Making Decisions Under Uncertainty

What an agent should do depends on:

- What the agent believes. Not only the most likely state of affairs, but all ways the world could be, given the agent's knowledge. Sensing the world updates the agent's beliefs by conditioning on what is sensed.
- The agent's goals. When an agent has to reason under uncertainty, it has to consider not only what will most likely happen but everything that may possibly happen.

Decision theory specifies how to trade off the desirability and probabilities of the possible outcomes for competing actions.

Decision Variables

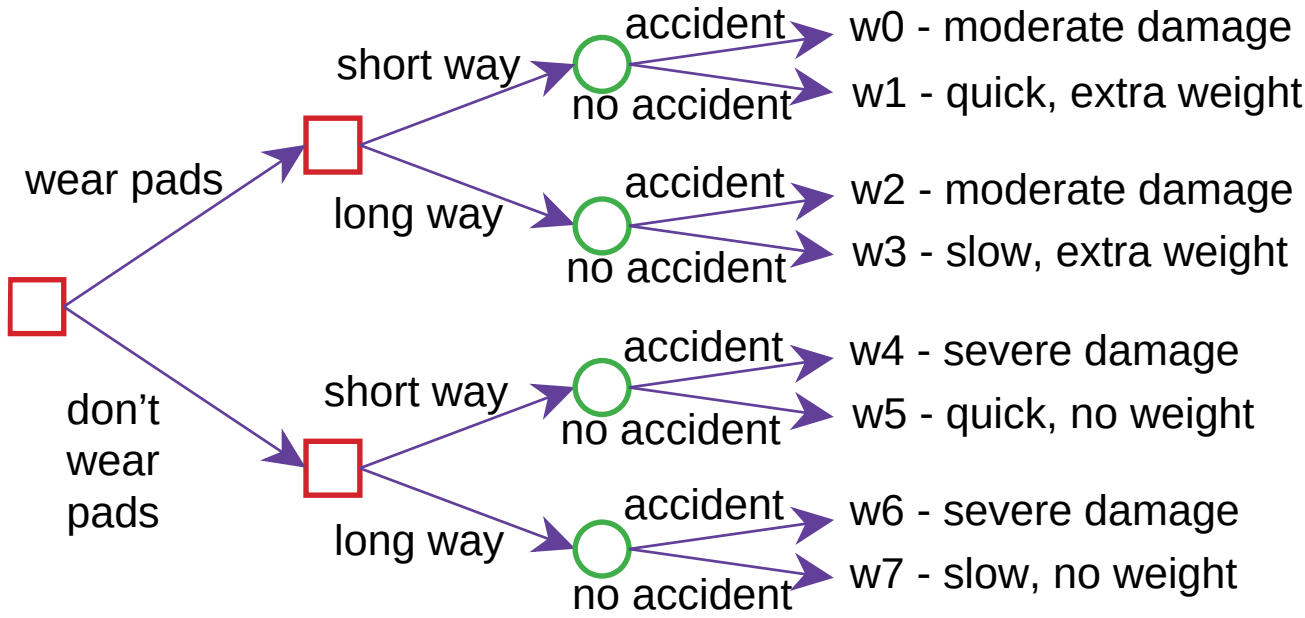
- Decision variables are like random variables that an agent gets to choose the value of.
- A possible world specifies the value for each decision variable and each random variable.
- For each assignment of values to all decision variables, the measures of the worlds satisfying that assignment sum to 1.
- The probability of a proposition is undefined unless you condition on the values of all decision variables.

Decision Tree for Delivery Robot

The robot can choose to wear pads to protect itself or not.

The robot can choose to go the short way past the stairs or a long way that reduces the chance of an accident.

There is one random variable of whether there is an accident.



Expected Values

The expected value of a numerical random variable is its average value, weighting possible worlds by their probability.

Suppose V is a numerical random variable and ω is a possible world. Let $\rho(V, \omega)$ be the value x such that $\omega \models V = x$.

The expected value of V is

$$\mathbf{E}(V) = \sum_{\omega \in \Omega} \rho(V, \omega) \times \mu(\omega).$$

The conditional expected value of V given e is

$$\mathbf{E}(V|e) = \sum_{\omega \models e} \rho(V, \omega) \times \mu_e(\omega).$$

Utility

- Utility is a measure of desirability of worlds to an agent.
- Let U be a real-valued random variable such that $\rho(U, \omega)$ represents how good the world is to an agent.
- Simple goals can be specified by: worlds that satisfy the goal have utility 1; other worlds have utility 0.
- Often utilities are more complicated: for example, made up from the amount of damage to a robot, how much energy it has used up, what goals are achieved, and how much time it has taken.

Single decisions

In a single decision, the agent chooses a value for each decision variable. Let compound decision variable d be the tuple of all original decision variables. The agent can choose $d = d_i$ for any $d_i \in dom(d)$.

