Modern Artificial Intelligence: Computer Science and Beyond

David Poole

University of British Columbia

☞

Caveat

- This is a personal view and not one of LCI.
- It does not attempt to cover the breadth of work in LCI.
- It does not attempt to cover the breadth of work in AI.
- My intention is to provoke thought and discussion.

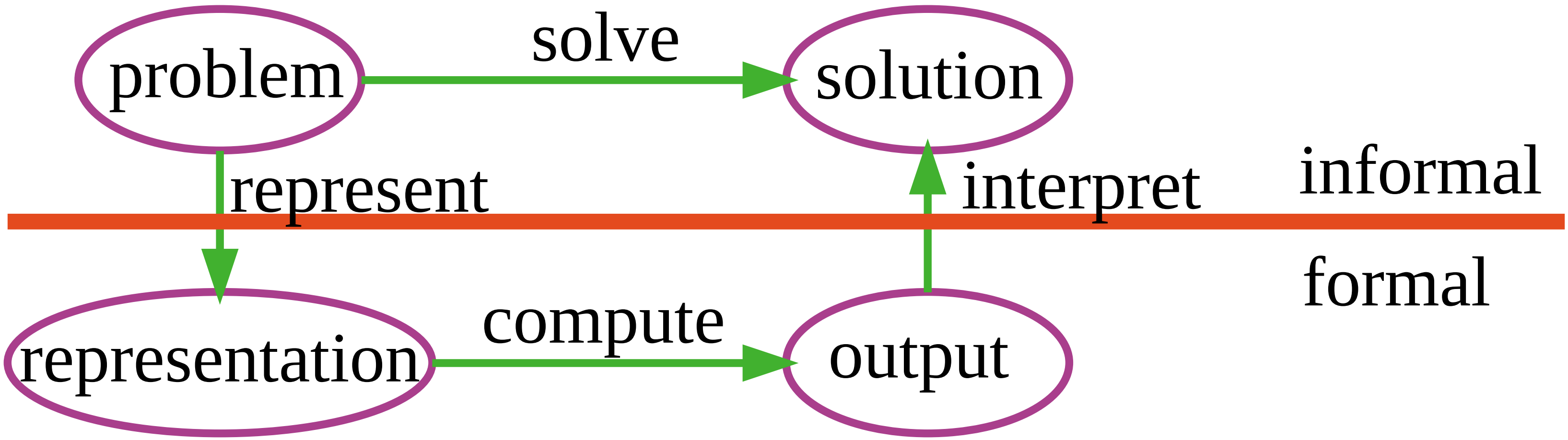
Overview

- **AI, Computer Science and representations**
  - What is Computer Science?
  - Representations
  - An example representation
- Agents in an environment
- Stochastic Dynamic Systems

What is Computer Science?

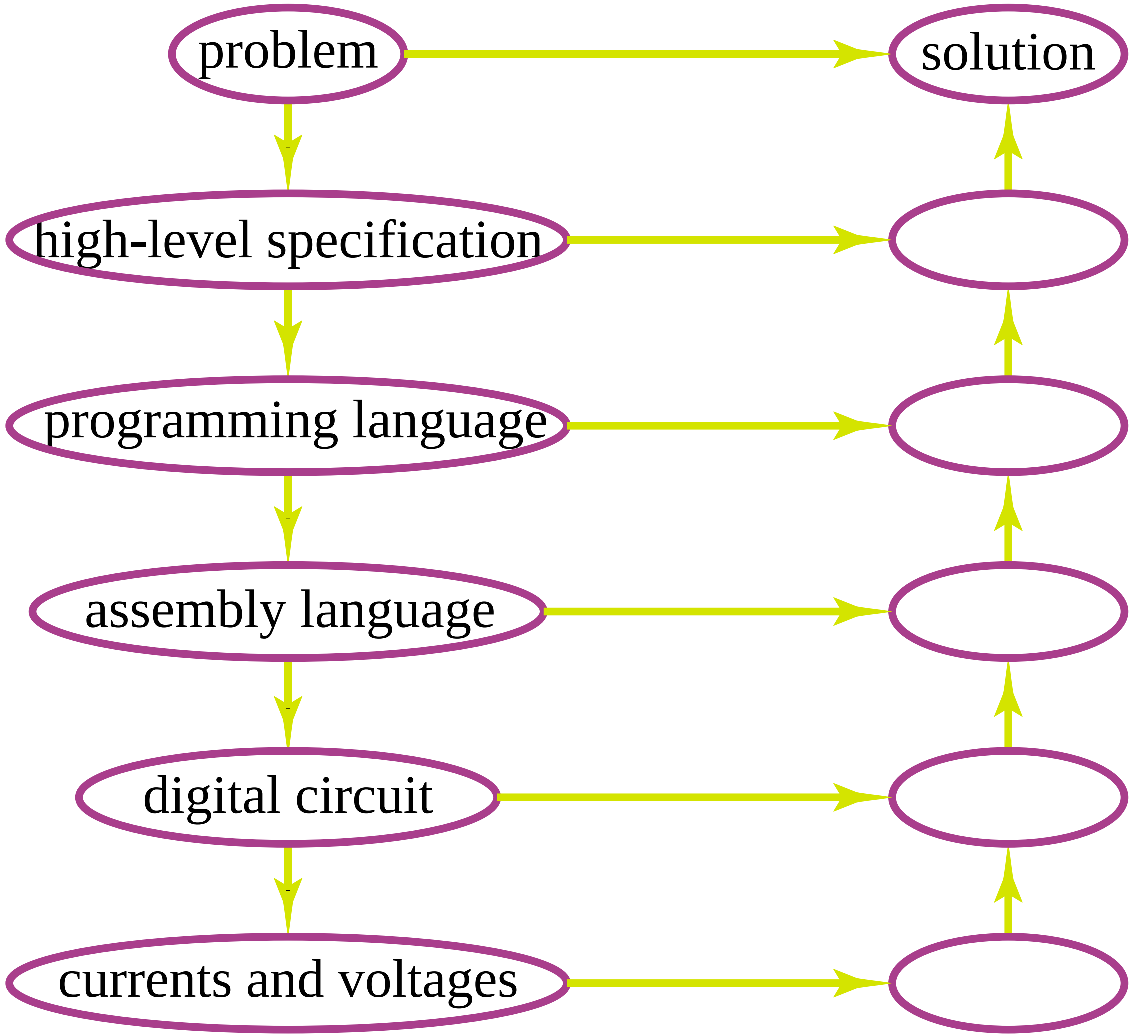
- the study of useful abstractions and how they can be practically realised on current and future computers.
- the study of representations and algorithms and their limitations.

