Chapter 10: Using Uncertain Knowledge

Using Uncertain Knowledge

- Agents don't have complete knowledge about the world.
- Agents need to make decisions based on their uncertainty.
- It isn't enough to assume what the world is like.
  Example: wearing a seat belt.
- An agent needs to reason about its uncertainty.
- When an agent makes an action under uncertainty it is gambling $\Longrightarrow$ probability.

Probability

- Probability is an agent's measure of belief in some proposition — subjective probability.
- Example: Your probability of a bird flying is your measure of belief in the flying ability of an individual based only on the knowledge that the individual is a bird.
  - Other agents may have different probabilities, as they may have had different experiences with birds or different knowledge about this particular bird.
  - An agent's belief in a bird's flying ability is affected by what the agent knows about that bird.

Numerical Measures of Belief

- Belief in proposition, f, can be measured in terms of a number between 0 and 1 — this is the probability of f.
  - The probability f is 0 means that f is believed to be definitely false.
  - The probability f is 1 means that f is believed to be definitely true.
- Using 0 and 1 is purely a convention.
- f has a probability between 0 and 1, doesn't mean f is true to some degree, but means you are ignorant of its truth value. Probability is a measure of your ignorance.

Random Variables

- A random variable is a term in a language that can take one of a number of different values.
- The domain of a variable X, written $dom(X)$, is the set of values X can take.
- A tuple of random variables $\langle X_1, \ldots, X_n \rangle$ is a complex random variable with domain $dom(X_1) \times \cdots \times dom(X_n)$. Often the tuple is written as $X_1, \ldots, X_n$.
- Assignment $X = x$ means variable X has value x.
- A proposition is a Boolean formula made from assignments of values to variables.

Possible World Semantics

- A **possible world** specifies an assignment of one value to each random variable.

- $w \models X = x$

  means variable X is assigned value x in world w.

- Logical connectives have their standard meaning:

$$w \models \alpha \wedge \beta \text{ if } w \models \alpha \text{ and } w \models \beta$$

$$w \models \alpha \vee \beta \text{ if } w \models \alpha \text{ or } w \models \beta$$

$$w \models \neg\alpha \text{ if } w \not\models \alpha$$

- Let Ω be the set of all possible worlds.

Semantics of Probability: finite case

For a finite number of possible worlds:

- Define a nonnegative measure $\mu(w)$ to each set of worlds w so that the measures of the possible worlds sum to 1.

  The measure specifies how much you think the world w is like the real world.

- The probability of proposition f is defined by:

$$P(f) = \sum_{w \models f} \mu(\omega).$$

Axioms of Probability: finite case

Four axioms define what follows from a set of probabilities:

Axiom 1 $P(f) = P(g)$ if $f \leftrightarrow g$ is a tautology. That is, logically equivalent formulae have the same probability.

Axiom 2 $0 \leq P(f)$ for any formula f.

Axiom 3 $P(\tau) = 1$ if τ is a tautology.

Axiom 4 $P(f \vee g) = P(f) + P(g)$ if $\neg(f \wedge g)$ is a tautology.

These axioms are sound and complete with respect to the semantics.

Semantics of Probability: general case

In the general case we have a measure on sets of possible worlds, satisfying:

- $\mu(S) \geq 0$ for all $S \subseteq \Omega$
- $\mu(\Omega) = 1$
- $\mu(S_1 \cup S_2) = \mu(S_1) + \mu(S_2)$ if $S_1 \cap S_2 = \{\}$.

  Or sometimes σ-additivity:

$$\mu(\bigcup_i S_i) = \sum_i \mu(S_i) \text{ if } S_i \cap S_j = \{\}$$

Then $P(\alpha) = \mu(\{w | w \models \alpha\})$.

Probability Distributions

- A probability distribution on a random variable X is a function $dom(X) \to [0, 1]$ such that

$$x \mapsto P(X = x).$$

  This is written as $P(X)$.

- This also includes the case where we have tuples of variables. E.g., $P(X, Y, Z)$ means $P(\langle X, Y, Z\rangle)$.

- When $dom(X)$ is infinite sometimes we need a probability density function...

Conditioning

- Probabilistic conditioning specifies how to revise beliefs based on new information.
- You build a probabilistic model taking all background information into account. This gives the **prior probability.**
- All other information must be conditioned on.
- If **evidence** e is the all of the information obtained subsequently, the **conditional probability** $P(h|e)$ of h given e is the **posterior probability** of h.

Semantics of Conditional Probability

Evidence e rules out possible worlds incompatible with e.

Evidence e induces a new measure, μ_e, over possible worlds

$$\mu_e(\omega) = \begin{cases} \frac{1}{P(e)} \times \mu(\omega) & \text{if } \omega \models e \\ 0 & \text{if } \omega \not\models e \end{cases}$$

The conditional probability of formula h given evidence e is

$$\begin{aligned} P(h|e) &= \sum_{\omega \models h} \mu_e(w) \\ &= \frac{P(h \wedge e)}{P(e)} \end{aligned}$$

Properties of Conditional Probabilities

- **Chain rule:**

$$
\begin{aligned}
& P(f_1 \wedge f_2 \wedge \ldots \wedge f_n) \\
&\quad = \; P(f_1) \times P(f_2|f_1) \times P(f_3|f_1 \wedge f_2) \\
&\qquad\quad \times \cdots \times P(f_n|f_1 \wedge \cdots \wedge f_{n-1}) \\
&\quad = \; \prod_{i=1}^{n} P(f_i|f_1 \wedge \cdots \wedge f_{i-1})
\end{aligned}
$$

Bayes' theorem

The chain rule and commutativity of conjunction ($h \wedge e$ is equivalent to $e \wedge h$) gives us:

$$\begin{aligned} P(h \wedge e) &= P(h|e) \times P(e) \\ &= P(e|h) \times P(h). \end{aligned}$$

If $P(e) \neq 0$, you can divide the right hand sides by $P(e)$:

$$P(h|e) = \frac{P(e|h) \times P(h)}{P(e)}.$$

This is Bayes' theorem.

Why is Bayes' theorem interesting?

- Often you have causal knowledge:

  $P(symptom \mid disease)$

  $P(light\ is\ off \mid status\ of\ switches\ and\ switch\ positions)$

  $P(alarm \mid fire)$

  $P($*image looks like*  $\mid a\ tree\ is\ in\ front\ of\ a\ car)$

- and want to do evidential reasoning:

  $P(disease \mid symptom)$

  $P(status\ of\ switches \mid light\ is\ off\ and\ switch\ positions)$

  $P(fire \mid alarm)$.

