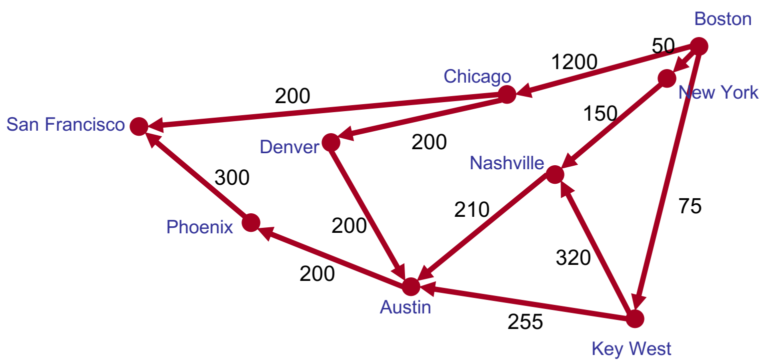
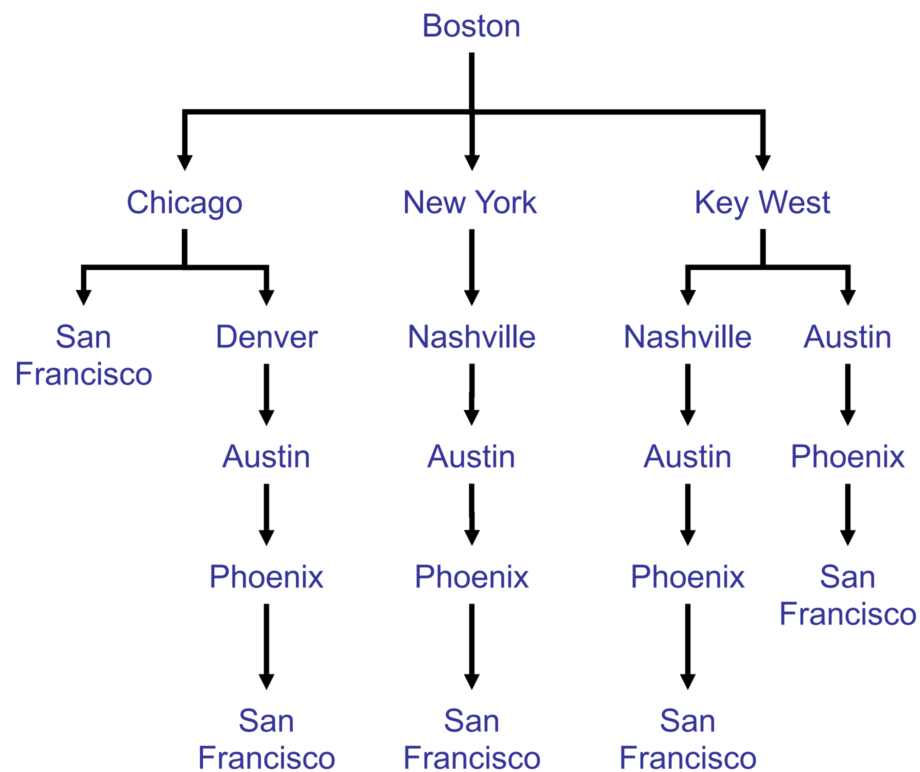
Learning from Observations

