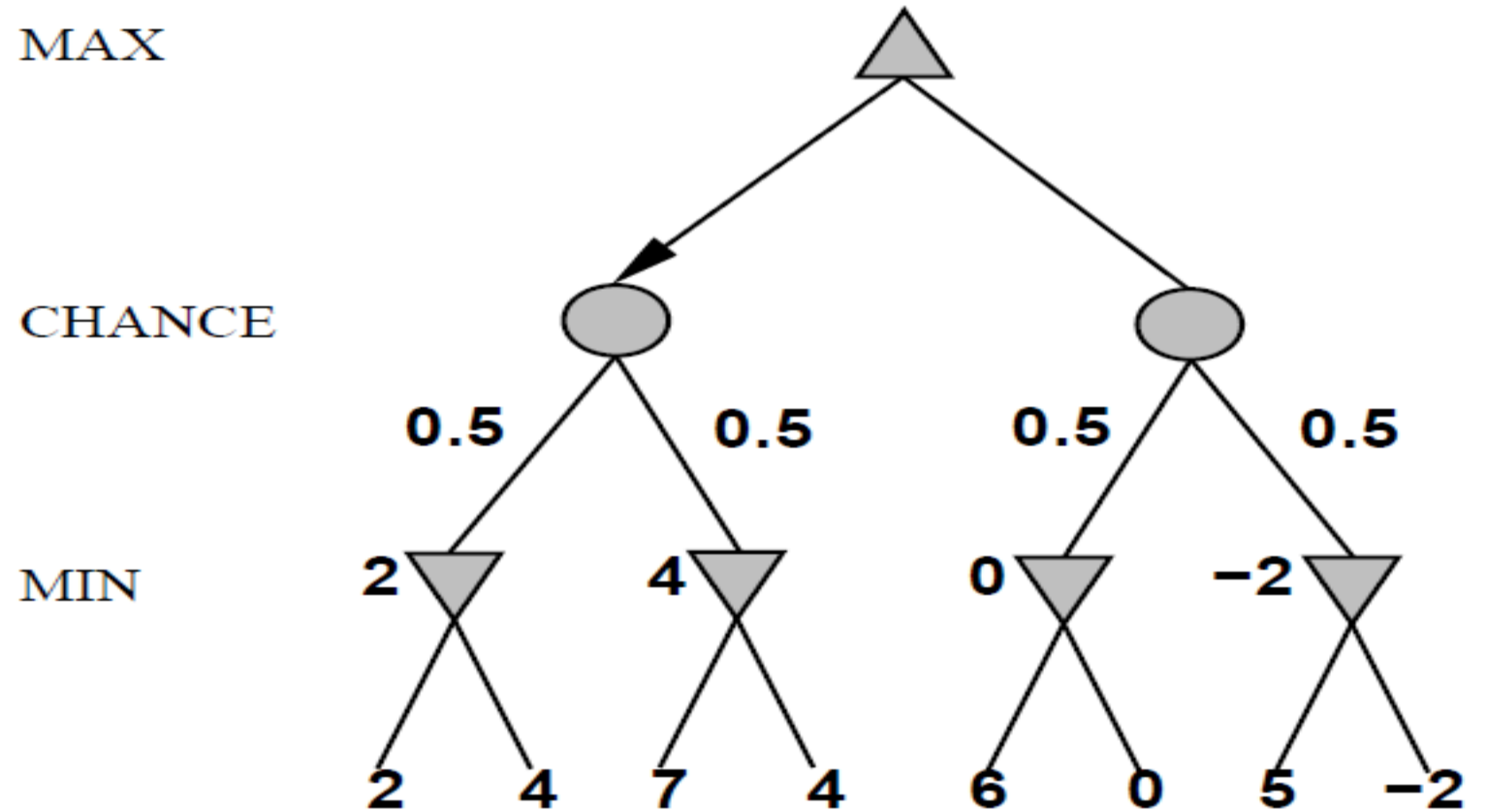
# Nondeterministic Games in General

- Simplified example with coin-flipping:
  - Minimax tree with chance nodes added



