Decision Analysis

CS4246/CS5446

AI Planning and Decision Making

Extra Slides

Expected Value of Imperfect Information

Perfect Information

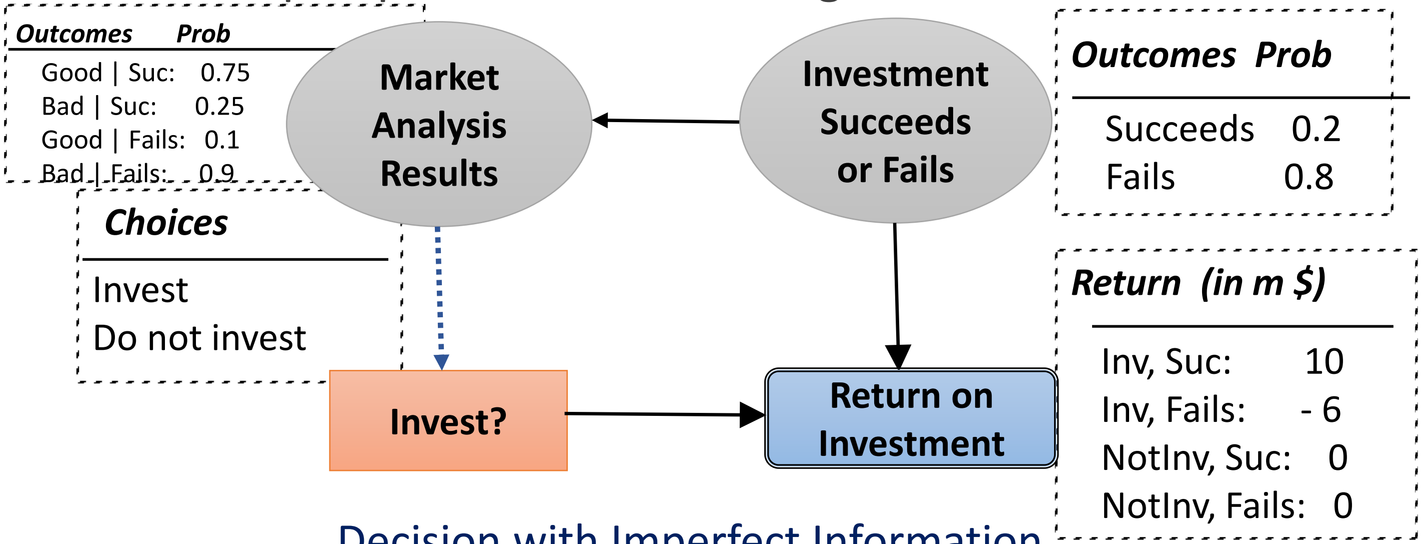
- Definition:
  - A knowledge source, or expert's information is perfect if it is always correct
  - When state S will occur, the expert always says so (and never says that some other state will occur)
  - P(Say "Good" | Event is good) = 1
  - Use conditional probabilities to model perfect information

Expected Value of Imperfect Information

- Max amount that the decision maker is willing to pay for hiring economist or imperfect information
- To find expected value of imperfect information (EVII):
  - Value collecting some information from a sample
    - Example: Hire economist to forecast stock market trends
  - Modify decision model to include imperfect information
  - Solve the model and find the EMV ($822)
  - EVII = EMV (with imperfect information) - EMV (original)
    = $822 - $580 = $242

