Lecture 21: Uncertainty 6

Victor R. Lesser

CMPSCI 683
Fall 2010

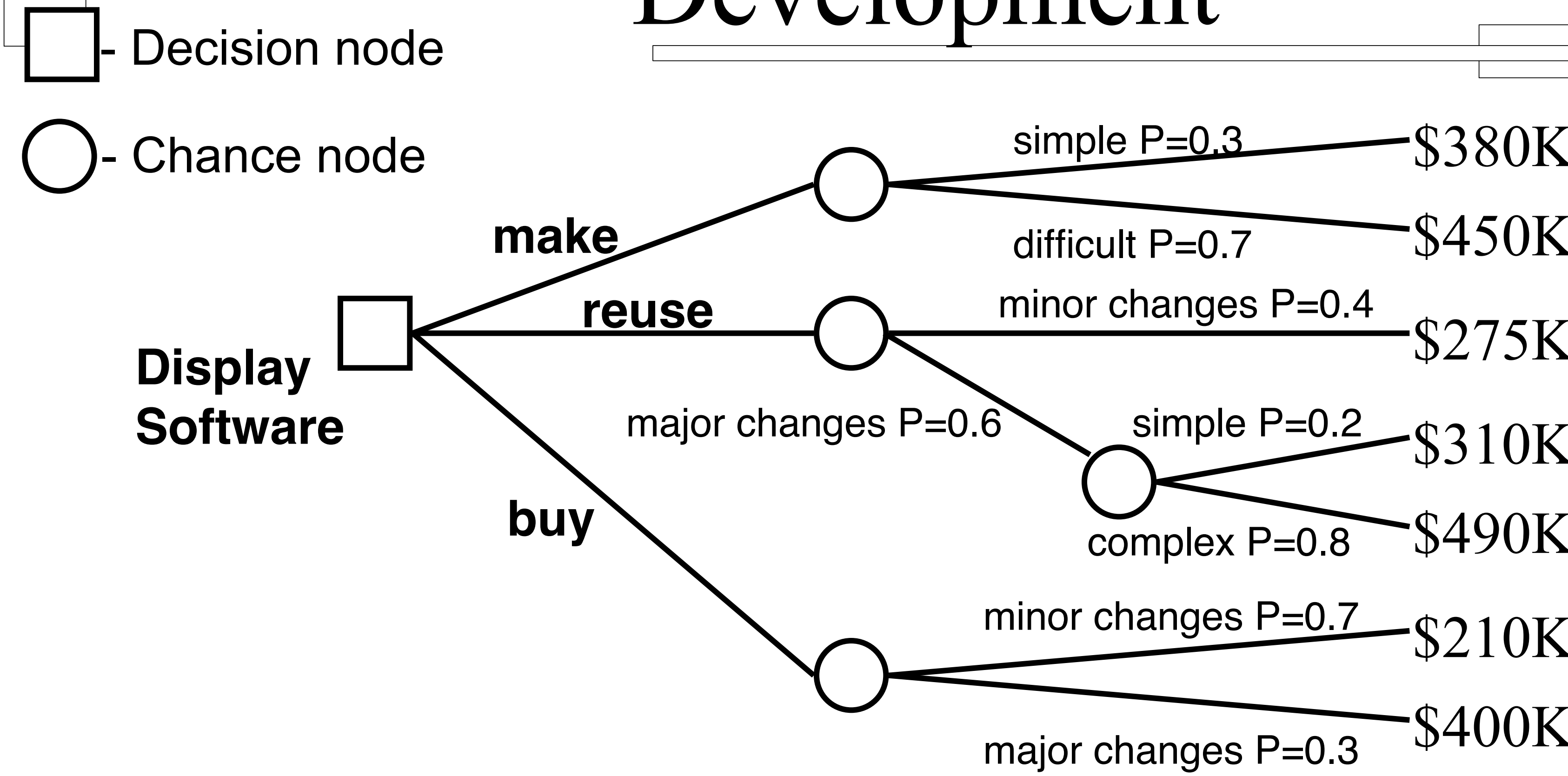
Today's Lecture

- Decision Trees and Networks

Decision Trees

- A decision tree is an explicit representation of all the possible scenarios from a given state.
- Each path corresponds to decisions made by the agent, actions taken, possible observations, state changes, and a final outcome node.
- Similar to a game played against "nature"

Example 1: Software Development



- **EU(**make**) = 0.3 ∗ $380K + 0.7 ∗ $450K** = $429K
- **EU(**reuse**) = 0.4 ∗ $275K + 0.6 ∗ [0.2 ∗ $310K + 0.8 ∗ $490K]** = $382.4K
- **EU(**buy**) = 0.7 ∗ $210K + 0.3 ∗ $400K** = $267K ; best choice

Example 2: Buying a car

- There are two candidate cars C_1 and C_2, each can be of good quality (+) or bad quality (−).
- There are two possible tests, T_1 on C_1 (costs \$50) and T_2 on C_2 (costs \$20).
- C_1 costs \$1500 (\$500 below market value) but if it is of bad quality repair cost is \$700.
  - 500 gain or 200 lost
- C_2 costs \$1150 (\$250 below market value) but if it is of bad quality repair cost is \$150.
  - 250 gain or 100 gain
- Buyer must buy one of the cars and can perform at most one test. -- What other information?

Example 2: Buying a car cont.

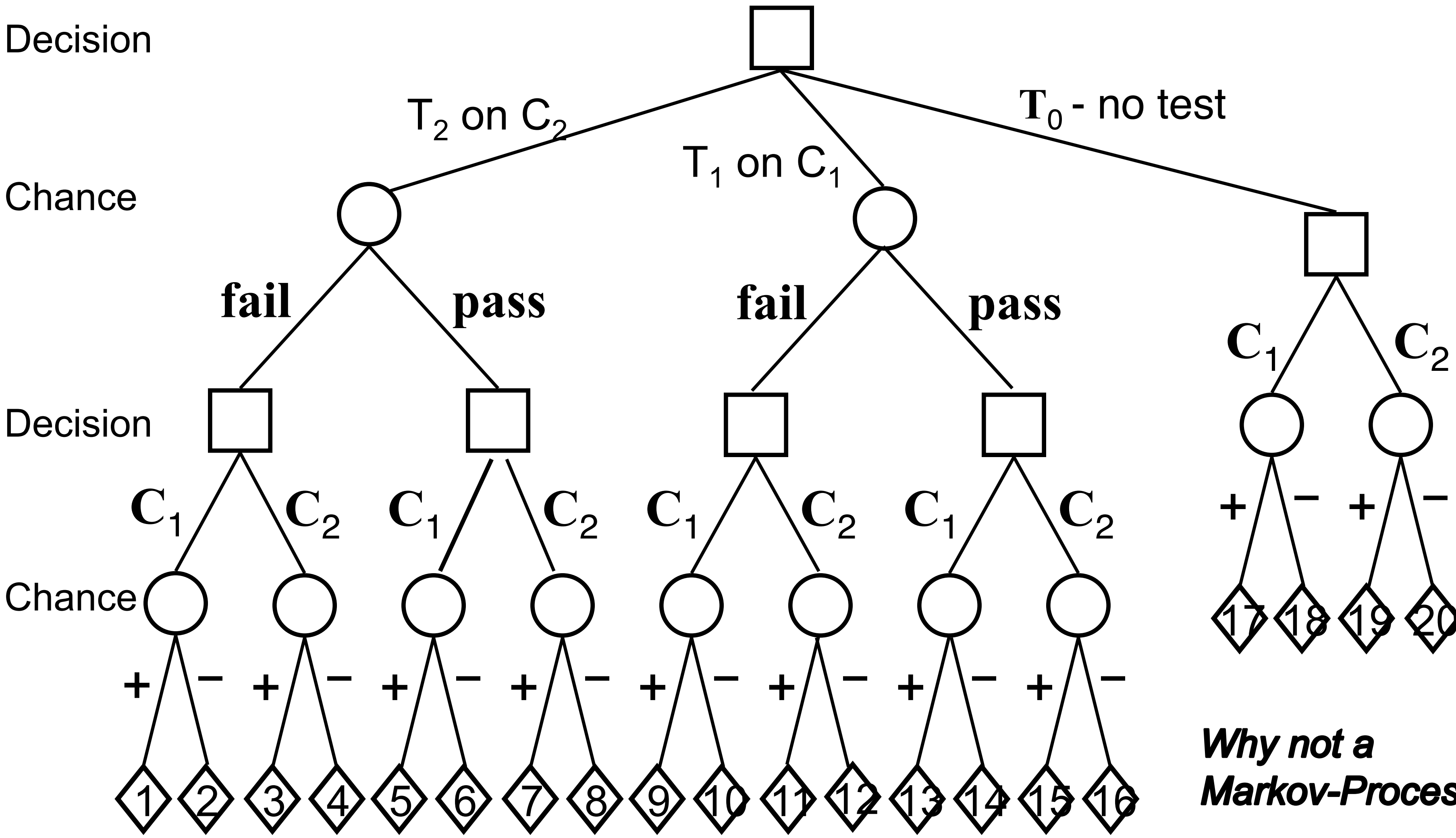
- The chances that the cars are of good quality are 0.70 for C_1 and 0.80 for C_2.