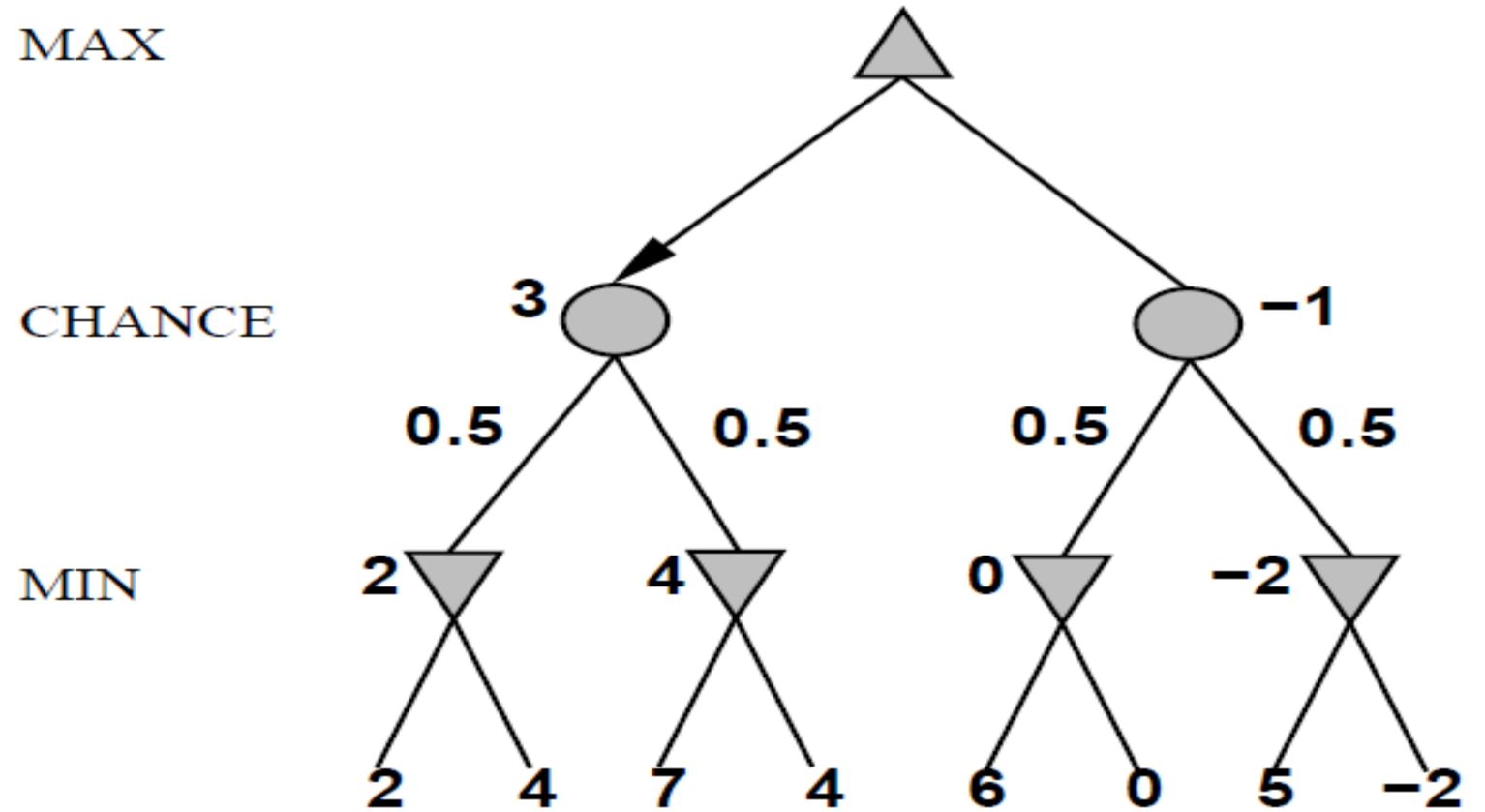
# Nondeterministic Games in General

- Simplified example with coin-flipping:
  - Min makes choices just like we'd expect



# Nondeterministic Games in General

- Simplified example with coin-flipping:
  - We use probability to calculate the expected value of each choice at the chance nodes



# Algorithm for Nondeterministic Games

- Expectiminimax is just like Minimax, except we must also handle chance nodes:

...