  $P(a\ tree\ is\ in\ front\ of\ a\ car \mid$ *image looks like* $)$

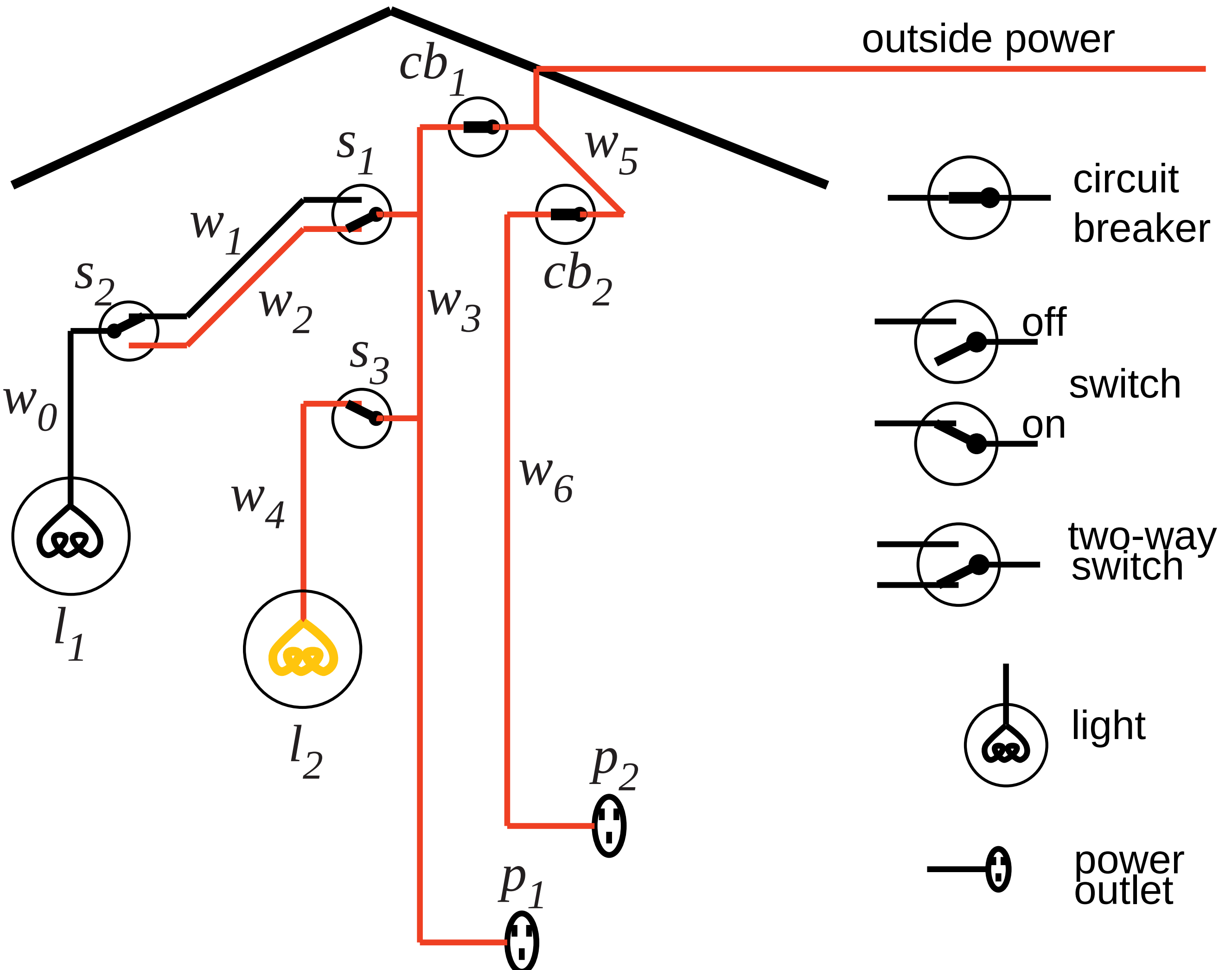
Conditional independence

Random variable X is **independent** of random variable Y **given** random variable Z if, for all $x_i \in dom(X)$, $y_j \in dom(Y)$, $y_k \in dom(Y)$ and $z_m \in dom(Z)$,

$$\begin{aligned} & P(X = x_i | Y = y_j \wedge Z = z_m) \\ = \quad & P(X = x_i | Y = y_k \wedge Z = z_m) \\ = \quad & P(X = x_i | Z = z_m). \end{aligned}$$

That is, knowledge of Y's value doesn't affect your belief in the value of X, given a value of Z.

Example domain (diagnostic assistant)

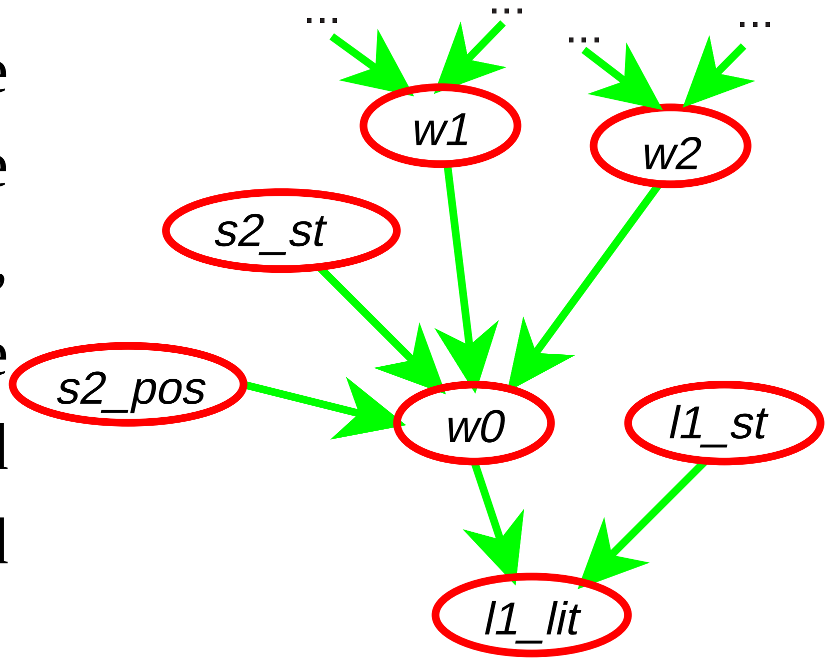


Examples of conditional independence

- The identity of the queen of Canada is independent of whether light $l1$ is lit given whether there is outside power.
- Whether there is someone in a room is independent of whether a light $l2$ is lit given the position of switch $s3$.
- Whether light $l1$ is lit is independent of the position of light switch $s2$ given whether there is power in wire w_0.
- Every other variable may be independent of whether light $l1$ is lit given whether there is power in wire w_0 and the status of light $l1$ (if it's *ok*, or if not, how it's broken).

Idea of belief networks

Whether $l1$ is lit ($l1\_lit$) depends only on the status of the light ($l1\_st$) and whether there is power in wire $w0$. Thus, $l1\_lit$ is independent of the other variables given $l1\_st$ and $w0$. In a belief network, $w0$ and $l1\_st$ are parents of $l1\_lit$.



Similarly, $w0$ depends only on whether there is power in $w1$, whether there is power in $w2$, the position of switch $s2$ ($s2\_pos$), and the status of switch $s2$ ($s2\_st$).

Belief networks

- Totally order the variables of interest: $X_1, \ldots, X_n$
- Theorem of probability theory (chain rule): $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | X_1, \ldots, X_{i-1})$
- The parents π_{X_i} of X_i are those predecessors of X_i that render X_i independent of the other predecessors. That is, $\pi_{X_i} \subseteq X_1, \ldots, X_{i-1}$ and $P(X_i | \pi_{X_i}) = P(X_i | X_1, \ldots, X_{i-1})$
- So $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | \pi_{X_i})$
- A belief network is a graph: the nodes are random variables; there is an arc from the parents of each node into that node.