The expected utility of decision $d = d_i$ is $\mathbf{E}(U|d = d_i)$.

An optimal single decision is the decision $d = d_{max}$ whose expected utility is maximal:

$$\mathbf{E}(U|d = d_{max}) = \max_{d_i \in dom(d)} \mathbf{E}(U|d = d_i).$$

Sequential Decisions

- An intelligent agent doesn't make a multi-step decision and carry it out without considering revising it based on future information.
- A more typical scenario is where the agent:
  observes, acts, observes, acts, …
- Subsequent actions can depend on what is observed.
  What is observed depends on previous actions.
- Often the sole reason for carrying out an action is to provide information for future actions.
  For example: diagnostic tests, spying.

Sequential decision problems

- A sequential decision problem consists of a sequence of decision variables $d_1, \ldots, d_n$.
- Each d_i has an information set of variables π_{d_i}, whose value will be known at the time decision d_i is made.
- A policy is a sequence $\delta_1, \ldots, \delta_n$ of decision functions

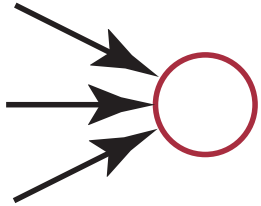
$$\delta_i : dom(\pi_{d_i}) \rightarrow dom(d_i).$$

This policy means that when the agent has observed $O \in dom(\pi_{d_i})$, it will do $\delta_i(O)$.

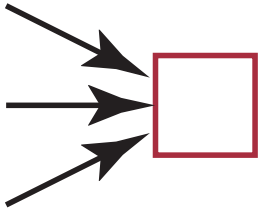
Decision Networks

- A decision network is a graphical representation of a finite sequential decision problem.
- Decision networks extend belief networks to include decision variables and utility.
- A decision network specifies what information is available when the agent has to act.
- A decision network specifies which variables the utility depends on.

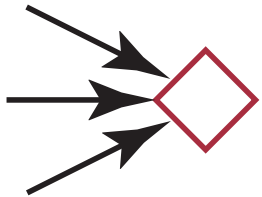
Decisions Networks



- A random variable is drawn as an ellipse. Arcs into the node represent probabilistic dependence.

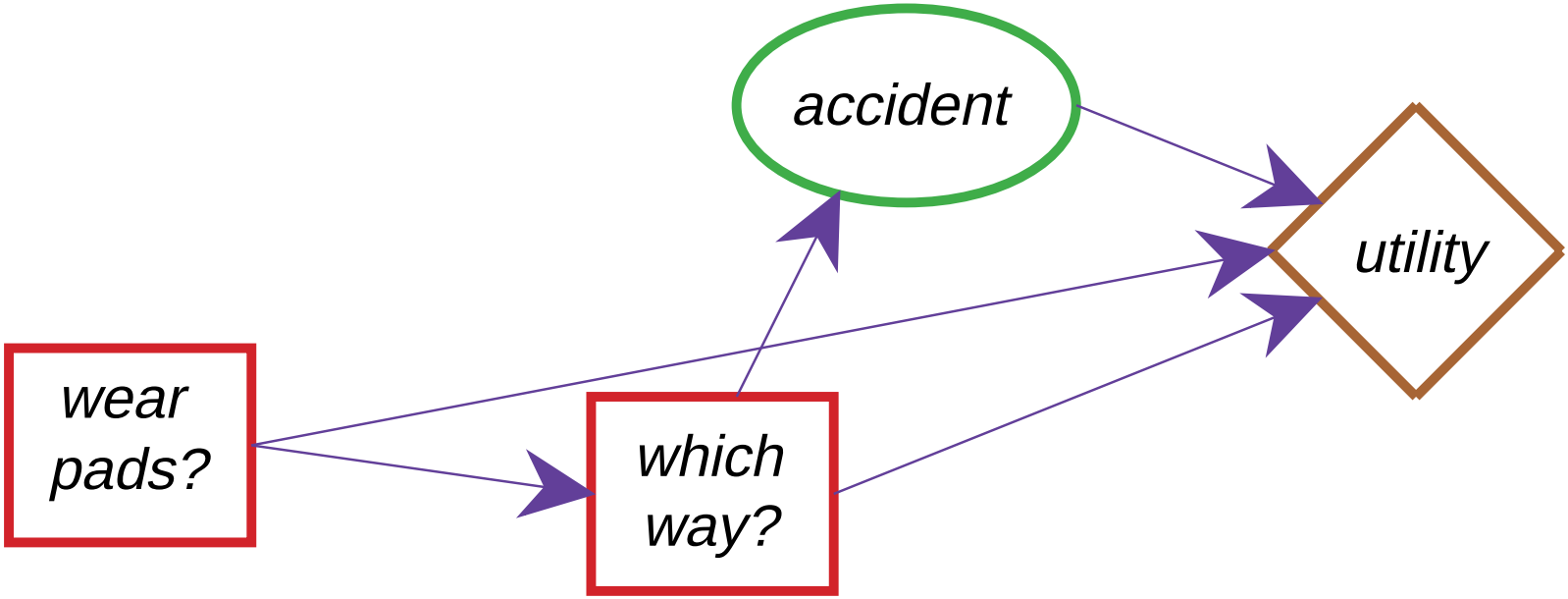


- A decision variable is drawn as an rectangle. Arcs into the node represent information available when the decision is make.



