Logic, Knowledge Representation and Bayesian Decision Theory

David Poole

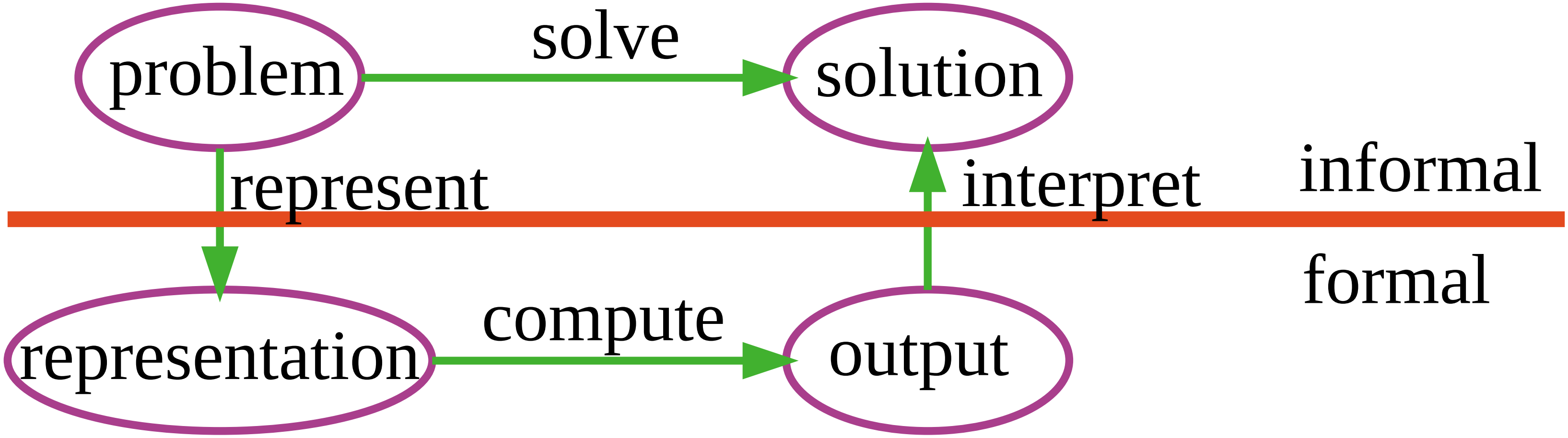
University of British Columbia

☞

Overview

- Knowledge representation, logic, decision theory.
- Belief networks
- Independent Choice Logic
- Stochastic Dynamic Systems
- Bayesian Learning

Knowledge Representation



- Find compact / natural representations
- Exploit features of representation for computational gain.
- Tradeoff representational adequacy, efficient (approximate) inference and learnability

What do we want in a representation?

We want a representation to be

- rich enough to express the knowledge needed to solve the problem.
- as close to the problem as possible: natural and maintainable.
- amenable to efficient computation;
  able to express features of the problem we can exploit for computational gain.
- learnable from data and past experiences.
- trade off accuracy and computation time

Normative Traditions

- Logic
  - Semantics (symbols have meaning)
  - Sound and complete proof procedures
  - Quantification over variables (relations amongst multiple individuals)
- Decision Theory
  - Tradeoffs under uncertainty
  - Probabilities and utilities

Bayesians

- Interested in action: what should an agent do?
- Role of belief is to make good decisions.
- Theorems (Von Neumann and Morgenstern):
  (under reasonable assumptions) a rational agent will act as though it has (point) probabilities and utilities and acts to maximize expected utilities.
- Probability as a measure of belief:
  study of how knowledge affects belief
  lets us combine background knowledge and data

Representations of uncertainty

We want a representation for

- probabilities
- utilities
- actions

that facilitates finding the action(s) that maximise expected utility.

Overview

- Knowledge representation, logic, decision theory.
- Belief networks
  - Independence
  - Inference
  - Causality
- Independent Choice Logic
- Stochastic Dynamic Systems
- Bayesian Learning

Belief networks (Bayesian networks)

- Totally order the variables of interest: $X_1, \ldots, X_n$
- Theorem of probability theory (chain rule):

$$\begin{aligned} P(X_1, \ldots, X_n) &= P(X_1)P(X_2|X_1) \cdots P(X_n|X_1, \ldots, X_{n-1}) \\ &= \prod_{i=1}^{n} P(X_i|X_1, \ldots, X_{i-1}) \end{aligned}$$

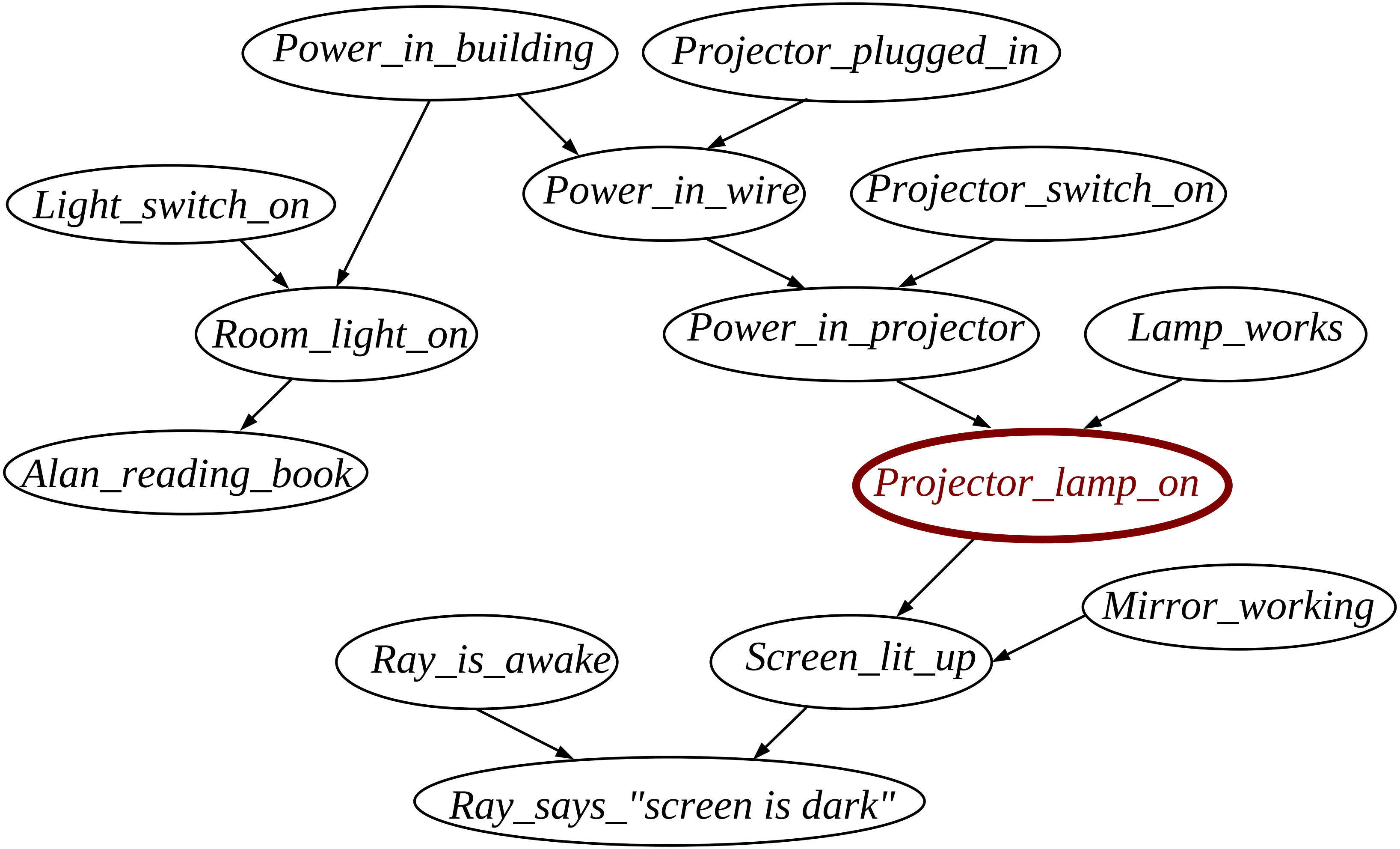
- The parents of X_i $\pi_i \subseteq X_1, \ldots, X_{i-1}$ such that

