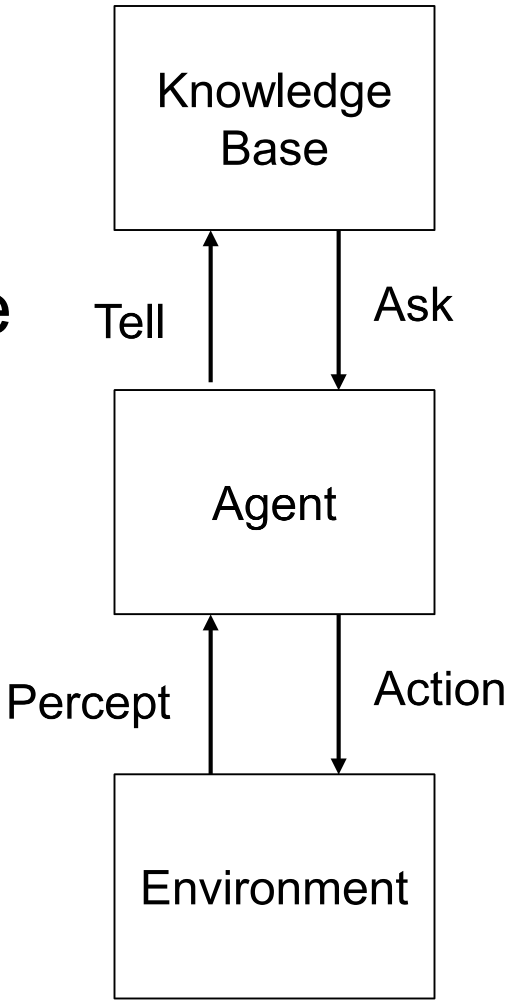
CS331 Overview II

Logical Agents

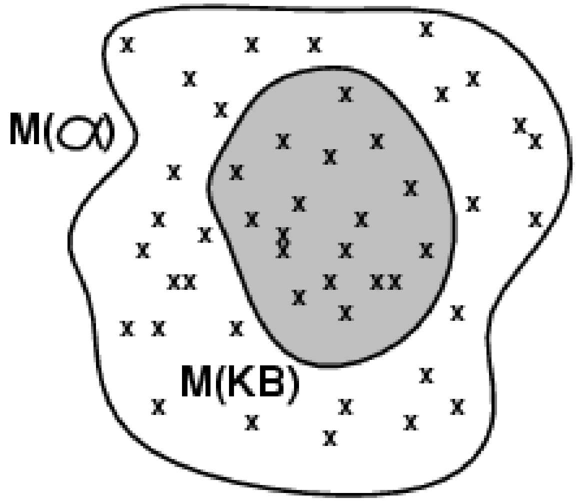
Knowledge-based Agents

- A knowledge base encodes knowledge about the world
- The agent communicates with the knowledge base thru Ask-Tell Interface
  - Tell(*KB*, Make-Percept-Sentence(percept,*t*)
  - Action := Ask(KB, Make-action-query(*t*))
  - Tell(KB,Make-Action-Sentence(action,*t*))

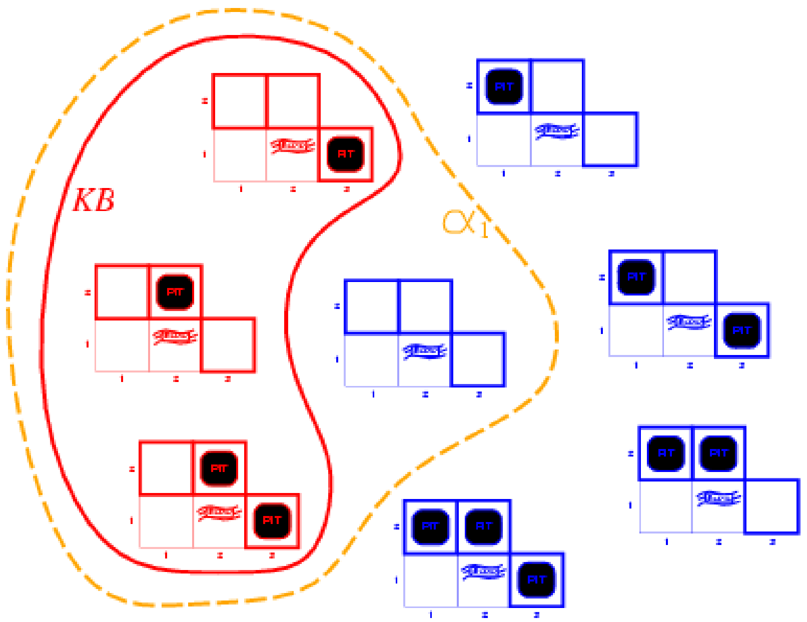


What is Logic?

- Syntax – says what sentences are legal
- Semantics – say what the sentences mean. Define whether a sentence is true or false in each ``possible world'' or "model."
  - `X+Y = 2' is true in a model where X=1 and Y=1.
  - False in a model where X=1 and Y=2.
  - We say *m* is ``a model of'' a sentence α if α is true in *m*. Let M(α) be models of α.
- Entailment
  - KB ⊨ α if and only if M(KB) ⊆ M(α)
  - {X=1,Y=X} ⊨ X+Y = 2

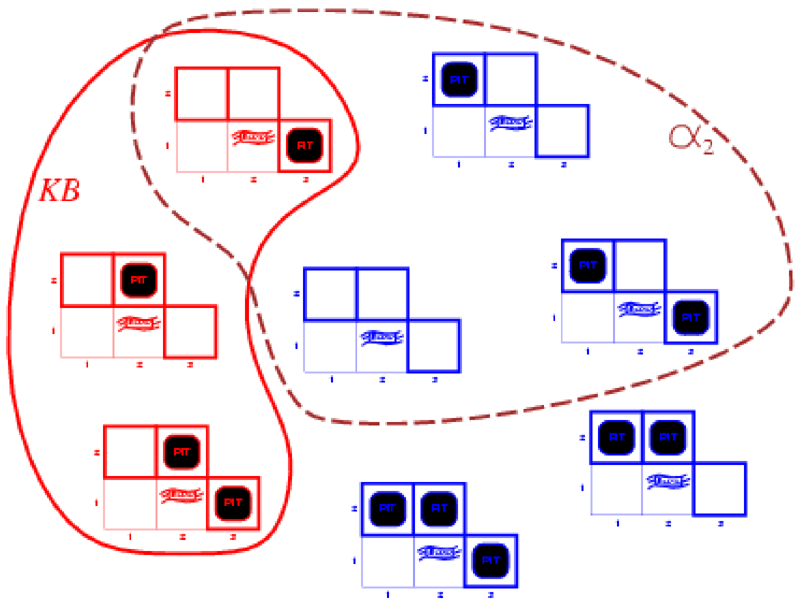


Wumpus models



- *KB* = wumpus-world rules + observations
- α_1 = "[1,2] is safe", *KB* $\models \alpha_1$, proved by model checking

Wumpus models



- *KB* = wumpus-world rules + observations
- α_2 = "[2,2] is safe", *KB* $\not\models \alpha_2$

Truth tables for connectives

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

Wumpus world sentences

Let $P_{i,j}$ be true if there is a pit in [i, j].

Let $B_{i,j}$ be true if there is a breeze in [i, j].

Let Percepts be:

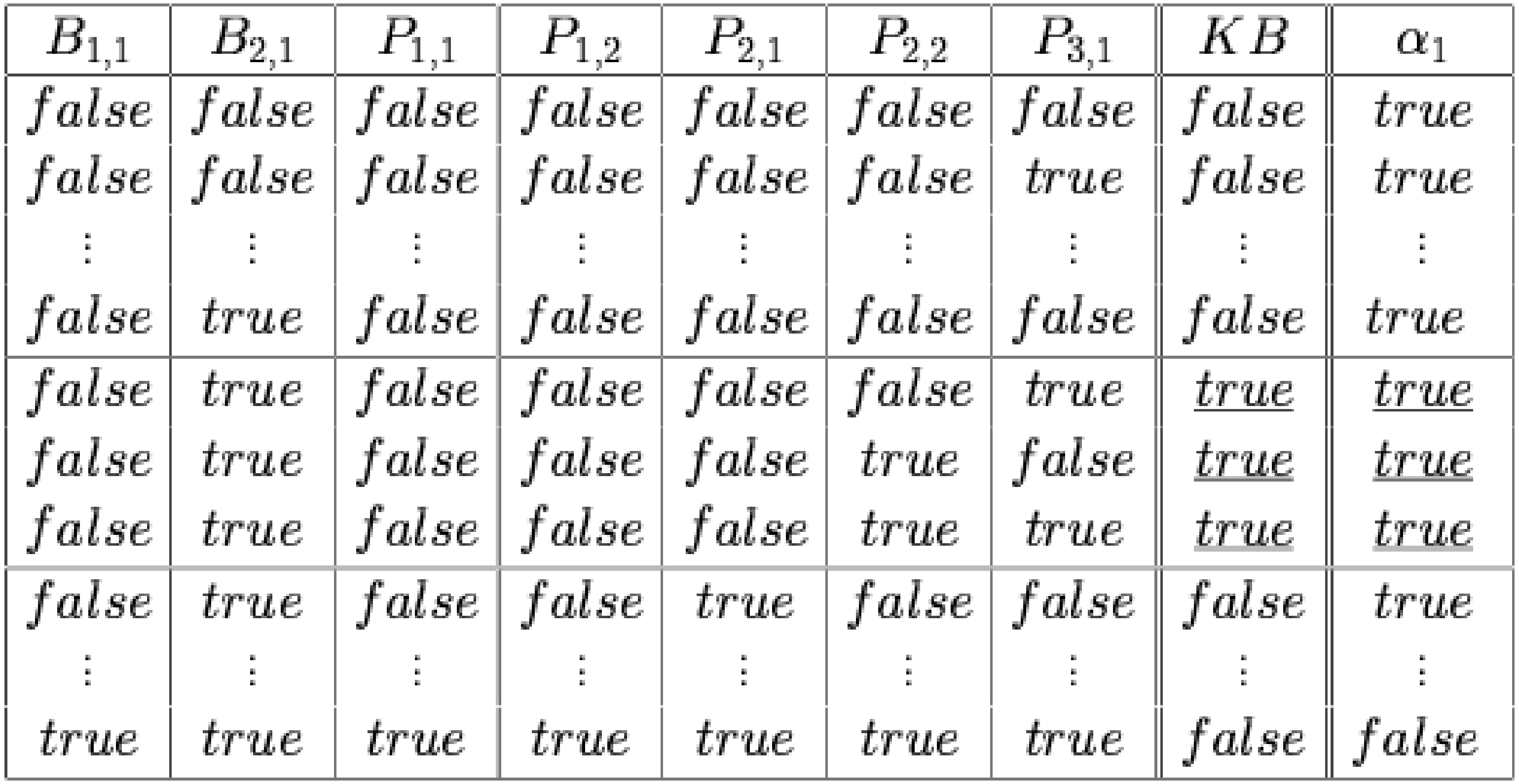
$\neg P_{1,1}$

$\neg B_{1,1}$

$B_{2,1}$

- "Pits cause breezes in adjacent squares (and nothing else does)"
- $(P_{1,2} \vee P_{2,1}) \Leftrightarrow B_{1,1}$
- $(P_{1,1} \vee P_{2,2} \vee P_{3,1}) \Leftrightarrow B_{2,1}$

Truth tables for inference



$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB	α_1
false	false	false	false	false	false	false	false	true
false	false	false	false	false	false	true	false	true
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	true	false	false	false	false	false	false	true
false	true	false	false	false	false	true	true	true
false	true	false	false	false	true	false	true	true
false	true	false	false	false	true	true	true	true
false	true	false	false	true	false	false	false	true
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	true	true	true	true	true	true	false	false

Inference by Enumeration: builds this table row by row using depth first search and returns **true** if α_1 is true whenever KB is true.

