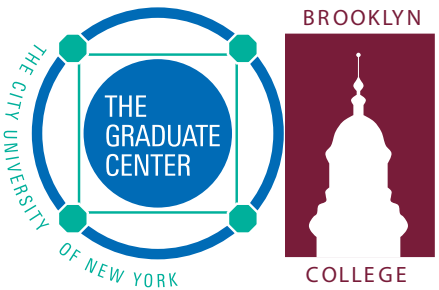
CIS 7414x Expert Systems

Lecture 1: Introduction to Expert Systems

Yuqing Tang

Doctoral Program in Computer Science
The Graduate Center
City University of New York
ytang@cs.gc.cuny.edu

September 1st, 2010



Instructor and course setting

- Instructor: Yuqing Tang
- Email: `ytang@cs.gc.cuny.edu`
- Course web:
  `http://web.cs.gc.cuny.edu/~tang/teachings/cis7414x`
- Course setting
  - $\approx$ 13 lectures
  - Grade policy: Homework $\approx 1/3$, mid-term exam and term project $\approx 1/3$, final exam $\approx 1/3$
- Textbook
  - Bayesian Artificial Intelligence (2004), Kevin B. Korb and Ann E. Nicholson, Chapman and Hall, CRC Press
- Supplemental Materials
  - Expert Systems: Principles and Programming (4th ed.) Joseph C. Giarratano, Gary D. Riley, Thomson Course Technology (from which the first 2 lectures depend on)
  - Artificial Intelligence: A Modern Approach, Stuart Russell, Peter Norvig, Prentice Hall
  - Papers and readings

Course objectives

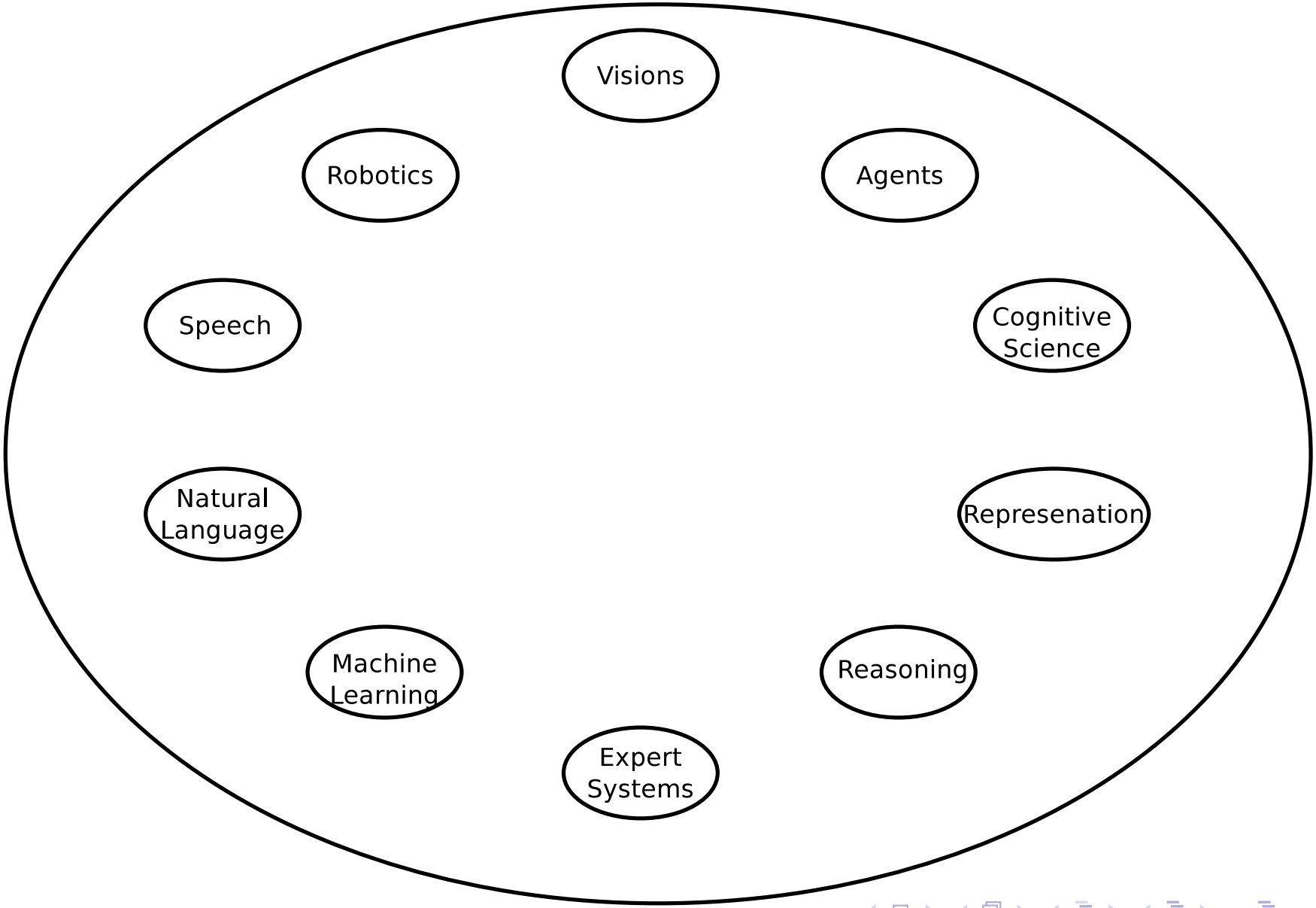
- Contribute to the very big picture
  - A thorough foundation in the discipline of Artificial Intelligence
  - Current trends and advances
  - Capability to select an appropriate approach for the problem in hand
- Learn general expert systems (the first two lectures)
  - The meaning of an expert system
  - The problem domain and knowledge domain
  - The advantage of an expert system
  - The stages in the development of an expert systems
  - The general characteristics of an expert system
  - Challenges
- Focus on probabilistic approaches (the rest of the course)
  - Bayesian Networks
  - Inferences in Bayesian Networks
  - Decision making using Bayesian Networks
  - Knowledge engineering with Bayesian Networks
  - Introduction to machine learning in Bayesian Networks (if time allows)

What is an expert system?

