BABEŞ-BOLYAI UNIVERSITY
Faculty of Computer Science and Mathematics

ARTIFICIAL INTELLIGENCE

Intelligent systems

Rule-based systems

Topics

- A. Short introduction in Artificial Intelligence (AI)
- A. Solving search problems
  - A. Definition of search problems
  - B. Search strategies
    - A. Uninformed search strategies
    - B. Informed search strategies
    - C. Local search strategies (Hill Climbing, Simulated Annealing, Tabu Search, Evolutionary algorithms, PSO, ACO)
    - D. Adversarial search strategies
- C. **Intelligent systems**
  - A. **Rule-based systems in certain environments**
  - B. Rule-based systems in uncertain environments (Bayes, Fuzzy)
  - C. Learning systems
    - A. Decision Trees
    - B. Artificial Neural Networks
    - C. Support Vector Machines
    - D. Evolutionary algorithms
  - D. Hybrid systems

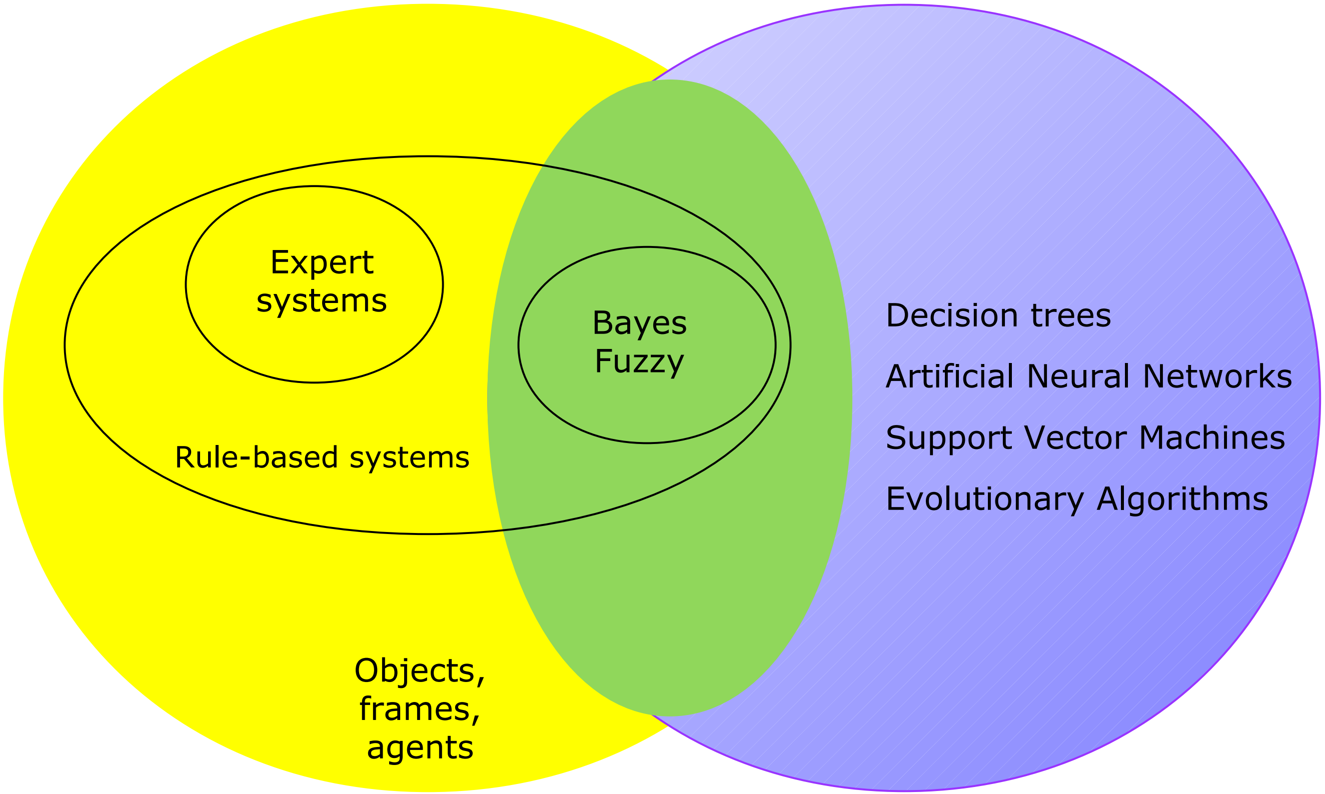
Useful information

- Chapter III of *S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995*
- Chapter 4 and 5 of *H.F. Pop, G. Şerban, Inteligenţă artificială, Cluj Napoca, 2004*
- Chapter 2 of *Adrian A. Hopgood, Intelligent Systems for Engineers and Scientists, CRC Press, 2001*
- Chapter 6 and 7 of *C. Groşan, A. Abraham, Intelligent Systems: A Modern Approach, Springer, 2011*

Content

- Intelligent systems
  - Knowledge-based systems
    - Logic-based systems
    - Rule-based systems in certainty

Intelligent systems



Sisteme bazate pe cunoştinţe (SBC) Inteligenţă computaţională

Intelligent systems – knowledge-based systems (KBS)

- Computational systems - 2 main modules
  - *knowledge base (KB)*
    - Specific information
  - *inference engine (IE)*
    - Rules that can determine new information
    - Domain independent algorithms