- Test T_1 on C_1 will confirm good quality with probability 0.80 if C_1=good and will confirm bad quality with probability 0.65 if C_1= bad.
  - Imperfect information

- Test T_2 on C_2 will confirm good quality with probability 0.75 and will confirm bad quality with probability 0.70.

Example 2: Buying a car cont.

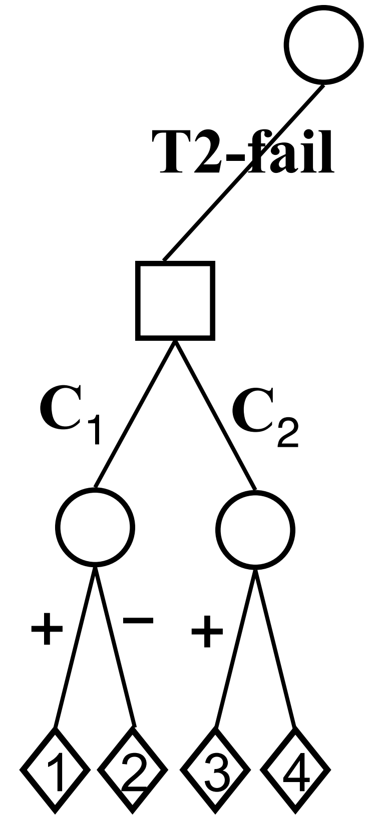
What are the decisions and how can you judge their outcomes?



Why not a Markov-Process?

Evaluating decision trees

1. Traverse the tree in a depth-first manner:
   (a) Assign a value to each *leaf node* based on the outcome, then back-up outcome values
   (b) Calculate the average utility at each *chance node* based on the likelihood of each outcome
   (c) Calculate the maximum utility at each *decision node*, while marking the maximum branch
2. Trace back the marked branches, from the root node down to find the desired optimal (conditional) plan.



Finding the value of (perfect or imperfect) information in a decision tree.

Additional Information

Buyer knows car c_1 is good quality

70% $P(c_1=\text{good}) = .7$

Buyer knows car c_2 is good quality

80% $P(c_2=\text{good}) = .8$

Test t_1 check quality of car c_1

$P(t_1=\text{pass}|c_1=\text{good}) = .8$

$P(t_1=\text{pass}|c_1=\text{bad}) = .35$

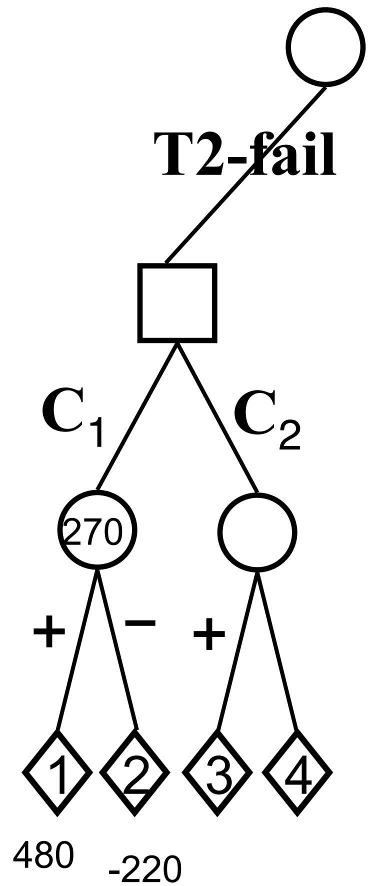
Test of t_2 check quality of car c_2

$P(t_2=\text{pass}|c_2=\text{good}) = .75$

$P(t_2=\text{pass}|c_2=\text{bad}) = .3$

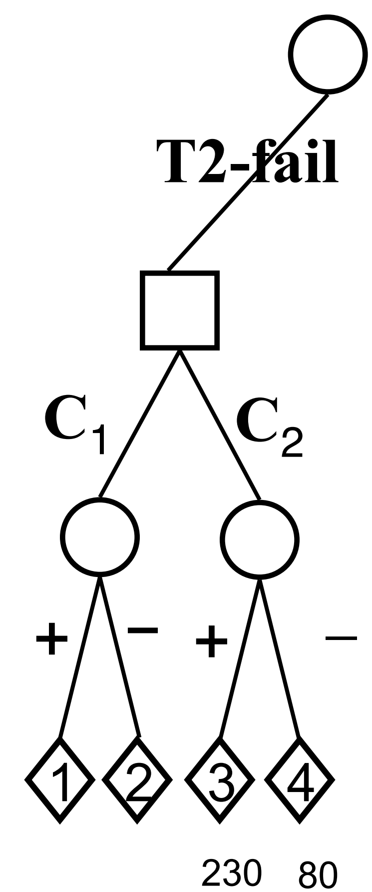
Details of Example

- Case 1
  - P(c1=good|t2=fail)=p(c1=good)=.7;
    test t2 does not say anything about c1
  - Utility = 2000(*value of car)*-1500(*cost of car*)-20(*cost of test*) =480
- Case 2
  - P(c1=bad|t2=fail) = p(c1=bad) =
    1- p(c1=good) = .3
  - Utility = 2000-1500-700(*cost of repair*)-20 = -220
- Expected Utility of Chance Node of 1&2
  - .7 x480 +.3x-220 = 270



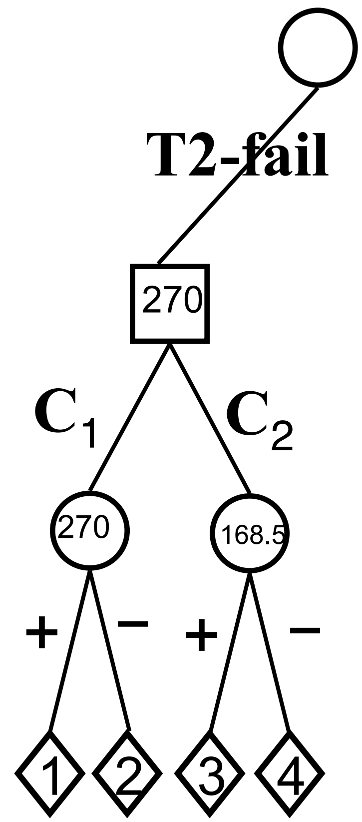
Details of Example cont

- Case 3
  - P(c2=good|t2=fail) =
  - P(t2=fail|c2=good) P(c2=good)/P(t2=fail) =
  - (.25x.8=.2)/ P(t2=fail) =
  - Normalize .2/.34 (=*.2+.14*), .14/.34 (over c2=bad case 4)
  - .59
  - Utility = 1400-1150-20= 230
- Case 4
  - P(c2=bad/t2=fail) =
  - P(t2=fail/c2=bad) P(c2=bad)/P(t2=fail) =
  - (.7x.2=.14) / P(t2=fail) =
  - .41
  - Utility = 1400-1150-20-150= 80
- Expected Utility of Chance Node of 3&4
  - .59 x230 +.41x80 =168.5



