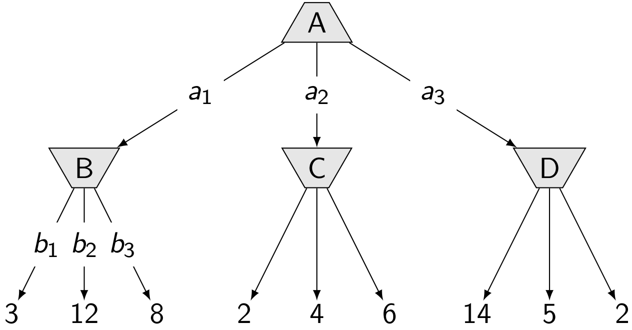
Uncertainty, Chances, and Utilities

Tomáš Svoboda

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

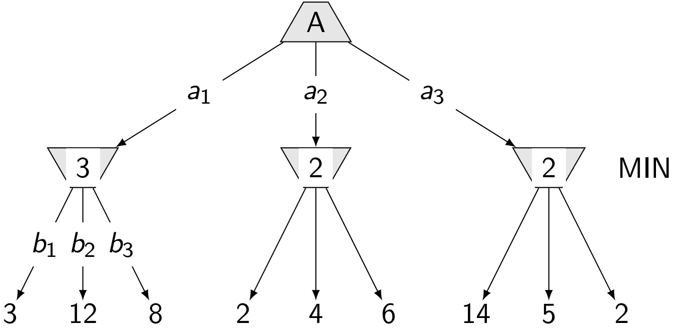
March 11, 2019

Deterministic opponent → stochastic environment



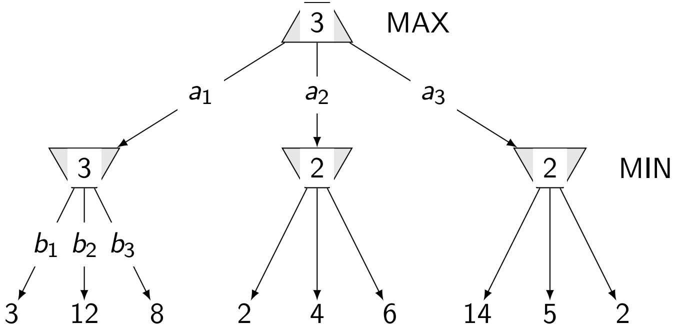
b_1, b_2, b_3 - probable branches, uncertain outcomes of a_1 action.

Deterministic opponent → stochastic environment



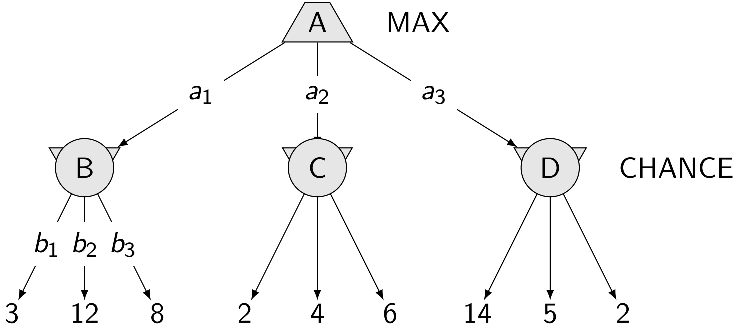
b_1, b_2, b_3 - probable branches, uncertain outcomes of a_1 action.

Deterministic opponent → stochastic environment



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

Deterministic opponent → stochastic environment



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

Why? Actions may fail, ...

Video: Slipping robot

Vision for Robotics and Autonomous Systems, `http://cyber.felk.cvut.cz/vras`

Why? Actions may fail, . . . , getting to work

A At home

tram *bike* *car*

Random variable: Situation on rails R

r_1 free rails

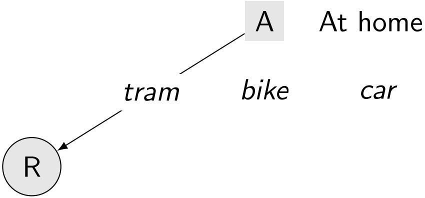
r_2 accident

r_3 congestion

MAX/MIN depends on what the $r_?$ otions and terminal numbers mean.

The goal may be to get to work as fast as possible.