Propositional Theorem Proving

- Can we avoid enumeration of all possible models to show $KB \models \alpha$?
- It seems we can: $\{(P_{1,2} \vee P_{2,1}) \Leftrightarrow B_{1,1}; P_{1,2}\} \models B_{1,1}$
- Theorem proving or Inference
  - Applies a sequence of <u>inference rules</u> (proof) to true sentences
  - Sound inference rules guarantee that if the premises of the rule are true, the conclusion is true
- But which sequence of inference rules?
- **Search Problem**: Initial state = KB, Actions = Inference Rules, Goal condition = KB includes α

Validity and satisfiability

A sentence is valid if it is true in all models,

e.g., *True*, $A \vee \neg A$, $A \Rightarrow A$, $(A \wedge (A \Rightarrow B)) \Rightarrow B$

Validity is connected to inference via the Deduction Theorem:

$KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

A sentence is satisfiable if it is true in some model

e.g., $A \vee B$, C

A sentence is unsatisfiable if it is true in no models

e.g., $A \wedge \neg A$

Satisfiability is connected to inference via the following:

$KB \models \alpha$ if and only if $(KB \wedge \neg \alpha)$ is unsatisfiable

Logical equivalence

Two sentences are *logically equivalent* iff true in same models: $\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$

$(\alpha \land \beta) \equiv (\beta \land \alpha)$	commutativity of $\land$
$(\alpha \lor \beta) \equiv (\beta \lor \alpha)$	commutativity of $\lor$
$((\alpha \land \beta) \land \gamma) \equiv (\alpha \land (\beta \land \gamma))$	associativity of $\land$
$((\alpha \lor \beta) \lor \gamma) \equiv (\alpha \lor (\beta \lor \gamma))$	associativity of $\lor$
$\neg(\neg\alpha) \equiv \alpha$	double-negation elimination
$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \lor \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \land \beta) \equiv (\neg\alpha \lor \neg\beta)$	de Morgan
$\neg(\alpha \lor \beta) \equiv (\neg\alpha \land \neg\beta)$	de Morgan
$(\alpha \land (\beta \lor \gamma)) \equiv ((\alpha \land \beta) \lor (\alpha \land \gamma))$	distributivity of $\land$ over $\lor$
$(\alpha \lor (\beta \land \gamma)) \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma))$	distributivity of $\lor$ over $\land$

Some Sound Inference Rules

- And Introduction: $\frac{\alpha,\beta}{\alpha \land \beta}$
- Modus Ponens: $\frac{\alpha,\alpha \Rightarrow \beta}{\beta}$
- Implication Elimination: $\frac{\alpha \Rightarrow \beta}{\neg\alpha \lor \beta}$
- Unit Resolution: $\frac{\neg\alpha \lor \beta,\ \alpha}{\beta}$
- Resolution: $\frac{\neg\alpha \lor \beta,\ \alpha \lor \gamma}{\beta \lor \gamma}$
- Biconditional Elimination: $\frac{\alpha \Leftrightarrow \beta}{(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)}$

Conversion to CNF (And of Or's)

$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
   $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

3. Move $\neg$ inwards using de Morgan's rules and double-negation:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

4. Apply distributive law ($\wedge$ over $\vee$) and flatten:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

Resolution example

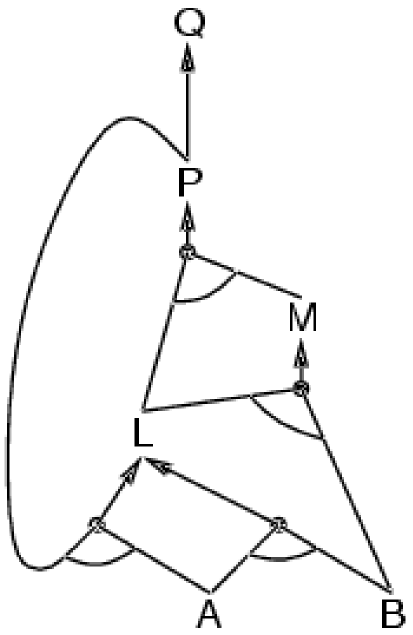
- *KB* = $(B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1}))$; $\neg B_{1,1}$
- $\alpha = \neg P_{1,2}$
- *Show KB* $\models \alpha$.
- Add $\neg\alpha$ to *KB*.
- Convert *KB* to CNF.
- Apply Resolution to derive a contradiction.

$$\begin{tikzcd}
\boxed{\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}} & \boxed{\neg P_{1,2} \vee B_{1,1}} \arrow[dr] & \boxed{\neg P_{2,1} \vee B_{1,1}} & \boxed{\neg B_{1,1}} \arrow[dl] & \boxed{P_{1,2}} \arrow[ddl] \\
 & & \boxed{\neg P_{1,2}} \arrow[dr] & & \\
 & & & \square &
\end{tikzcd}$$

Forward Chaining for Horn KBs

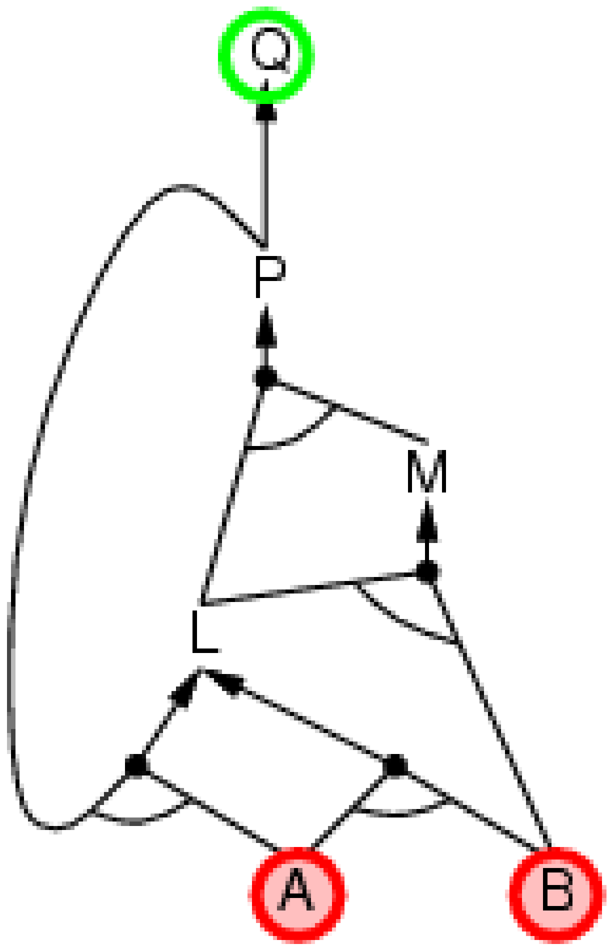
- Do until query is in the KB
  - Fire any rule whose premises are satisfied in the *KB*
  - Add its conclusion to the *KB*