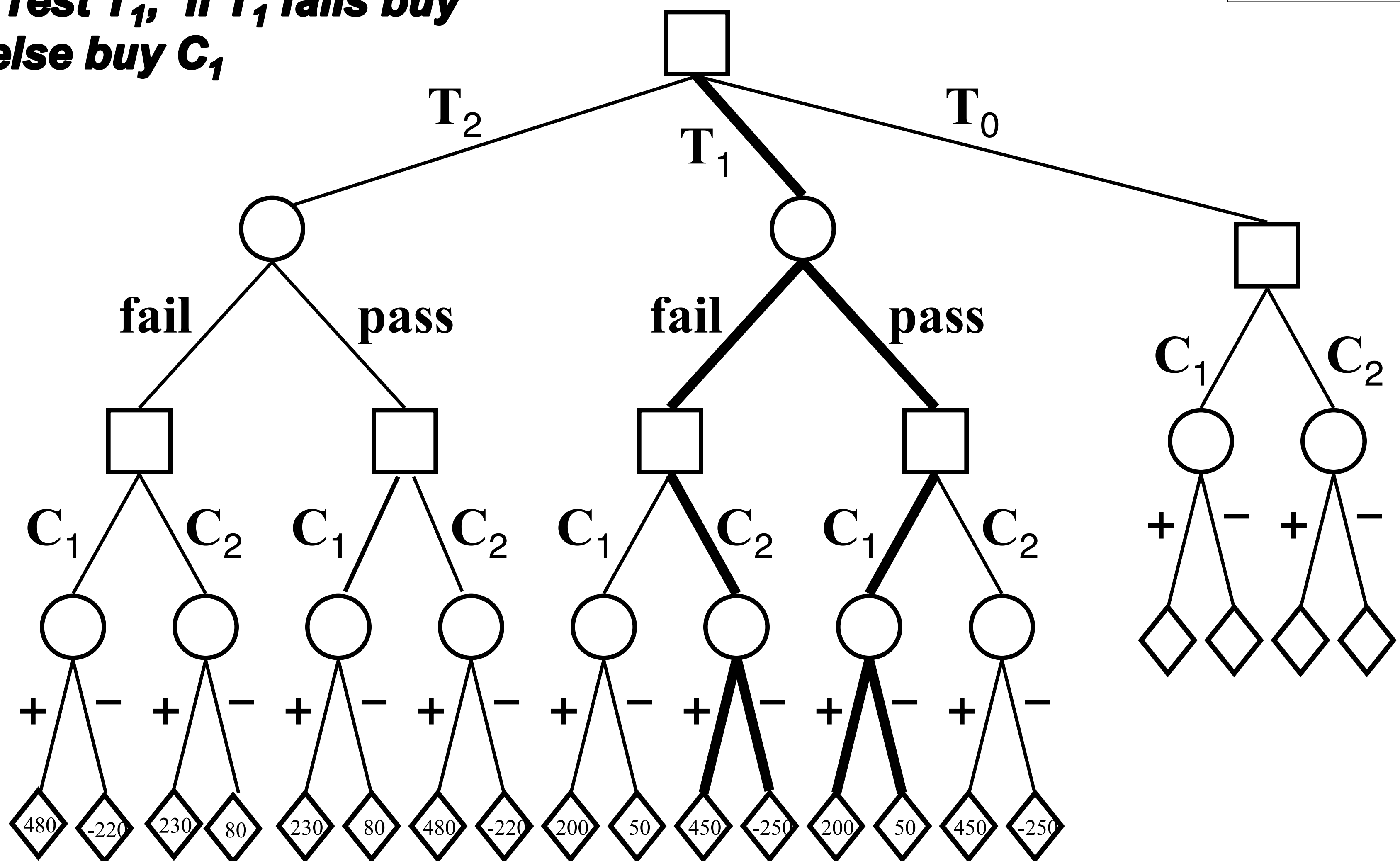
Details of Example cont

- What is the decision if
  - Decide to do test t2
  - It comes out false
  - Do you buy c1 or c2?
    - E(c1|test t2=fail) = Expected Utility of Chance Node of 1&2 = 270
    - E(c2|test t2=fail) = Expected Utility of Chance Node of 3&4 = 168.5



Example 2: Buying a car cont.

Do Test T_1; If T_1 fails buy C_2 else buy C_1



Decision Networks/Influence Diagrams

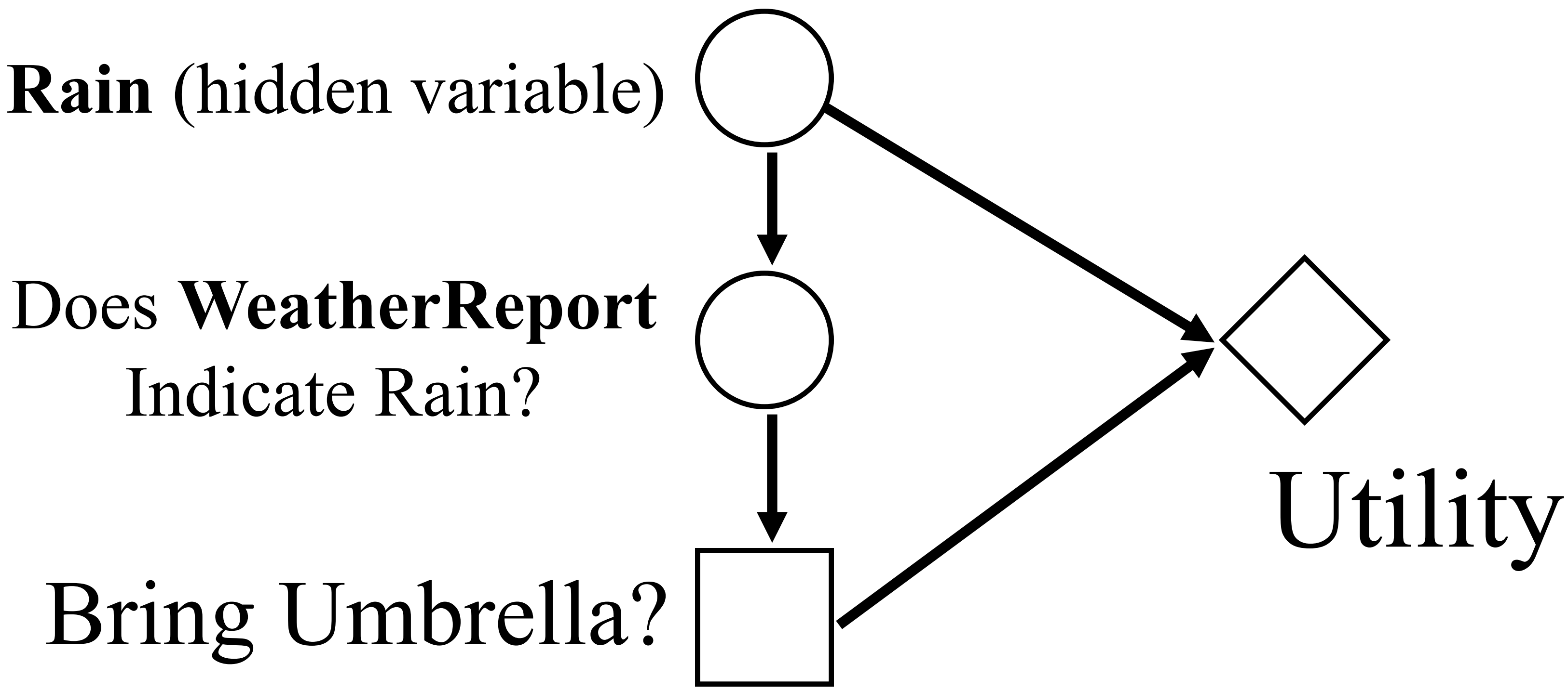
- Decision networks or influence diagrams are an extension of belief networks that allow for reasoning about actions and utility.

- The network represents information about the agent's current state, its possible actions, the possible outcome of those actions, and their utility.

Nodes in a Decision Network

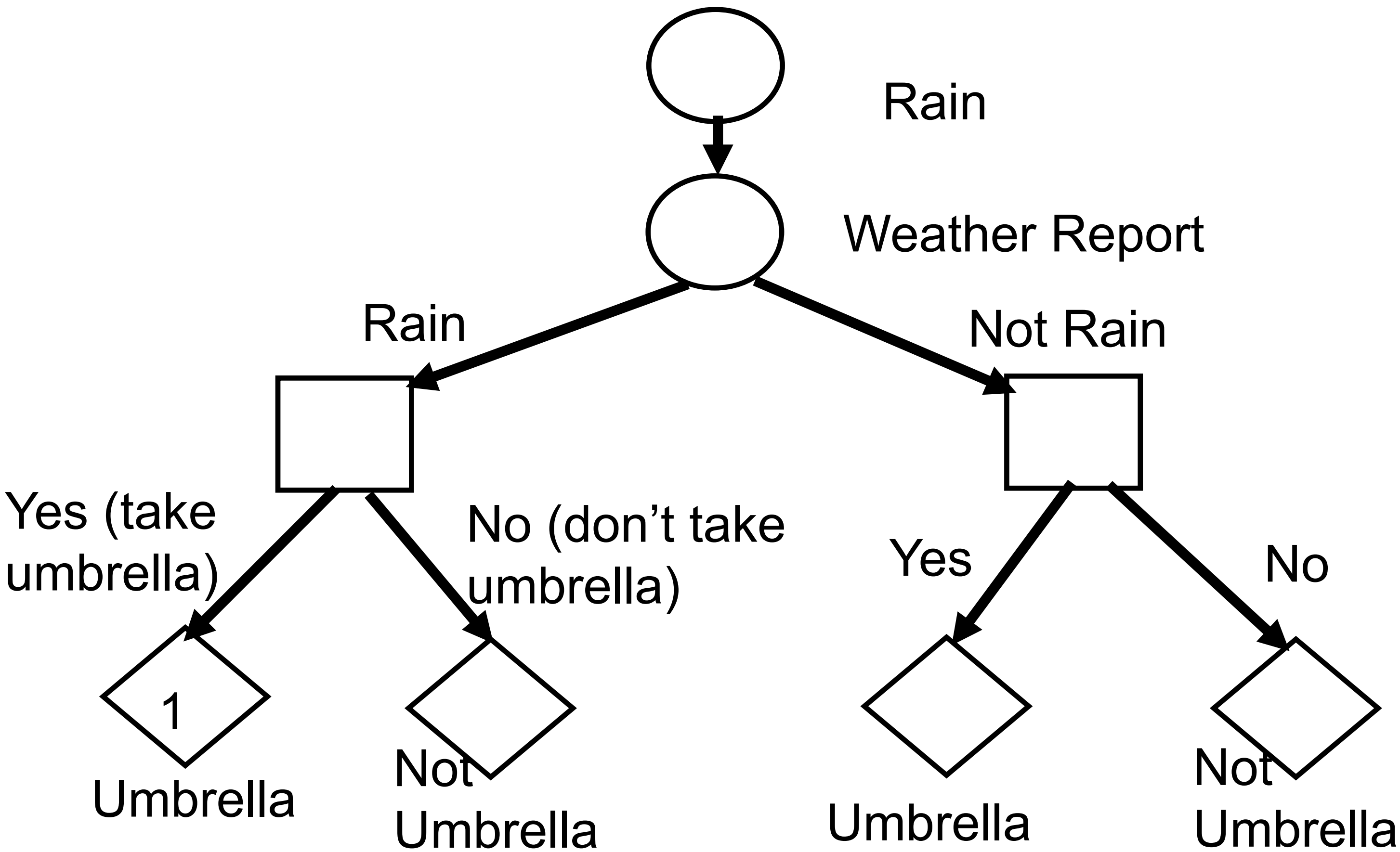
- Chance nodes **(ovals)** have CPTs (conditional probability tables) that depend on the states of the parent nodes (chance or decision).
- Decision nodes **(squares)** represent options available to the decision maker.
- Utility nodes **(Diamonds)** or value nodes represent the overall utility *based on the states of the parent nodes*.

