Decision Graphs / Influence Diagrams

Preference Orderings

- A *preference ordering* $\succeq$ is a ranking of all possible states of affairs (worlds) S
  - these could be outcomes of actions, truth assts, states in a search problem, etc.
  - $s \succeq t$: means that state s is *at least as good as* t
  - $s \succ t$: means that state s *is strictly preferred to* t
- We insist that $\succeq$ is
  - reflexive: i.e., s $\succeq$ s for all states s
  - transitive: i.e., if s $\succeq$ t and t $\succeq$ w, then s $\succeq$ w
  - connected: for all states s,t, either s $\succeq$ t or t $\succeq$ s

Why Impose These Conditions?

- Structure of preference ordering imposes certain "rationality requirements" (it is a weak ordering)
- E.g., why transitivity?
  - Suppose you (strictly) prefer coffee to tea, tea to OJ, OJ to coffee
  - If you prefer X to Y, you will trade me Y plus $1 for X
  - I can construct a "money pump" and extract arbitrary amounts of money from you

Utilities

- Rather than just ranking outcomes, we must quantify our degree of preference
  - e.g., how much more important is *chc* than ∼*mess*
- A *utility function* $U : S \to \mathbb{R}$ associates a realvalued *utility* with each outcome.
  - $U(s)$ measures your *degree* of preference for s
- Note: U induces a preference ordering $\succeq_U$ over S defined as: s $\succeq_U$ t iff $U(s) \geq U(t)$
  - obviously $\succeq_U$ will be reflexive, transitive, connected

Expected Utility

- Under conditions of uncertainty, each decision d induces a distribution Pr_d over possible outcomes
  - $Pr_d(s)$ is probability of outcome s under decision d

- The *expected utility* of decision d is defined

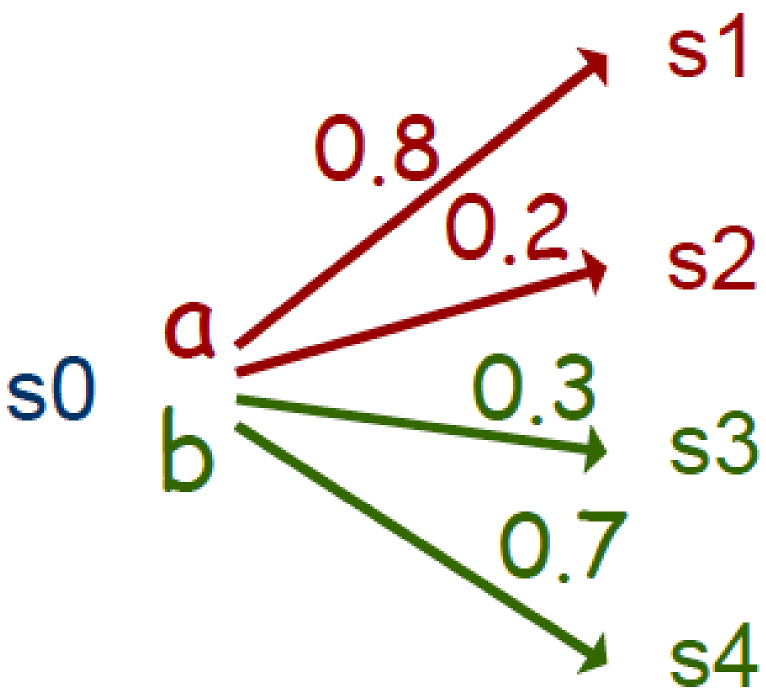
- The *principle of maximum expected utility (MEU)* states that the optimal decision under conditions of uncertainty is that with the greatest expected utility.

$$EU(d) = \sum_{s \in S} Pr_d(s)U(s)$$

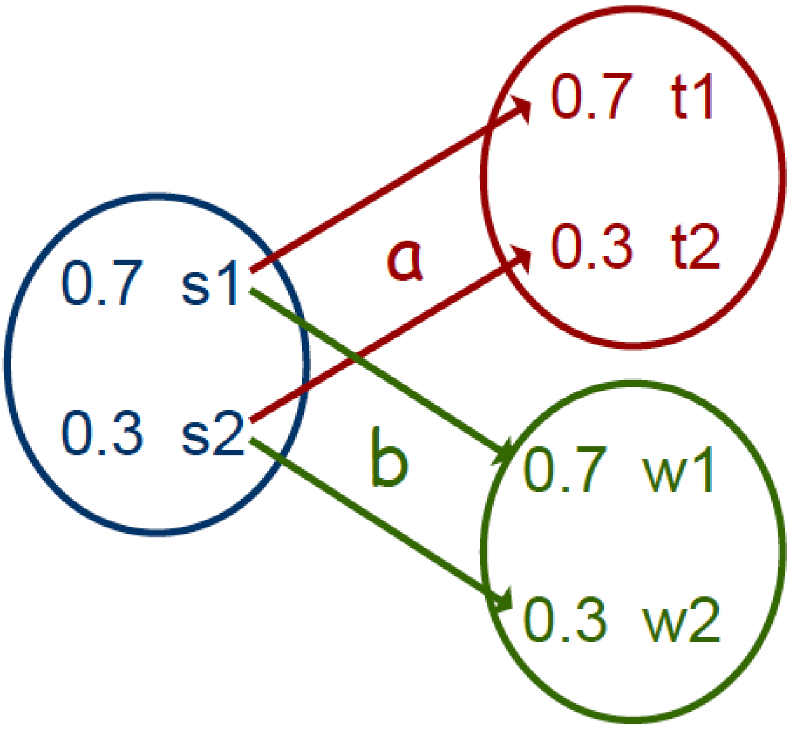
Decision Problems: Uncertainty

- A *decision problem under uncertainty* is:
  - a set of *decisions* D
  - a set of *outcomes* or states S
  - an *outcome function* $Pr : D \to \Delta(S)$
    - $\Delta(S)$ is the set of distributions over S (e.g., Prd)
  - a *utility function* U over S
- A solution to a decision problem under uncertainty is any $d^* \in D$ such that $EU(d^*) \succeq EU(d)$ for all $d \in D$
- Again, for single-shot problems, this is trivial

- Note that this viewpoint accounts for both:
  - uncertainty in action outcomes
  - uncertainty in state of knowledge
  - any combination of the two



Stochastic actions



Uncertain knowledge

Expected Utility: Notes

- Why MEU? Where do utilities come from?
  - underlying foundations of utility theory tightly couple utility with action/choice
  - a utility function can be determined by asking someone about their preferences for actions in specific scenarios (or "lotteries" over outcomes)
- Utility functions needn't be unique
  - if I multiply U by a positive constant, all decisions have same relative utility
  - if I add a constant to U, same thing
  - *U is unique up to positive affine transformation*

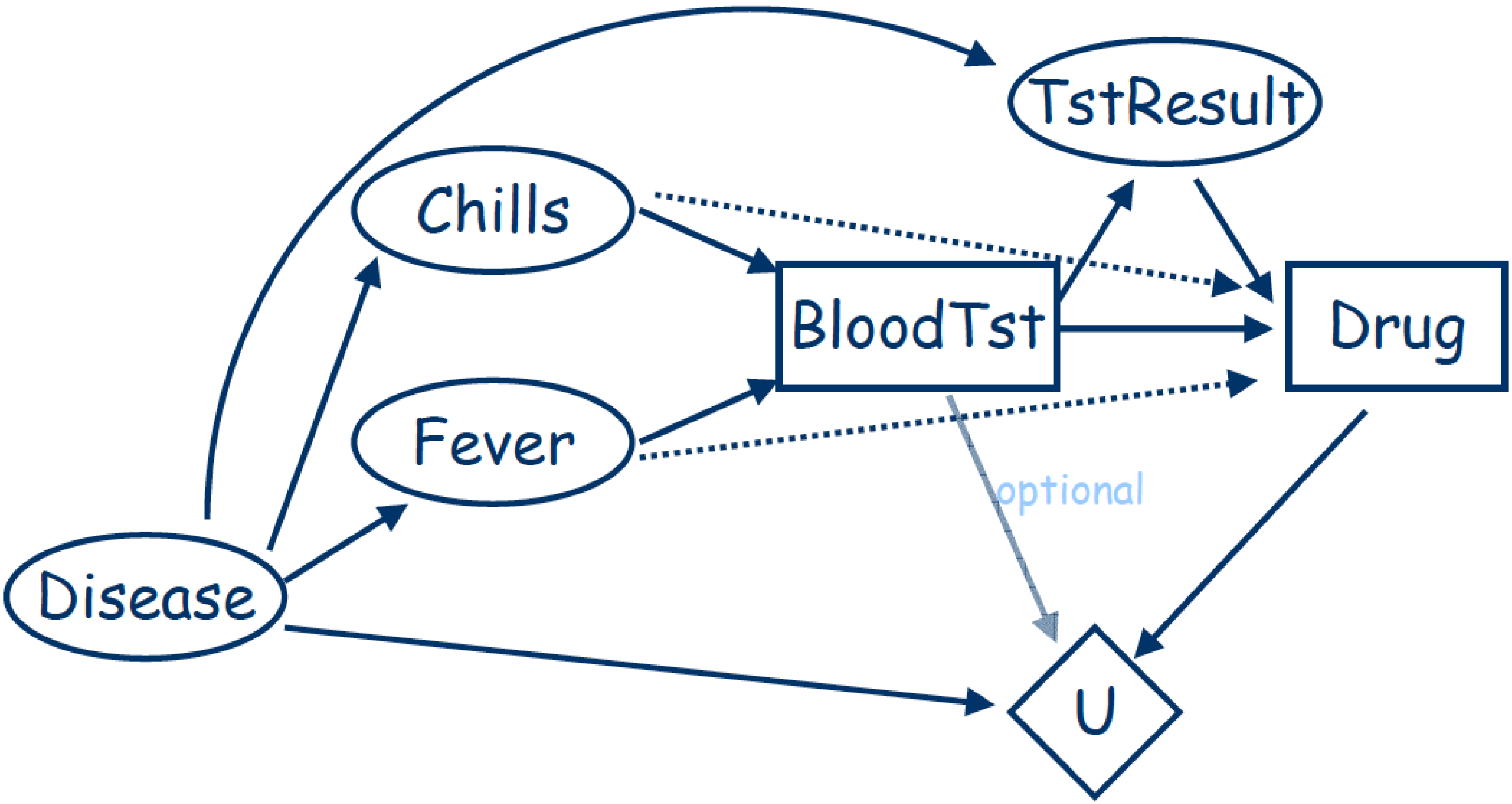
So What are the Complications?

