

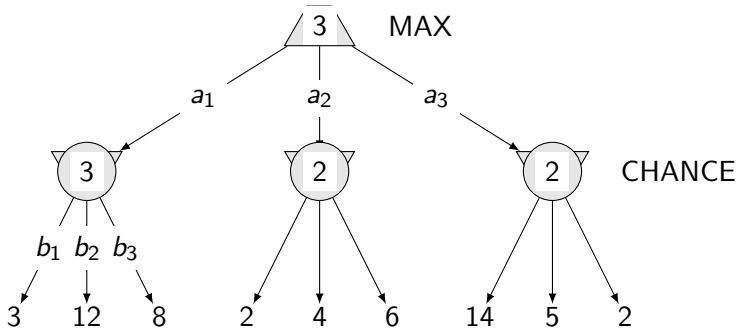
# Uncertainty, Chance, and Utilities

Tomáš Svoboda and Petr Pošík

Vision for Robots and Autonomous Systems, Center for Machine Perception  
Department of Cybernetics  
Faculty of Electrical Engineering, Czech Technical University in Prague

March 22, 2023

# Deterministic opponent → stochastic environment



$b_1, b_2, b_3$  - stochastic branches, uncertain outcomes of  $a_1$  action.

CHANCE nodes are "virtual",  $b_1, b_2, b_3$  are not actions!

---

## Notes

Stochastic environment or stochastic opponent. Simply something that is playing against us. CHANCE nodes are virtual – we use them to represent uncertain outcome of actions.

## Why? Actions may fail, ...



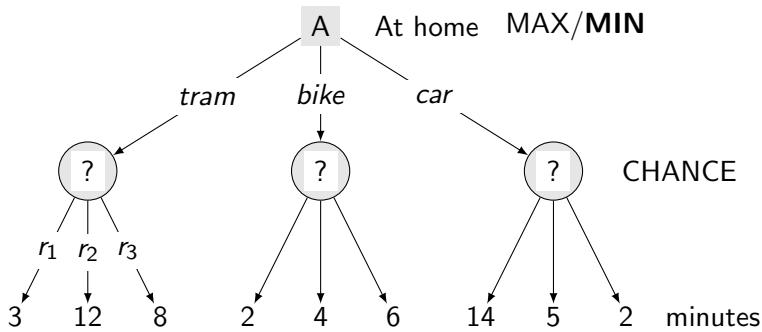
Video: Slipping robot. Vision for Robotics and Autonomous Systems, <http://cyber.felk.cvut.cz/vras>, <https://youtu.be/kvEEHnyCHMs>

3 / 33

### Notes

At a certain moment, command is forward, flippers are rolling but the outcome is different, robot does not move – it is slipping a bit until it catches the grip again.

# Why? Action costs not deterministic, ..., getting to work



**Random variable:** Function mapping situation on rails to values  $T(r_i) = t_i$ :

$t_1 = T(r_1) = 3$  mins (free rails)

$t_2 = T(r_2) = 12$  mins (accident)

$t_3 = T(r_3) = 8$  mins (congestion)

MAX/MIN depends on what the  $t_i$  options and terminal numbers mean. The goal may be to get to work as fast as possible.

## Notes

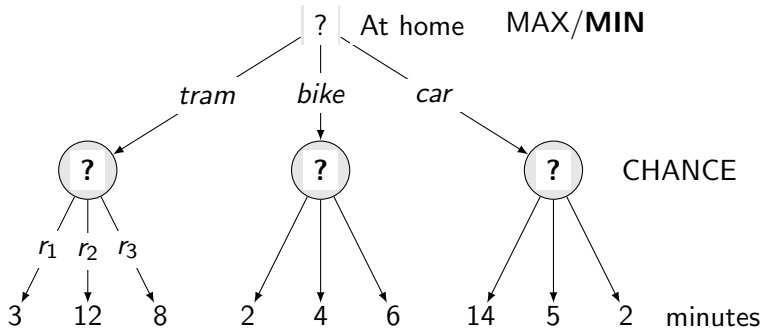
We talk about games. However, game model may be well used for modeling real world problems.

This is just a two-ply game/tree. But think sequentially, or, recursively.

The numbers can be seen as journey duration - then A is the MIN node - min value is the best (MAX) for me.

We can convert it to a classical MAX thinking by changing the Utilities to Working hours-delay - and we want to maximize the working hours.

# Chance nodes values



- ▶ Average case, not the worst case.
- ▶ Calculate expected utilities ...
- ▶ i.e. take weighted average (expectation) of successors

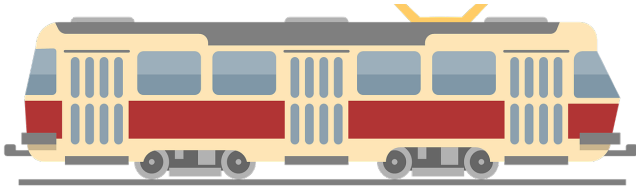
---

## Notes

Later we will learn how to formalize all this as Markov Decision Processes.

