UIN SUSKA RIAU
Universitas Islam Negeri Sultan Syarif Kasim Riau
Berubah Untuk Maju ..

EXPERT SYSTEM

ASSALAMU'ALAIKUM

# CHAPTER 1

# INTRODUCTION OF EXPERT SYSTEM


RAHMAD KURNIAWAN, S.T., M.I.T.

UIN SUSKA RIAU


A part of the course slides have been obtained and adapted with permission from Dr. Franz Kurfess, CalPoly, San Luis Obispo

# General Info

## Course Material

- Course web page:
  - http://tif.uin-suska.ac.id/es
  - Textbooks (see below)
- Lecture Notes*
  - PowerPoint Slides available on the course web page
  - Will be updated during the term if necessary

## Assessment

- Quiz
- Homework Assignments
- Individual Research Report
- Group Project
- Mid Test (Computer Assisted Test)
- Final (Proposal Defense)

# Mr. Rahmad Kurniawan, S.T., M.I.T.


- **FST Building, L-III, TIF Room**
- **Phone:** +62 853 5550 5833
- **E-mail:**
  rahmadkurniawan[at]uin-suska.ac.id
- **Home page:**
  - www.rahmadkurniawan.co.nr
  - http://fst.uin-suska.ac.id/rk
  - http://tif.uin-suska.ac.id/rk

- **Office hours: M, T: 12:30-03:30pm**
  **W : 08:00-12:30pm**
  **Th, F: 08:00-04:00pm**

# Course Overview

- **Introduction**
- **CLIPS Overview**
  - Concepts, Notation, Usage
- **Knowledge Representation**
  - Semantic Nets, Frames, Logic
- **Reasoning and Inference**
  - Predicate Logic, Inference Methods, Resolution
- **Reasoning with Uncertainty**
  - Probability, Bayesian Decision Making
- **Pattern Matching**
  - Variables, Functions, Expressions, Constraints
- **Expert System Design**
  - Expert SystemLife Cycle
- **Expert System Implementation**
  - Salience, Rete Algorithm
- **Thesis Preparation**
  - Topic: Expert System

# Textbooks

## Main Textbook

- Joseph Giarratano and Gary Riley. *Expert Systems - Principles and Programming.* 4th ed., PWS Publishing, Boston, MA, 2004

## Secondary Textbook

- Peter Jackson. *Introduction to Expert Systems.* 3rd ed., Addison-Wesley, 1999.

# Overview Introduction

- **Motivation**
- **Objectives**
- **What is an Expert System (XPS)?**
  - knowledge, reasoning
- **General Concepts and Characteristics of Expert System**
  - knowledge representation, inference, knowledge acquisition, explanation
- **Expert System Technology**
- **Expert SystemTools**
  - shells, languages
- **Expert System Elements**
  - facts, rules, inference mechanism
- **Important Concepts and Terms**
- **Chapter Summary**

# Motivation

- **Utilization of computers to deal with knowledge**
  - Quantity of knowledge increases rapidly
  - Knowledge might get lost if not captured
  - Relieves humans from tedious tasks
- **Computers have special requirements for dealing with knowledge**
  - Acquisition, representation, reasoning
- **Some knowledge-related tasks can be solved better by computers than by humans**
  - Cheaper, faster, easily accessible, reliable

# Objectives

- **To know and comprehend the main principles, components, and application areas for expert systems**
- **To understand the structure of expert systems**
  - Knowledge base, inference engine
- **To be familiar with frequently used methods for knowledge representation and reasoning in computers**
- **To apply expert system techniques for specific tasks**
  - Application of methods in certain scenarios