An expert system is a computer system that emulates, or acts in all respects, with the decision-making capabilities of a human expert.

Professor Edward Feigenbaum
Stanford University

Some areas of Artificial Intelligence



Approaches to AI

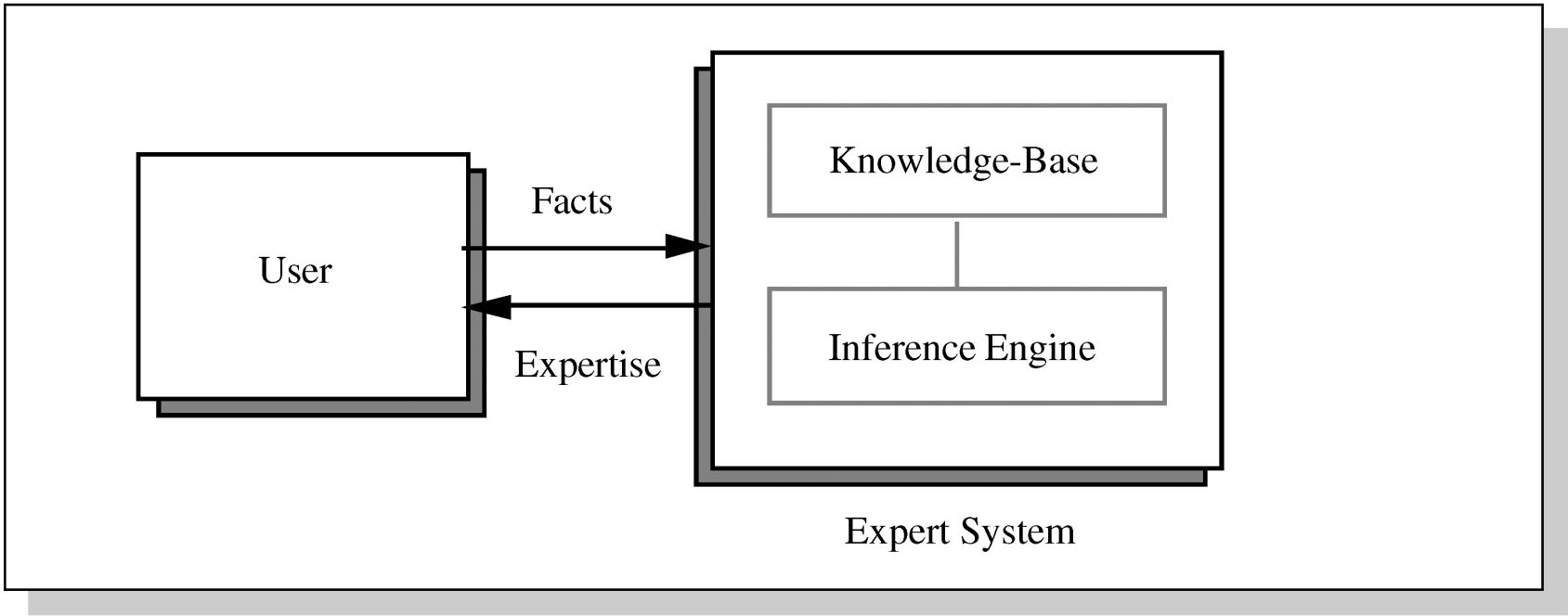
- Symbolic
- Sub-symbolic
- Connectionist
- Statistical
- Intelligent agent (integrating the above and various other approaches, e.g. Market mechanisms)

Expert System Main Components

- Knowledge base
  - Obtainable from books, magazines, knowledgeable persons, etc.
- Inference engine
  - Draws conclusions from the knowledge base