CPSC 470 – Artificial Intelligence

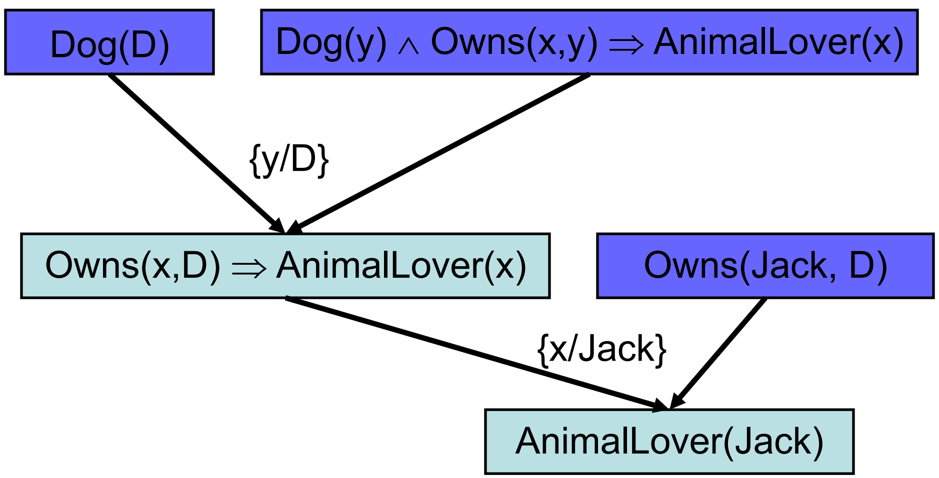
Brian Scassellati

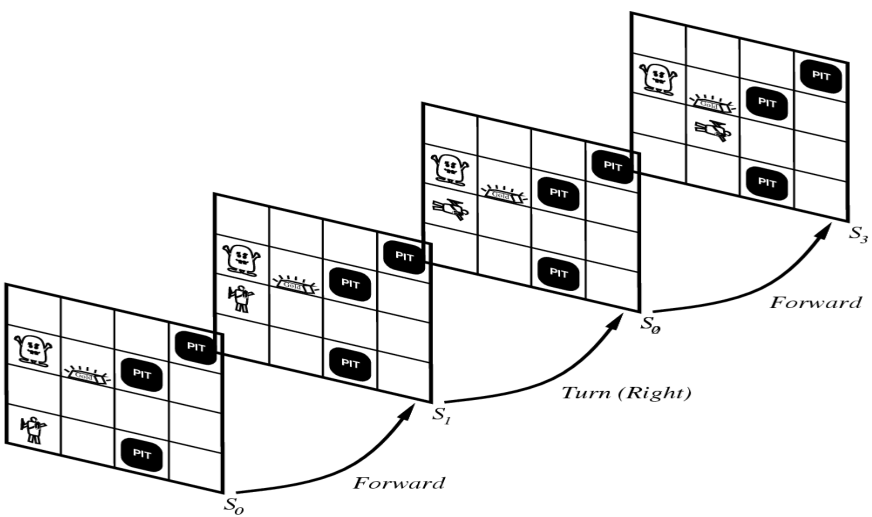
Problem Solving via Search



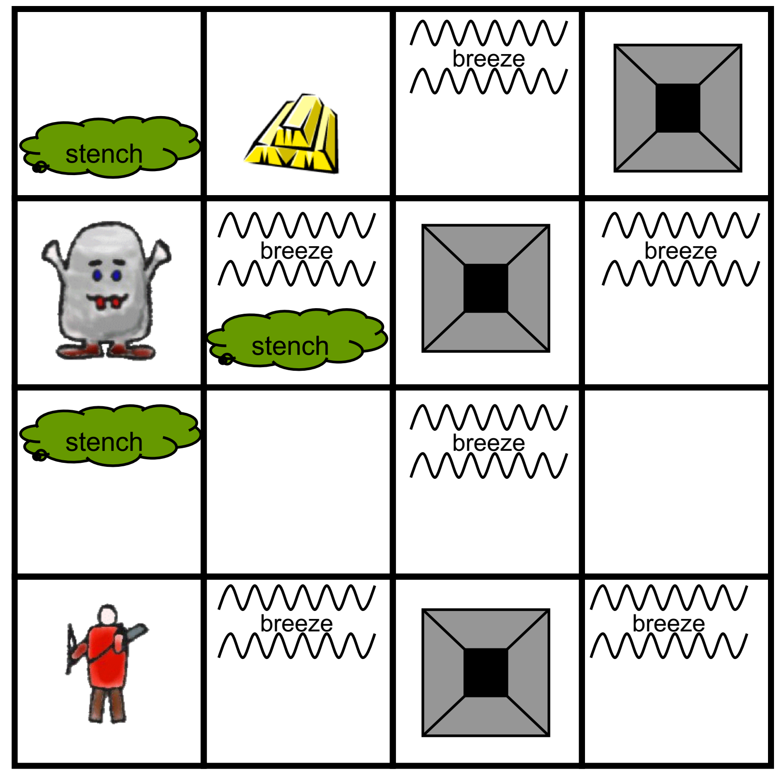


Knowledge Representation and Logical Reasoning

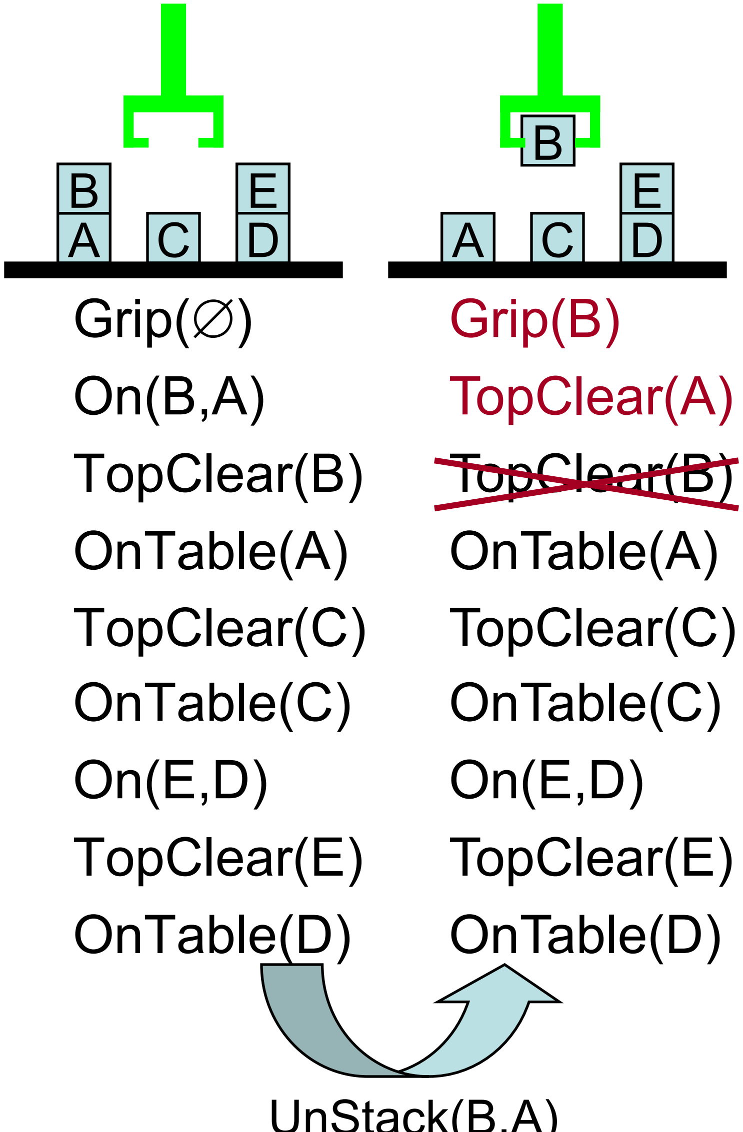