- Outcome space is large
  - like all of our problems, states spaces can be huge
  - don't want to spell out distributions like Pr_d explicitly
  - Solution: Bayes nets (or related: *influence diagrams*)
- Decision space is large
  - usually our decisions are not one-shot actions
  - rather they involve sequential choices (like plans)
  - if we treat each plan as a distinct decision, decision space is too large to handle directly
  - Soln: use dynamic programming methods to *construct* optimal plans (actually generalizations of plans, called policies... like in game trees)

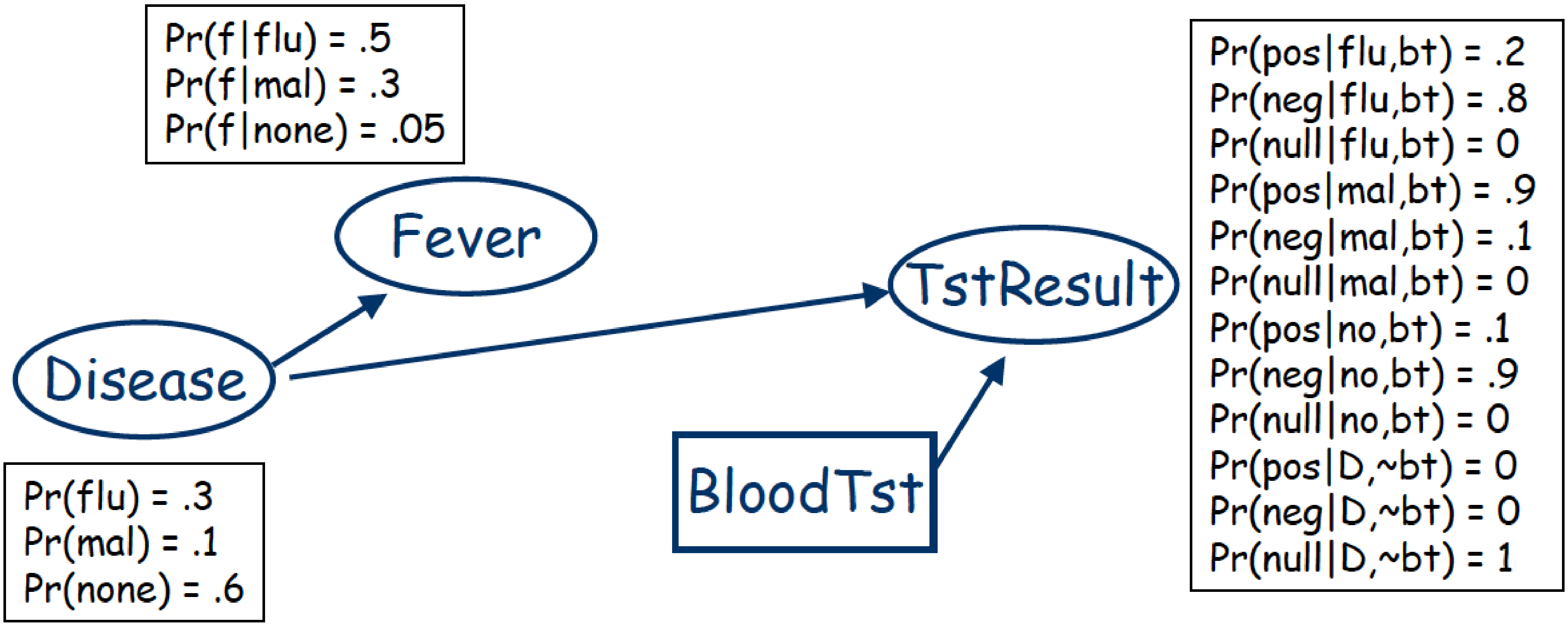
So What are the Complications?

- *Decision networks* (more commonly known as *influence diagrams*) provide a way of representing sequential decision problems
  - basic idea: represent the variables in the problem as you would in a BN
  - add decision variables – variables that you "control"
  - add utility variables – how good different states are

Sample Decision Network

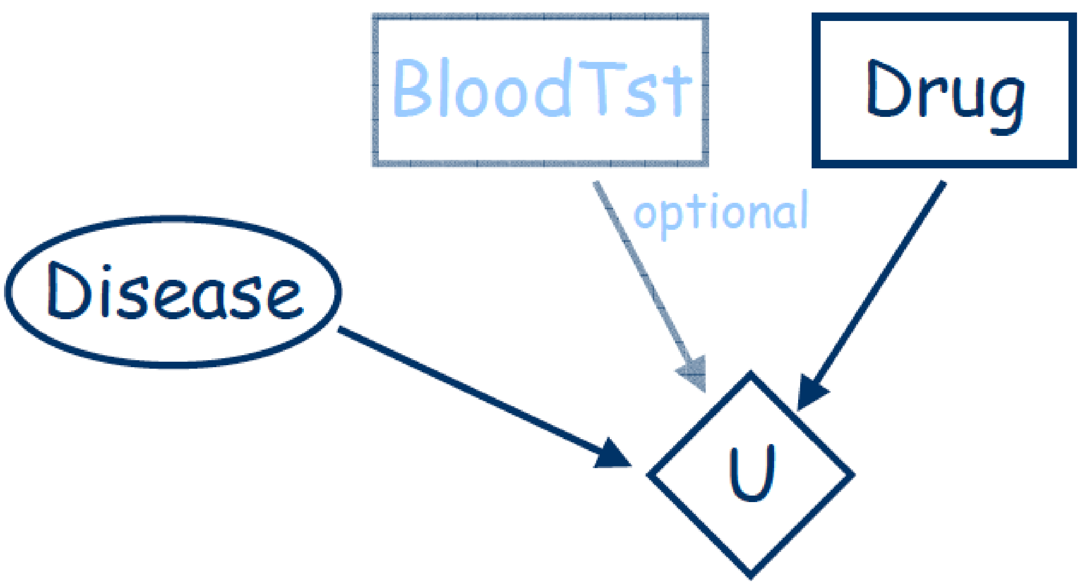


- **Chance nodes**
  - random variables, denoted by circles
  - as in a BN, probabilistic dependence on parents



- **Decision nodes**
  - variables decision maker sets, denoted by squares
  - parents reflect *information available* at time decision is to be made
- In example decision node: the actual values of Ch and Fev will be observed before the decision to take test must be made
  - agent can make different decisions for each instantiation of parents (i.e., policies)

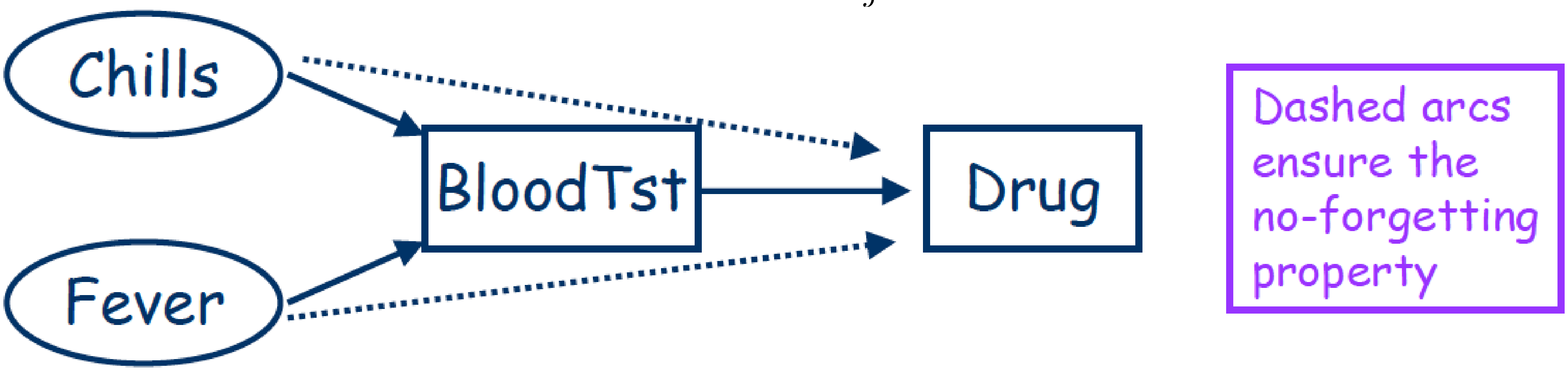
- **Value node**
  - specifies utility of a state, denoted by a diamond
  - utility depends *only on state of parents* of value node
  - generally: only one value node in a decision network
- Utility depends only on disease and drug



U(fludrug, flu) = 20
U(fludrug, mal) = -300
U(fludrug, none) = -5
U(maldrug, flu) = -30
U(maldrug, mal) = 10
U(maldrug, none) = -20
U(no drug, flu) = -10
U(no drug, mal) = -285
U(no drug, none) = 30

Decision Networks: Assumptions

- Decision nodes are totally ordered
  - decision variables $D_1, D_2, \ldots, D_n$
  - decisions are made in sequence
  - e.g., BloodTst (yes,no) decided before Drug (fd,md,no)
- *No-forgetting property*
  - any information available when decision D_i is made is available when decision D_j is made (for $i < j$)
  - thus all parents of D_i are parents of D_j