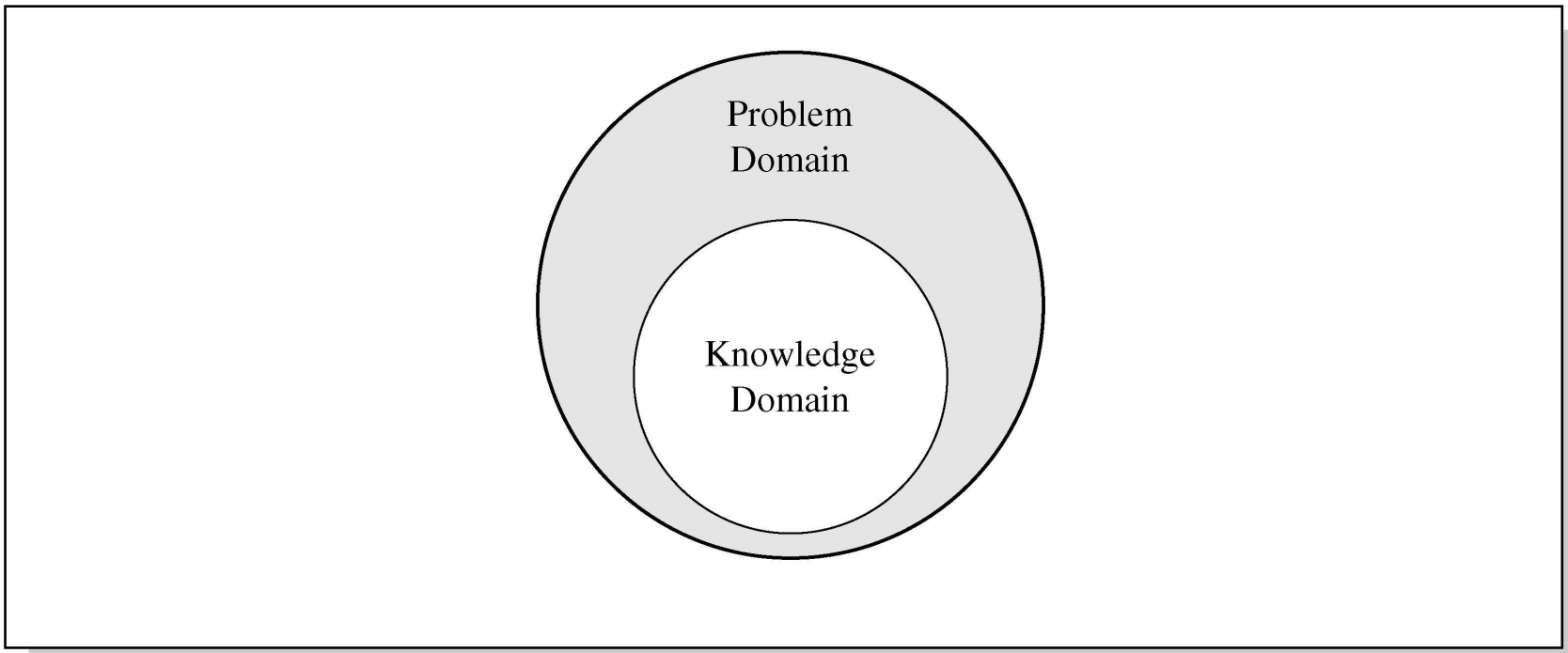
Basic Functions of Expert Systems

Figure 1.2 Basic Function of an Expert System



Problem and knowledge domain relationship

Figure 1.3 Problem and Knowledge Domain Relationship



Advantages of Expert Systems

- Increased availability
- Reduced cost
- Reduced danger
- Performance
- Multiple expertise
- Increased reliability
- Explanation
- Fast response
- Steady, unemotional, and complete responses at all times
- Intelligent tutor
- Intelligent database

Representing the Knowledge

The knowledge of an expert system can be represented in a number of ways, including IF-THEN rules:

IF you are hungry THEN eat

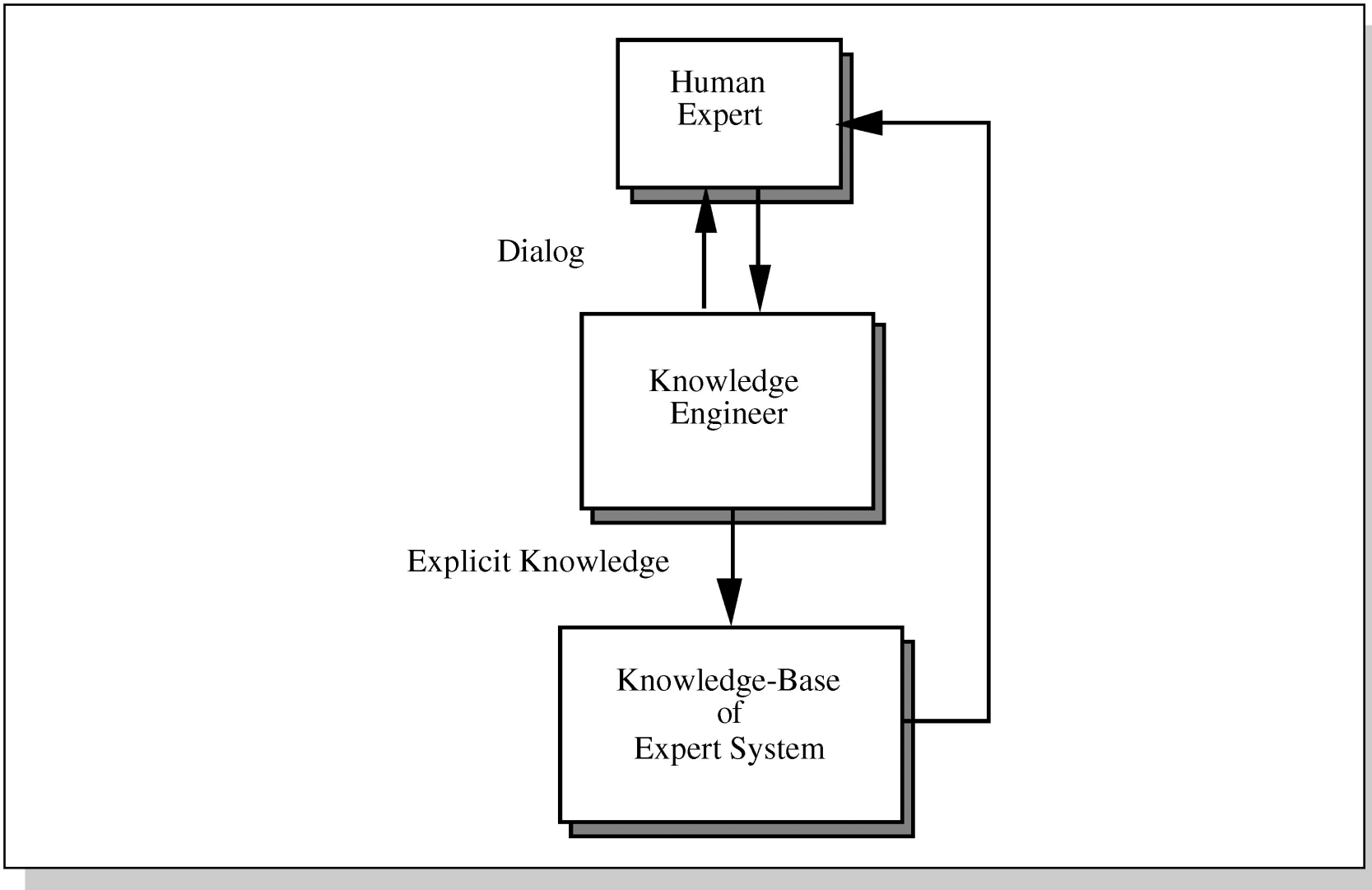
Knowledge Engineering

The process of building an expert system:

- The knowledge engineer establishes a dialog with the human expert to elicit knowledge.
- The knowledge engineer codes the knowledge explicitly in the knowledge base.
- The expert evaluates the expert system and gives a critique to the knowledge engineer.

Development of an Expert System

Figure 1.4 Development of an Expert System



The Role of AI

- An algorithm is an ideal solution guaranteed to yield a solution in a finite amount of time.
- When an algorithm is not available or is insufficient, we rely on artificial intelligence (AI).
- Expert system relies on inference – we accept a "reasonable solution."