$$\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}$$

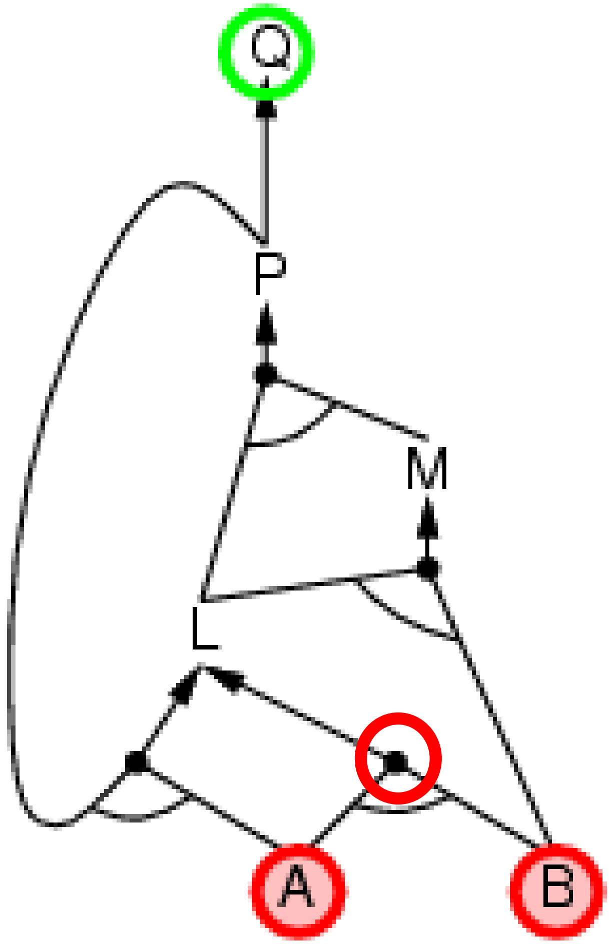


- Forward chaining is sound and complete for Horn clauses

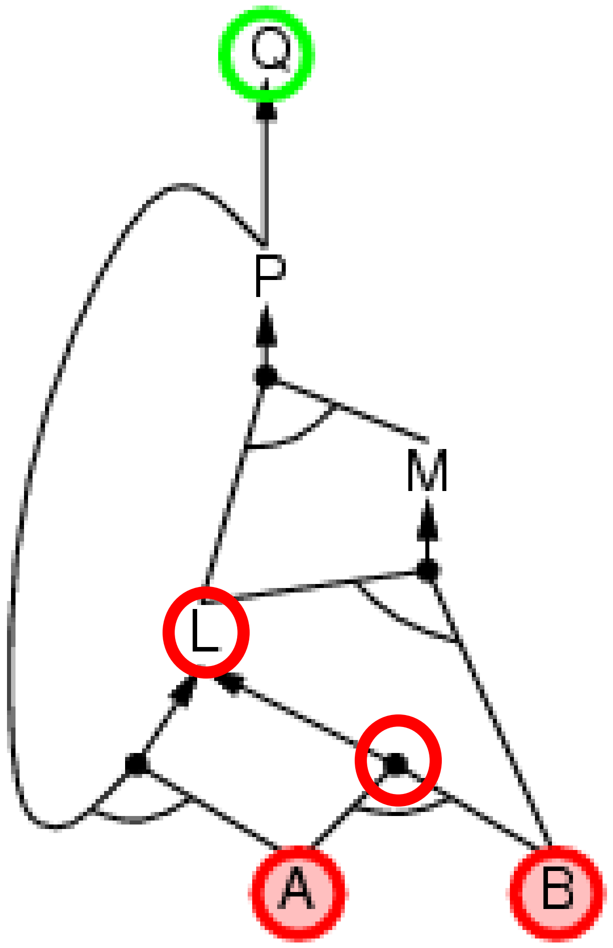
Forward chaining example



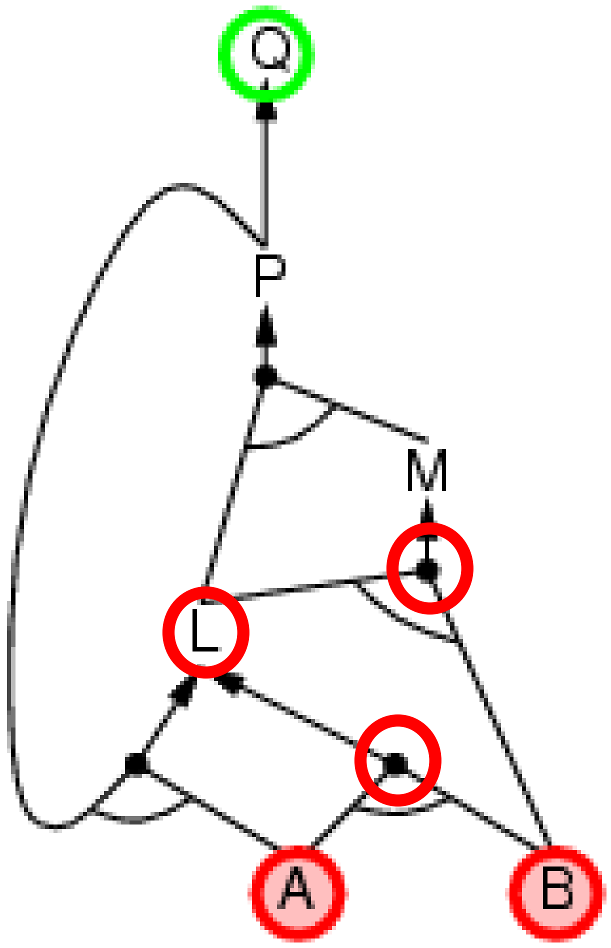
Forward chaining example



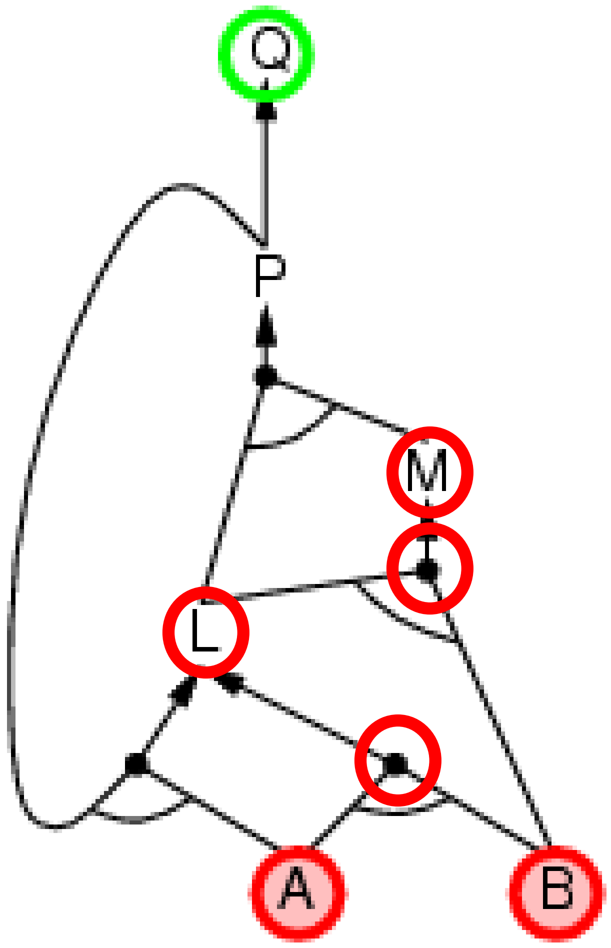
Forward chaining example



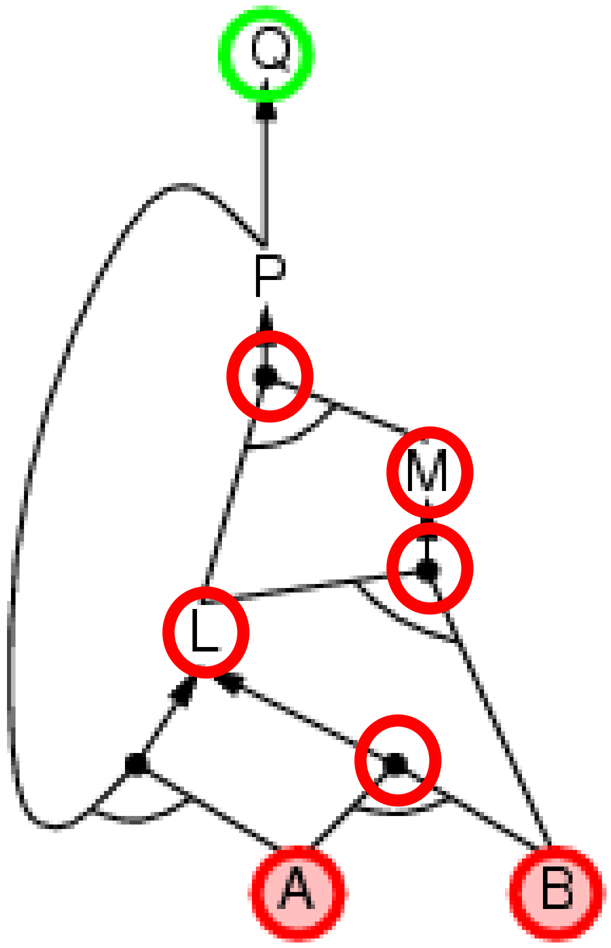
Forward chaining example



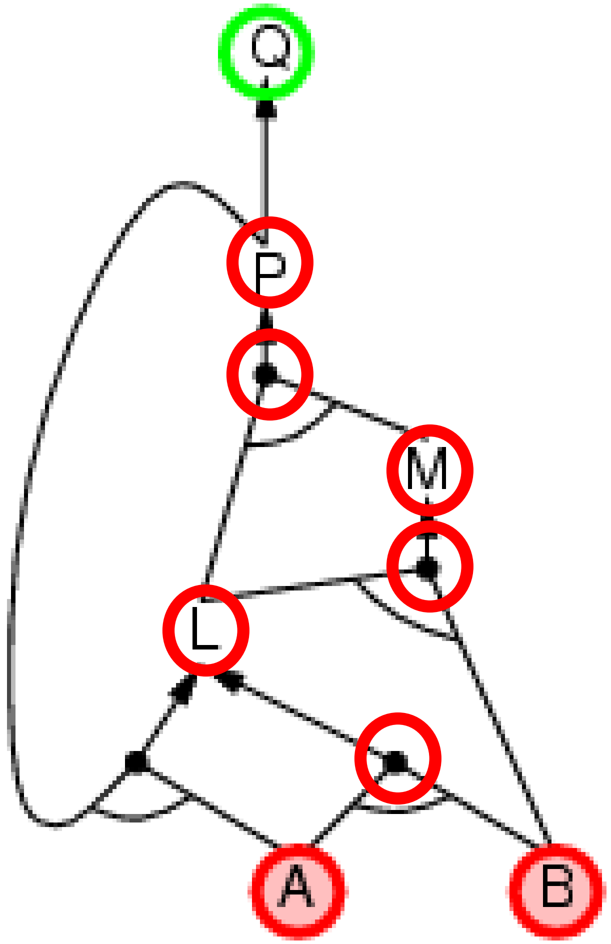
Forward chaining example



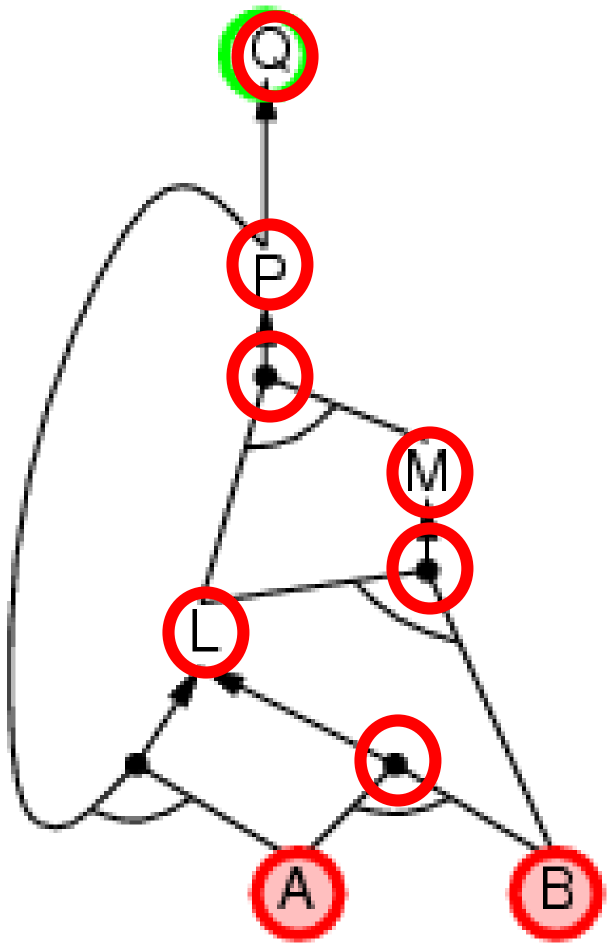
Forward chaining example



Forward chaining example



Forward chaining example

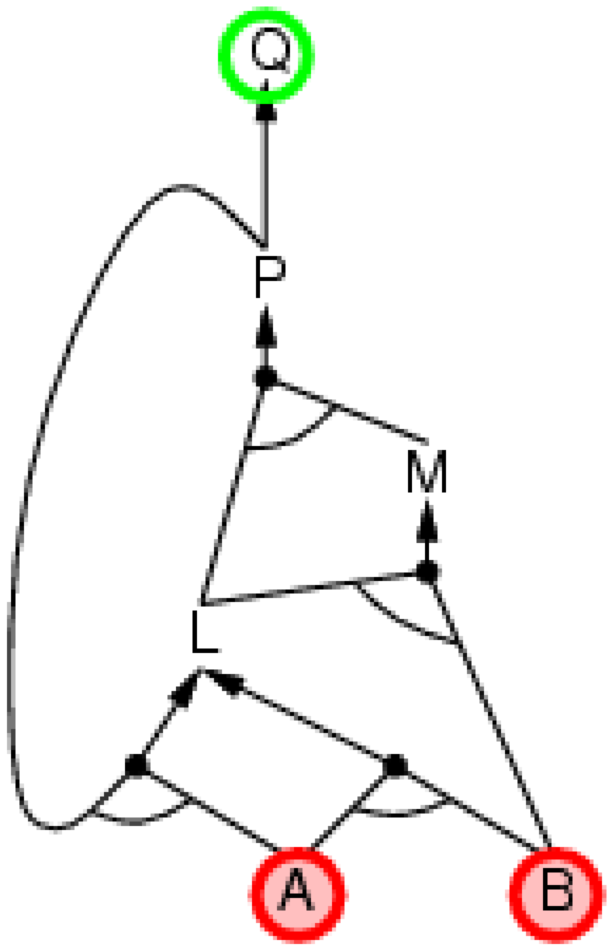


Backward Chaining

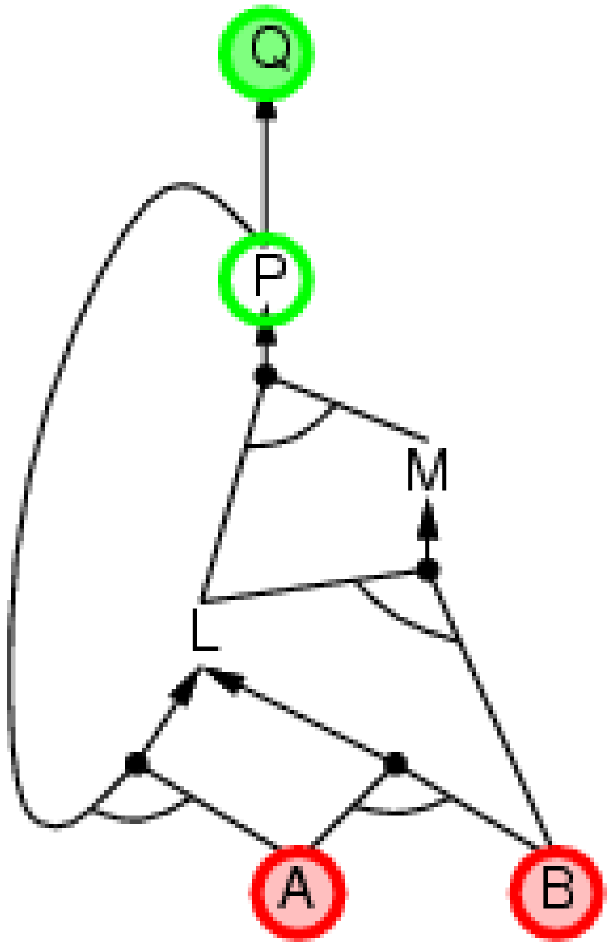
- Identical to And/Or Graph Search
- Idea: work backwards from the query *q*:

  to prove *q* by BC,
  - check if *q* is known already, or
  - prove by BC all premises of some rule concluding *q*
- Avoid loops: check if new subgoal is already on the goal stack
- Avoid repeated work (what makes this *graph* search!)
  - If new subgoal has already been proved true go to the next subgoal
  - If it has already failed return fail