Belief network summary

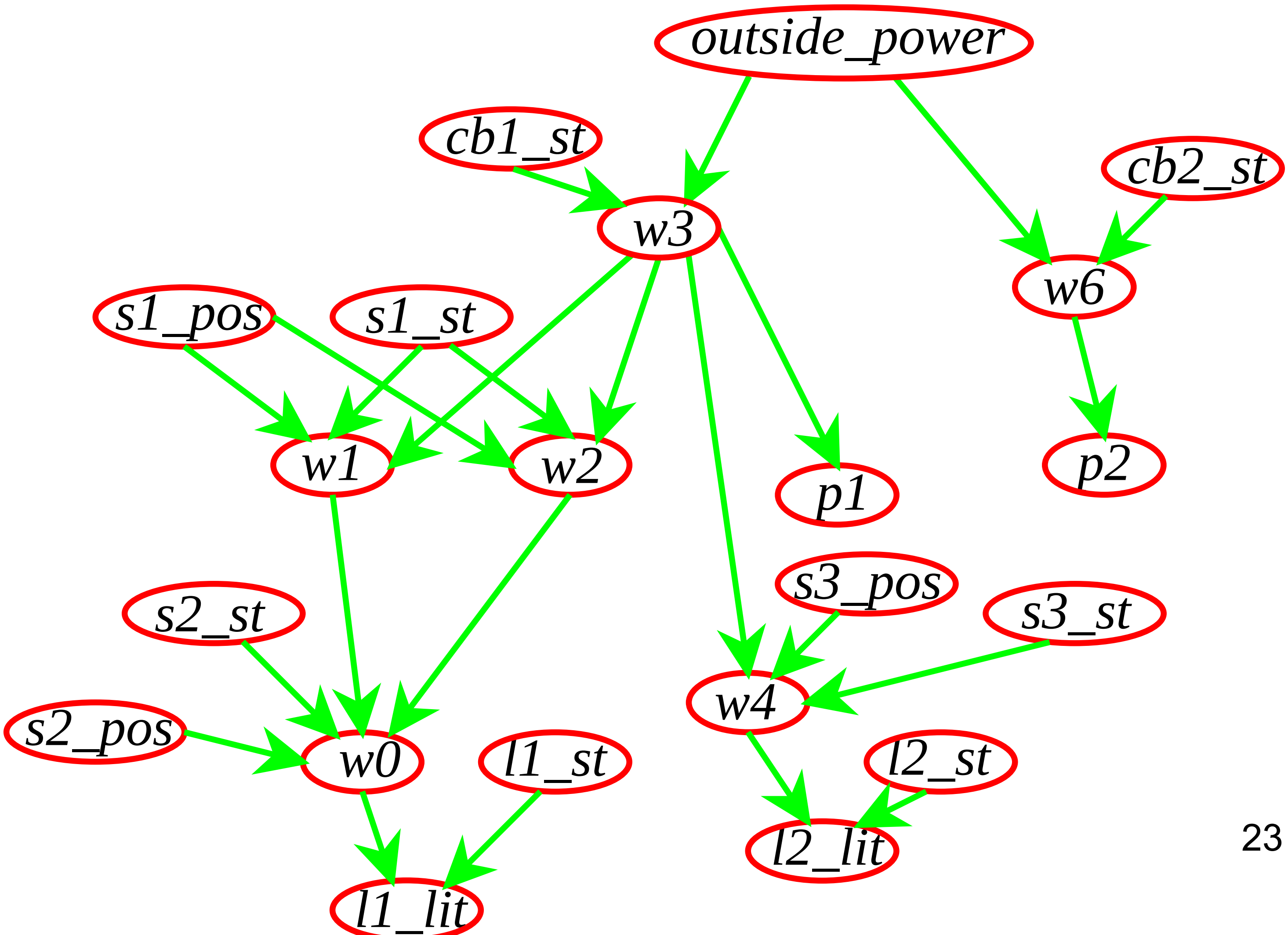
- A belief network is automatically acyclic by construction.
- A belief network is a directed acyclic graph (DAG) where nodes are random variables.
- The parents of a node n are those variables on which n directly depends.
- A belief network is a graphical representation of dependence and independence:
  - A variable is independent of its nondescendants given its parents.

Components of a belief network

A belief network consists of:

- a directed acyclic graph with nodes labeled with random variables
- a domain for each random variable
- a set of conditional probability tables for each variable given its parents (including prior probabilities for nodes with no parents).

Example belief network



Example belief network (continued)

The belief network also specifies:

- The domain of the variables:

  $w_0, \ldots, w_6$ have domain $\{live, dead\}$

  s_1_pos, s_2_pos, and s_3_pos have domain $\{up, down\}$

  s_1_st has $\{ok, upside\_down, short, intermittent, broken\}$.

- Conditional probabilities, including:

  $P(w_1 = live | s_1\_pos = up \wedge s_1\_st = ok \wedge w_3 = live)$

  $P(w_1 = live | s_1\_pos = up \wedge s_1\_st = ok \wedge w_3 = dead)$

  $P(s_1\_pos = up)$

  $P(s_1\_st = upside\_down)$

Constructing belief networks

To represent a domain in a belief network, you need to consider:

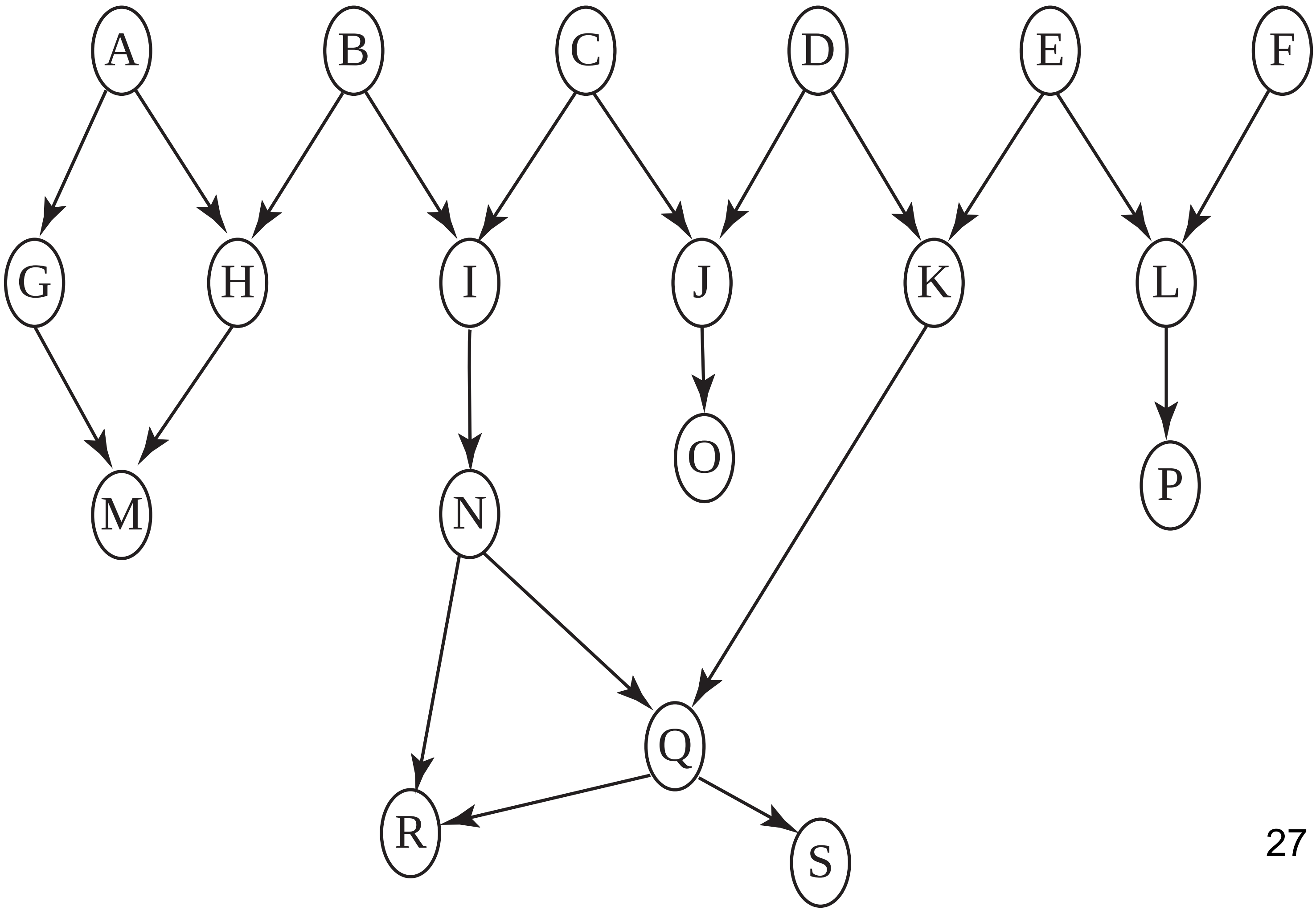
- What are the relevant variables?
- What values should these variables take?
- What is the relationship between them? This should be expressed in terms of local influence.
- How does the value of one variable depend on the variables that locally influence it (its parents)? This is expressed in terms of the conditional probability tables.

Using belief networks

The power network can be used in a number of ways:

- Conditioning on the status of the switches and circuit breakers, whether there is outside power and the position of the switches, you can simulate the lighting.
- Given values for the switches, the outside power, and whether the lights are lit, you can determine the posterior probability that each switch or circuit breaker is *ok* or not.
- Given some switch positions and some outputs and some intermediate values, you can determine the probability of any other variable in the network.

