CS 332 Artificial Intelligence

SCHOOL OF COMPUTER ENGINEERING AND TECHNOLOGY

CS 332 Artificial Intelligence

Teaching Scheme **Credits: 2 + 1 = 3**

Theory: 4 Hrs / Week **Practical:** 2 Hrs / Week

Course Objectives:

1) To understand the concept of Artificial Intelligence (AI)
2) To learn various peculiar search strategies for AI
3) To develop a mind to solve real world problems unconventionally with optimality

Course Outcomes:

1) Identify and apply suitable Intelligent agents for various AI applications
2) Design smart system using different informed search / uninformed search or heuristic approaches.
3) Identify knowledge associated and represent it by ontological engineering to plan a strategy to solve given problem.

Syllabus

Knowledge Inference and Expert System

Basics of Probability, Markov Model, Statistical reasoning, Bayes' Theorem and its use, Bayesian learning and network

Expert systems: Architecture of Expert system, Role of Expert system, Inference engine, Knowledge acquisition, Typical Expert systems- MYCIN, Expert systems shells, Applications of Expert systems.

Knowledge Inference and Expert Systems

Contents

- Basics of Probability,
- Markov Model,
- Statistical reasoning,
- Bayes' Theorem and its use, Bayesian learning and network
- Architecture of Expert system, Role of Expert system,
- Inference engine, Knowledge acquisition,
- Typical Expert systems- MYCIN, Expert systems shells,
- Applications of Expert systems.

Acting under uncertainty

A logical agent uses propositions that are true, false or unknown

When the logical agent knows enough facts about it environment it derives plans that are guaranteed to work.

Unfortunately, agents almost never have access to the whole truth about their environment and therefore **agents must act under uncertainty**

Uncertainty

Let action A_t = leave for airport t minutes before flight . Will A_t get me there on time?

Problems:

1. partial observability (road state, other drivers' plans, noisy sensors)
2. uncertainty in action outcomes (flat tire, etc.)
3. immense complexity of modeling and predicting traffic

Hence a purely logical approach either

1. risks falsehood: “A_{25} will get me there on time”, or
2. leads to conclusions that are too weak for decision making:

“A_{25} will get me there on time if there's no accident on the bridge and it doesn't rain and my tires remain intact etc etc.” (A_{1440} might reasonably be said to get me there on time but I'd have to stay overnight in the airport …)

Probability to the Rescue

Probability

- Model agent's degree of belief, given the available evidence.

- A_{25} will get me there on time with probability 0.04

Probability in AI models our ignorance, not the true state of the world.

The statement “With probability 0.7 I have a cavity” means:
I either have a cavity or not, but I don’t have all the necessary information to know this for sure.

Probability

1. All probability statements must indicate the evidence with respect to which the probability is being assessed
2. As agent receives new percepts, the probability assessments are updated to reflect new evidence
3. Before the evidence is obtained , we have **prior** or **unconditional probability**
4. After the evidence is obtained, we have **posterior or conditional probability**
5. Generally, the **agent** will have some evidence from its percepts and will be interested in computing the posterior probabilities of the outcomes it cares about

Probability

Subjective probability:

Probabilities relate propositions to agent's own state of knowledge

e.g., $P(A_{25} \mid \text{no reported accidents at 3 a.m.}) = 0.06$

Probabilities of propositions change with new evidence:

e.g., $P(A_{25} \mid \text{no reported accidents at 5 a.m.}) = 0.15$

Making decisions under uncertainty

Suppose I believe the following:

$P(A_{25}$ gets me there on time | …) = 0.04

$P(A_{90}$ gets me there on time | …) = 0.70

$P(A_{120}$ gets me there on time | …) = 0.95

$P(A_{1440}$ gets me there on time | …) = 0.9999

Which action to choose?

Depends on my preferences for missing flight vs. time spent waiting, etc.

- Utility theory is used to represent and infer preferences
- Decision theory = probability theory + utility theory

Probability Basics

We begin with a set Ω—the sample space

- e.g., 6 possible rolls of a die.
- Ω can be infinite

$\omega \in \Omega$ is a sample point/possible world/atomic event

A probability space or probability model is a sample space with an assignment $P(\omega)$ for every $\omega \in \Omega$ such that:

$0 \leq P(\omega) \leq 1$

$\Sigma_\omega P(\omega) = 1$

e.g., $P(1) = P(2) = P(3) = P(4) = P(5) = P(6) = 1/6.$

An event A is any subset of Ω:

$P(A) = \Sigma_{\{\omega \in A\}} P(\omega)$

e.g., $P(\text{die roll} < 4) = P(1) + P(2) + P(3) = 1/6 + 1/6 + 1/6 = 1/2$

Probability Basics

- A random variable is a function from sample points to some range
  - e.g., $Odd(1) = true$, has a boolean-valued range.

P induces a probability distribution for any r.v. X:

$$P(X = x_i) = \Sigma_{\{\omega : X(\omega) = x_i\}} P(\omega)$$

e.g., $P(Odd = true) = P(1) + P(3) + P(5) = 1/6 + 1/6 + 1/6 = 1/2$

Capital letter: random variable
lower case: single value

Probability Basics

Basic element: random variable

Similar to propositional logic: possible worlds defined by assignment of values to random variables.

Boolean random variables

e.g., *Cavity* (do I have a cavity?)

Discrete random variables

e.g., *Weather* is one of <*sunny,rainy,cloudy,snow*>

Elementary proposition constructed by assignment of a value to a random variable: e.g., *Weather* = *sunny*, *Cavity* = *false* (abbreviated as ¬*cavity*)

Complex propositions formed from elementary propositions and standard logical connectives e.g., *Weather* = *sunny* ∨ *Cavity* = *false*

Syntax

Atomic event: A complete specification of the state of the world about which the agent is uncertain (i.e. a full assignment of values to all variables in the universe, a unique single world).

E.g., if the world consists of only two Boolean variables *Cavity* and *Toothache*, then there are 4 distinct atomic events:

$Cavity = false \wedge Toothache = false$

$Cavity = false \wedge Toothache = true$

$Cavity = true \wedge Toothache = false$

$Cavity = true \wedge Toothache = true$

if some atomic event is true, then all other other atomic events are false.

There is always some atomic event true.

Hence, there is exactly 1 atomic event true.

Atomic events are mutually exclusive and the set of all possible atomic events is exhaustive

Prior probability

Prior or unconditional probabilities of propositions

e.g., P(*Cavity* = true) = 0.1 and P(*Weather* = sunny) = 0.72 correspond to belief prior to arrival of any (new) evidence

Probability distribution gives values for all possible assignments:

P(*Weather*) = <0.72,0.1,0.08,0.1> (normalized, i.e., sums to 1)

Joint probability distribution for a set of random variables gives the probability of every atomic event of those random variables

P(*Weather*,*Cavity*) = a 4 × 2 matrix of values:

Weather =	sunny	rainy	cloudy	snow
Cavity = true	0.144	0.02	0.016	0.02
Cavity = false	0.576	0.08	0.064	0.08

Every question about a domain can be answered by the joint distribution

Conditional probability

Conditional or posterior probabilities
e.g., P(*cavity* | *toothache*) = 0.8 i.e., given that *Toothache*=*true* is all I know.

Note that **P**(Cavity|Toothache) is a 2x2 array, normalized over columns.

If we know more, e.g., cavity is also given, then we have
P(*cavity* | *toothache*,*cavity*) = 1

New evidence may be irrelevant, allowing simplification, e.g.,
P(*cavity* | *toothache*, *sunny*) = P(*cavity* | *toothache*) = 0.8

Conditional probability

Definition of conditional probability:

$P(a \mid b) = P(a \wedge b) / P(b) \quad \text{if} \quad P(b) > 0$

Product rule gives an alternative formulation:

$P(a \wedge b) = P(a \mid b) P(b) = P(b \mid a) P(a)$

Bayes Rule: $P(a|b) = P(b|a) P(a) / P(b)$

A general version holds for whole distributions, e.g.,

$\mathbf{P}(Weather, Cavity) = \mathbf{P}(Weather \mid Cavity)\, \mathbf{P}(Cavity)$

(View as a set of 4 × 2 equations, not matrix multiplication)

Chain rule is derived by successive application of product rule:

$$
\begin{aligned}
\mathbf{P}(X_1, \ldots, X_n) &= \mathbf{P}(X_1,\ldots,X_{n-1})\, \mathbf{P}(X_n \mid X_1,\ldots,X_{n-1}) \\
&= \mathbf{P}(X_1,\ldots,X_{n-2})\, \mathbf{P}(X_{n-1} \mid X_1,\ldots,X_{n-2})\, \mathbf{P}(X_n \mid X_1,\ldots,X_{n-1}) \\
&= \ldots \\
&= \pi_{i=1}\text{^}n\, \mathbf{P}(X_i \mid X_1, \ldots, X_{i-1})
\end{aligned}
$$