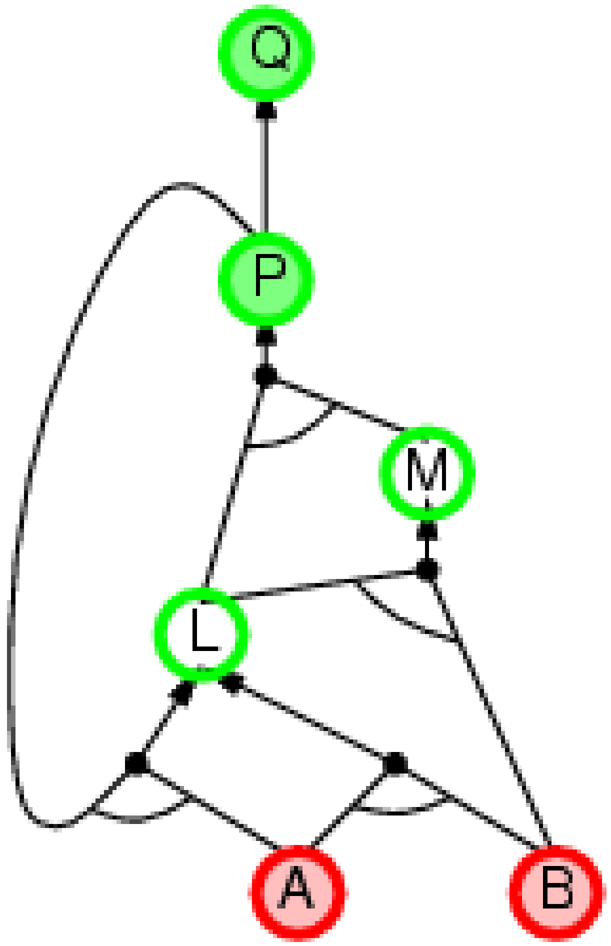
Backward chaining example



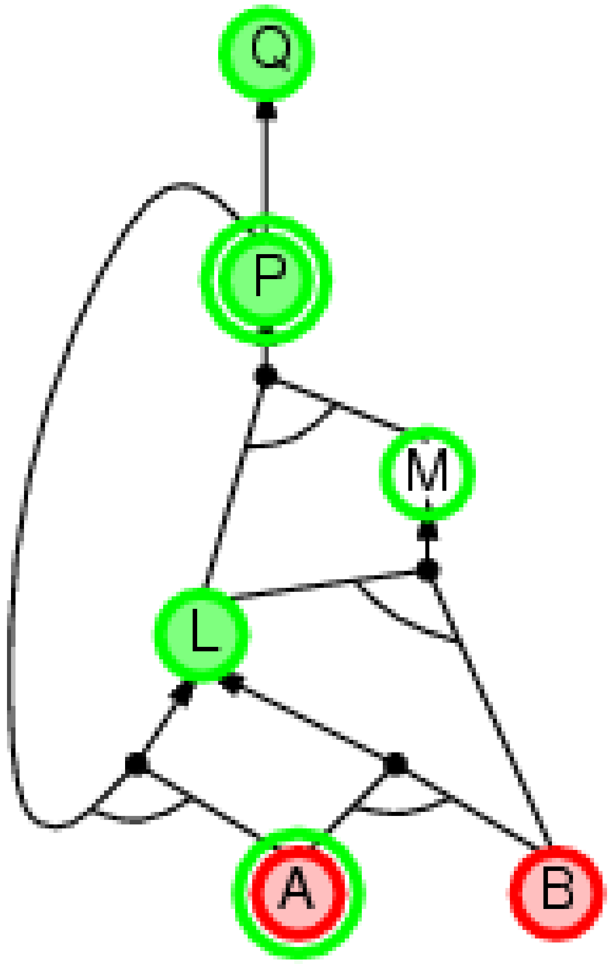
Backward chaining example



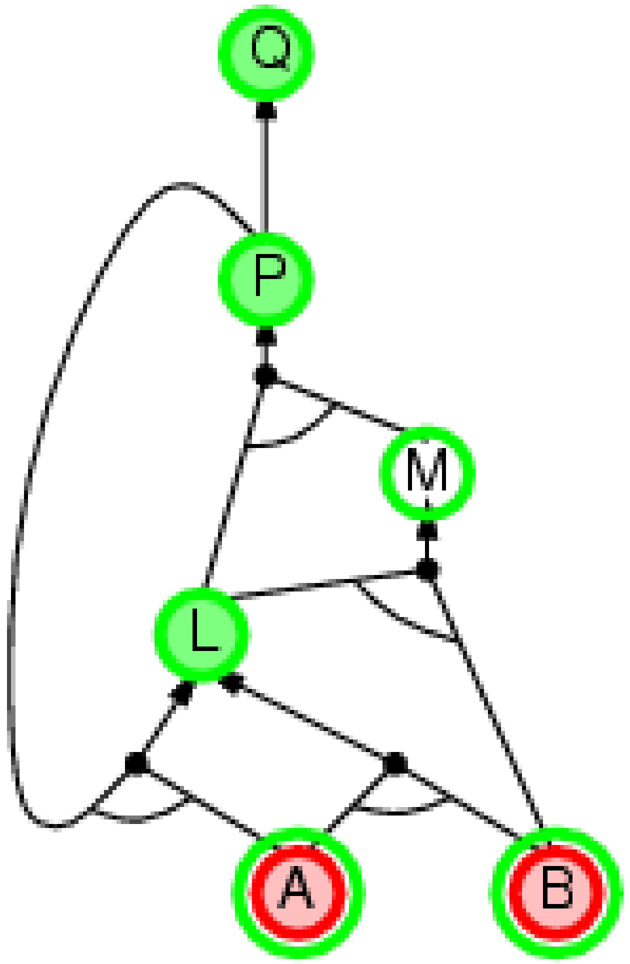
Backward chaining example



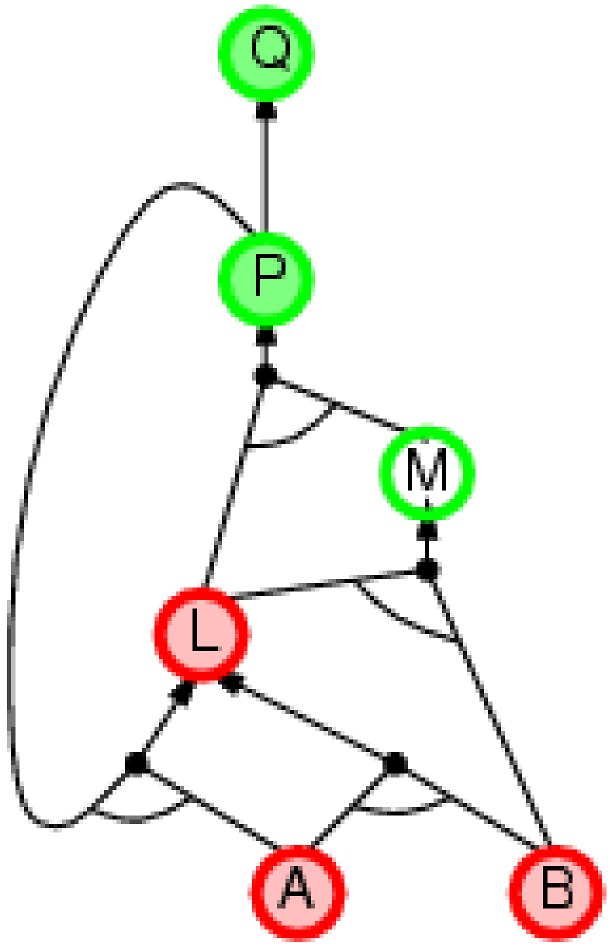
Backward chaining example



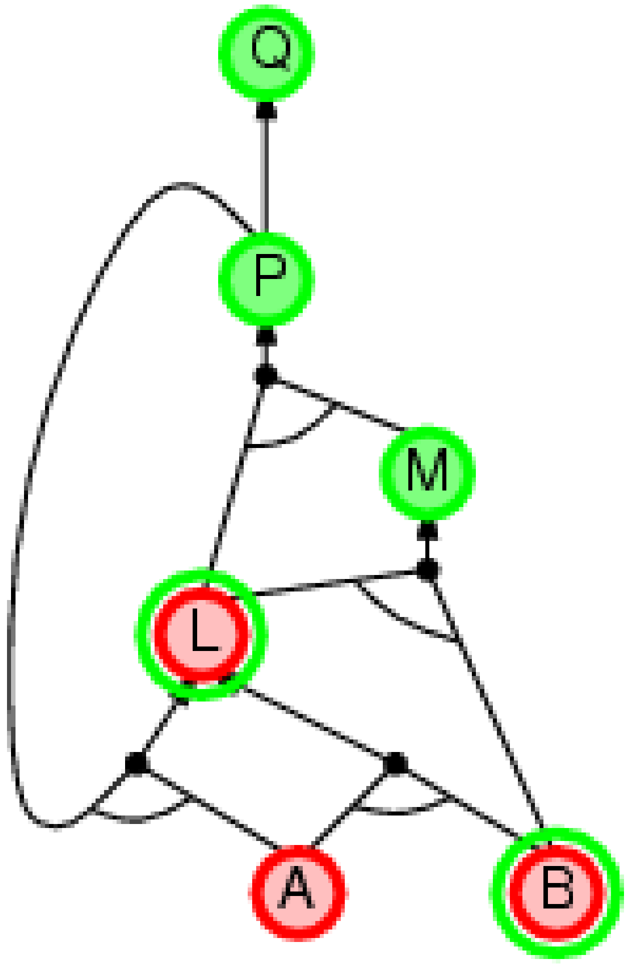
Backward chaining example



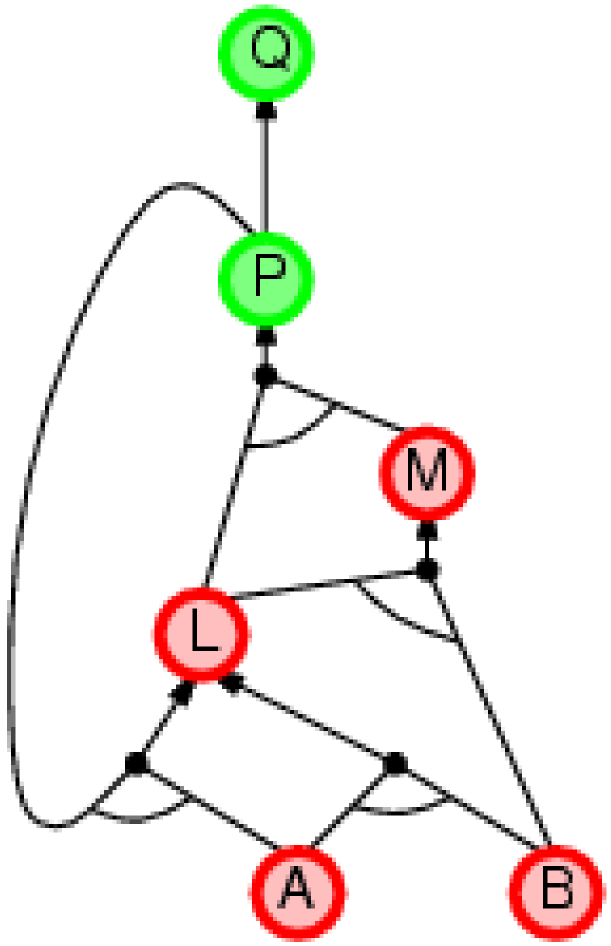
Backward chaining example



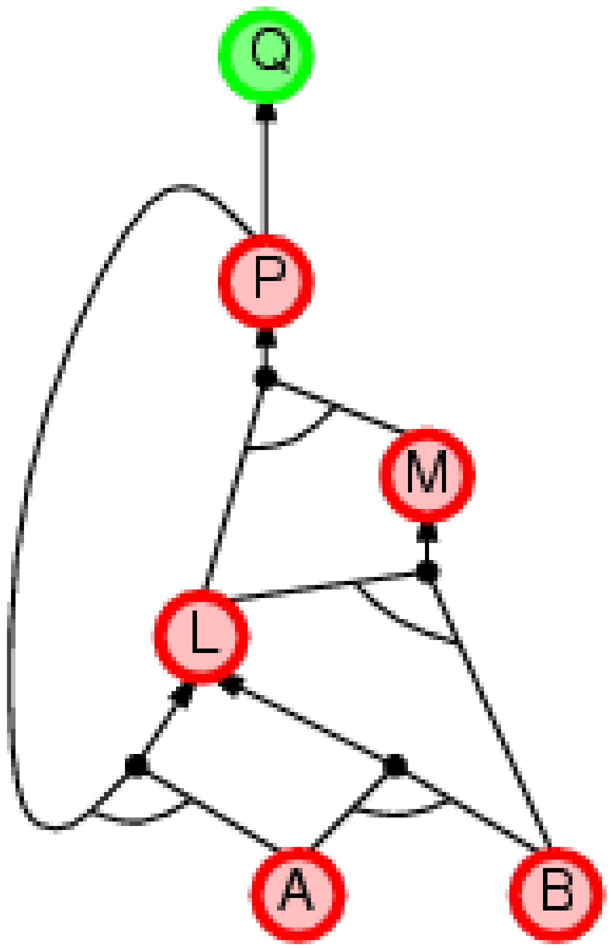
Backward chaining example



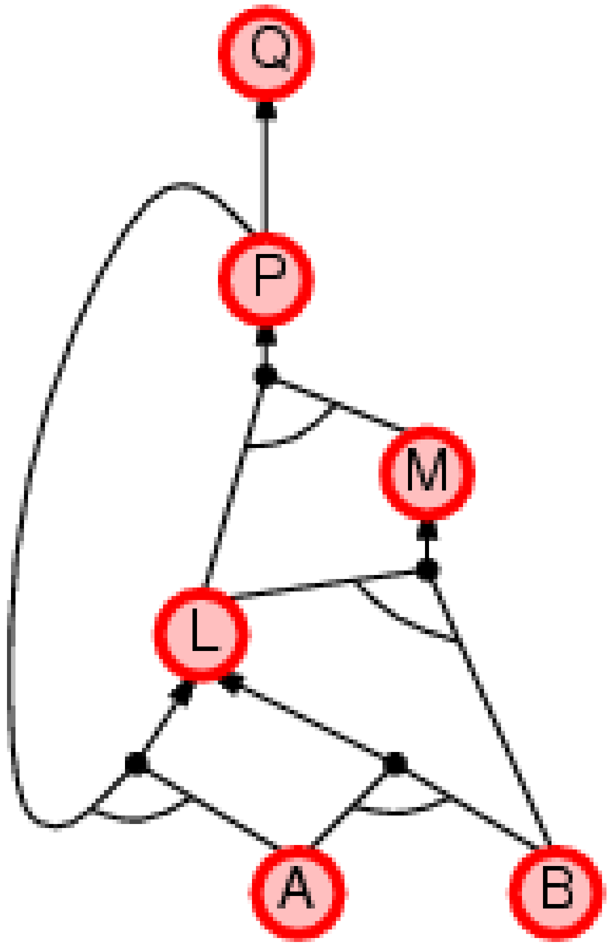
Backward chaining example



Backward chaining example



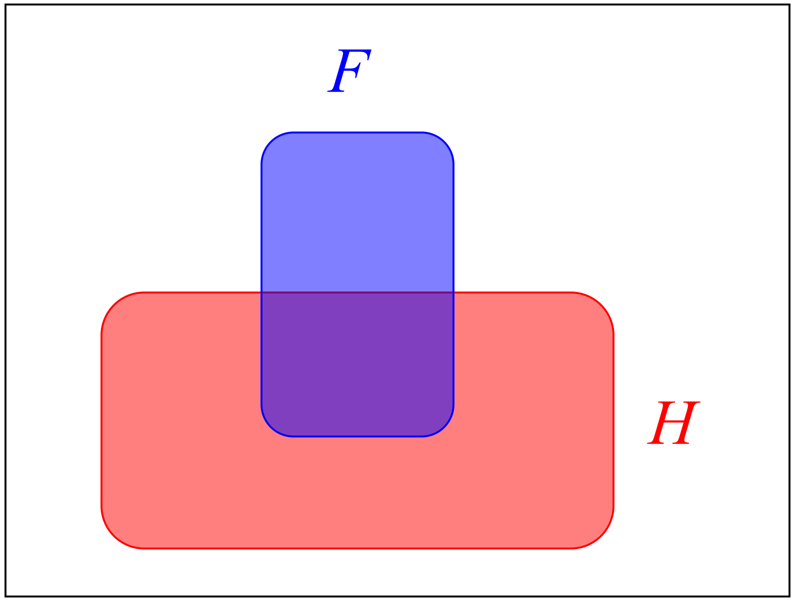
Backward chaining example



Probability

Thanks to Andrew Moore for some course material

Conditional Probability



H = "Have a headache"
F = "Coming down with Flu"

P(*H*) = 1/10
P(*F*) = 1/40
P(*H* | *F*) = 1/2

P(*H*|*F*) = Fraction of flu-inflicted worlds in which you have a headache

$$= \frac{\text{\# worlds with flu and headache}}{\text{\# worlds with flu}}$$

$$= \frac{\text{Area of "H and F" region}}{\text{Area of "F" region}}$$

$$= \frac{\text{P(H, F)}}{\text{P(F)}}$$

Definition of Conditional Probability

$$P(A \mid B) = \frac{P(A,B)}{P(B)}$$

Corollary: The Chain Rule (aka The Product Rule)

$$P(A,B) = P(A \mid B)P(B)$$

Full Joint Probability Distributions

Toothache	Cavity	Catch	P(Toothache, Cavity, Catch)
false	false	false	0.576
false	false	true	0.144
false	true	false	0.008
false	true	true	0.072
true	false	false	0.064
true	false	true	0.016
true	true	false	0.012
true	true	true	0.108

"Catch" means the dentist's probe catches in my teeth

This cell means P(Toothache = true, Cavity = true, Catch = true) = 0.108

Joint Probability Distribution

From the full joint probability distribution, we can calculate any probability involving the three random variables in this world e.g.

P(*Toothache* = *true* OR *Cavity* = true) =

P(*Toothache=true, Cavity=false, Catch=false*) +
P(*Toothache=true, Cavity=false*, *Catch=true*) +
P(*Toothache=false, Cavity=true, Catch=false*) +
P(*Toothache=false, Cavity=true*, *Catch=true*) +
P(*Toothache=true, Cavity=true, Catch=false*) +
P(*Toothache=true, Cavity = true, Catch=true*) +

= 0.064 + 0.016 + 0.008 + 0.072 + 0.012 + 0.108 = 0.28

Marginalization

We can even calculate marginal probabilities (the probability distribution over a subset of the variables) e.g:

P(*Toothache=true, Cavity=true*) =

P(*Toothache=true, Cavity=true, Catch=true*) +

P(*Toothache=true, Cavity=true, Catch=false*)

= 0.108 + 0.012 = 0.12

Marginalization

The general marginalization rule for any **sets** of variables ***Y*** and ***Z***:

$$\boldsymbol{P}(\boldsymbol{Y}) = \sum_{\mathbf{z}} \boldsymbol{P}(\boldsymbol{Y}, \mathbf{z})$$

z is over all possible combinations of values of ***Z*** (remember ***Z*** is a set)

or

$$\boldsymbol{P}(\boldsymbol{Y}) = \sum_{\mathbf{z}} \boldsymbol{P}(\boldsymbol{Y} \mid \mathbf{z}) P(\mathbf{z})$$

Normalization

$$P(Cavity = true \mid Toothache = true)$$
$$= \frac{P(Cavity = true, Toothache = true)}{P(Toothache = true)}$$
$$= \frac{0.108 + 0.012}{0.108 + 0.012 + 0.016 + 0.064} = 0.6$$

$$P(Cavity = false \mid Toothache = true)$$
$$= \frac{P(Cavity = false, Toothache = true)}{P(Toothache = true)}$$
$$= \frac{0.016 + 0.064}{0.108 + 0.012 + 0.016 + 0.064} = 0.4$$

Note that 1/P(*Toothache*=*true*) remains constant in the two equations and is called normalization constant α.