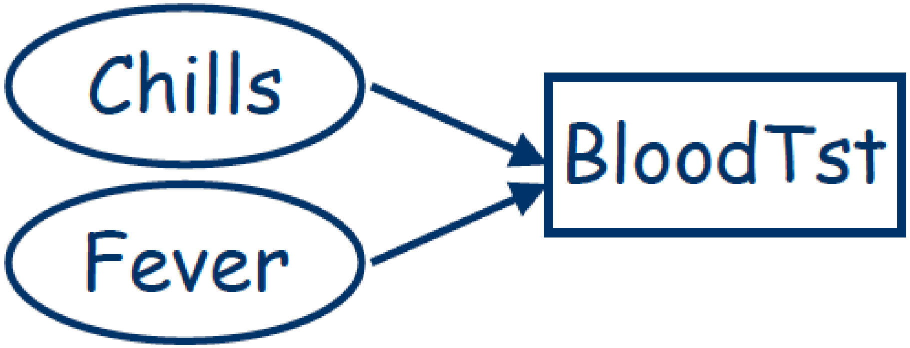
Policies

- Let $Par(D_i)$ be the parents of decision node D_i
  - $Dom(Par(Di))$ is the set of assignments to parents

- A policy δ is a set of mappings δ_i, one for each decision node D_i
  - $\delta_i : Dom(Par(D_i)) \to (D_i)$
  - δ_i associates a decision with each parent asst for D_i

- For example, a policy for BT might be:

$$\begin{aligned} \delta_{BT}(c, f) &= bt \\ \delta_{BT}(c, \sim f) &=\sim bt \\ \delta_{BT}(\sim c, f) &= bt \\ \delta_{BT}(\sim c, \sim f) &=\sim bt \end{aligned}$$



Value of a Policy

- Value of a policy δ is the expected utility given that decision nodes are executed according to δ
- Given associates $\boldsymbol{x}$ to the set $\boldsymbol{X}$ of all chance variables, let $\delta(\boldsymbol{x})$ denote the asst to decision variables dictated by δ
  - e.g., asst to D_1 determined by it's parents' asst in $\boldsymbol{x}$
  - e.g., asst to D_2 determined by it's parents' asst in $\boldsymbol{x}$ along with whatever was assigned to D1
  - etc.
- Value of δ:

$$EU(\delta) = \sum_{\boldsymbol{X}} P(\boldsymbol{X}, \delta(\boldsymbol{X}) U(\boldsymbol{X}, \delta(\boldsymbol{X}))$$

Optimal Policies

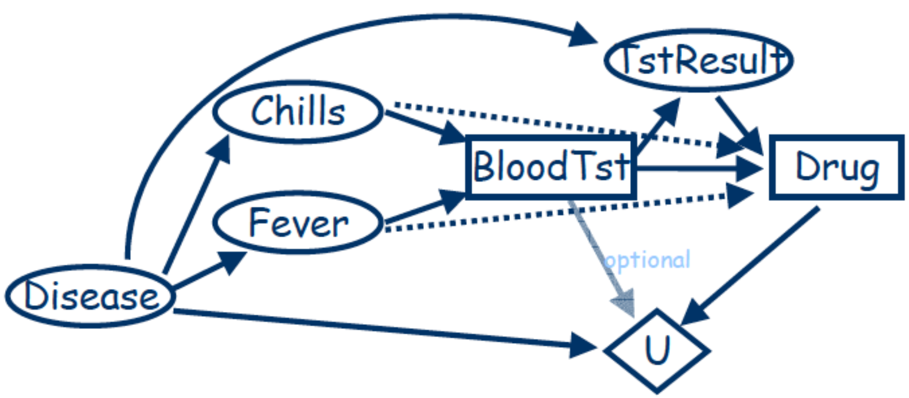
- An *optimal policy* is a policy δ^* such that $EU(\delta^*) \geq EU(\delta)$ for all policies δ
- We can use the dynamic programming principle yet again to avoid enumerating all policies
- We can also use the structure of the decision network to use variable elimination to aid in the computation

- We can work backwards as follows

- First compute optimal policy for Drug (last dec'n)
  - for each asst to parents (C,F,BT,TR) and for each decision value (D = md,fd,none), *compute the expected value* of choosing that value of D

  - set policy choice for each value of parents to be the value of D that has max value

  - eg: $\delta_D(c, f, bt, pos) = md$

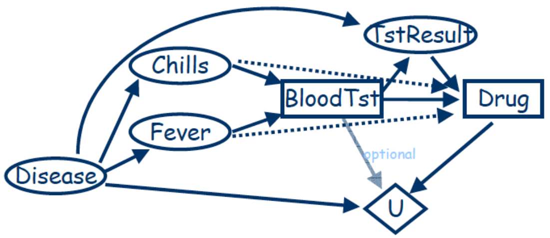


Computing the Best Policy

- Next compute policy for BT given policy $\delta_D(C, F, BT, TR)$ just determined for Drug
  - since $\delta_D(C, F, BT, TR)$ is fixed, we can treat Drug as a normal random variable with deterministic probabilities
  - i.e., for any instantiation of parents, value of Drug is fixed by policy δ_D
  - this means we can solve for optimal policy for BT just as before
  - only uninstantiated vars are random vars (once we fix *its* parents)

Computing the Best Policy

- How do we compute these expected values?
  - suppose we have asst $< c, f, bt, pos >$ to parents of $Drug$
  - we want to compute EU of deciding to set $Drug = md$
  - we can run variable elimination!
- Treat C, F, BT, TR, Dr as evidence
  - this reduces factors (e.g., U restricted to bt, md: depends on Dis)
  - eliminate remaining variables (e.g., only Disease left)
  - left with factor: $U() = \sum_{Dis} P(Dis|c, f, bt, pos, md)U(Dis)$
- We now know EU of doing $Dr = md$ when c, f, bt, pos true
- Can do same for fd, no to decide which is best

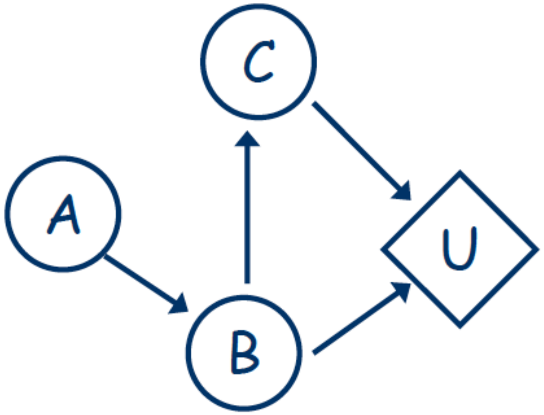


Computing Expected Utilities

- The preceding illustrates a general phenomenon
  - computing expected utilities with BNs is quite easy
  - utility nodes are just factors that can be dealt with using variable elimination

$$
\begin{aligned}
EU &= \sum_{A,B,C} P(A,B,C)U(B,C) \\
&= \sum_{A,B,C} P(C|B)P(B|A)P(A)U(B,C)
\end{aligned}
$$

- Just eliminate variables in the usual way



Optimizing Policies: Key Points

- If a decision node D has no decisions that follow it, we can find its policy by instantiating each of its parents and computing the expected utility of each decision for each parent instantiation
  - no-forgetting means that all other decisions are instantiated (they must be parents)
  - its easy to compute the expected utility using VE
  - the number of computations is quite large: we run expected utility calculations (VE) for each parent instantiation together with each possible decision D might allow
  - policy: choose max decision for each parent instant'n

Optimizing Policies: Key Points

- When a decision D node is optimized, it can be treated as a random variable
  - for each instantiation of its parents we now know what value the decision should take
  - just treat policy as a new CPT: for a given parent instantiation $\boldsymbol{x}$, D gets $\delta(\boldsymbol{x})$ with probability 1 (all other decisions get probability zero)
- If we optimize from last decision to first, at each point we can optimize a specific decision by (a bunch of) simple VE calculations
  - it's successor decisions (optimized) are just normal nodes in the BNs (with CPTs)

Decision Network Notes

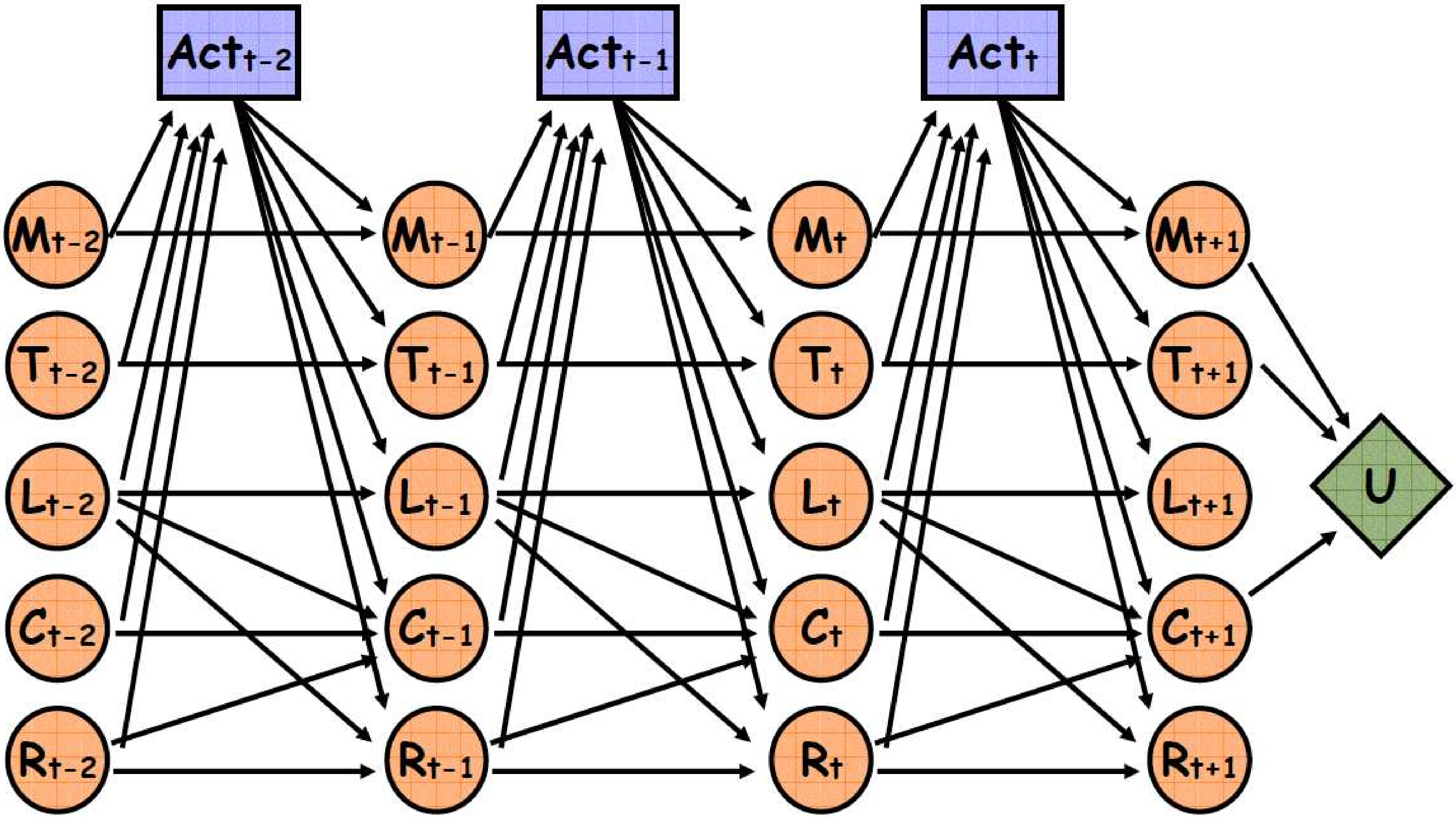
- Decision networks commonly used by decision analysts to help structure decision problems