$$P(X_i|\pi_i) = P(X_i|X_1, \ldots, X_{i-1})$$

- So $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|\pi_i)$

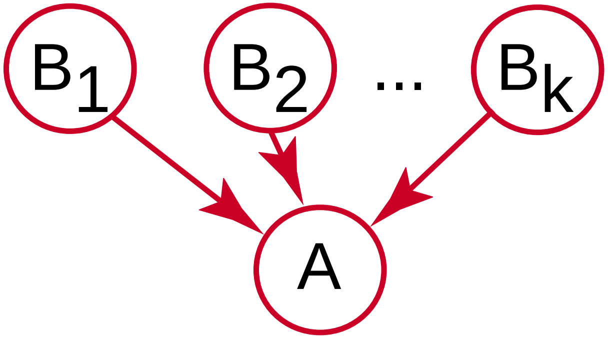
→ Belief network nodes are variables, arcs from parents

Belief Network for Overhead Projector



Belief Network

- Graphical representation of dependence.
- DAG with nodes representing random variables.
- If $B_1, B_2, \cdots, B_k$ are the parents of A:



we have an associated conditional probability:

$$P(A|B_1, B_2, \cdots, B_k)$$

Causality

Belief networks are not necessarily causal. However:

- If the direct causes of a variable are its parents, one would expect that causation would follow the independence of belief networks.
- Conjecture: representing knowledge causally results in a sparser network that is more stable to changing contexts.
- A causal belief network also lets us predict the effect of an intervention: what happens of we change the value of a variable.

Overview

- Knowledge representation, logic, decision theory.
- Belief networks
- Independent Choice Logic
  - Logic programming + arguments
  - Belief networks + first-order rule-structured conditional probabilities
  - Abduction
- Stochastic Dynamic Systems
- Bayesian Learning

Independent Choice Logic

- **C**, the choice space is a set of alternatives.
  An alternative is a set of atomic choices.
  An atomic choice is a ground atomic formula.
  An atomic choice can only appear in one alternative.
- **F**, the facts is an acyclic logic program.
  No atomic choice unifies with the head of a rule.
- P_0 a probability distribution over alternatives:

$$\forall A \in \mathbf{C} \ \sum_{a \in A} P_0(a) = 1.$$

Meaningless Example

$$\mathbf{C} = \{\{c_1, c_2, c_3\}, \{b_1, b_2\}\}$$

$$\begin{aligned} \mathbf{F} = \{\ & f \leftarrow c_1 \wedge b_1, && f \leftarrow c_3 \wedge b_2, \\ & d \leftarrow c_1, && d \leftarrow \overline{c}_2 \wedge b_1, \\ & e \leftarrow f, && e \leftarrow \overline{d}\} \end{aligned}$$

$$P_0(c_1) = 0.5 \quad P_0(c_2) = 0.3 \quad P_0(c_3) = 0.2$$

$$P_0(b_1) = 0.9 \quad P_0(b_2) = 0.1$$

Semantics of ICL

- A total choice is a set containing exactly one element of each alternative in **C**.
- For each total choice τ there is a possible world w_τ.
- Proposition f is true in w_τ (written $w_\tau \models f$) if f is true in the (unique) stable model of $\mathbf{F} \cup \tau$.
- The probability of a possible world w_τ is

$$\prod_{a \in \tau} P_0(a).$$

- The probability of a proposition f is the sum of the probabilities of the worlds in which f is true.

Meaningless Example: Semantics

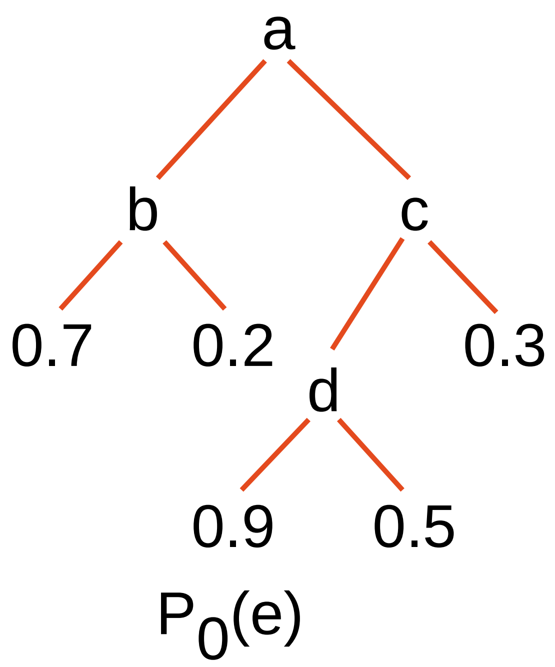
There are 6 possible worlds:

$$
\begin{array}{lllllllr}
w_1 & \models & c_1 & b_1 & f & d & e & P(w_1) = 0.45 \\
w_2 & \models & c_2 & b_1 & \overline{f} & \overline{d} & e & P(w_2) = 0.27 \\
w_3 & \models & c_3 & b_1 & \overline{f} & d & \overline{e} & P(w_3) = 0.18 \\
w_4 & \models & c_1 & b_2 & \overline{f} & d & \overline{e} & P(w_4) = 0.05 \\
w_5 & \models & c_2 & b_2 & \overline{f} & \overline{d} & e & P(w_5) = 0.03 \\
w_6 & \models & c_3 & b_2 & f & \overline{d} & e & P(w_6) = 0.02
\end{array}
$$