Intelligent systems – KBS

Knowledge base

- Content
- Typology
- Representation
- Storing

Intelligent systems – KBS

Knowledge base (KB)

- Content
  - Information (in a given representation – eg. Sentences) about domain
  - Required information for problem solving
  - Set of sentences (represented in a formal way) that describe an environment
    - Interpretability of representation → knowledge representation language
    - Mechanism for obtaining new sentences from the old ones → inference

- Typology
  - Perfect knowledge
  - Imperfect knowledge
    - Not-exactly
    - Incomplete
    - Non-measurable

- Knowledge representation
  - Formal logic (formal languages)
  - Rules
  - Semantic nets

Intelligent systems – KBS

Knowledge base (KB)

- Knowledge representation
  - Formal logic (formal languages)
    - Definition
      - Science of formal principles for rationing
    - Components
      - Syntax – atomic symbols used by language and the constructing rules of the language
      - Semantic – associates a meaning to each symbol and a truth value (true or false) to each rule
      - Syntactic inference – rules for identifying a subset of logic expressions → theorems (for generating new expressions)
    - Typology
      - True value
        - Dual logic
        - Polyvalent logic
      - Basic elements
        - Classic → primitives = sentences (predicates)
        - Probabilistic → primitives = random variables
      - Working manner
        - Propositional logic → declarative propositions and fix or unique objects (Ionica is student)
        - First-order logic → declarative propositions, predicates and quantified variables, unique objects or variables associated to a unique object
  - Rules
  - Semantic nets

Intelligent systems – KBS

Knowledge base (KB)

- Knowledge representation
  - Formal logic
  - Rules
    - Special heuristics that generate information
    - A possible representation of information
    - Ex. If Ionica works at Facebook, then he earns a lot, but he has little free time.
    - Inter-dependencies among rules → inference network
    - Link the cause and effect → if clause then effect
  - Semantic nets
    - Oriented graphs with nodes that contain concepts and edges that represent semantic relations among concepts
      - *Meronymy* (A is a meronym of B if A is a part of B)
        - Eg. Finger is a meronym of hand, wheel is a meronym of car
      - *Holonymy* (A is holonym of B dacă B is a part of A)
        - Ex. Tree is a holonym of branch
      - *Hyponymy* (A is hyponym of B if A is a kind of B)
        - Ex. Tractor is a hyponym of vehicle
      - *Hypernymy* (A is hypernym of B if A is a generalization of B)
        - Ex. Fruit is a hypernym of orange
      - *Synonymy* (A is a synonym of B if A and B have the same meaning)
        - Ex. run is synonym to jog
      - *Antonymy* (A is an antonym of B if A and B reflect opposed things)
        - Ex. Dry is an antonym to wet

Intelligent systems – KBS

Knowledge base (KB)

- Storage
  - Relations
    - Simple → data bases
    - Hierarchical → semantic nets
  - Formal logic
    - Rules
  - Procedural logic
    - Algorithms

Intelligent systems – KBS

Inference engine (IE)

- Content
  - Determine a conclusion by taking into account some premises and by applying some inference rules
  - IE depends on the complexity and type of knowledge

- Typology
  - Inference direction
    - IE with *forward chaining*
      - Start from available information and determine a conclusion
      - Based on data (*data driven*)
    - IE with *backward chaining*
      - Start from a possible conclusion and identify some explanations for it
      - Based on aim *(goal driven*)

- Techniques for inference
  - Certainty environment
    - Logic-base
    - Rule-based
  - Uncertainty environment
    - Based on theory of probability
    - Based on theory of possibility

Intelligent systems – KBS

Typology

- **Logic based systems (LBS)**
- **Rule based systems (RBS)**
- *Case-based reasoning*
- *Hypertext manipulating systems*
- *Data bases and intelligent UI*
- *Intelligent tutoring systems*

Intelligent systems – KBS

Logic based systems (LBS)

- Content and aim
- Architecture
- Typology
- Tools
- Advantages and limits

Intelligent systems – KBS – LBS

Content and aim

- Content
  - Explore a lot of information for obtaining conclusions about difficult activities by using methods of formal logic
  - A logic system is composed of
    - Language (syntax and semantic)
    - Deduction method (inference)
- Aim of a LBS
  - Problem solving by help of declarative programming
  - Ex. Automatic theorems proving
- Why LBS are studied?
  - Formal logic is precise and well-defined

Intelligent systems – KBS – LBS

Architecture

- Knowledge base (KB)
  - Syntax
    - Atomic symbols used by a language and construction rules of expressions
  - Semantic
    - Meaning of symbols and the truth value of rules

- Inference engine (IE)
  - Inference, as method, - rules for determining a subset of logic expressions → theorems

Intelligent systems – KBS – LBS

Typology

- **Systems based on propositional logic**
  - Declarative propositions only
  - Described objects are unique and fixed (*Ionica is student*)

- Systems based on first-order logic
  - Declarative propositions, predicates (*Student(a)*) and quantified variables (*for any a, Student(a) → WifiAccess(a)*)
  - Described objects can be unique or dynamic (*all the students are present*)
  - Predicates have simple arguments (*Student(a)*)

- Systems based on higher-order logic (≥ 2)
  - Declarative propositions, predicates and quantified variables
  - Variables can represents more relations among objects
  - Predicates can have simple arguments or predicates arguments (*StudentSenator*(*Student*(*a*))) or function arguments (*Bursier(a has average over 9.50)*)