Imperfect Information

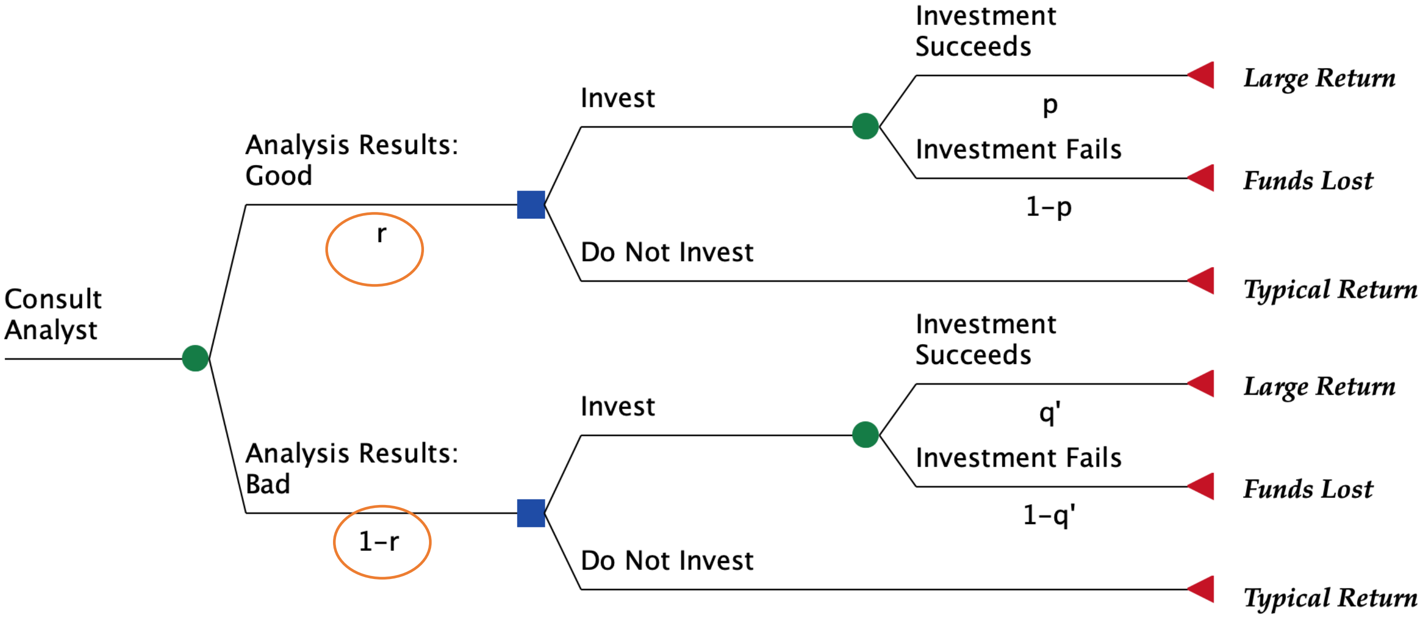
- Decision maker will obtain information before making decision
  - e.g., The investor hires a consulting firm to do an analysis of the investment prospects before deciding to invest



Decision with Imperfect Information

Imperfect Information

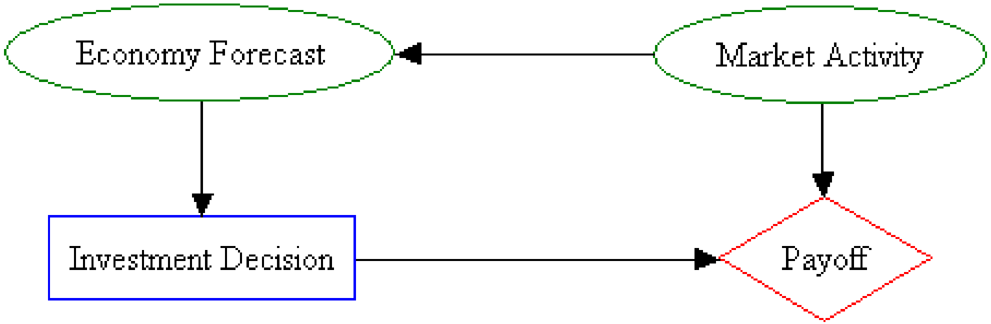
- Decision maker will obtain information before making a decision