Why? Actions may fail, . . . , getting to work



Random variable: Situation on rails R

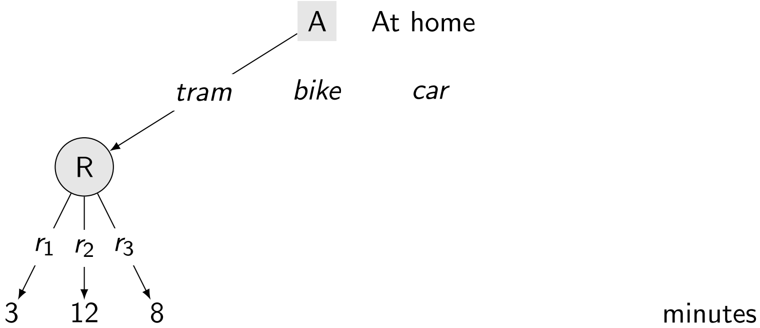
r_1 free rails

r_2 accident

r_3 congestion

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

Why? Actions may fail, . . . , getting to work



Random variable: Situation on rails R

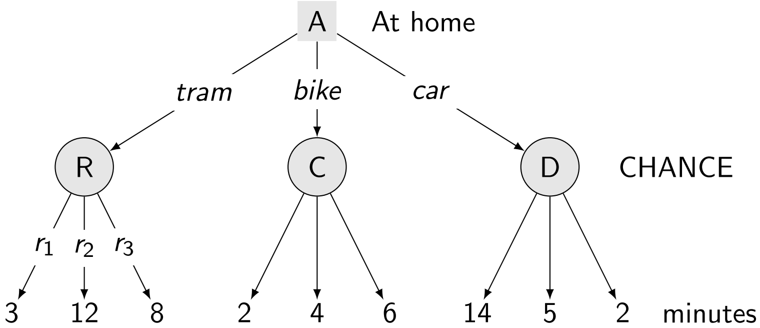
r_1 free rails

r_2 accident

r_3 congestion

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

Why? Actions may fail, . . . , getting to work



Random variable: Situation on rails R

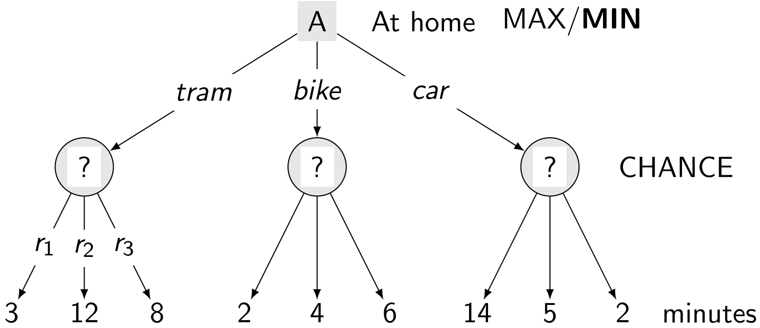
r_1 free rails

r_2 accident

r_3 congestion

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

Why? Actions may fail, . . . , getting to work



Random variable: Situation on rails R

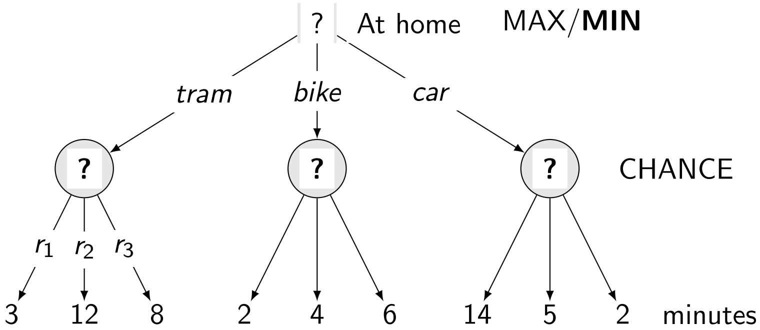
r_1 free rails

r_2 accident

r_3 congestion

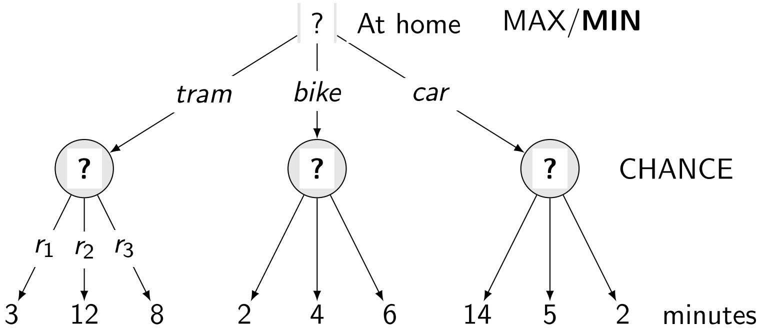
MAX/MIN depends on what the $r_?$ otions and terminal numbers mean.
The goal may be to get to work as fast as possible.

Chance nodes values



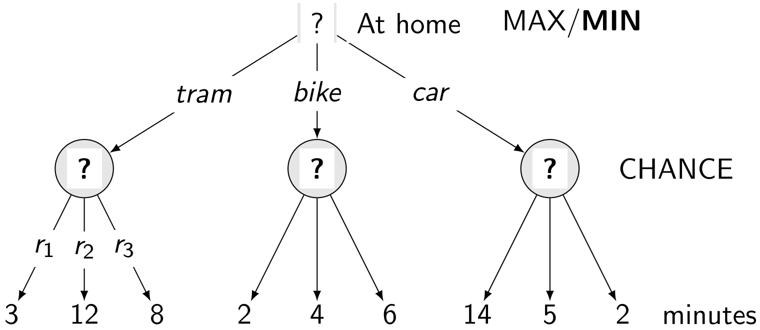
- Average case, not the worst case.
- Calculate expected utilities . . .
- i.e. take weighted average (expectation) of successors

Chance nodes values



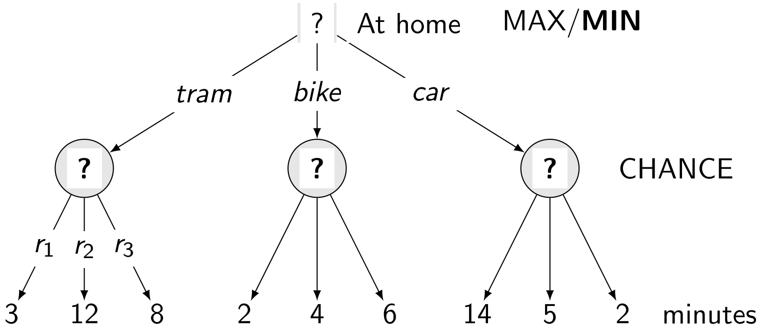
- Average case, not the worst case.
- Calculate expected utilities . . .
- i.e. take weighted average (expectation) of successors