- Much work put into computationally effective techniques to solve these
  - common trick: replace the decision nodes with random variables at outset and solve a plain Bayes net (a subtle but useful transformation)

- Complexity much greater than BN inference
  - we need to solve a number of BN inference problems

  - one BN problem for each setting of decision node parents and decision node value

DBN-Decision Nets for Planning



Decision Network Notes

- In example on previous slide:
  - we assume the state (of the variables at any stage) is fully observable
    - hence all time t vars point to time t decision
  - this means the state at time t d-separates the decision at time t-1 from the decision at time t-2
  - so we ignore "no-forgetting" arcs between decisions
    - once you *know* the state at time t, what you *did* at time t-1 to get there is irrelevant to the decision at time t-1
- If the state were not fully observable, we could not ignore the "no-forgetting" arcs

A Detailed Decision Net Example

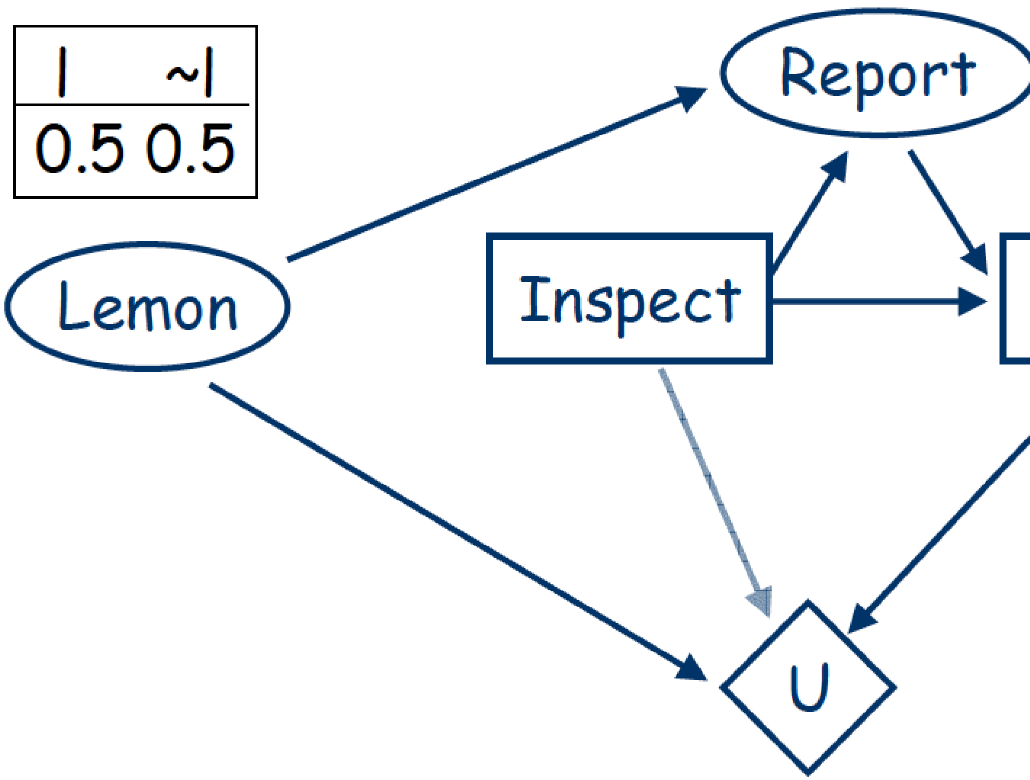
- Setting: you want to buy a used car, but there's a good chance it is a "lemon" (i.e., prone to breakdown). Before deciding to buy it, you can take it to a mechanic for inspection. S/he will give you a report on the car, labelling it either "good" or "bad". A good report is positively correlated with the car being sound, while a bad report is positively correlated with the car being a lemon.

- The report costs $50 however. So you could risk it, and buy the car without the report.

- Owning a sound car is better than having no car, which is better than owning a lemon.

Car Buyer's Network

l	~l
0.5	0.5



Rep: good,bad,none

	g	b	n
l i	0.2	0.8	0
~l i	0.9	0.1	0
l ~i	0	0	1
~l ~i	0	0	1

Utility

b l	-600
b ~l	1000
~b l	-300
~b~l	-300

-50 if inspect

Evaluate Last Decision: Buy (1)

- $EU(B|I,R) = \sum_L P(L|I,R,B)U(L,B)$
- $I = i, R = g$:

$$\begin{aligned} EU(buy) &= P(l|i,g)U(l,buy) + P(\sim l|i,g)U(\sim l,buy) - 50 \\ &= .18 \cdot -600 + .82 \cdot 1000 - 50 = 662 \\ EU(\sim buy) &= P(l|i,g)U(l,\sim buy) + P(\sim l|i,g)U(\sim l,\sim buy) - 50 \\ &= -300 - 50 = -350 (-300 \text{ indep. of lemon}) \end{aligned}$$

- So optimal $\delta_{Buy}(i,g) = buy$

Evaluate Last Decision: Buy (2)

- $I = i, R = b$:

$$\begin{aligned}
EU(buy) &= P(l|i,b)U(l,buy) + P(\sim l|i,b)U(\sim l, buy) - 50 \\
&= .89 \cdot -600 + .11 \cdot 1000 - 50 = -474 \\
EU(\sim buy) &= P(l|i,b)U(l,\sim buy) + P(\sim l|i,b)U(\sim l, \sim buy) - 50 \\
&= -300 - 50 = -350 (-300 \text{ indep. of lemon})
\end{aligned}$$

- So optimal $\delta_{Buy}(i,b) =\sim buy$

Evaluate Last Decision: Buy (3)

- $I = \sim i, R = g$ (note: no inspection cost subtracted):

$$\begin{aligned} EU(buy) &= P(l|\sim i, g)U(l, buy) + P(\sim l|\sim i, g)U(\sim l, buy) \\ &= .5 \cdot -600 + .5 \cdot 1000 = 200 \\ EU(\sim buy) &= P(l|\sim i, g)U(l, \sim buy) + P(\sim l|\sim i, g)U(\sim l, \sim buy) - 50 \\ &= -300 - 50 = -350(-300 \text{ indep. of lemon}) \end{aligned}$$

- So optimal $\delta_{Buy}(\sim i, g) = \sim buy$

- So optimal policy for Buy is:
  - $\delta_{Buy}(i, g) = buy; \delta_{Buy}(i, b) = \sim buy; \delta_{Buy}(\sim i, n) = buy$

- Note: we don't bother computing policy for $(i, \sim n)$, $(\sim i, g)$, or $(\sim i, b)$, since these occur with probability 0

Evaluate First Decision: Inspect

- $EU(I) = \sum_{L,R} P(L, R|I)U(L, \boldsymbol{\delta_{Buy}(I, R)})$
  - where $P(R, L|I) = P(R|L, I)P(L|I)$

$$\begin{aligned} EU(i) &= .1 \cdot -600 + .4 \cdot -300 + .45 \cdot 1000 + .05 \cdot -300 - 50 \\ &= 237.5 - 50 = 187.5 \\ EU(\sim i) &= P(l| \sim i, n)U(l, buy) + P(\sim l| \sim i, n)U(\sim l, buy) \\ &= .5 \cdot -600 + .5 \cdot 1000 = 200 \end{aligned}$$

- So optimal $\delta_{Inspect}(\sim i) = buy$

	$P(R, L\|I)$	$\boldsymbol{\delta_{Buy}}$	$U(L, \boldsymbol{\delta_{Buy}})$
g, l	0.1	buy	$-600 - 50 = -650$
$g, \sim l$	0.45	buy	$1000 - 50 = 950$
b, l	0.4	$\sim buy$	$-300 - 50 = -350$
$b, \sim l$	0.05	$\sim buy$	$-300 - 50 = -350$

Value of Information

- So optimal policy is: don't inspect, buy the car
  - EU = 200
  - Notice that the EU of inspecting the car, then buying it iff you get a good report, is 237.5 less the cost of the inspection (50). So inspection not worth the improvement in EU.
  - But suppose inspection cost \$25: then it would be worth it ($EU = 237.5 - 25 = 212.5 > EU(\sim i)$)
  - The *expected value of information* associated with inspection is 37.5 (it improves expected utility by this amount ignoring cost of inspection). How? Gives opportunity to change decision ($\sim$ *buy* if bad).
  - You should be willing to pay up to \$37.5 for the report