Representations



Example representations: machine language, C, Java, Prolog, natural language

Representations in Traditional CS



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: compact, natural and maintainable. Elaboration tolerant.
- amenable to efficient computation;
  able to express features of the problem we can exploit for computational gain.
- learnable from data and past experiences.
- able to trade off accuracy and computation time.

Example Representation: belief networks

A **belief network** (or Bayesian network) is a graphical representation of independence amongst random variables.

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

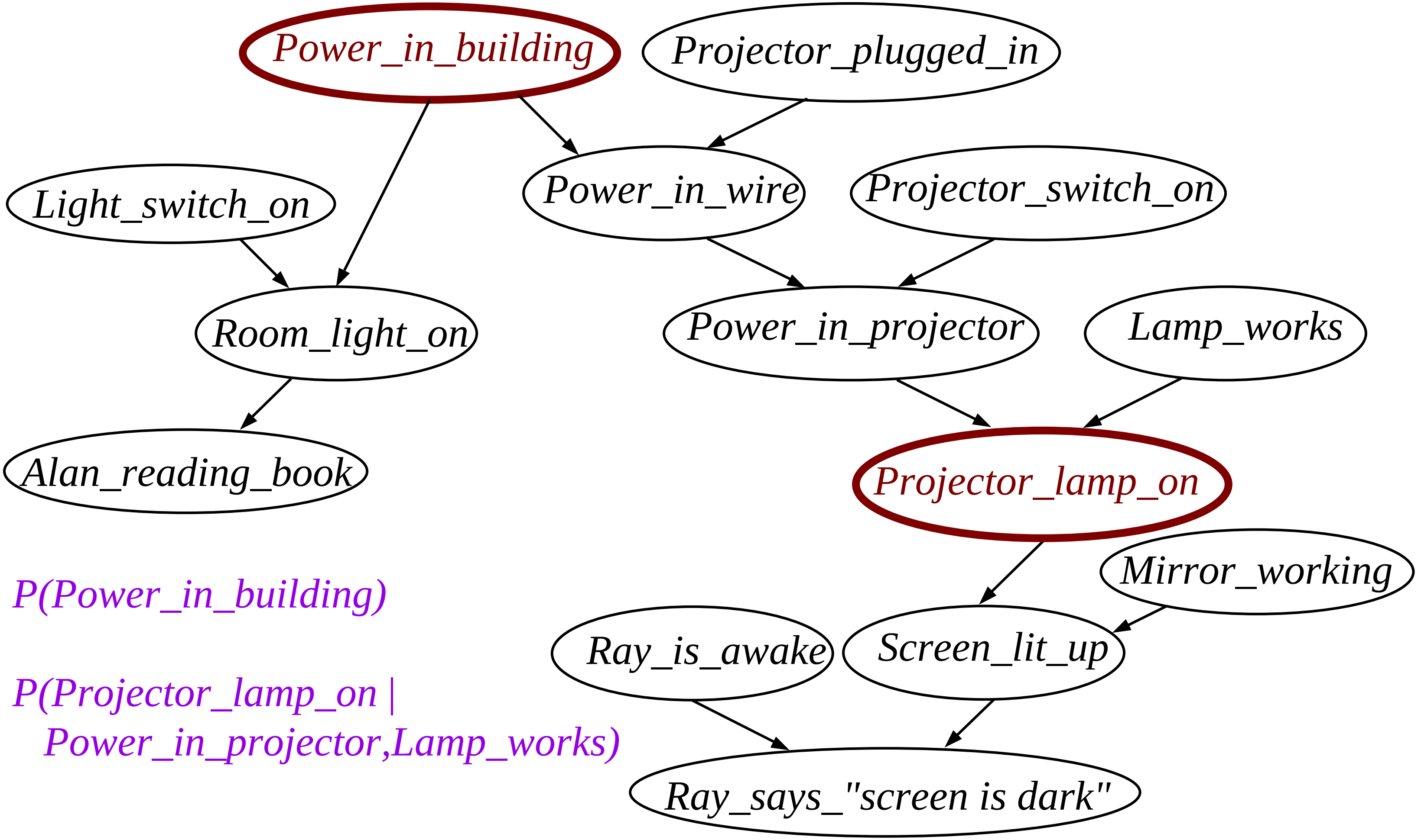
$$\begin{aligned} P(X_1, \ldots, X_n) &= P(X_1)P(X_2|X_1) \cdots P(X_n|X_1, \ldots, X_{n-1}) \\ &= \prod_{i=1}^{n} P(X_i|X_1, \ldots, X_{i-1}) \end{aligned}$$