Independence

We say that variables X and Y are independent if any of the following hold: (they are all equivalent)

$$P(X \mid Y) = P(X) \quad \text{or}$$

$$P(Y \mid X) = P(Y) \quad \text{or}$$

$$P(X, Y) = P(X)P(Y)$$

Assume that Weather is independent of toothache, catch, cavity i.e.,

P(*Weather=cloudy* | *Toothache* = *toothache*, *Catch* = *catch*, *Cavity* = *cavity*)
= P(*Weather=cloudy*)

Bayes' Rule

The product rule can be written in two ways:

$P(A, B) = P(A \mid B)P(B)$

$P(A, B) = P(B \mid A)P(A)$

You can combine the equations above to get:

$$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)}$$

Bayes Rule Example

Meningitis causes stiff necks with probability 0.5. The prior probability of having meningitis is 0.00002. The prior probability of having a stiff neck is 0.05. What is the probability of having meningitis given that you have a stiff neck?

Let *m* = patient has meningitis

Let *s* = patient has stiff neck

P(*s* | *m*) = 0.5

P(*m*) = 0.00002

P(*s*) = 0.05

$$P(m \mid s) = \frac{P(s \mid m)P(m)}{P(s)} = \frac{(0.5)(0.00002)}{0.05} = 0.0002$$

Bayes Rule Example

Meningitis causes stiff necks with probability 0.5. The prior probability of having meningitis is 0.00002. The prior probability of having a stiff neck is 0.05. What is the probability of having meningitis given that you have a stiff neck?

Note: Even though P(s|m) = 0.5, P(m|s) = 0.0002

Let *m* = patient has meningitis

Let *s* = patient has stiff neck

P(*s* | *m*) = 0.5

P(*m*) = 0.00002

P(*s*) = 0.05

$$P(m \mid s) = \frac{P(s \mid m)P(m)}{P(s)} = \frac{(0.5)(0.00002)}{0.05} = 0.0002$$

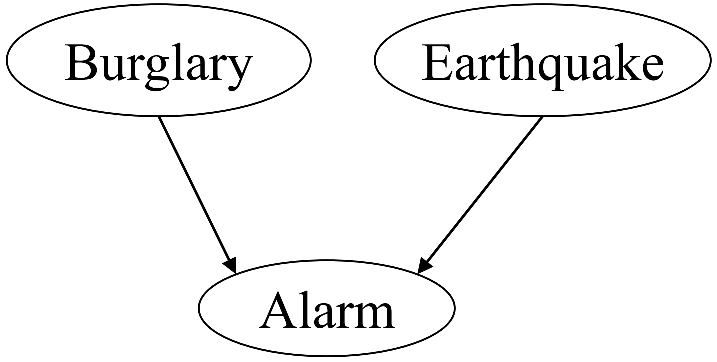
Bayesian Networks

Thanks to Andrew Moore for some course material

A Bayesian Network

B	P(B)
false	0.999
true	0.001

E	P(E)
false	0.998
true	0.002



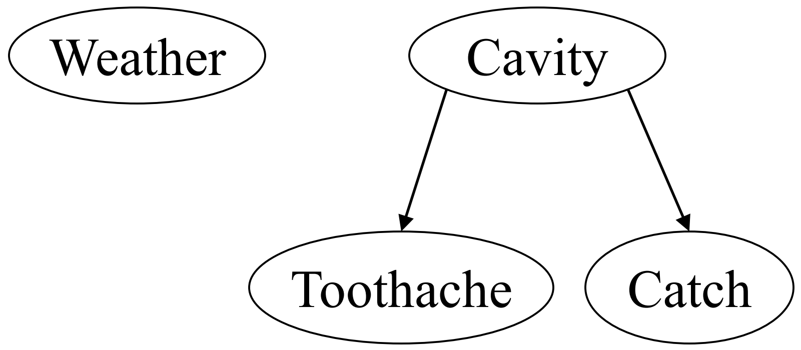
B	E	A	P(A|B,E)
false	false	false	0.999
false	false	true	0.001
false	true	false	0.71
false	true	true	0.29
true	false	false	0.06
true	false	true	0.94
true	true	false	0.05
true	true	true	0.95

A Bayesian network is made up of two parts:

1. A directed acyclic graph

2. A set of parameters: Conditional probability of each node given its parents

Bayesian Network Example



Things of note:

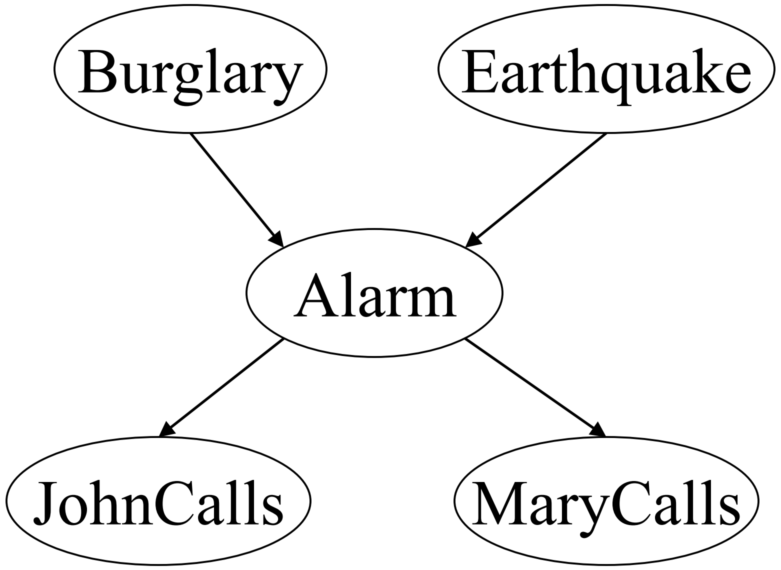
- Weather is independent of the other variables
- Toothache and Catch are conditionally independent given Cavity (this is represented by the fact that there is no link between Toothache and Catch and by the fact that they have Cavity as a parent)

A Representation of the Full Joint Distribution

- We will use the following abbreviations:
  - P(x_1, …, x_n) for P($X_1 = x_1 \wedge \ldots \wedge X_n = x_n$)
  - *parents(X_i)* for the values of the parents of X_i
- From the Bayes net, we can calculate:

$$P(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

Example



$\boldsymbol{P}(\mathit{JohnCalls}, \mathit{MaryCalls}, \mathit{Alarm}, \mathit{Burglary}, \mathit{Earthquake})$

$= \boldsymbol{P}(\mathit{JohnCalls} \mid \mathit{Alarm})\ \boldsymbol{P}(\mathit{MaryCalls} \mid \mathit{Alarm})\ \boldsymbol{P}(\mathit{Alarm} \mid \mathit{Burglary}, \mathit{Earthquake})\ \boldsymbol{P}(\mathit{Burglary})\ \boldsymbol{P}(\mathit{Earthquake})$

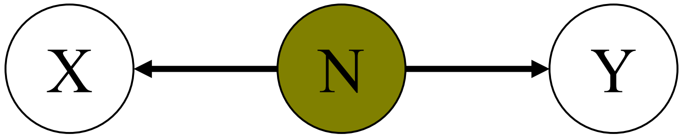
D-separation

- Let evidence nodes $E \subseteq V$ (where V are the vertices or nodes in the graph), and X and Y be distinct nodes in $V - E$.
- We say X and Y are d-separated by E in the Bayesian network if every undirected path between X and Y is blocked by E.
- What does it mean for a path to be blocked? There are 3 cases…

Case 1: Forks

There exists a node N on the path such that

- It is in the evidence set *E* (shaded grey)
- The arcs putting *N* in the path are "tail-to-tail".



The path between X and Y is blocked by N

Case 2: Chains

There exists a node N on the path such that

- It is in the evidence set *E*
- The arcs putting N in the path are “tail-to-head”.

$$\begin{tikzcd} X \arrow[r] & N \arrow[r] & Y \end{tikzcd}$$

Or

$$\begin{tikzcd} X & N \arrow[l] & Y \arrow[l] \end{tikzcd}$$

The path between X and Y is blocked by N

Case 3: Colliders

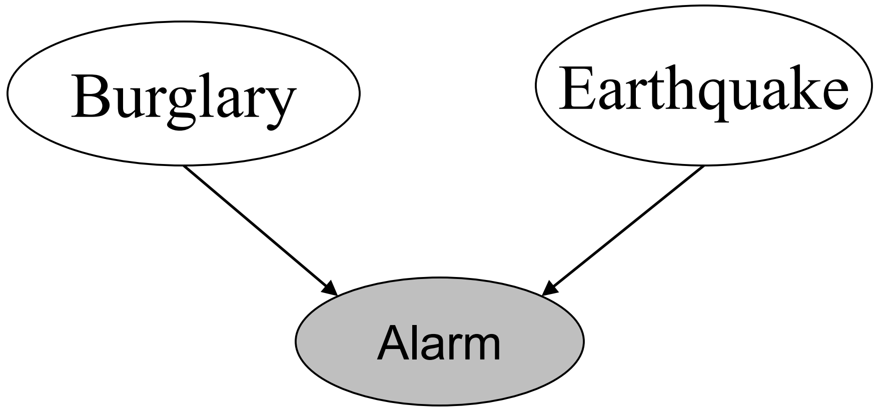
There exists a node N on the path such that

- It is NOT in the evidence set *E* (not shaded)
- Neither are any of its descendants
- The arcs putting *N* in the path are "head-to-head".

$$\begin{tikzcd} X \arrow[r] & N & Y \arrow[l] \end{tikzcd}$$

The path between X and Y is blocked by N (Note N is not in the evidence set)

Case 3 (Explaining Away)



Suppose that while you are on vacation, your neighbor lets you know your alarm went off. If you knew that a medium-sized earthquake happened, then you're probably relieved that it's probably not a burglar

The earthquake "explains away" the hypothetical burglar

This means that Burglary and Earthquake are not independent given Alarm.

d-separation Recipe

- To determine if I(X, Y | E), ignore the directions of the arrows, find all paths between X and Y
- Now pay attention to the arrows. Determine if the paths are blocked according to the 3 cases
- If all the paths are blocked, X and Y are d-separated given E
- Which means they are conditionally independent given E

Bayesian Networks (Inference)

Thanks to Andrew Moore for some of the slides.

Queries Formalized

We will use the following notation:

- X = query variable
- $E = \{E_1, \ldots, E_m\}$ is the set of evidence variables
- $\mathbf{e}$ = observed event
- $Y = \{Y_1, \ldots, Y_l)$ are the non-evidence (or hidden) variables
- The complete set of variables $\boldsymbol{X} = \{X\} \cup \boldsymbol{E} \cup \boldsymbol{Y}$

Need to calculate the query $\boldsymbol{P}(X \mid \boldsymbol{e})$

Inference by Enumeration

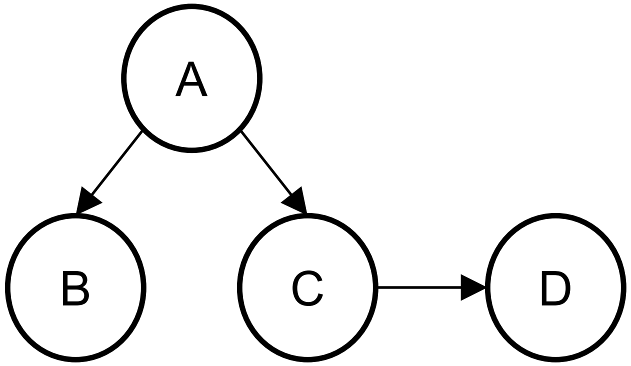
- Recall that:

$$\mathbf{P}(X \mid \mathbf{e}) = \alpha \mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$$

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

This means you can answer queries by computing sums of products of conditional probabilities from the network

Example #1

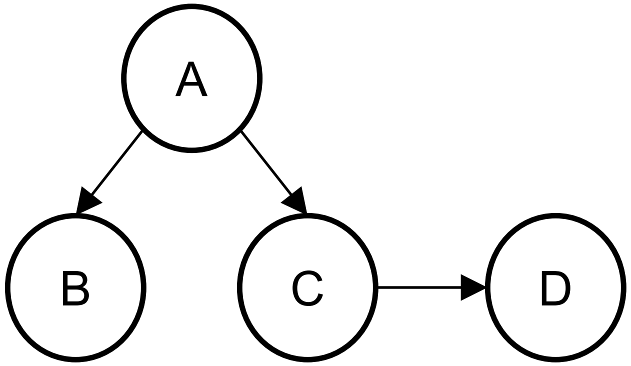


Query: P(B=true | C=true)

How do you solve this? 2 steps:

1. Express it in terms of the joint probability distribution P(A, B, C,D)
2. Express the joint probability distribution in terms of the entries in the CPTs of the Bayes net