Influence Diagrams

Up to now, we used Bayesian networks for

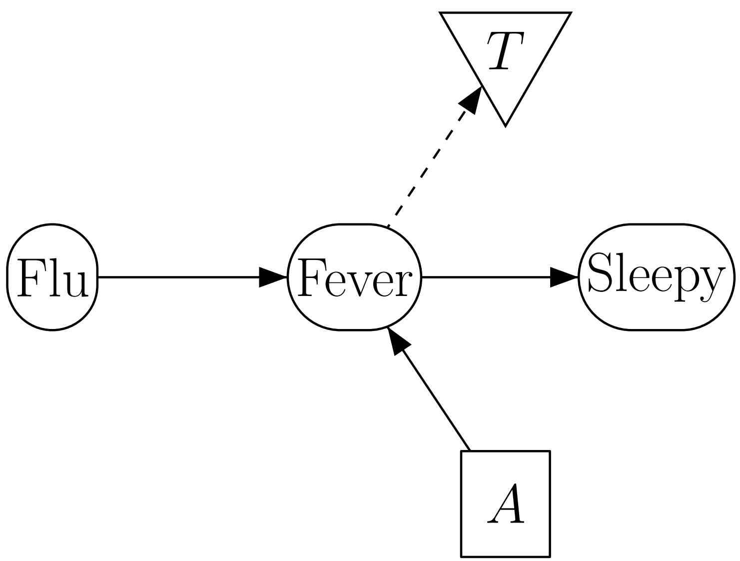
- modeling (in)dependence relations between random/chance variables
- quantifying the strength of these relations by assigning (conditional) probabilities
- update these probabilities after evidence observations

However, in practical, this is only a part of a more complex task: **decision making under uncertainty**.

If a set of actions solves a problem, we have to choose one particular action based on predefined criteria, e. g. costs and/or gains.

Therefore, we will now augment the current framework with special nodes that serve these purposes.

Example: Observations and Actions



$T\ldots$ Temperature

$A\ldots$ Aspirine

- Rectangular nodes: intervening actions/decisions
- Triangular nodes: test actions/observations
- Observations may change probabilities of nodes that are causes:

  Observing $T = 37°C$ decreases probability of Fever and Flu (and, of course, Sleepy).

- The impact of intervening actions can only follow the direction of the (causal) edges:

  Taking Aspirine (A) decreases the probability of Fever and Sleepy and may result in an alike observation for T. However, it cannot change the state for Flu since Aspirine only eases the pain and does not kill viruses.

Example: Utilities

Mildew Fungus Infestation (dt. Mehltau-Befall)

Before the harvest, a farmer checks the state of his crop and decides whether to apply a fungi treatment or not.

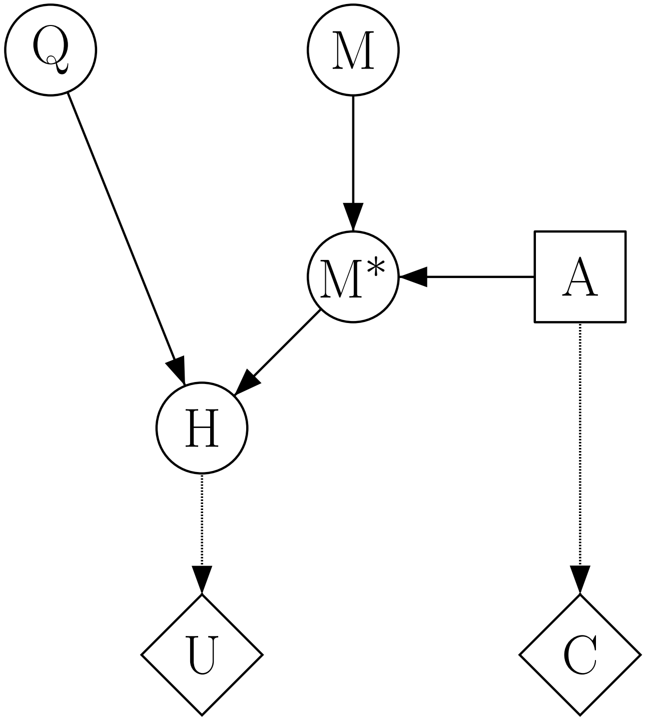
- Q — Quality of the crop
- M — Mildew infestation severity
- H — Harvest quality
- A — Action to be taken
- M* — Mildew infestation after action A
- U — Utility function of the harvest (i. e. the benefit)
- C — Utility functon of the action (i. e. the treatment costs)

$\longrightarrow$ edges leading to chance nodes

$\dashrightarrow$ edges leading to decision nodes

$\cdots\rightarrow$ edges leading to utility nodes

Example: Utilities (2)



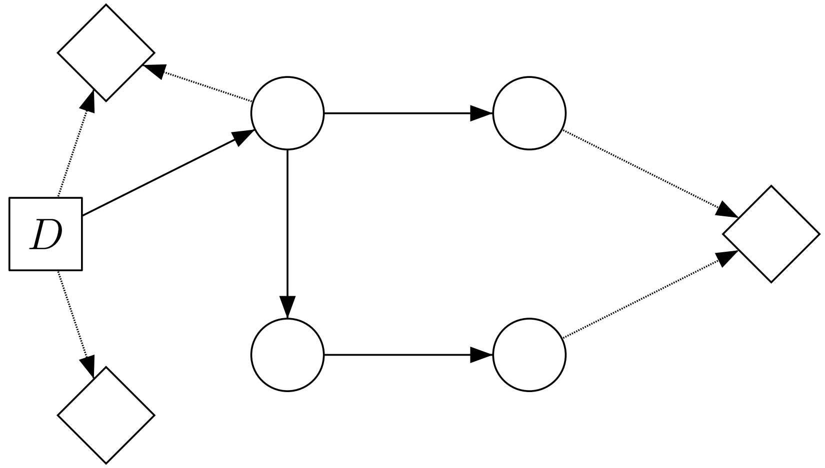
- Diamond-shaped nodes: utility functions (costs/benefits)
- Given the quality of the crops and the mildew state, which action maximizes the benefit?
- $C(\mathrm{A}) < 0$
- $U(\mathrm{H}) \geq 0$
- Expected total utility of action $\mathrm{A} = a$:

$$\mathrm{E}(U(a \mid q, m)) = C(a) + \sum_h U(h) \cdot P(h \mid a, q, m)$$

Single-Action Models

A single-action model consists of

- a Bayesian network representing the chance nodes
- one decision (action) node
- a set of utility nodes
- decision nodes can affect chance and utility nodes
- utility nodes can be affected by chance and decision nodes



Single-Action Models (2)

Given n utility nodes $U_1, \dots, U_n$ and assuming they all depend on only one respective chance node X_i, the total expected utility given a decision $D = d$ and (chance node) evidence e is defined as:
vskip-2mm

$$\mathrm{E}(U(d \mid e)) = \sum_{i=1}^{n} \sum_{x \in \mathrm{dom}(X_i)} U_1(x_1) \cdot P(x_1 \mid d, e)$$

The optimal decision d^* is then chosen:

$$d^* = \operatorname*{arg\,max}_{d \in \mathrm{dom}(D)} \mathrm{E}(U(d \mid e))$$

Influence Diagrams

An influence diagram consists of a directed acyclic graph over chance nodes, decision nodes and utility nodes that obey the following structural properties:

- there is a directed path comprising all decision nodes
- utility nodes cannot have children
- decision and chance nodes are discrete
- utility nodes do not have states
- chance nodes are assigned potential tables given their parents (including decision nodes)
- each utility node U gets assigned a real-valued utility function over its parents

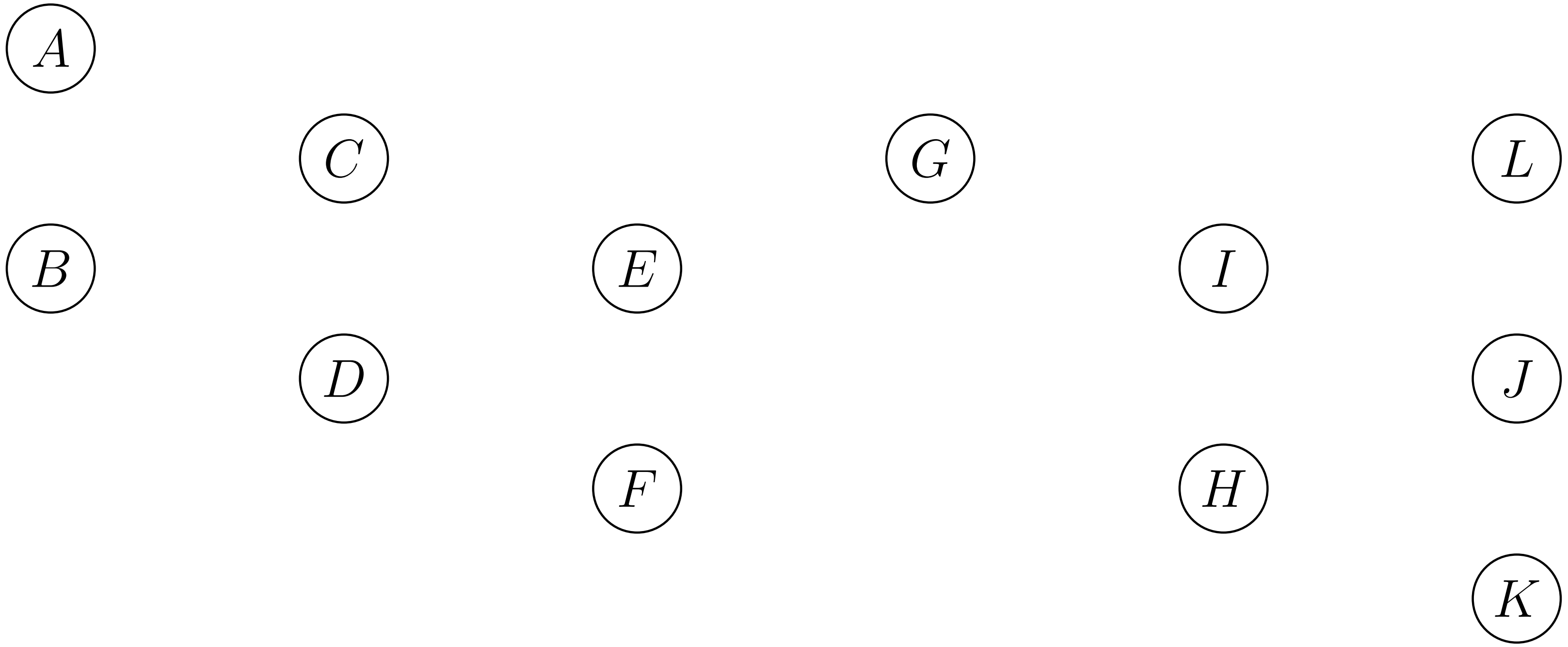
$$U : \bigtimes_{X \in \mathrm{parents}(U)} \mathrm{dom}(X) \to \mathbb{R}$$