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

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

↪ **Belief network** nodes are variables, arcs from parents

Belief Network for Overhead Projector



P(Power_in_building)

P(Projector_lamp_on | Power_in_projector,Lamp_works)

Overview

- AI, Computer Science and representations
- Agents in an environment
  - What is an agent?
  - Examples of agents.
  - What should an agent do?
  - Representations for deciding that to do.
- Stochastic Dynamic Systems

What is the problem we want to solve?

What is the problem we want to solve?

What should an (intelligent) agent do?

What is the problem we want to solve?

What should an (intelligent) agent do?

An agent is something that acts in an environment.

What is the problem we want to solve?

What should an (intelligent) agent do?

An agent is something that acts in an environment.

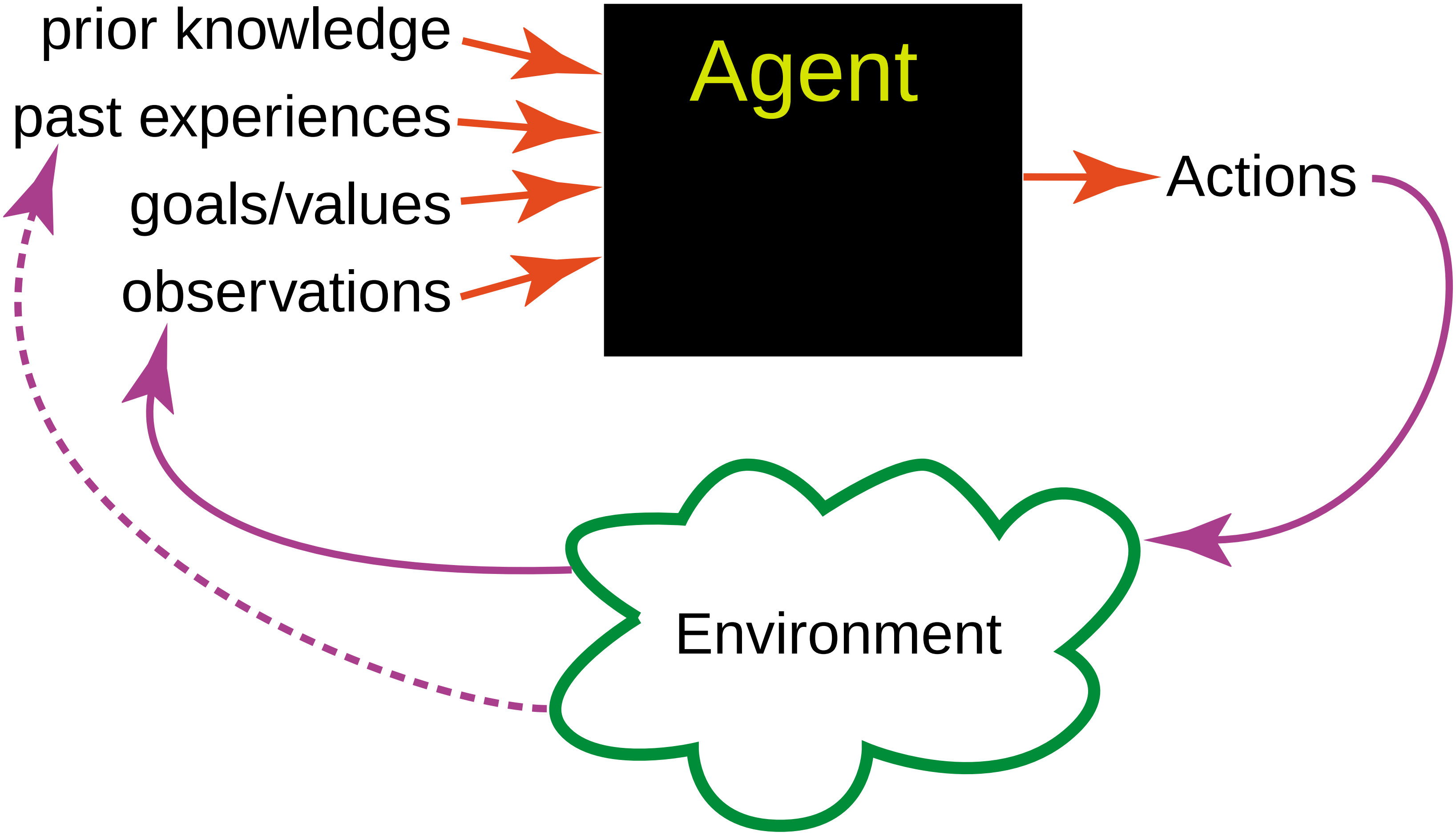
An intelligent agent acts intelligently:

- what it does is appropriate for its circumstances and goals
- it is flexible to changing circumstances and goals
- it learns from experience
- it makes appropriate choices given perceptual limitations and finite computation

Examples of Agents

- **Organizations** Microsoft, Al Qaeda, Government of Canada, UBC, CS Dept,...
- **People** teachers, physicians, stock traders, engineers, researchers, farmers, waiters...
- **Computers** user interfaces, airplane controllers, network controllers, games, advising systems, tutoring systems, diagnostic assistants...
- **Animals** dogs, mice, birds, insects, worms,...

An agent acting in an environment



Example agent: robot

- **actions:** movement, grippers, speech, facial expressions,…
- **observations:** vision, sonar, sound, speech recognition, gesture recognition,…
- **goals:** deliver food, rescue people, score goals, explore,…
- **past experiences:** effect of steering, slipperiness, how people move,…
- **prior knowledge:** what is important feature, categories of objects, what a sensor tell us,…

Example agent: teacher

- **actions:** present new concept, drill, give test, explain concept,…
- **observations:** test results, facial expressions, errors, focus,…
- **goals:** particular knowledge, skills, inquisitiveness, social skills,…
- **past experiences:** prior test results, effects of teaching strategies, …
- **prior knowledge:** subject material, teaching strategies,…

Example agent: medical doctor

- **actions:** operate, test, prescribe drugs, explain instructions,…
- **observations:** verbal symptoms, test results, visual appearance…
- **goals:** remove disease, relieve pain, increase life expectancy, reduce costs,…
- **past experiences:** treatment outcomes, effects of drugs, test results given symptoms…
- **prior knowledge:** possible diseases, symptoms, possible causal relationships…

Example agent: user interface

- actions: present information, ask user, find another information source, filter information, interrupt,…
- observations: users request, information retrieved, user feedback, facial expressions…
- goals: present information, maximize useful information, minimize irrelevant information, privacy,…
- past experiences: effect of presentation modes, reliability of information sources,…
- prior knowledge: information sources, presentation modalities…

What should an agent do?

It depends on its:

- capabilities
- beliefs
- preferences (values/goals)
- perceptions

Preferences

Alice . . . went on “Would you please tell me, please, which way I ought to go from here?”

“That depends a good deal on where you want to get to,” said the Cat.

“I don’t much care where —” said Alice.

“Then it doesn’t matter which way you go,” said the Cat.

Lewis Carroll, 1832–1898

Alice’s Adventures in Wonderland, 1865

Chapter 6

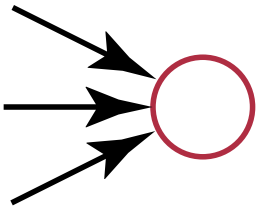
Acting as Gambling

- When an agent doesn't have complete and accurate knowledge of the world and the effect of its actions, it is gambling.

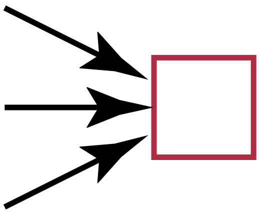
- Usually you can't opt out and not gamble — doing nothing is another action with consequences.

- Probability and (Bayesian) decision theory are the appropriate calculi for gambling.
  - probability as a measure of belief
  - utility as a measure for preferences