Example 3: Taking an Umbrella



Parameters: P(Rain), P(WeatherReport|Rain), P(WeatherReport|¬Rain), Utility(Rain,Umbrella)

"Taking an Umbrella" as Decision Tree

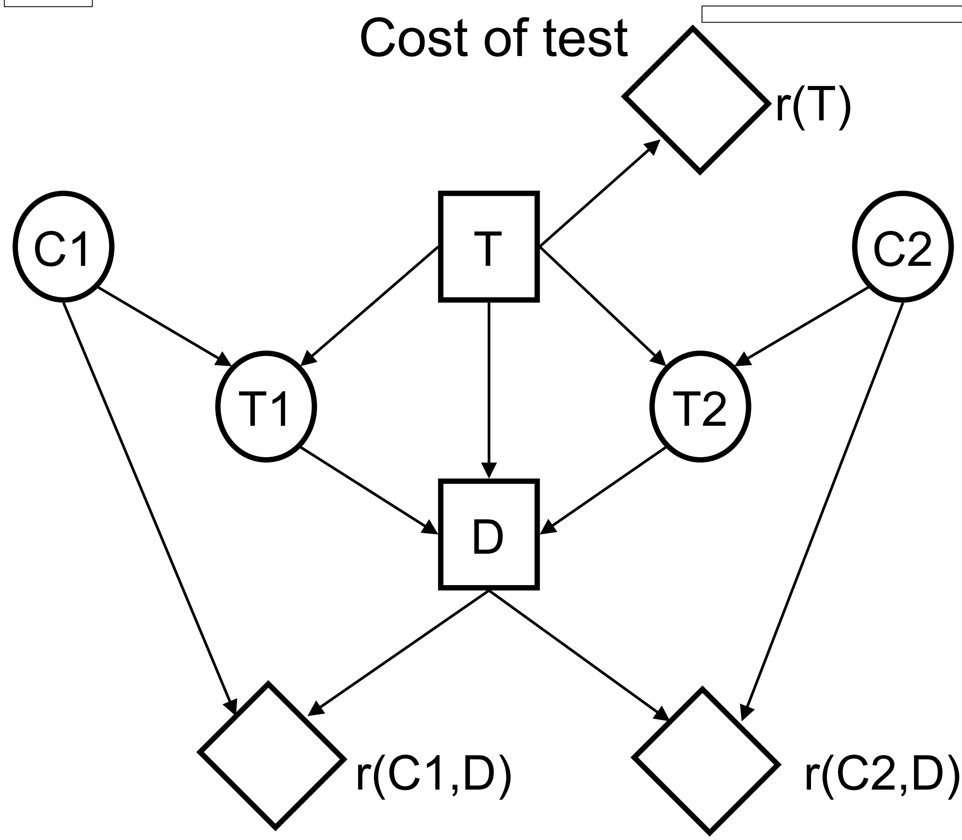


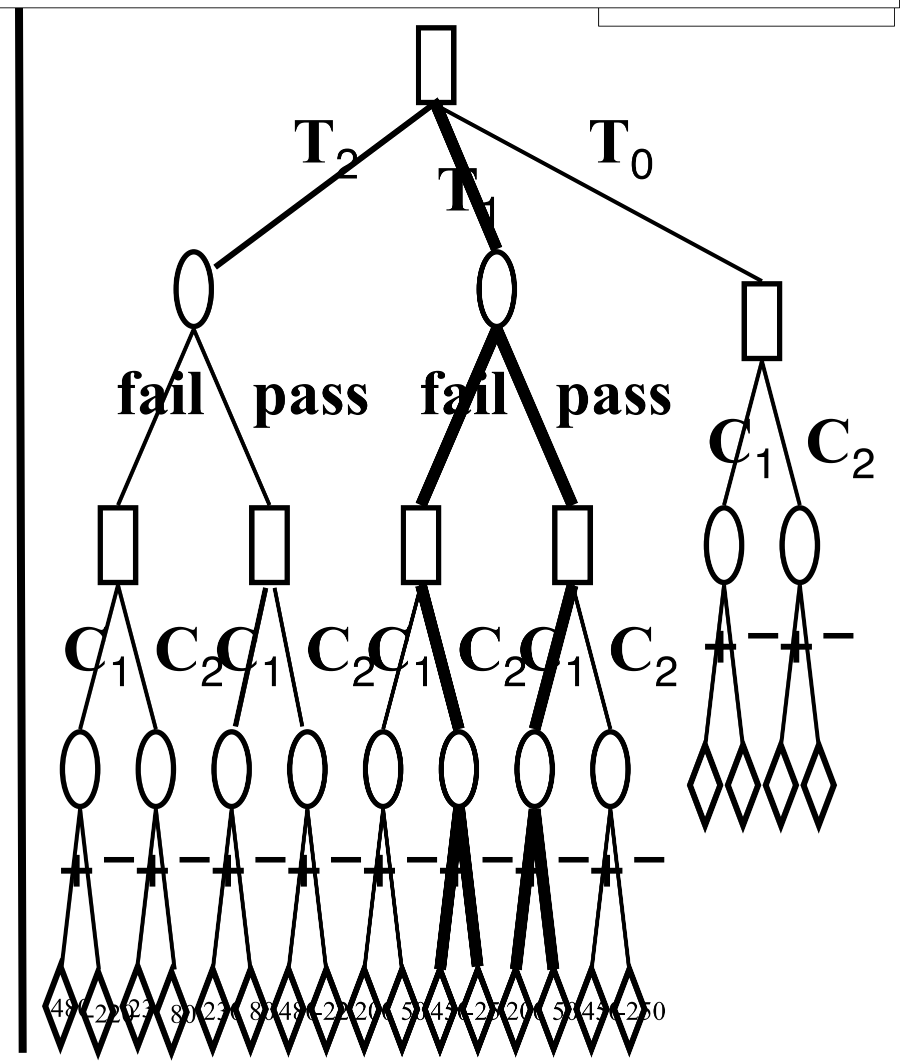
Case 1:U(Umbrella|W=Rain)*P(W=Rain|WR=Rain) +U(Umbrella|W=not Rain)*P(W= not Rain|WR=Rain)

Knowledge in an Influence Diagram

- Causal knowledge about how events influence each other in the domain
- Knowledge about what action sequences are feasible in any given set of circumstances
  - Lays out possible temporal ordering of decisions
- Normative (Utility) knowledge about how desirable the consequences are

Example 2 as an Influence Diagram





T is decision whether to do a Test or not and which one

D is the decision of which car to buy

Decision Trees vs Influence Diagrams

- Decision trees are not convenient for representing domain knowledge
  - Requires tremendous amount of storage
    - Multiple decisions nodes -- expands tree
    - Duplication of knowledge along different paths
      - *Joint Probability Distribution vs Bayes Net*
- Generate decision tree on the fly from more economical forms of knowledge
  - Depth-first expansion of tree for computing optimal decision