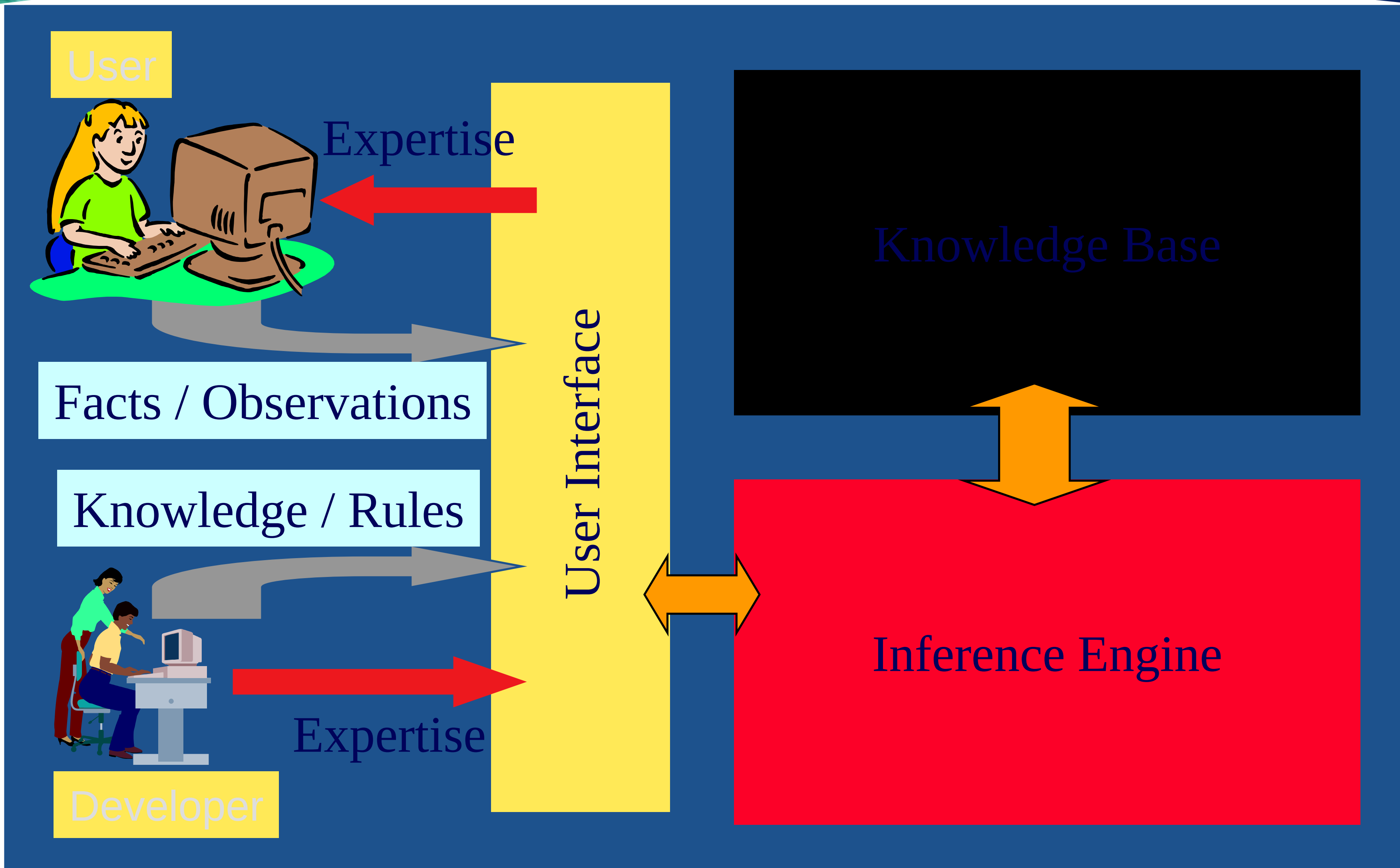
# Expert Systems (ES)

- **Rely on internally represented knowledge to perform tasks**
- **Utilizes reasoning methods to derive appropriate new knowledge**
- **Are usually restricted to a specific *problem domain***
- **Some systems try to capture more general knowledge**
  - General problem solver (newell, shaw, simon)
  - Cyc (lenat)

# What is an “Expert System”?

- **A computer system that emulates the decision-making ability of a human expert in a restricted domain [Giarratano & Riley 1998]**
- **Edward Feigenbaum**
  - “An intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solutions.” [Giarratano & Riley 1998]
- **Sometimes, we also refer to *knowledge-based system***

# Main Components of an ES



# Main Expert SystemComponents

## ❖ knowledge base

- contains essential information about the problem domain
- often represented as facts and rules

## ❖ inference engine

- mechanism to derive new knowledge from the knowledge base and the information provided by the user
- often based on the use of rules

## ❖ user interface

- interaction with end users
- development and maintenance of the knowledge base

# Concepts and Characteristics of ES

- **Knowledge acquisition**
  - Transfer of knowledge from humans to computers
  - Sometimes knowledge can be acquired directly from the environment
    - Machine learning, neural networks
- **Knowledge representation**
  - Suitable for storing and processing knowledge in computers
- **Inference**
  - Mechanism that allows the generation of new conclusions from existing knowledge in a computer
- **Explanation**
  - Illustrates to the user how and why a particular solution was generated