The challenge of uncertainty

- Both human experts and expert systems must be able to deal with uncertainty.
- It is easier to program expert systems with shallow knowledge than with deep knowledge.
- Shallow knowledge – based on empirical and heuristic knowledge.
- Deep knowledge – based on basic structure, function, and behavior of objects.

Limitations of Expert Systems

- Typical expert systems cannot generalize through analogy to reason about new situations in the way people can.
- A knowledge acquisition bottleneck results from the time-consuming and labor intensive task of building an expert system.

Early Expert Systems

- DENDRAL – used in chemical mass spectroscopy to identify chemical constituents
- MYCIN – medical diagnosis of illness
- DIPMETER – geological data analysis for oil
- PROSPECTOR – geological data analysis for minerals
- XCON/R1 – configuring computer systems

Broad Classes of Expert Systems

Class	General Area
Configuration	Assemble proper components of a system in the proper way.
Diagnosis	Infer underlying problems based on observed evidence.
Instruction	Intelligence teaching so that a student can ask *why*, *how*, and *what if* questions just as if a human were teaching.
Interpretation	Explain observed data.
Monitoring	Compares observed data to expected data to judge performance
Predicting	Predict the outcome of a given situation.
Planning	Devise actions to yield a desired outcome
Remedy	Prescribe treatment for a problem.
Control	Regulate a process. May require interpretation, diagnosis, monitoring, predicting, planning, and remedies.

Problems with Algorithmic Solutions

- Conventional computer programs generally solve problems having algorithmic solutions.
- Algorithmic languages include C, Java, and C#.
- Classical AI languages include LISP and PROLOG.

Considerations for Building Expert Systems

- Can the problem be solved effectively by conventional programming?
- Is there a need and a desire for an expert system?
- Is there at least one human expert who is willing to cooperate?
- Can the expert explain the knowledge to the knowledge engineer can understand it?
- Is the problem-solving knowledge mainly heuristic and uncertain?

Languages, Shells, and Tools

- Expert system languages are post-third generation.
- Procedural languages (e.g., C) focus on techniques to represent data.
- More modern languages (e.g., Java) focus on data abstraction.
- Expert system languages (e.g. CLIPS) focus on ways to represent knowledge.

Expert systems Vs conventional programs

Characteristic	Conventional Program	Expert System
Control by...	Statement order	Inference engine
Control and data	Implicit integration	Explicit separation
Control strength	Strong	Weak
Solution by...	Algorithm	Rules and inference
Solution search	Small or none	Large
Problem solving	Algorithm is correct	Rules
Input	Assumed correct	Incomplete, incorrect
Unexpected input	Difficult to deal with	Very responsive
Output	Always correct	Varies with problem
Explanation	None	Usually
Applications	Numeric, file, and text	Symbolic reasoning
Execution	Generally sequential	Opportunistic rules
Program design	Structured design	Little or no structure
Modifiability	Difficult	Reasonable
Expansion	Done in major jumps	incremental

Elements of an Expert System

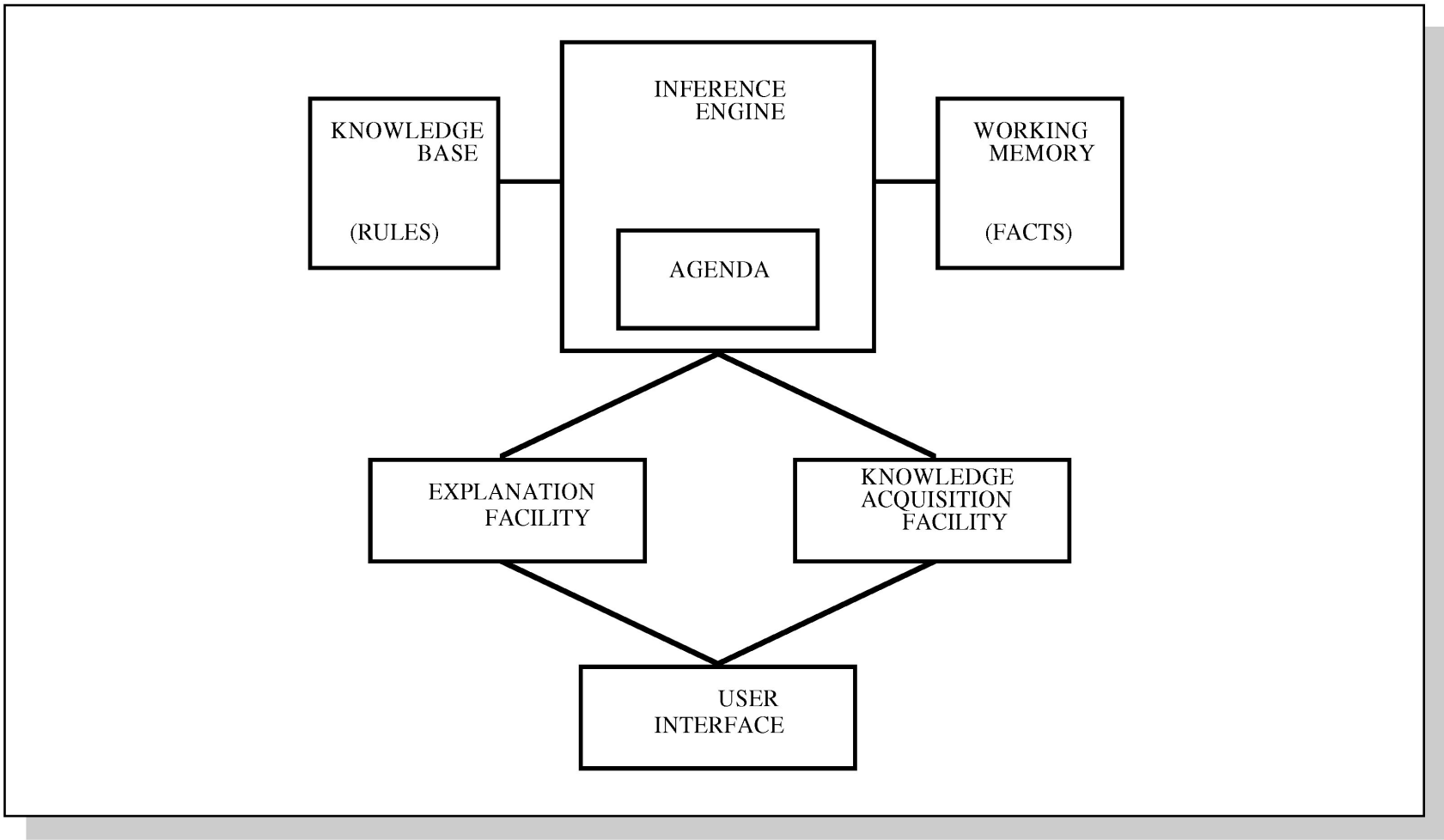
- User interface – mechanism by which user and system communicate.
- Explanation facility – explains reasoning of expert system to user.
- Working memory – global database of facts used by rules.
- Inference engine – makes inferences deciding which rules are satisfied and prioritizing.
- Agenda – a prioritized list of rules created by the inference engine, whose patterns are satisfied by facts or objects in working memory.
- Knowledge acquisition facility – automatic way for the user to enter knowledge in the system bypassing the explicit coding by knowledge engineer.
- Knowledge Base – includes the rules of the expert system