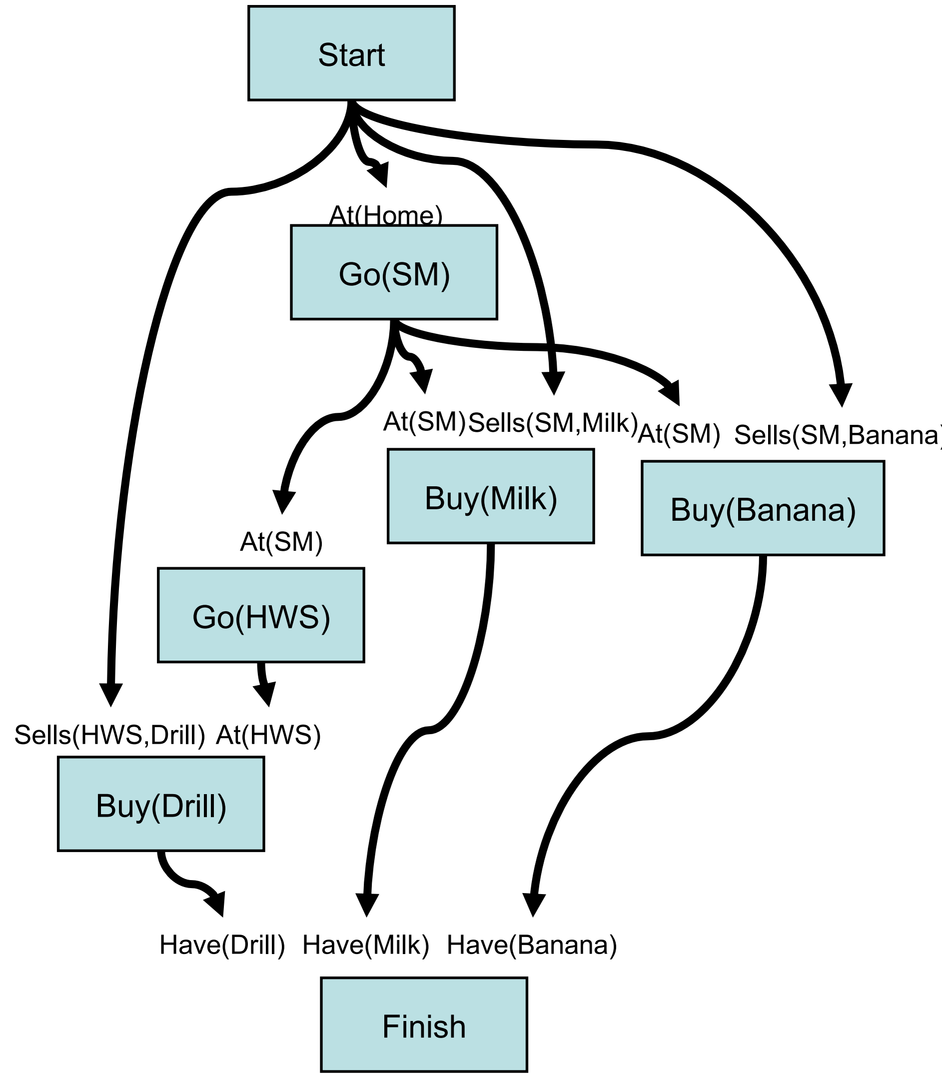


Wumpus World

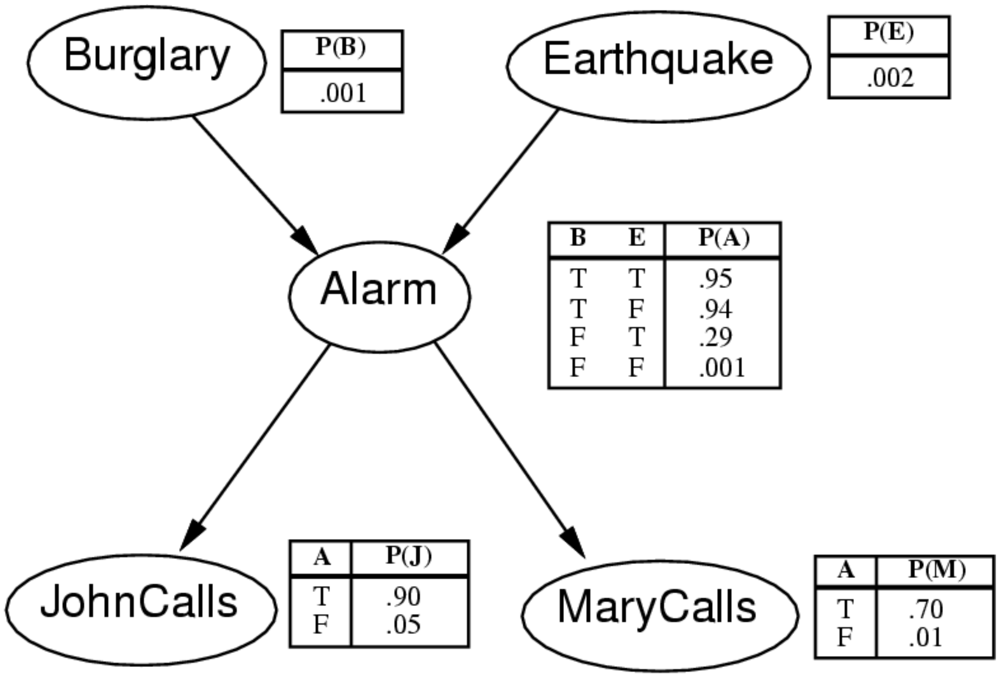


Planning


B
A
C
E
D
B
A
C
E
D
Grip(∅)
On(B,A)
TopClear(B)
OnTable(A)
TopClear(C)
OnTable(C)
On(E,D)
TopClear(E)
OnTable(D)
Grip(B)
TopClear(A)
~~TopClear(B)~~
OnTable(A)
TopClear(C)
OnTable(C)
On(E,D)
TopClear(E)
OnTable(D)
UnStack(B,A)