Modifying ID for EVII

- Insert uncertain event node for forecast

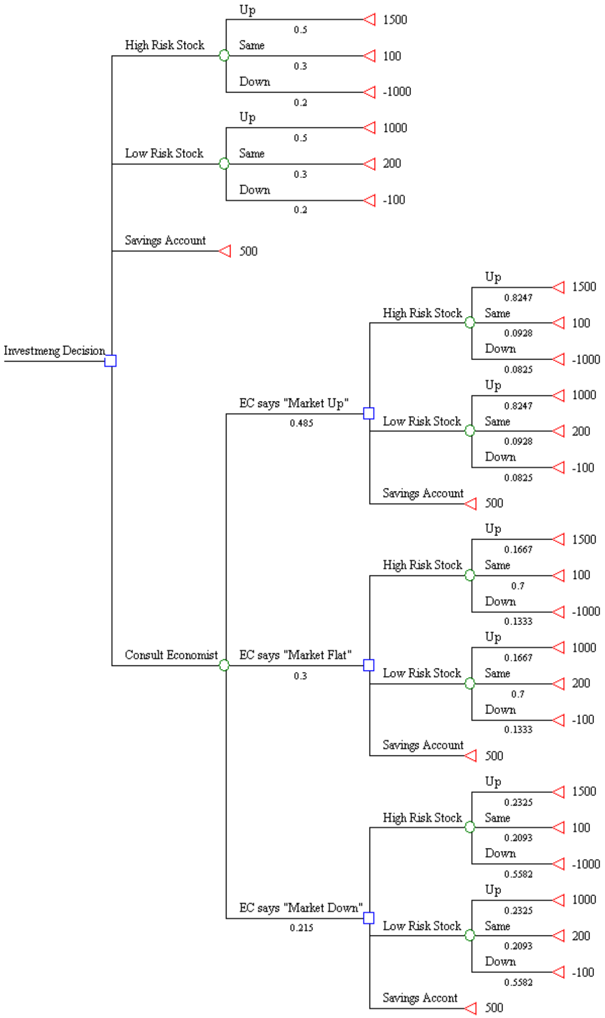
	Economist says Up	Flat	Down
M. Up	0.80	0.10	0.10
M. Flat	0.15	0.70	0.15
M. Down	0.20	0.20	0.60



Influence diagram with imperfect information

Modifying DT for EVII

- Insert uncertain event node
- Calculate missing probabilities

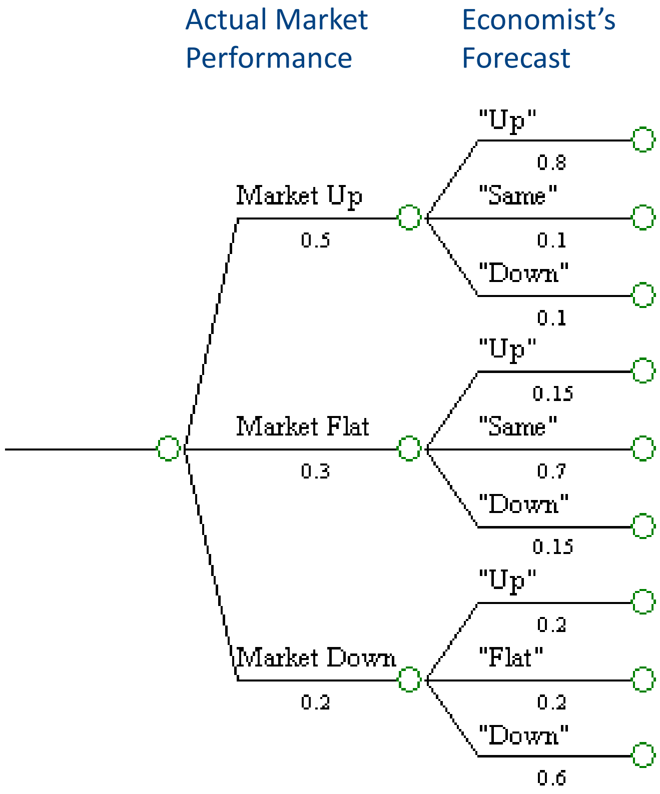


Stock Investment Example (cont.)