Production Rules

- Knowledge base is also called production memory.
- Production rules can be expressed in IF-THEN pseudocode format.
- In rule-based systems, the inference engine determines which rule antecedents are satisfied by the facts.

Structure of a Rule-Based Expert System

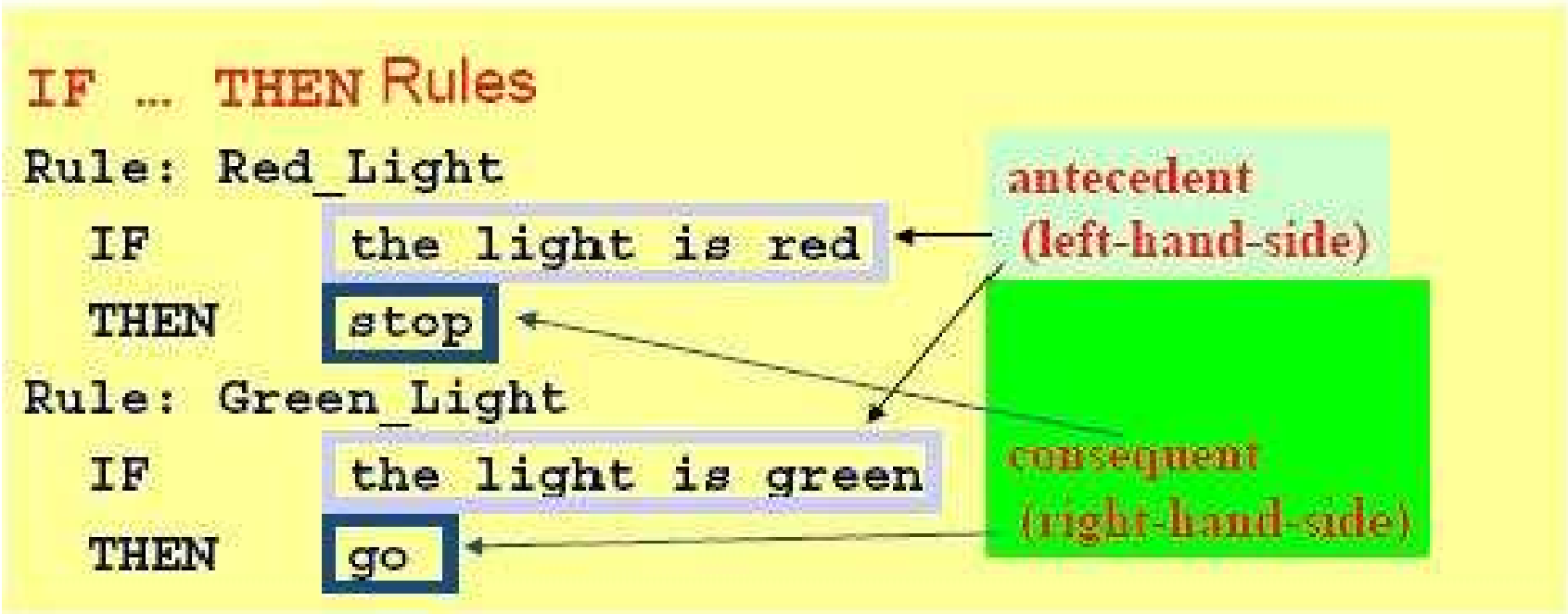
Figure 1.6 Structure of a Rule-Based Expert System