## Random variables, probability distribution, . . .

- ▶ Random variable - a function that maps experiment outcomes to values
- ▶ Probability distribution - assignment of probabilities (weights) to the values



- ▶ Random variable:  $T(s)$  - maps situation on rails to values
- ▶ Values of  $T(s)$ :  $T(s) \in \{3, 12, 8\}$ , corresponding to outcomes  $s$  (free rails, accident, congestion)
- ▶ Probability distribution:  $P(T = 3) = 0.3$ ,  $P(T = 12) = 0.1$ ,  $P(T = 8) = 0.6$

A few reminders from laws of probability, **Probabilities**:

- ▶ always non-negative,
- ▶ sum over all possible outcomes is equal to 1.

6 / 33

---

### Notes

You may also think about situation on rails  $s$  in continuous world:

- $T(s)$  maps situation to a travel time,  $T : s \rightarrow \mathbb{R}^+$ , depending on traffic intensity
- “free rails” corresponds to  $T(s) \leq$  some threshold number.
- Probability has a meaning of: what is the chance that the traffic intensity will be higher than something and smaller than something else.  $P(t_{\text{low}} \leq T(s) < t_{\text{high}})$ .

# Expectations, ...

How long does it take to go to work by tram?

- ▶ Depends on the random variable  $T$  with possible values  $t_1, t_2, t_3$  (corresponding to situation on rails).
- ▶ What is the **expectation** of the time?  
Using values  $t_1, t_2, t_3$  of random variable  $T$ :

$$E(T) = P(t_1)t_1 + P(t_2)t_2 + P(t_3)t_3$$

Or, using random outcomes  $r_1, r_2, r_3$ :

$$E(T) = P(r_1)T(r_1) + P(r_2)T(r_2) + P(r_3)T(r_3)$$

Expected value of a discrete r.v.: **Weighted average**

---

## Notes

The Expectation is a kind of long-horizon/many-realizations value. Think about trials/simulations.

# Expectimax

```
function EXPECTIMAX(state) return a value
  if IS-TERMINAL(state): return UTILITY(state)
  if state (next agent) is MAX: return MAX-VALUE(state)
  if state (next agent) is CHANCE: return EXP-VALUE(state)
end function
```

---

```
function MAX-VALUE(state) return value  $v$ 
   $v \leftarrow -\infty$ 
  for  $a$  in ACTIONS(state) do
     $v \leftarrow \max(v, \text{EXPECTIMAX}(\text{RESULT}(\text{state}, a)))$ 
  end for
end function
```

---

```
function EXP-VALUE(state) return value  $v$ 
   $v \leftarrow 0$ 
  for all  $r \in$  random outcomes do
     $v \leftarrow v + P(r) \text{EXPECTIMAX}(\text{RESULT}(\text{state}, r))$ 
  end for
end function
```

8 / 33

---

## Notes

The scheme very much resembles the MINIMAX algorithm. Before, we had the deterministic opponent – MIN node.



# How about the Reversi game?

- ▶ Is there any space for randomness?
- ▶ Is the opponent really greedy and clever enough?
- ▶ Hope for chance when there is adversarial world – **Dangerous optimism** .
- ▶ Assuming worst case even if it is not likely – **Dangerous pessimism** .

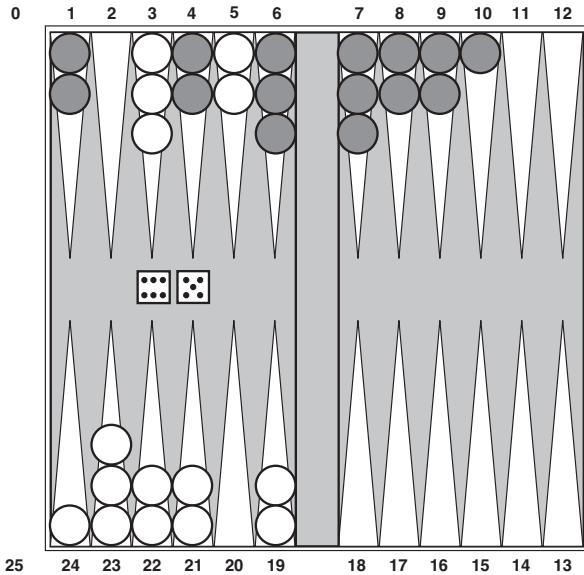
---

## Notes

For games where there is only a single final outcome (value)—e.g., you win, loose, or draw—and no bonus for winning fast, it does not pay off to be optimistic and assume your opponent is a fool. Such optimism can be dangerous.

For other games, like the Pacman example in the UC Berkeley lecture where there is cost for every move you make, assuming the ghost is a perfect adversary while it is behaving randomly may cost you some points.

# Games with chance and strategy



## Notes

Read the rules at: <https://en.wikipedia.org/wiki/Backgammon> or elsewhere.

White moves clockwise - toward 25, black counterclockwise - toward 0.

Moving step defined by the dice, one after another.

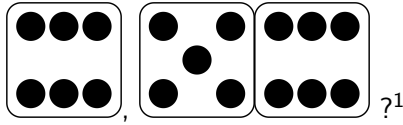
Moving out the gameboard from last quarter only after all stones are there.

No move to position where more than one opp stone.

One stone can be captured (see position 10).

## Random variable: Throwing two dice

Do we care which die comes first?