- Temporal systems
  - Represent the truth value of fact along time (*Ionică is sometimes nervous*)

- Modal systems
  - Represents doubtful facts also (*Ionică could pass the exam*)

Intelligent systems – KBS – LBS

Systems based on propositional logic

- Knowledge base
- Composed by
    - symbols (A, B, P, Q, ...)
    - Propositions (formula)
        - Defined as
            - A symbol
            - Is P is a proposition, then ¬ P is a proposition too
            - If P and Q are propositions, then $P \wedge Q$, $P \vee Q$, $P \Rightarrow Q$, $P \Leftrightarrow Q$ are propositions too
            - A finite number of applying the rules (1)-(3)
        - Meaning of a proposition → truth value
        - Model → interpretation of a set of propositions such as all the propositions are true

- Inference engine
- Performs inference
    - Establish the truth value of a proposition taking into account the KB
- Typology
    - Model checking
        - All the possible combinations of truth value for all the involved symbols and propositions
    - Model deduction

Intelligent systems – KBS – LBS

Systems based on propositional logic – inference engine

- Problem
- For a KB = {P1, P2, ..., Pm} composed by symbols {X1, X2, ..., Xn} and an objective proposition O
- It is possible to deduce O from KB?

- Model checking
- Steps
  - Construct the table for all possible combinations of truth values for all the symbols
  - Determine if all the models of KB are models for O too
    - KB models – that models where all propositions are true
- Example
  - P = *Afară este foarte cald*
  - Q = *Afară este umezeală*
  - R = *Afară plouă*

  - BC = {P∧Q⇒R, Q⇒P, Q}
  - R = *Va ploua?*

P	Q	R	P∧Q⇒R	Q⇒P	Q	BC	R	BC⇒R
T	T	T	T	T	T	T	T	T
T	T	F	F	T	T	F	F	T
T	F	T	T	T	F	F	T	T
T	F	F	T	T	F	F	F	T
F	T	T	T	F	T	F	T	T
F	T	F	T	F	T	F	F	T
F	F	T	T	T	F	F	T	T
F	F	F	T	T	F	F	F	T

- Difficulties
  - Exponential number for all combinations → large computing time
  - Solution: deduction by using inference rules

Intelligent systems – KBS – LBS

Intelligent systems – KBS – LBS

- Problem
- For a KB = {P1, P2, ..., Pm} composed by symbols {X1, X2, ..., Xn} and an objective proposition O
- It is possible to deduce O from KB?

- Model deduction by using inference rules
- Steps
  - Construct a demonstration (proof) of truth value for the objective proposition based on:
    - Propositions
      - Original (that from KB)
      - Derivative
    - Inference rules

Regulă de inferenţă	Premisă	Propoziţia derivată
Modus ponens	$A, A \Rightarrow B$	B
Şi introductiv	A, B	$A \wedge B$
Şi eliminativ	$A \wedge B$	A
Negaţie dublă	$\neg\neg A$	A
Rezoluţie unitară	$A \vee B, \neg B$	A
Rezoluţie	$A \vee B, \neg B \vee C$	$A \vee C$

- Example
  - Problem
    - P = Afară este foarte cald
    - Q = Afară este umezeală
    - R = Afară plouă
    - BC = $\{P \wedge Q \Rightarrow R, Q \Rightarrow P, Q\}$
    - R = Va ploua?

Solution

1.	Q	Premise
2.	$Q \Rightarrow P$	Premise
3.	P	Modus Ponens (1,2)
4.	$(P \wedge Q) \Rightarrow R$	Premise
5.	$P \wedge Q$	introductiv AND(1,3)
6.	R	Modus Ponens (4,5)

Intelligent systems – KBS

Typology

- Logic based systems (LBS)
- **Rule based systems (RBS)**
- *Case-based reasoning*
- *Hypertext manipulating systems*
- *Data bases and intelligent UI*
- *Intelligent tutoring systems*

Intelligent systems – KBS

- Rule-based systems (RBS)
  - Content and objectives
  - Design
  - Architecture
  - Tools and example
  - Advantages and limits

Intelligent systems – KBS – RBS

Content and objectives

- Content
  - Explore a lot of knowledge for new conclusions about activities that are difficult to be analyzed by using methods as humans do
  - Can solve problems for which there is no deterministic algorithm
  - try to simulate a human expert (in a given domain).
  - RBSs do not replace human experience, but they facilitate the work of non-experts → Expert Systems

- RBS's aim
  - Solving problems that require human experts
    - To transfer expertise from an expert to a computer system and
    - Than on to other humans (non-experts)

  - Examples of problems solved by RBSs → consulting problems
    - Detector problems in operating systems - Microsoft Windows troubleshooting
    - Financial consultant
    - Medical diagnosis - program takes place of a doctor; given a set of symptoms the system suggests a diagnosis and treatment
    - Car fault diagnosis - given car's symptoms, suggest what is wrong with it

- Why RBS are investigated?
  - In order to understand the human reason
  - Human experts need holidays, can go to other companies, can be sick, can ask more money
  - Have a big commercial success