Start
At(Home)
Go(SM)
At(SM)
Sells(SM,Milk)
At(SM)
Sells(SM,Banana)
Buy(Milk)
Buy(Banana)
At(SM)
Go(HWS)
Sells(HWS,Drill)
At(HWS)
Buy(Drill)
Have(Drill)
Have(Milk)
Have(Banana)
Finish

Making Decisions under Uncertainty



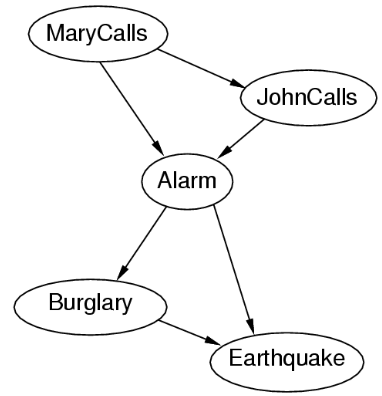
P(B)
.001

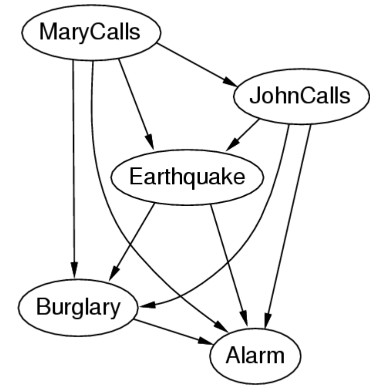
P(E)
.002

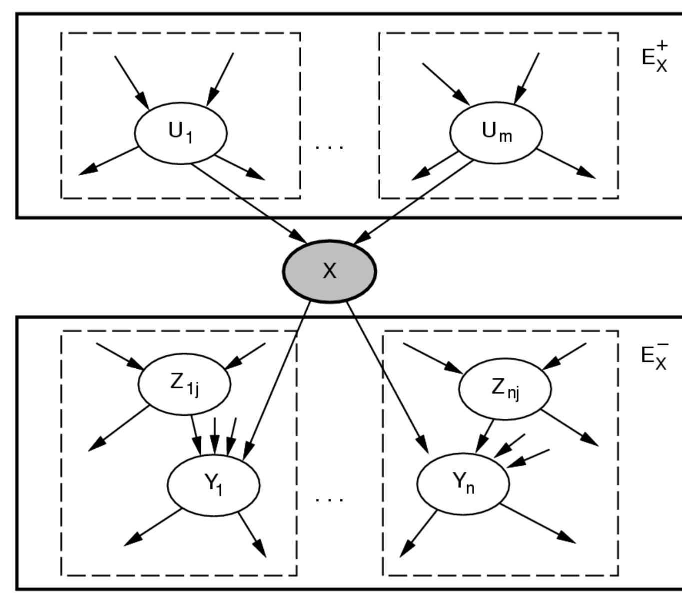
B	E	P(A)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J)
T	.90
F	.05

A	P(M)
T	.70
F	.01





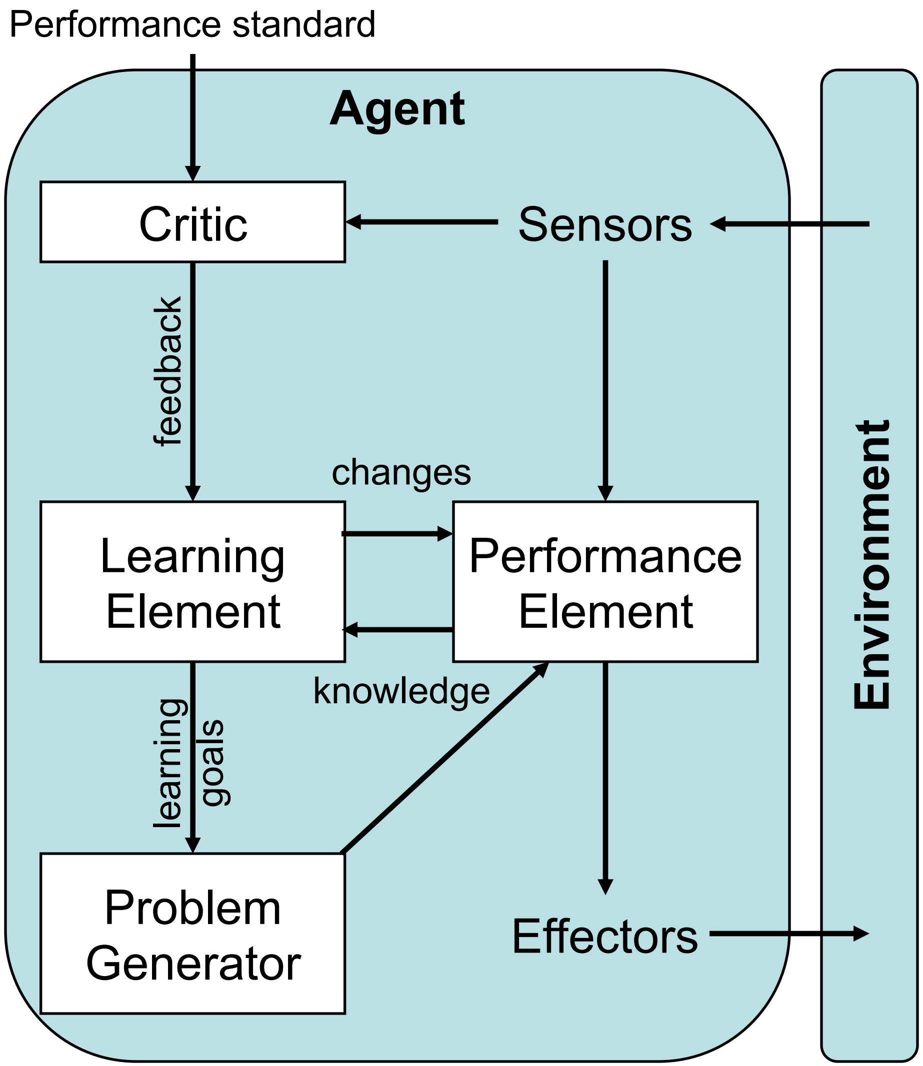


A conditional probability table gives the likelihood of a particular combination of values