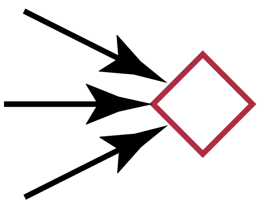
Decision Networks



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

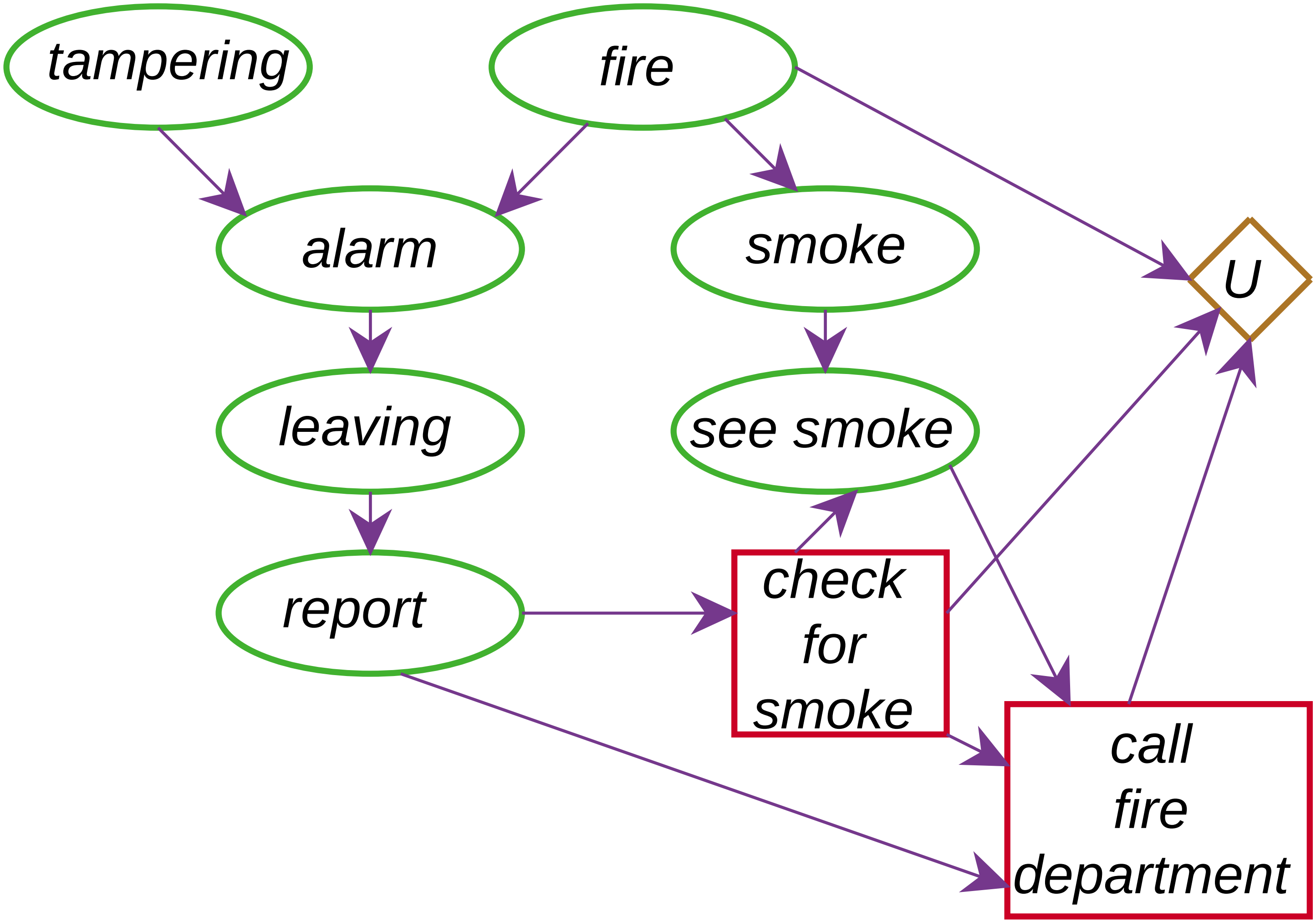


- A **decision variable** is drawn as an rectangle. Arcs into the node represent information available when the decision is make.



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

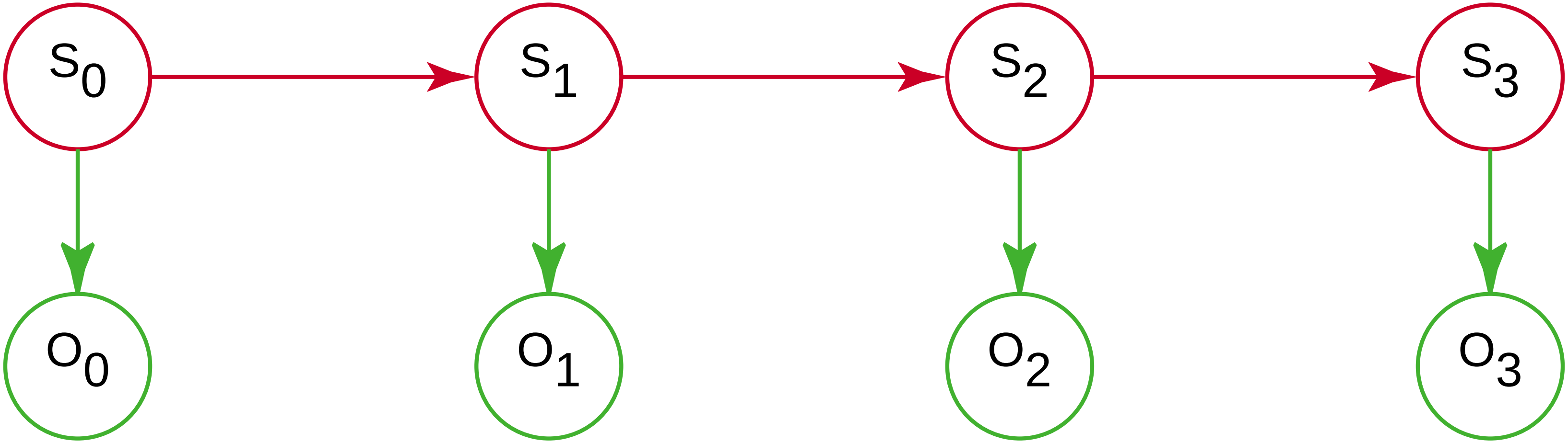
Example Decision Network



Markov Process

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

Hidden Markov Model



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

Overview

- AI, Computer Science and representations
- Agents in an environment
- Stochastic Dynamic Systems
  - Dimensions in modelling dynamic systems
  - What we know how to do
  - Limits of our understanding

Dimensions of Representations

- deterministic or stochastic dynamics
- finite stage or infinite stage
- fully observable or partially observable
- explicit state space or properties
- zeroth-order or first-order
- dynamics and rewards given or learned
- single level of abstraction or multiple levels of abstraction
- single agent or multiple agents
- perfect rationality or bounded rationality

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.