Intelligent systems – KBS – RBS

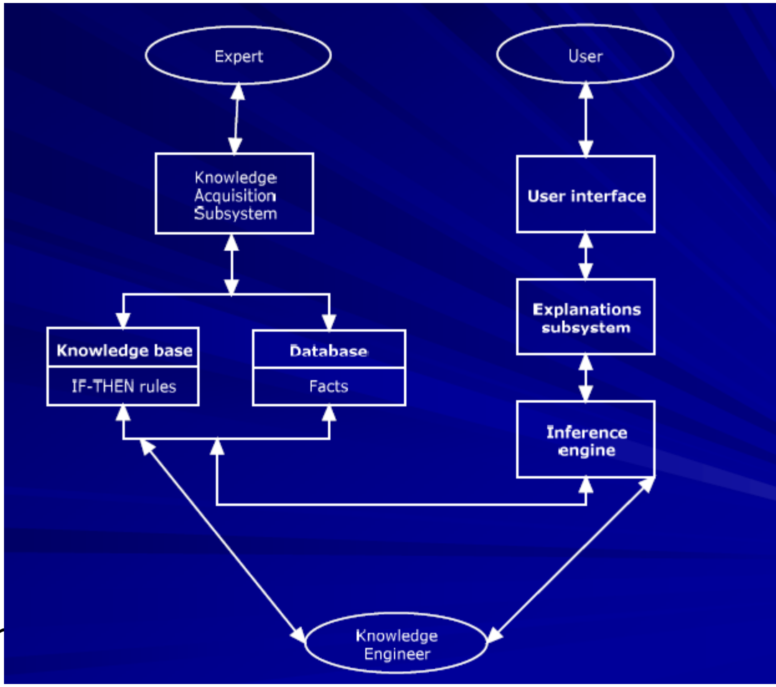
Design

- Knowledge acquisition
- Knowledge representation
- Knowledge inference
- Knowledge transfer to the user

Intelligent systems – KBS – RBS

Architecture

- Knowledge base (KB)
  - Specific information
- Inference engine (IE)
  - Rules utilized for generating new information
- User interface
  - Allows dialog with users and accessing facts and knowledge from KB (to be added or updated)
  - A facility for the user to interact with the system
- Knowledge Acquisition Facility
  - An automatic way to acquire knowledge
- Explanation Facility
  - Explains reasoning of the system to the user



Intelligent systems – KBS – RBS

Architecture → knowledge base

- Content
  - knowledge necessary for understanding, formulating, and solving problems
    - Facts – correct sentences
    - Rules - special heuristics, or rules that directly generates knowledge
    - Knowledge is the primary raw material of ES
    - Incorporates knowledge representation
- Aim
  - Storing all knowledge (facts, rules, solving methods, heuristics) about a specific domain taken from human experts or other sources

Intelligent systems – KBS – RBS

Architecture → knowledge base

- Facts
  - Definition
    - Correct affirmations (sentences, propositions)
    - Memorised as some data structures
  - Example
    - *Ionică works at Facebook*
  - Typology
    - Persistence (changing rate)
      - Static facts - ~ permanently *(Ionica works at Facebook)*
      - Dynamic facts – specific for an instance/run *(Ionica takes a lunch brake)*
    - Generation
      - Given facts *(Ionica assists to a meeting)*
      - Derived facts – resulted by applying some rules *(If Ionica is PM, then he must lead the meeting)*

Intelligent systems – KBS – RBS

Architecture → knowledge base

- Rules
  - Definition
    - Special heuristics that generate knowledge (information)
    - A possible knowledge representation
    - Inter-dependencies among rules → inference network
    - Link a cause to an effect → clauses *IF cond THEN effect1 ELSE effect2*
      - IF this condition (or premise or antecedent) occurs,
      - THEN some action (or result, or conclusion, or consequence) will (or should) occur
      - Example
        - IF temperature < -20oC AND you are in Romania, THEN is winter.
      - Deduction – cause + rule → effect
      - Abduction – effect + rule → cause
      - Induction – cause + effect → rule
    - Rules can be viewed as a simulation of the cognitive behavior of human experts
    - Rules represent a model of actual human behavior

Intelligent systems – KBS – RBS

Architecture → knowledge base

- Rules
  - Examples
    - A single cause (antecedent) and more consequences (combined by AND)
      - *IF Ionica works at Facebook, THEN he has many money AND little free time*
    - A single cause (antecedent) and more consequences (combined by OR)
      - *IF it is winter THEN it is cold OR it is snow.*
    - More causes/antecedents (combined by AND) and a single consequence
      - IF it is winter AND temperature < 0 AND it is wind THEN we can not walk.
    - More causes/antecedents (combined by OR) and a single consequence
      - IF it is winter OR temperature < 0 OR it is wind THEN we can not walk.
    - More causes/antecedents (combined by ANDOR/) and more consequences
      - IF it is winter AND temperature < 0 OR it is wind THEN the airplanes can not land.