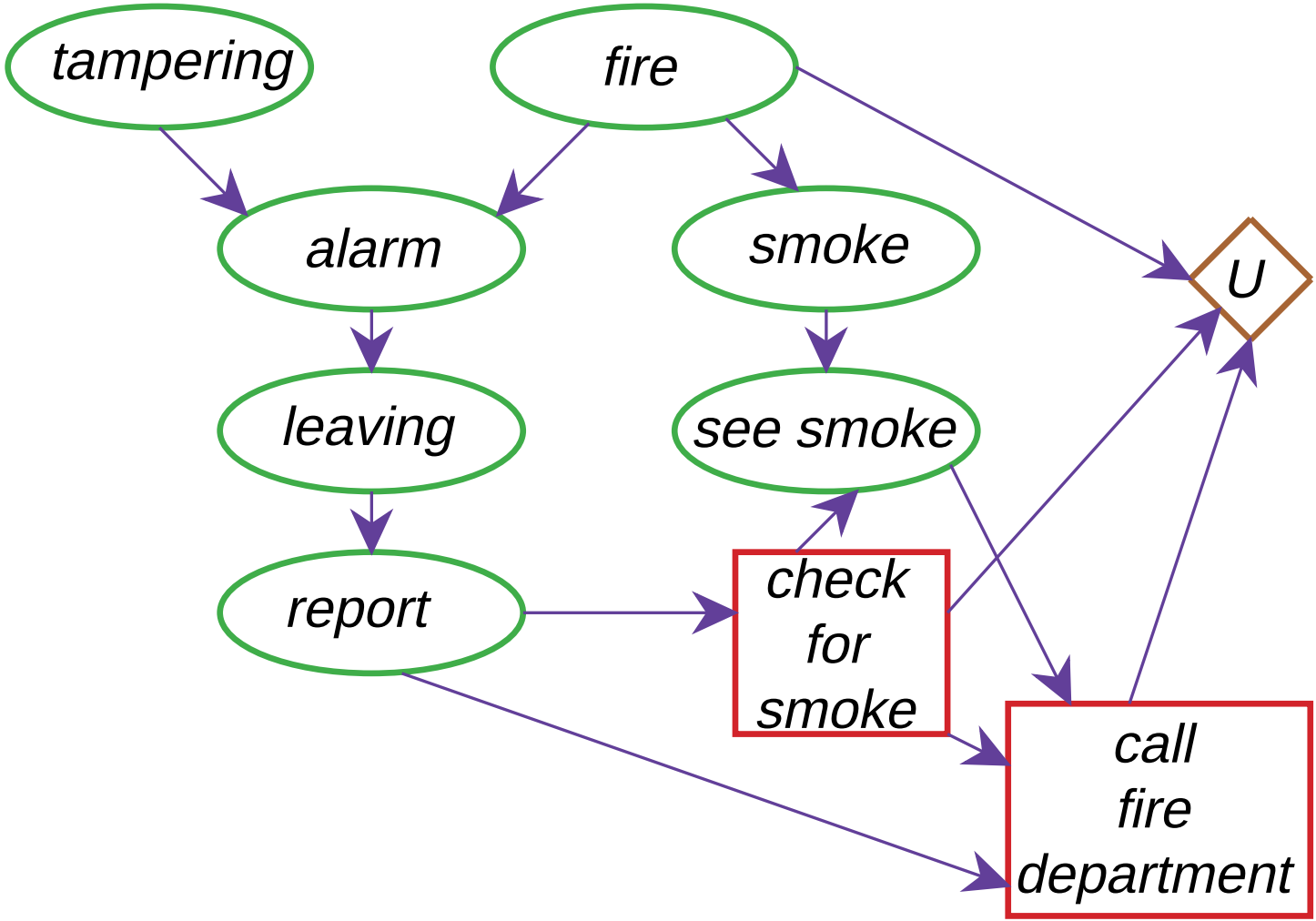
- A value node is drawn as a diamond. Arcs into the node represent values that the value depends on.

Example Decision Network



This shows explicitly which nodes affect whether there is an accident.