Influence Diagrams (2)

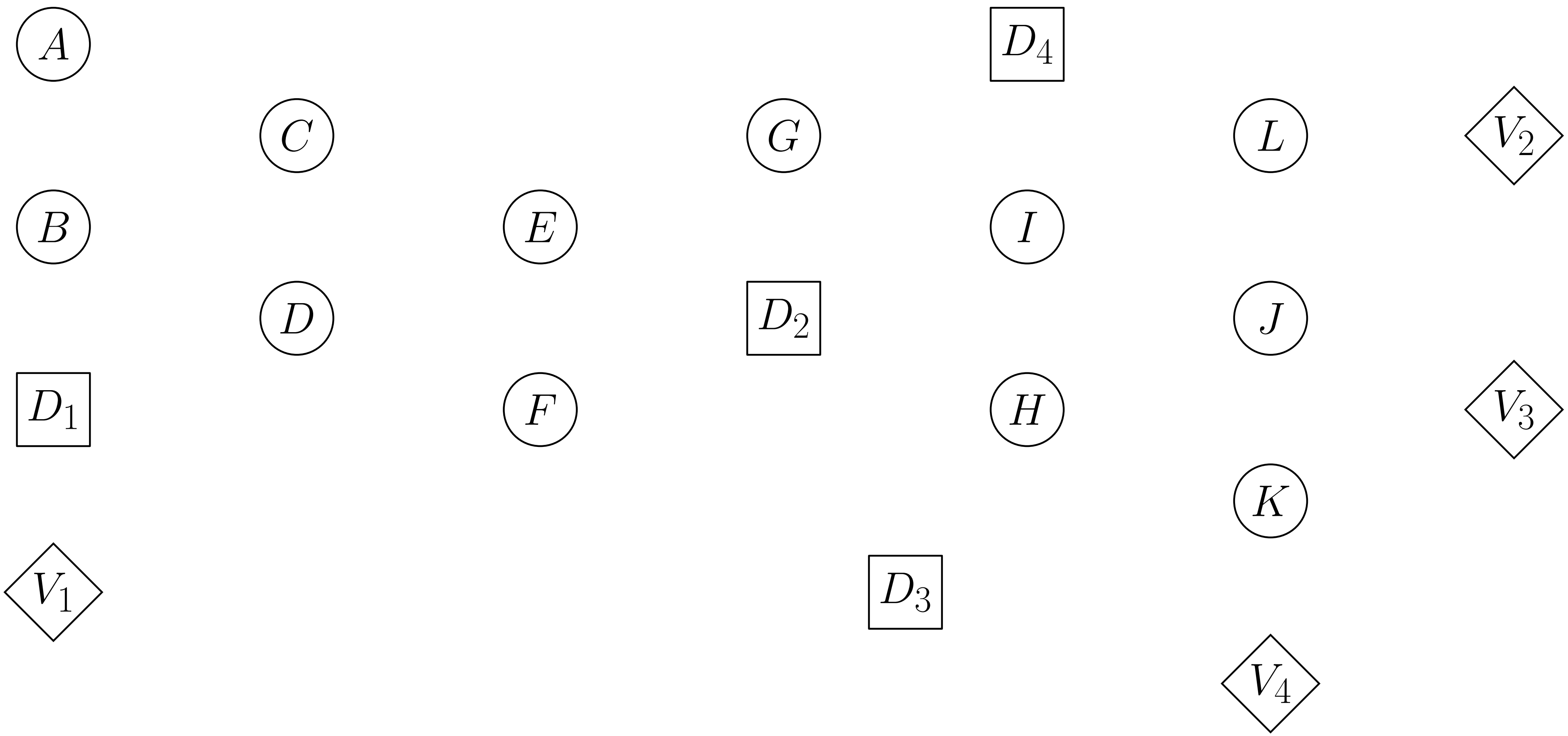
- Links into decision nodes carry no quantitative information, they only introduce a temporal ordering.
- The required path between the decision nodes induces a temporal partition of the chance nodes:

  If there are n decision nodes, then for $1 \leq i < n$ the set I_i represents all chance nodes that have to be observed after decision D_i but before decision D_{i+1}.
- I_0 is the set of chance nodes to be observed before any decision.
- I_n is the set of chance nodes that are not observed.

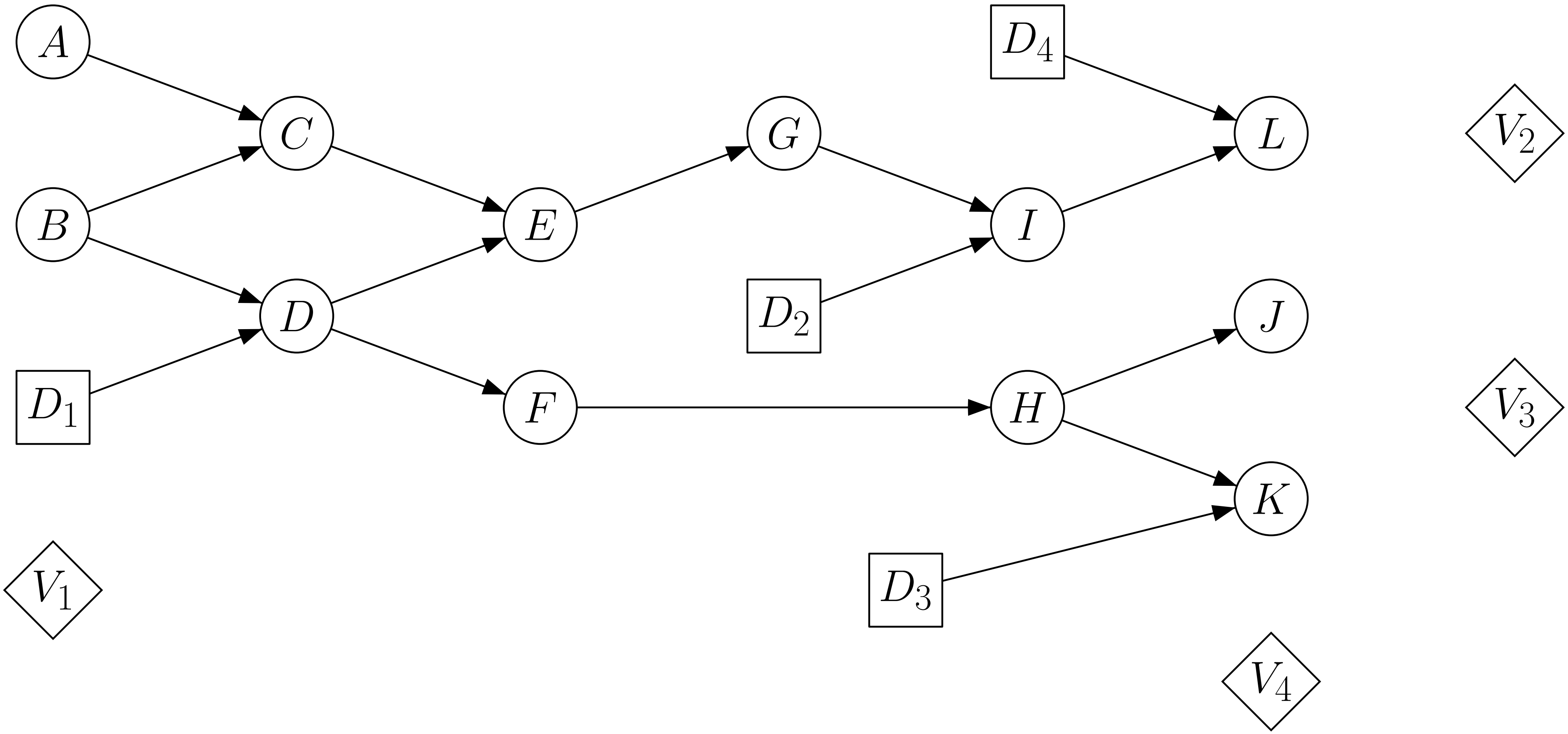
Influence Diagrams (3)