Understanding independence: example



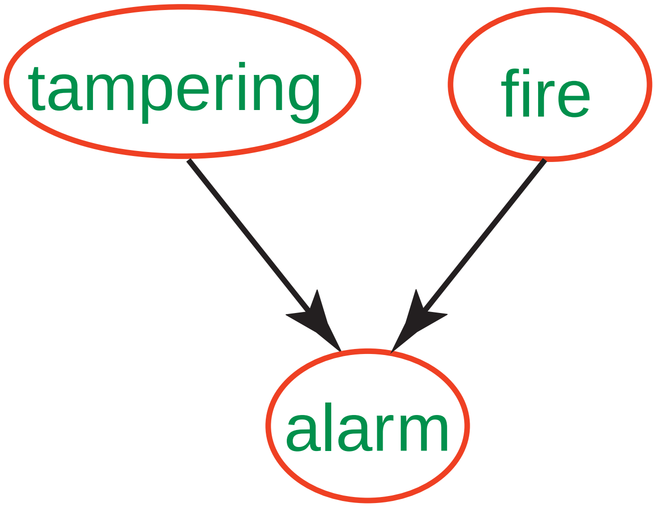
Understanding independence: questions

- On which given probabilities does $P(N)$ depend?
- If you were to observe a value for B, which variables' probabilities will change?
- If you were to observe a value for N, which variables' probabilities will change?
- Suppose you had observed a value for M; if you were to then observe a value for N, which variables' probabilities will change?
- Suppose you had observed B and Q; which variables' probabilities will change when you observe N?

What variables are affected by observing?

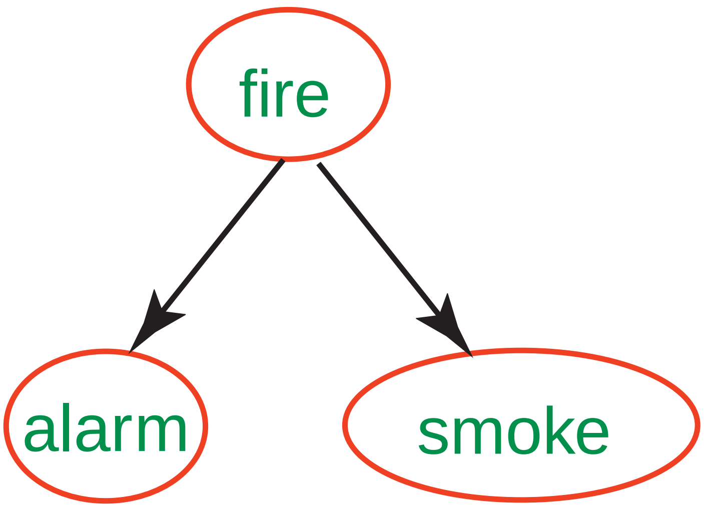
- If you observe variable $\overline{Y}$, the variables whose posterior probability is different from their prior are:
  - The ancestors of $\overline{Y}$ and
  - their descendants.
- Intuitively (if you have a causal belief network):
  - You do abduction to possible causes and
  - prediction from the causes.

Common descendants



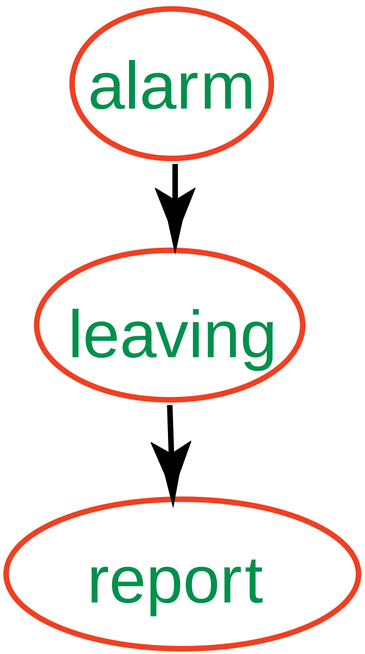
- *tampering* and *fire* are independent
- *tampering* and *fire* are dependent given *alarm*
- Intuitively, *tampering* can explain away *fire*

Common ancestors



- *alarm* and *smoke* are dependent
- *alarm* and *smoke* are independent given *fire*
- Intuitively, *fire* can explain *fire* and *smoke*; learning one can affect the other by changing your belief in *fire*.

Chain



- *alarm* and *report* are dependent
- *alarm* and *report* are independent given *leaving*
- Intuitively, the only way that the *alarm* affects *report* is by affecting *leaving*.

d-separation

- $\overline{X}$ is d-separated from $\overline{Y}$ given $\overline{Z}$ if there is no path from an element of $\overline{X}$ to an element of $\overline{Y}$, where:
  - If there are paths $A \rightarrow B$ and $B \rightarrow C$ such that $B \notin \overline{Z}$, there is a path $A \rightarrow C$.
  - If there are paths $B \rightarrow A$ and $B \rightarrow C$ such that $B \notin \overline{Z}$, there is a path $A \rightarrow C$.
  - If there are paths $A \rightarrow B$ and $C \rightarrow B$ such that $B \in \overline{Z}$, there is a path $A \rightarrow C$.
- $\overline{X}$ is independent $\overline{Y}$ given $\overline{Z}$ for some conditional probabilities iff $\overline{X}$ is d-separated from $\overline{Y}$ given $\overline{Z}$

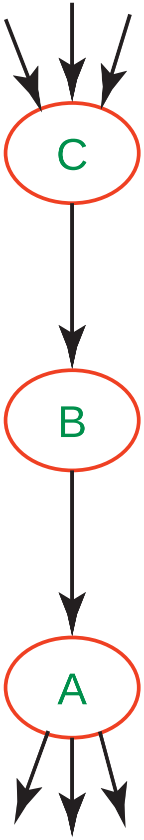
Belief network inference

Three main approaches to determine posterior distributions in belief networks:

- Exploiting the structure of the network to eliminate (sum out) the non-observed, non-query variables one at a time.
- Search-based approaches that enumerate some of the possible worlds, and estimate posterior probabilities from the worlds generated.
- Stochastic simulation where random cases are generated according to the probability distributions.

Summing out a variable: intuition

Suppose B is Boolean ($B = true$ is b and $B = false$ is $\neg b$)



$$
\begin{aligned}
& P(C|A) \\
= \quad & P(C \wedge b|A) + P(C \wedge \neg b|A) \\
= \quad & P(C|b \wedge A)P(b|A) + P(C|\neg b \wedge A)P(\neg b|A) \\
= \quad & P(C|b)P(b|A) + P(C|\neg b)P(\neg b|A) \\
= \quad & \textstyle\sum_B P(C|B)P(B|A)
\end{aligned}
$$

We can compute the probability of some of the variables by summing out the other variables.

Factors

A **factor** is a representation of a function from a tuple of random variables into a number.

We will write factor f on variables $X_1, \ldots, X_j$ as $f(X_1, \ldots, X_j)$.

We can assign some or all of the variables of a factor:

- $f(X_1 = v_1, X_2, \ldots, X_j)$, where $v_1 \in dom(X_1)$, is a factor on $X_2, \ldots, X_j$.
- $f(X_1 = v_1, X_2 = v_2, \ldots, X_j = v_j)$ is a number that is the value of f when each X_i has value v_i.

The former is also written as $f(X_1, X_2, \ldots, X_j)_{X_1 = v_1}$, etc.

Example factors

$r(X, Y, Z)$:

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

$r(X{=}t, Y, Z)$:

Y	Z	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

$r(X{=}t, Y, Z{=}f)$:

Y	val
t	0.9
f	0.8

$$r(X{=}t, Y{=}f, Z{=}f) = 0.8$$

Multiplying factors

The **product** of factor $f_1(\overline{X}, \overline{Y})$ and $f_2(\overline{Y}, \overline{Z})$, where $\overline{Y}$ are the variables in common, is the factor $(f_1 \times f_2)(\overline{X}, \overline{Y}, \overline{Z})$ defined by:

$$(f_1 \times f_2)(\overline{X}, \overline{Y}, \overline{Z}) \quad = \quad f_1(\overline{X}, \overline{Y}) f_2(\overline{Y}, \overline{Z}).$$

Multiplying factors example

f_1:

A	B	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

f_2:

B	C	val
t	t	0.3
t	f	0.7
f	t	0.6
f	f	0.4

$f_1 \times f_2$:

A	B	C	val
t	t	t	0.03
t	t	f	0.07
t	f	t	0.54
t	f	f	0.36
f	t	t	0.06
f	t	f	0.14
f	f	t	0.48
f	f	f	0.32