A  $1/24$

B  $1/36$

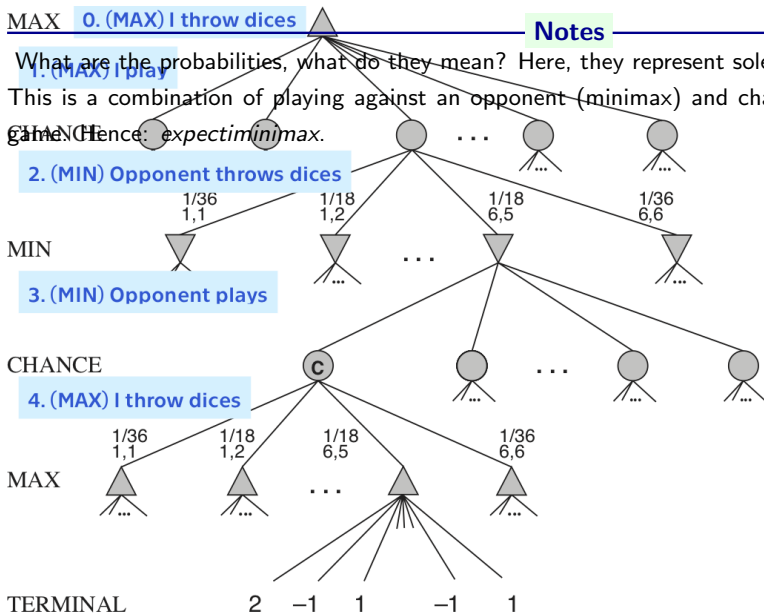
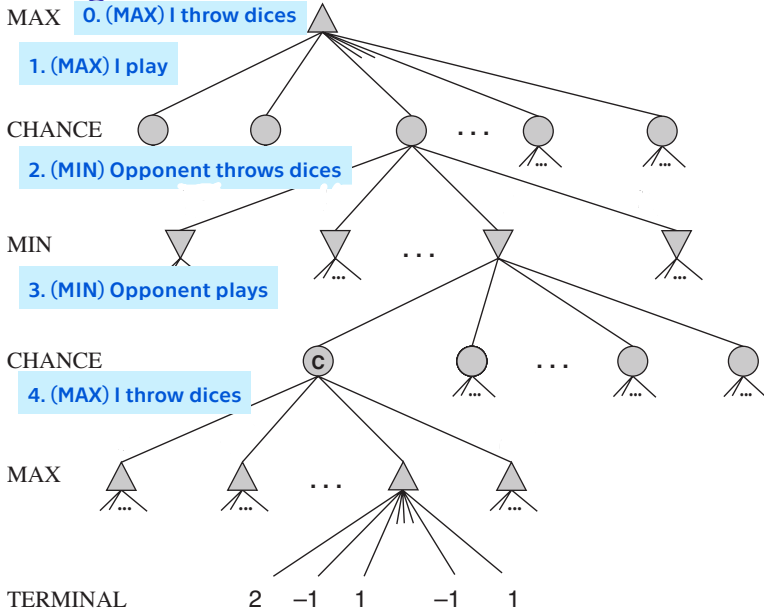
C  $1/18$

D  $1/6$

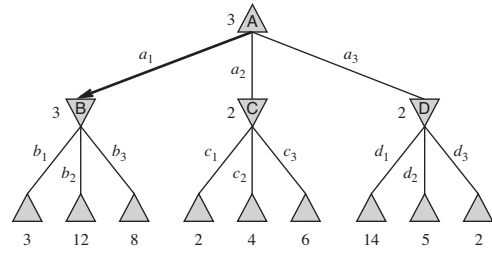
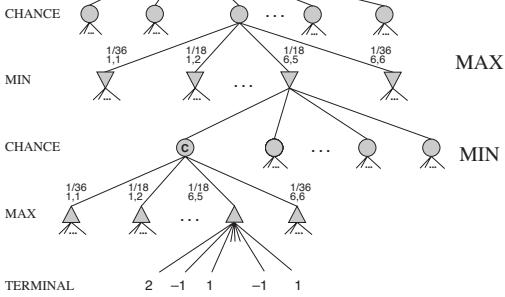
---

<sup>1</sup>Source of dice images: <https://flyclipart.com/dice-clipart-tool-rolling-dice-clipart-248574>

