CS 1571 Introduction to AI
Lecture 26

Decision making in the presence of uncertainty

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Administration

- **Solutions to HW-9 and HW-10**
  - **will be posted on the course web page**
- **Office hours:**
  - **Milos: Thursday 2:00-3:30pm**
  - **Swapna: Thursday 5:45 pm - 7:15 pm**
- **Final exam:**
  - December 11, 2006
  - 12:00-1:50pm, 5129 Sennott Square

Decision-making in the presence of uncertainty

- Computing the probability of some event may not be our ultimate goal
- Instead we are often interested in **making decisions about our future actions so that we satisfy goals**
- **Example: medicine**
  - Diagnosis is typically only the first step
  - The ultimate goal is to manage the patient in the best possible way. Typically many options available:
    - Surgery, medication, collect the new info (lab test)
    - There is an **uncertainty in the outcomes** of these procedures: patient can be improve, get worse or even die as a result of different management choices.

Decision-making in the presence of uncertainty

Main issues:

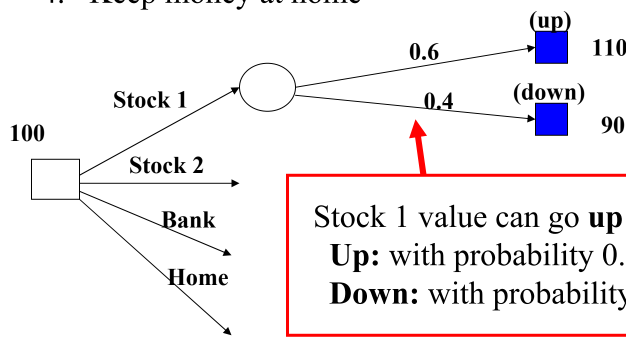
- **How to model the decision process with uncertain outcomes in the computer ?**
- **How to make decisions about actions in the presence of uncertainty?**

The field of **decision-making** studies ways of making decisions in the presence of uncertainty.

Decision making example.

Assume we want to invest $100 for 6 months

- **We have 4 choices:**
  1. Invest in Stock 1
  2. Invest in Stock 2
  3. Put money in bank
  4. Keep money at home



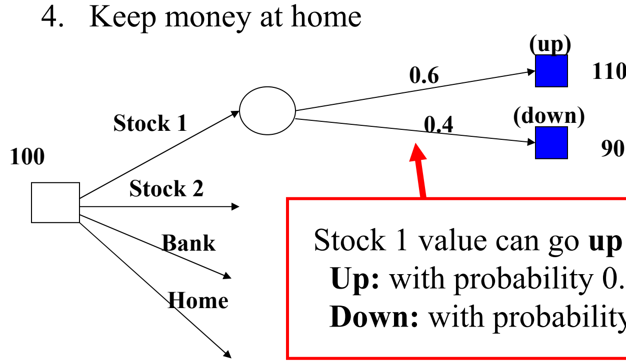
Stock 1 value can go **up** or **down**:
Up: with probability 0.6
Down: with probability 0.4

Decision making example.

Assume we want to invest $100 for 6 months

- **We have 4 choices:**
  1. Invest in Stock 1
  2. Invest in Stock 2
  3. Put money in bank
  4. Keep money at home

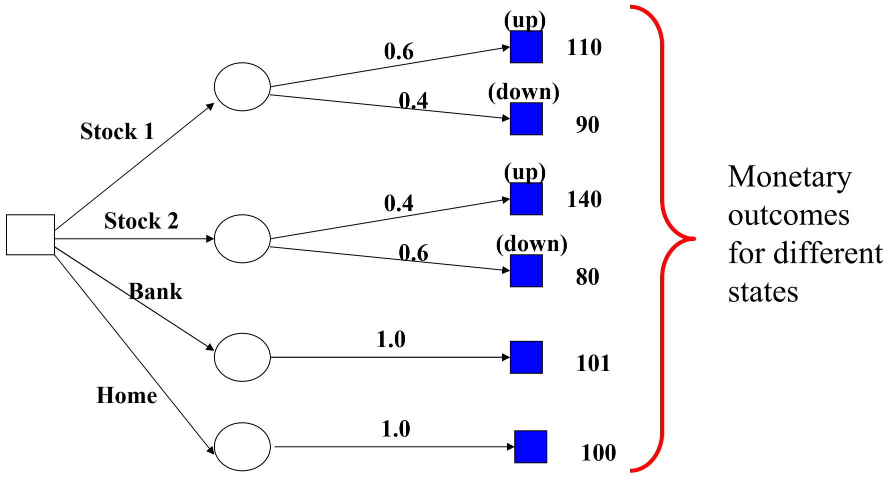
Monetary Outcomes for up and down states



Stock 1 value can go **up** or **down**:
Up: with probability 0.6
Down: with probability 0.4