Summing out variables

We can sum out a variable, say X_1 with domain $\{v_1, \dots, v_k\}$, from factor $f(X_1, \dots, X_j)$, resulting in a factor on $X_2, \dots, X_j$ defined by:

$$
\begin{aligned}
&(\sum_{X_1} f)(X_2, \dots, X_j) \\
&\quad = \; f(X_1 = v_1, \dots, X_j) + \cdots + f(X_1 = v_k, \dots, X_j)
\end{aligned}
$$

Multiplying factors example

f_3:

A	B	C	val
t	t	t	0.03
t	t	f	0.07
t	f	t	0.54
t	f	f	0.36
f	t	t	0.06
f	t	f	0.14
f	f	t	0.48
f	f	f	0.32

$\sum_B f_3$:

A	C	val
t	t	0.57
t	f	0.43
f	t	0.54
f	f	0.46

Evidence

If we want to compute the posterior probability of Z given evidence $Y_1 = v_1 \wedge \ldots \wedge Y_j = v_j$:

$$\begin{aligned}
& P(Z|Y_1 = v_1, \ldots, Y_j = v_j) \\
&= \frac{P(Z, Y_1 = v_1, \ldots, Y_j = v_j)}{P(Y_1 = v_1, \ldots, Y_j = v_j)} \\
&= \frac{P(Z, Y_1 = v_1, \ldots, Y_j = v_j)}{\sum_Z P(Z, Y_1 = v_1, \ldots, Y_j = v_j)}.
\end{aligned}$$

So the computation reduces to the probability of $P(Z, Y_1 = v_1, \ldots, Y_j = v_j)$.

We normalize at the end.

Probability of a conjunction

Suppose the variables of the belief network are $X_1, \ldots, X_n$.

To compute $P(Z, Y_1 = v_1, \ldots, Y_j = v_j)$, we sum out the other variables, $Z_1, \ldots, Z_k = \{X_1, \ldots, X_n\} - \{Z\} - \{Y_1, \ldots, Y_j\}$.

We order the Z_i into an elimination ordering.

$$
\begin{aligned}
& P(Z, Y_1 = v_1, \ldots, Y_j = v_j) \\
= \; & \sum_{Z_k} \cdots \sum_{Z_1} P(X_1, \ldots, X_n)_{Y_1 = v_1, \ldots, Y_j = v_j}. \\
= \; & \sum_{Z_k} \cdots \sum_{Z_1} \prod_{i=1}^{n} P(X_i | \pi_{X_i})_{Y_1 = v_1, \ldots, Y_j = v_j}.
\end{aligned}
$$

Computing sums of products

Computation in belief networks reduces to computing the sums of products.

- How can we compute $ab + ac$ efficiently?

Computing sums of products

Computation in belief networks reduces to computing the sums of products.

- How can we compute $ab + ac$ efficiently?
- Distribute out the a giving $a(b + c)$

Computing sums of products

Computation in belief networks reduces to computing the sums of products.

- How can we compute $ab + ac$ efficiently?
- Distribute out the a giving $a(b + c)$
- How can we compute $\sum_{Z_1} \prod_{i=1}^{n} P(X_i | \pi_{X_i})$ efficiently?

Computing sums of products

Computation in belief networks reduces to computing the sums of products.

- How can we compute $ab + ac$ efficiently?
- Distribute out the a giving $a(b + c)$
- How can we compute $\sum_{Z_1} \prod_{i=1}^{n} P(X_i | \pi_{X_i})$ efficiently?
- Distribute out those factors that don't involve Z_1.

Variable elimination algorithm

To compute $P(Z|Y_1 = v_1 \wedge \ldots \wedge Y_j = v_j)$:

- Construct a factor for each conditional probability.
- Set the observed variables to their observed values.
- Sum out each of the other variables (the $\{Z_1, \ldots, Z_k\}$) according to some elimination ordering.
- Multiply the remaining factors. Normalize by dividing the resulting factor $f(Z)$ by $\sum_Z f(Z)$.

Summing out a variable

To sum out a variable Z_j from a product $f_1, \ldots, f_k$ of factors:

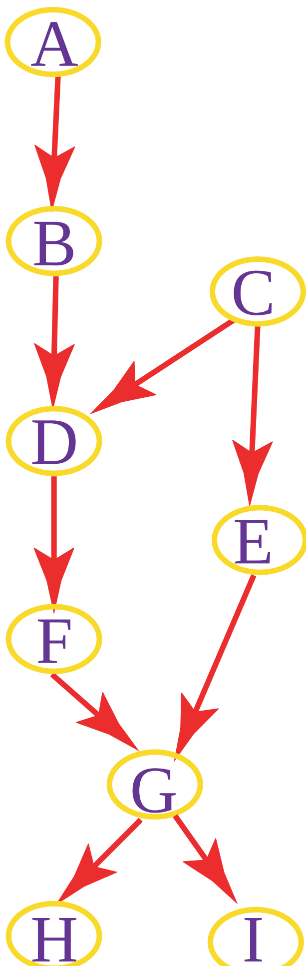
- Partition the factors into
  - those that don't contain Z_j, say $f_1, \ldots, f_i$,
  - those that contain Z_j, say $f_{i+1}, \ldots, f_k$

We know:

$$\sum_{Z_j} f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_i \times \left(\sum_{Z_j} f_{i+1} \times \cdots \times f_k \right).$$

- Explicitly construct a representation of the rightmost factor. Replace the factors $f_{i+1}, \ldots, f_k$ by the new factor.

Variable elimination example



$$\left.\begin{array}{l} P(A) \\ P(B|A) \end{array}\right\} \xrightarrow{\text{elim } A} f_1(B)$$

$$\left.\begin{array}{l} P(C) \\ P(D|BC) \\ P(E|C) \end{array}\right\} \xrightarrow{\text{elim } C} f_2(BDE)$$

$$P(F|D)$$

$$P(G|FE)$$

$$P(H|G) \;\Big\}\; \xrightarrow{\text{obs } H} f_3(G)$$

$$P(I|G) \;\Big\}\; \xrightarrow{\text{elim } I} f_4(G)$$

Markov chain

- A Markov chain is a special sort of belief network:

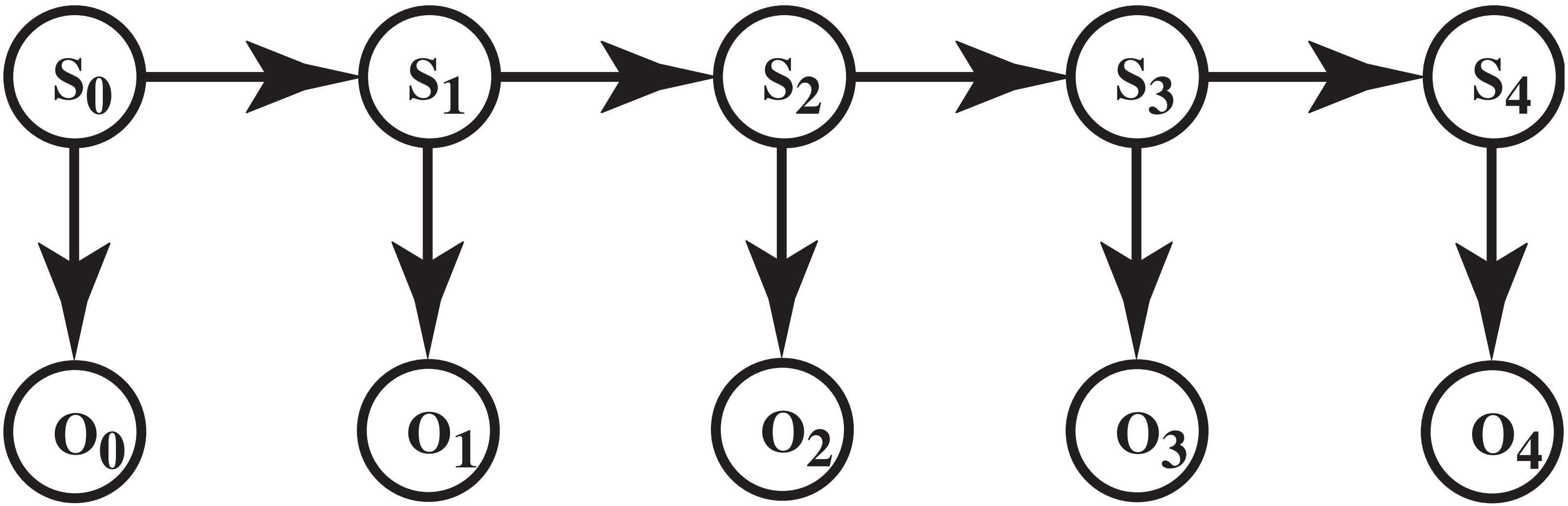
- Thus $P(S_{t+1}|S_0, \ldots, S_t) = P(S_{t+1}|S_t)$.
- Often S_t represented the state at time t. Intuitively S_t conveys all of the information about the history that can affect the future states.
- "The past is independent of the future given the present."