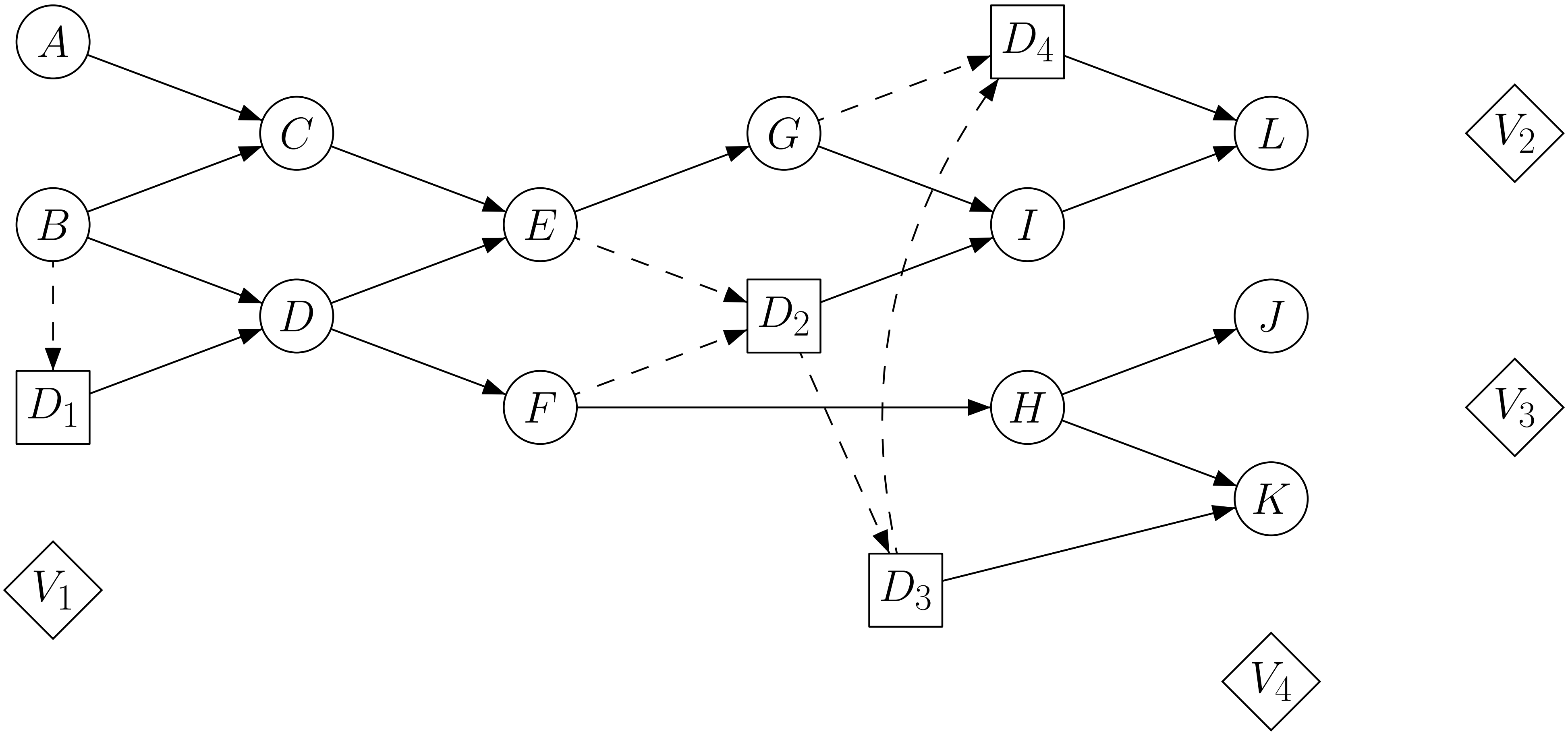
Influence Diagrams (3)



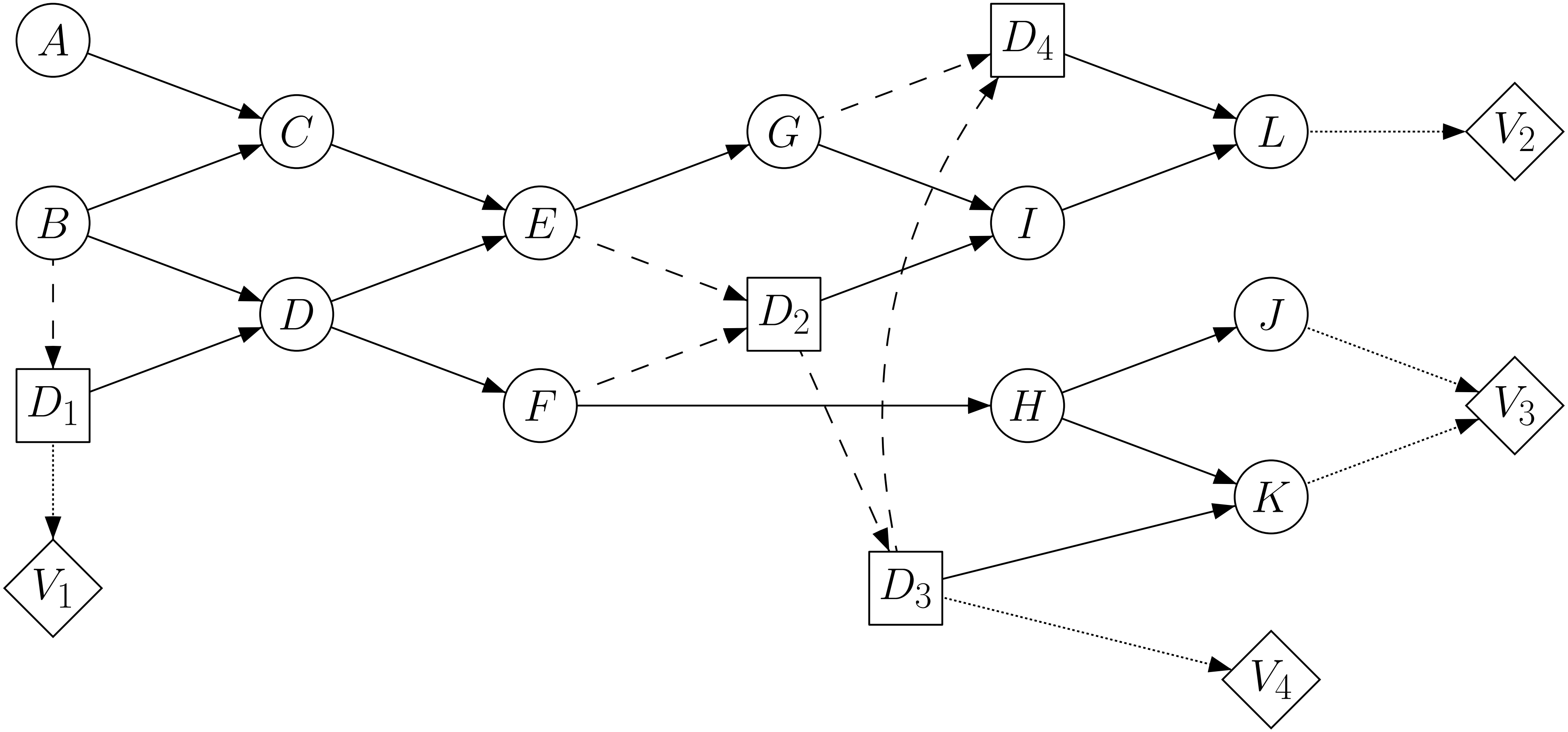
Influence Diagrams (3)



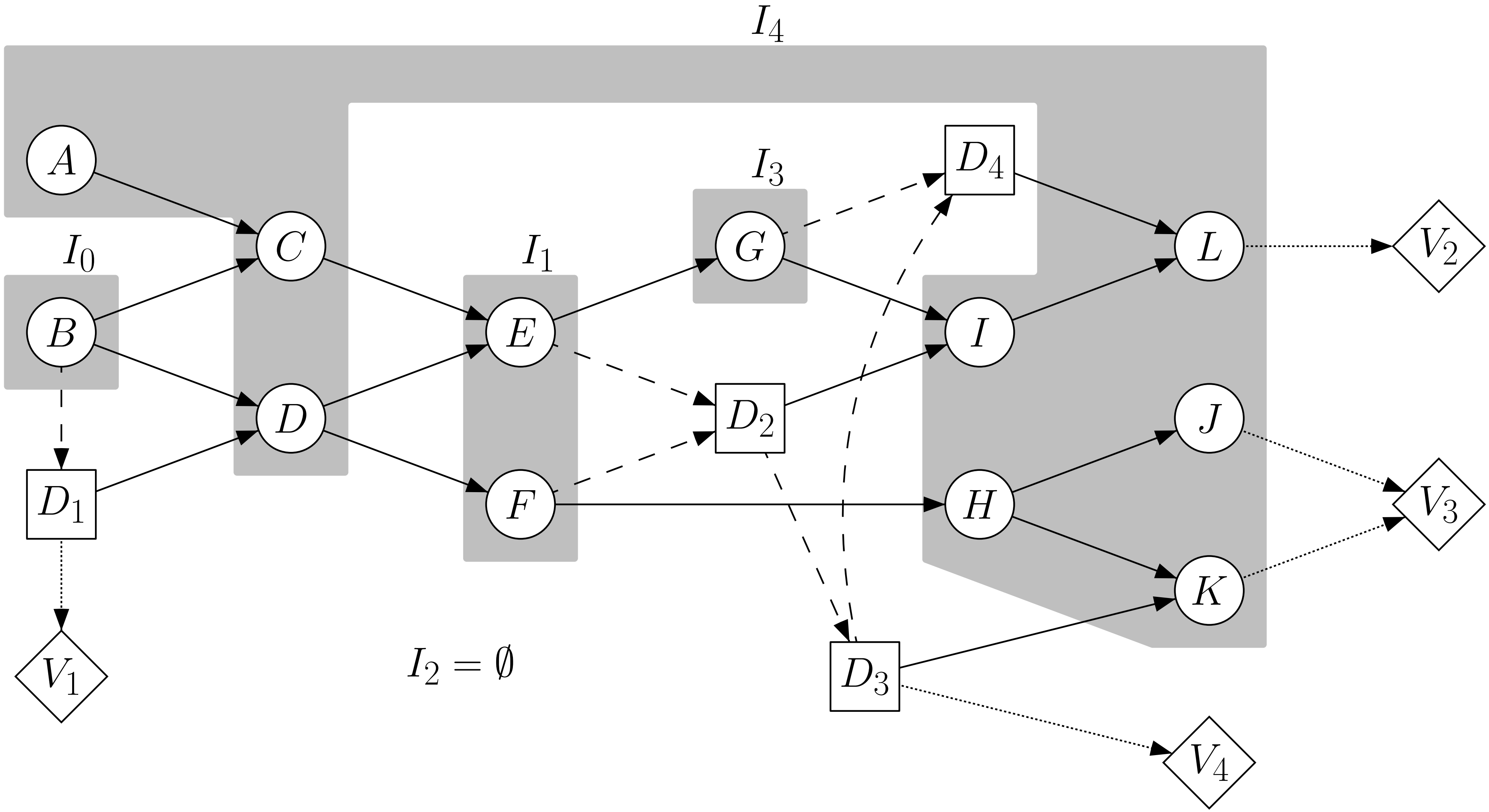
Influence Diagrams (3)