$$P(e) = 0.45 + 0.27 + 0.03 + 0.02 = 0.77$$

Decision trees and ICL rules

Decision trees with probabilities on leaves → ICL rules:



$$e \leftarrow a \wedge b \wedge h_1. \qquad P_0(h_1) = 0.7$$

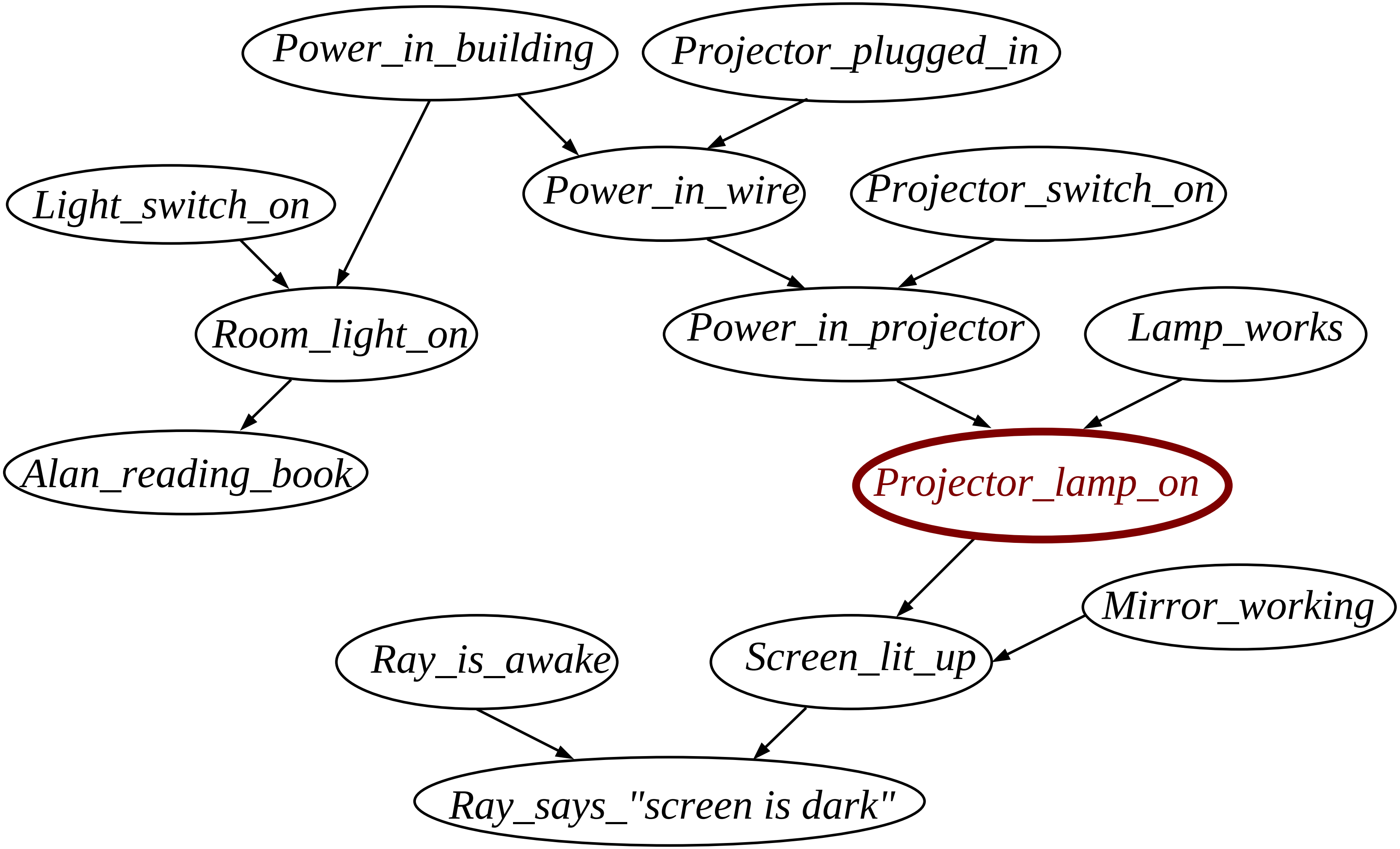
$$e \leftarrow a \wedge \overline{b} \wedge h_2. \qquad P_0(h_2) = 0.2$$

$$e \leftarrow \overline{a} \wedge c \wedge d \wedge h_3. \qquad P_0(h_3) = 0.9$$

$$e \leftarrow \overline{a} \wedge c \wedge \overline{d} \wedge h_4. \qquad P_0(h_4) = 0.5$$

$$e \leftarrow \overline{a} \wedge \overline{c} \wedge h_5. \qquad P_0(h_5) = 0.3$$

Belief Network for Overhead Projector



Belief networks as logic programs

$$projector\_lamp\_on \leftarrow$$

$$power\_in\_projector \land$$

$$lamp\_works \land$$

$$projector\_working\_ok. \longleftarrow \text{atomic choice}$$

$$projector\_lamp\_on \leftarrow$$

$$power\_in\_projector \land$$

$$\overline{lamp\_works} \land$$

$$working\_with\_faulty\_lamp. \longleftarrow \text{atomic choice}$$

Probabilities of hypotheses

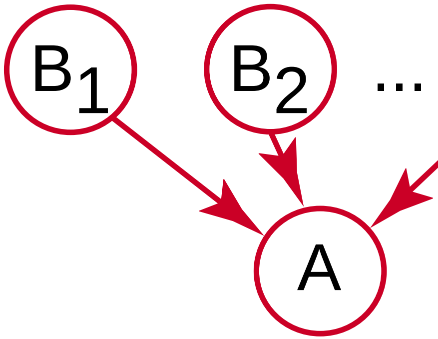
$$P_0(projector\_working\_ok)$$

$$= \quad P(projector\_lamp\_on \mid power\_in\_projector \wedge lamp\_works)$$

— provided as part of Belief network

Mapping belief networks into ICL

There is a local mapping from belief networks into ICL:



is translated into the rules

$$a(V) \leftarrow b_1(V_1) \wedge \cdots \wedge b_k(V_k) \wedge h(V, V_1, \ldots, V_k).$$

and the alternatives

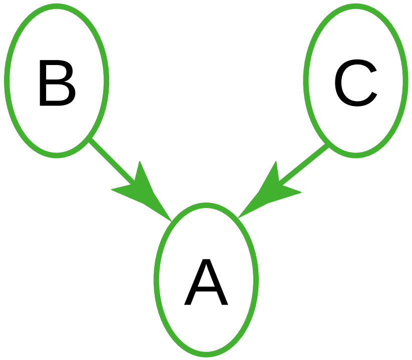
$$\forall v_1 \cdots \forall v_k \{h(v, v_1, \ldots, v_k) | v \in domain(a)\} \in \mathbf{C}$$

Rule-based Inference

Suppose the only rule for a is:

$$a \leftarrow b \wedge c$$

Can we compute the probability of a from the probabilities of b and c?