Stationary Markov chain

- A stationary Markov chain is when for all $t > 0$, $u > 0$, $P(S_{t+1}|S_t) = P(S_{u+1}|S_u)$ we have .
- We specify $P(S_0)$ and $P(S_{t+1}|S_t)$.
- It is of interest because:
  - Simple model, easy to specify
  - Natural
  - The network can extend indefinitely

Hidden Markov Model

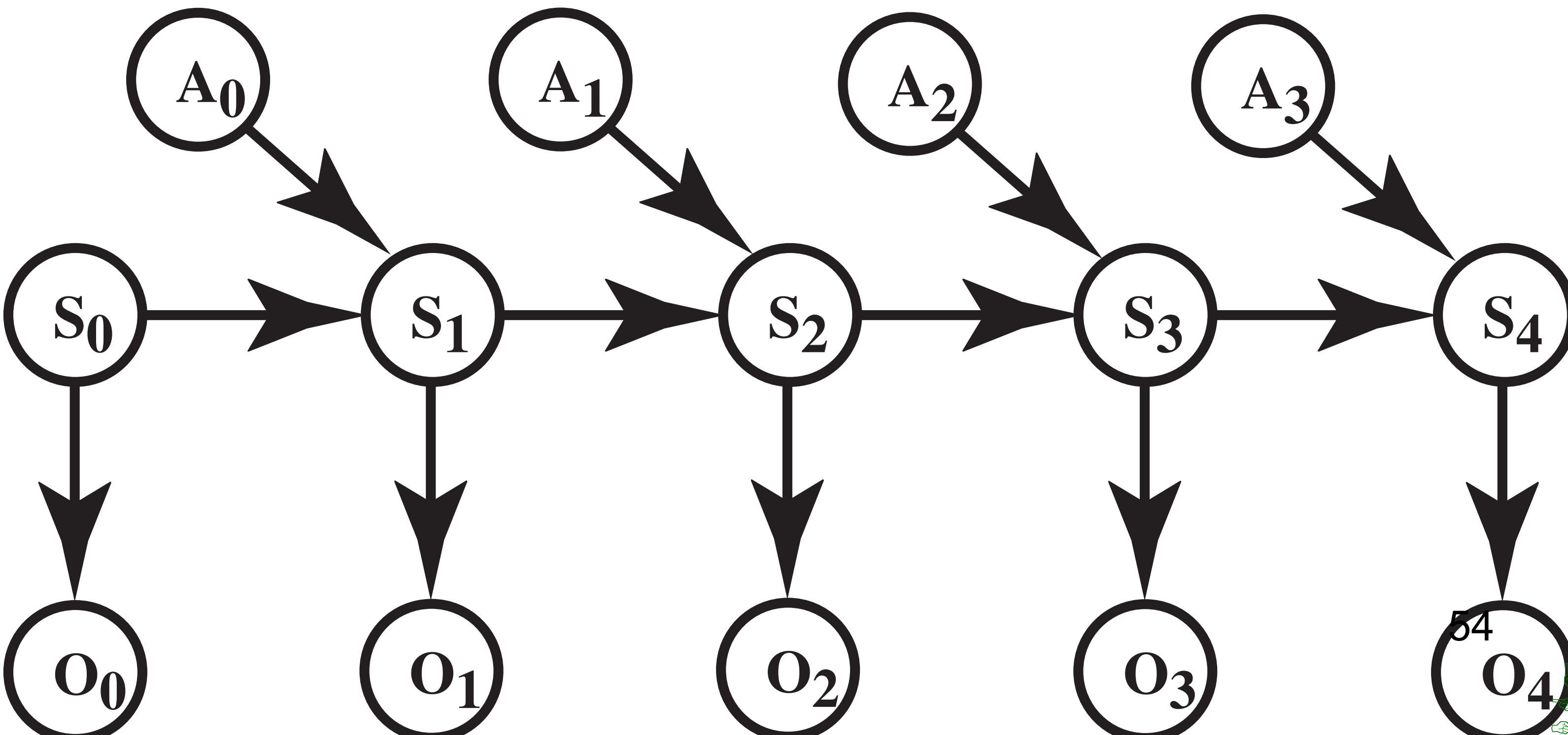
- A Hidden Markov Model (HMM) is a belief network:



- $P(S_0)$ specifies initial conditions
- $P(S_{t+1}|S_t)$ specifies the dynamics
- $P(O_t|S_t)$ specifies the sensor model

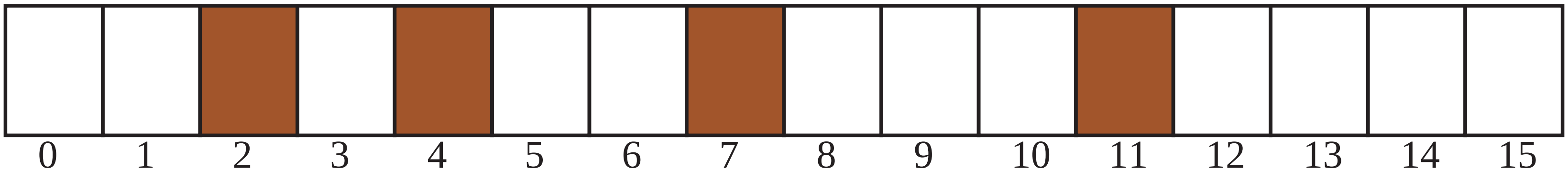
Example: localization

- Suppose a robot wants to determine its location based on its actions and its sensor readings. Called Localization
- This can be represented by the augmented HMM:



Example localization domain

- Circular corridor, with 16 locations:



- Doors at positions: 2, 4, 7, 11.
- Noisy Sensors
- Stochastic Dynamics
- Robot starts at an unknown location and must determine where it is.

Example Sensor Model

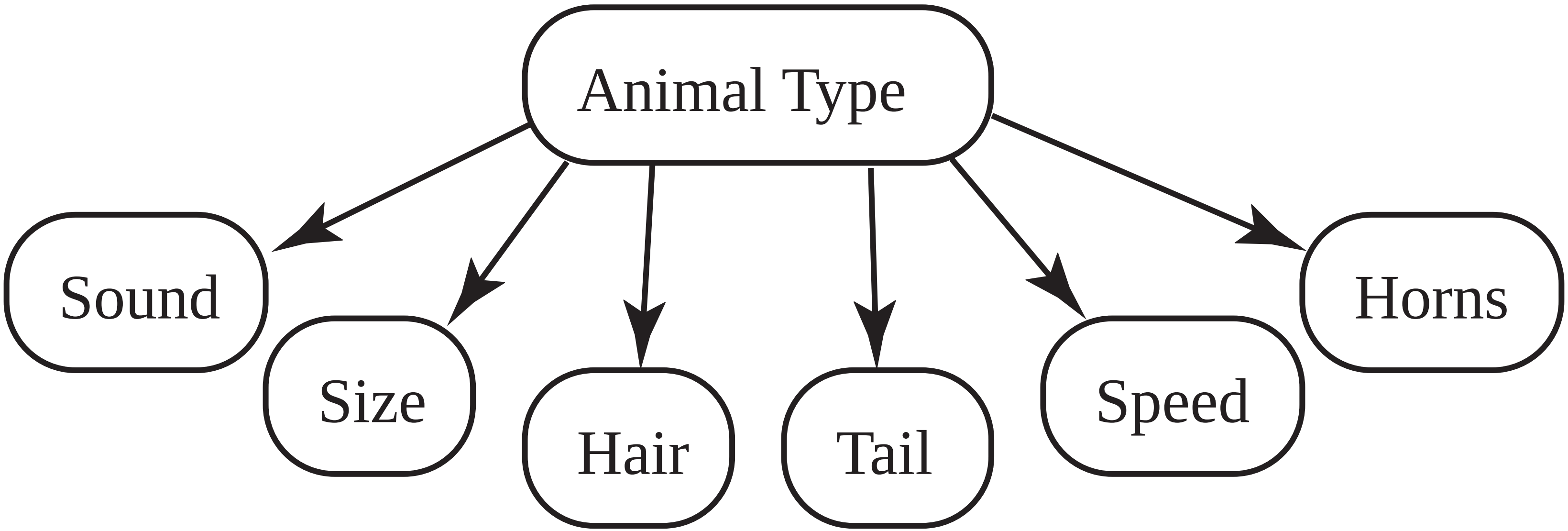
- $P(Observe\ Door \mid At\ Door) = 0.8$
- $P(Observe\ Door \mid Not\ At\ Door) = 0.1$