Solving Problems

- How to *Do the Right Thing*™
  - Try all possibilities (search)
  - Build a Knowledge Base and Apply logical rules (inference)
- Dealing with the difficulties of the world
  - Dealing with uncertainty
  - Attempting to perform a plan
- What do you do when
  - You don’t know what the right answer really is
  - There are too many choices for search
  - Attempt to automatically learn the correct function

Parts of Learning Agents



- Performance element
  - Maps sensory states to actions (may use internal state, etc.)
- Learning element
  - Uses feedback to modify the performance element in order to improve future action selection
- Critic
  - Maps percepts to performance measures to provide feedback (optional)
- Problem generator
  - Suggests actions that will allow for better learning (optional)

Questions when Designing a Learning Agent

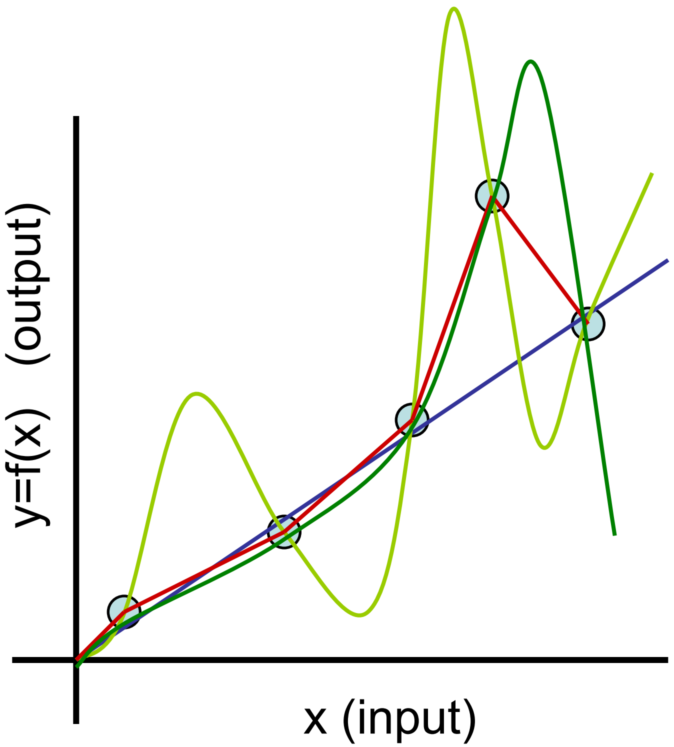
- Taxi-Driver Agent example
- Which components of the performance element are to be improved?
  - Steering angle, acceleration rules
  - Knowledge of road conditions
  - Navigation
- What representation is used for those components?
  - Polynomial function? Logical format? Search tree?
- What feedback is available?
  - Instructor? Honking horns? Crashes?
- What prior information is available?
  - First time behind the wheel, drove three years ago or drove yesterday?
  - Did we have a driving course or read a manual?

The Machine Learning Toolbox

Feedback is Critical

- Supervised learning: when an error occurs, agent receives the correct output
- Reinforcement learning: when an error occurs, agent receives an evaluation of its output, but is not told the correct output
- Unsupervised learning: no indication is given whether an output was correct or incorrect