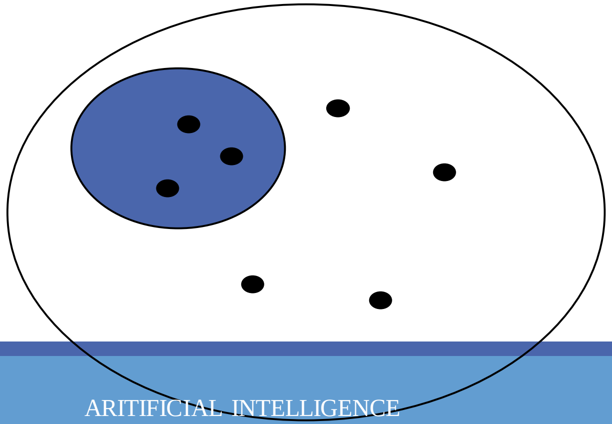
Inference by enumeration

Start with the joint probability distribution:

	$toothache$		$\neg\ toothache$	
	$catch$	$\neg\ catch$	$catch$	$\neg\ catch$
$cavity$	.108	.012	.072	.008
$\neg\ cavity$	.016	.064	.144	.576

For any proposition a, sum the atomic events where it is true: $P(a) = \Sigma_{\omega \text{ s.t. } a=true} P(\omega)$

P(a)=1/7 + 1/7 + 1/7 = 3/7



Inference by enumeration

Start with the joint probability distribution:

	toothache		¬ *toothache*	
	catch	¬ *catch*	*catch*	¬ *catch*
cavity	.108	.012	.072	.008
¬ *cavity*	.016	.064	.144	.576

For any proposition a, sum the atomic events where it is true: $P(a) = \Sigma_{\omega:\omega \text{ s.t. } a=\text{true}} \; P(\omega)$

P(*toothache*) = 0.108 + 0.012 + 0.016 + 0.064 = 0.2

Inference by enumeration

Start with the joint probability distribution:

	toothache		*¬ toothache*	
	catch	*¬ catch*	*catch*	*¬ catch*
cavity	.108	.012	.072	.008
¬ cavity	.016	.064	.144	.576

Can also compute conditional probabilities:

$P(\neg cavity \mid toothache) = P(\neg cavity \wedge toothache) / P(toothache)$

$= (0.016+0.064) / (0.108 + 0.012 + 0.016 + 0.064)$

$= 0.4$

Calculate (i) P(*cavity*) (ii) P(*cavity* | *toothache*)

Inference by enumeration

	toothache		¬ toothache	
	catch	¬ catch	catch	¬ catch
cavity	.108	.012	.072	.008
¬ cavity	.016	.064	.144	.576

Denominator can be viewed as a normalization constant α

$\mathbf{P}(Cavity \mid toothache) = \alpha \times \mathbf{P}(Cavity, toothache)$

$= \alpha \times [\mathbf{P}(Cavity, toothache, catch) + \mathbf{P}(Cavity, toothache, \neg catch)]$

$= \alpha \times [\langle 0.108, 0.016\rangle + \langle 0.012, 0.064\rangle]$

$= \alpha \times \langle 0.12, 0.08\rangle = \langle 0.6, 0.4\rangle$

General idea: compute distribution on query variable by fixing evidence variables and summing over hidden variables

Inference by enumeration

Typically, we are interested in

the posterior joint distribution of the query variables **Y**

given specific values **e** for the evidence variables **E**

Let the hidden variables be **H** = **X** - **Y** - **E**

Then the required summation of joint entries is done by summing out the hidden variables:

$$\mathbf{P}(\mathbf{Y} \mid \mathbf{E} = \mathbf{e}) = \alpha \mathbf{P}(\mathbf{Y},\mathbf{E} = \mathbf{e}) = \alpha \Sigma_{\mathrm{h}} \mathbf{P}(\mathbf{Y},\mathbf{E}= \mathbf{e}, \mathbf{H} = \mathbf{h})$$

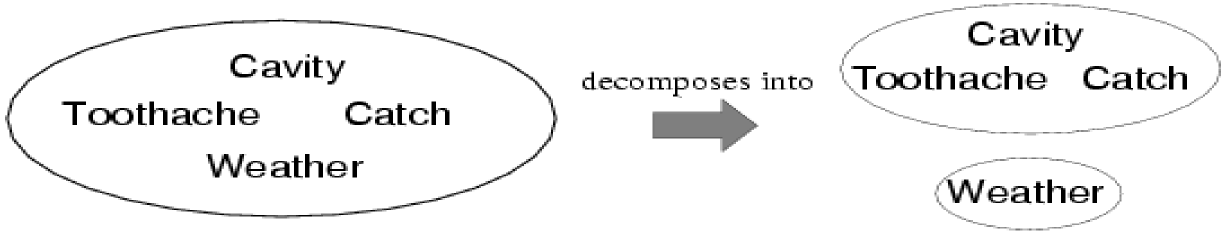
The terms in the summation are joint entries because **Y**, **E** and **H** together exhaust the set of random variables

Obvious problems:

1. Worst-case time complexity $O(d^n)$ where d is the largest arity
2. Space complexity $O(d^n)$ to store the joint distribution
3. How to find the numbers for $O(d^n)$ entries

Independence

A and *B* are independent iff $\mathbf{P}(A|B) = \mathbf{P}(A)$ or $\mathbf{P}(B|A) = \mathbf{P}(B)$ or $\mathbf{P}(\mathrm{A, B}) = \mathbf{P}(A)\,\mathbf{P}(B)$



$\mathbf{P}(Toothache, Catch, Cavity, Weather)$
$= \mathbf{P}(Toothache, Catch, Cavity)\,\mathbf{P}(Weather)$

32 entries reduced to 12;

for *n* independent biased coins, $O(2^n) \rightarrow O(n)$

Absolute independence powerful but rare

Dentistry is a large field with hundreds of variables, none of which are independent. What to do?

Bayes' Rule

Product rule $P(a \wedge b) = P(a \mid b)\, P(b) = P(b \mid a)\, P(a)$

$\Rightarrow$ Bayes' rule: $P(a \mid b) = P(b \mid a)\, P(a) / P(b)$

or in distribution form

$$\mathbf{P}(Y|X) = \mathbf{P}(X|Y)\, \mathbf{P}(Y) / \mathbf{P}(X) = \alpha \mathbf{P}(X|Y)\, \mathbf{P}(Y)$$

- E.g., let m be meningitis, s be stiff neck: Let $P(s \mid m)=0.8$, $P(m)=0.0001$, $P(s)=0.1$

  $P(m|s) = P(s|m)\, P(m) / P(s) = 0.8 \times 0.0001 / 0.1 = 0.0008$

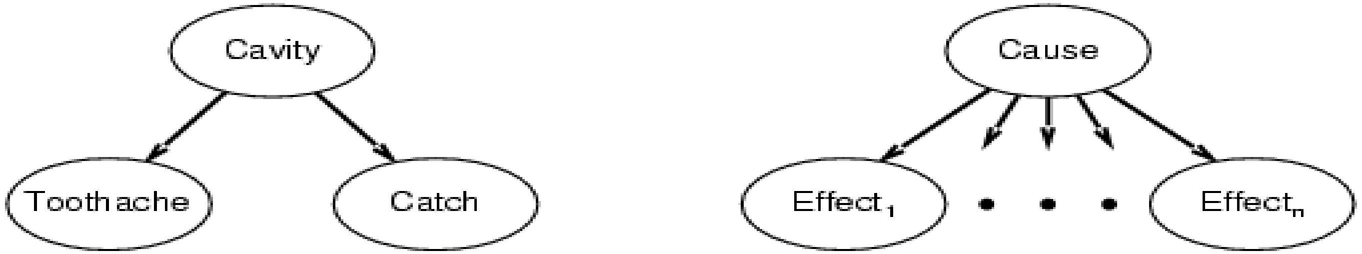
- Note: even though the probability of having a stiff neck given meningitis is very large (0.8), the posterior probability of meningitis given a stiff neck is still very small

- $P(s|m)$ is more 'robust' than $P(m|s)$. Imagine a new disease appeared which would also cause a stiff neck, then $P(m|s)$ changes but $P(s|m)$ not.