# Development of Expert SystemTechnology

- **Strongly influenced by cognitive science and mathematics / logic**
  - The way humans solve problems
  - Formal foundations, especially logic and inference
- **Production rules as representation mechanism**
  - `IF … THEN` type rules
  - Reasonably close to human reasoning
  - Can be manipulated by computers
  - Appropriate granularity
    - Knowledge "chunks" are manageable for humans and computers

[Dieng et al. 1999]

# Rules and Humans

- **Rules can be used to formulate a theory of human information processing (newell & simon)**
  - Rules are stored in long-term memory
  - Temporary knowledge is kept in short-term memory
  - (External) sensory input triggers the activation of rules
  - Activated rules may trigger further activation (internal input; “thinking”)
  - A cognitive processor combines evidence from currently active rules
- **This model is the basis for the design of many rule-based systems (*production systems*)**

# Early Expert SystemSuccess Stories

- **DENDRAL (Feigenbaum, Lederberg, and Buchanan, 1965)**
  - deduce the likely molecular structure of organic chemical compounds from known chemical analyses and mass spectrometry data
- **MYCIN (Buchanan and Shortliffe, 1972-1980)**
  - diagnosis of infectious blood diseases and recommendation for use of antibiotics
  - "empty" MYCIN = EMYCIN = Expert Systemshell
- **PROSPECTOR**
  - analysis of geological data for minerals
  - discovered a mineral deposit worth $100 million
- **XCON/R1 (McDermott, 1978)**
  - configuration of DEC VAX computer systems
  - 2500 rules; processed 80,000 orders by 1986; saved DEC $25M a year

# The Key to Expert SystemSuccess

- **Convincing ideas**
  - Rules, cognitive models
- **Practical applications**
  - Medicine, computer technology, …
- **Separation of knowledge and inference**
  - Expert system *shell*
    - Allows the re-use of the “machinery” for different domains
- **Concentration on domain knowledge**
  - General reasoning is too complicated

# When (Not) to Use an ES

- **Expert systems are not suitable for all types of domains and tasks**
- **They are not useful or preferable, when …**
  - efficient conventional algorithms are known
  - the main challenge is computation, not knowledge
  - knowledge cannot be captured efficiently or used effectively
  - users are reluctant to apply an expert system, e.g. due to  criticality of task, high risk or high security demands

# Expert SystemDevelopment Tools

## ❖ Expert Systemshells

- an Expert Systemdevelopment tool / environment where the user provides the knowledge base
- CLIPS, JESS, EMYCIN, Babylon, ...