Intelligent systems – KBS – RBS

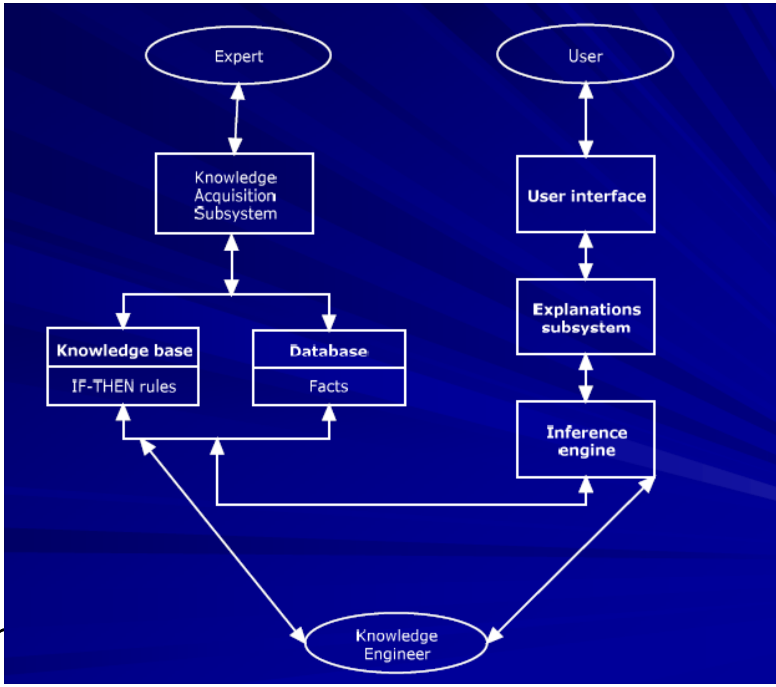
Architecture → knowledge base

- Rules
  - Typology
    - Certainty degree
      - Exact rules - *If you are employed then you have a salary*
      - Uncertain rules – *If it is winter, then temperature < 0*
    - Rules can express
      - Relations – ex. *Dacă studentul are media peste 9.50, atunci el primeşte bursă*
      - Recommendations – ex. *Dacă plouă, atunci să luăm umbrela*
      - Directives – ex. *Dacă bateria telefonului este gata, atunci trebuie pusă la încărcat*
      - Heuristics – ex. *Dacă lumina telefonului este stinsă, atunci bateria este plină*
  - Rules’ advantages
    - Easy to understand (natural form of knowledge)
    - Easy to derive inference and explanations
    - Easy to modify and maintain
  - Rules’ limits
    - Complex knowledge requires many rules
    - Search limitations in systems with many rules

Intelligent systems – KBS – RBS

Architecture

- Knowledge base (KB)
  - Specific information
- Inference engine (IE)
  - Rules utilized for generating new information
- User interface
  - Allows dialog with users and accessing facts and knowledge from KB (to be added or updated)
  - A facility for the user to interact with the system
- Knowledge Acquisition Facility
  - An automatic way to acquire knowledge
- Explanation Facility
  - Explains reasoning of the system to the user



Intelligent systems – KBS – RBS

Architecture – inference engine

- Content
  - Rules that can generate new information
  - Algorithms
  - RBS’s brain – a deduction algorithm based on KB and specific to reasoning method
  - Depends on complexity and type of knowledge
- Aim
  - Helps to explore KB for reasoning → solutions, recommendations or conclusions
- Typology
  - IE with *forward chaining*
  - IE with *backward chaining*

Intelligent systems – KBS – RBS

Architecture – inference engine with *forward chaining*

- Main idea
  - starts with the initial facts and keep using the rules to draw new conclusions (or take certain actions) given those facts.
  - The rules are of the form
    - left hand side (LHS) ==> right hand side (RHS).
  - Data-driven - Start from available information as it becomes available, then try to draw conclusions

- Example
  - **Question:** *Does employee Popescu get a phone?*
  - **Rule:** *If Popescu is an employee, he gets a phone.*
  - **Current Fact:** *Popescu is an employee.*
  - **Conclusion:** *Popescu gets a phone.*

Intelligent systems – KBS – RBS

Architecture – inference engine with *forward chaining*

- Algorithm
  - The execution cycle is
    - Repeat
      - Select a rule whose left hand side conditions match the current state as stored in the working memory.
      - Execute the right hand side of that rule, thus somehow changing the current state.
    - Until there are no rules which apply.
  - Remarks
    - Facts are represented in a working memory which is continually updated.
    - Rules represent possible actions to take when specified conditions hold on items in the working memory.
    - The conditions are usually patterns that must match items in the working memory,
    - The actions usually involve adding or deleting items from the working memory.

Intelligent systems – KBS – RBS

Architecture – inference engine with *backward chaining*

- Main idea
  - Starts from a potential conclusion (hypothesis) and search facts that explain it
  - The rules are of the form
    - left hand side (LHS) ==> right hand side (RHS).
  - Goal-driven - Start from a potential conclusion (hypothesis), then seek evidence that supports (or contradicts) it

- Example
  - **Question:** *Does employee John get a computer?*
  - **Statement**: *John gets a computer.*
  - **Current Fact**: *John is a programmer.*
  - **Rule**: *If employee is a programmer, then he gets a computer.*
  - Check the rule base to see what has to be “true” for John to get a computer. A programmer. Is it a fact that john is programmer. If true, then he gets a computer

Intelligent systems – KBS – RBS

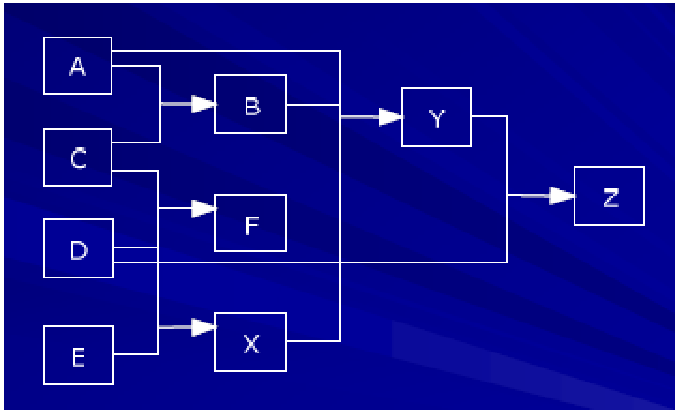
Architecture – inference engine with *backward chaining*