Chance nodes values



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

Chance nodes values



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

Expectimax

function EXPECTIMAX(state) **return** a value
    if TERMINAL-TEST(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, EXPECTIMAX(RESULT(state,a)))
    end for
end function

---

function EXP-VALUE(state) **return** value v
    $v \leftarrow 0$
    for all $r \in$ random events **do**
        $v \leftarrow v + P(r)$ EXPECTIMAX(RESULT(state,r))
    end for
end function

Expectimax

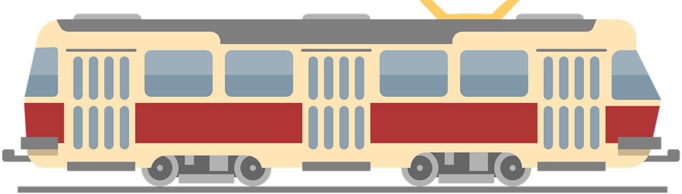
```
function EXPECTIMAX(state) return a value
    if TERMINAL-TEST(state): return UTILITY(state)
    if state (next agent) is MAX: return MAX-VALUE(state)
    if state (next agent) is CHANCE: return EXP-VALUE(state)
end function
```

```
function MAX-VALUE(state) return value v
    v ← −∞
    for a in ACTIONS(state) do
        v ← max(v, EXPECTIMAX(RESULT(state,a)))
    end for
end function
```

```
function EXP-VALUE(state) return value v
    v ← 0
    for all r ∈ random events do
        v ← v + P(r) EXPECTIMAX(RESULT(state,r))
    end for
end function
```

Random variables, probability distribution, . . .

- Random variable - an event with unknown outcome
- Probability distribution - assignment of weights to the outcomes



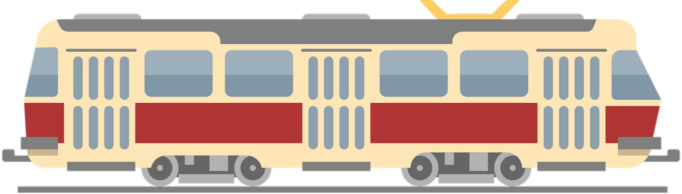
- Random variable: R - situation on rails
- Outcomes/events: $r \in \{\text{free rails, accindent, congestion}\}$
- Probability distribution: $P(R = \text{free rails}) = 0.3$, $P(R = \text{accident}) = 0.1$, $P(R = \text{congestion}) = 0.6$

Few reminders from laws of probability, Probabilities:

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

Random variables, probability distribution, ...

- Random variable - an event with unknown outcome
- Probability distribution - assignment of weights to the outcomes



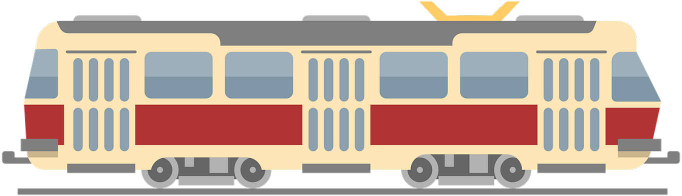
- Random variable: R - situation on rails
- Outcomes/events: $r \in \{\text{free rails, accindent, congestion}\}$
- Probability distribution: $P(R = \text{free rails}) = 0.3$, $P(R = \text{accident}) = 0.1$, $P(R = \text{congestion}) = 0.6$

Few reminders from laws of probability, Probabilities:

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

Random variables, probability distribution, . . .

- Random variable - an event with unknown outcome
- Probability distribution - assignment of weights to the outcomes



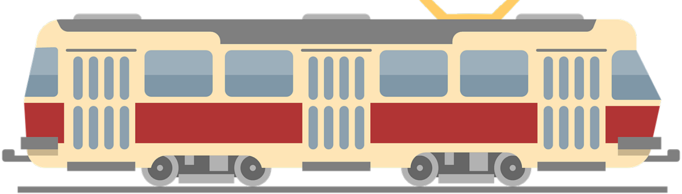
- Random variable: R - situation on rails
- Outcomes/events: $r \in \{\text{free rails, accindent, congestion}\}$
- Probability distribution: $P(R = \text{free rails}) = 0.3$, $P(R = \text{accident}) = 0.1$, $P(R = \text{congestion}) = 0.6$

Few reminders from laws of probability, Probabilities:

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

Random variables, probability distribution, . . .

- Random variable - an event with unknown outcome
- Probability distribution - assignment of weights to the outcomes



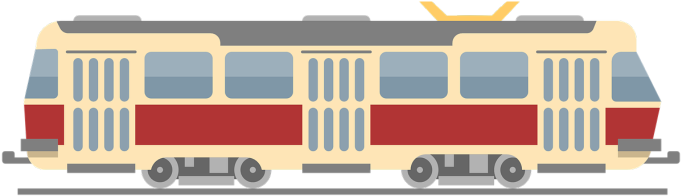
- Random variable: R - situation on rails
- Outcomes/events: $r \in \{\text{free rails, accindent, congestion}\}$
- Probability distribution: $P(R = \text{free rails}) = 0.3$, $P(R = \text{accident}) = 0.1$, $P(R = \text{congestion}) = 0.6$

Few reminders from laws of probability. Probabilities:

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

Random variables, probability distribution, . . .