Example Dynamics Model

- $P(loc_{t+1} = L | action_t = goRight \wedge loc_t = L) = 0.1$
- $P(loc_{t+1} = L + 1 | action_t = goRight \wedge loc_t = L) = 0.8$
- $P(loc_{t+1} = L + 2 | action_t = goRight \wedge loc_t = L) = 0.074$
- $P(loc_{t+1} = L' | action_t = goRight \wedge loc_t = L) = 0.002$ for any other location L'.
  - All location arithmetic is modulo 16.
  - The action *goLeft* works the same but to the left.

Sensor Fusion

- We can have many (noisy) sensors for a property.
- Example:



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

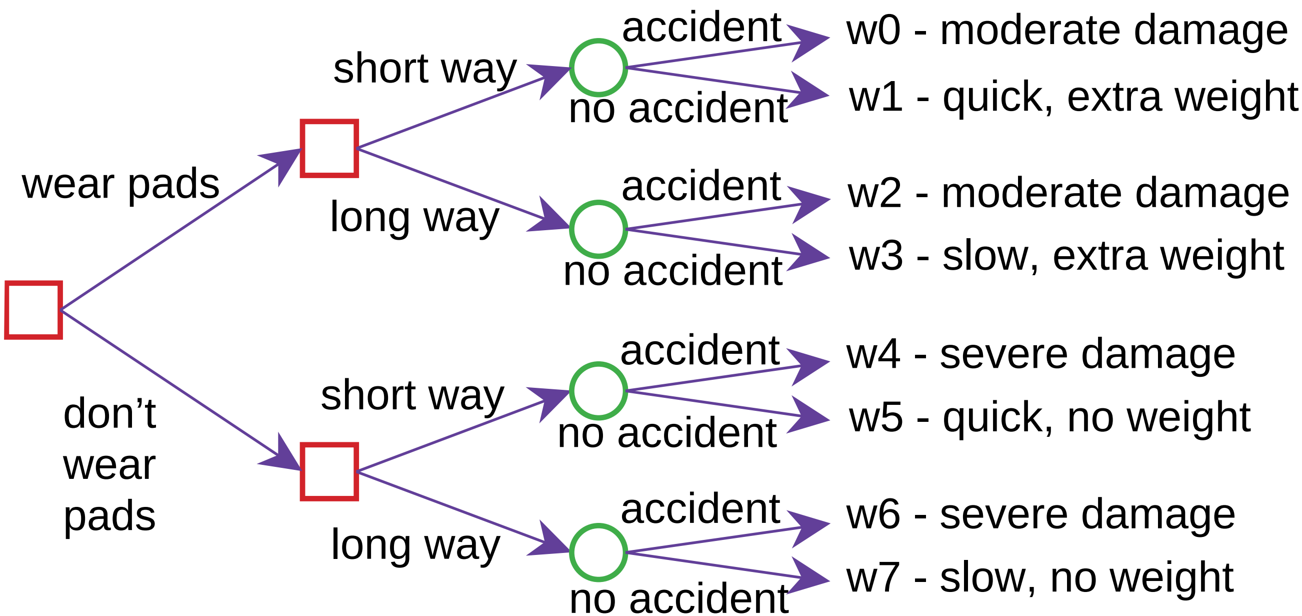
- **Decision variables** are like random variables that an agent gets to choose the value of.
- A possible world specifies the value for each decision variable and each random variable.
- For each assignment of values to all decision variables, the measures of the worlds satisfying that assignment sum 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

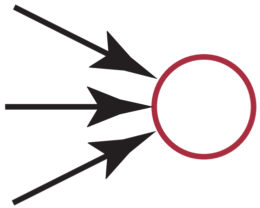
$$\delta_i : dom(\pi_{d_i}) \to dom(d_i).$$

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

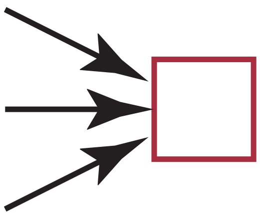
Decision Networks

- 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.

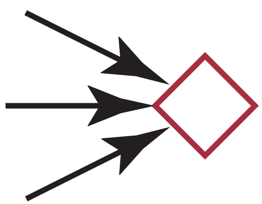
Decisions Networks



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

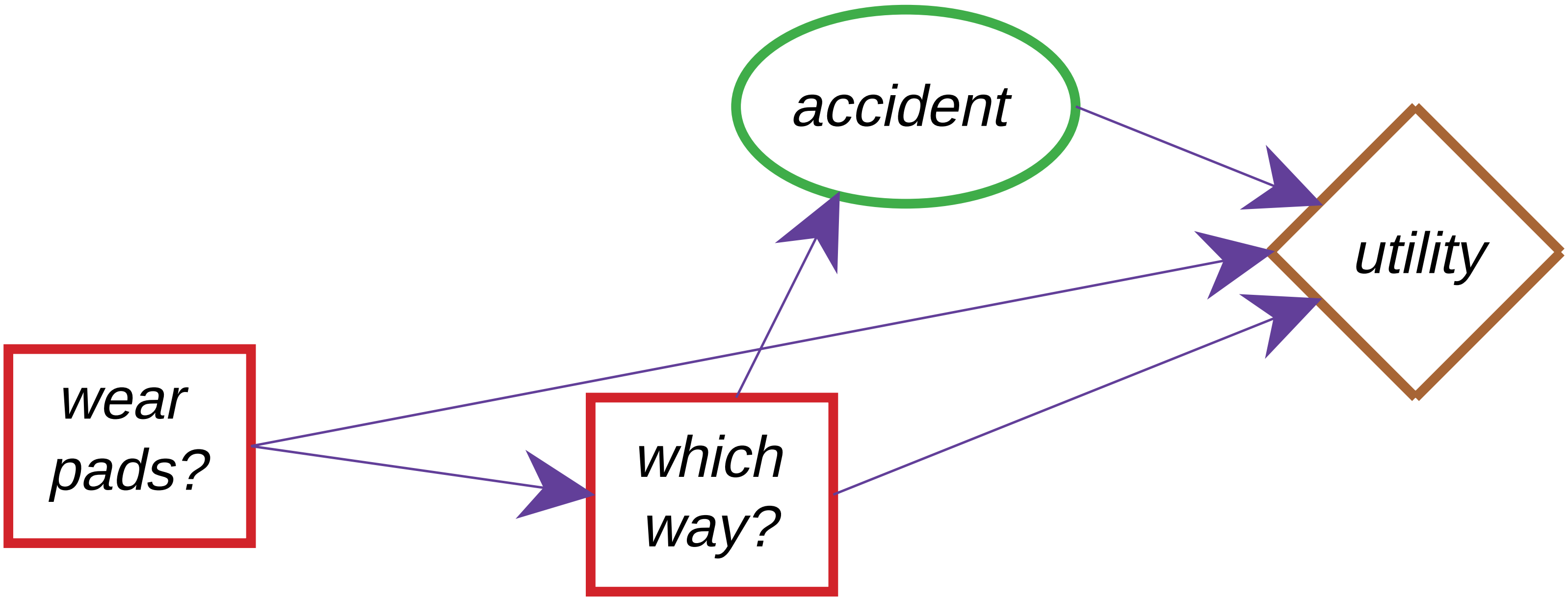


- A decision variable is drawn as an rectangle. Arcs into the node represent information available when the decision is make.