# Mixing MAX, CHANCE, and MIN nodes



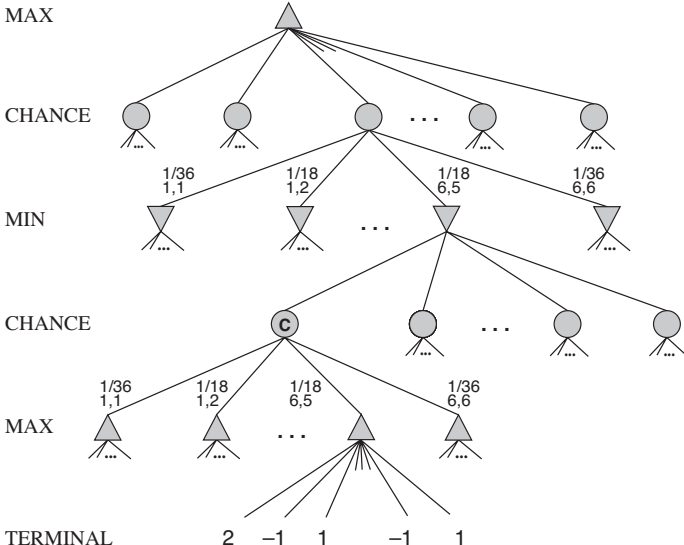
# Mixing layer types - chances inserted



Extra random agent that moves after each MAX and MIN agent

$$\begin{aligned}
 \text{EXPECTIMINIMAX}(s) &= \\
 &= \begin{cases} \text{UTILITY}(s, \text{MAX}) & \text{if IS-TERMINAL}(s) \\ \max_a \text{EXPECTIMINIMAX}(\text{RESULT}(s, a)) & \text{if TO-PLAY}(s) = \text{MAX} \\ \min_a \text{EXPECTIMINIMAX}(\text{RESULT}(s, a)) & \text{if TO-PLAY}(s) = \text{MIN} \\ \sum_r P(r) \text{EXPECTIMINIMAX}(\text{RESULT}(s, r)) & \text{if TO-PLAY}(s) = \text{CHANCE} \end{cases}
 \end{aligned}$$

# Mixing chance into min/max tree. How big is the tree going to be?



- ▶  $b$  branching factor
- ▶  $m$  maximum depth
- ▶  $n$  number of distinct rolls

What is the time complexity of EXPECTIMINIMAX?

- A  $O(b^{mn})$
- B  $O(b^m n)$
- C  $O(b^m n^b)$
- D  $O(b^m n^m)$

## Notes

$$O(b^m n^m)$$

There are actually  $n^m$  different minimax trees. Each layer of  $n$  distinct rolls multiplies the number of min-max trees.

It is BIG! With roughly 20 legal moves in every position and 21 possible rolls of 2 dice, for expectimax search into depth = 2, we already have:

$$20 * (21 * 20)^3 = 1.2 * 10^9 \text{ possibilities.}$$

So we cannot get very far with search. At the same time, given the stochasticity, the fact that we cannot search so deep is less damaging.

We need an evaluation function.

Computer program for playing Backgammon – TD-Gammon, see Chapter 16.1 [4] for a thorough explanation. We will discuss the Reinforcement learning and learning of linear classifiers later in the course.

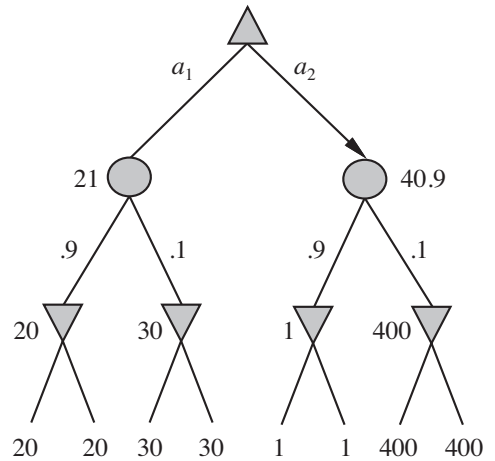
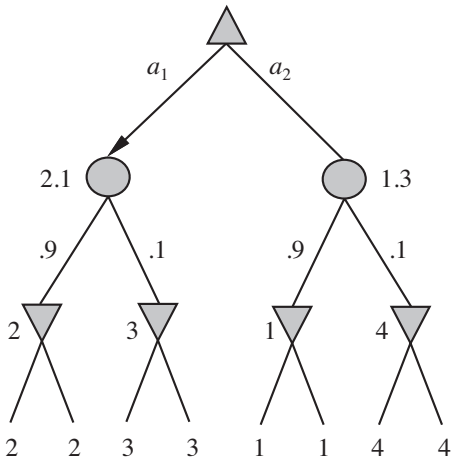
- depth 2
- good evaluation function + reinforcement learning
- 1st AI world champion in any game

# Evaluation function

MAX

CHANCE

MIN



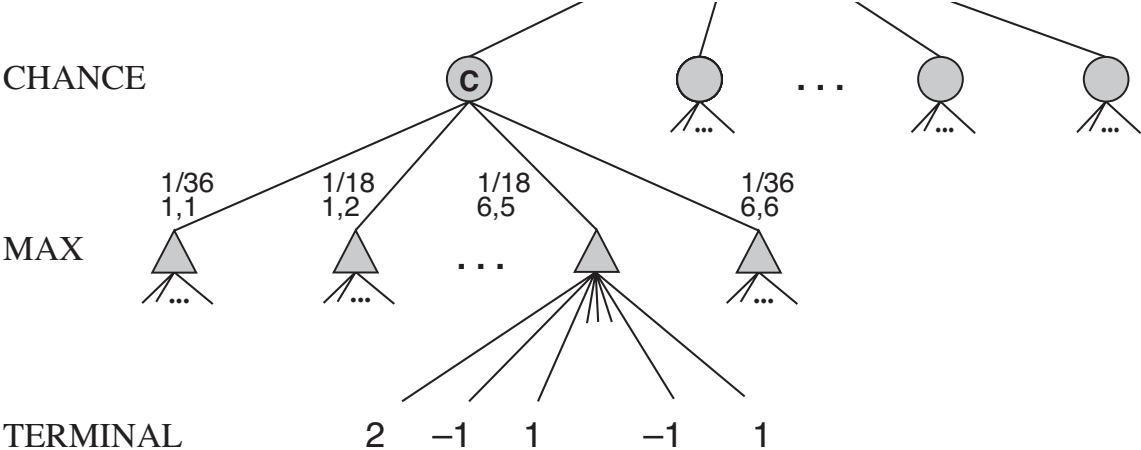
- ▶ Left:  $a_1$  is the best. Right:  $a_2$  is the best. Ordering of the (terminal) leaves is the same.
- ▶ Scale matters! Not only ordering.
- ▶ Can we prune the tree? ( $\alpha, \beta$  like?)