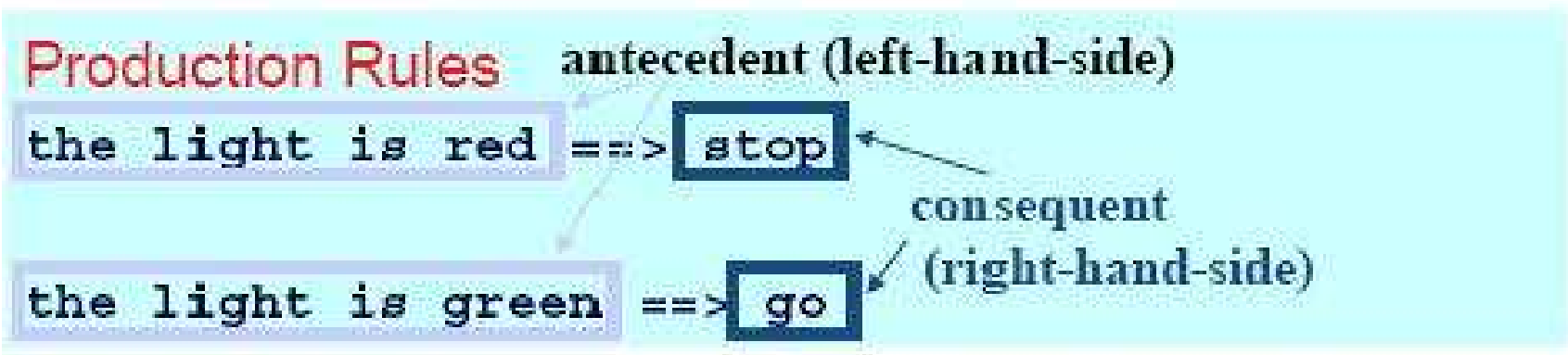


Rule-Based Expert Systems

- Knowledge is encoded as *IF . . . THEN* rules
  - these rules can also be written as *production rules*
- The inference engine determines which rule antecedents are satisfied
  - The left-hand side mush "match" a fact in the working memory
- Satisfied rules are placed on the agenda
- Rules on the agenda can be activated ("fired")
  - An activated rule may generate new facts through its right-hand side
  - The activation of one rule may subsequently cause the activation of other rules

Example rules

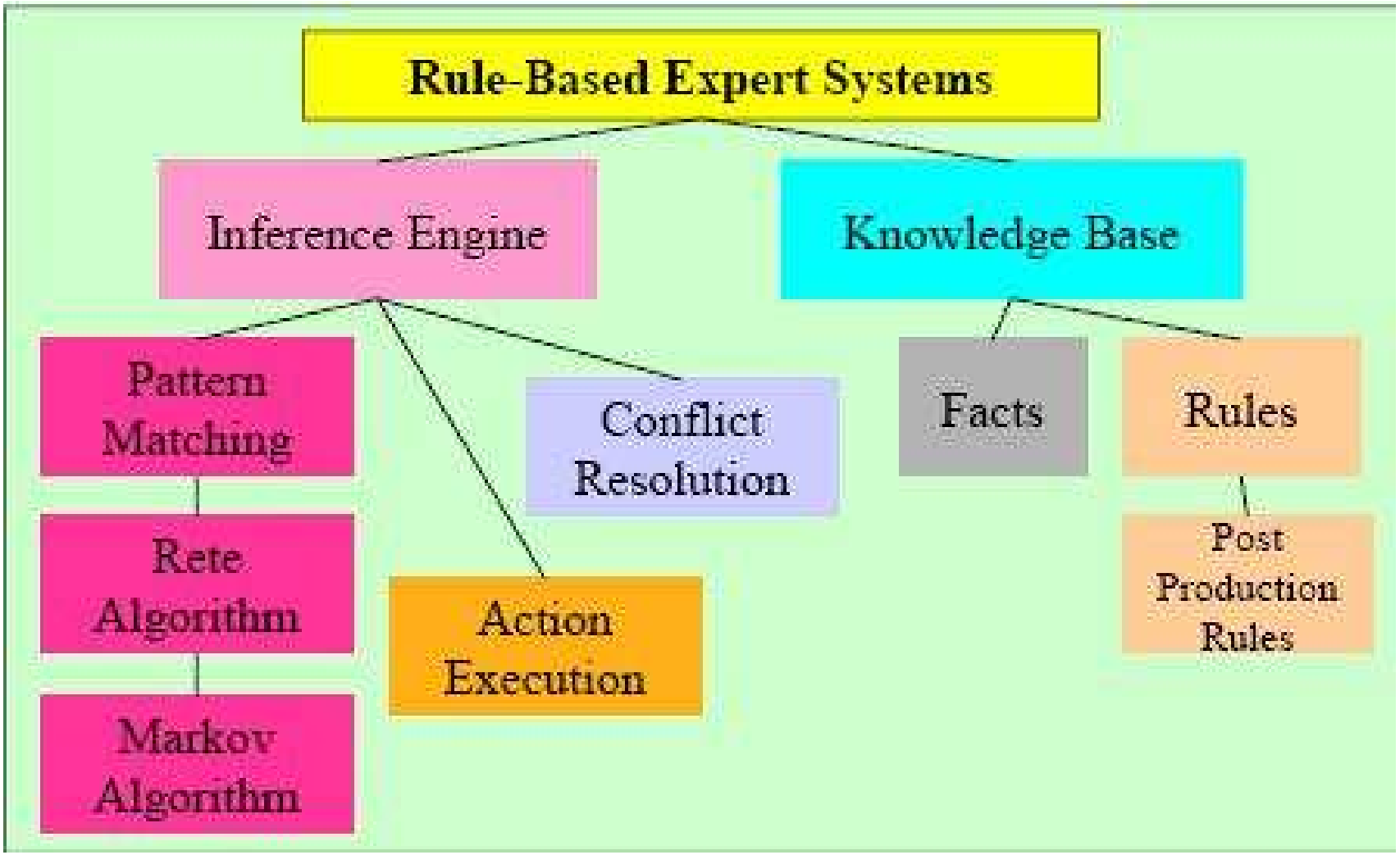




Inference Engine Cycle

- The inference engine determines the execution of the rules by the following cycle:
  - Conflict resolution
    - ★ Select the rule with the highest priority from the agenda
  - Execution (act)
    - ★ Perform the actions on the consequent of the selected rule
    - ★ Remove the rule from the agenda
  - Match: Update the agenda
    - ★ Add rules whose antecedents are satisfied to the agenda
    - ★ Remove rules with non-satisfied agendas
- The cycle ends when no more rules are on the agenda, or when an explicit stop command is encountered

Foundation of Classic Expert Systems



General Methods of Inferencing

- Forward chaining (data-driven)– reasoning from facts to the conclusions resulting from those facts – best for predicting, monitoring, and control.
  - Examples: CLIPS, OPS5
- Backward chaining (query/Goal driven)– reasoning in reverse from a hypothesis, a potential conclusion to be proved to the facts that support the hypothesis – best for diagnosis problems.
  - Examples: MYCIN

Production Systems

- Rule-based expert systems – most popular type today.
- Knowledge is represented as multiple rules that specify what should/not be concluded from different situations.
- Forward chaining – start w/facts and use rules do draw conclusions/take actions.
- Backward chaining – start w/hypothesis and look for rules that allow hypothesis to be proven true.