- Calculating EVII in:
  - influence diagram -- simple and clear
  - decision tree -- full details with proper orders
- Examples: The stock investment problem
  - “Market Activity” node follows economist’s forecast
  - Use Bayes’ theorem to find posterior probabilities reflected in the new decision tree structure
  - We have:
    - P (Economist says “Up” | Market Up)
    - P (Economist says “Flat” | Market Up)
    - P (Economist says “Down” | Market Up)
    - ...
  - We want:
    - P (Market Up | Economist says “Up”)
    - P (Market Flat | Economist says “Up”)
    - P (Market Down | Economist says “Up”)
    - ...

The Original Probability Tree

	Economist says		
	Up	Flat	Down
M. Up	0.80	0.10	0.10
M. Flat	0.15	0.70	0.15
M. Down	0.20	0.20	0.60



Flipping the Probability Tree

P(Market Up | Exp says "Up")

$$= \frac{P(\text{"Up"} \mid Up)\,P(Up)}{[P(\text{"Up"}|Up)P(Up) + P(\text{"Up"}|Flat)P(Flat) + P(\text{"Up"}|Down)P(Down)]}$$

$$= \frac{0.8(0.5)}{0.8(0.5)+0.15(0.3)+0.2(0.2)}$$

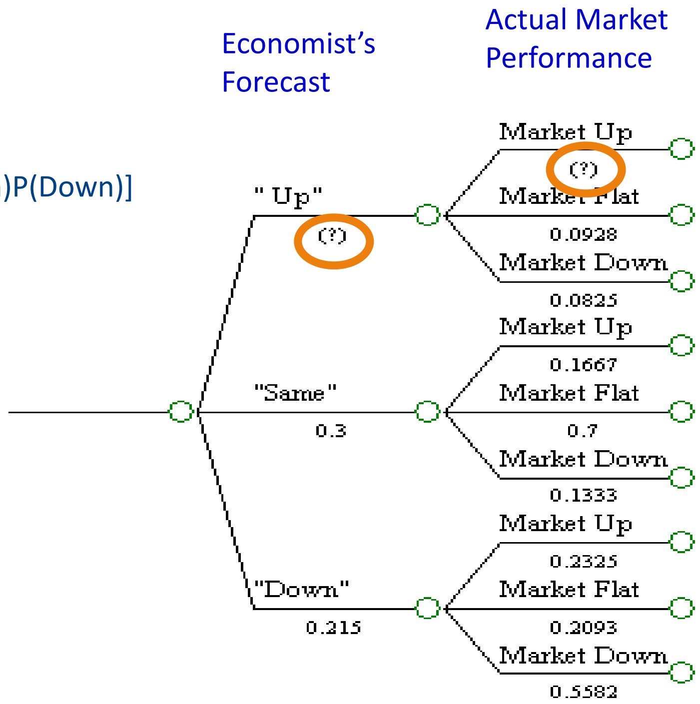
= 0.400/0.485

= *0.8247*

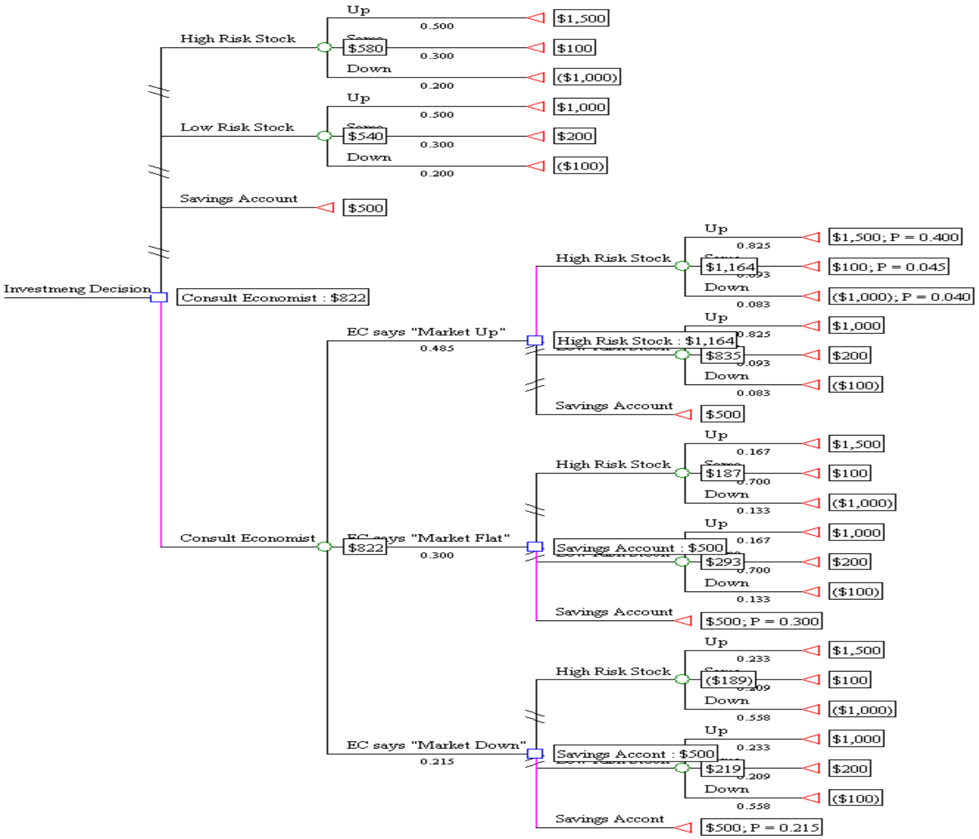
P(Exp says "Up")

= [P("Up"|Up)P(Up)+ P("Up"|Flat)P(Flat) + P("Up"|Down)P(Down)]