- Random variable - an event with unknown outcome
- Probability distribution - assignment of weights to the outcomes



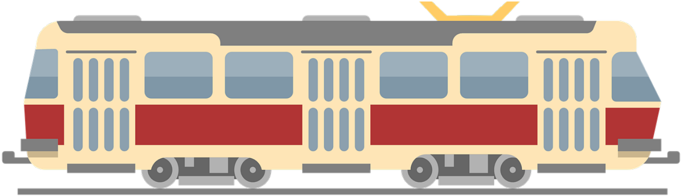
- Random variable: R - situation on rails
- Outcomes/events: $r \in \{$free rails, accindent, congestion$\}$
- Probability distribution: $P(R = \text{free rails}) = 0.3$, $P(R = \text{accident}) = 0.1$, $P(R = \text{congestion}) = 0.6$

Few reminders from laws of probability. Probabilities:

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

Random variables, probability distribution, ...

- Random variable - an event with unknown outcome
- Probability distribution - assignment of weights to the outcomes



- Random variable: R - situation on rails
- Outcomes/events: $r \in \{\text{free rails, accindent, congestion}\}$
- Probability distribution: $P(R = \text{free rails}) = 0.3$, $P(R = \text{accident}) = 0.1$, $P(R = \text{congestion}) = 0.6$

Few reminders from laws of probability, Probabilities:

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

Expectations, ...

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

- Depends on the random variable R - situation on rails with possible events r_1, r_2, r_3.
- What is the expectation of the time?

$$t = P(r_1)t_1 + P(r_2)t_2 + P(r_3)t_3$$

Weighted average.

Expectations, . . .

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

- Depends on the random variable R - situation on rails with possible events r_1, r_2, r_3.
- What is the expectation of the time?

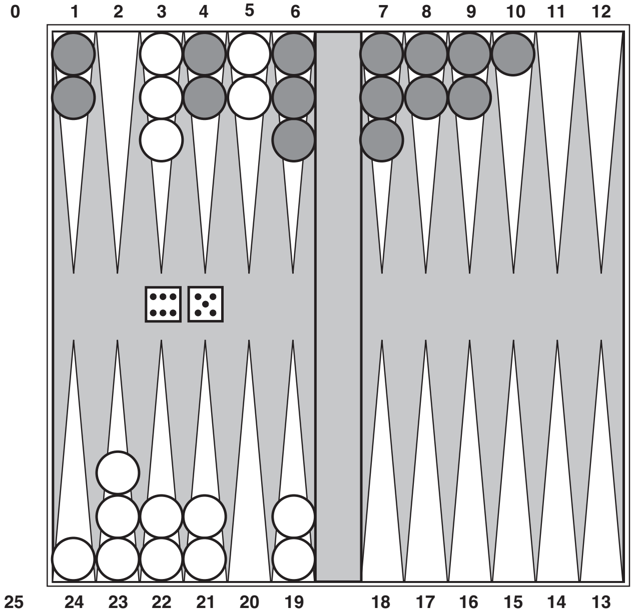
$$t = P(r_1)t_1 + P(r_2)t_2 + P(r_3)t_3$$

Weighted average.

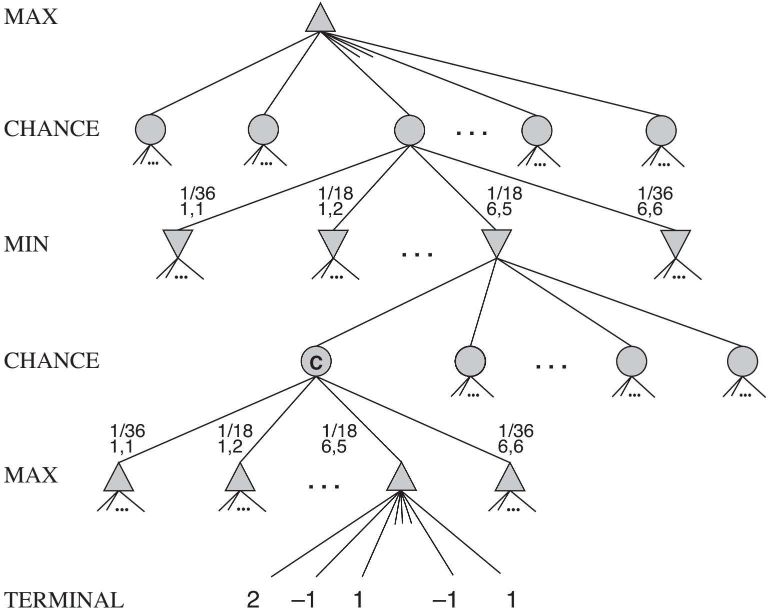
How about the Reversi game?