## Notes

About the scale. Utilities will be discussed later in this lecture.

Note: Only linear transformations of utilities preserve the same optimal choice of actions.

# Pruning expectiminimax tree



- ▶ Bounds on terminal utilities needed. Terminal values from  $-2$  to  $2$ .
- ▶ Monte Carlo simulation for evaluation of a position (state).

---

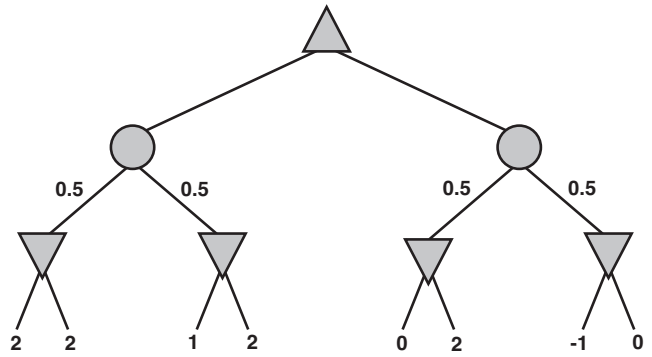
## Notes

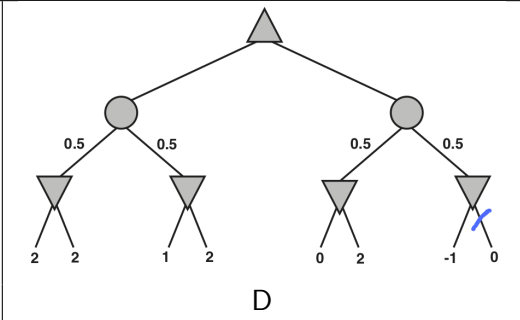
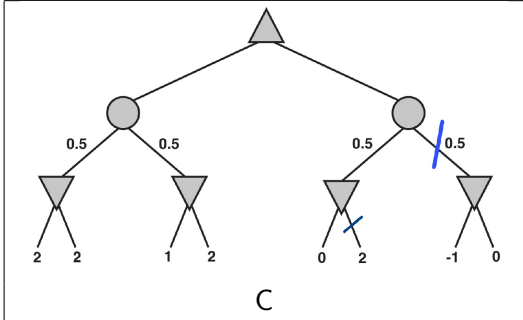
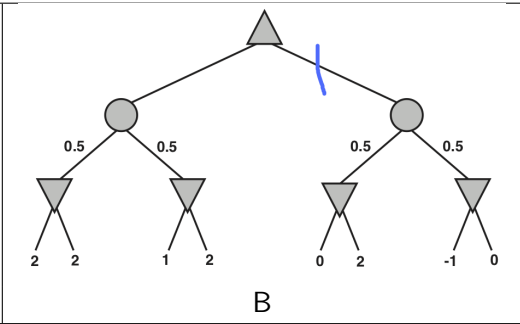
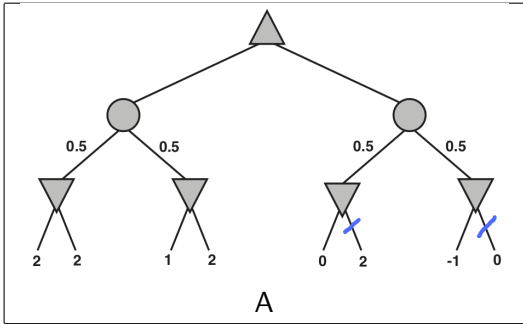
**Monte Carlo Simulation** . From a given position play against itself, many times, use random dice rolls. Collect results. Compute state value.



# Where to prune the Expectimax tree

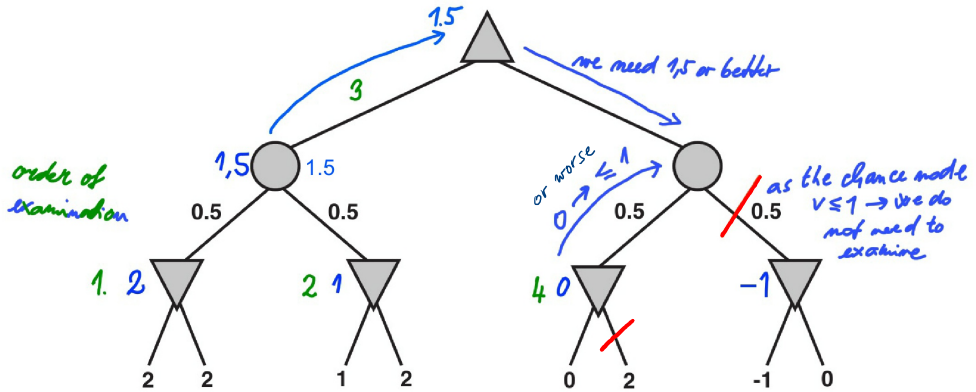
- ▶ Assume terminal nodes bounded to  $-2$  to  $2$ , inclusive
- ▶ Going from left to right.
- ▶ Which branches can be pruned out?