Influence Diagrams (3)



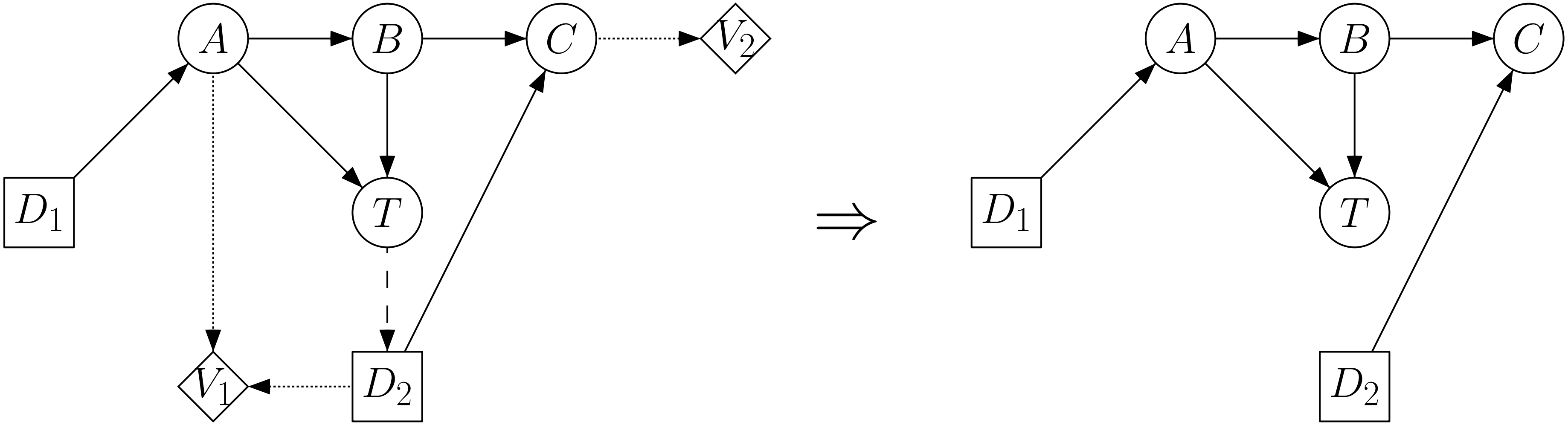
Influence Diagrams (3)



d-Separation in Influence Diagrams

To be able to use the d-separation, we need to preprocess the graphical structure of an influence diagram as follows:

- remove all utility nodes (and the edges towards them)
- remove edges that point to decision nodes



For example: $C \perp\!\!\!\perp T \mid B$ or $\{A, T\} \perp\!\!\!\perp D_2 \mid \emptyset$.

Chain Rule

The semantics of an influence diagram disallow some probabilities:

- $P(D)$ for a decision node D has no meaning
- $P(A \mid D)$ has no meaning unless a decision $d \in \mathrm{dom}(D)$ has been chosen

Given an influence diagram G with U_C being the set of chance nodes and U_D being the set of decision nodes, we can factorize P as follows:

$$P(U_C \mid U_D) = \prod_{X \in U_C} P(X \mid \mathrm{parents}(X))$$

Solutions to Influence Diagrams

- Given: an influence diagram
- Desired: a strategy which decision(s) to make

Policy

A *policy* for decision D_i is a mapping σ_i, which for any configuration of the past of D_i yields a decision for D_i, i. e.

$$\sigma_i(I_0, D_1, I_1, \ldots, D_{i-1}, I_{i-1}) \in \mathrm{dom}(D_i)$$

Strategy

A *strategy* for an influence diagram is a set of policies, one for each decision node.

Solution

A *solution* to an influence diagram is a strategy maximizing the expected utility.

Solutions to Influence Diagrams (2)

Assume, we are given an influence diagram G over $U = U_C \cup U_D$ and U_V.

- U_C …set of chance nodes
- U_D …set of decision nodes and
- $U_V = \{V_i\}$ …set of utility nodes

Further, we know the following temporal order:

$$I_0 \prec D_1 \prec I_1 \prec \cdots \prec D_n \prec I_n$$

The total utility V be defined as the sum of all utility nodes: $V = \sum_i V_i$

Solutions to Influence Diagrams (3)

- An optimal policy for D_i is

$$\sigma_i(I_0, D_1, \ldots, I_{i-1}) = \arg\max_{d_i} \sum_{I_i} \max_{d_{i+1}} \cdots \max_{d_n} \sum_{I_n} P(U_C \mid U_D) \cdot V$$

  where $d_x \in \mathrm{dom}(D_x)$.

- The expected utility from following policy σ_i (and acting optimally in the future) is

$$\rho_i(I_0, D_1, \ldots, I_{i-1}) = \frac{\max_{d_i} \sum_{I_i} \max_{d_{i+1}} \cdots \max_{d_n} \sum_{I_n} P(U_C \mid U_D) \cdot V}{P(I_0, \ldots, I_{i-1} \mid D_1, \ldots, D_{i-1})}$$

  where $d_x \in \mathrm{dom}(D_x)$.

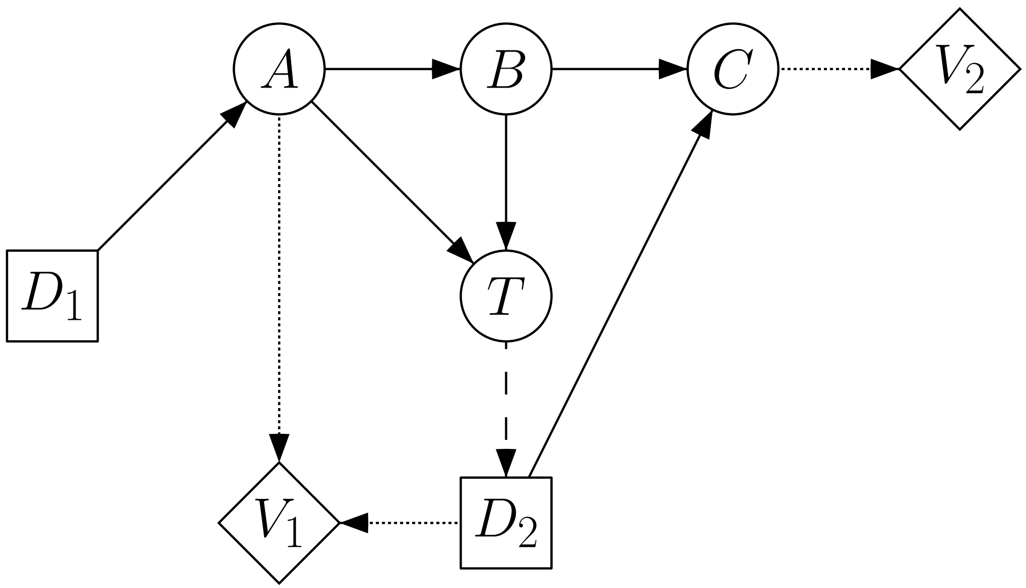
Solutions to Influence Diagrams (4)

- An optimal strategy yields the maximum expected utility of

$$\mathrm{MEU}(G) = \sum_{I_0} \max_{d_1} \sum_{I_1} \max_{d_2} \cdots \max_{d_n} \sum_{I_n} P(U_C \mid U_D) \cdot V$$

- $\sum_{I_i}$ means (sum-)marginalizing over all nodes in I_i
- $\max_{d_i}$ means taking the maximum over all $d_i \in \mathrm{dom}(D_i)$ and thus (max-)marginalizing over D_i
- Everytime I_i is marginalized out, the result is used to determine a policy for D_i.
- Marginalization in reverse temporal order
- $\Rightarrow$ use simplification techniques from the Bayesian network realm to simplify the joint probability distribution $P(U_C \mid U_D)$

Example



$P(A \mid D_1)$	$d_1^{(1)}$	$d_1^{(2)}$
y	0.2	0.8
n	0.8	0.2

$P(B \mid A)$	y	n
y	0.8	0.2
n	0.2	0.8

$P(T \mid A, B)$	y, y	y, n	n, y	n, n
y	0.9	0.5	0.5	0.1
n	0.1	0.5	0.5	0.9

$V_2(C)$	
y	10
n	0

$V_1(A, D_2)$	$d_2^{(1)}$	$d_2^{(2)}$
y	3	0
n	0	2

Utility functions

$P(C \mid B, D_2)$	y, $d_2^{(1)}$	y, $d_2^{(2)}$	n, $d_2^{(1)}$	n, $d_2^{(2)}$
y	0.9	0.5	0.5	0.9
n	0.1	0.5	0.5	0.1

Chance potentials

Example (2)

For D_2 we can read from the graph:

$$I_0 = \emptyset \qquad I_1 = \{T\} \qquad I_2 = \{A, B, C\}$$

Thus, σ_2 can be solved to the following strategy:

$\sigma_2(\emptyset, D_1, \{T\})$	$d_1^{(1)}$	$d_1^{(2)}$
y	$d_2^{(1)}$	$d_2^{(1)}$
n	$d_2^{(2)}$	$d_2^{(2)}$

$\rho_2(\emptyset, D_1, \{T\})$	$d_1^{(1)}$	$d_1^{(2)}$
y	9.51	11.29
n	10.34	8.97

Finally, $\sigma_1 = d_1^{(2)}$ and $\mathrm{MEU}(G) = 10.58$.