- Dangerous optimism
- Dangerous pessimism

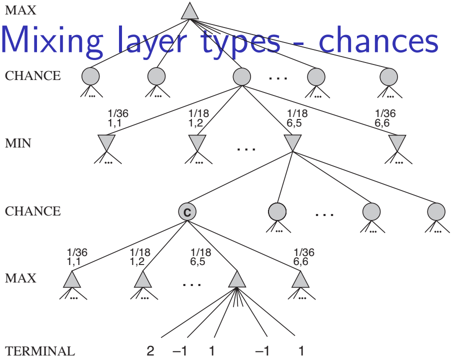
Games with chances **and** strategy

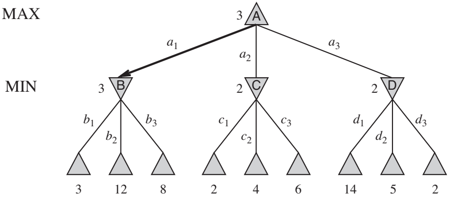


Mixing MAX, CHANCE, and MIN nodes



Mixing layer types - chances inserted

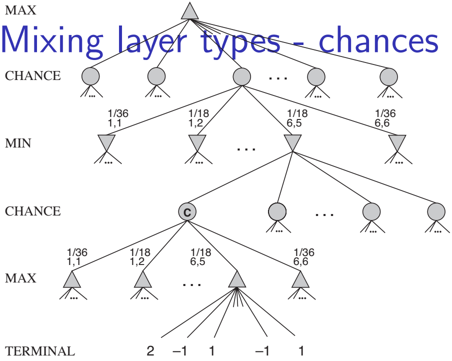


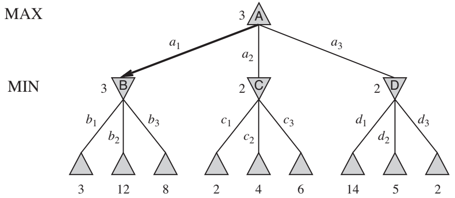


Extra random agent that moves after each MAX and MIN agent

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

Mixing layer types - chances inserted

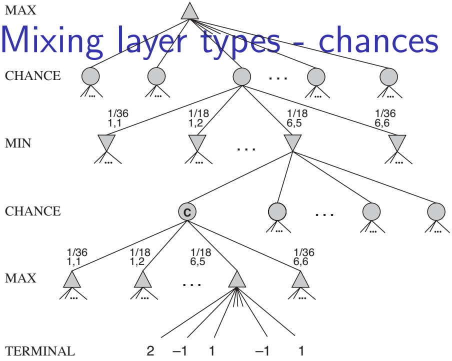


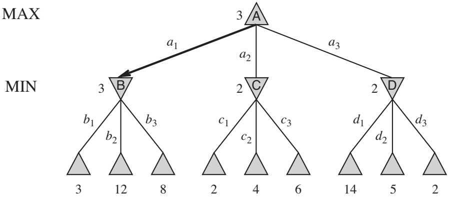


Extra random agent that moves after each MAX and MIN agent

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

Mixing layer types - chances inserted

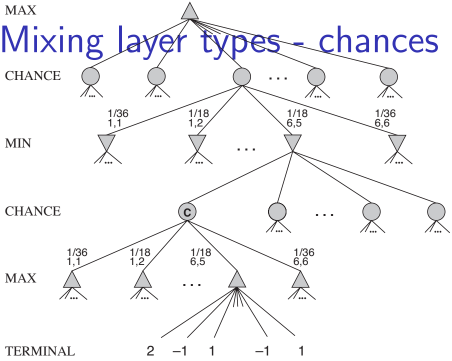


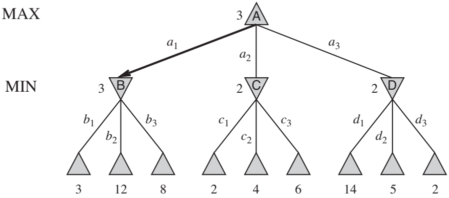


Extra random agent that moves after each MAX and MIN agent

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

Mixing layer types - chances inserted

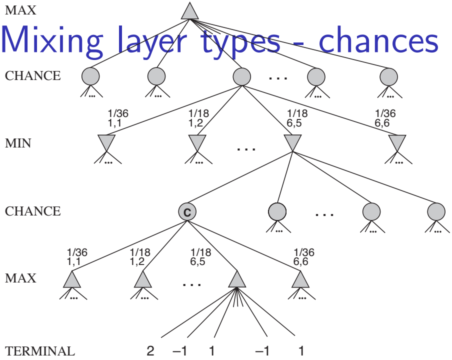


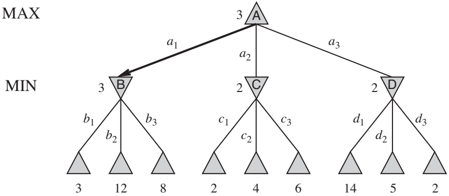


Extra random agent that moves after each MAX and MIN agent

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

Mixing layer types - chances inserted

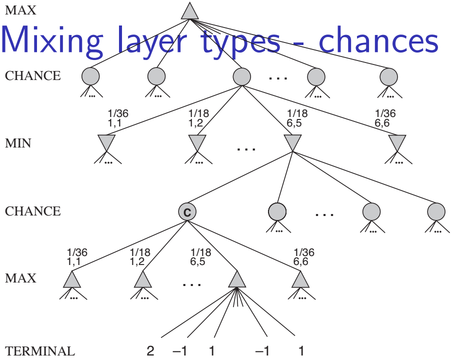


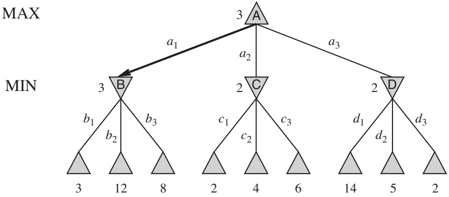


Extra random agent that moves after each MAX and MIN agent

$$\textsc{Expectiminimax}(s) = \begin{cases} \textsc{Utility}(s) & \text{if } \textsc{Terminal-Test}(s) \\ \max_a \textsc{Expectiminimax}(\textsc{Result}(s,a)) & \text{if } \textsc{Player}(s) = \textsc{Max} \\ \min_a \textsc{Expectiminimax}(\textsc{Result}(s,a)) & \text{if } \textsc{Player}(s) = \textsc{Min} \\ \sum_r P(r)\textsc{Expectiminimax}(\textsc{Result}(s,r)) & \text{if } \textsc{Player}(s) = \textsc{Chance} \end{cases}$$