Assume terminal nodes bounded to  $-2$  to  $2$ , inclusive. Going from left to right.

Notes



# Multi-player games

to move

A

(1, 2, 6)

B

(1, 2, 6)

(1, 5, 2)

C

(1, 2, 6)

(6, 1, 2)

(1, 5, 2)

(5, 4, 5)

A

(1, 2, 6)

(4, 2, 3)

(6, 1, 2)

(7, 4, 1)

(5, 1, 1)

(1, 5, 2)

(7, 7, 1)

(5, 4, 5)

- ▶ Utility tuples
- ▶ Each player maximizes its own
- ▶ Coalitions, cooperations, competitions may be dynamic

---

## Notes

I bet everybody remembers playing this kind of game ... Remember the games you played when being kids.

# Uncertainty recap (enough games, back to the robots/agents)



- ▶ Uncertain outcome of an action.
- ▶ Robot/Agent may not know the current state!

20 / 33

---

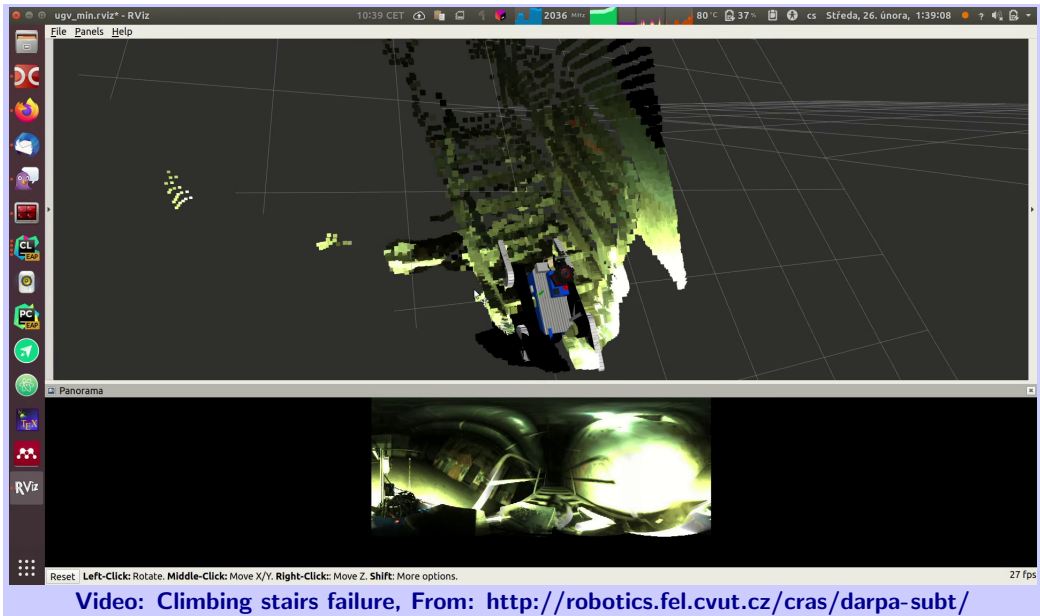
## Notes

What is state for the robot?

- inner state of the robot (**interoceptive** measurement)
  - speed
  - inclination, orientation (N,E,S,W)
  - battery status
  - ...
- environment (**exteroceptive** measurement/sensing)
  - terrain profile close to robot
  - robot position within the world frame
  - ...

All of this may influence the decision about the best next action(s).

# Uncertain outcome of an action



**Video: Climbing stairs failure, From: <http://robotics.fel.cvut.cz/cras/darpa-subt/>**

21 / 33

## Notes

Climbing up, rear flipper got too weak, gave up supporting the robot and it flipped back. Reason unknown, the robot climbed up similar stairs successfully many times.

# Uncertain, partially observable environment

- ▶ Current state  $s$  may be unknown, observations  $\mathbf{e}$
- ▶ Take action  $a$
- ▶ Uncertain outcome  $\text{RESULT}(a)$
- ▶ Probability of outcome  $s'$  given  $\mathbf{e}$  is

$$P(\text{RESULT}(a) = s' | a, \mathbf{e})$$

- ▶ Utility function  $U(s)$  corresponds to agent preferences.
- ▶ **Expected utility** of an action  $a$  given  $\mathbf{e}$ :

$$EU(a|\mathbf{e}) = \sum_{s'} P(\text{RESULT}(a) = s' | a, \mathbf{e}) U(s')$$



Amatrice, Italy, 2016.

---

## Notes

See [3], Ch. 16 Making simple decisions.

# Rational agent

Agent's expected utility of an action  $a$  given  $\mathbf{e}$ :

$$EU(a|\mathbf{e}) = \sum_{s'} P(\text{RESULT}(a) = s' | a, \mathbf{e}) U(s')$$

What should a rational agent do?