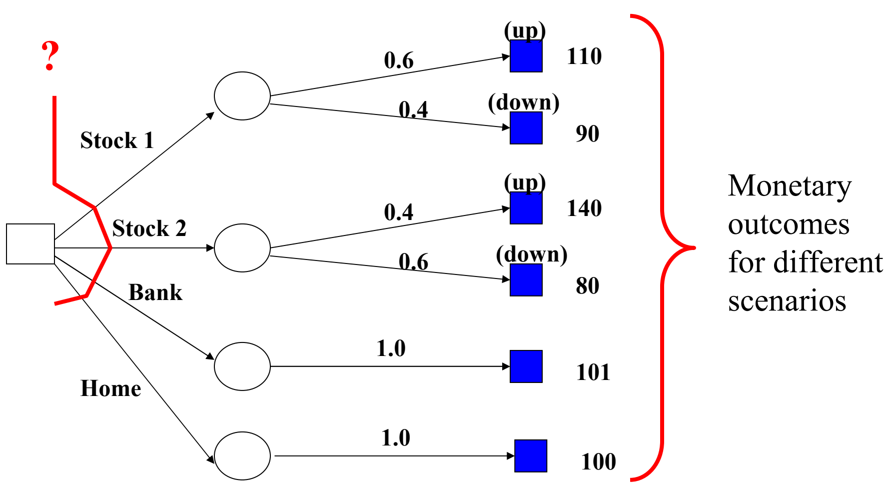
Decision making example.

Investing of $100 for 6 months



Decision making example.

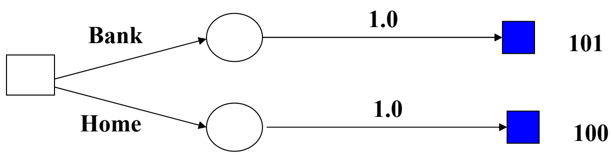
We need to make a choice whether to invest in Stock 1 or 2, put money into bank or keep them at home. But how?



Decision making example.

Assume the simplified problem with the Bank and Home choices only.

The result is guaranteed – the outcome is deterministic

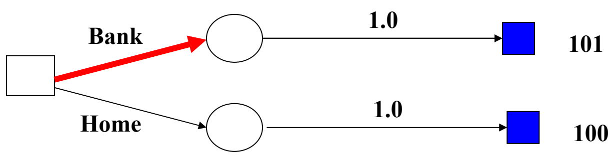


What is the rational choice assuming our goal is to make money?

Decision making. Deterministic outcome.

Assume the simplified problem with the Bank and Home choices only.

These choices are deterministic.



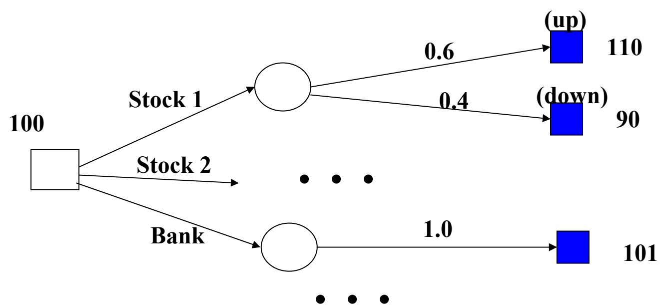
Our goal is to make money. What is the rational choice?

Answer: **Put money into the bank. The choice is always strictly better in terms of the outcome**

But what to do if we have uncertain outcomes?

Decision making. Stochastic outcome

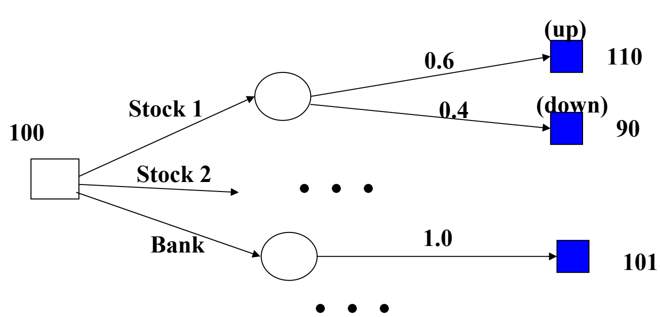
- **How to quantify the goodness of the stochastic outcome?**
  We want to compare it to deterministic and other stochastic outcomes.



?

Decision making. Stochastic outcome

- **How to quantify the goodness of the stochastic outcome?**
  We want to compare it to deterministic and other stochastic outcomes.



Idea: Use expected value of the outcome

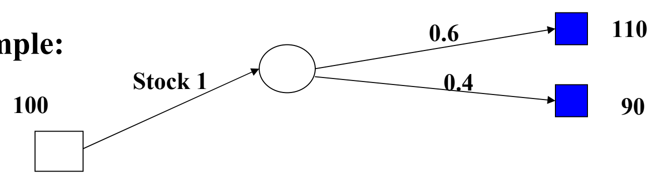
Expected value

- Let X be a random variable representing the monetary outcome with a discrete set of values Ω_X.
- **Expected value** of X is:

$$E(X) = \sum_{x \in \Omega_X} x P(X = x)$$

Intuition: Expected value summarizes all stochastic outcomes into a single quantity.

- **Example:**



- What is the expected value of the outcome of Stock 1 option?