Rule-based Inference

Suppose the only rule for a is:

$$a \leftarrow b \wedge c$$

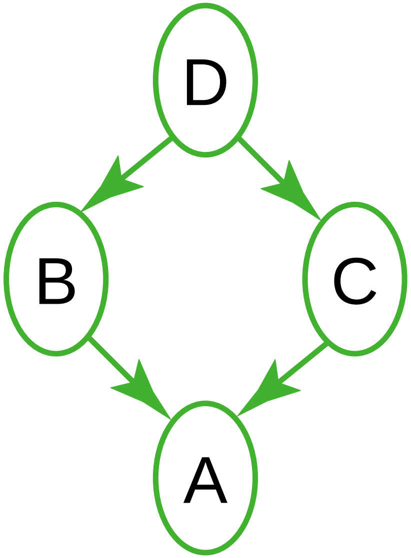
Can we compute the probability of a from the probabilities of b and c?

NO! Consider the rules:

$$b \leftarrow d$$

$$c \leftarrow d$$

$$P_0(d) = 0.5$$



...but you can simply combine explanations.

Assumption-based reasoning

- Given background knowledge / facts F and assumables / possible hypotheses H,
- An explanation of g is a set D of assumables such that

  $F \cup D$ is consistent

  $F \cup D \models g$
- abduction is when g is given and you want D
- default reasoning / prediction is when g is unknown

Abductive Characterization of ICL

- The atomic choices are assumable.
- The elements of an alternative are mutually exclusive.

Suppose the rules are disjoint

$$\left.\begin{array}{l} a \leftarrow b_1 \\ \cdots \\ a \leftarrow b_k \end{array}\right\} \quad b_i \wedge b_j \text{ for } i \neq j \text{ can't be true}$$

$$P(g) = \sum_{E \text{ is a minimal explanation of } g} P(E)$$

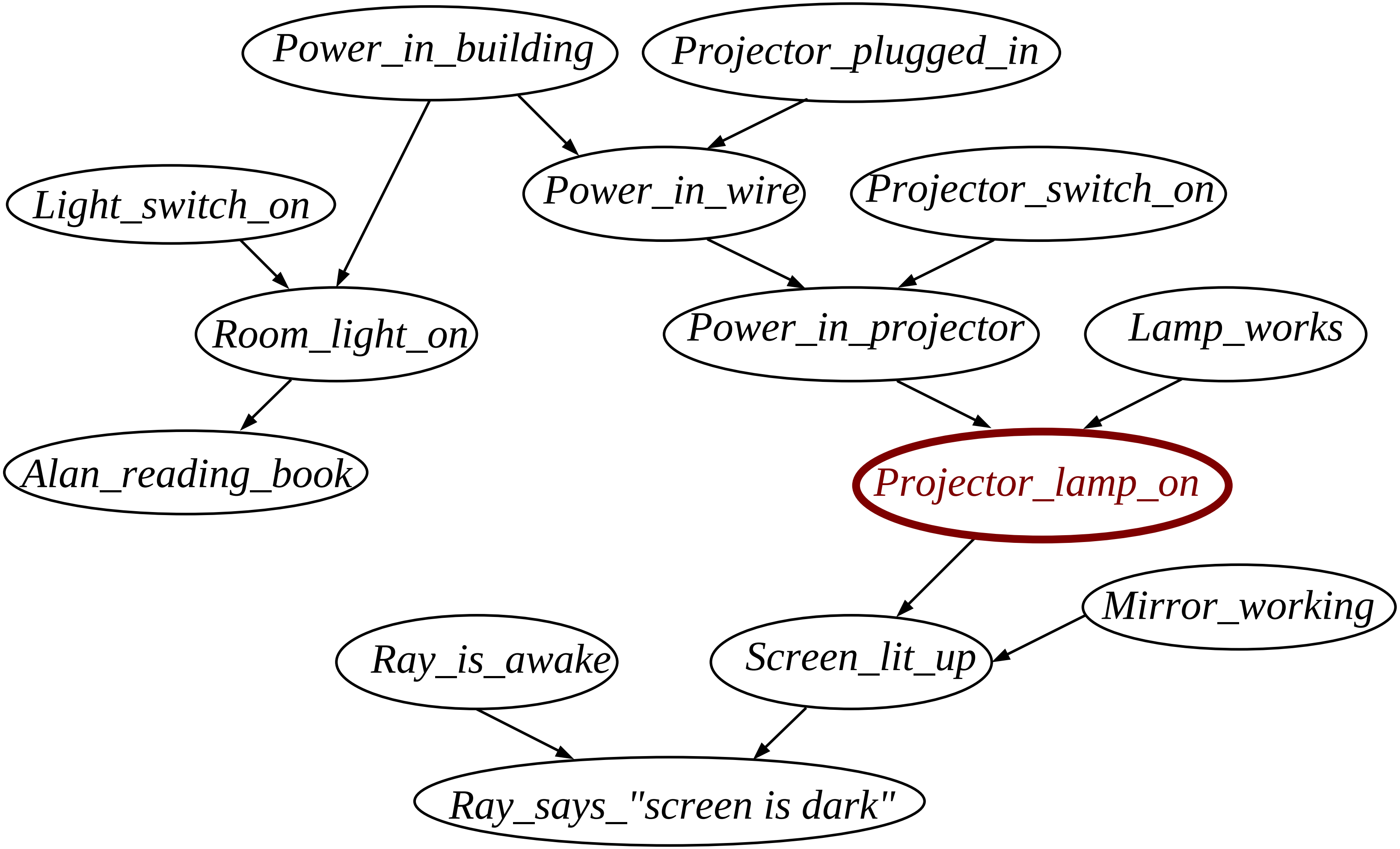
$$P(E) = \prod_{h \in E} P_0(h)$$

Probabilistic Conditioning

$$P(g|e) = \frac{P(g \wedge e)}{P(e)} \quad \begin{array}{l} \longleftarrow \text{explain } g \wedge e \\ \longleftarrow \text{explain } e \end{array}$$

- Given evidence e, explain e then try to explain g from these explanations.
- The explanations of $g \wedge e$ are the explanations of e extended to also explain g.
- Probabilistic conditioning is abduction + prediction.