Is it then all solved? Do we know all what we need?

- ▶  $P(\text{RESULT}(a) = s' | a, \mathbf{e})$
- ▶  $U(s')$

---

## Notes

Well, obviously take the action that maximizes the expected utility.

In some realms is the utility  $U(s)$  replaced by a **loss**  $L(s)$ , and the rational agent picks the minimum loss. Complete causal model is needed to compute the probabilities  $P$ , and a complete search/planning to the end required for computing the utility  $U$ . And, eh, the state space may be, and often is, infinite. Enough pessimism, we will come back to this in next lectures/courses.

# Utilities



- ▶ Where do utilities come from?
- ▶ Does averaging make sense?
- ▶ Do they exist?
- ▶ What if our preferences can't be described by utilities?

---

## Notes

Before we start solving all this, let's talk about utilities. Where do they come from, are they unique, . . . . Actually, let's talk about preferences first, we all have some **preferences** . Later, we will derive utilities from them.



# Agent/Robot Preferences

- ▶ Prizes  $A, B$
- ▶ Lottery: uncertain prizes  $L = [p, A; (1 - p), B]$

Preference, indifference, ...

- ▶ Robot prefers  $A$  over  $B$ :  $A \succ B$
- ▶ Robot has no preferences:  $A \sim B$
- ▶ in between:  $A \succsim B$

---

## Notes

You may use agent/robot/algorithm/..., according to your preferences.

Lottery can be seen as a chance node.

# Rational preferences

- ▶ Transitivity:  $(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$
- ▶ Completeness:  $(A \succ B) \vee (B \succ A) \vee (A \sim B)$
- ▶ Continuity:  $(A \succ B \succ C) \Rightarrow \exists p [p, A; 1 - p, C] \sim B$
- ▶ Substitutability:  $A \sim B \Rightarrow [p, A; 1 - p, C] \sim [p, B; 1 - p, C]$ . The same for  $\succ$  and  $\sim$ .
- ▶ Monotonicity:  $A \succ B \Rightarrow (p > q) \Leftrightarrow [p, A; 1 - p, B] \succ [q, A; 1 - q, B]$ . Agent must prefer a lottery with higher chance to win.
- ▶ Decomposability, compressing compound lotteries into one:  
 $[p, A; 1 - p, [q, B; 1 - q, C]] \sim [p, A; (1 - p)q, B; (1 - p)(1 - q), C]$

Axioms of utility theory.

Motivation: if agent/robot violates an axiom  $\Rightarrow$  irrational agent/robot.

26 / 33

---

## Notes

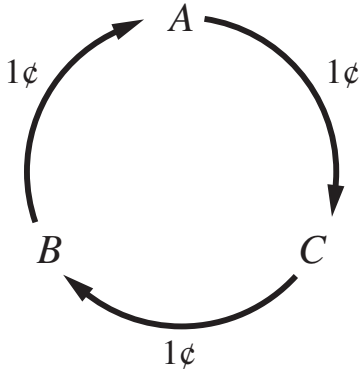
If you think it through you will see that the properties of rational preferences are quite logical, *rational* if you want ;-)

- Transitivity:  $(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$
- Completeness:  $(A \succ B) \vee (B \succ A) \vee (A \sim B)$
- Continuity:  $(A \succ B \succ C) \Rightarrow \exists p [p, A; 1 - p, C] \sim B$
- Substitutability:  $A \sim B \Rightarrow [p, A; 1 - p, C] \sim [p, B; 1 - p, C]$ . The same for  $\succ$  and  $\sim$ .
- Monotonicity:  $A \succ B \Rightarrow (p > q) \Leftrightarrow [p, A; 1 - p, B] \succ [q, A; 1 - q, B]$ . Agent must prefer a lottery with higher chance to win.
- Decomposability, compressing compound lotteries into one:  
 $[p, A; 1 - p, [q, B; 1 - q, C]] \sim [p, A; (1 - p)q, B; (1 - p)(1 - q), C]$

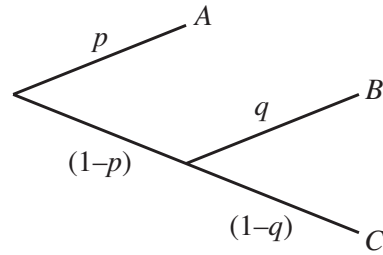
Any agent that breaks the rules can be shown irrational.

## Transitivity and decomposability

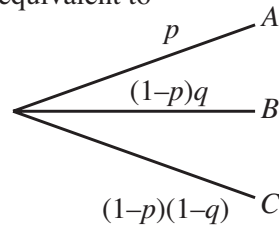
Goods  $A, B, C$  and (nontransitive) preferences of an (irrational) agent  $A \succ B \succ C \succ A$ .



(a)



is equivalent to



(b)

### Notes

$A, B, C$  are goods. Suppose an agent has  $A$ . As the agent prefers  $C \succ A$  we offer him/her the exchange plus the agent gives one cent (the smallest currency unit). The same for  $B \succ C$ , and  $A \succ B$ . At the end of the round, the agent has  $A$  again but also 3 cents less. And this can continue until the poor agent has no money at all.