Mixing layer types - chances inserted

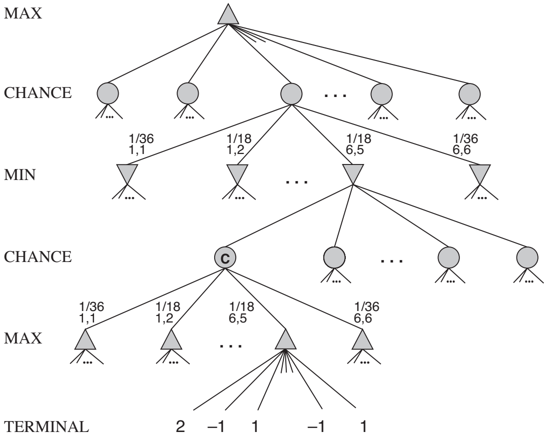




Extra random agent that moves after each MAX and MIN agent

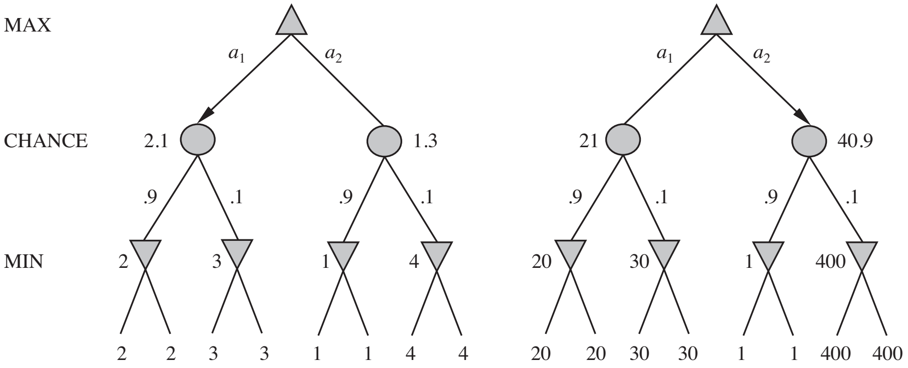
$$\text{EXPECTIMINIMAX}(s) = \begin{cases} \text{UTILITY}(s) & \text{if } \text{TERMINAL-TEST}(s) \\ \max_a \text{EXPECTIMINIMAX}(\text{RESULT}(s,a)) & \text{if } \text{PLAYER}(s) = \text{MAX} \\ \min_a \text{EXPECTIMINIMAX}(\text{RESULT}(s,a)) & \text{if } \text{PLAYER}(s) = \text{MIN} \\ \sum_r P(r)\,\text{EXPECTIMINIMAX}(\text{RESULT}(s,r))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

Mixing chances into min/max tree, how big?



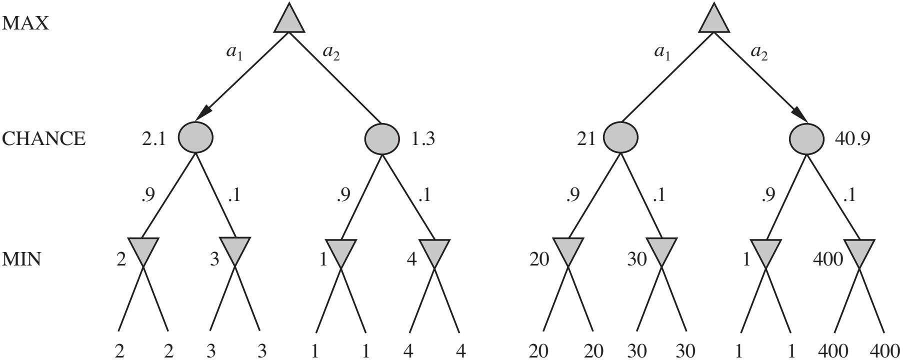
- b branching factor
- m maximumm depth
- n number of distinct rolls

Evaluation function



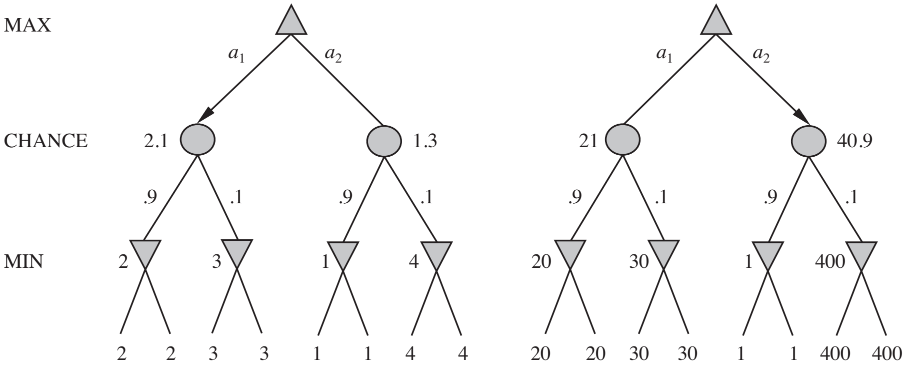
- Scale matters! Not only ordering.
- Can we prune the tree? (α, β like?)