= *0.485*



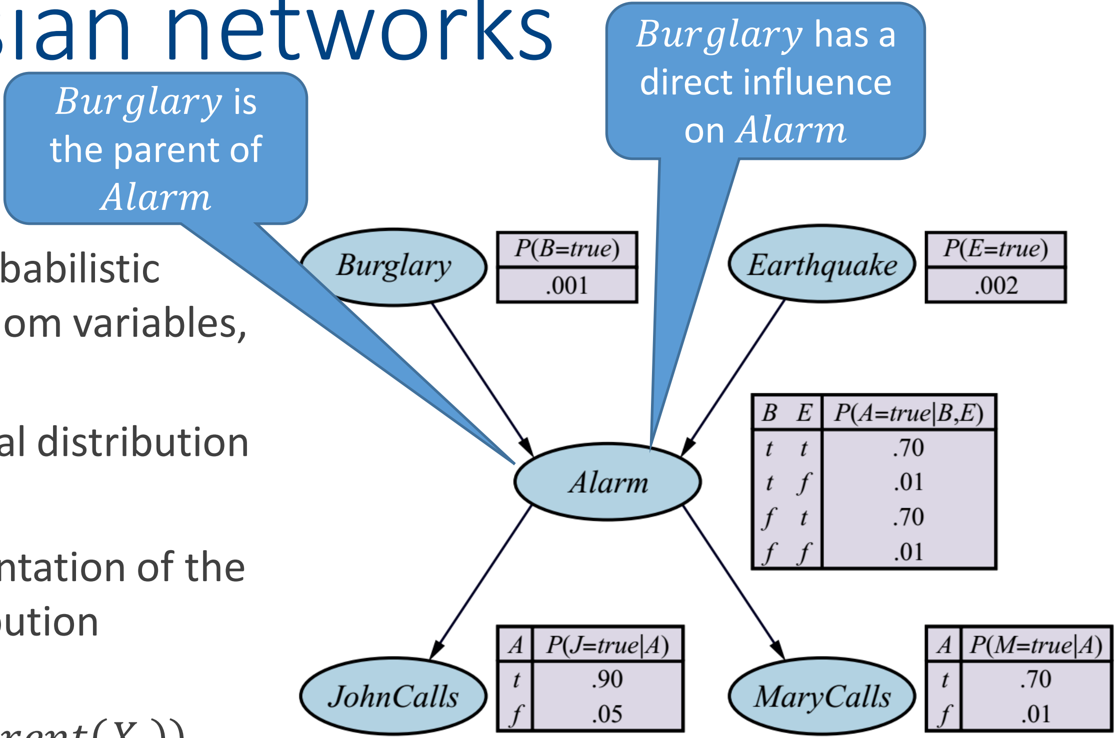
Calculating EVII



Review: Bayesian networks

- A Bayesian network is:
  - A DAG that represents probabilistic dependencies among random variables, aka chance nodes
  - Each node has a conditional distribution
    $$P(X_i|Parent(X_i))$$
  - Compact factored representation of the full joint probability distribution

$$P(x_1, \dots, x_n) = \prod_{i=1}^{n} P(x_i|Parent(X_i))$$

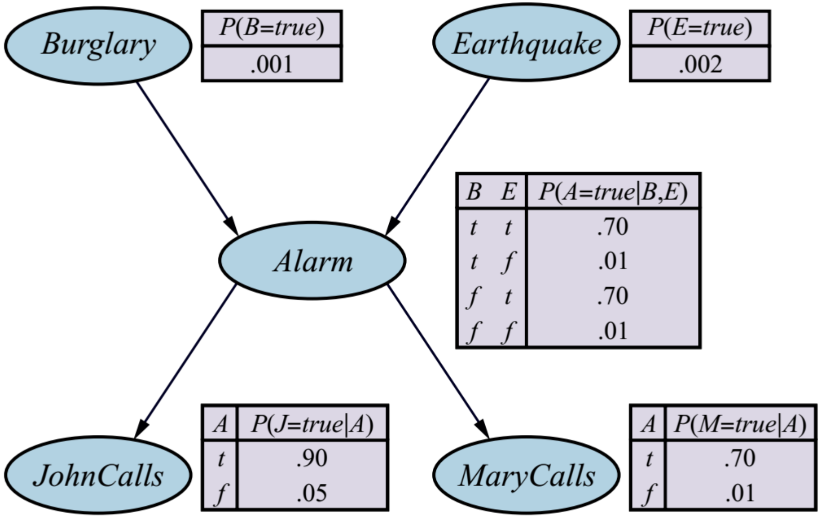


Source: RN: Figure 13.2

Charniak, E., *Bayesian networks without tears: making Bayesian networks more accessible to the probabilistically unsophisticated.* AI Mag., 1991. **12**(4): p. 50–63.

Review: Bayesian networks

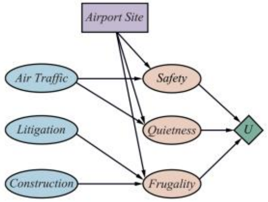
- For Boolean variables:
  - Let each variable be affected by at most k variables,
  - When $k \ll n$, need to specify $< n2^k$ parameters rather than 2^n
- Example:
  - $n = 30$ and $k = 5$; the BN uses 960 numbers;
  - a full joint probability table requires 2^{30} numbers!
- Question:
  - What is the probability to get a call from both John and Mary about alarm sounded when there was no burglary or earthquake?



Source: RN: Figure 13.2

Charniak, E., *Bayesian networks without tears: making Bayesian networks more accessible to the probabilistically unsophisticated.* AI Mag., 1991. **12**(4): p. 50–63.