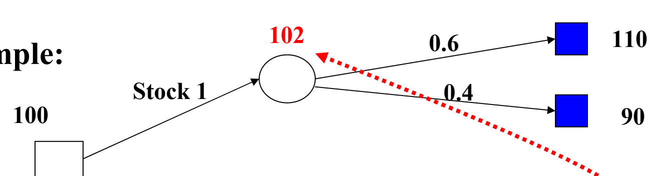
Expected value

- Let X be a random variable representing the monetary outcome with a discrete set of values Ω_X.
- **Expected value** of X is:

$$E(X) = \sum_{x \in \Omega_X} x P(X = x)$$

  - **Expected value** summarizes all stochastic outcomes into a single quantity

- **Example:**

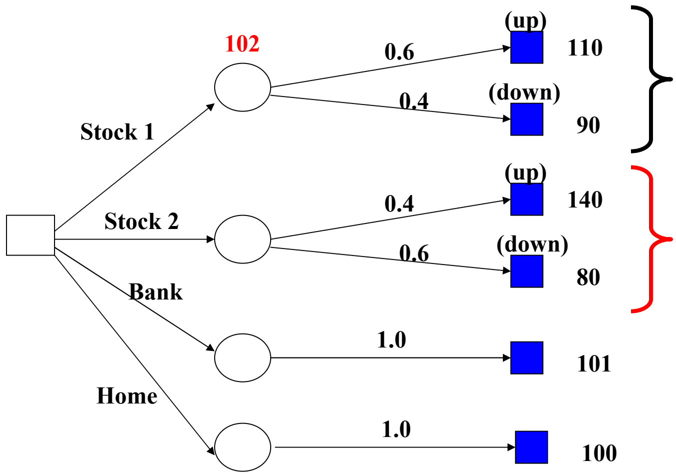


Expected value for the outcome of the Stock 1 option is:

$$0.6 \times 110 + 0.4 \times 90 = 66 + 36 = 102$$

Expected values

Investing $100 for 6 months

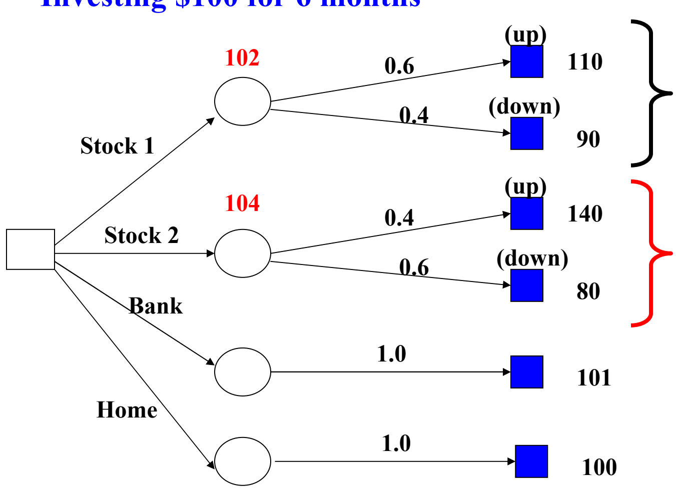


$0.6 \times 110 + 0.4 \times 90 =$
$66 + 36 = 102$

?

Expected values

Investing $100 for 6 months

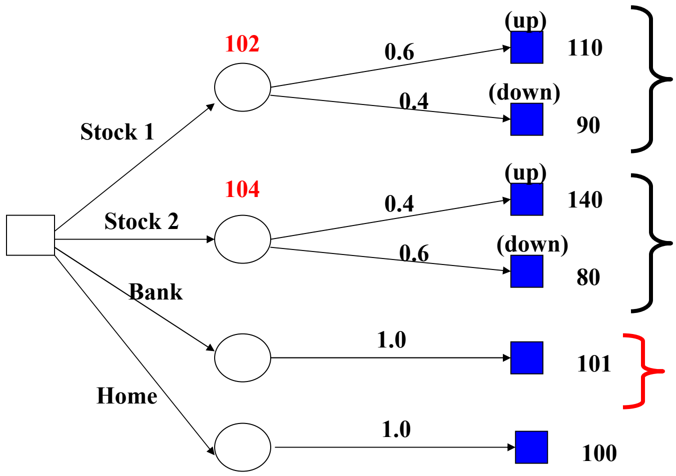


$0.6 \times 110 + 0.4 \times 90 =$
$66 + 36 = 102$

$0.4 \times 140 + 0.6 \times 80 =$
$56 + 48 = 104$

Expected values

Investing $100 for 6 months



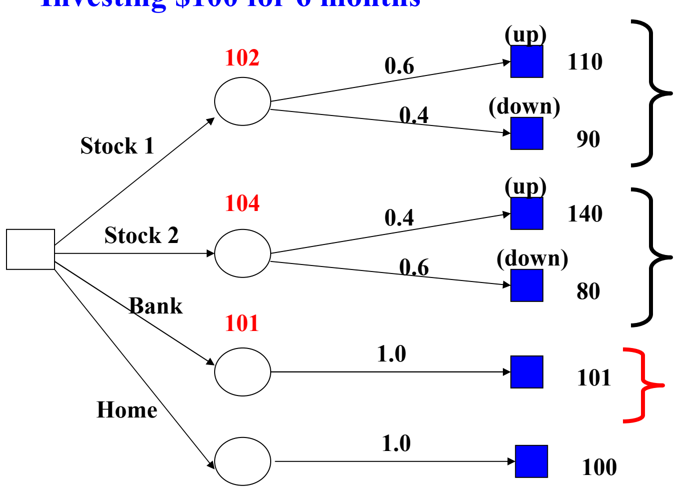
$0.6 \times 110 + 0.4 \times 90 =$
$66 + 36 = 102$

$0.4 \times 140 + 0.6 \times 80 =$
$56 + 48 = 104$

?