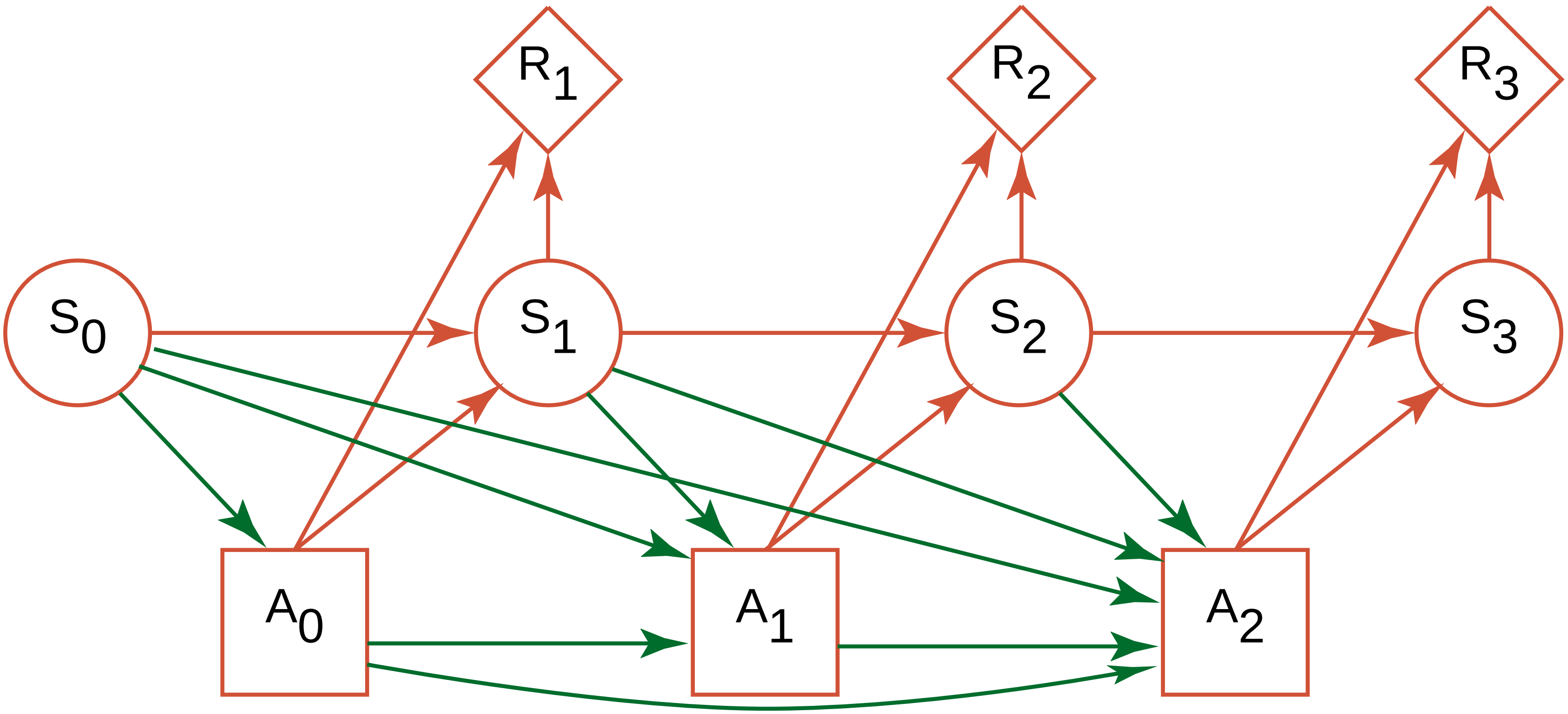
Finite stage or infinite stage

- **Finite stage** there is a given number of sequential decisions
- **Infinite stage** indefinite number (perhaps infinite) number of sequential decisions.
- Infinite stages let us model ongoing processes as well as problems with unknown number of stages.

Fully observable or partially observable

- Fully observable = the agent 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 …
- Partial observability means having theories of perception such as vision or natural language.