**if** *state* is a MAX node **then**

**return** the highest EXPECTIMINIMAX-VALUE of SUCCESSORS(*state*)

**if** *state* is a MIN node **then**

**return** the lowest EXPECTIMINIMAX-VALUE of SUCCESSORS(*state*)

**if** *state* is a chance node **then**

**return** average of EXPECTIMINIMAX-VALUE of SUCCESSORS(*state*)

...

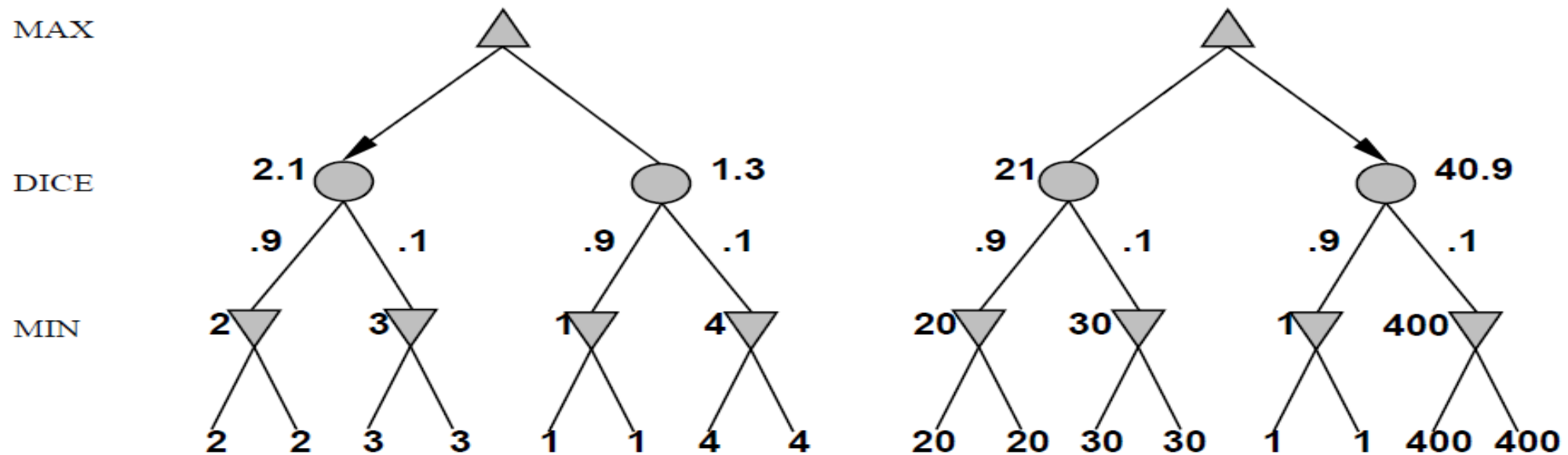
# Nondeterministic Games in Practice

---

- Dice rolls increase branching factor: 21 possible rolls with 2 dice
- Backgammon  $\sim$  20 legal moves
  - Depth 4 =  $20 \times (21 \times 20)^3 \sim 1.2 \times 10^9$
- As depth increases, probability of reaching a given node shrinks
  - Value of lookahead is seriously diminished
- $\alpha$ - $\beta$  pruning is much less effective
- TDGammon uses depth-2 search + very good Eval
  - World-champion level