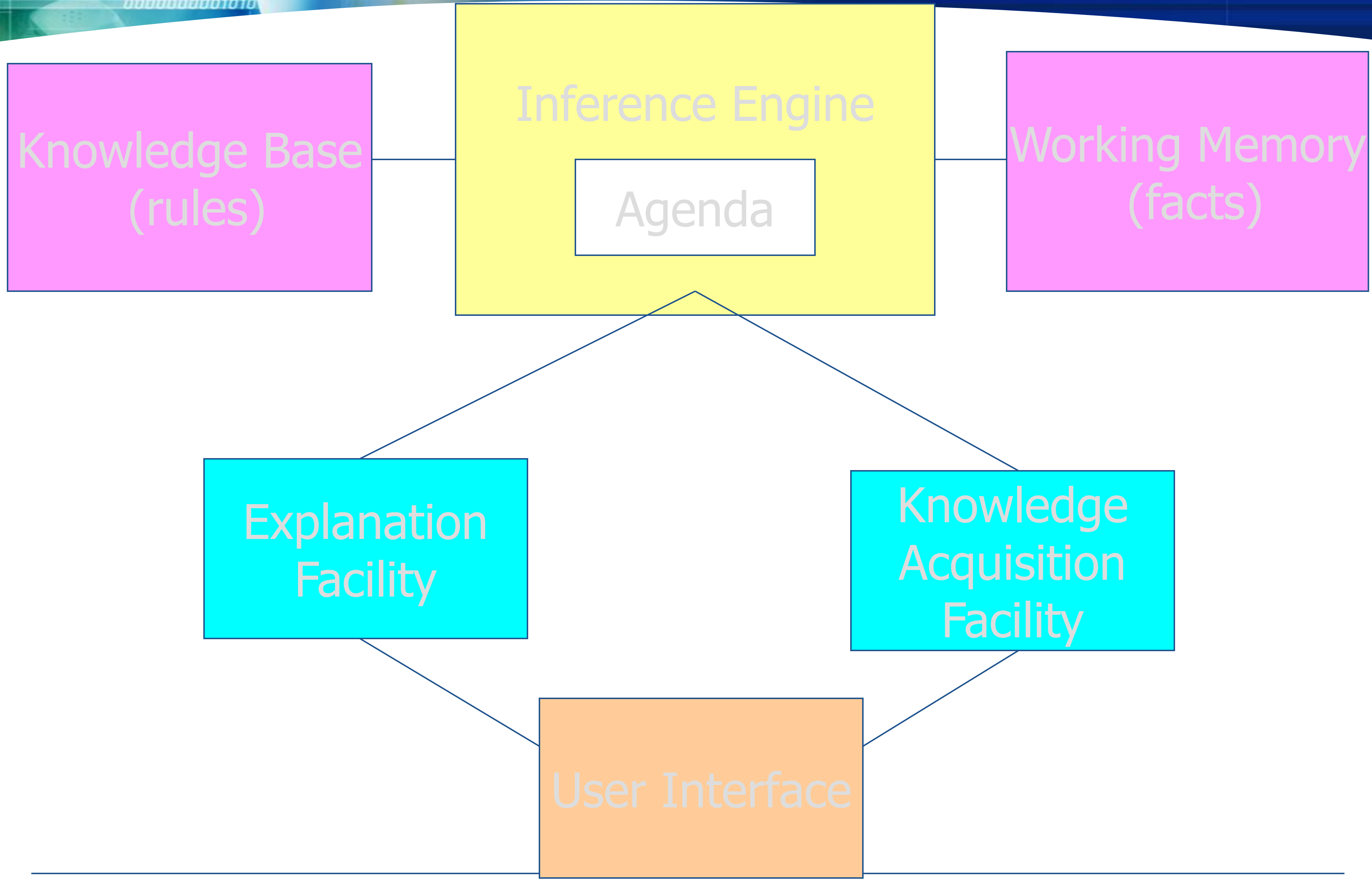
## ❖ Knowledge representation languages; ontologies

- higher-level languages specifically designed for knowledge representation and reasoning
- KRL, KQML, KIF, DAML, OWL, Cyc

# Expert SystemElements

- **Knowledge base**
- **Inference engine**
- **Working memory**
- **Agenda**
- **Explanation facility**
- **Knowledge acquisition facility**
- **User interface**

# Expert System Structure



# Architecture of Rule-Based Expert System

## Knowledge-Base / Rule-Base

- store expert knowledge as **condition-action-rules** (aka: **if-then-** or **premise-consequence-rules**)

## Working Memory

- stores **initial facts** and **generated facts** derived by inference engine; maybe with additional parameters like the "degree of trust" into the truth of a fact ≅ **certainty factor**

## Inference Engine

- matches **condition-part** of rules against facts stored in Working Memory (**pattern matching**);
- rules with satisfied condition are **active rules** and are placed on the **agenda**;
- among the active rules on the agenda, one is **selected** (see **conflict resolution, priorities of rules**) as next rule for
- **execution** (“**firing**”) – consequence of rule is added as new fact(s) to Working Memory

# Cont…

**Inference Engine + additional components**

**might be necessary for other functions, like**

- **calculation of certainty values,**
- **determining priorities of rules,**
- **conflict resolution mechanisms,**
- **a truth maintenance system (TMS) if reasoning with defaults and beliefs is requested**

## Cont...

**Explanation Facility**

provides justification of solution to user (reasoning chain)

**Knowledge Acquisition Facility**

helps to integrate new knowledge; also automated knowledge acquisition

**User Interface**

allows user to interact with the Expert System- insert facts, query the system, solution presentation

# Rule-Based Expert System

- **Knowledge is encoded as `IF … THEN` rules**
  - Condition-action pairs
- **The inference engine determines which rule antecedents (condition-part) are satisfied**
  - The left-hand condition-part must “match” facts in the working memory
- **Matching rules are “activated”, i.E. Placed on the agenda**
- **Rules on the agenda can be executed (“fired”)**
  - An activated rule may generate new facts and/or cause actions through its right-hand side (action-part)
  - The activation of a rule may thus cause the activation of other rules through added facts based on the right-hand side of the fired rule

# Example Rules

**IF** … **THEN** Rules

```
Rule: Red_Light
   IF          the light is red
   THEN        stop
Rule: Green_Light
   IF          the light is green
   THEN        go
```

antecedent
(left-hand-side)

consequent
(right-hand-side)