Bayes' Rule and conditional independence

This is an example of a naïve Bayes model:

$\mathbf{P}(\text{Cause}, \text{Effect}_1, \ldots, \text{Effect}_n) = \mathbf{P}(\text{Cause})\ \pi_i \mathbf{P}(\text{Effect}_i|\text{Cause})$



Total number of parameters is linear in *n*

A naive Bayes classifier computes: P(cause|effect1, effect2...)

Graphical Models

If no assumption of independence is made, then an exponential number of parameters must be estimated for sound probabilistic inference.

No realistic amount of training data is sufficient to estimate so many parameters.

If a blanket assumption of conditional independence is made, efficient training and inference is possible, but such a strong assumption is rarely warranted.

Graphical models use directed or undirected graphs over a set of random variables to explicitly specify variable dependencies and allow for less restrictive independence assumptions while limiting the number of parameters that must be estimated.

- **Bayesian Networks**: Directed acyclic graphs that indicate causal structure.
- **Markov Networks**: Undirected graphs that capture general dependencies.

Bayesian network

A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions

Syntax:

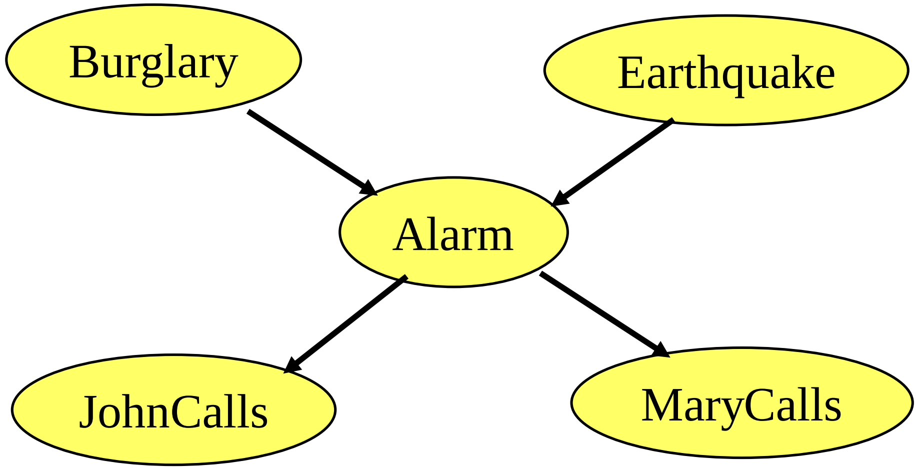
- a set of nodes, one per variable
- a directed, acyclic graph (link $\approx$ "directly influences")
- a conditional distribution for each node given its parents:
  $\mathbf{P}(X_i | Parents(X_i))$

In the simplest case, the conditional distribution is represented as a conditional probability table (CPT) giving the distribution over X_i for each combination of parent values

Bayesian Networks

Directed Acyclic Graph (DAG)

- Nodes are random variables
- Edges indicate causal influences



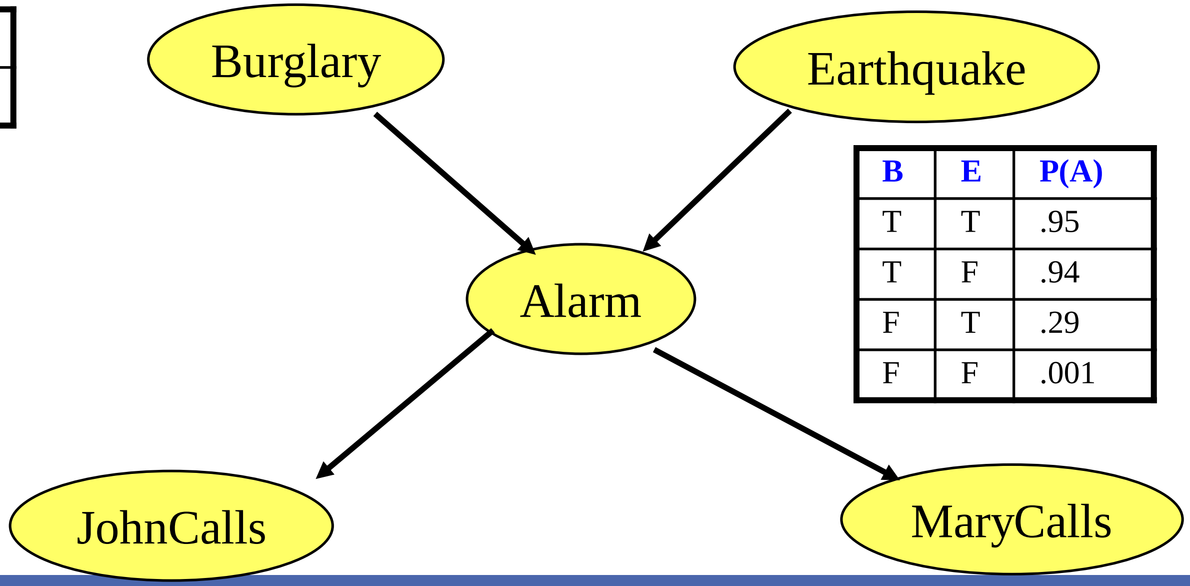
Conditional Probability Tables

Each node has a **conditional probability table** (**CPT**) that gives the probability of each of its values given every possible combination of values for its parents (conditioning case).

- Roots (sources) of the DAG that have no parents are given prior probabilities.

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

Joint Distributions for Bayes Nets

A Bayesian Network implicitly defines a joint distribution.

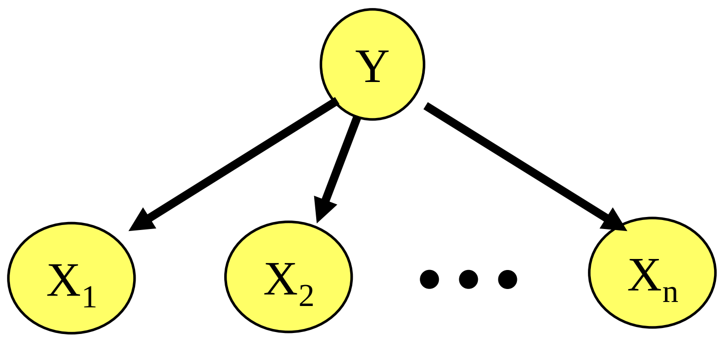
$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i \mid \text{Parents}(X_i))$$

- Example

$$\begin{aligned} & P(J \wedge M \wedge A \wedge \neg B \wedge \neg E) \\ & = P(J \mid A) P(M \mid A) P(A \mid \neg B \wedge \neg E) P(\neg B) P(\neg E) \\ & = 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 = 0.00062 \end{aligned}$$