# Exact Values DO Matter

- Behavior is preserved only by positive linear transformation of Eval
- Hence Eval should be proportional to the expected payoff





# Nondeterministic Games with Partial Observability

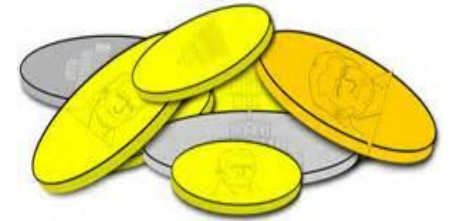
- E.g., card games, where opponent's initial cards are unknown
- Typically we can calculate a probability for each possible deal
- Seems just like having one big dice roll at the beginning of the game
- Idea: compute the minimax value of each action in each deal, then choose the action with highest expected value over all deals
- Special case: if an action is optimal for all deals, it's optimal.
- GIB, current best bridge program, approximates this idea by
  - 1) generating 100 deals consistent with bidding information
  - 2) picking the action that wins most tricks on average
    - “On average” as determined by Monte Carlo simulation, not a calculated expected value



# Commonsense Example

- Day 1: Deterministic

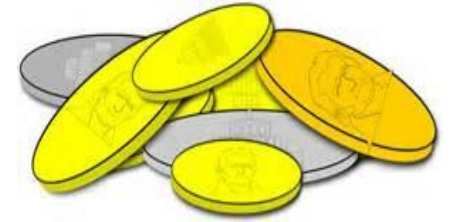
- Road A leads to a small heap of gold pieces (Eval = \$1,000)
- Road B leads to a fork:
  - take the left fork and you'll find a mound of jewels; (Eval = \$1,000,000)
  - take the right fork and you'll be run over by a bus.



# Commonsense Example

- Day 1: Deterministic

- Road A leads to a small heap of gold pieces (Eval = \$1,000)
- Road B leads to a fork:
  - take the left fork and you'll find a mound of jewels; (Eval = \$1,000,000)
  - take the right fork and you'll be run over by a bus.



- Day 2: Deterministic

- Road A leads to a small heap of gold pieces (Eval = \$1,000)
- Road B leads to a fork:
  - take the left fork and you'll be run over by a bus;
  - take the right fork and you'll find a mound of jewels. (Eval = \$1,000,000)



# Commonsense Example

- Day 1 - Deterministic:

- Road A leads to a small heap of gold pieces (Eval = \$1,000)
- Road B leads to a fork:
  - take the left fork and you'll find a mound of jewels; (Eval = \$1,000,000)
  - take the right fork and you'll be run over by a bus.



- Day 2 - Deterministic:

- Road A leads to a small heap of gold pieces (Eval = \$1,000)
- Road B leads to a fork:
  - take the left fork and you'll be run over by a bus;
  - take the right fork and you'll find a mound of jewels. (Eval = \$1,000,000)



- Day 3 – Stochastic (50/50 chance on the road B fork):

- Road A leads to a small heap of gold pieces (Eval = \$1,000)
- Road B leads to a fork:
  - guess correctly and you'll find a mound of jewels; (Eval = \$1,000,000)
  - guess incorrectly and you'll be run over by a bus.



# Proper Analysis

- Intuition that the value of an action is the average of its values in all actual states is WRONG
- With partial observability, value of an action depends on the information state or belief state the agent is in
- Can generate and search a tree of information states
- Leads to rational behaviors such as
  - Acting to obtain information
  - Signaling to one's partner
  - Acting randomly to minimize information disclosure
  - Cooperation and/or alliances



# Summary

---

- Games are fun to work on! (and dangerous)
- They illustrate several important points about AI
  - Perfection is unattainable => must approximate
  - Good idea to think about what to think about
  - Uncertainty constrains the assignment of values to states
  - Optimal decisions depend on information state, not real state

# Semester Project

- The projects we talked about are now posted on the website
- Keep thinking about these – and bring up new ideas if you have them
  - I'll continue to update the website