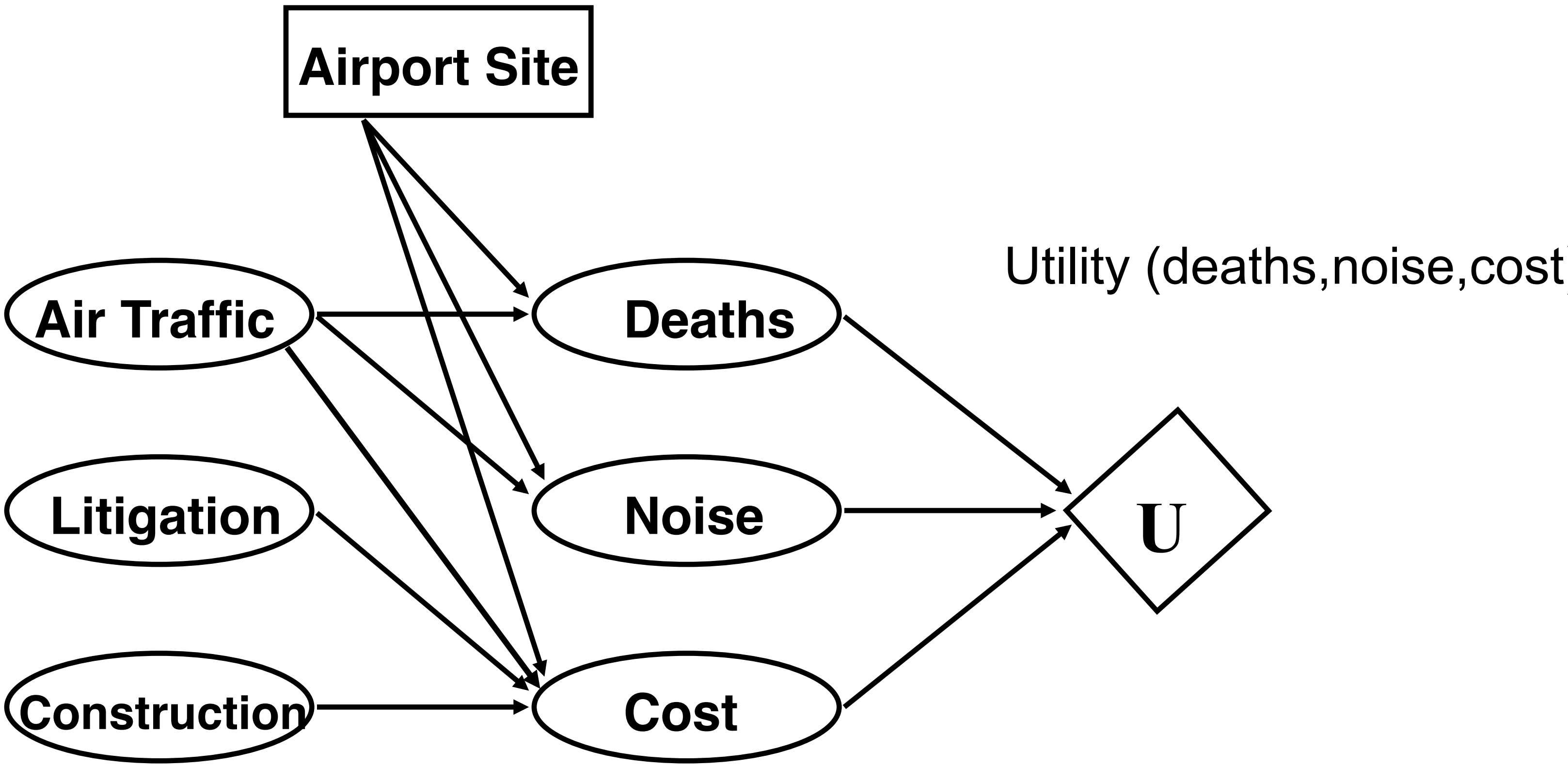
Topology of decision networks

1. The directed graph has no cycles.
2. The utility nodes have no children.
3. There is a directed path that contains all of the decision nodes.
4. A CPT is attached to each chance node specifying P(A|parents(A)).
5. A real valued function over parents(U) is attached to each utility node.

Semantics

- Links into decision nodes are called "information links," and they indicate that the state of the parent is known prior to the decision.
- The directed path that goes through all the decision nodes defines a temporal sequence of decisions.
- It also partitions the chance variables into sets: I_0 is the vars observed before any decision is made, I_1 is the vars observed after the first and before the second decision, etc. I_n is the set of unobserved vars.
- The "no-forgetting" assumption is that the decision maker remembers all past observations and decisions. -- Non Markov Assumption

Example 4: Airport Siting Problem



- P(cost=high|airportsite=Darien, airtraffic=low, litigation=high, construction=high)

Evaluating Decision Networks

1. Set the evidence variables for the current state.
2. For each possible value of the decision node(s):
   (a) Set the decision node to that value.
   (b) Calculate the posterior probabilities for the parent nodes of the utility node.
   (c) Calculate the expected utility for the action.
3. Return the action/decision with the highest utility.

Similar to Cutset Conditioning of a Multiply Connected Belief Network

Imperfect Information Example 5: Mildew

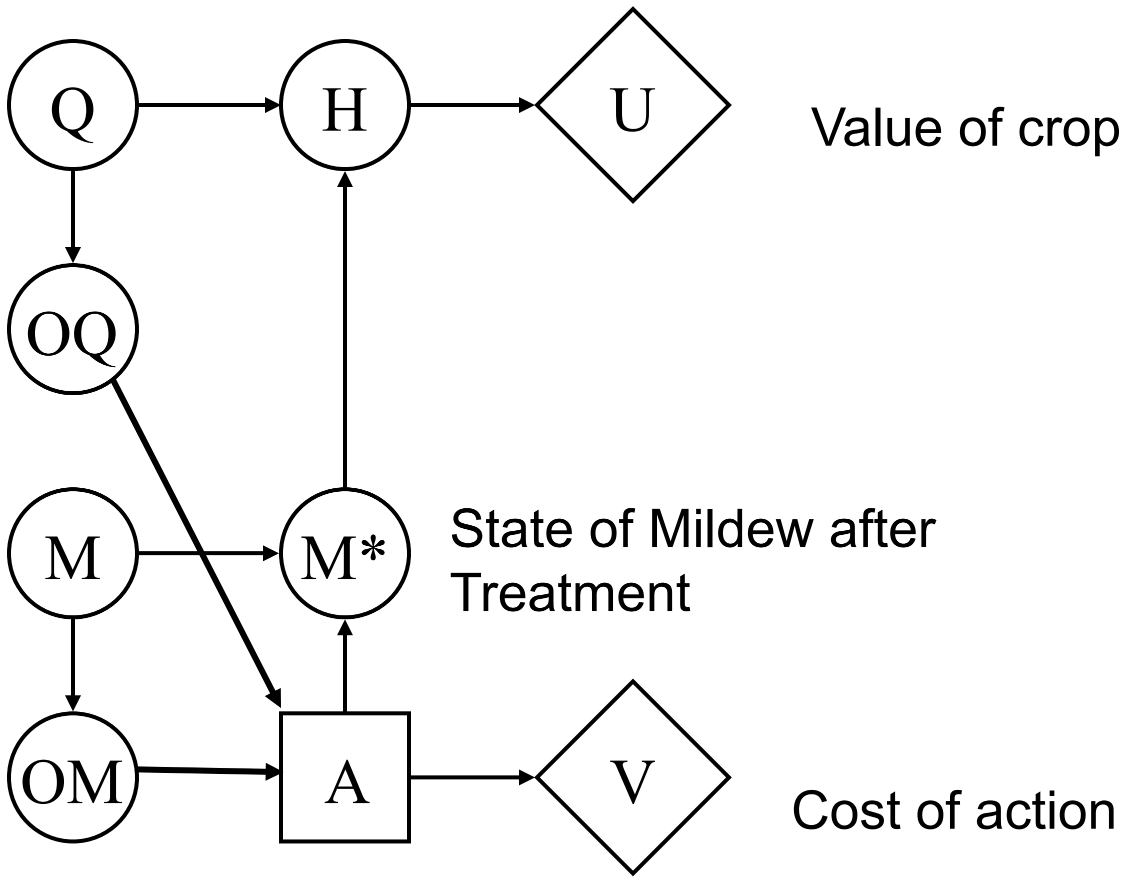
Two months before the harvest of a wheat field, the farmer observes the state Q of the crop, and he observes whether it has been attacked by mildew, M. If there is an attack, he will decide on a treatment with fungicides.

There are five variables:

- Q: fair (f), not too bad (n), average (a), good (g)
- M: no (no), little (l), moderate (m), severe (s)
- H: state of Q plus M: rotten (r),bad (b), poor (p)
    - **State after action taken whether to treatment or not**
- OQ: observation of Q; imperfect information on Q
- OM: observation of M; imperfect information on M

Mildew decision model

Maximize (‘value of crop” - “cost of action”)