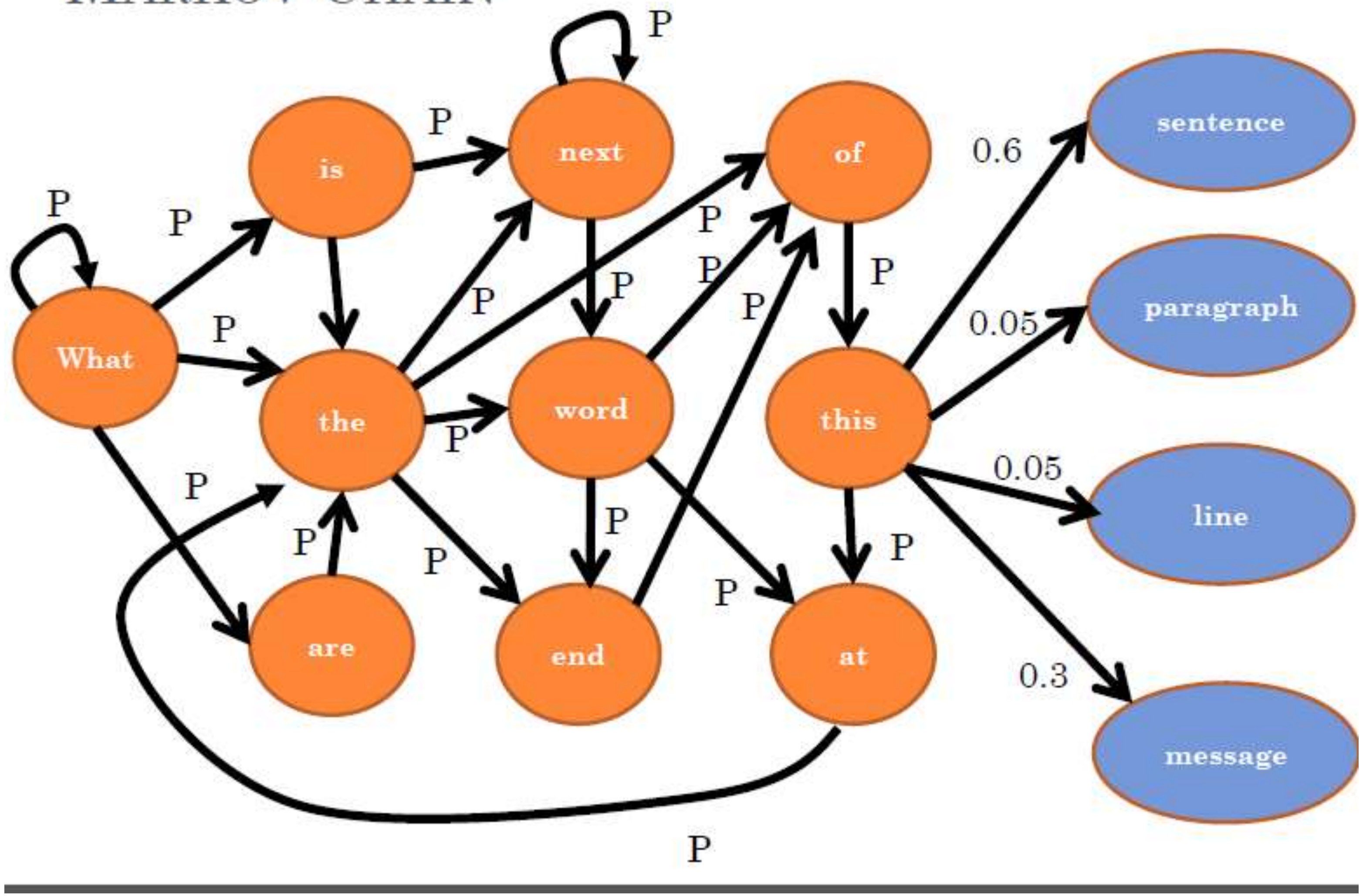
Naïve Bayes as a Bayes Net

Naïve Bayes is a simple Bayes Net



- Priors $P(Y)$ and conditionals $P(X_i|Y)$ for Naïve Bayes provide CPTs for the network.

MARKOV CHAIN



WHAT IS A MARKOV MODEL?

- A Markov Model is a stochastic model which models temporal or sequential data, i.e., data that are ordered.
- It provides a way to model the dependencies of current information (e.g. weather) with previous information.
- It is composed of states, transition scheme between states, and emission of outputs (discrete or continuous).

- Several goals can be accomplished by using Markov models:
  - Learn statistics of sequential data.
  - Do prediction or estimation.
  - Recognize patterns.

Markov Model

Design a Markov Model to predict the weather of tomorrow using previous information of the past days.

•Our model has only 3 states: $S=S1,S2,S3$, and the name of each state is $S1=Sunny$, $S2=Rainy$, $S3=Cloudy$.

•To establish the transition probabilities relationship between states we will need to collect data.

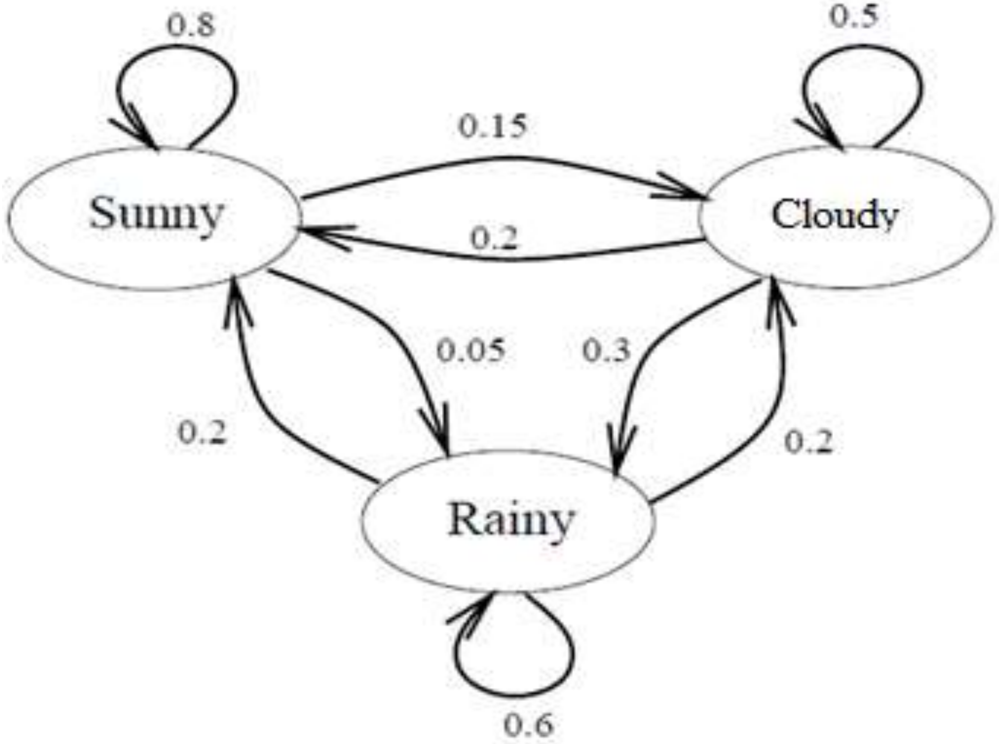
Assume the data produces the following transition probabilities:

Markov Models

$P(Sunny|Sunny)$=0.8
$P(Rainy|Sunny)$=0.05
$P(Cloudy|Sunny)$=0.15

$P(Sunny|Rainy)$=0.2
$P(Rainy|Rainy)$=0.6
$P(Cloudy|Rainy)$=0.2

$P(Sunny|Cloudy)$=0.2
$P(Rainy|Cloudy)$=0.3
$P(Cloudy|Cloudy)$=0.5



Markov Models

Let's say we have a sequence: Sunny, Rainy, Cloudy, Cloudy, Sunny, Sunny, Sunny, Rainy, ….; so, in a day we can be in any of the three states.

- We can use the following state sequence notation:

  $q1$, $q2$, $q3$, $q4$, $q5$,.., where $qi \epsilon \{Sunny, Rainy, Cloudy\}$.

- In order to compute the probability of tomorrow's weather we can use the Markov property:

$$P(q_1, \dots, q_n) = \prod_{i=1}^{n} P(q_i | q_{i-1})$$

Markov Models

Exercise 1: Given that today is Sunny, what's the probability that tomorrow is Sunny and the next day Rainy?

$P(q_2, q_3 | q_1) = P(q_2 | q_1)\, P(q_3 | q_1, q_2)$
$= P(q_2 | q_1)\, P(q_3 | q_2)$
$= P(Sunny \mid Sunny)\, P(Rainy \mid Sunny)$
=0.8(0.05)
=0.04

Exercise 2: Assume that yesterday's weather was Rainy, and today is Cloudy, what is the probability that tomorrow will be Sunny?

Problems with Logical Reasoning

Brittleness: one axiom/fact wrong, system can prove anything (Correctness)

Large proof spaces (Efficiency)

“Probably” not representable (Representation)

No notion of combining evidence

Doesn’t provide confidence in conclusion.

Expert System

Attempt to model expert decision making in a limited domain

Examples: medical diagnosis, computer configuration, machine fault diagnosis

Requires a willing Expert

Requires knowledge representable as rules

- Doesn’t work for chess

Architecture

Domain Knowledge as “if-then” rules

Inference Engine

- Backward chaining
- Forward chaining

Calculus for combining evidence

- Construct all proofs, not just one

Explanation Facility: can answer “why?”

MYCIN: 1972-1980

50-500 rules, acquired from expert by interviewing. Blood disease diagnosis.

Example rule:

if stain of organism is gramneg and morphology is rod and aerobicity is aerobic then strongly suggestive (.8) that organism is enterocabateriacease.

Rules matched well knowledge in domain: medical papers often present a few rules

Rule = nugget of independent knowledge

Facts are not facts

Morphology is rod requires microscopic evaluation.

- Is a bean shape a rod?
- Is an “S” shape a rod?

Morphology is rod is assigned a confidence.

All “facts” assigned confidences, from 0 to 1.

MYCIN: Shortliffe

Begins with a few facts about patient

- Required by physicians but irrelevant

Backward chains from each possible goal (disease).

Preconditions either match facts or set up new subgoals. Subgoals may involve tests.

Finds all “proofs” and weighs them.

Explains decisions and combines evidence

Worked better than average physician.

Never used in practice.

Methodology used.

Examples and non-examples

Soybean diagnosis

- Expert codified knowledge in form of rules
- System almost as good
- When hundreds of rules, system seems reasonable.