Markov Decision Process (MDP)

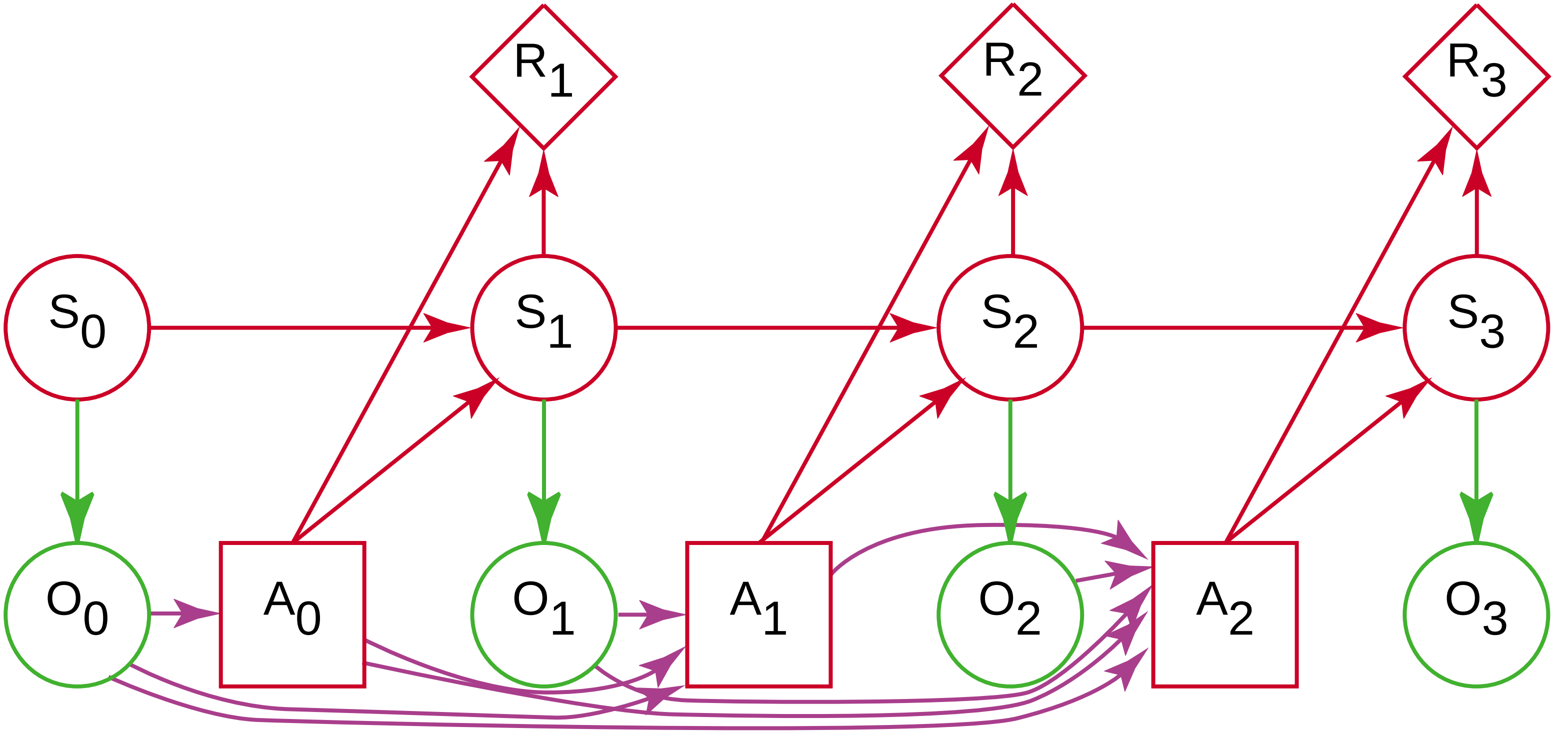


$P(S_{t+1}|S_t, A_t)$ specified the dynamics

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

Value is $R_1 + R_2 + R_3$.

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

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*.
- Planning is concerned with reasoning about properties to fulfill goals.

Zeroth-order or first-order

- The traditional decision-theoretic methods are zero-order; there is no logical quantification. All of the individuals must be part of the explicit model.

- For many domains the individuals present are not known when the model is created and change during run time.

- Example: a robot may want to reason about particular individuals (e.g., those it has served).

- Example: an intelligent tutoring system may want to reason about particular examples given to particular students; the examples are customised for each student.

Single or multiple levels of abstraction

- Do you reason at a single level of abstraction (e.g., single discretization of time) or at multiple levels of abstraction at once?
- Example: a robot may need reason about its next high-level task as well as which steering angle to choose next.
- Example: an intelligent tutoring system may need to reason about the high-level course plan as well as which example to present next.
- A particular case is for Hybrid systems where you need to reason at continuous and discrete levels.

Dynamics and rewards given or learned

- Often we don't know a priori the model, the probabilities or the rewards, but only observe the system while controlling it
  - reinforcement learning.
- Exploration—exploitation tradeoff.
- We learn models of dynamics and preferences where some variables are observable, some are controllable and some are neither.
- This is part of the large field of Machine Learning.

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.
- Agents can be cooperative, competitive or somewhere in between.
- Finding optimal multi-agent strategies is exponentially harder than finding single-agent strategies (even if the agents share the same values)
- Often stochastic policies are optimal

Perfect or Bounded Rationality

- We cannot assume agents have unlimited computation time and space.
- It may be better to find a reasonable decision fast than take a long time to find what (was) the best decision.
- Value of computation. Value of space. How much is thinking worth to the agent?
- Offline versus online computation.