Example #1

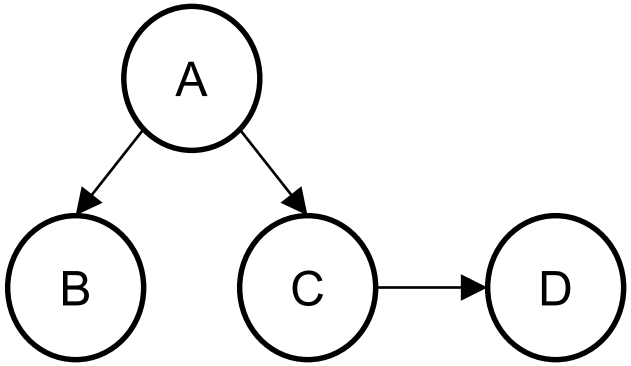


Whenever you see a conditional like P(B=true | C=true), use the Chain Rule:

P(B | C) = P(B, C) / P(C)

$$P(B = true \mid C = true)$$
$$= \frac{P(B = true, C = true)}{P(C = true)}$$

Example #1



Whenever you need to get a subset of the variables e.g. P(B,C) from the full joint distribution P(A,B,C,D), use marginalization:

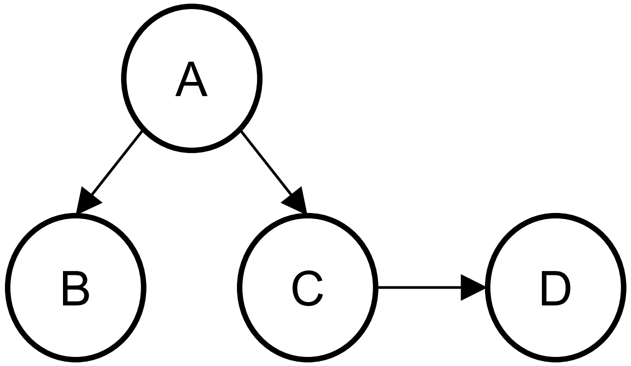
$$P(X) = \sum_{y} P(X, Y = y)$$

$$P(B = true \mid C = true)$$

$$= \frac{P(B = true, C = true)}{P(C = true)}$$

$$= \frac{\sum_{a}\sum_{d} P(A = a, B = true, C = true, D = d)}{\sum_{a}\sum_{b}\sum_{d} P(A = a, B = b, C = true, D = d)}$$

Example #1



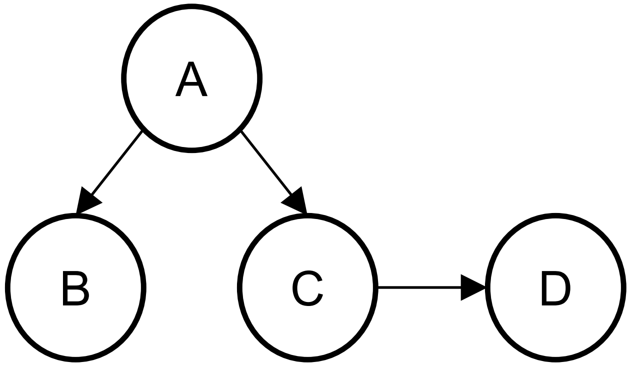
To express the joint probability distribution as the entries in the CPTs, use:

$$P(X_1, \ldots, X_N) = \prod_{i=1}^{N} P(X_i \mid Parents(X_i))$$

$$= \frac{\sum_a \sum_d P(A=a, B=true, C=true, D=d)}{\sum_a \sum_b \sum_d P(A=a, B=b, C=true, D=d)}$$

$$= \frac{\sum_a \sum_d P(A=a)P(B=true \mid A=a)P(C=true \mid A=a)P(D=d \mid C=true)}{\sum_a \sum_b \sum_d P(A=a)P(B=b \mid A=a)P(C=true \mid A=a)P(D=d \mid C=true)}$$

Example #1

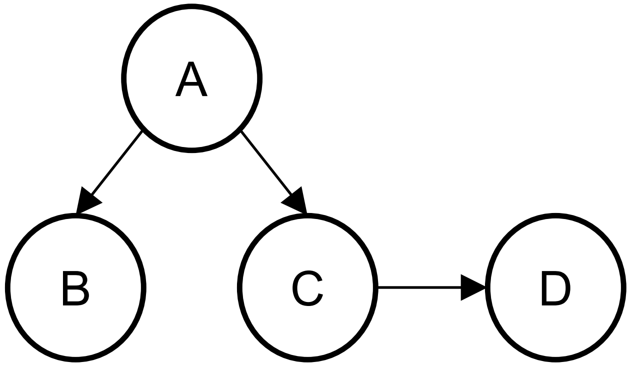


Take the probabilities that don't depend on the terms in the summation and move them outside the summation

$$= \frac{\sum_a \sum_d P(A=a)P(B=true \mid A=a)P(C=true \mid A=a)P(D=d \mid C=true)}{\sum_a \sum_b \sum_d P(A=a)P(B=b \mid A=a)P(C=true \mid A=a)P(D=d \mid C=true)}$$

$$= \frac{\sum_a P(A=a)P(B=true \mid A=a)P(C=true \mid A=a)\sum_d P(D=d \mid C=true)}{\sum_a P(A=a)\sum_b P(B=b \mid A=a)P(C=true \mid A=a)\sum_d P(D=d \mid C=true)}$$

Example #1



Take the probabilities that don't depend on the terms in the summation and move them outside the summation

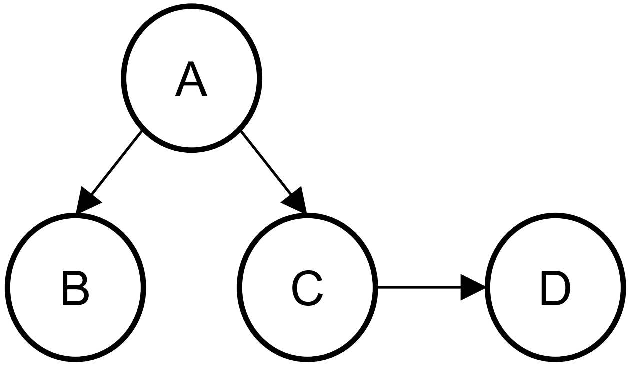
Sums to 1

$$= \frac{\sum_a \sum_d P(A=a)P(B=true \mid A=a)P(C=true \mid A=a)P(D=d \mid C=true)}{\sum_a \sum_b \sum_d P(A=a)P(B=b \mid A=a)P(C=true \mid A=a)P(D=d \mid C=true)}$$

$$= \frac{\sum_a P(A=a)P(B=true \mid A=a)P(C=true \mid A=a)\underline{\sum_d P(D=d \mid C=true)}}{\sum_a P(A=a)\sum_b P(B=b \mid A=a)P(C=true \mid A=a)\underline{\sum_d P(D=d \mid C=true)}}$$

Sums to 1

Example #1



Take the probabilities that don't depend on the terms in the summation and move them outside the summation

$$= \frac{\sum_a P(A=a)P(B=true \mid A=a)P(C=true \mid A=a)}{\sum_a P(A=a)\sum_b P(B=b \mid A=a)\underline{P(C=true \mid A=a)}}$$

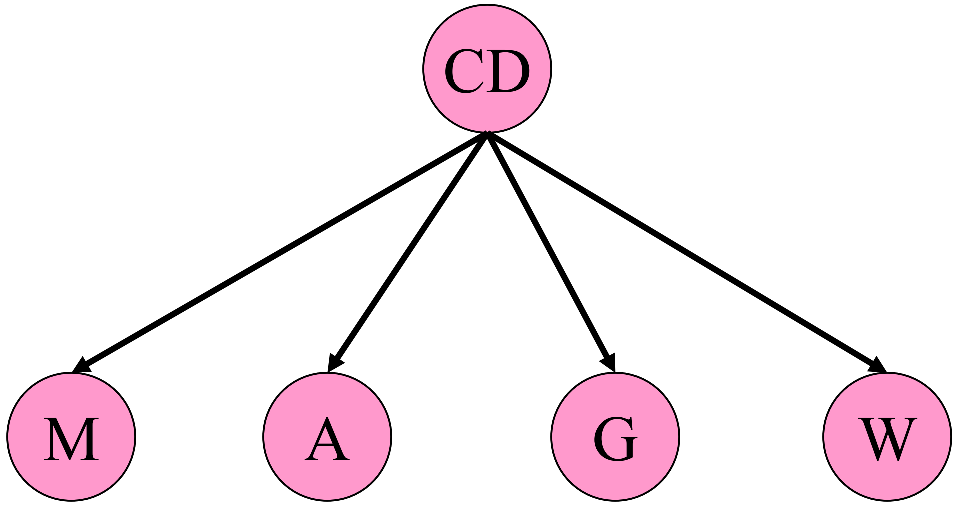
$\longleftarrow$ Doesn't depend on b. Can move to the left

$$= \frac{\sum_a P(A=a)P(B=true \mid A=a)P(C=true \mid A=a)}{\sum_a P(A=a)P(C=true \mid A=a)\underline{\sum_b P(B=b \mid A=a)}}$$

$\longleftarrow$ Sums to 1

$$= \frac{\sum_a P(A=a)P(B=true \mid A=a)P(C=true \mid A=a)}{\sum_a P(A=a)P(C=true \mid A=a)}$$

Naïve Bayes Structure

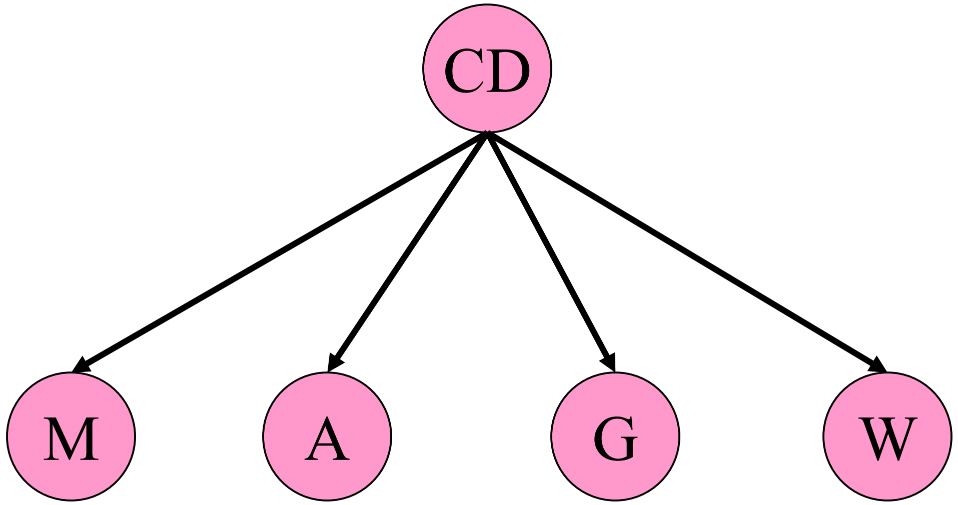


Notice the conditional independence assumption:

The features are conditionally independent given the class variable.

Naïve Bayes Parameters

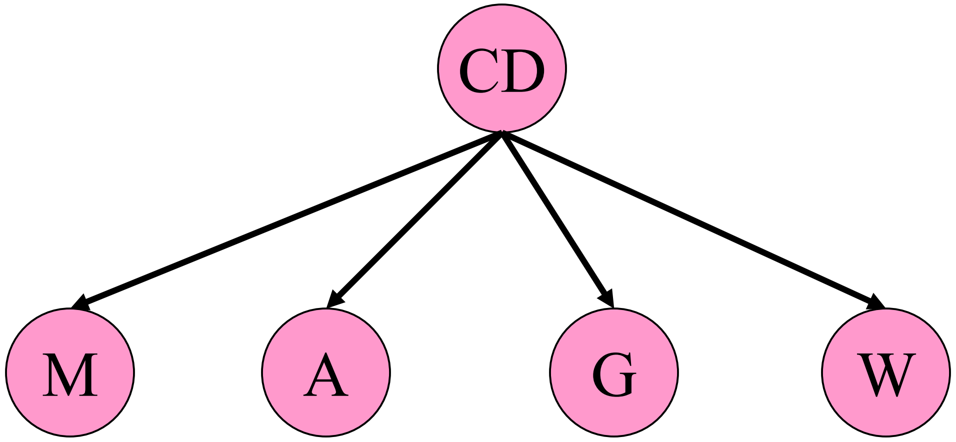
$P(CD) = ?$



$P(M \mid CD) = ?$ $P(A \mid CD) = ?$ $P(G \mid CD) = ?$ $P(W \mid CD) = ?$

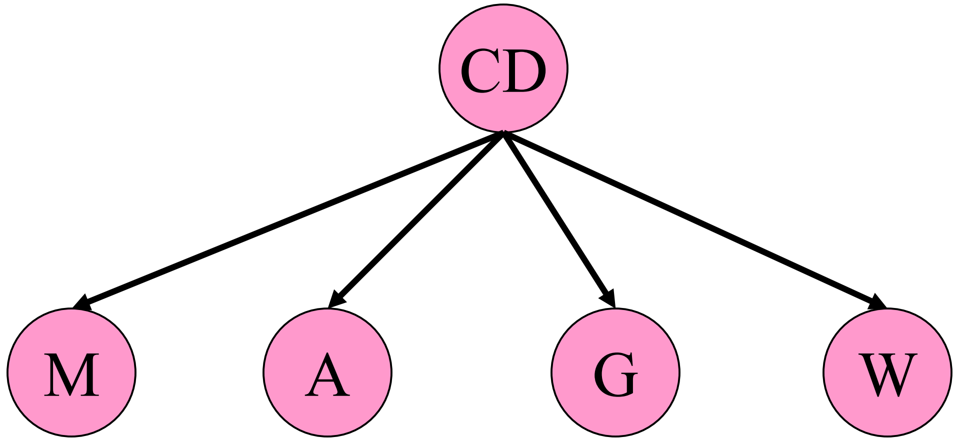
How do you get these parameters from the training data?