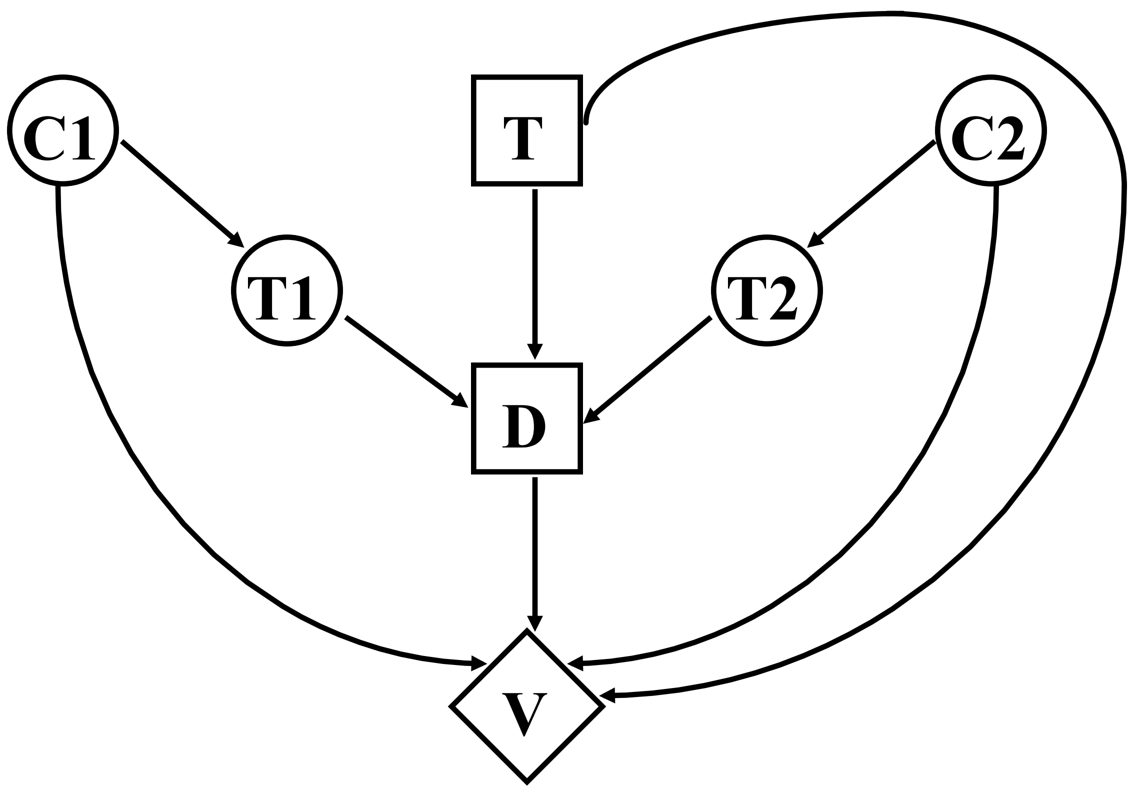
One action in general

- A single decision node D may have links to some chance nodes.
- A set of utility functions $U_1,\ldots,U_n$ over domains $X_1, \ldots, X_n$.
- Goal: find the decision d that maximizes $EU(D=d \mid e)$:

$$EU(D \mid e) = \sum_{X_1} U_1(X_1) P(X_1 \mid D, e) + \ldots + \sum_{X_n} U_n(X_n) P(X_n \mid D, e)$$

- How to solve such problems using a standard Bayesian network package?

Multiple decisions -- Policy Generation



Need a more complex evaluation technique since generating a policy (sequence of decisions)

The Domain of Decision Nodes:

Options At Decision Node D

- T: t_0, t_1, t_2
- D:

If T = t_0 then { Buy 1, Buy 2 }

If T = t_1 then {

Buy 1 if t_1=pass else Buy 2,

Buy 2 if t_1=pass else Buy 1,

always Buy 1,

always Buy 2 }

If T = t_2 then {

Buy 1 if t_2=pass else Buy 2,

Buy 2 if t_2=pass else Buy 1,

always Buy 1, always Buy 2 }

The Next Set of Slides were not covered in detail in class and thus will not be tested on the final exam

Evaluation by Graph Reduction

Basic idea: (Ross Shachter) Perform a sequence of transformations to the diagram that preserve the optimal policy and its value, until only the UTILITY node remains.

- **Similar to ideas of transformation into polytree**

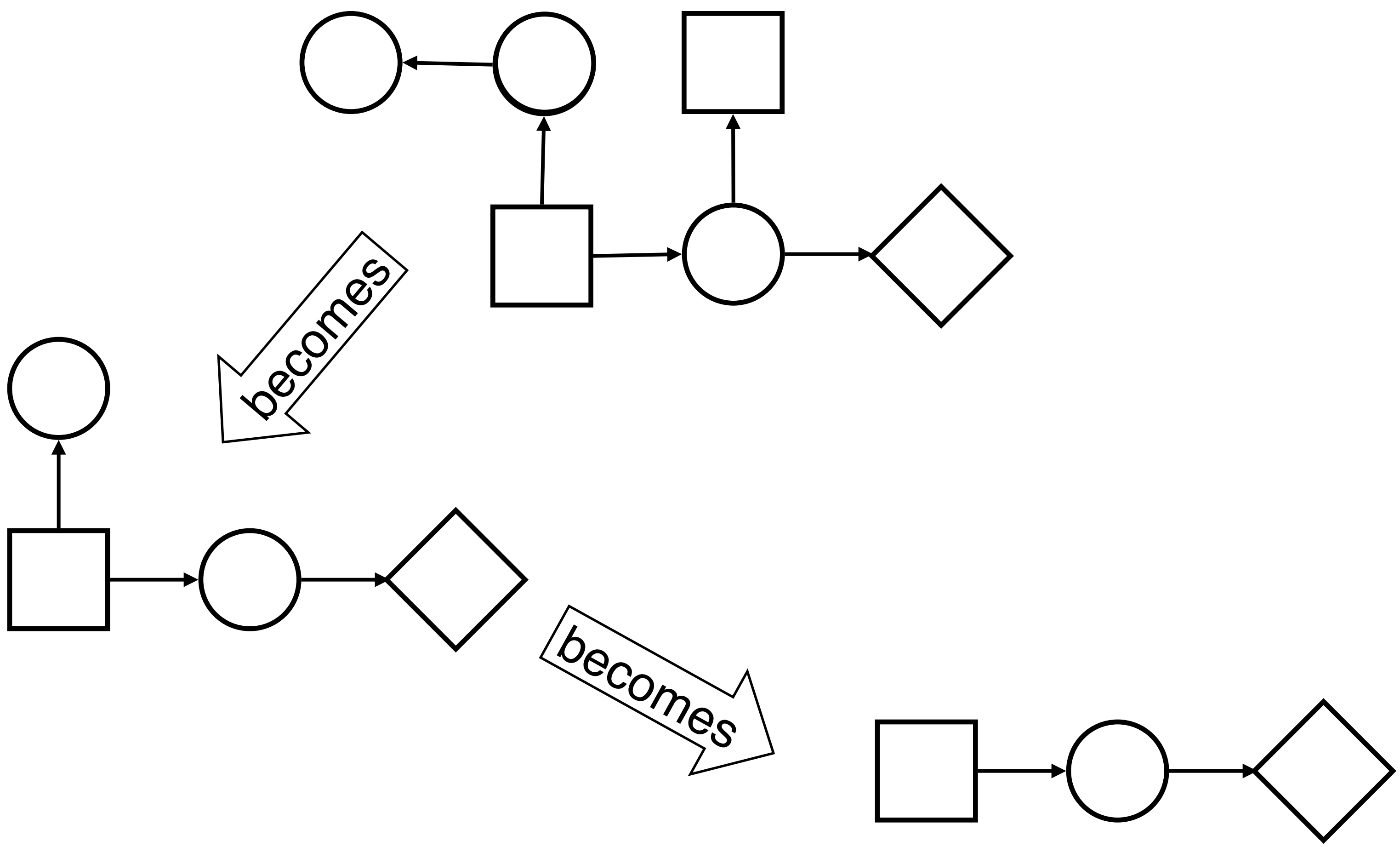
Four basic value/utility-preserving reductions:

- Barren node removal
- Chance node removal (marginalization)
- Decision node removal (maximization)
- Arc reversal (Bayes' rule)

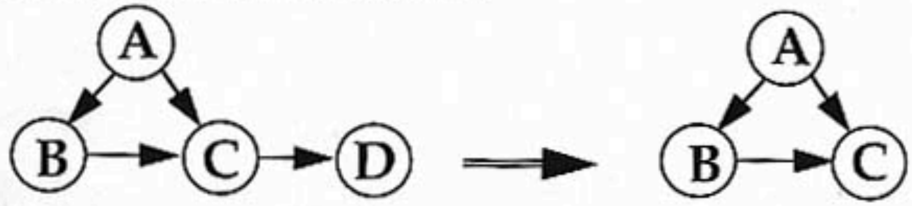
Barren node reduction

- Let X_j represent a subset of nodes of interest in an influence diagram.
- Let X_k represent a subset of evidence nodes.
- We are interested in $P(f(X_j) \mid X_k)$
- A node is "barren" if it has no successors and it is not a member of X_j or X_k.
- The elimination of barren nodes does not affect the value of $P(f(X_j) \mid X_k)$

Barren Node Removal

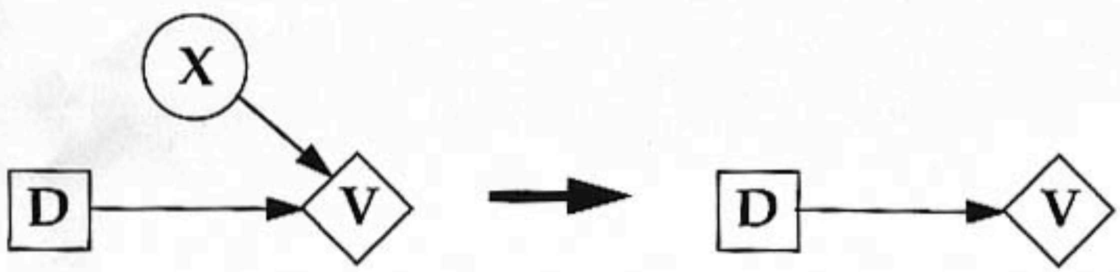


1 **Barren Node Removal**



$$\sum_D P(A,B,C,D) = P(A)P(B \mid A)P(C \mid B,A)\underbrace{\sum_D P(D \mid C)}_{=1}$$