Autoclade placement

- Expert but no codification

Chess

- Experts but no codification in terms of rules

Forward Chaining Interpreter

Repeat

- Apply all the rules to the current facts.
- Each rule firing may add new facts

Until no new facts are added.

Comprehensible

Trace of rule applications that lead to conclusion is explanation. Answers why.

Forward Chaining Example

Facts:

- F1: Ungee gives milk
- F2: Ungee eats meat
- F3: Ungee has hoofs

Rules:

- R1: If X gives milk, then it is a mammal
- R2: If X is a mammal and eats meat, then carnivore.
- R3: If X is a carnivore and has hoofs, then ungulate

Easy to see: Ungee is ungulate.

Backward Chaining

Start with Goal G1: is Ungee an ungulate?

G1 matches conclusion of R3

Sets up premises as subgoals

- G2: Ungee is carnivore
- G3: Ungee has hoofs

G3 matches fact F3 so true.

G2 matches conclusion of R2. etc.

The good and the bad

Forward chaining allows you to conclude anything

Forward chaining is expensive

Backward chaining requires known goals.

Premises of backward chaining directs which facts (tests) are needed.

Rule trace provides explanation.

Simple Confidence Calculus

This will yield an intuitive degree of belief in system conclusion.

To each fact, assign a confidence or degree of belief. A number between 0 and 1.

To each rule, assign a rule confidence: also a number between 0 and 1.

Simple Confidence Calculus

Confidence of premise of a rule =

- Minimum confidence of each condition
- Intuition: strength of argument is weakest link

Confidence in conclusion of a rule =

- (confidence in rule premise)*(confidence in rule)

Confidence in conclusion from several rules: r1,r2,..rm with confidences c1,c2,..cm =

- c1 @ c2 @... cm
- Where x @ y is   1- (1-x)*(1-y).

And now with confidences

Facts:

- F1: Ungee gives milk: .9
- F2: Ungee eats meat: .8
- F3: Ungee has hoofs: .7

Rules:

- R1: If X gives milk, then it is a mammal: .6
- R2: If X is a mammal and eats meat, then carnivore: .5
- R3: If X has hoofs, then X is carnivore: .4

Simple Confidence Calculus

R1 with F1: Ungee is mammal. (F4)

Confidence F4: C(F4) = .9*.6 = .54

R2 using F2 and F4 yields: Ungee is carnivore (F5).

C(F5) from R2 = min(.54, .8)*.5 = .27

R3 using F3 conclude F5 from R3

C(F5) from R3 = .7*.4 = .28

C(F5) from R3 and R2 = .27 @ .28 = 1 –(1-.28)*(1-.27) = .48

MYCIN

History and Overview

MYCIN Architecture

Consultation System

- Knowledge Representation & Reasoning

Explanation System

Knowledge Acquisition

Results, Conclusions

History

Thesis Project by Shortliffe @ Stanford

Davis, Buchanan, van Melle, and others

- Stanford Heuristic Programming Project
- Infectious Disease Group, Stanford Medical

Project Spans a Decade

- Research started in 1972
- Original implementation completed 1976
- Research continues into the 80’s

Tasks and Domain

Disease DIAGNOSIS and Therapy SELECTION

Advice for non-expert physicians with time considerations and <u>in</u>complete evidence on:

- Bacterial infections of the blood
- Expanded to meningitis and other ailments

System Goals

Utility

- Be useful, to attract assistance of experts
- Demonstrate competence
- Fulfill domain need (i.e. penicillin)

Flexibility

- Domain is complex, variety of knowledge types
- Medical knowledge rapidly evolves, must be easy to maintain K.B.

System Goals (continued)

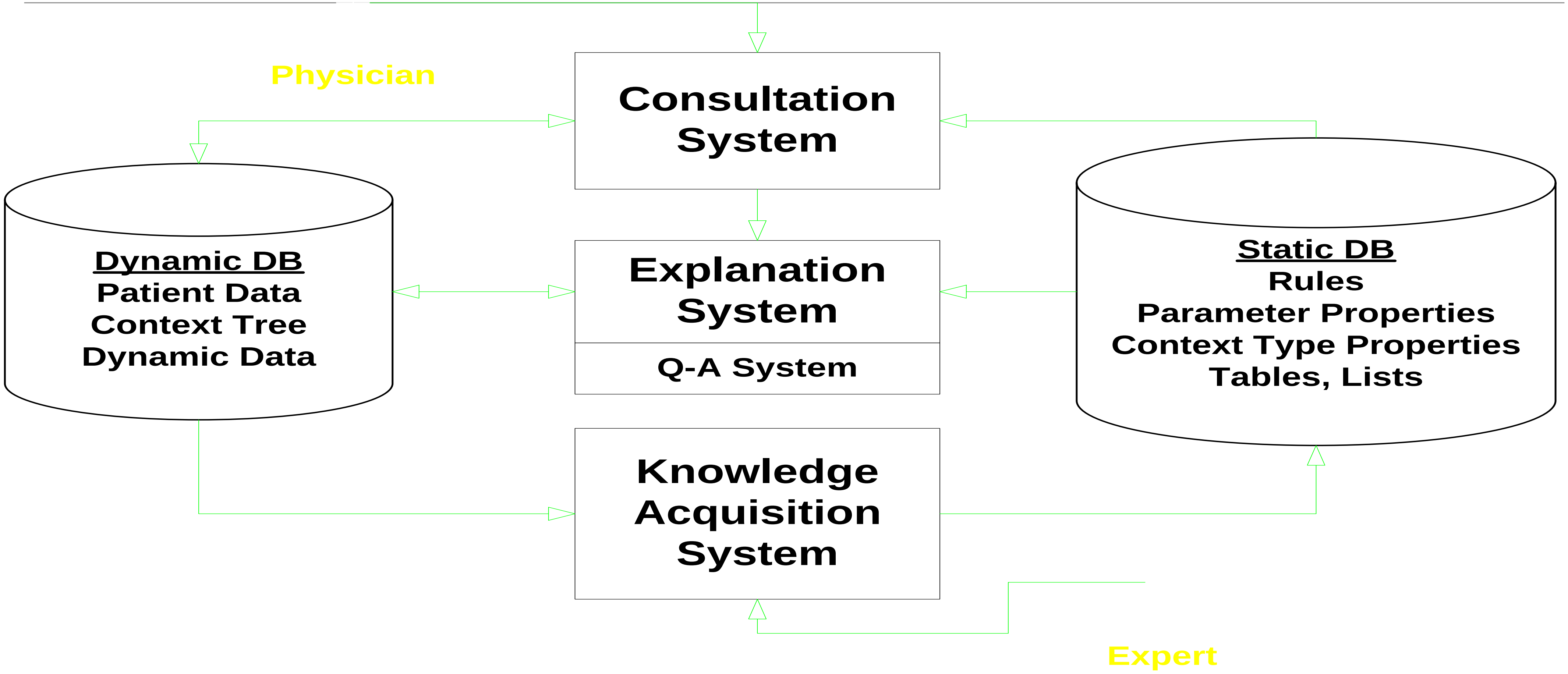
Interactive Dialogue

- Provide coherent explanations (symbolic reasoning paradigm)
- Allow for real-time K.B. updates by experts