Naïve Bayes Parameters



CD	P(CD)
false	(# of records in training data with CD = false) / (# of records in training data)
true	(# of records in training data with CD = true) / (# of records in training data)

Naïve Bayes Parameters



M	CD	P(M | CD)
false	false	(# of records with M = false and CD = false) / (# of records with CD = false)
false	true	(# of records with M = false and CD = true) / (# of records with CD = true)
true	false	(# of records with M = true and CD = false) / (# of records with CD = false)
true	true	(# of records with M = true and CD = true) / (# of records with CD = true)

Inference in Naïve Bayes

$$\boldsymbol{P}(CD \mid M, A, G, W)$$

$$= \frac{\boldsymbol{P}(M, A, G, W \mid CD) P(CD)}{\boldsymbol{P}(M, A, G, W)}$$

By Bayes Rule

$$= \alpha \boldsymbol{P}(M, A, G, W \mid CD) \boldsymbol{P}(CD)$$

Treat denominator as constant

$$= \alpha \boldsymbol{P}(CD) \boldsymbol{P}(M \mid CD) \boldsymbol{P}(A \mid CD) \boldsymbol{P}(G \mid CD) \boldsymbol{P}(W \mid CD)$$

From conditional independence

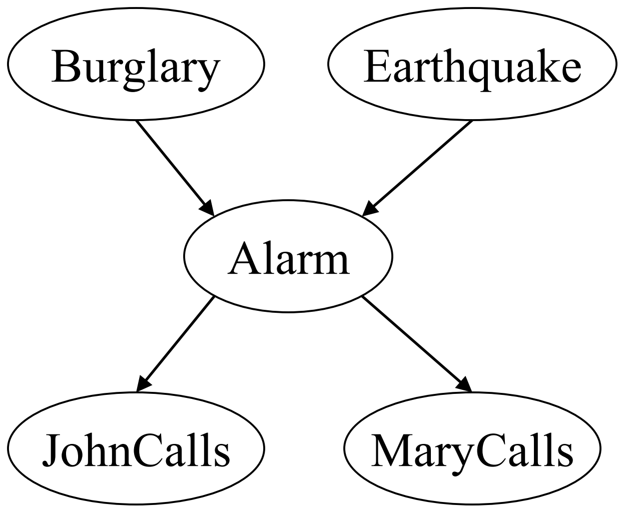
Bayesian Networks Learning

Constructing a Locally Structured Bayesian Network

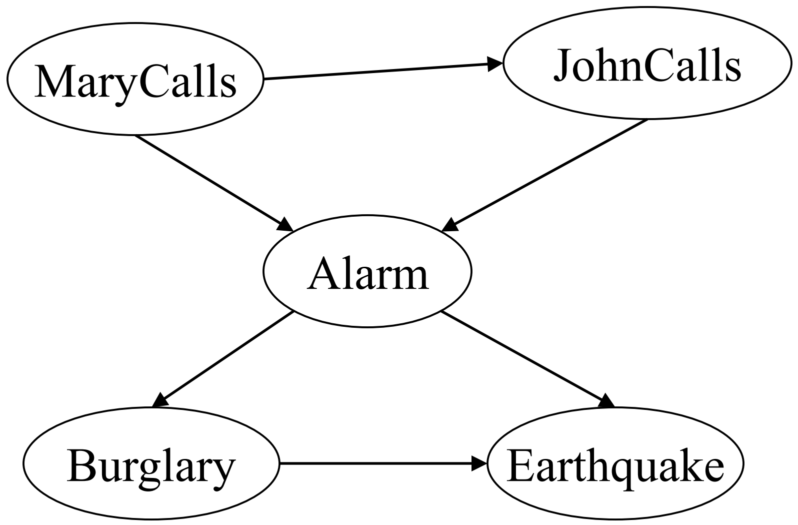
- Needs:
  1. Each variable to be directly influenced by a few others
  2. Parents are the direct influences of a node
- Process:
  - Add “root causes” first (those that are not influenced by any others)
  - For each new variable, add the smallest subset of all previous variables as parents such that it is independent of the rest given the parents
  - Keep going until you reach the “leaves” which do not have a direct causal influence on the other variables

Choosing the Wrong Order

Compact network



Not-So-Compact Network



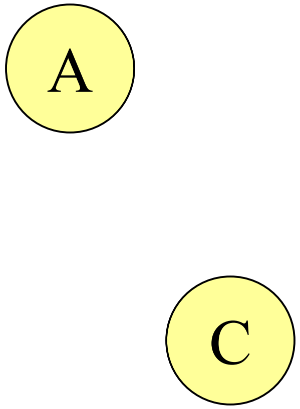
Note: Both networks can represent the same joint probability distribution. The problem is that the one on the right doesn't represent all the conditional independence relationships and some links need not be there

Learning the Structure

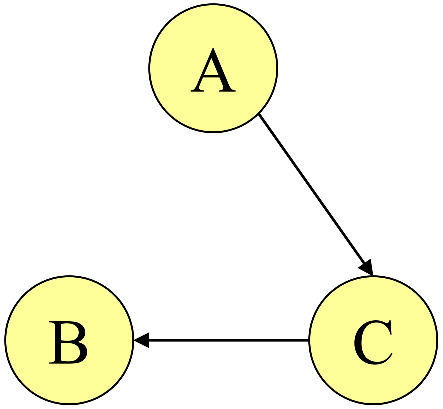
- It is impossible to do an exhaustive search to find the optimal structure in large networks
- Need to resort to local search methods e.g. hill-climbing, simulated annealing
- We'll illustrate this using a 3 node example.

Local Search Methods

Initial State:



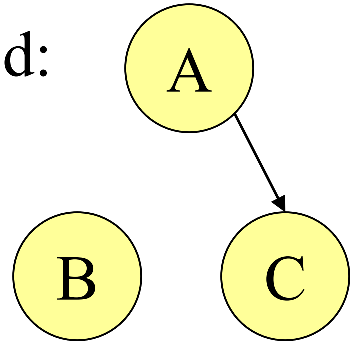
Start with no links



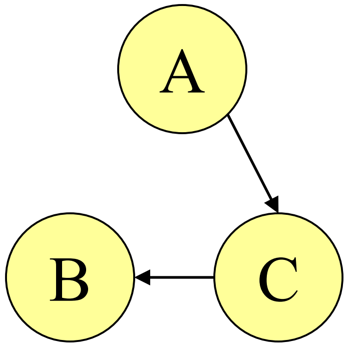
Start with a random set of links

Local Search Methods

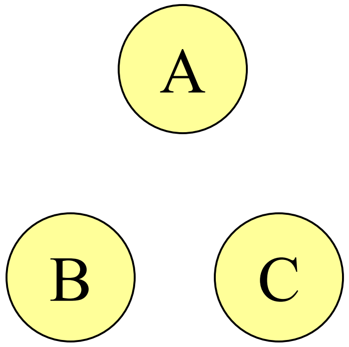
Neighborhood:



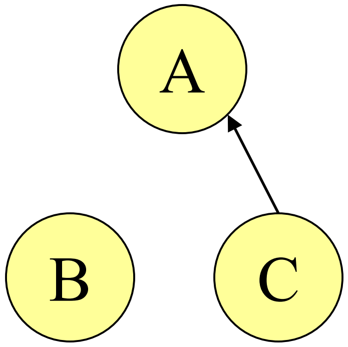
Current State



Add a link



Remove a link



Reverse a link

Things to Watch Out For

- Need to avoid introducing cycles
- Need to re-estimate parameters everytime you modify a link in the Bayes net
  - Do you need to re-estimate the parameters for all nodes?
  - No, just the ones that are affected by the modified link
- Lots of local optima problems. Use random restarts.

The Evaluation Function

- How do we know if a Bayes net structure is good?
- Two types of evaluation functions:
  1. Evaluate if conditional independence relationships in the learned network match those in the data
  2. Evaluate how well the learned network explains the data (in the probabilistic sense).

Machine Learning Decision Tree Induction

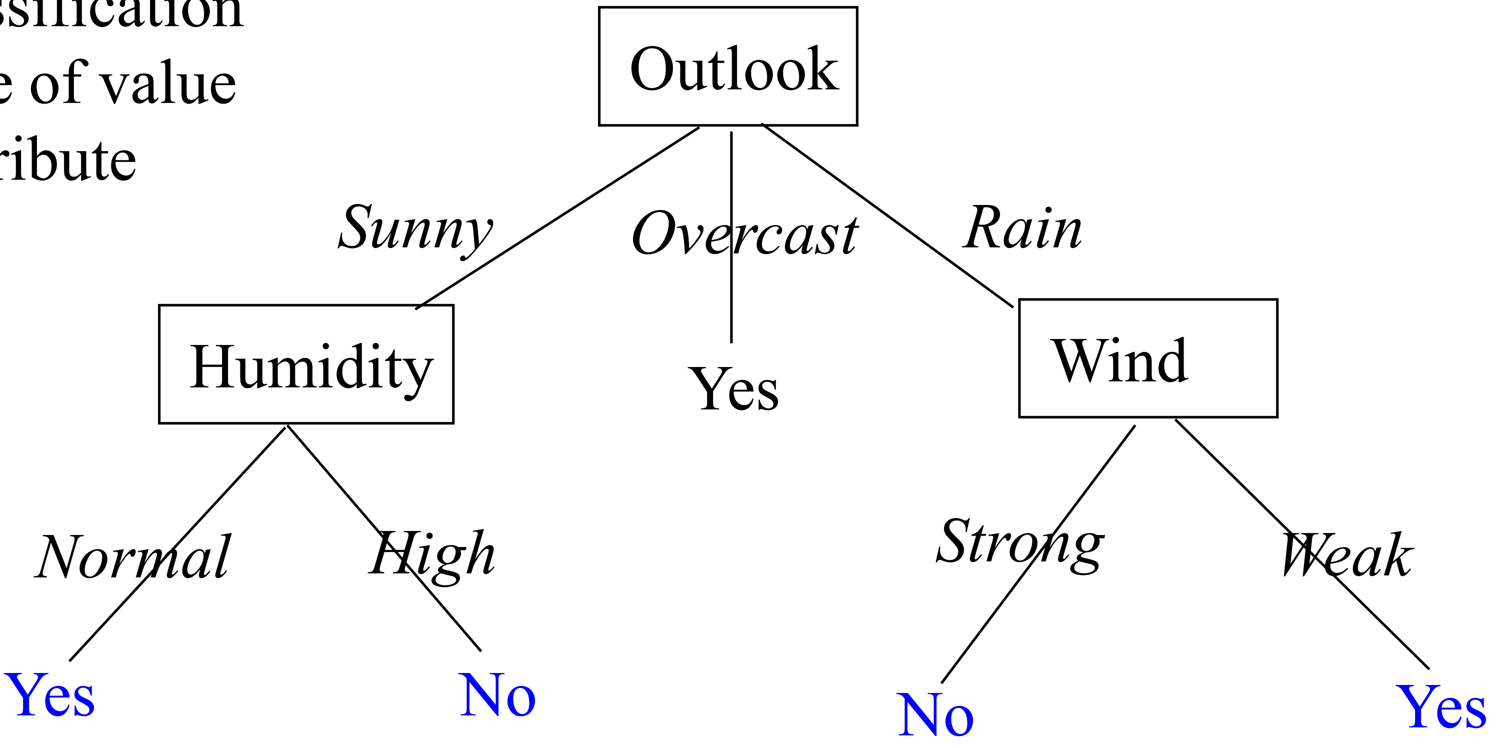
Thanks to: Dan Weld, University of Washington

Decision Tree Representation

Good day for tennis?