Production Rules

antecedent (left-hand-side)

```
the light is red ==> stop

the light is green ==> go
```

consequent
(right-hand-side)

# MYCIN Sample Rule

## Human-Readable Format

```
IF    the stain of the organism is gram negative
AND   the morphology of the organism is rod
AND   the aerobiocity of the organism is gram anaerobic
THEN  there is strong evidence (0.8)
      that the class of the organism is enterobacteriaceae
```

## MYCIN Format

```
IF   (AND (SAME CNTEXT GRAM GRAMNEG)
          (SAME CNTEXT MORPH ROD)
          (SAME CNTEXT AIR AEROBIC)
THEN (CONCLUDE CNTEXT CLASS ENTEROBACTERIACEAE
      TALLY .8)
```

[Durkin 94, p. 133]

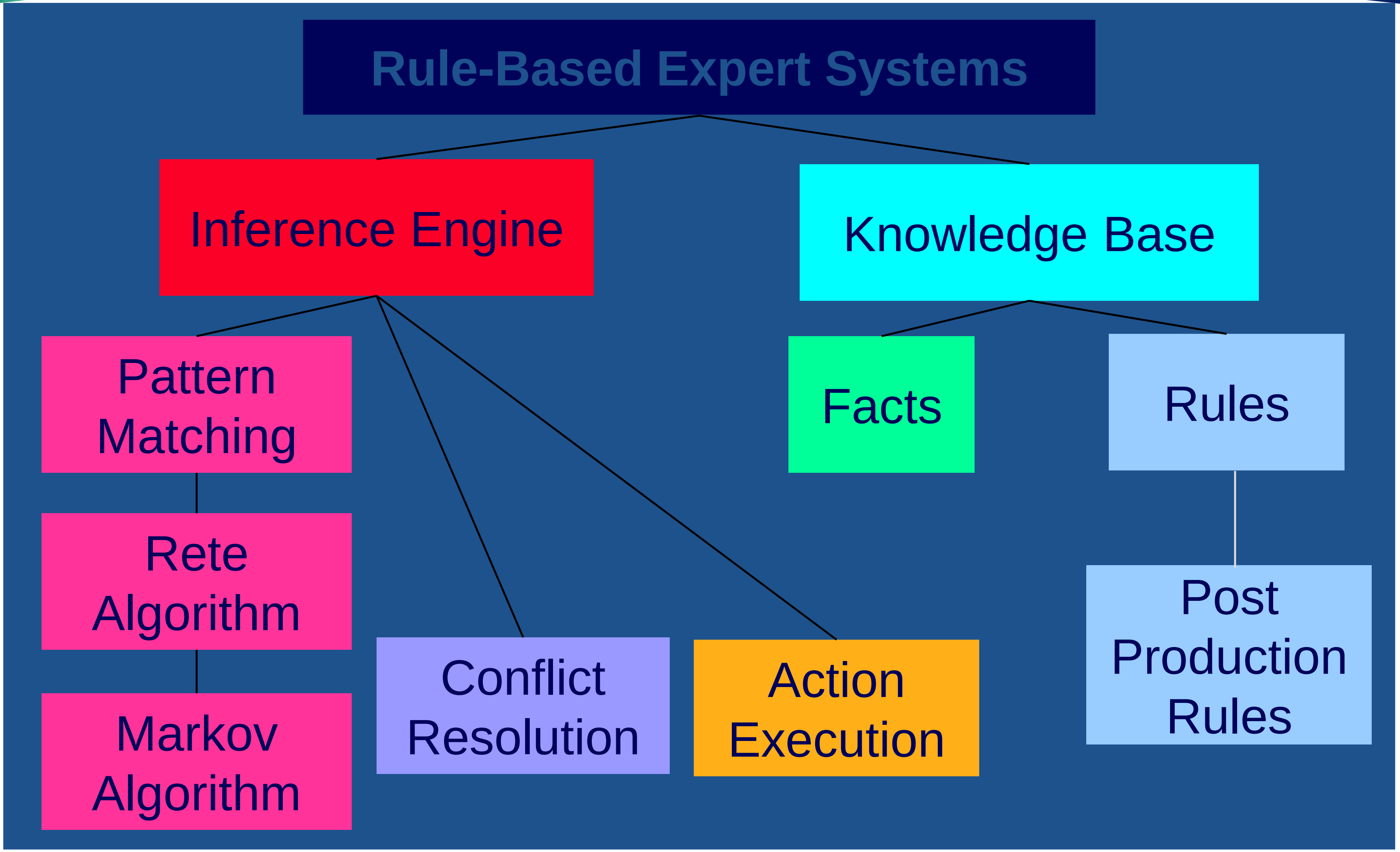
# Inference Engine Cycle

- **Describes the execution of rules by the inference engine**
- **"Recognize-act cycle"**
  - Pattern matching
    - Update the agenda (= conflict set)
      - Add rules, whose antecedents are satisfied
      - Remove rules with non-satisfied antecedents
  - Conflict resolution
    - Select the rule with the highest priority from the agenda
  - Execution
    - Perform the actions in the consequent part of the selected rule
    - Remove the rule from the agenda
- **The cycle ends when no more rules are on the agenda, or when an explicit stop command is encountered**

# Forward and Backward Chaining

- **Different methods of reasoning and rule activation**
  - forward chaining (data-driven)
    - reasoning from facts to the conclusion
    - as soon as facts are available, they are used to match antecedents of rules
    - a rule can be activated if all parts of the antecedent are satisfied
    - often used for real-time expert systems in monitoring and control
    - examples: CLIPS, OPS5
  - backward chaining (query-driven)
    - starting from a hypothesis (query), supporting rules and facts are sought until all parts of the antecedent of the hypothesis are satisfied
    - often used in diagnostic and consultation systems
    - examples: EMYCIN

# Foundations of Expert Systems





Rahmad Kurniawan, S.T., M.I.T.

# Post Production Systems

- **Production rules were used by the logician emil L. Post in the early 40s in symbolic logic**
- **Post's theoretical result**
  - Any system in mathematics or logic can be written as a production system
- **Basic principle of production rules**
  - A set of rules governs the conversion of a set of strings into another set of strings
    - These rules are also known as ***rewrite rules***
    - Simple syntactic string manipulation
    - No understanding or interpretation is required

# Markov Algorithms

- **In the 1950s, A. A. Markov introduced *priorities* as a control structure for production systems**