1 **Removal into Value Node (by Expectation)**



$$V(D) = \sum_X V(X,D) * P(X)$$

Notation for Shachter’s algorithm

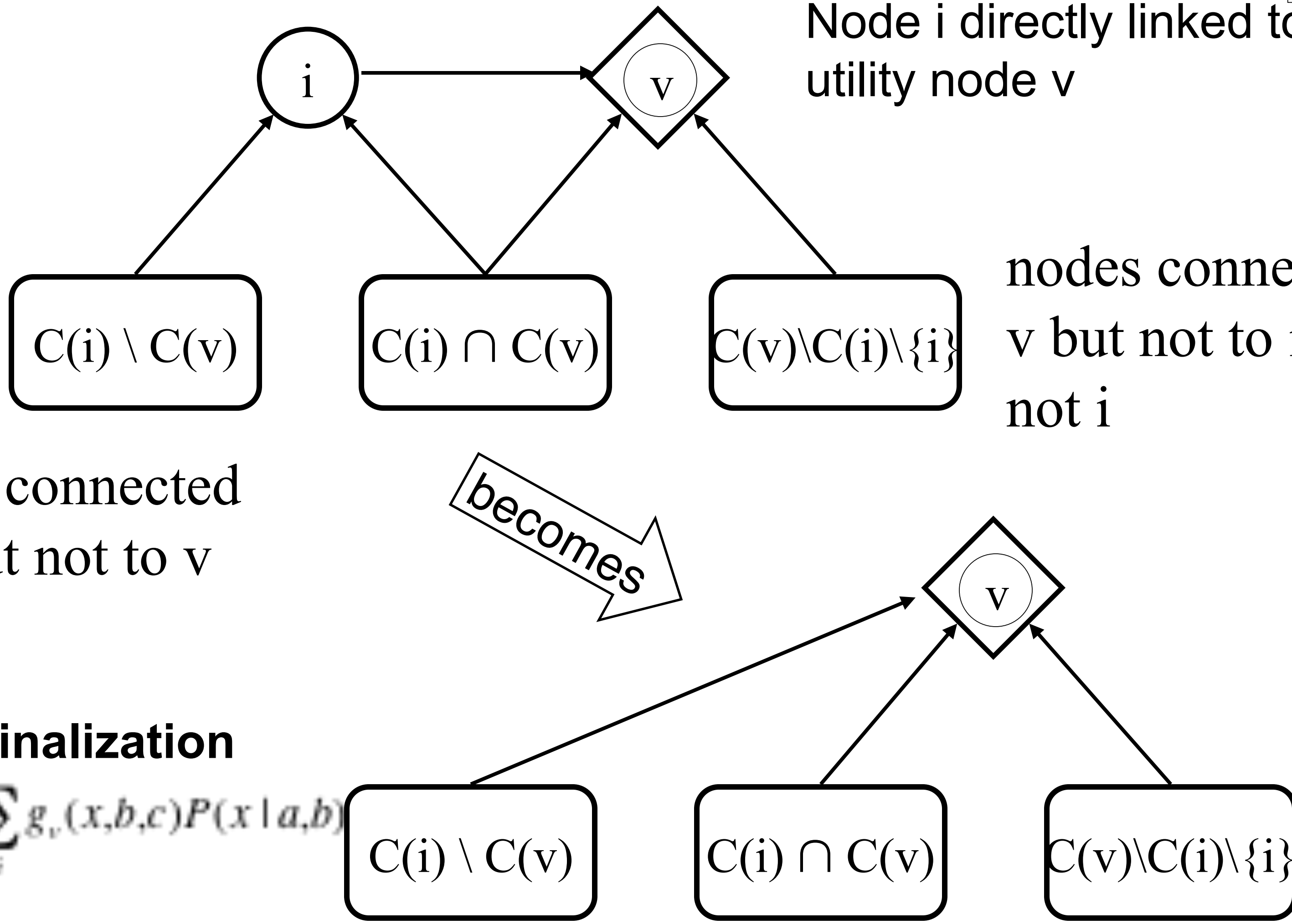
For chance nodes:

- S(i) = direct successors = children
- C(i) = conditional predecessors = parents

For decision nodes

- I(i) = information predecessors = parents

Chance Node Removal



Node i directly linked to utility node v

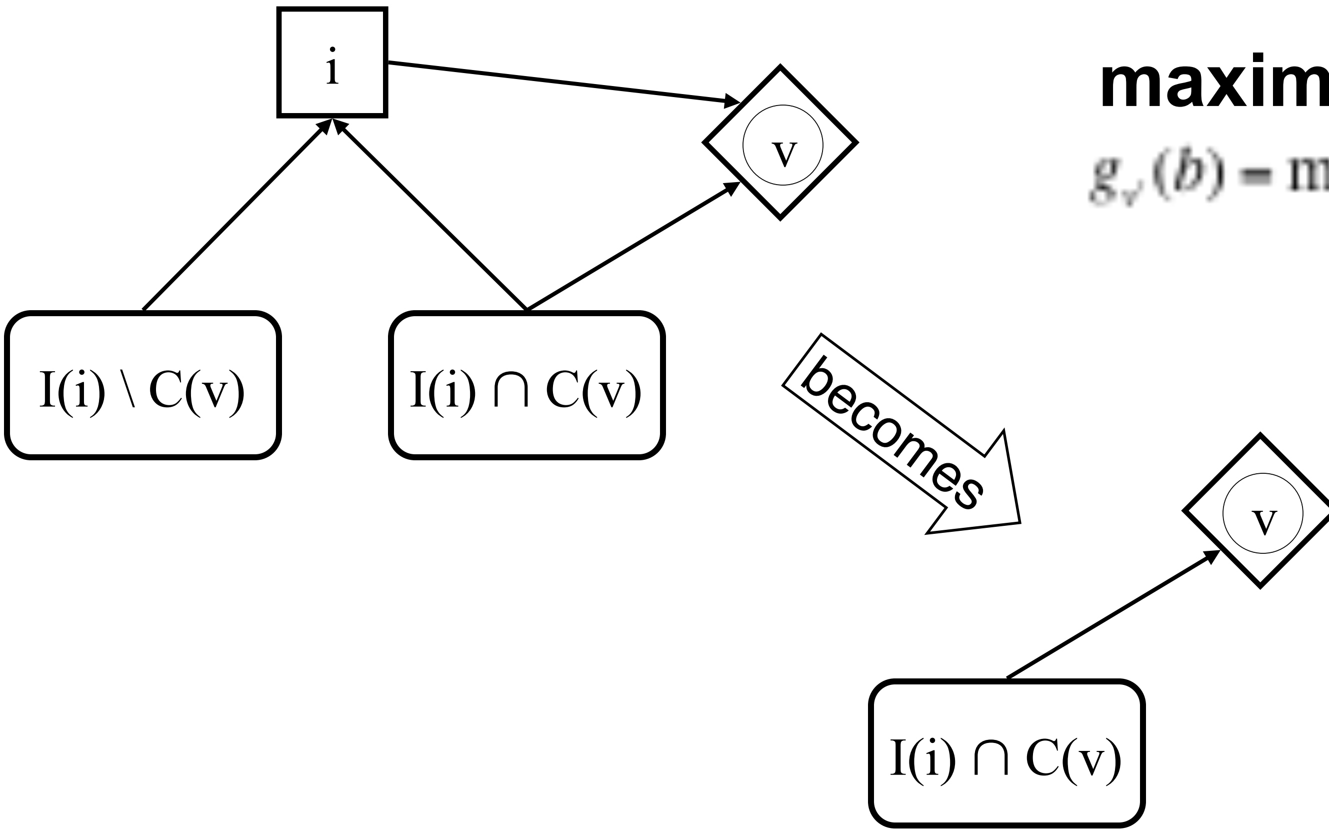
nodes connected to v but not to i and not i