Evaluation function



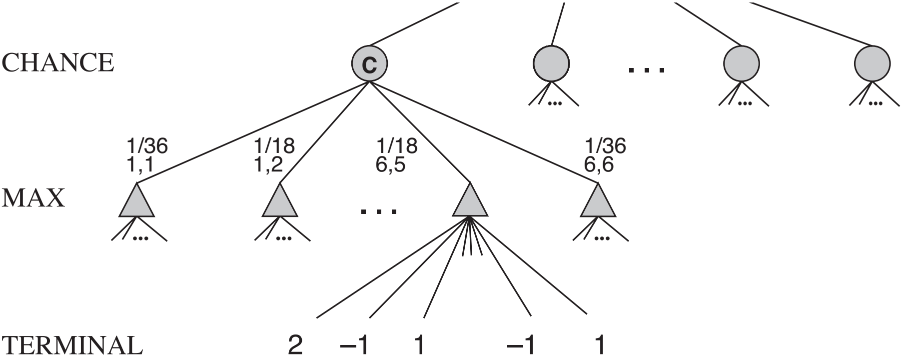
- Scale matters! Not only ordering.
- Can we prune the tree? (α, β like?)

Evaluation function



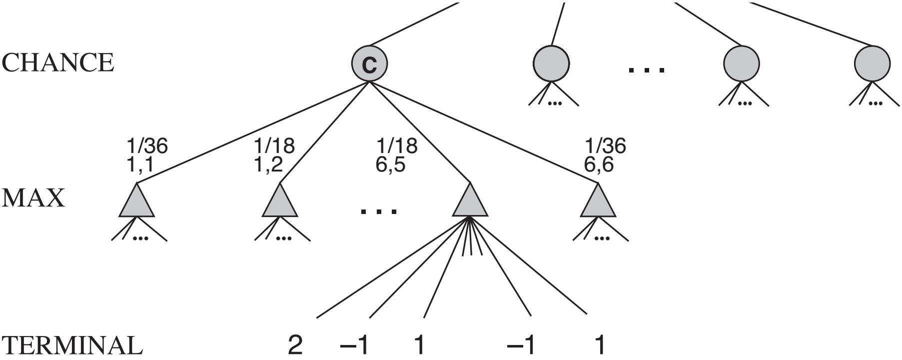
- Scale matters! Not only ordering.
- Can we prune the tree? (α, β like?)

Prunning expectiminimax tree



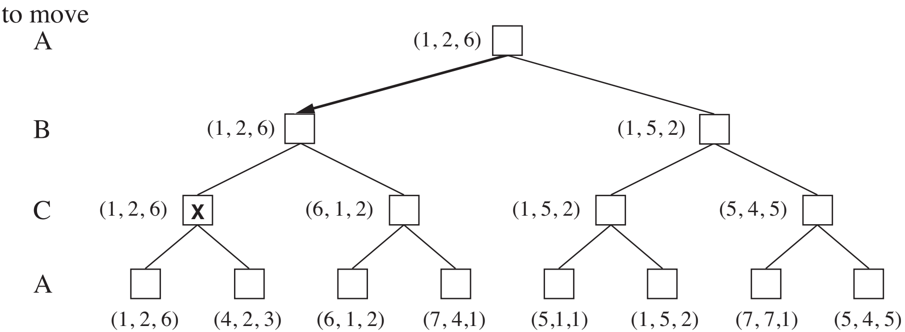
Bounds on terminal utilities needed.

Prunning expectiminimax tree



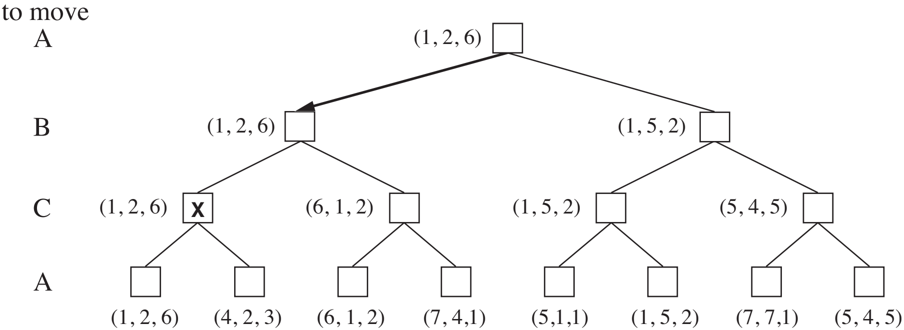
Bounds on terminal utilities needed.

Multi player games



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

Multi player games



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

Uncertainty recap

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

Uncertainty recap

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

Uncertain, partially observable environment

- Currents state s may be unknown, observations $\mathbf{e}$
- Uncertain outcome, random variable $\textsc{Result}(a)$
- Probability of outcome s' given $\mathbf{e}$ is

$$P(\textsc{Result}(a) = s' | a, \mathbf{e})$$

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

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

Amatrice, Italy, 2016.

Rational agent

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

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

What should a rational agent do?

Is it then all solved?