  - rules with higher priorities are applied first
  - allows more efficient execution of production systems
  - but still not efficient enough for expert systems with large sets of rules

# Rete Algorithm

- **Rete is a Latin word and means network, or net**
- **The Rete Algorithm was developed by Charles L. Forgy in the late 70s for CMU's OPS (Official Production System) shell**
  - stores information about the antecedents in a network
  - in every cycle, it only checks for changes in the networks
  - this greatly improves efficiency

# Expert System Advantages

- **Economical**
  - Lower cost per user
- **Availability**
  - Accessible anytime, almost anywhere
- **Response time**
  - Often faster than human experts
- **Reliability**
  - Can be greater than that of human experts
  - No distraction, fatigue, emotional involvement, …
- **Explanation**
  - Reasoning steps that lead to a particular conclusion
- **Intellectual property**
  - Can’t walk out of the door

# Expert System Problems

- **limited knowledge**
  - "shallow" knowledge
    - no "deep" understanding of the concepts and their relationships
  - no "common-sense" knowledge
  - no knowledge from possibly relevant related domains
  - "closed world"
    - the Expert System knows only what it has been explicitly "told"
    - it doesn't know what it doesn't know
- **mechanical reasoning**
  - may not have or select the most appropriate method for a particular problem
  - some "easy" problems are computationally very expensive
- **lack of trust**
  - users may not want to leave critical decisions to machines

# Summary Introduction

- **Expert systems or knowledge based systems are used to represent and process knowledge in a format that is suitable for computers but still understandable by humans**
  - If-then rules are a popular format
- **The main components of an expert system are**
  - Knowledge base
  - Inference engine
- **Expert system can be cheaper, faster, more accessible, and more reliable than humans**
- **Expert system have limited knowledge (especially "common-sense"), can be difficult and expensive to develop, and users may not trust them for critical decisions**

# Important Concepts and Terms

- Agenda
- Backward Chaining
- Common-sense Knowledge
- Conflict Resolution
- Expert System (ES)
- Expert System Shell
- Explanation
- Forward Chaining
- Inference
- Inference Mechanism
- If-then Rules
- Knowledge
- Knowledge Acquisition
- Knowledge Base
- Knowledge-based System
- Knowledge Representation
- Markov Algorithm
- Matching
- Post Production System
- Problem Domain
- Production Rules
- Reasoning
- RETE Algorithm
- Rule
- Working Memory

## ❖ Thank you