Belief Network for Overhead Projector



Overview

- Knowledge representation, logic, decision theory.
- Belief networks
- Independent Choice Logic
- Stochastic Dynamic Systems
  - Issues in modelling dynamical systems
  - Representations based on Markov Decision Processes
- Bayesian Learning

Modelling Assumptions

- deterministic or stochastic dynamics
- goals or utilities
- finite stage or infinite stage
- fully observable or partial observable
- explicit state space or properties
- zeroth-order or first-order
- dynamics and rewards given or learned
- single agent or multiple agents

Deterministic or stochastic dynamics

If you knew the initial state and the action, could you predict the resulting state?

Stochastic dynamics are needed if:

- you don't model at the lowest level of detail (e.g., modelling wheel slippage of robots or side effects of drugs)
- exogenous actions can occur during state transitions

Goals or Utilities

- With goals, there are some equally preferred goal states, and all other states are equally bad.
- Not all failures are equal. For example: a robot stopping, falling down stairs, or injuring people.
- With uncertainty, we have to consider how good and bad all possible outcomes are.
  - ➥ utility specifies a value for each state.
- With utilities, we can model goals by having goal states having utility 1 and other states have utility 0.

Finite stage or infinite stage

- **Finite stage** there is a given number of sequential decisions
- **Infinite stage** indefinite number (perhaps infinite) number of sequential decisions.
- With infinite stages, we can model stopping by having an absorbing state — a state s_i so that $P(s_i|s_i) = 1$, and $P(s_j|s_i) = 0$ for $i \neq j$.
- Infinite stages let us model ongoing processes as well as problems with unknown number of stages.

Fully observable or partial observable

- Fully observable = can observe actual state before a decision is made
- Full observability is a convenient assumption that makes computation much simpler.
- Full observability is applicable only for artificial domains, such as games and factory floors.
- Most domains are partially observable, such as robotics, diagnosis, user modelling …

Explicit state space or properties

- Traditional methods relied on explicit state spaces, and techniques such as sparse matrix computation.
- The number of states is exponential in the number of properties or variables. It may be easier to reason with 30 binary variables than 1,000,000,000 states.
- Bellman labelled this the *Curse of Dimensionality*.

Zeroth-order or first-order

- The traditional methods are zero-order, there is no logical quantification. All of the individuals must be part of the explicit model.
- There is some work on automatic construction of probabilistic models — they provide macros to construct ground representations.
- Naive use of unification does not work, as we can't treat the rules separately.

Dynamics and rewards given or learned

- Often we don't know a priori the probabilities and rewards, but only observe the system while controlling it
  ➥ reinforcement learning.
- Credit and blame attribution.
- Exploration—exploitation tradeoff.

Single agent or multiple agents

- Many domains are characterised by multiple agents rather than a single agent.
- Game theory studies what agents should do in a multi-agent setting.
- Even if all agents share a common goal, it is exponentially harder to find an optimal multi-agent plan than a single agent plan.

Overview

- Knowledge representation, logic, decision theory.
- Belief networks
- Independent Choice Logic
- Stochastic Dynamic Systems
  - Issues in modelling dynamical systems
  - Representations based on Markov Decision Processes
- Bayesian Learning

Markov Process

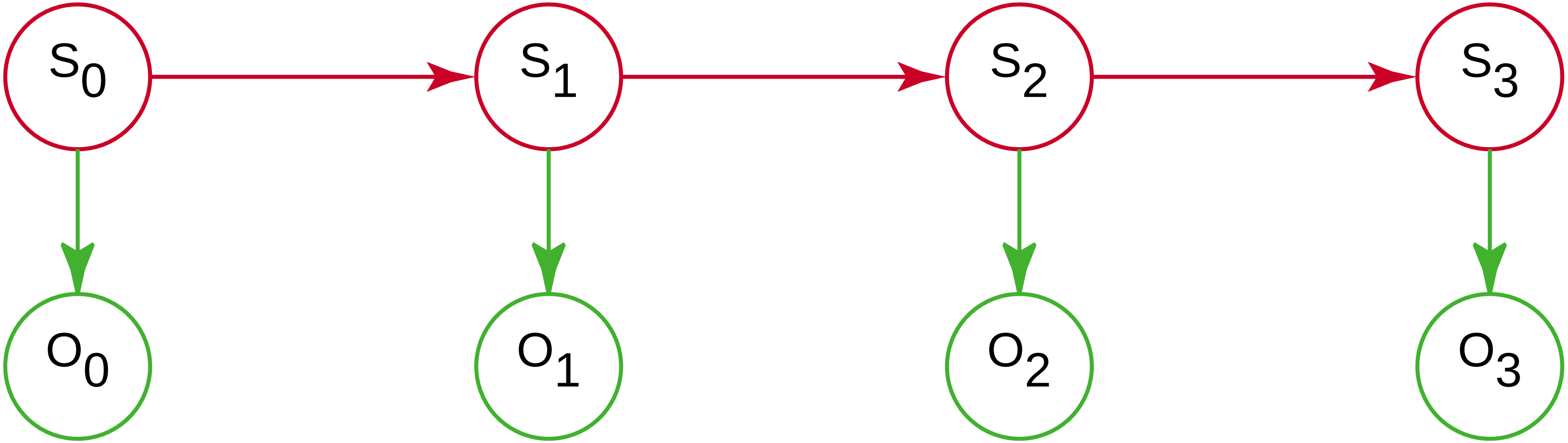
- $P(S_{t+1}|S_t)$ specified the dynamics
- In the ICL:

$$state(S, T+1) \leftarrow$$

$$state(S0, T) \wedge trans(S0, S).$$

$$\forall s \{trans(s, s_0), \ldots, trans(s, s_n)\} \in \mathbf{C}$$

Hidden Markov Model



$P(S_{t+1}|S_t)$ specified the dynamics

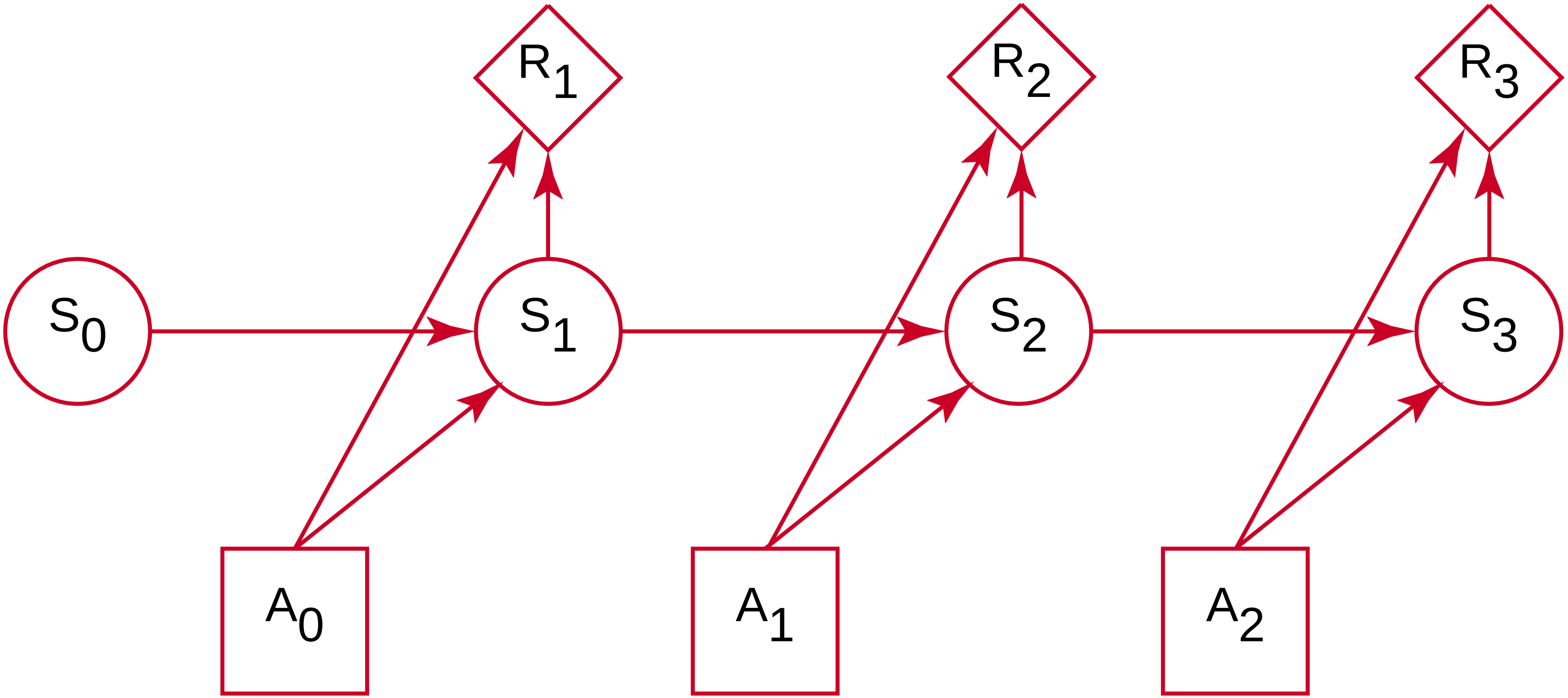
$P(O_t|S_t)$ specifies the sensor model.

$$observe(O, T) \leftarrow state(S, T) \wedge obs(S, O).$$

For each state s, there is an alternative:

$\{obs(s, o_1), \ldots, obs(s, o_k)\}$.

Markov Decision Process



$P(S_{t+1}|S_t, A_t)$ specified the dynamics

$R(S_t, A_{t-1})$ specifies the reward at time t

Discounted value is $R_1 + \gamma R_2 + \gamma^2 R_3 + \ldots.$

Dynamics for MDP

$P(S_{t+1}|S_t, A_t)$ represented in the ICL as:

$$state(S, T+1) \leftarrow$$
$$state(S0, T) \wedge$$
$$do(A, T) \wedge$$
$$trans(S0, A, S).$$

$$\forall s \forall a \{trans(s, a, s_0), \ldots, trans(s, a, s_n)\} \in \mathbf{C}$$

Policies

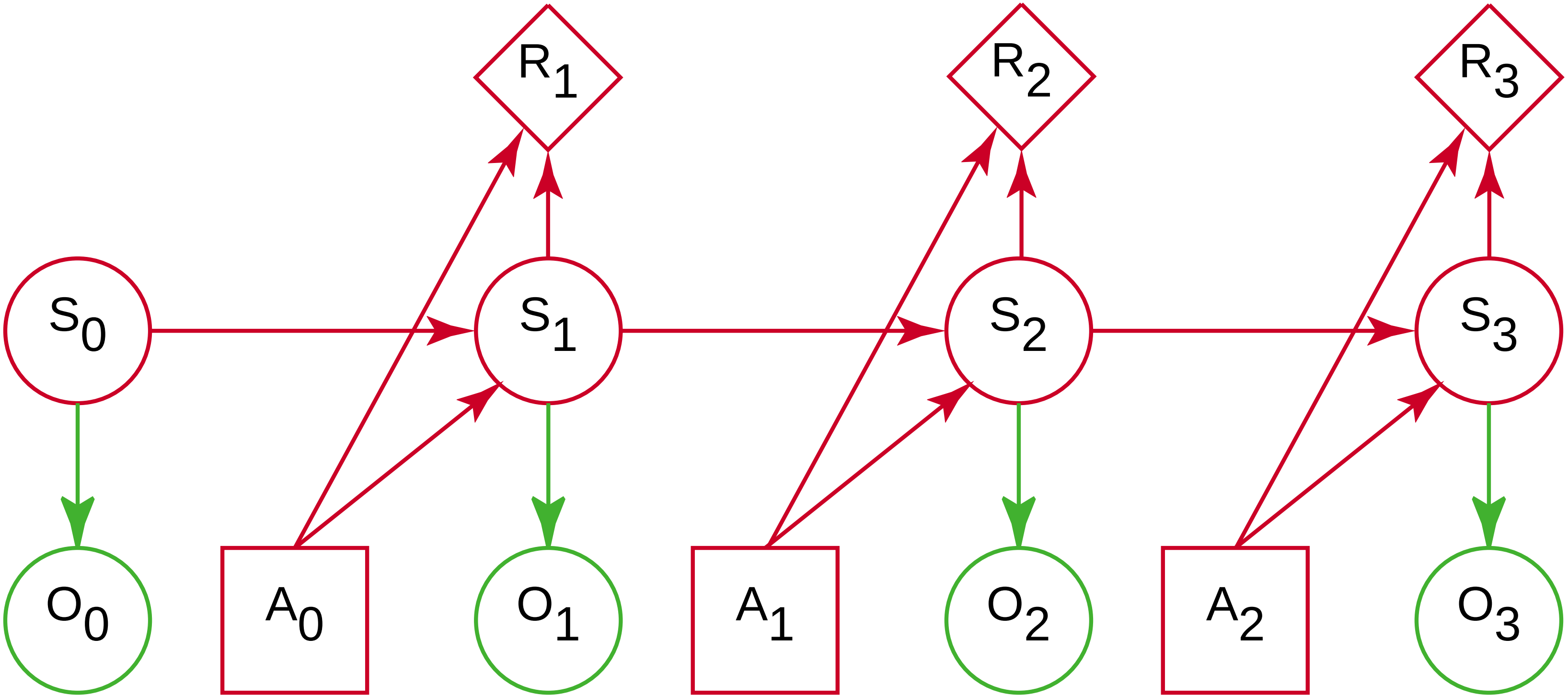
- What the agent does based on its perceptions is specified by a **policy.**
- For fully observable MDPs, a policy is a function from observed state into actions:

$$policy : S_t \rightarrow A_t$$

- A policy can be represented by rules of the form:

$$do(a, T) \leftarrow$$
$$state(s, T).$$

Partially Observable MDP (POMDP)



$P(S_{t+1}|S_t, A_t)$ specified the dynamics

$P(O_t|S_t)$ specifies the sensor model.