- Algorithm
  - Execution cycle
    - Start with a goal state
    - System will first check if the goal matches the initial facts given. If it does, the goal succeeds. If it doesn't, the system will looks for rules whose conclusions match the goal.
    - One such rule will be chosen, and the system will then try to prove any facts in the preconditions of the rule using the same procedure, setting these as new goals to prove.
  - Remarks
    - Needs to keep track of what goals it needs to prove to prove its main hypothesis.

Intelligent systems – KBS – RBS

Architecture – inference engine – example

- Knowledge base
  - Facts
    - A – runny nose
    - B – sinusitis
    - C – headache
    - D - vertigos
    - E – fever
    - F – blood problems
    - X – infection
    - Y – antibiotic
    - Z – bed rest
  - Rules
    - R1: *if A is true and C is true, then B is true*
    - R2: *if C is true and D is true then F is true*
    - R3: *if C is true and D is true then E is true*
    - R4: *if A is true and B is true and X is true then Y is true*
    - R5: *if Y is true and D is true then Z is true*
  - Goal
    - Fact Z



Intelligent systems – KBS – RBS

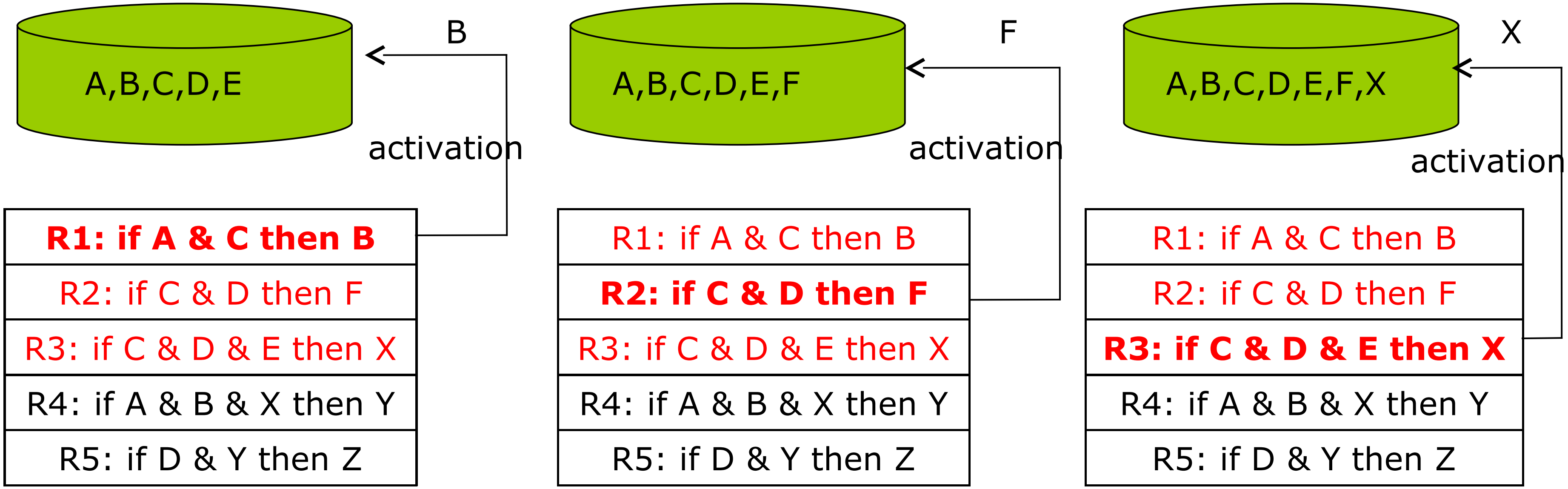
Architecture – inference engine – example

- *forward chaining* algorithm
  - repeat
    - Select rules that can be applied for facts of KB
      - Rules that contain in LHS facts from KB
    - If for a fact more rules can be applied, choose one of them (that has not been used)
    - Apply the selected rules and add into the KB the new resulted facts
  - Until the goal fact or a stop rule is reached