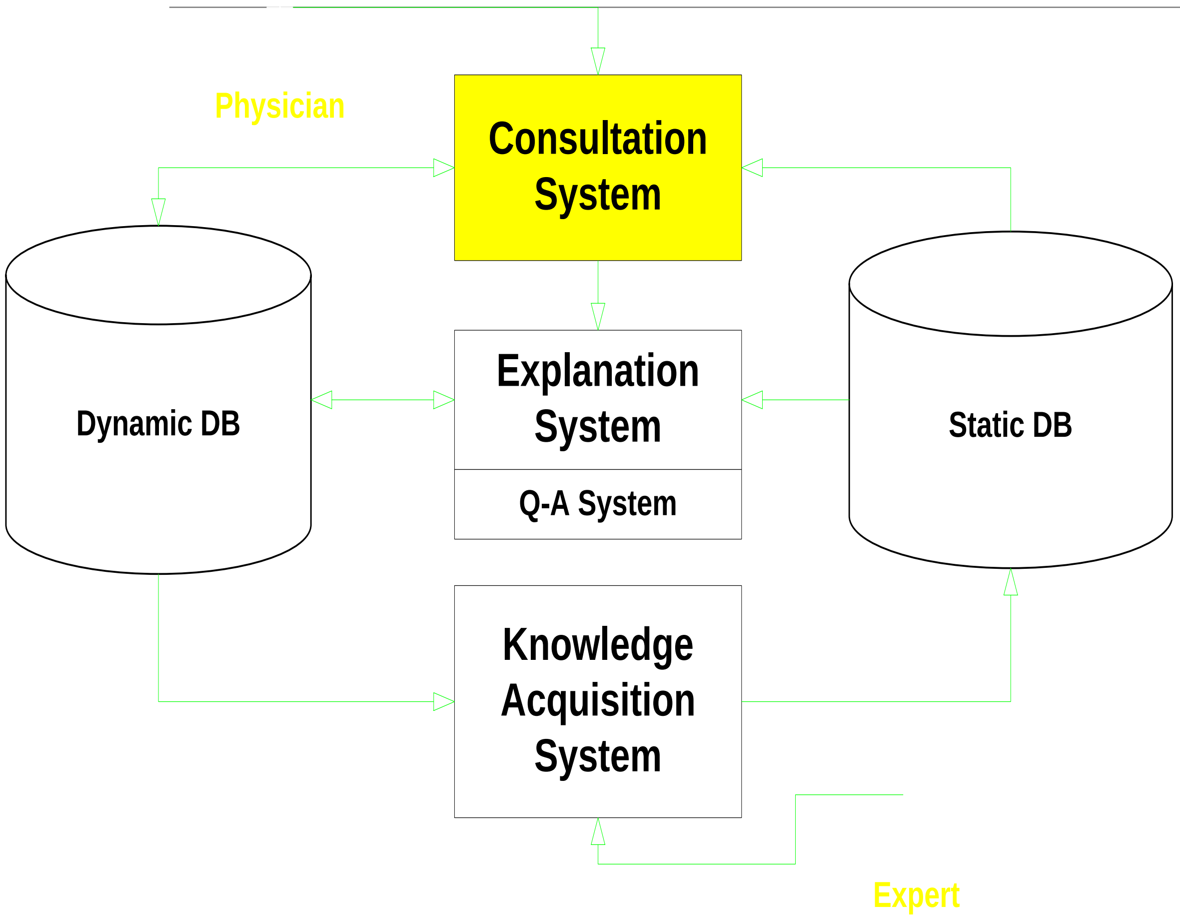
Fast and Easy

- Meet time constraints of the medical field

MYCIN Architecture



Consultation System



Performs Diagnosis and Therapy Selection

Control Structure reads Static DB (rules) and read/writes to Dynamic DB (patient, context)

Linked to Explanations

Terminal interface to Physician

Consultation System

User-Friendly Features:

- Users can request rephrasing of questions
- Synonym dictionary allows latitude of user responses
- User typos are automatically fixed

Questions are asked when more data is needed

- If data cannot be provided, system ignores relevant rules

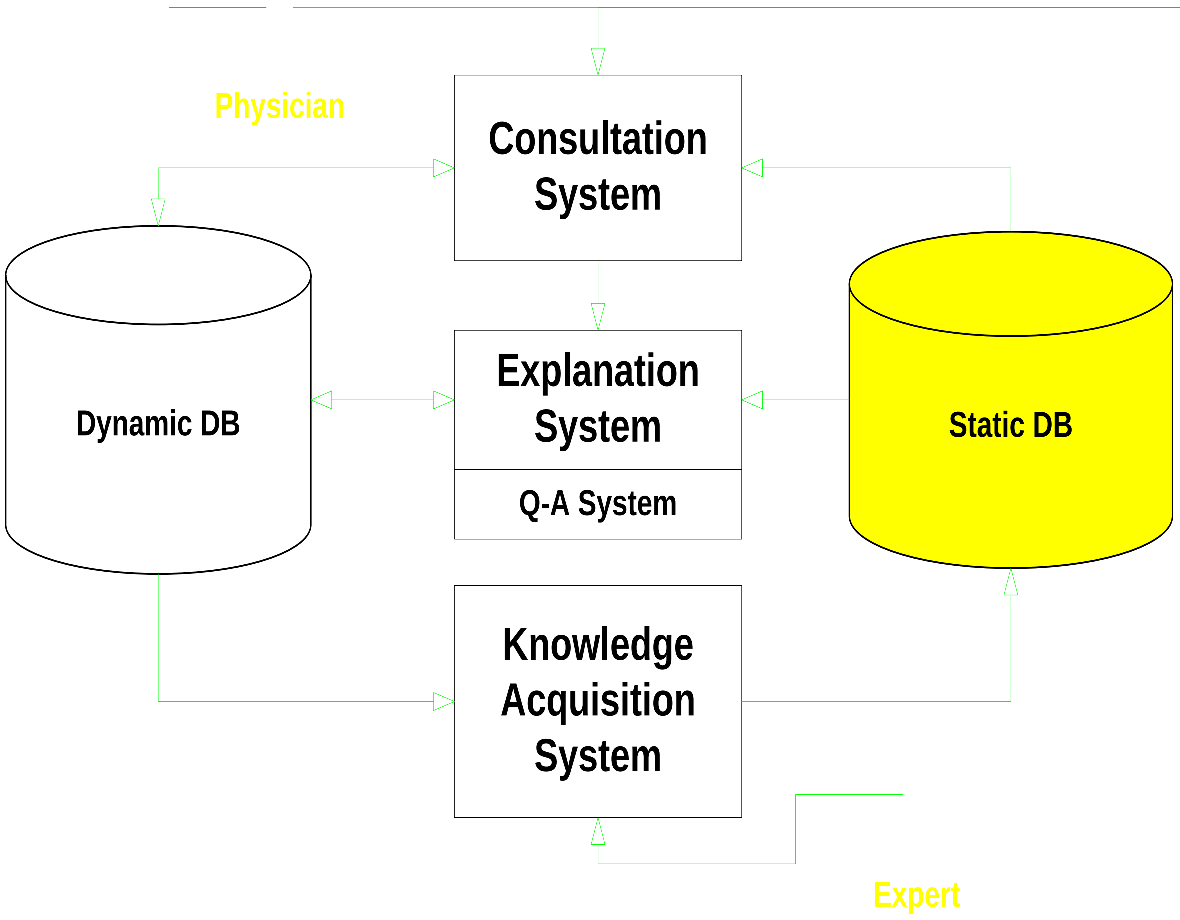
Consultation “Control Structure”

Goal-directed Backward-chaining Depth-first Tree Search

High-level Algorithm:

1. Determine if Patient has significant infection
2. Determine likely identity of significant organisms
3. Decide which drugs are potentially useful
4. Select best drug or coverage of drugs

Static Database



- Rules
- Meta-Rules
- Templates
- Rule Properties
- Context Properties
- Fed from Knowledge Acquisition System

Production Rules

Represent Domain-specific Knowledge

Over 450 rules in MYCIN

Premise-Action (If-Then) Form:

<predicate function><object><attrib><value>

Each rule is completely modular, all relevant context is contained in the rule with explicitly stated premises

MYCIN P.R. Assumptions

Not every domain can be represented, requires formalization (EMYCIN)

Only small number of simultaneous factors (more than 6 was thought to be unwieldy)

IF-THEN formalism is suitable for Expert Knowledge Acquisition and Explanation sub-systems

Judgmental Knowledge

Inexact Reasoning with Certainty Factors (CF)

CF are not Probability!

Truth of a Hypothesis is measured by a sum of the CFs

- Premises and Rules added together
- Positive sum is confirming evidence
- Negative sum is disconfirming evidence

Sub-goals

At any given time MYCIN is establishing the value of some parameter by sub-goaling

Unity Paths: a method to bypass sub-goals by following a path whose certainty is known (CF==1) to make a definite conclusion

Won't search a sub-goal if it can be obtained from a user first (i.e. lab data)

Preview Mechanism

Interpreter reads rules before invoking them

Avoids unnecessary deductive work if the sub-goal has already been tested/determined

Ensures self-referencing sub-goals do not enter recursive infinite loops

Meta-Rules

Alternative to exhaustive invocation of all rules

Strategy rules to suggest an approach for a given sub-goal

- Ordering rules to try first, effectively pruning the search tree

Creates a search-space with embedded information on which branch is best to take

Meta-Rules (continued)

High-order Meta-Rules (i.e. Meta-Rules for Meta-Rules)

- Powerful, but used limitedly in practice

Impact to the Explanation System:

- (+) Encode Knowledge formerly in the Control Structure
- (-) Sometimes create “murky” explanations