Expected values

Investing $100 for 6 months



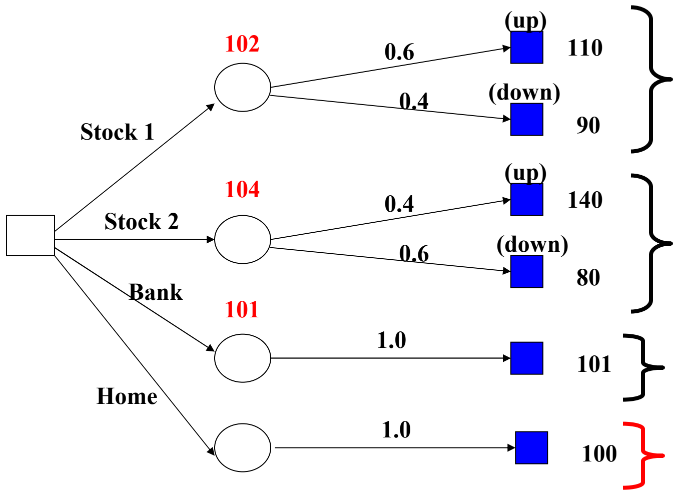
$0.6 \times 110 + 0.4 \times 90 =$
$66 + 36 = 102$

$0.4 \times 140 + 0.6 \times 80 =$
$56 + 48 = 104$

$1.0 \times 101 = 101$

Expected values

Investing $100 for 6 months



$0.6 \times 110 + 0.4 \times 90 = 66 + 36 = 102$

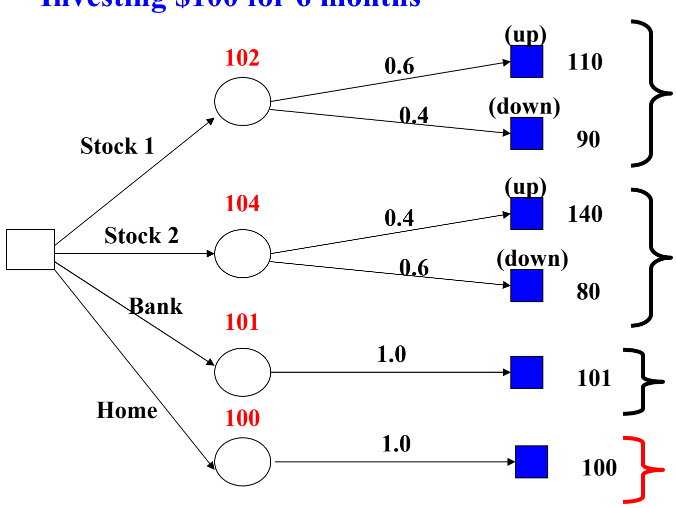
$0.4 \times 140 + 0.6 \times 80 = 56 + 48 = 104$

$1.0 \times 101 = 101$

?

Expected values

Investing $100 for 6 months



$0.6 \times 110 + 0.4 \times 90 = 66 + 36 = 102$

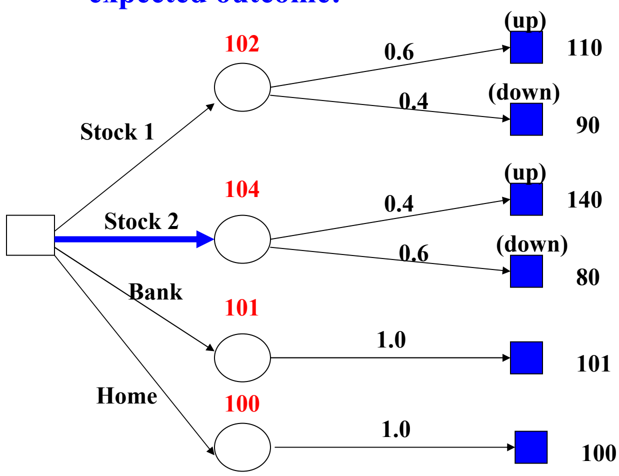
$0.4 \times 140 + 0.6 \times 80 = 56 + 48 = 104$

$1.0 \times 101 = 101$

$1.0 \times 100 = 100$

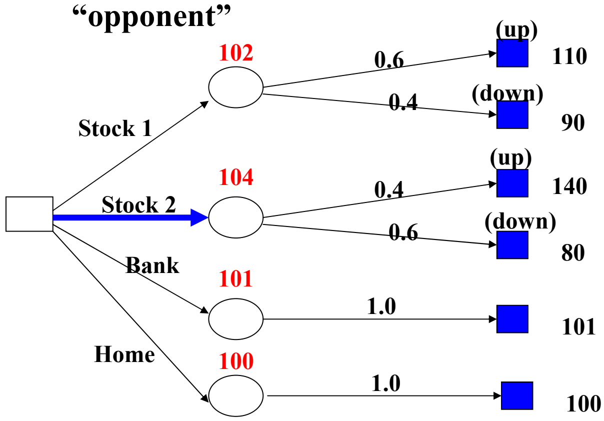
Selection based on expected values

The optimal action is the option that maximizes the expected outcome:



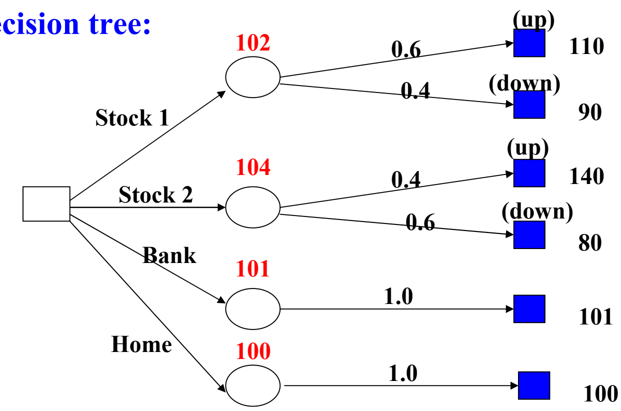
Relation to the game search

- Game search: minimax algorithm
  - considers the rational opponent and its best move
- Decision making: maximizes the expectation
  - play against the nature - stochastic non-malicious "opponent"



(Stochastic) Decision tree

- **Decision tree:**



- **decision node**
- **chance node**
- **outcome (value) node**

Sequential (multi-step) problems

The decision tree can be build to capture multi-step decision problems:

- Choose an action
- Observe the stochastic outcome
- And repeat

How to make decisions for multi-step problems?