Intelligent systems – KBS – RBS

Architecture – inference engine – example

- Iteration 1



A,B,C,D,E

B

activation

R1: if A & C then B

R2: if C & D then F

R3: if C & D & E then X

R4: if A & B & X then Y

R5: if D & Y then Z

A,B,C,D,E,F

F

activation

R1: if A & C then B

R2: if C & D then F

R3: if C & D & E then X

R4: if A & B & X then Y

R5: if D & Y then Z

A,B,C,D,E,F,X

X

activation

R1: if A & C then B

R2: if C & D then F

R3: if C & D & E then X

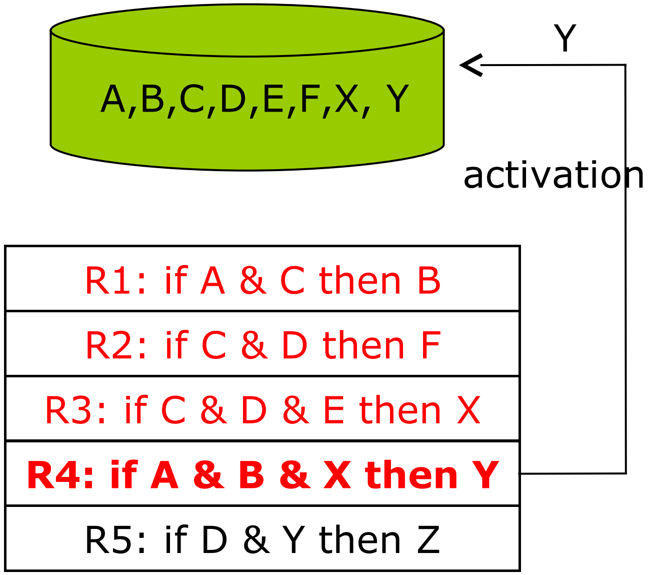
R4: if A & B & X then Y

R5: if D & Y then Z

Intelligent systems – KBS – RBS

Architecture – inference engine – example

- Iteration 2



A,B,C,D,E,F,X, Y

Y

activation

R1: if A & C then B

R2: if C & D then F

R3: if C & D & E then X

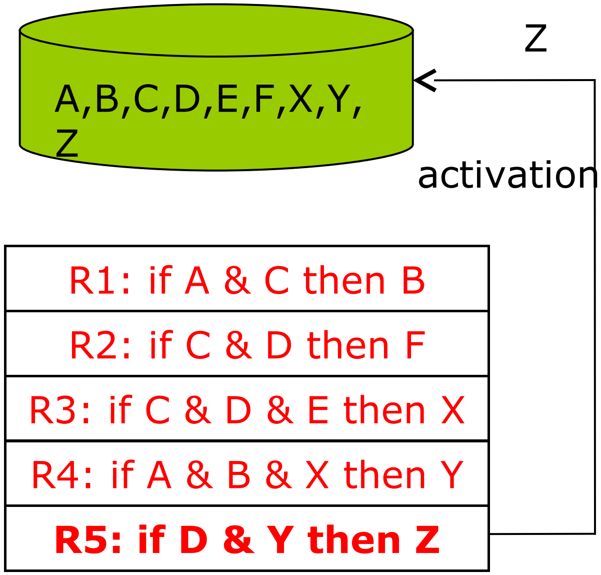
R4: if A & B & X then Y

R5: if D & Y then Z

Intelligent systems – KBS – RBS

Architecture – inference engine – example

- Iteration 3



R1: if A & C then B

R2: if C & D then F

R3: if C & D & E then X

R4: if A & B & X then Y

R5: if D & Y then Z

Intelligent systems – KBS – RBS

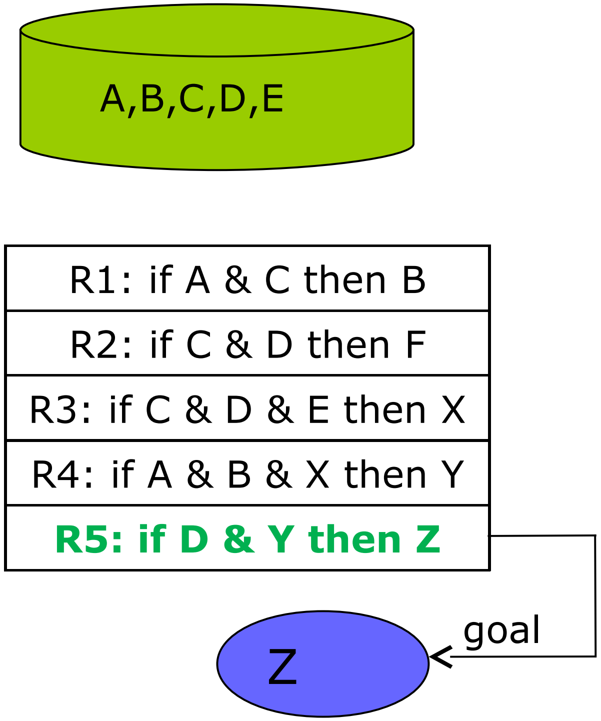
Architecture – inference engine – example

- *backward chaining* algorithm
  - repeat
    - Select the rules that match the goal
      - Rules that contains in RHS the goal fact
      - If more rules can be applied for a given goal, choose one of them
    - Check the selected rules
      - Replace the goal by causes in the selected rules (these causes become the new goals)
  - Until all the goals are true
    - Are known facts (from the KB)
    - Are information provided by user
  - Repeat
    - Apply the previous selected rules (in reverse order)
  - Until all the rules are applied and the goal is reached (as fact in KB)