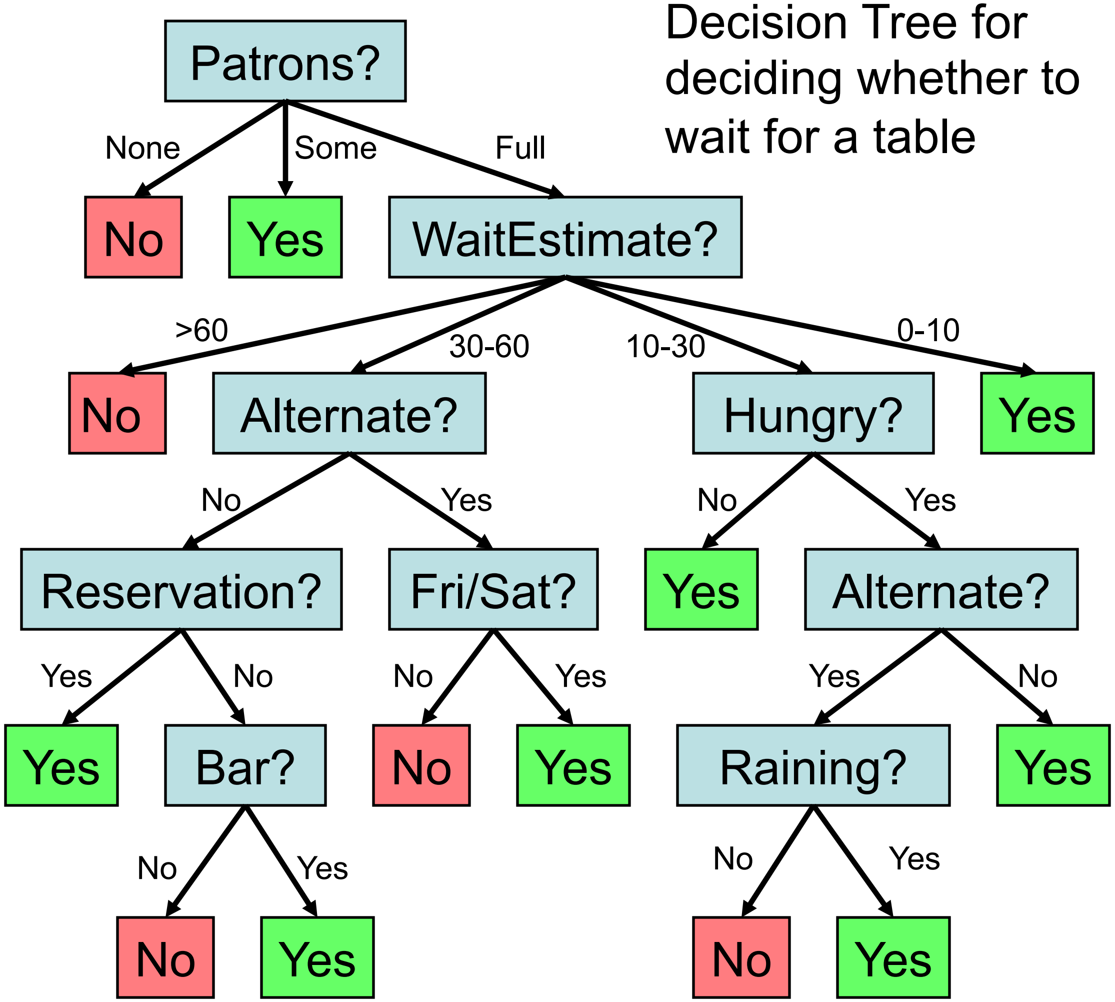
Inductive Learning



- A form of supervised learning
- Output is some function of the input

  y=f(x)
- Examples are samples of the function f
- Hypothesis (h) is an estimate of f
- Many hypotheses are possible
- Bias is a preference of one hypothesis over another

Decision Trees



Decision Tree for deciding whether to wait for a table

- Represents a Boolean function (the goal predicate WillWaitForTable())
- Internal nodes are tests of a feature/property
- Leaves are Boolean values
- Represent a propositional logic statement
  - Each path could be a line in a truth table

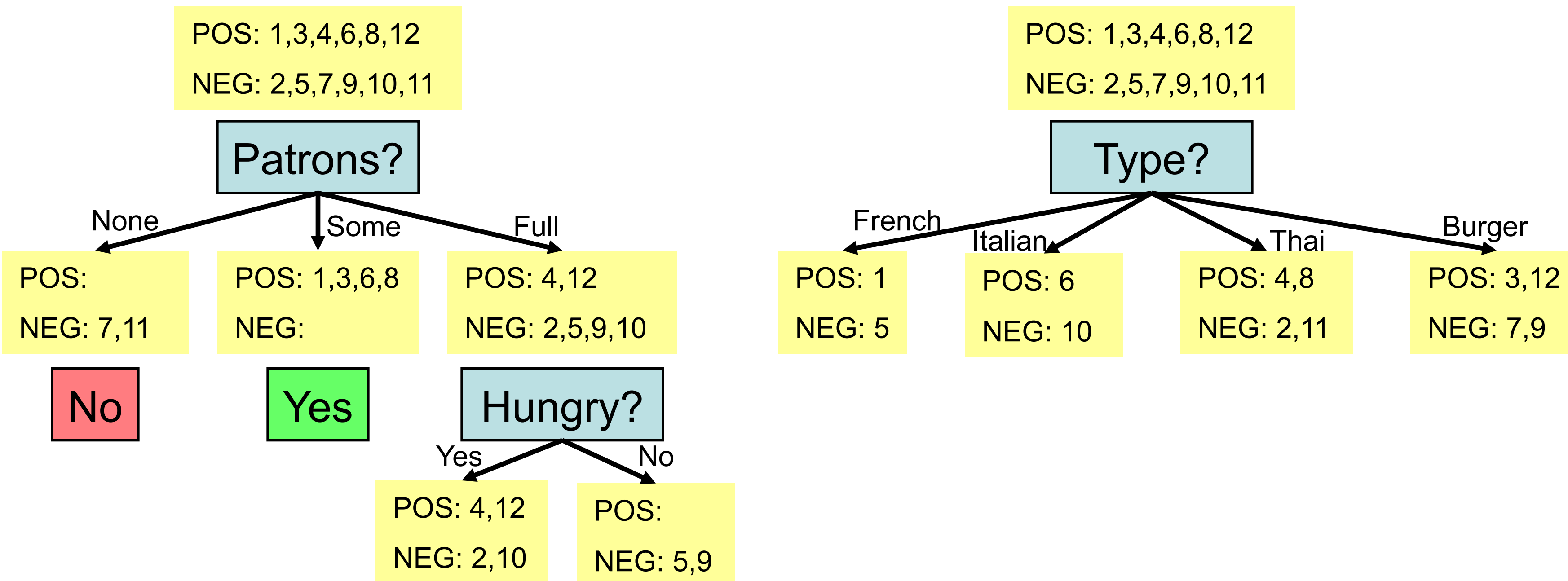
Inducing Decision Trees from Examples

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	*Yes*
X_2	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	*No*
X_3	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	*Yes*
X_4	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*10–30*	*Yes*
X_5	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	*No*
X_6	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	*Yes*
X_7	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	*No*
X_8	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	*Yes*
X_9	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	*No*
X_{10}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	*No*
X_{11}	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	*No*
X_{12}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	*Yes*

Goal Predicate
(classification)