Post Production System

- Basic idea – any mathematical / logical system is simply a set of rules specifying how to change one string of symbols into another string of symbols.
  - these rules are also known as rewrite rules
  - simple syntactic string manipulation
  - no understanding or interpretation is required
  - also used to define grammars of languages
    - ★ e.g BNF grammars of programming languages.
- Basic limitation – lack of control mechanism to guide the application of the rules.

Markov Algorithm

- An ordered group of productions applied in order or priority to an input string.
- If the highest priority rule is not applicable, we apply the next, and so on.
- Inefficient algorithm for systems with many rules.
- Termination on
  - Last production not applicable to a string, or
  - Production ending with period applied Can be applied to substrings, beginning at left

Markov Algorithm

(1) αxy → yαx
(2) α → ^
(3) ^ → α

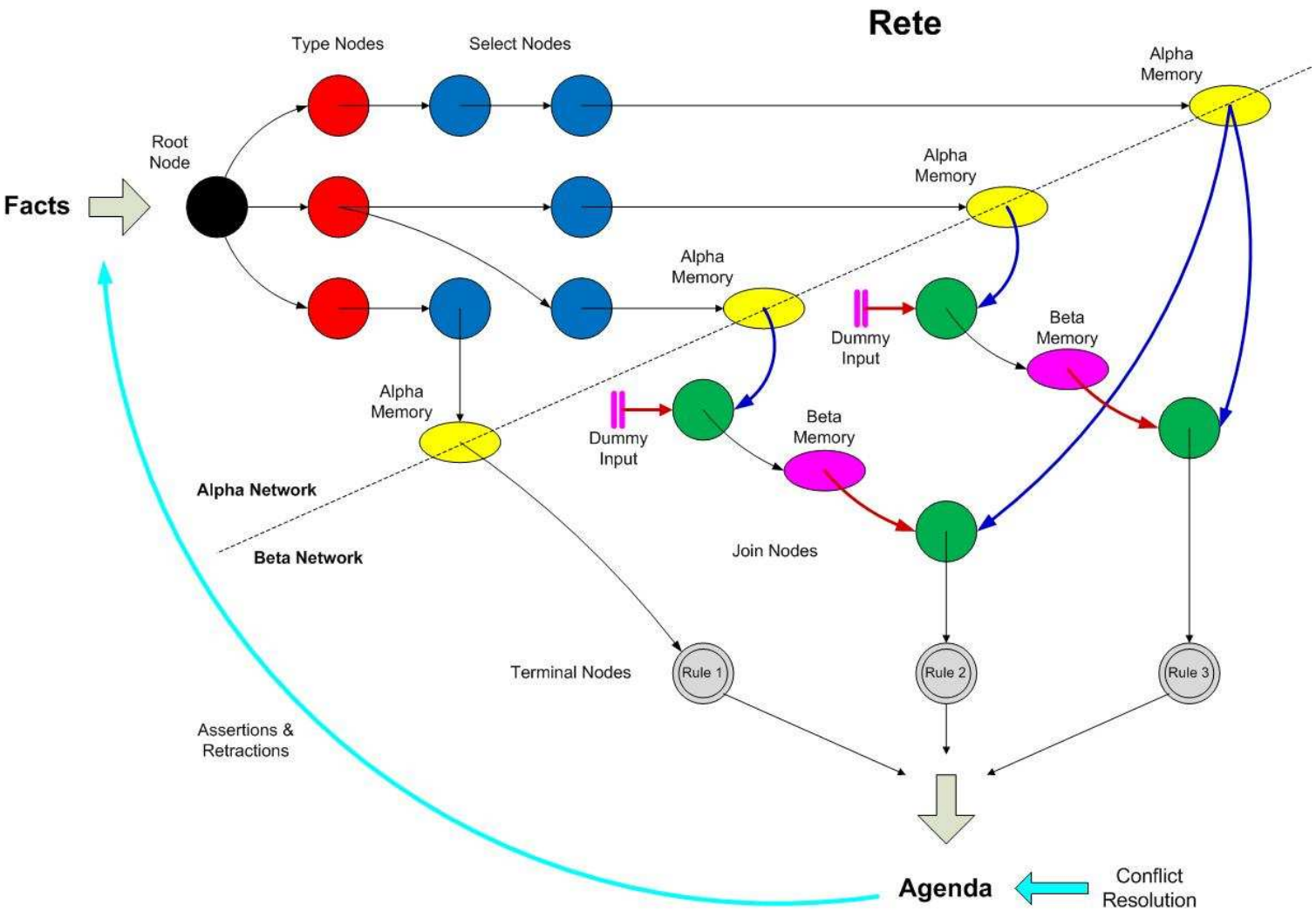
Rule	Success or Failure	String
1	F	ABC
2	F	ABC
3	S	α ABC
1	S	B α AC
1	S	BC α A
1	F	BC α A
2	S	BCA

Table 1.11 Execution Trace of a Markov Algorithm

Rete Algorithm

- Markov: too inefficient to be used with many rules
- Functions like a net – holding a lot of information.
- Much faster response times and rule firings can occur compared to a large group of IF-THEN rules which would have to be checked one-by-one in conventional program.
- Takes advantage of temporal redundancy and structural similarity.
- Looks only for changes in matches (ignores static data)
- Drawback is high memory space requirements.

Rete Algorithm



[from wikipedia]

Summary of Classical Expert Systems

- During the 20th Century various definitions of AI were proposed.
- In the 1960s, a special type of AI called expert systems dealt with complex problems in a narrow domain, e.g., medical disease diagnosis.
- Today, expert systems are used in a variety of fields.
- Expert systems solve problems for which there are no known algorithms.

Reasoning under uncertainty

- Uncertainty: The quality or state of being not clearly known.
  This encompasses most of what we understand about the world — and most of what we would like our AI systems to understand.

  Distinguishes deductive knowledge (e.g., mathematics) from inductive belief (e.g., science).

- Sources of uncertainty
  - Ignorance
    (which side of this coin is up?)
  - Complexity
    (meteorology)
  - Physical randomness
    (which side of this coin will land up?)
  - Vagueness
    (which tribe am I closest to genetically? Picts? Angles? Saxons? Celts?)