What we know how to do:

- finite stage or infinite stage
- fully observable or partially observable
- explicit state space or properties
- zeroth-order or first-order
- single or multiple-levels of abstraction
- dynamics and rewards given or learned
- single agent or multiple agents
- perfect rationality or bounded rationality

The limit of our understanding (1):

- finite stage or infinite stage
- fully observable or partially observable
- explicit state space or properties
- zeroth-order or first-order
- single or multiple-levels of abstraction
- dynamics and rewards given or learned
- single agent or multiple agents
- perfect rationality or bounded rationality

The limit of our understanding (2):

- finite stage or infinite stage
- fully observable or partially observable
- explicit state space or properties
- zeroth-order or first-order
- single or multiple-levels of abstraction
- dynamics and rewards given or learned
- single agent or multiple agents
- perfect rationality or bounded rationality

The limit of our understanding (3):

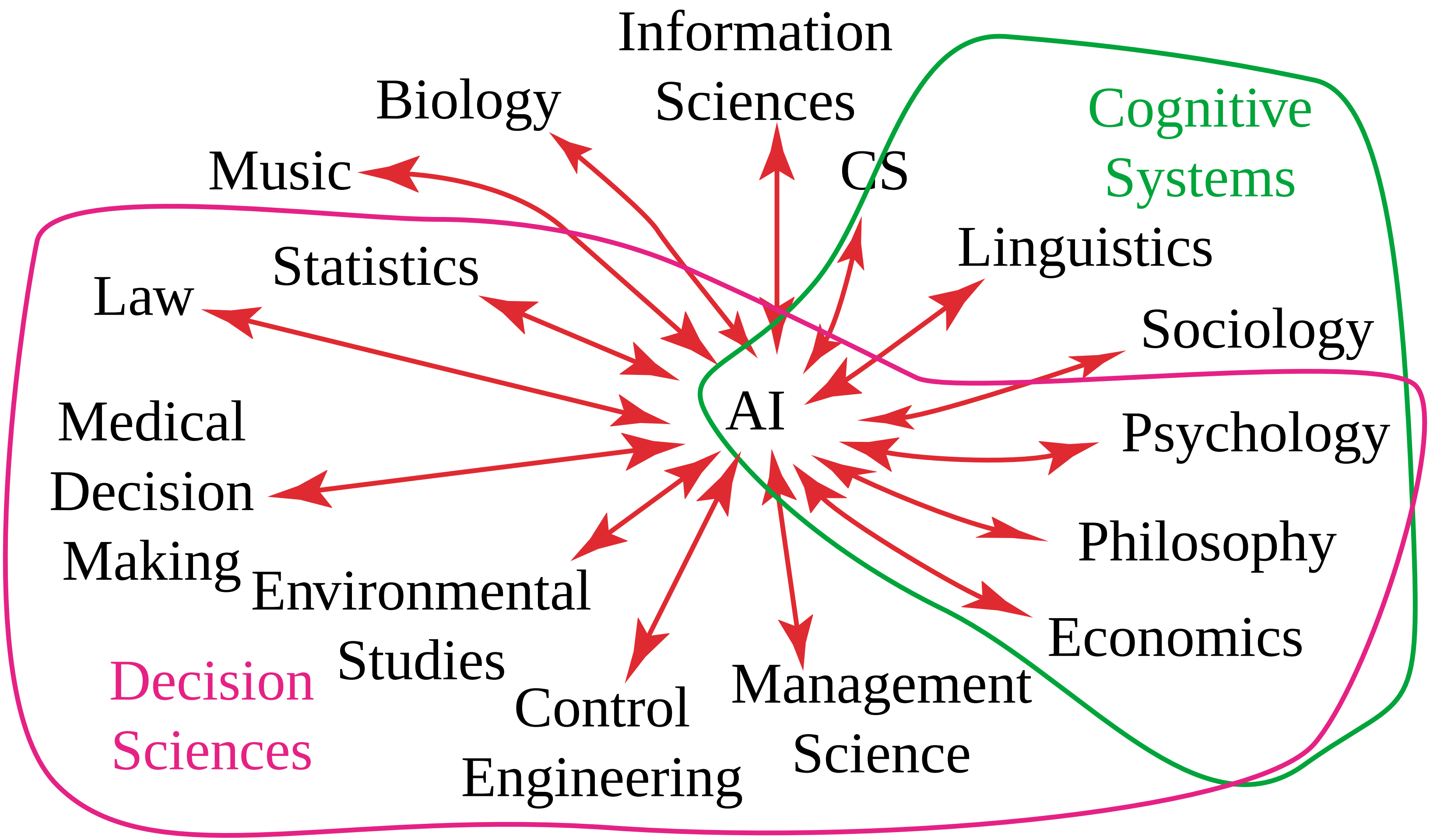
- **deterministic** or stochastic dynamics
- finite stage or **infinite stage**
- fully observable or **partially observable**
- explicit state space or **properties**
- **zeroth-order** or first-order
- single or **multiple** levels of abstraction
- **dynamics and rewards given** or learned
- **single agent** or multiple agents

Algorithms

There have been a revolution in algorithms for difficult (e.g., NP-complete) problems:

- Exploiting the structure found in real problems
- Randomized algorithms

AI is interdisciplinary



Who is the perfect candidate?

- they are making progress in one of the difficult problems
- they are making fundamental progress on problems in an interesting domain
- they are doing things that we haven't anticipated

What do **YOU** think?