nodes connected to i but not to v

marginalization

$$g_v(a,b,c) = \sum_i g_v(x,b,c)P(x \mid a,b)$$

Decision node removal



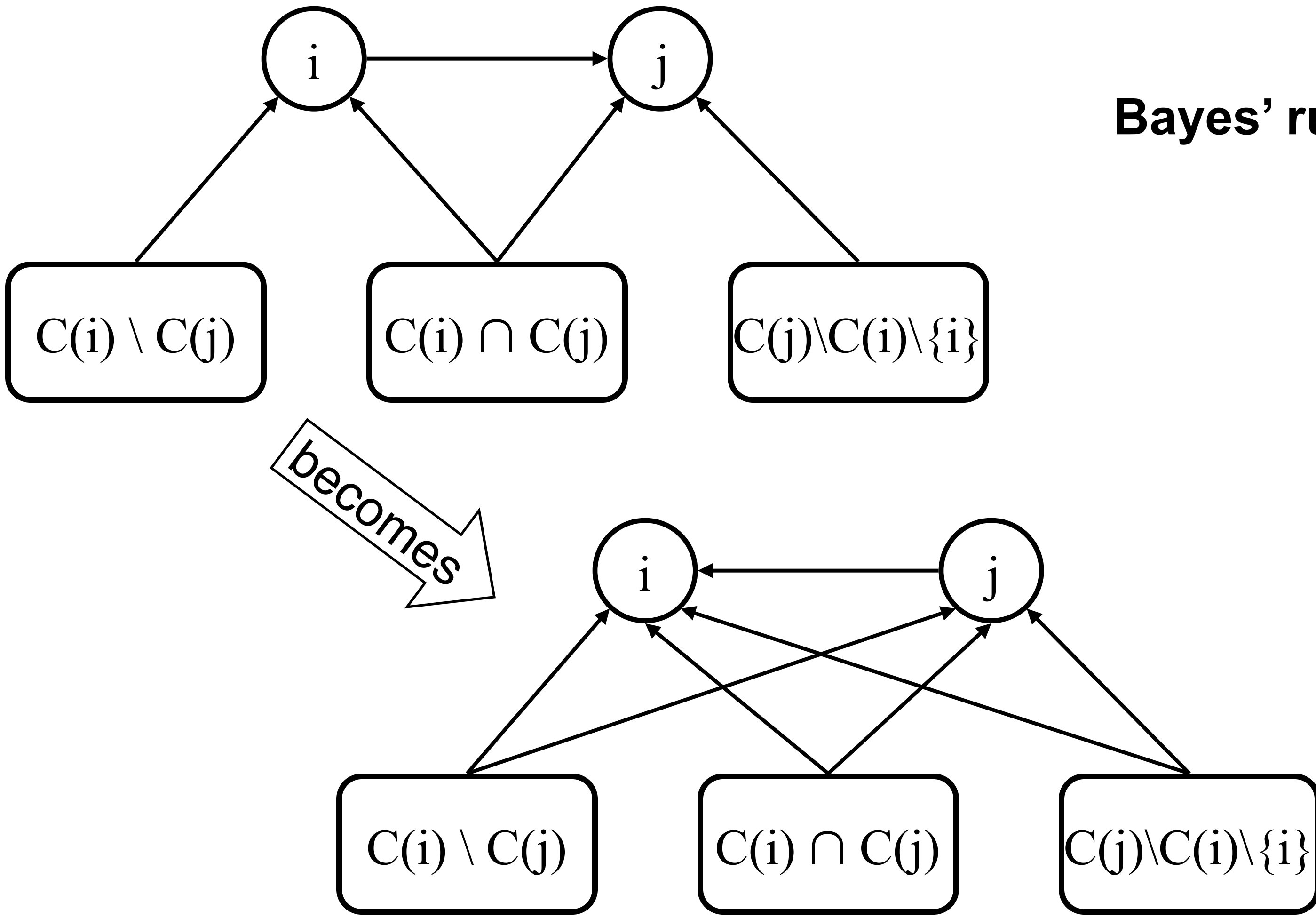
maximization

$$g_v(b) = \max_i \; g_v(i,b)$$

Arc reversal

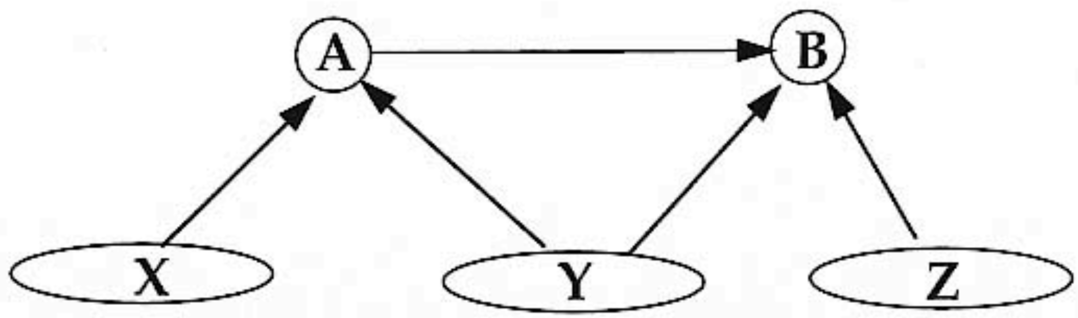
- Given an influence diagram containing an arc from i to j, but no other directed path from i to j, it is possible to to transform the diagram to one with an arc from j to i. (If j is deterministic, then it becomes probabilistic.)

Arc Reversal

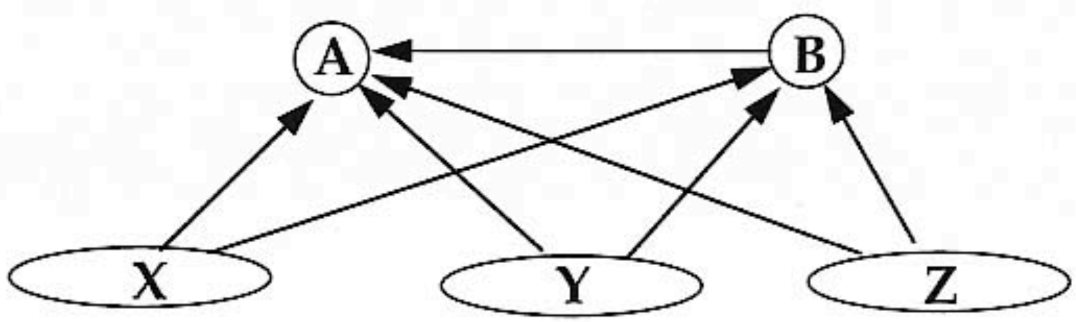


Bayes' rule

Arc Reversal



X = Pa(A)\Pa(b)   Y = Pa(A)vPa(b)   X = Pa(B)\Pa(A)

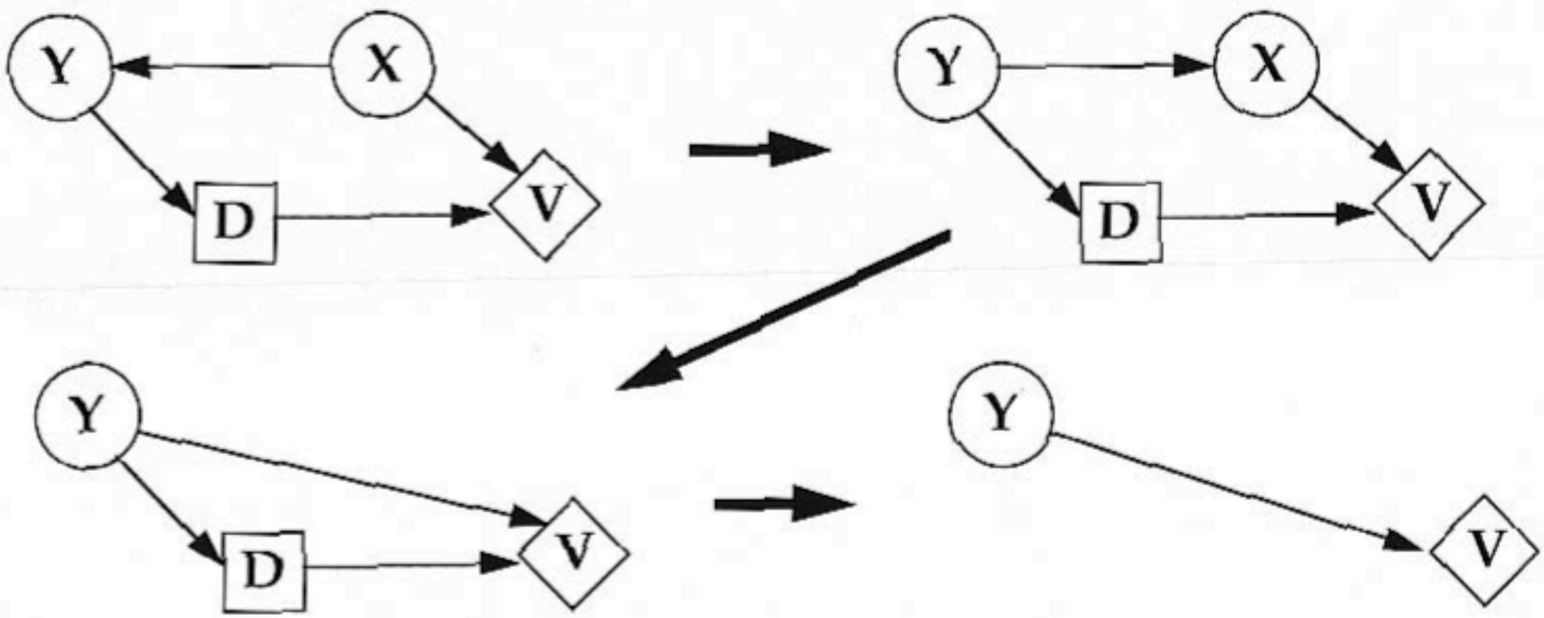


P(A | B,X,Y,Z) = P(B | A,Y,Z)*P(A | X,Y)/P(B | X,Y,Z)

Pa=Parents Pa(A)\Pa(B) parents of A who are not parents of B

Decision Example

- Reverse X-Y Arc
- Removal of X by Expectation into V
- Removal of D by maximization into V
  - gives you the optimal policy for D given Y



Next Lecture

- Introduction to Different Forms of Learning