Decision Tree

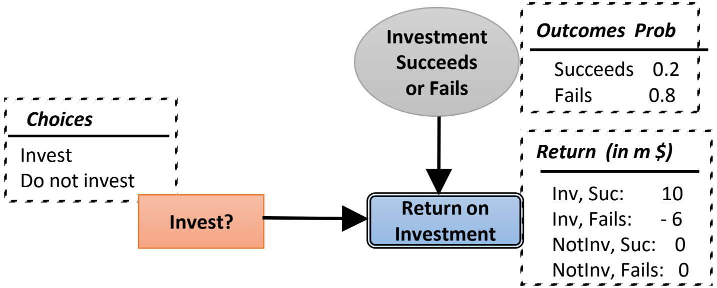


- "Expanded" form of Decision Networks
  - Represents all possible paths through time
  - Decisions and chance events are most naturally placed in a time order from left to right
  - Implicit probabilistic and information dependencies
- Chance node:
  - Arcs denote chance outcomes; each chance node has a set of mutually exclusive and collectively exhaustive outcomes
- Decision node:
  - Arcs denote decision alternatives; only one option can be chosen at each decision node
- Utility node:
  - Utility (terminal) node represents conditional utility associated with path of action-alternative-chance-outcome combinations
  - Collapse multidimensional objective description into a single score for final consequence

$$U = Q(AT, L, C, AS)$$

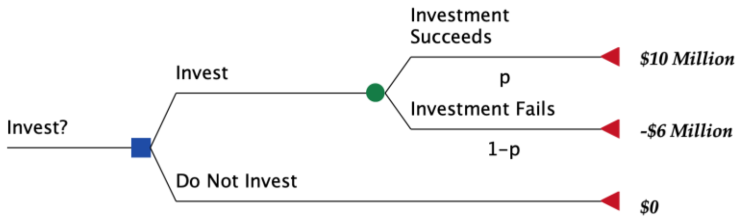
Pros and Cons

- Advantages:
  - Compact representation of decision structure; easy to understand
  - Explicit representation of probabilistic relevance and informational dependence
  - Size grows linearly with number of decision factors; suitable for sequential problems
  - Can capture both discrete and continuous decisions and probabilities
  - Effective as an interactive modeling tool
- Disadvantages:
  - Does not show all possible consequences of decisions and outcomes
  - Can represent only symmetric decisions and event sequences
  - Solution algorithms less straight-forward