- $P(\textsc{Result}(a) = s'|a, \mathbf{e})$
- $U(s')$

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?

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

Rational agent

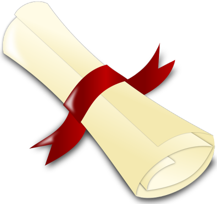
Agent's expected utility of an action a given $\mathbf{e}$:

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

What should a rational agent do?
Is it then all solved?

- $P(\textsc{Result}(a) = s'|a, \mathbf{e})$
- $U(s')$

Utilities

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

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: $A \succsim B$

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: $A \succsim B$

Rational preferences

- Transitivity
- Orderability (Completeness)
- Continuity
- Substituability
- Monotonocity
- Decomposability (Reduction)

Axioms of utility theory.

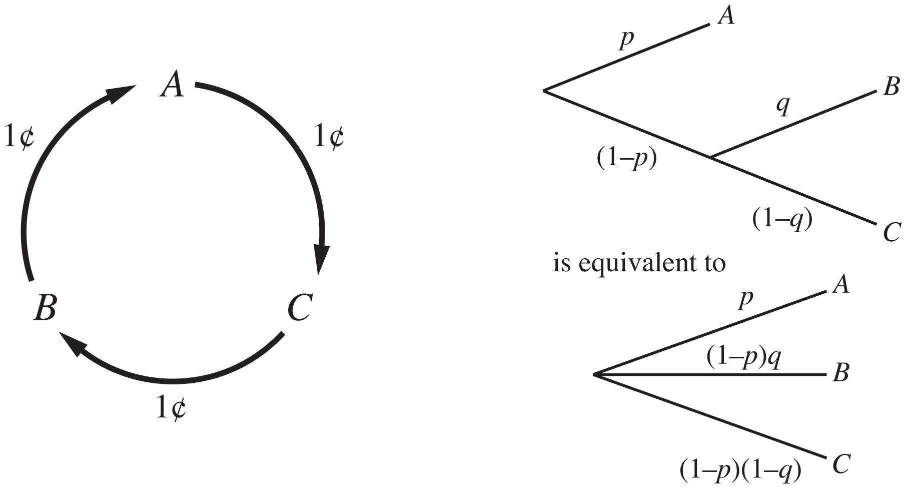
Rational preferences

- Transitivity
- Orderability (Completeness)
- Continuity
- Substituability
- Monotonocity
- Decomposability (Reduction)

Axioms of utility theory.

Transitivity and decomposability

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



(a) (b)

Maximum expected utility principle

Given the rational preferences (contraints), there exists a real valued function U such that:

$$U(A) > U(B) \quad \Leftrightarrow \quad A \succ B$$
$$U(A) = U(B) \quad \Leftrightarrow \quad A \sim B$$

Expected utility of a Lotery L:

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

Proof in [3].
Is a utility U unique?

Maximum expected utility principle

Given the rational preferences (contraints), there exists a real valued function U such that:

$$
\begin{aligned}
U(A) > U(B) &\Leftrightarrow A \succ B \\
U(A) = U(B) &\Leftrightarrow A \sim B
\end{aligned}
$$

Expected utility of a Lotery L:

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

Proof in [3].

Is a utility U unique?

Maximum expected utility principle

Given the rational preferences (contraints), there exists a real valued function U such that:

$$U(A) > U(B) \quad \Leftrightarrow \quad A \succ B$$
$$U(A) = U(B) \quad \Leftrightarrow \quad A \sim B$$

Expected utility of a Lotery L:

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

Proof in [3].
Is a utility U unique?

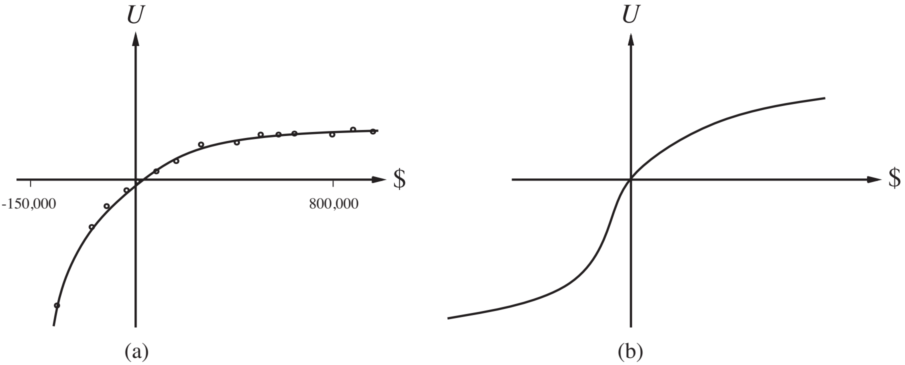
Human utilities

Utility of money

You triumphed in a TV show!

a) Take $\$1,000,000$ ... or

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



The (human) Utility of money.

References

Some figures from [2], Chapters 5, 16. Human utilities are discussed in [1]. This lecture has been also greatly inspired by the 7th lecture of CS 188 at `http://ai.berkeley.edu` as it convenietly bridges the world of deterministic search and sequential decisions in uncertain worlds.

[1] Daniel Kahneman.
Thinking, Fast and Slow.
Farrar, Straus and Giroux, 2011.

[2] Stuart Russell and Peter Norvig.
Artificial Intelligence: A Modern Approach.
Prentice Hall, 3rd edition, 2010.
`http://aima.cs.berkeley.edu/`.

[3] 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`, Utility theorem.