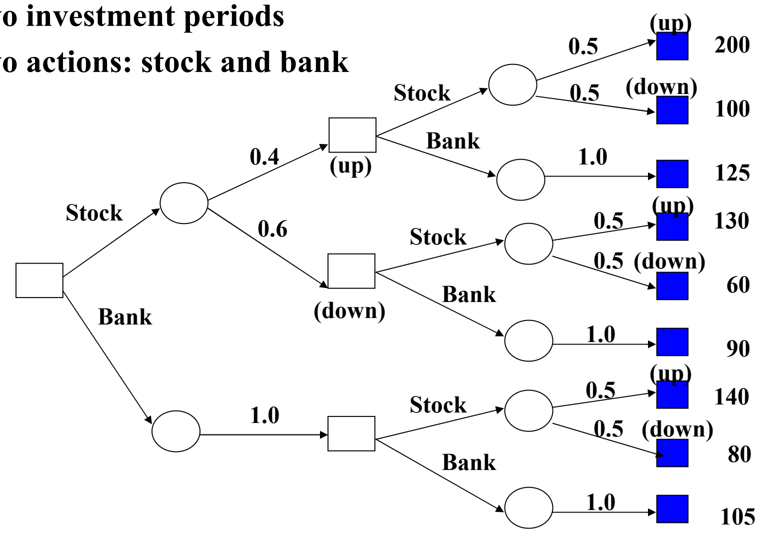
- Start from the leaves of the decision tree (outcome nodes)
- Compute expectations at chance nodes
- Maximize at the decision nodes

Algorithm is sometimes called **expectimax**

Multi-step problem example

Assume:

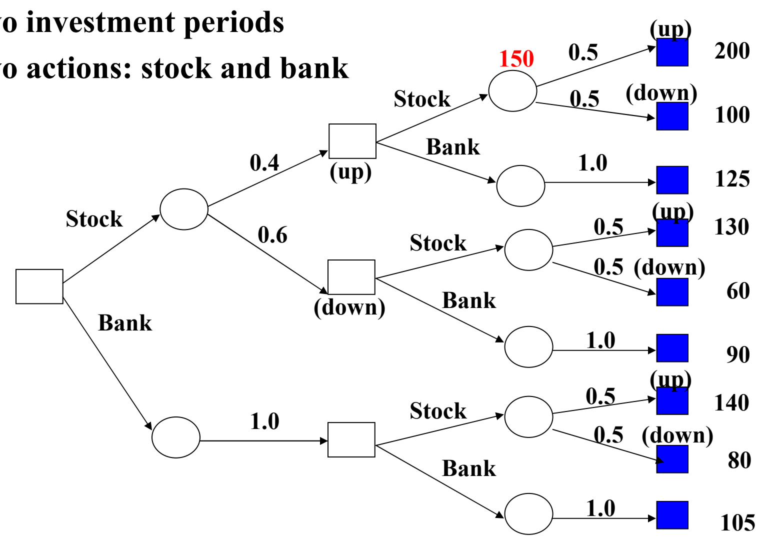
- Two investment periods
- Two actions: stock and bank



Multi-step problem example

Assume:

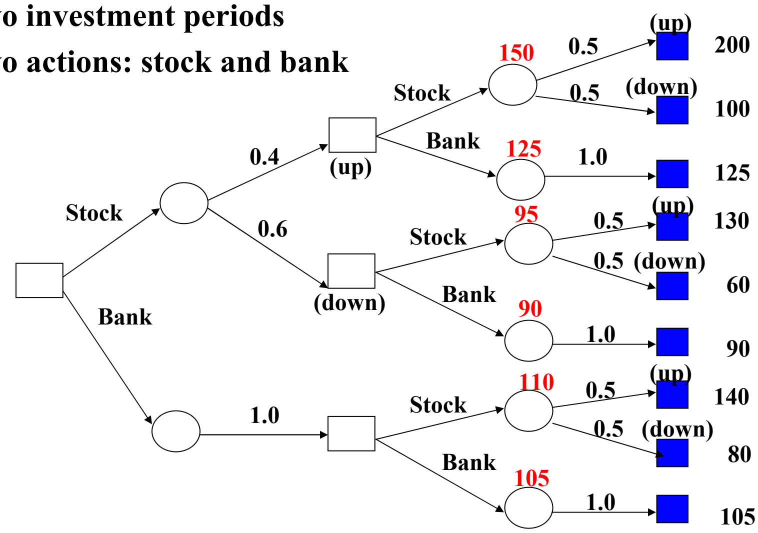
- Two investment periods
- Two actions: stock and bank



Multi-step problem example

Assume:

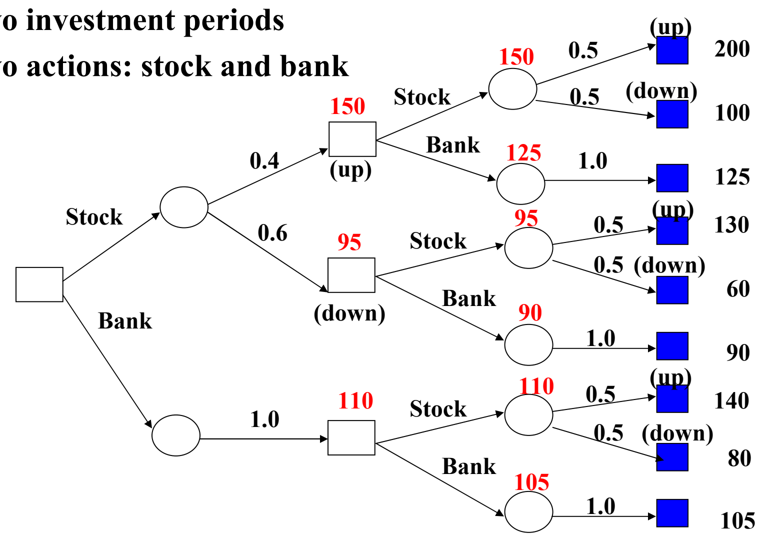
- Two investment periods
- Two actions: stock and bank



Multi-step problem example

Assume:

- Two investment periods
- Two actions: stock and bank

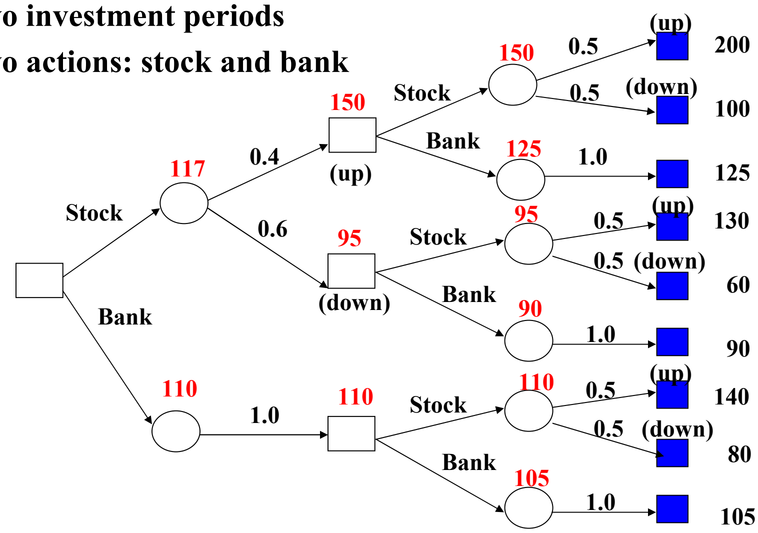


CS 1571 Intro to AI
M. Hauskrecht

Multi-step problem example

Assume:

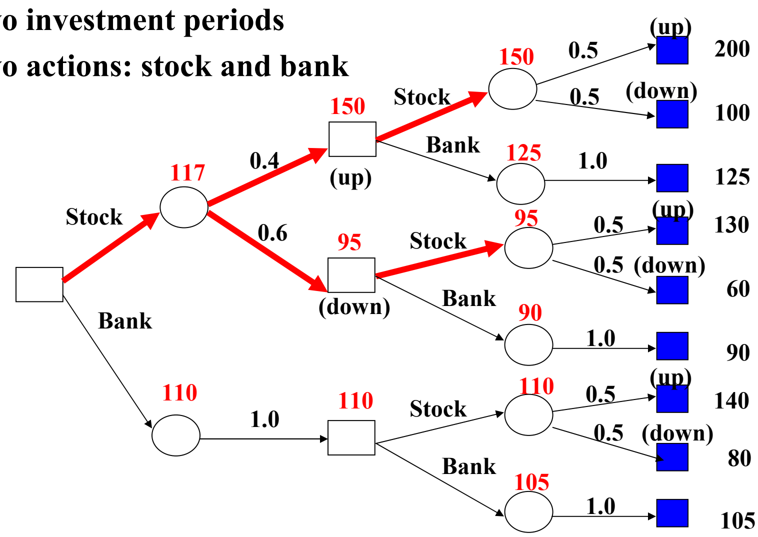
- Two investment periods
- Two actions: stock and bank



Multi-step problem example

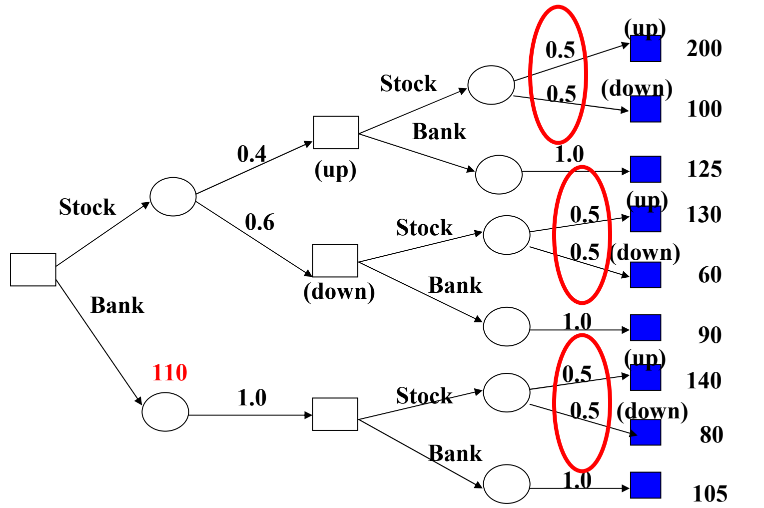
Assume:

- Two investment periods
- Two actions: stock and bank



Multi-step problems. Conditioning.