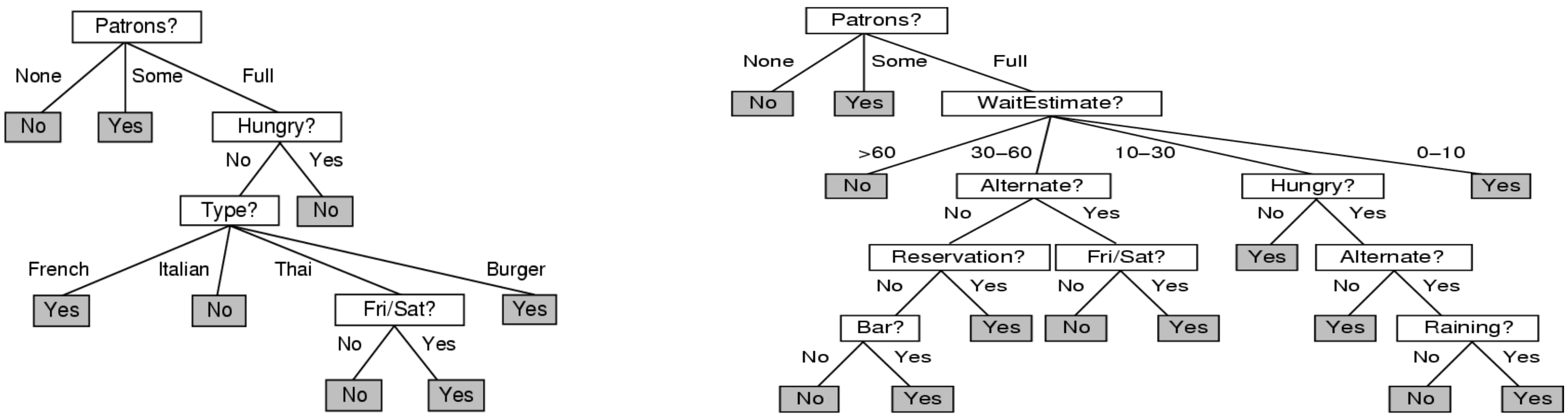
- A set of examples is a training set
  - Positive examples (YES result) and negative examples (NO result)
- A trivial solution:
  - Build a tree that has one path for each example
- Ockham's Razor
  - The most likely hypothesis is the simplest one that is consistent with the data

Guidelines for Finding a Small Decision Tree



- Test the most important feature first
- If you have only one type of example, return a leaf
- Else, choose the next most important feature
- If you run out of examples, return a default value (no data)
- If you run out of features, you are in trouble (two examples have same description: noisy data)

Decision Tree Learning



- Following this algorithm, we generate the tree at left
- But the examples were generated by the agent acting on the original tree at right
- There is nothing wrong with the learning algorithm…
  - The algorithm generates a hypothesis that matches the examples, not necessarily the underlying function
    - May be considerably simpler
    - May uncover unexpected regularities

How do you Determine which Feature is Better?

- Information Theory!
- Information is measured in bits

$$I(P(v_1),\ldots,P(v_n)) = \sum_{i=1}^{n} -P(v_i)\log_2 P(v_i)$$

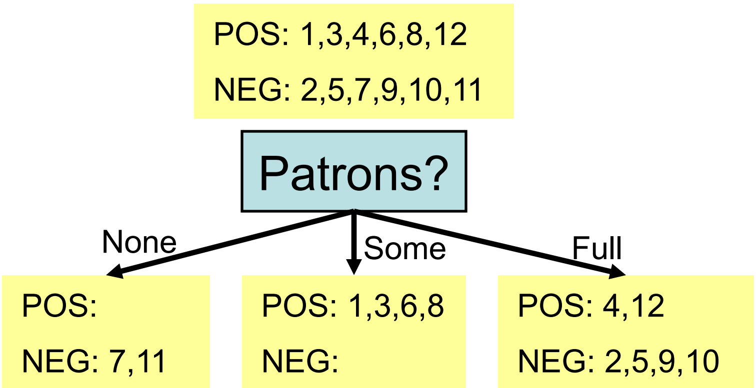
- Flipping a fair coin gives one bit of information

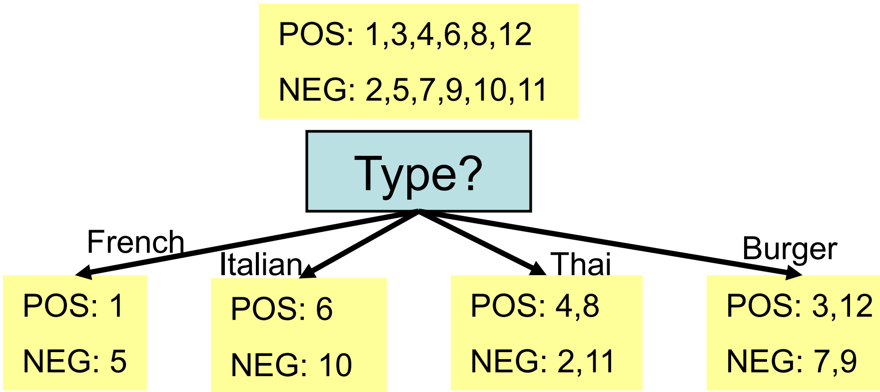
$$I(\tfrac{1}{2},\tfrac{1}{2}) = -\tfrac{1}{2}\log_2\tfrac{1}{2} - \tfrac{1}{2}\log_2\tfrac{1}{2} = \tfrac{1}{2} + \tfrac{1}{2} = 1\text{ bit}$$

- After we make a choice, we still need additional info to make the correct choice

$$Remainder(A) = \sum_{i=1}^{v} \tfrac{p_i+n_i}{p+n} I(\tfrac{p_i}{p_i+n_i},\tfrac{n_i}{p_i+n_i})$$