Motivation

- Huge variety of cases where
  - Uncertainty dominates considerations
  - Getting it right is crucial
- Examples and consequences:
  - Medicine: death, injury, disease
  - Law: false imprisonment, wrongful execution
  - Space shuttle: explosion
  - Hiring: wasted time and money

Approaches to uncertainty

- MYCIN's certainty factors
- Fuzzy logic
- Default logic
- Probability
- Dempster-Shafer theory

Probability Calculus

- Classic approach to reasoning under uncertainty. (origin: Blaise Pascal and Fermat).
- Kolmogorov's Axioms:
  1. $P(U) = 1$
  2. $\forall X \subseteq U \; P(X) \geq 0$
  3. $\forall X, Y \subseteq U$
     if $X \cap Y = \emptyset$
     then $P(X \vee Y) = P(X) + P(Y)$
- Conditional Probability $P(X|Y) = \frac{P(X \wedge Y)}{P(Y)}$
- Independence $X \perp\!\!\!\perp Y$ iff $P(X|Y) = P(X)$

Bayes' Theorem; Conditionalization as Posterior

- Due to Reverend Thomas Bayes (1764)

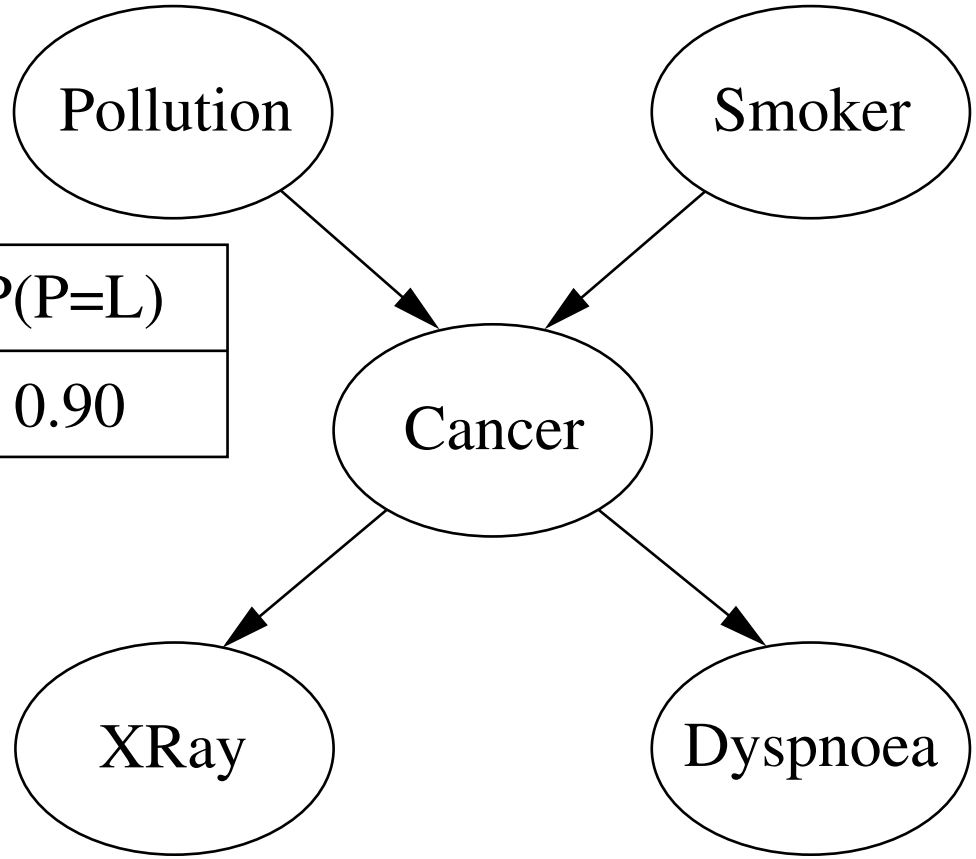
$$P(h|e) = \frac{P(e|h)P(h)}{P(e)}$$

- Conditionalization as posterior: $P'(h) = P(h|e)$
  Or, read Bayes' theorem as:

$$Posterior = \frac{Likelihood \times Prior}{Prob\ of\ \ evidence}$$

- Assumptions:
  - Joint priors over $\{h_i\}$ and e exists.
  - Total evidence: e, and only e, is learned.

Lung cancer example: Bayesian Net and its CPTs



P(S=T)
0.30

P(P=L)
0.90

P	S	P(C=T|P,S)
H	T	0.05
H	F	0.02
L	T	0.03
L	F	0.001

C	P(X=pos|C)
T	0.90
F	0.20

C	P(D=T|C)
T	0.65
F	0.30

Bayesian Decision Theory

- Frank Ramsey (1926)
  Decision making under uncertainty: what action to take (plan to adopt) when future state of the world is not known.

Bayesian answer: Find utility of each possible outcome (action-state pair) and take the action that maximizes expected utility.

Example

action	Rain ($p = 0.4$)	Shine ($1 - p = 0.6$)
Take umbrella	30	10
Leave umbrella	-100	50

Expected utilities:

$$\begin{aligned} E(\textit{Take umbrella}) &= (30)(0.4) + (10)(0.6) = 18 \\ E(\textit{Leave umbrella}) &= (-100)(0.4) + (50)(0.6) = -10 \end{aligned}$$

Summary [1]

- Classical expert systems
  - Rule-based
  - Explanation
  - The challenge of uncertainty
- Probabilistic approaches: Bayesian Networks

[1] The materials of Lecture 1 are taken from [Giarratano and Riley, 2005, Chapter 1] and [Korb and Nicholson, 2003, Chapter 1] with the instructor's own interpretations. The instructor takes full responsibility of any mistakes in the slides.

References I

Joseph C. Giarratano and Gary Riley.
Expert systems : principles and programming.
Thomson Course Technology, c2005., October 2005.

K. Korb and A. E. Nicholson.
Bayesian Artificial Intelligence.
Chapman & Hall /CRC, 2003.