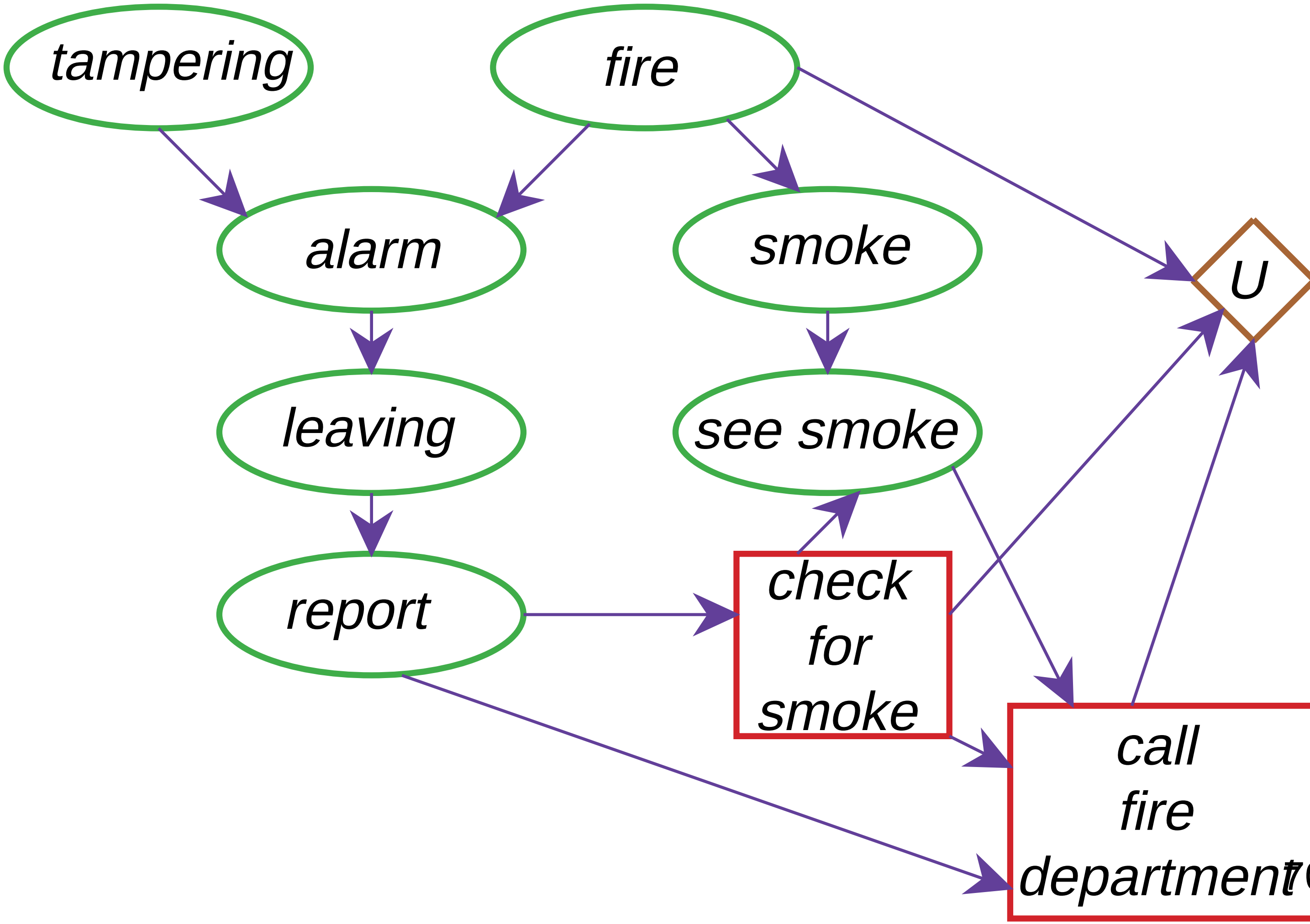
- A value node is drawn as a diamond. Arcs into the node represent values that the value depends on.

Example Decision Network



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

Decision Network for the Alarm Problem



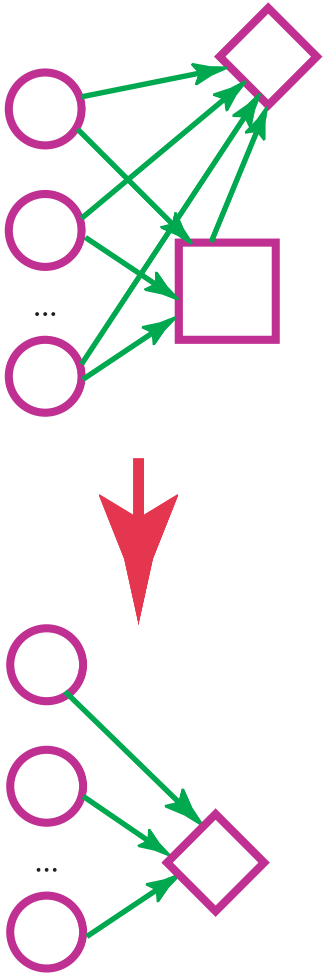
Expected Value of a Policy

- A policy δ is an assignment of a decision function $\delta_i : dom(\pi_{d_i}) \to dom(d_i)$ 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.

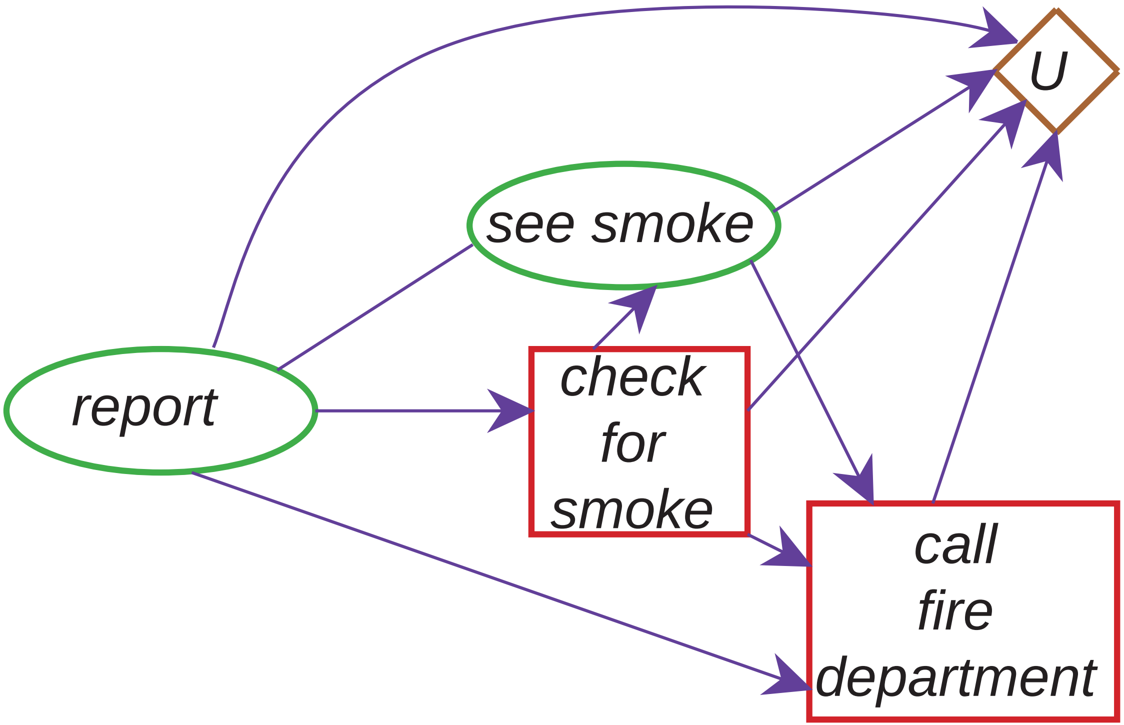
Finding the optimal policy



- 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.