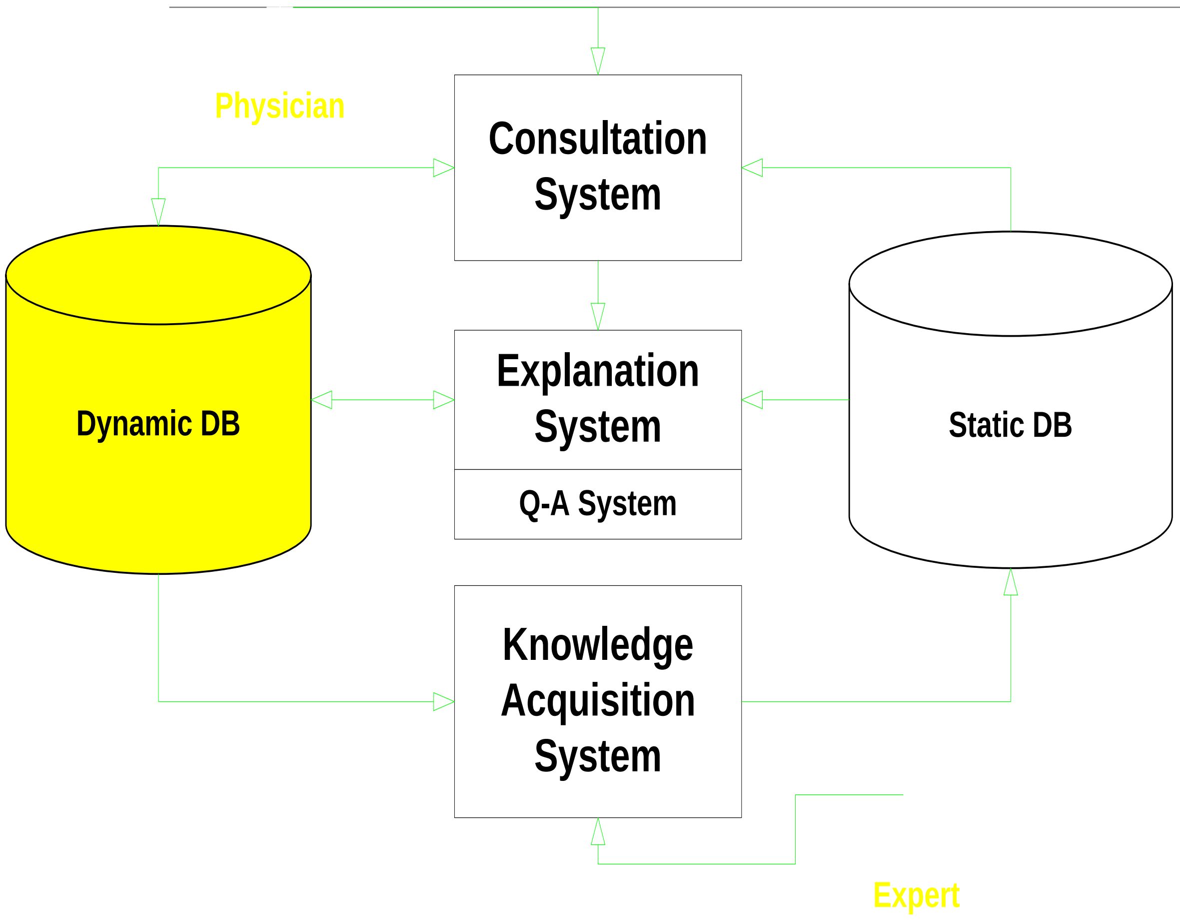
Templates

The Production Rules are all based on Template structures

This aids Knowledge-base expansion, because the system can “understand” its own representations

Templates are updated by the system when a new rule is entered

Dynamic Database



Patient Data

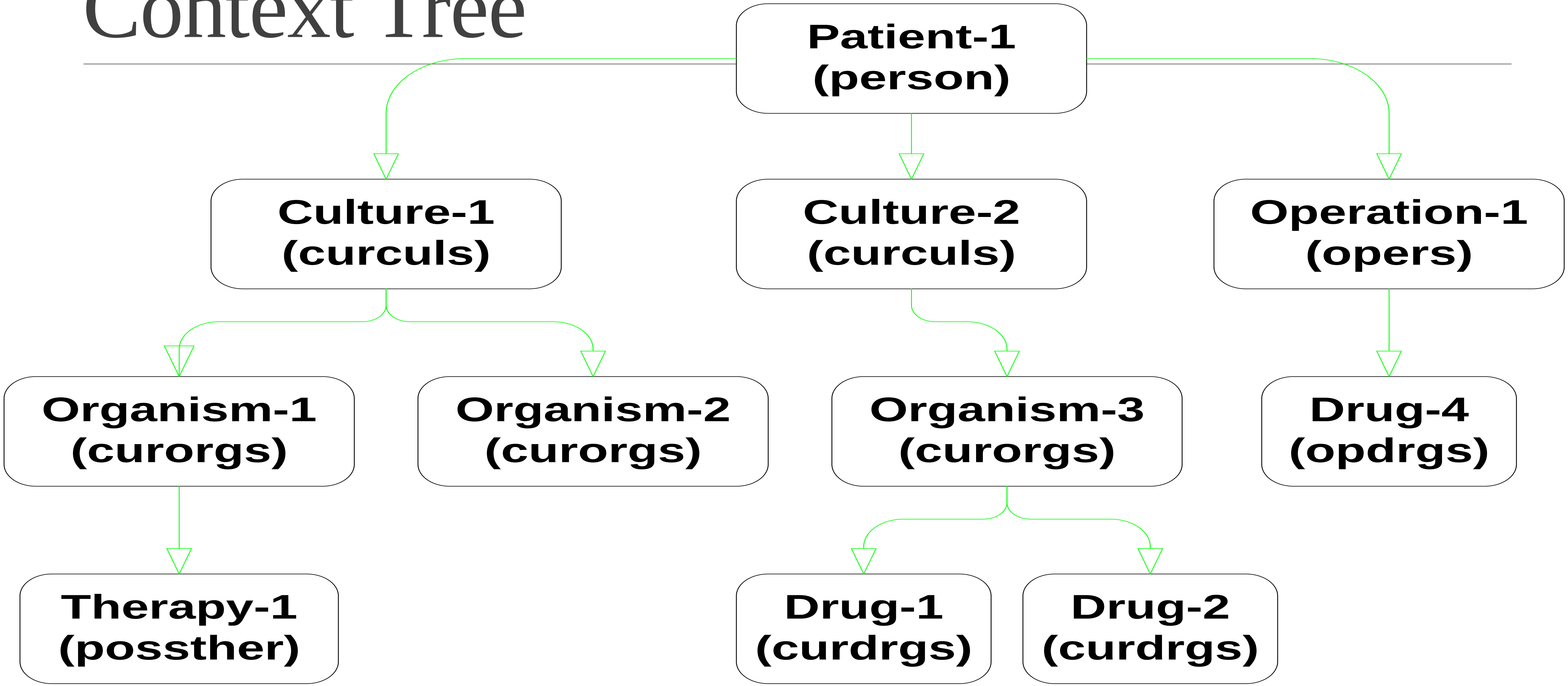
Laboratory Data

Context Tree

Built by Consultation System

Used by Explanation System

Context Tree



Therapy Selection

Plan-Generate-and-Test Process

Therapy List Creation

- Set of specific rules recommend treatments based on the probability (not CF) of organism sensitivity
- Probabilities based on laboratory data
- One therapy rule for every organism

Therapy Selection

Assigning Item Numbers

- Only hypothesis with organisms deemed “significantly likely” (CF) are considered
- Then the most likely (CF) identity of the organisms themselves are determined and assigned an Item Number
- Each item is assigned a probability of likelihood and probability of sensitivity to drug