- Notice that the probability of stock going up and down in the 2nd step is independent of the 1st step (=0.5)



Conditioning in the decision tree

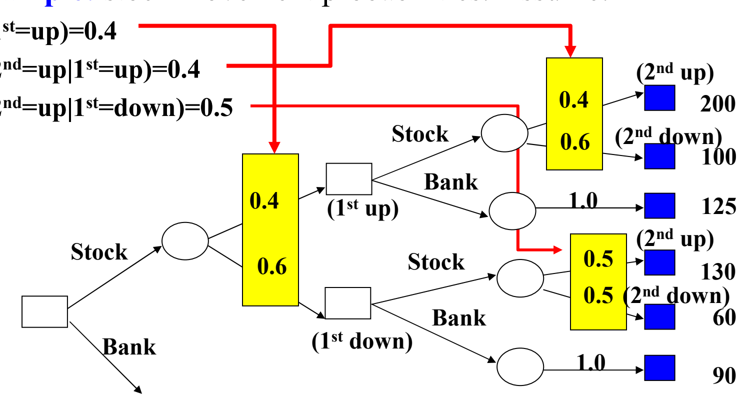
- **But this may not be the case. In decision trees:**
  - **Later outcomes can be conditioned on the earlier stochastic outcomes and actions**

Example: stock movement probabilities. Assume:

P(1st=up)=0.4

P(2nd=up|1st=up)=0.4

P(2nd=up|1st=down)=0.5



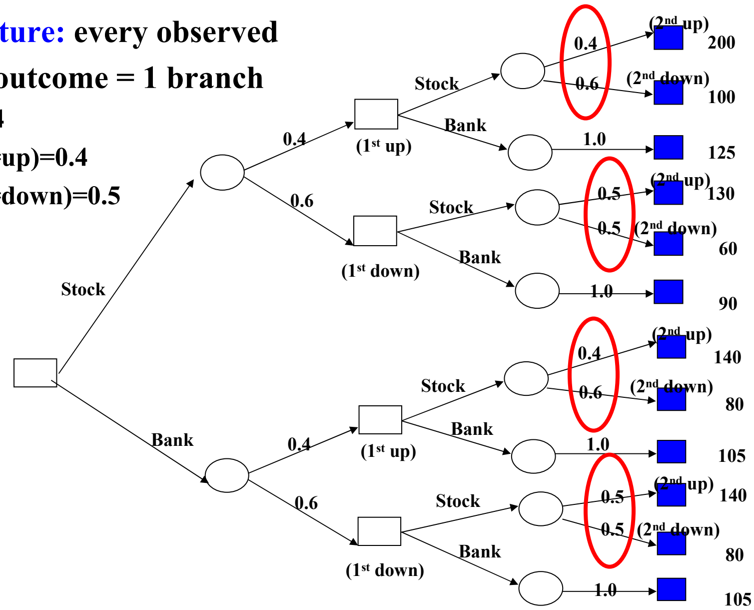
Multi-step problems. Conditioning.

Tree Structure: every observed stochastic outcome = 1 branch

P(1st=up)=0.4

P(2nd=up|1st=up)=0.4

P(2nd=up|1st=down)=0.5



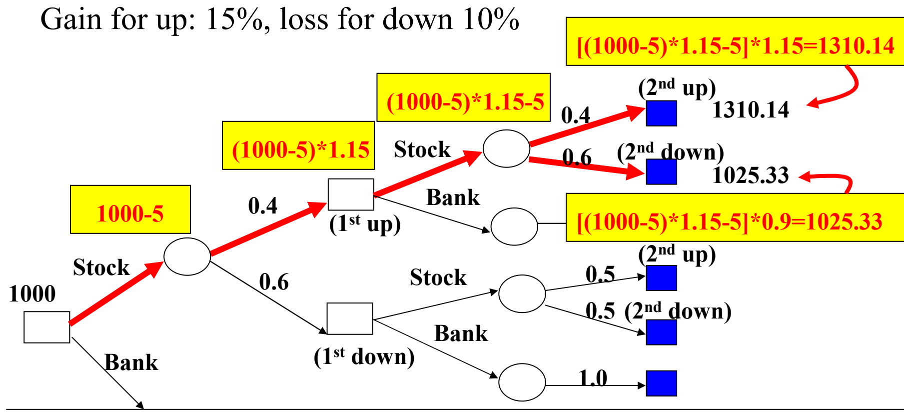
Trajectory payoffs

- **Outcome values at leaf nodes (e.g. monetary values)**
  - **Rewards and costs for the path trajectory**

Example: stock fees and gains. **Assume:**

Fee per period: $5 paid at the beginning

Gain for up: 15%, loss for down 10%



Constructing a decision tree

- **The decision tree is rarely given to you directly.**
  - Part of the problem is to construct the tree.