$R(S_t, A_{t-1})$ specifies the reward at time i

Policies

- What the agent does based on its perceptions is specified by a policy a function from history into actions:

$$O_0, A_0, O_1, A_1, \ldots, O_{t-1}, A_{t-1}, O_t \to A_t$$

- For POMDPs, a belief state is a probability distribution over states. A belief state is an adequate statistic about the history.

$$policy : B_t \to A_t$$

If there are n states, this is a function on $\Re^n$.

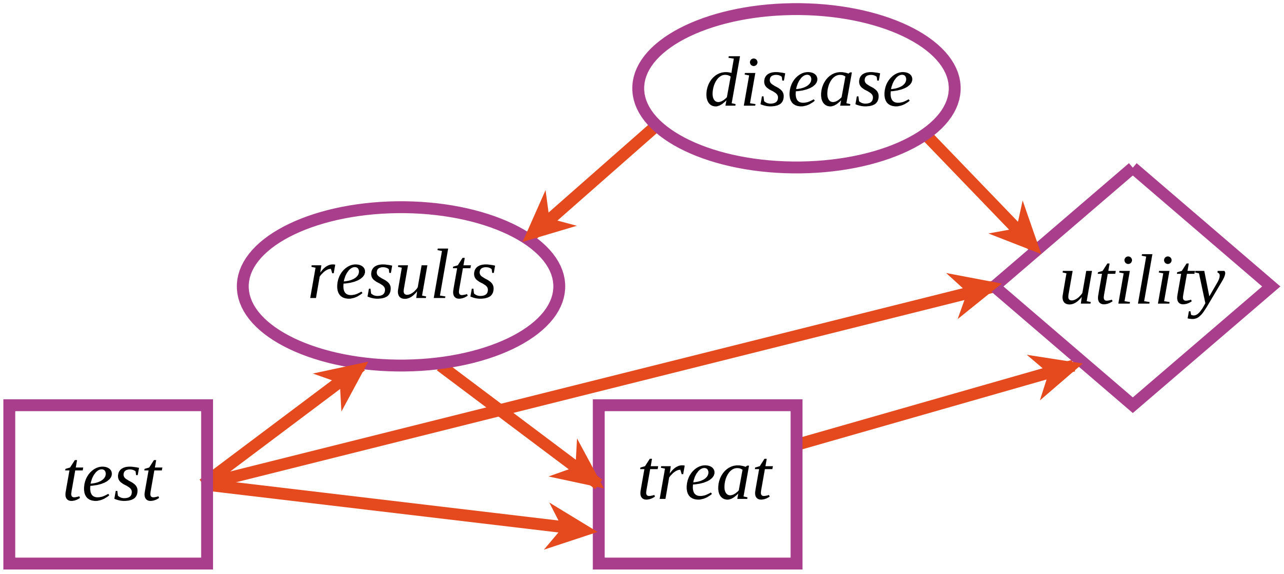
Reinforcement Learning

Use (fully observable) MDP model, but the state transition function and the reward function are not given, but must be learned from acting in the environment.

- exploration versus exploitation
- model-based algorithms (learn the probabilities) or model-free algorithms (don't learn the state transition or reward functions).
- The use of properties is common in reinforcement learning. For example, using a neural network to model the dynamics and reward functions or the value function.

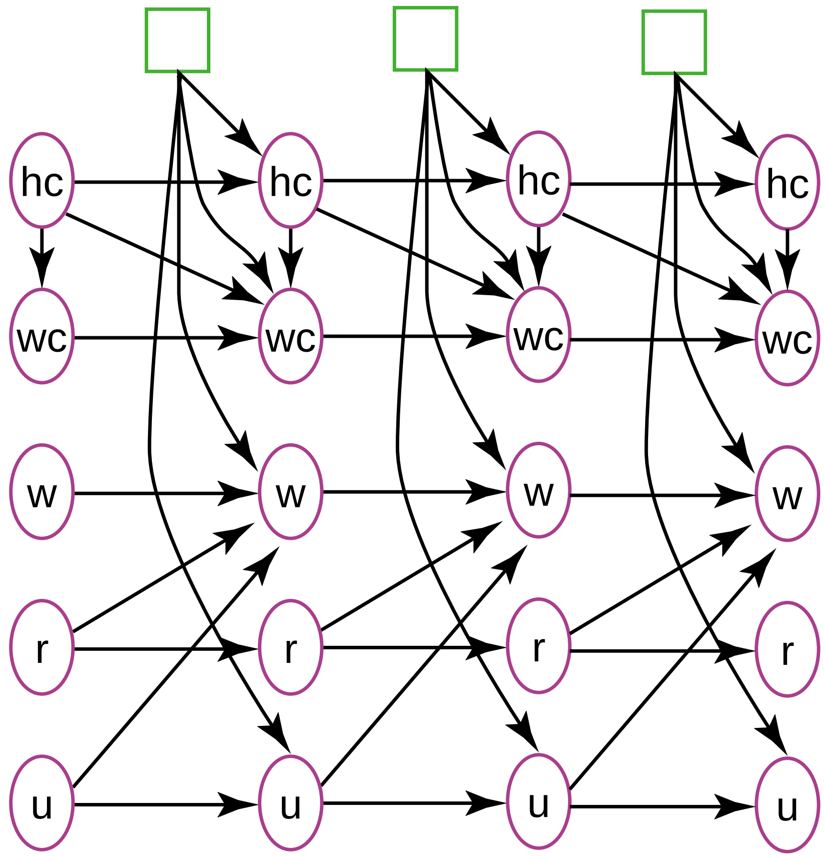
Influence Diagrams

An influence diagram is a belief network with decision nodes (rectangles) and a value node (diamond).



Dynamic Belief Networks

Idea: represent the state in terms of random variables / propositions.