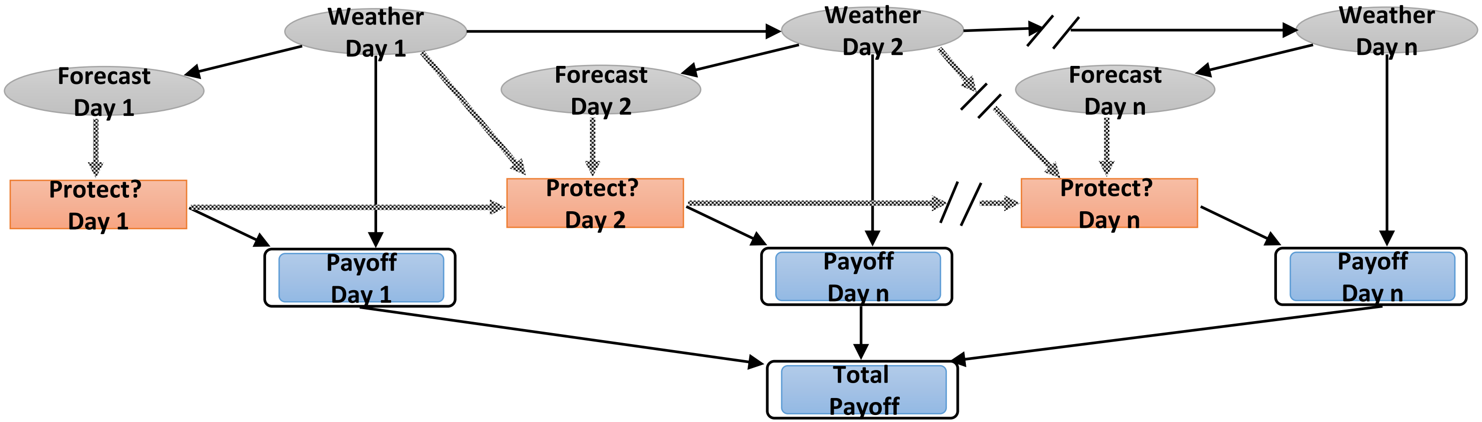
Pros and Cons of Decision Trees

- Advantages:
  - Show details of all possible sequences of decisions and uncertain outcomes
  - Portraits the multiplicity of paths or scenarios that lead to possible outcomes
  - Direct specification of asymmetric outcomes



- Disadvantages:
  - Independence among uncertain events implicit
  - Difficult to display and manage when number of decisions and chance events become large; not suitable for sequential problems
  - Model chance events as discrete random variables
  - Efficiency of rollback solution algorithm constrained by model structure and size

Sequential Decisions



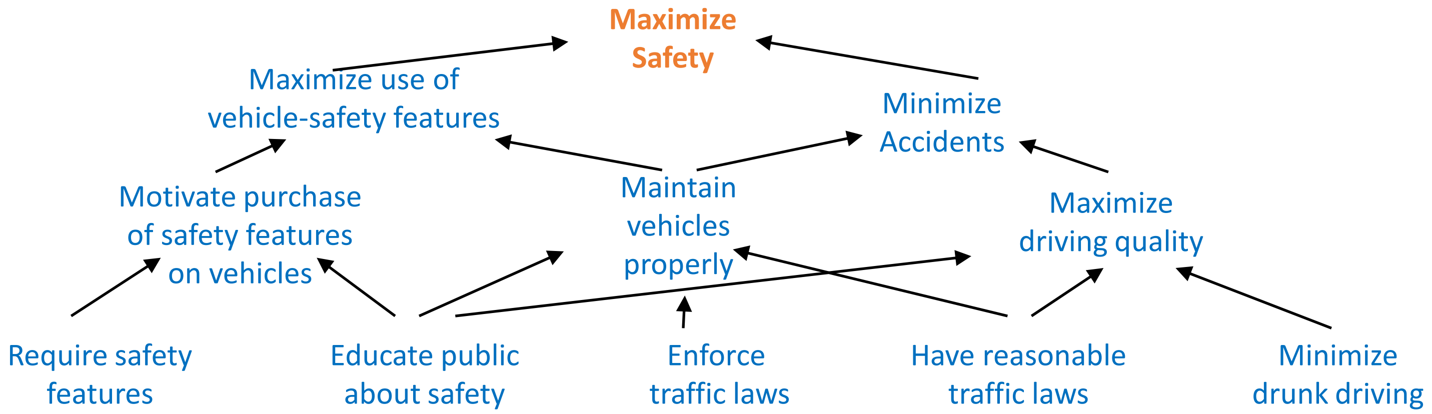
<u>**Sequential Decisions in The Orchard Problem**</u>

- Sequential structure explicitly shown
- No cycles allowed
- "No-forgetting" arcs implied but not shown
- Final value is a function of individual values over all stages

Example: Product Development

- Problem Definition:
  - Decision on whether to develop a new type of food processor
  - Two alternative power sources may be used: electricity and gas
  - The designs have different development costs
  - Either design may succeed or fail with some uncertainty
  - Only one design can be pursued initially due to resource constraints
  - If either design fails, company would still consider modifying design, with potential more investment, and would still not guarantee success
- Two stage decision problem:
  - Decision between designs and not to develop the new processor at all
  - Decision on whether the design should be modified

Example: Introducing New Vehicle Regulation



- How to construct a means objectives network?
  - Ultimately, for any objective: Why is it important?
  - To move away from fundamental objectives: How could you achieve this?
  - To move toward fundamental objectives: Why is that important?
- The means network can suggest creative new alternatives

Example: The Rollback Algorithm