Example: stocks, bonds, bank for k periods

Stock:
- Probability of stocks going up in the first period: 0.3
- Probability of stocks going up in subsequent periods:
  - P(kth step=Up| (k -1)th step =Up)=0.4
  - P(kth step =Up| (k -1)th step=Down)=0.5
- Return if stock goes up: 15 % if down: 10%
- Fixed fee per investment period: $5

Bonds:
- Probability of value up: 0.5, down: 0.5
- Return if bond value is going up: 7%, if down: 3%
- Fee per investment period: $2

Bank:
- Guaranteed return of 3% per period, no fee

Review

KR and logic

- **Knowledge representation:**
  - **Syntax** (how sentences are build), **Semantics** (meaning of sentences), **Computational aspect** (how sentences are manipulated)
- **Logic:**
  - A formal language for expressing knowledge and ways of reasoning
  - **Three components:**
    - A set of sentences
    - A set of interpretations
    - The valuation (meaning) function

Propositional logic

- A language for symbolic reasoning
- **Language:**
  - Syntax, Semantics
- **Satisfiability** of a sentence: at least one interpretation under which the sentence can evaluate to ***True***.
- **Validity** of a sentence: ***True*** in all interpretations
- **Entailment:** $KB \models \alpha$

  α is true in all worlds in which KB is true
- **Inference procedure**
  - Soundness If $KB \vdash_i \alpha$ then $KB \models \alpha$
  - Completeness If $KB \models \alpha$ then $KB \vdash_i \alpha$

Propositional logic

- **Logical inference problem:** $KB \models \alpha$?
  - Does KB entail the sentence α ?
- Logical inference problem for the propositional logic is **decidable**.
  - A procedure (program) that stops in finite time exists
- **Approaches:**
  - Truth table approach
  - Inference rule approach
  - Resolution refutation

$KB \models \alpha$	if and only if
$(KB \wedge \neg \alpha)$	is **unsatisfiable**

- **Normal forms**: DNF, CNF, Horn NF (conversions)

First order logic

- Deficiencies of propositional logic
- **First order logic (FOL):** allows us to represent objects, their properties, relations and statements about them
  - Variables, predicates, functions, quantifiers
  - Syntax and semantics of the sentences in FOL
- **Logical inference problem** $KB \models \alpha$?
  - **Undecidable.** No procedure that can decide the entailment for all possible input sentences in a finite number of steps.
- **Inference approaches:**
  - Inference rules
  - Resolution refutation

First order logic

- **Methods for making inferences work with variables:**
  - **Variable substitutions**
  - **Unification** process that takes two similar sentences and computes the substitution that makes that makes them look the same, if it exists

- **Conversions to CNF** with universally quantified variables
  - Used by resolution refutation
    - The procedure is refutation- complete

Knowledge-based systems with HNF

- **KBs in Horn normal form:**
  - Not all sentences in FOL can be translated to HNF
  - Modus ponens is complete for Horn databases

- **Inferences** with KBs in Horn normal form (HNF)
  - Forward chaining
  - Backward chaining

- **Production systems**
  - Conflict resolution

Planning

- **Find a sequence of actions** that lead to a goal
  - Much like path search, but for very large domains
  - Need to represent the dynamics of the world
- **Two basic approaches** planning problem representation:
  - **Situation calculus**
    - Explicitly represents situations (extends FOL)
    - **Solving:** theorem proving
  - **STRIPS**
    - Add and delete list
    - **Solving:** Search
      (Goal progression, Goal regression)
- **Frame problem**

Planning

- **Divide and conquer approach** (Sussman's anomaly)
- **State space vs. plan space search**
- **Partial order (non-linear) planners:**
  - Search the space of partially build plans
  - Progressive or regressive mode
- **Hierarchical planners**

Uncertainty

- **Basics of probability:**
  - random variable, values, probability distribution
- **Joint probability distribution**
  - Over variables in a set, **full joint** over all variables
  - Marginalization (summing out)
- **Conditional probability distribution**

$$P(A \mid B) = \frac{P(A, B)}{P(B)} \text{ s.t. } P(B) \neq 0$$

- **Product rule** $P(A, B) = P(A \mid B)P(B)$
- **Chain rule**
- **Bayes rule**

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Uncertainty

Full joint probability distribution

- Over variables in a set, **full joint** over all variables

Two important things to remember:

- Any probabilistic query can be computed from the full joint distribution
- Full joint distribution can be expressed as a product of conditionals via the chain rule

Bayesian belief networks

- **Full joint distribution** over all random variables defining the domain can be very large
  – Complexity of a model, inferences, acquisition
- **Solution:** Bayesian belief networks (BBNs)
- **Two components of BBNs:**
  – Structure (directed acyclic graph)
  – Parameters (conditional prob. distributions)
- **BBN** build upon conditional independence relations:

$$P(A,B \mid C) = P(A \mid C)P(B \mid C)$$

- **Joint probability distribution for BBNs**:
  – Product of local (variable-parents) conditionals

$$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

Bayesian belief networks

- **Model of joint distribution:**
  – Reduction in the number of parameters
- **Inferences:**
  – Queries on joint probabilities
  – Queries on conditionals expressed as ratios of joint probabilities
  – Joint probabilities can be expressed in terms of full joints
  – Full joints are product of local conditionals
- **Smart way to do inferences:**
  – Interleave sums and products (variable elimination)

Decision-making in the presence of uncertainty

- **Decision tree:**
  - Decision nodes (choices are made)
  - Chance nodes (reflect stochastic outcome)
  - Outcomes (value) nodes (value of the end-situation)
- **Rational choice:**
  - Decision-maker tries to optimize the expected value