Leaves = classification

Arcs = choice of value for parent attribute



Decision tree is equivalent to logic in disjunctive normal form

G-Day $\Leftrightarrow$ (Sunny $\wedge$ Normal) $\vee$ Overcast $\vee$ (Rain $\wedge$ Weak)

Decision Tree Algorithm

```
BuildTree(TraingData)
        Split(TrainingData)

Split(D)
        If (all points in D are of the same class)
                Then Return
        For each attribute A
                Evaluate splits on attribute A
        Use best split to partition D into D1, D2
        Split (D1)
        Split (D2)
```

Splitting heuristic: Mutual Information

Mutual information or Information gain (A = class, B = feature)

$$I(A;B) = H(A) - H(A|B), \text{where}$$

$$H(A) = -P(A=1)\log(P(A=1)) - P(A=0)\log(P(A=0))$$

$$H(A|B) = \sum_i P(B=v_i)H(A|B=v_i)$$

	A=0	A=1
B=0	12	8
B=1	8	2

Counts table

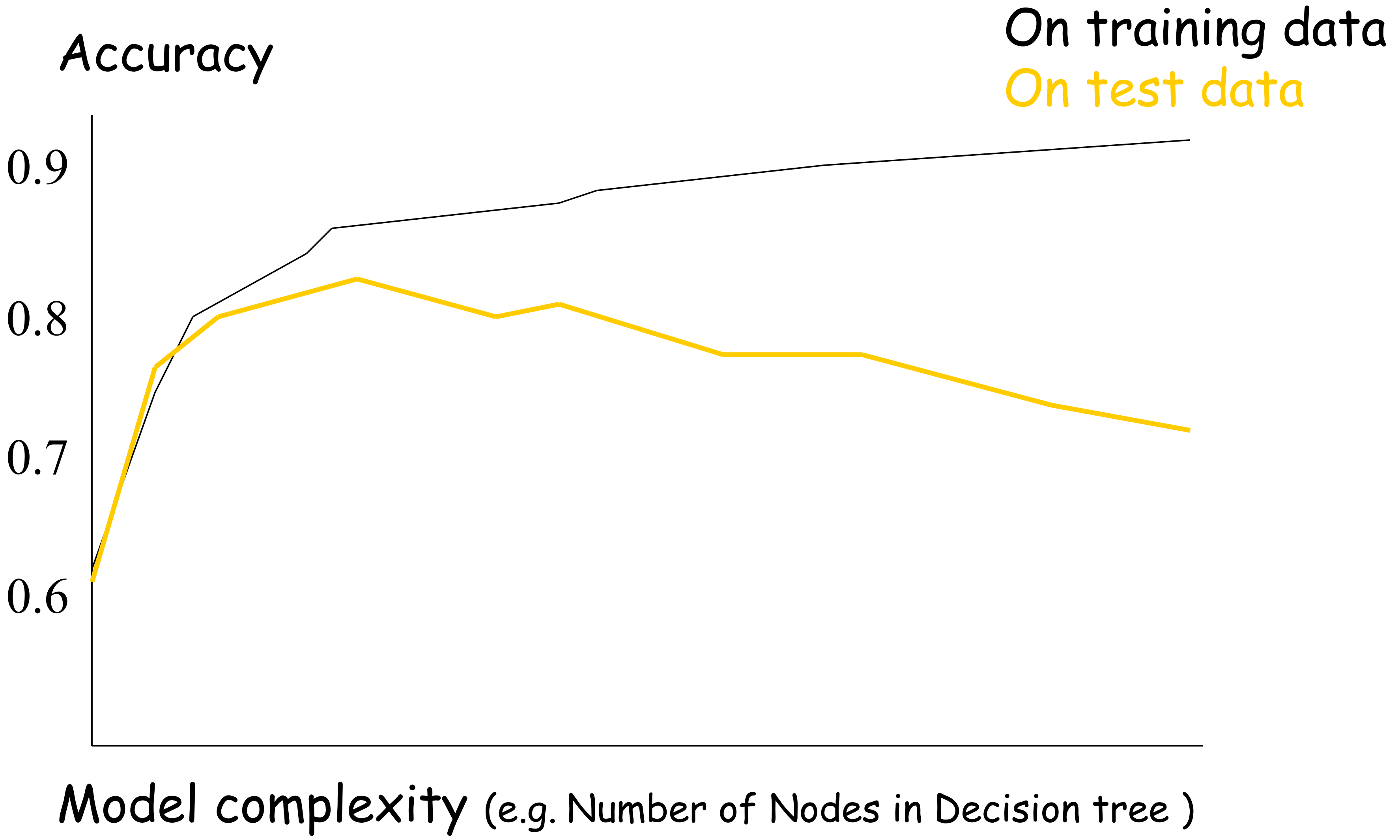
$$H(A) = -P(A=0)\log P(A=0) \; - P(A=1)\log P(A=1)$$
= -(20/30) log (20/30) – (10/30) log (10/30) = 0.9183

$$H(A|B=0\,)$$
$$= -P(A=0|B=0)\log P(A=0|B=0)$$
$$-\,P(A=1|B=0)\log P(A=1|B=0)$$
= -(12/20) log (12/20) – (8/20) log (8/20) = 0.9709

$$H(A|B=1\,) = -P(A=0|B=1)\log P(A=0|B=1) \; - P(A=1|B=1)\log P(A=1|B=1)$$
= -(8/10) log (8/10) – (2/10) log (2/10) = 0.7219

$$I(A;B\,) = H(A) - P(B=0)H(A|B=0) - P(B=1)H(A|B=1)$$
= 0.9183 – (20/30)*0.9709 – (10/30)*0.7219 = 0.0304

Overfitting



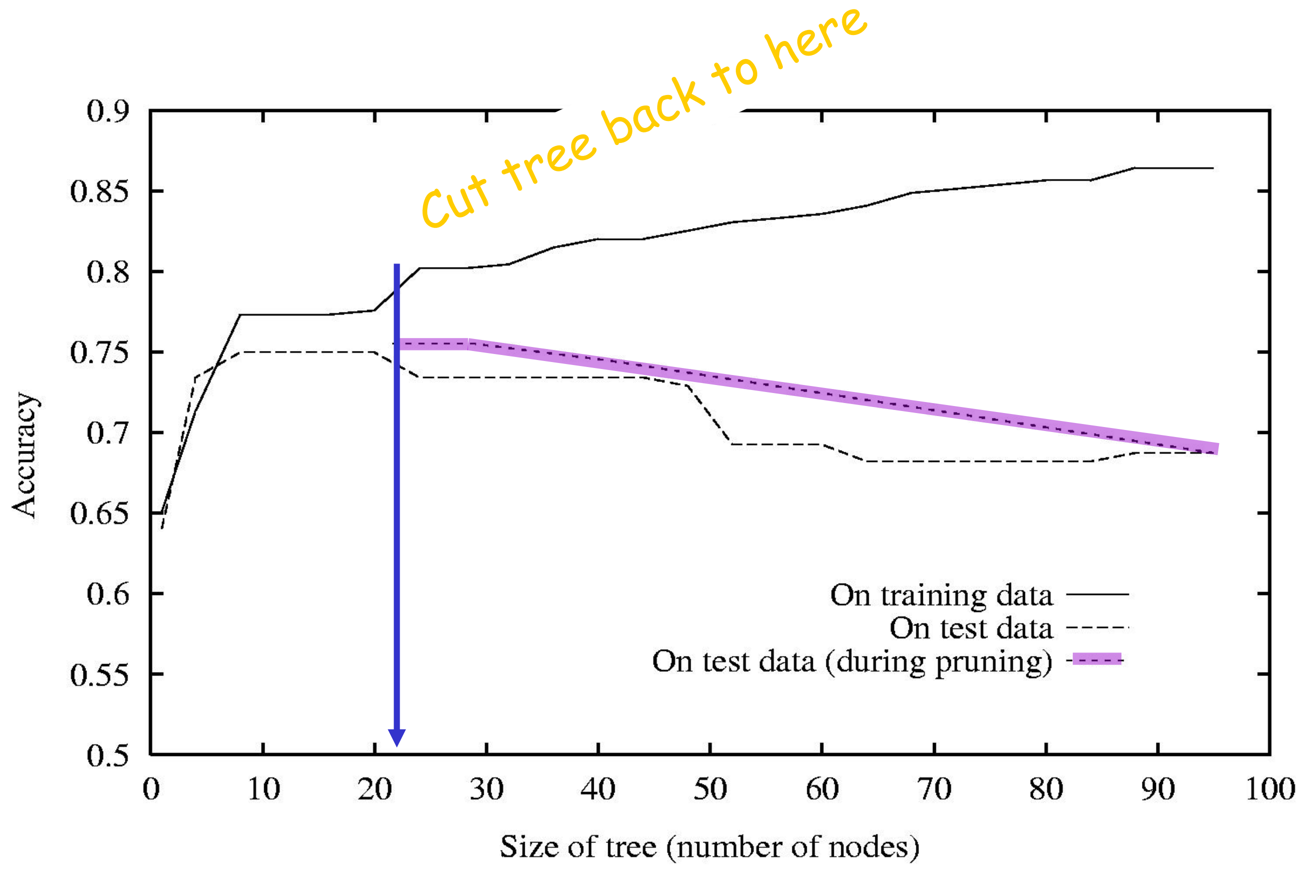
Reduced-Error Pruning

Split data into *training* and *validation* set

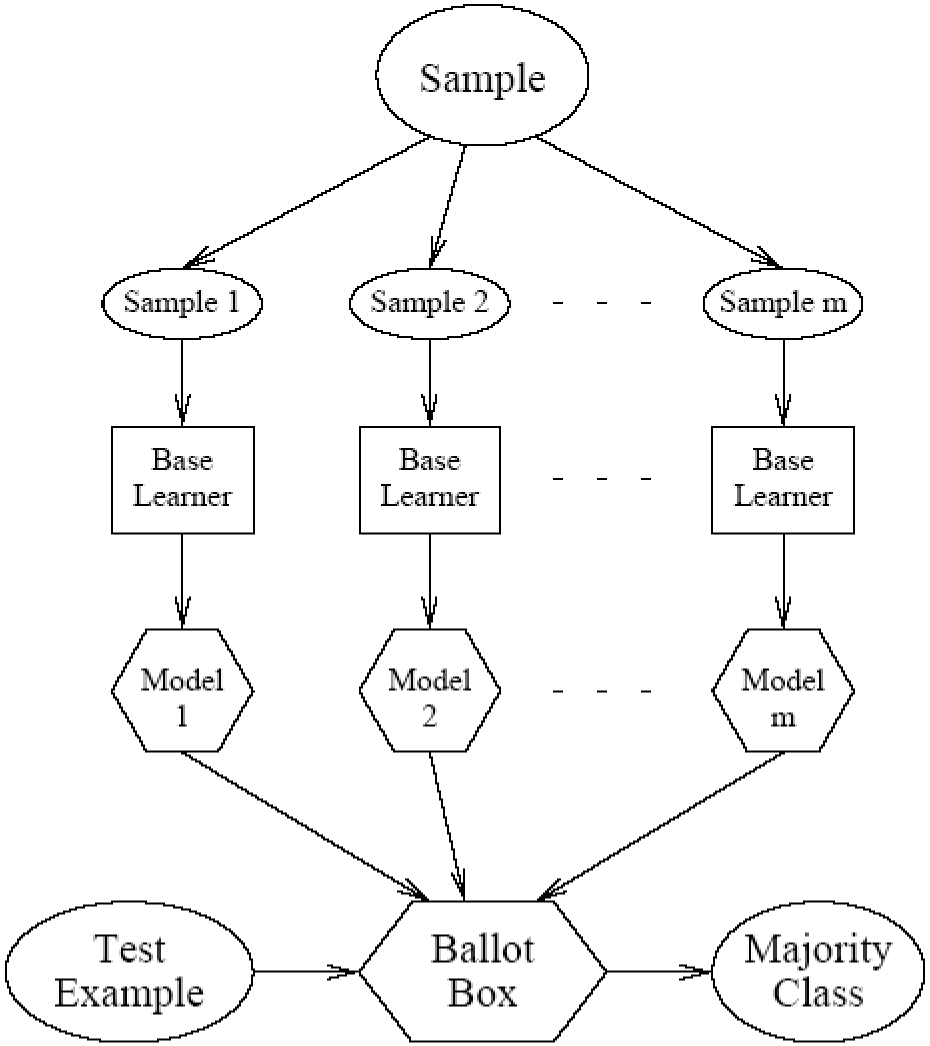
Do until further pruning is harmful:

1. Evaluate impact on *validation* set of pruning each possible node (plus those below it)
2. Greedily remove the one that most improves *validation* set accuracy

Effect of Reduced-Error Pruning



Voting


Sample
Sample 1
Sample 2
Sample m
Base Learner
Base Learner
Base Learner
Model 1
Model 2
Model m
Test Example
Ballot Box
Majority Class

Ensemble Construction II

Bagging

- Generate k sets of training examples
- For each set
  - Draw m examples randomly (with replacement)
  - From the original set of m examples
- Each training set corresponds to
  - 63.2% of original
  - (+ duplicates)
- Now train classifier on each set

Ensemble Creation III

Oregon State UNIVERSITY

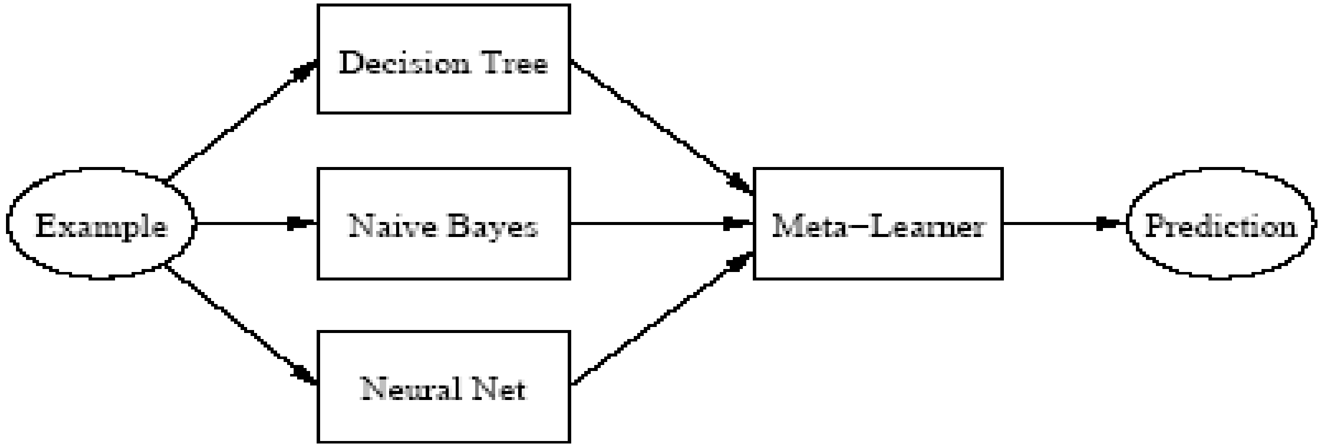
Boosting

- Maintain prob distribution over set of training ex
- Create k sets of training data iteratively:
- On iteration *i*
  - Draw m examples randomly (like bagging)
  - But use probability distribution to bias selection
  - Train classifier number *i* on this training set
  - Test partial ensemble (of *i* classifiers) on all training exs
  - Modify distribution: increase P of each error ex
- Create harder and harder learning problems...

Ensemble Creation IV

Stacking

- Train several base learners
- Next train meta-learner
  - Learns when base learners are right / wrong
  - Now meta learner arbitrates



Train using cross validated committees

- Meta-L inputs = base learner predictions
- Training examples = 'test set' from cross validation