# Maximum expected utility principle

Given the rational preferences (constraints), there exists a real valued function  $u$  such that:

$$u(A) > u(B) \Leftrightarrow A \succ B$$

$$u(A) = u(B) \Leftrightarrow A \sim B$$

Expected utility of a Lottery  $L$  (outcomes  $s_i$  with probabilities  $p_i$ ):

$$L([p_1, S_1; \dots; p_n, S_n]) = \sum_i p_i u(S_i)$$

Proof in [5].

Is a utility  $u$  function unique?

---

## Notes

In other words, we can find a utility to any preferences.

No, it is not unique:

$$u'(S) = au(S) + b$$

$a > 0$  makes the agent behavior the same. Think about Fahrenheit to Celsius conversion.



---

## Notes

# Utility of money

You triumphed in a TV show!

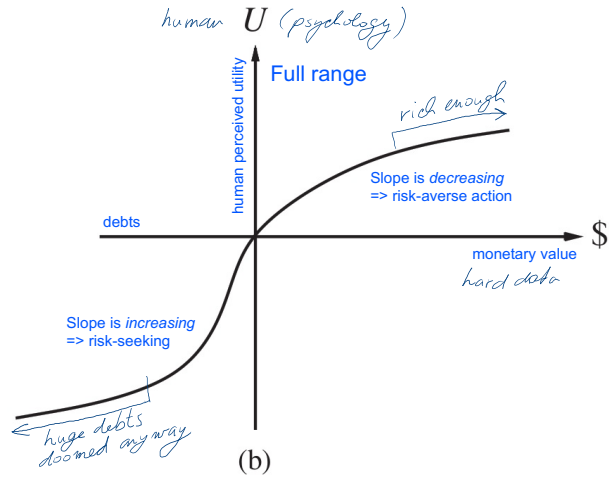
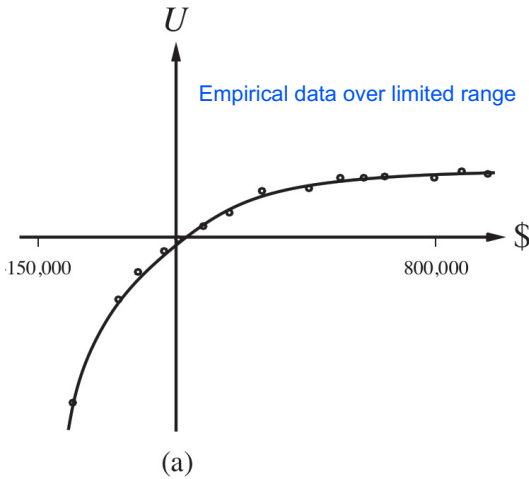
- a) Take \$1,000,000 . . . or
- b) Flip a coin and loose all or win \$2,500,000

---

## Notes

Lottery b) Expected monetary value (EMP) vs. utility. Clearly  $EMP(b)$  is bigger than  $EMP(a)$ . But what about the (human) Utility?

# Utility of money: human psychology vs. hard data



## Notes

Lottery b) Expected monetary value (EMP) vs. utility. Clearly EMP(b) is bigger than EMP(a). But what about the (human) Utility?

$$u(a) = u(S_{k+1,000,000})$$

$$u(b) = \frac{1}{2}u(S_k) + \frac{1}{2}u(S_{k+2,500,000}),$$

where  $S_k$  is the state of possessing  $k\$$  (current wealth).

E.g., imagine  $u(S_k) = 5$ ,  $u(S_{k+1000000}) = 8$ ,  $u(S_{k+2500000}) = 9$ . Then the rational decision is to decline the gamble.

Based on empirical studies, the human utility of money is rather logarithmic. People are in general *risk-averse*. This also motivates insurances.

# References I

Some figures from [3], Chapters 5, 16. Human utilities are discussed in [2]. This lecture has been also greatly inspired by the 7th lecture of CS 188 at <http://ai.berkeley.edu> as it conveniently bridges the world of deterministic search and sequential decisions in uncertain worlds.

[1] Christopher M. Bishop.

*Pattern Recognition and Machine Learning.*

Springer Science+Business Media, New York, NY, 2006.

[https://www.microsoft.com/en-us/research/uploads/prod/2006/01/](https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf)

[Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf](https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf).

[2] Daniel Kahneman.

*Thinking, Fast and Slow.*

Farrar, Straus and Giroux, 2011.



## References II

- [3] Stuart Russell and Peter Norvig.  
*Artificial Intelligence: A Modern Approach*.  
Prentice Hall, 3rd edition, 2010.  
<http://aima.cs.berkeley.edu/>.
- [4] Richard S. Sutton and Andrew G. Barto.  
*Reinforcement Learning; an Introduction*.  
MIT Press, 2nd edition, 2018.  
<http://www.incompleteideas.net/book/the-book-2nd.html>.
- [5] John von Neumann and Oskar Morgenstern.  
*Theory of Games and Economic Behavior*.  
Princeton, 1944.  
[https://en.wikipedia.org/wiki/Theory\\_of\\_Games\\_and\\_Economic\\_Behavior](https://en.wikipedia.org/wiki/Theory_of_Games_and_Economic_Behavior), Utility theorem.