Decision Network for the Alarm Problem



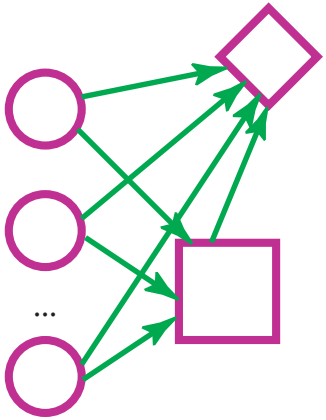
Expected Value of a Policy

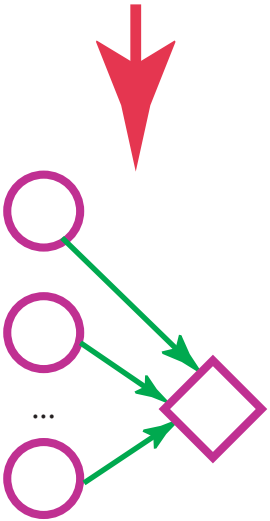
- A policy δ is an assignment of a decision function $\delta_i : dom(\pi_{d_i}) \rightarrow dom(d_i)$ to each decision variable d_i.
- Possible world ω satisfies policy δ, written $\omega \models \delta$ if the world assigns the value to each decision node that the policy specifies.
- The expected utility of policy δ is

$$\mathbf{E}(\delta) = \sum_{\omega \models \delta} \rho(U, \omega) \times \mu(\omega),$$

- An optimal policy is one with the highest expected utility.

Finding the optimal policy

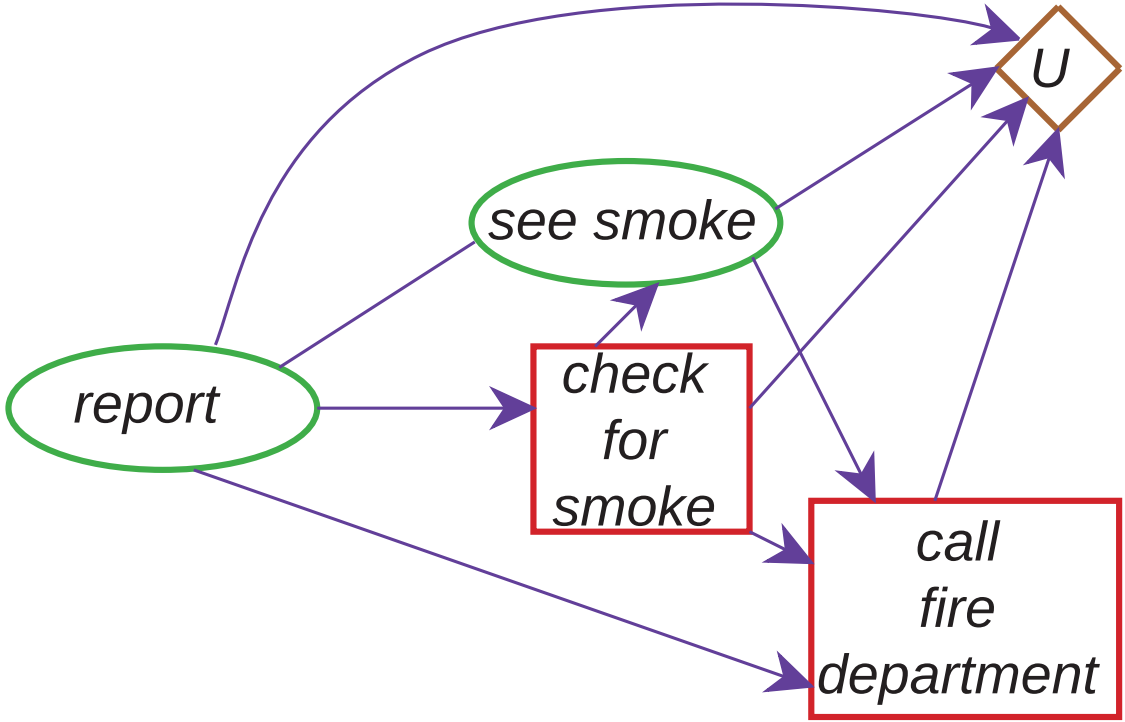




- If value node is only connected to a decision node and (some of) its parents
  - ↪ select a decision to maximize value for each assignment to the parent.
- If it isn't of this form, eliminate the non-observed variables.
- Replace decision node with value node.
- Repeat till there are no more decision nodes.

Reduced Alarm Example

Eliminate the non-observed variables for the final decision.



Complexity of finding the optimal policy

- If there are k binary parents, there are 2^k optimizations.
- If there are b possible actions, there are b^{2^k} policies.
- The dynamic programming algorithm is much more efficient than searching through policy space.

Value of Information

- We can determine the value of information X for a certain decision D is utility of the the network with an arc from X to D minus the utility of the network without the arc.
- The value of information is always non-negative.
- It is positive only if the agent changes its action depending on X.