Example of Information Content





$$Remainder(A) = \sum_{i=1}^{v} \frac{p_i+n_i}{p+n} I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

$$I(P(v_1),\ldots,P(v_n)) = \sum_{i=1}^{n} -P(v_i)\log_2 P(v_i)$$

$$Remainder(Patrons) = \tfrac{2}{12} I(0,1) + \tfrac{4}{12} I(1,0) + \tfrac{6}{12} I(\tfrac{2}{6}, \tfrac{4}{6})$$

$$Remainder(Patrons) \approx 0 + 0 + \tfrac{6}{12}(-\tfrac{2}{6}\log\tfrac{2}{6} - \tfrac{4}{6}\log\tfrac{4}{6})$$

$$Remainder(Patrons) \approx 0.459 \text{ bits}$$

Needs less info to make a perfect choice

$$Remainder(Type) = \tfrac{2}{12} I(\tfrac{1}{2}, \tfrac{1}{2}) + \tfrac{2}{12} I(\tfrac{1}{2}, \tfrac{1}{2}) + \tfrac{4}{12} I(\tfrac{2}{4}, \tfrac{2}{4}) + \tfrac{4}{12} I(\tfrac{2}{4}, \tfrac{2}{4})$$

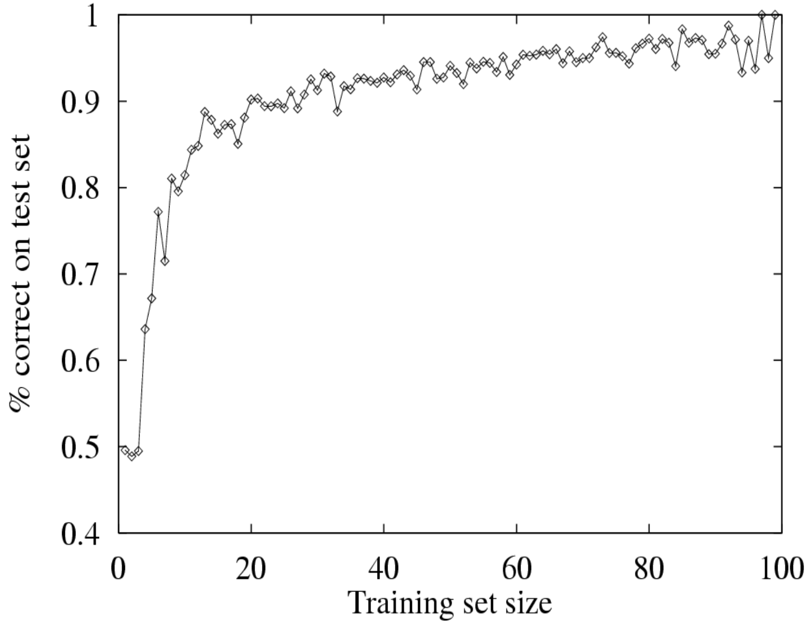
$$Remainder(Type) = \tfrac{2}{12}(1) + \tfrac{2}{12}(1) + \tfrac{4}{12}(1) + \tfrac{4}{12}(1)$$

$$Remainder(Type) = 1$$

Practical Examples of Decision Trees

- Designing oil platform equipment
  - GASOIL (1986 – BP)
  - Designing gas-oil separation systems for offshore platforms
  - 2500 rules
  - Would have taken 10 person-years to build by hand
  - Decision tree took 100 person-days to implement and train
- Learning to Fly
  - C4.5 (1992 - Sammut et al.)
  - Cessna on a flight simulator
  - Observe 3 human pilots make 30 assigned flights
  - Create training example every time a control is touched
  - Flies better than the human instructors!
    - (allows generalization across errors)

Assessing the Performance of a Learning Algorithm



- Divide the examples into a training set and a test set
  - Determine the percentage of examples in each set
  - Randomly select examples for each
- Vary the percentage
- Plot this data as a learning curve

Limits of Decision Trees

- Propositional logic limits
  - Difficult to express existential quantifiers
  - But can be done by defining new operators
- May be exponentially large (i.e. Parity function)
  - Consider a function with n features/attributes
    - 2^n rows in a truth table (can define a function with 2^n bits)
    - 2^{2^n} different functions
    - For example, with n=6 then $2^{2^n} = 2 \times 10^{19}$
- Needed extensions
  - Missing data attributes (what if you don't know all the relevant features?)
  - Multi-valued attributes (if there are too many choices, the information content gives an irrelevant measure)
  - Continuous-valued attributes (height, weight)

Noise and Overfitting

- In the presence of noise, some feature vectors will have multiple examples with conflicting results
- If there are many possible hypotheses, you must be careful to avoid finding meaningless “regularity” in the data (overfitting)
  - Every time I flip a coin with my left hand it comes up heads
  - I always encounter less traffic on Mondays (but I’m always late on Mondays)
- There are techniques for dealing with overfitting, but they rely on domain information

Administrivia

- Coming up next:
  - Wednesday: Guest Lecture (Marynel Vazquez)
  - Friday
    - Supervised Learning: Neural Networks
  - Friday 3:30-4:30 in Davies
    - Q&A session. Email questions by Thursday at 9pm
  - Monday
    - Midterm exam

Midterm Exam

- Monday, during class.
  - CS470: Report to Davies
  - CS570: Report to ML 211
- 50 minute exam (10:30-11:20). Do not be late.
- No calculators, textbooks, notes, phones, or computers
- You MAY bring one 8.5x11in sheet of paper
- Coverage:
  - Lectures up to and including 2/22 (Uncertainty)
  - Problem sets #0-4 (inclusive)
  - Reading up to and including 2/22 (CH 14)
  - NOT including Motion planning or CH 25