Therapy Selection

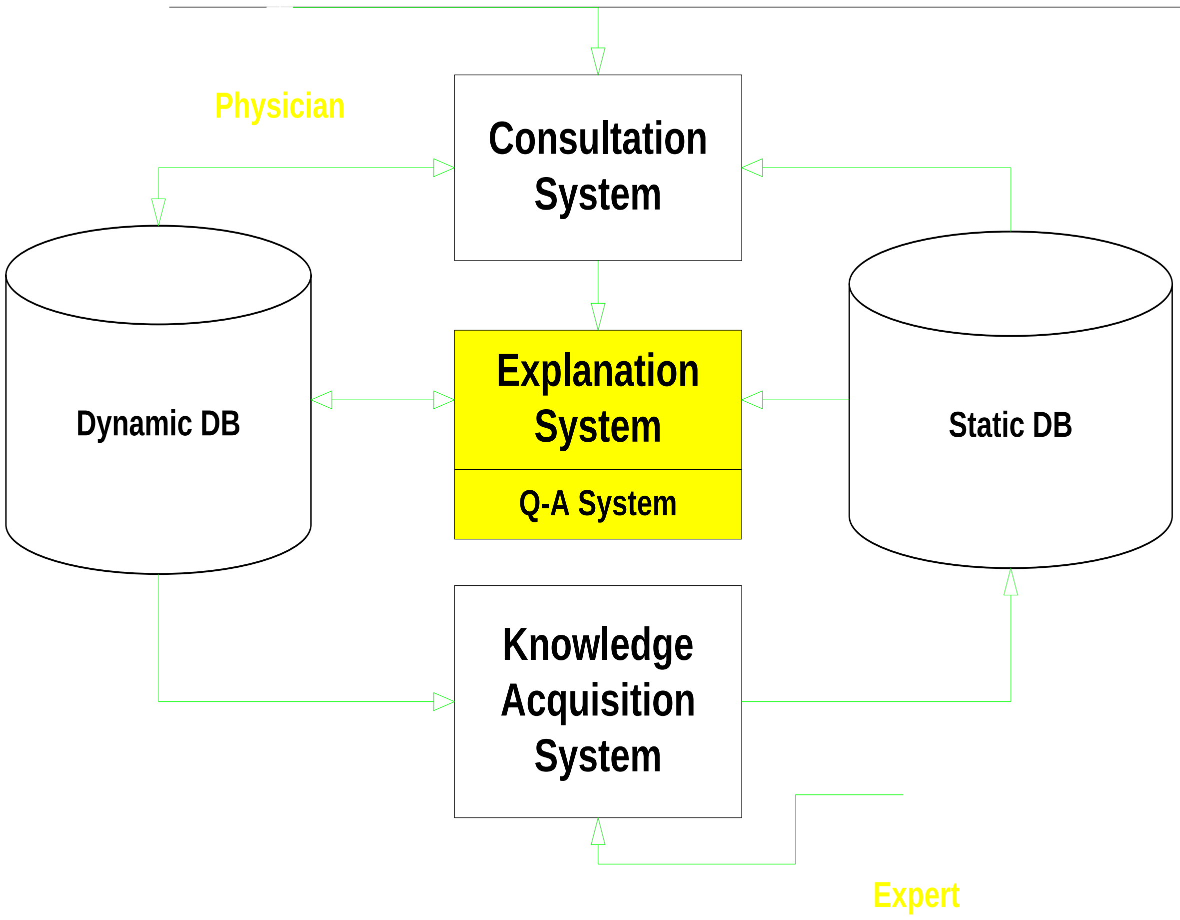
Final Selection based on:

- Sensitivity
- Contraindication Screening
- Using the minimal number of drugs and maximizing the coverage of organisms

Experts can ask for alternate treatments

- Therapy selection is repeated with previously recommended drugs removed from the list

Explanation System



Provides reasoning why a conclusion has been made, or why a question is being asked

Q-A Module

Reasoning Status Checker

Explanation System

Uses a trace of the Production Rules for a basis, and the Context Tree, to provide context

- Ignores Definitional Rules (CF == 1)

Two Modules

- Q-A Module
- Reasoning Status Checker

Q-A Module

Symbolic Production Rules are readable

Each <predicate function> has an associated translation pattern:

```
GRID (THE (2) ASSOCIATED WITH (1) IS KNOWN)

VAL (((2 1)))

PORTAL (THE PORTAL OF ENTRY OF *)

PATH-FLORA (LIST OF LIKELY PATHOGENS)
```

i.e. `(GRID (VAL CNTXT PORTAL) PATH-FLORA)` becomes:

"The list of likely pathogens associated with the portal of entry of the organism is known."

Reasoning Status Checker

Explanation is a tree traversal of the traced rules:

- WHY – moves up the tree
- HOW – moves down (possibly to untried areas)

Question is rephrased, and the rule being applied is explained with the translation patterns

Reasoning Status Checker (Example)

32) Was penicillinase added to this blood culture (CULTURE-1)?

****WHY**

[i.e. WHY is it important to determine whether penicillinase was added to CULTURE-1?]

[3.0] This will aid in determining whether ORGANISM-1 is a contaminant. It has already been established that

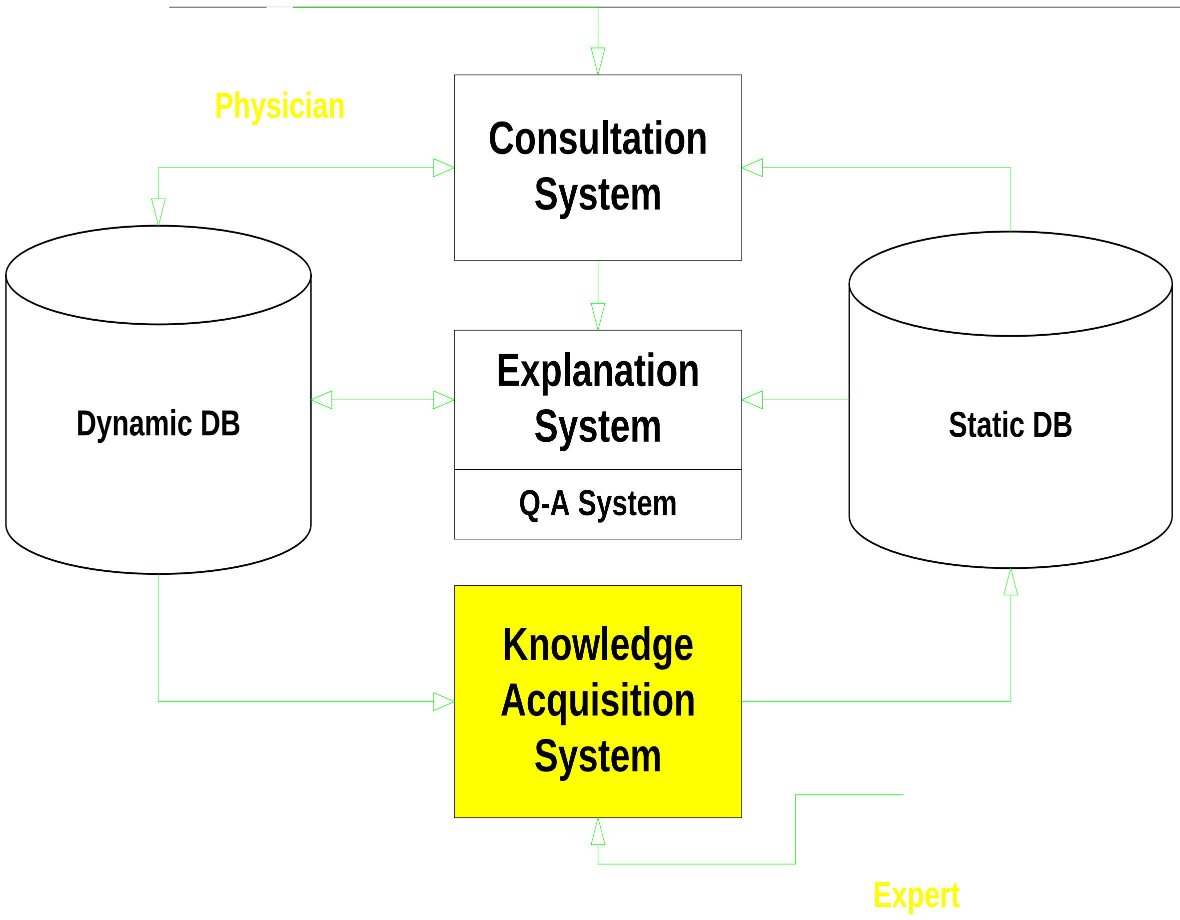
[3.1] the site of CULTURE-1 is blood, and

[3.2] the gram stain of ORGANISM-1 is grampos

Therefore, if

[3.3] penicillinase was added to this blood culture then there is weakly suggestive evidence...

Knowledge Acquisition System



Extends Static DB via Dialogue with Experts

Dialogue Driven by System

Requires minimal training for Experts

Allows for Incremental Competence, NOT an All-or-Nothing model

Knowledge Acquisition

IF-THEN Symbolic logic was found to be easy for experts to learn, and required little training by the MYCIN team

When faced with a rule, the expert must either except it or be forced to update it using the education process

Education Process

1. Bug is uncovered, usually by Explanation process
2. Add/Modify rules using *subset of English* by experts
3. Integrating new knowledge into KB
   - Found to be difficult in practice, requires detection of contradictions, and complex concepts become difficult to express

Results

Never implemented for routine clinical use

Shown to be competent by panels of experts, even in cases where experts themselves disagreed on conclusions

Key Contributions:

- Reuse of Production Rules (explanation, knowledge acquisition models)
- Meta-Level Knowledge Use

THANK YOU FOR LISTENING

ANY QUESTION ?