DBN in ICL

$$r(T+1) \leftarrow r(T) \wedge rain\_continues(T).$$

$$r(T+1) \leftarrow \overline{r(T)} \wedge rain\_starts(T).$$

$$hc(T+1) \leftarrow hc(T) \wedge do(A,T) \wedge A \neq pass\_coffee$$
$$\wedge\, keep\_coffee(T).$$

$$hc(T+1) \leftarrow hc(T) \wedge do(pass\_coffee,T)$$
$$\wedge\, keep\_coffee(T) \wedge passing\_fails(T).$$

$$hc(T+1) \leftarrow do(get\_coffee,T) \wedge get\_succeeds(T).$$

$$\forall T\{rain\_continues(T), rain\_stops(T)\} \in \mathbf{C}$$

$$\forall T\{keep\_coffee(T), spill\_coffee(T)\} \in \mathbf{C}$$

$$\forall T\{passing\_fails(T), passing\_succeeds(T)\} \in \mathbf{C}$$

Modelling Assumptions

- deterministic or stochastic dynamics
- goals or utilities
- finite stage or infinite stage
- fully observable or partial observable
- explicit state space or properties
- zeroth-order or first-order
- dynamics and rewards given or learned
- single agent or multiple agents

Comparison of Some Representations

	CP	DTP	IDs	RL	HMM	GT
stochastic dynamics		✔	✔	✔	✔	✔
values		✔	✔	✔		✔
infinite stage	✔	✔		✔	✔	
partially observable			✔		✔	✔
properties	✔	✔	✔	✔		✔
first-order	✔					
dynamics not given				✔	✔	
multiple agents						✔

Other Issues

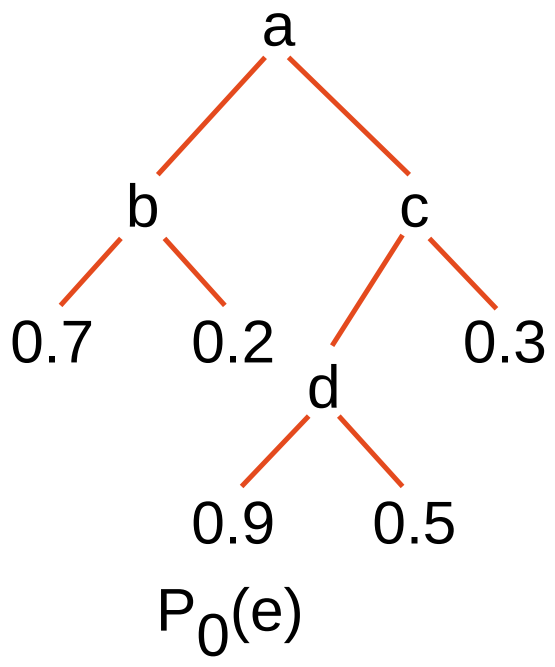
- Modelling and reasoning at multiple levels of abstraction abstracting both states and times
- Approximate reasoning and approximate modelling
- Bounded rationality: how to balance acting and thinking. Value of thinking.

Overview

- Knowledge representation, logic, decision theory.
- Belief networks
- Independent Choice Logic
- Stochastic Dynamic Systems
- Bayesian Learning
  - Learning belief networks
  - Belief networks for learning

Decision trees and rules

Decision trees with probabilities on leaves → rules:



$$e \leftarrow a \wedge b \wedge h_1. \qquad P_0(h_1) = 0.7$$

$$e \leftarrow a \wedge \overline{b} \wedge h_2. \qquad P_0(h_2) = 0.2$$

$$e \leftarrow \overline{a} \wedge c \wedge d \wedge h_3. \qquad P_0(h_3) = 0.9$$

$$e \leftarrow \overline{a} \wedge c \wedge \overline{d} \wedge h_4. \qquad P_0(h_4) = 0.5$$

$$e \leftarrow \overline{a} \wedge \overline{c} \wedge h_5. \qquad P_0(h_5) = 0.3$$

A common way to learn belief networks

- Totally order the variables.
- Build a decision tree for each for each variable based on its predecessors.
- Search over different orderings.

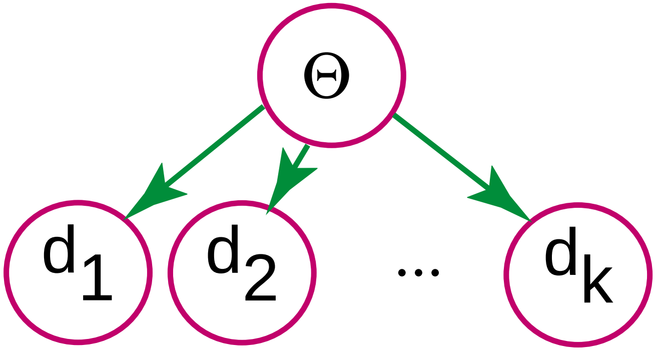
Issues in learning belief networks

There is a good understanding of:

- noisy data
- combining background knowledge and data
- observational and experimental data
- hidden variables
- missing data

Belief networks for learning

Suppose we observe data $d_1, d_2, \ldots, d_k$, i.i.d.

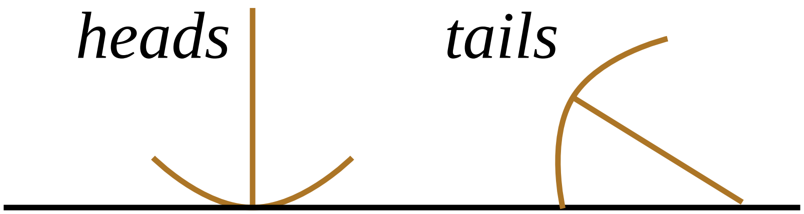


Domain of Θ is the set of all models (sometimes model parameters).

Bayesian learning compute $P(\Theta | d_1, d_2, \ldots, d_k)$

Classic example

Estimate the probability a drawing pin lands "heads"



$$heads(E) \leftarrow prob\_heads(P) \wedge lands\_heads(P, E).$$

$$tails(E) \leftarrow prob\_heads(P) \wedge lands\_tails(P, E).$$

$$\forall P \forall E \{lands\_heads(P, E), lands\_tails(P, E)\} \in \mathbf{C}$$

$$\{prob\_heads(V) : 0 \leq V \leq 1\} \in \mathbf{C}$$

$$P_0(lands\_heads(P, E) = P.$$

$$P_0(lands\_tails(P, E) = 1 - P.$$

Explaining the data

To explain data:

$$heads(e_1), tails(e_2), tails(e_3), heads(e_4), \ldots$$

there is an explanation:

$$\{lands\_heads(p, e_1), lands\_tails(p, e_2), lands\_tails(p, e_3), lands\_heads(p, e_4), \ldots, prob\_heads(p)\}$$

for each $p \in [0, 1]$.

This explanation has probability:

$$p^{\#heads}(1-p)^{\#tails}P_0(prob\_heads(p))$$

Where to now?

- Keep the representation as simple as possible to solve your problem, but no simpler.
- Approximate. Bounded rationality.
- Approximate the solution, not the problem (Sutton).
- We want everything, but only as much as it is worth to us.
- Preference elicitation.

Conclusions

- If you are interested in acting in real domains you need to treat uncertainty seriously.
- There is a large community working on stochastic dynamical systems for robotics, factory control, diagnosis, user modelling, multimedia presentation, collaborative filtering …
- There is much the computational logic community can contribute to this endeavour.