Intelligent systems – KBS – RBS

Architecture – inference engine – example

- Iteration 1.1



A,B,C,D,E

R1: if A & C then B

R2: if C & D then F

R3: if C & D & E then X

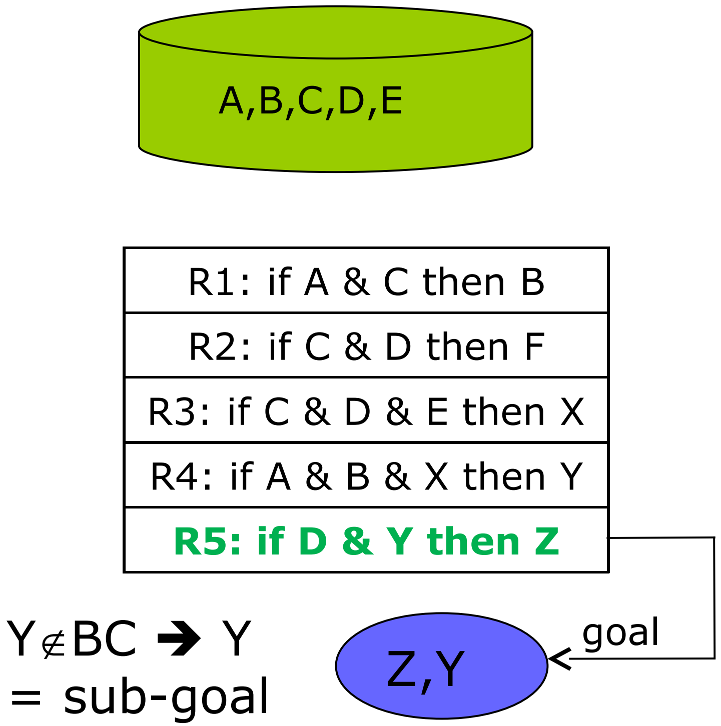
R4: if A & B & X then Y

R5: if D & Y then Z

Intelligent systems – KBS – RBS

Architecture – inference engine – example

- Iteration 1.1



A,B,C,D,E

R1: if A & C then B

R2: if C & D then F

R3: if C & D & E then X

R4: if A & B & X then Y

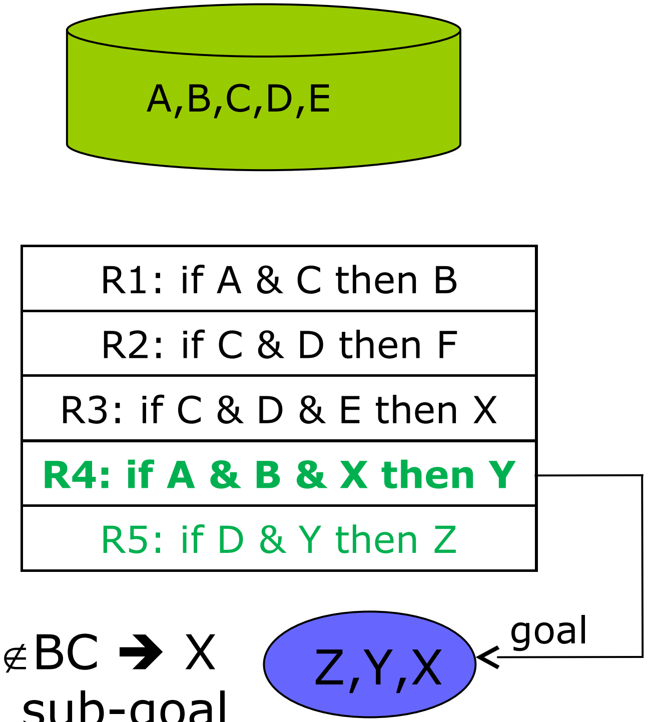
R5: if D & Y then Z

Y∉BC ➔ Y
= sub-goal

Intelligent systems – KBS – RBS

Architecture – inference engine – example

- Iteration 1.2



A,B,C,D,E

R1: if A & C then B

R2: if C & D then F

R3: if C & D & E then X

R4: if A & B & X then Y

R5: if D & Y then Z

goal

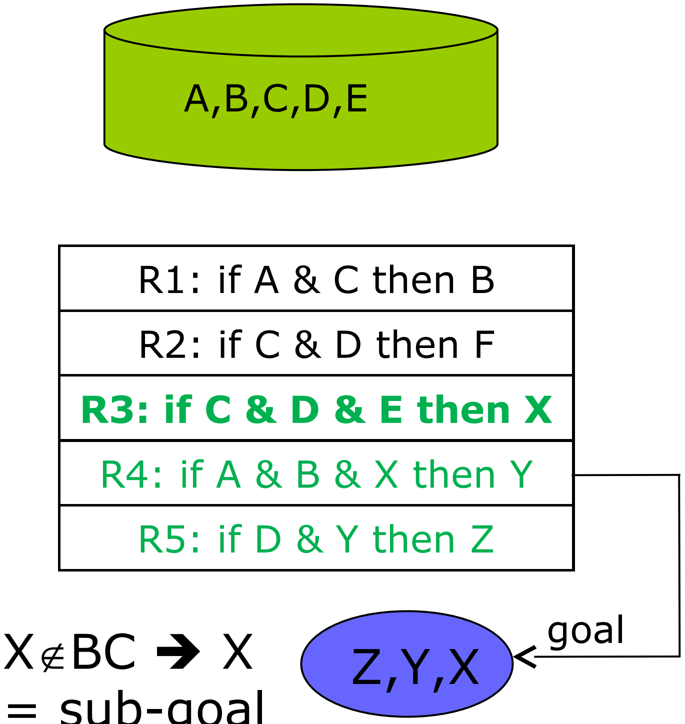
Z,Y,X

X∉BC ➔ X = sub-goal

Intelligent systems – KBS – RBS

Architecture – inference engine – example

- Iteration 1.2



A,B,C,D,E

R1: if A & C then B

R2: if C & D then F

R3: if C & D & E then X

R4: if A & B & X then Y

R5: if D & Y then Z

goal

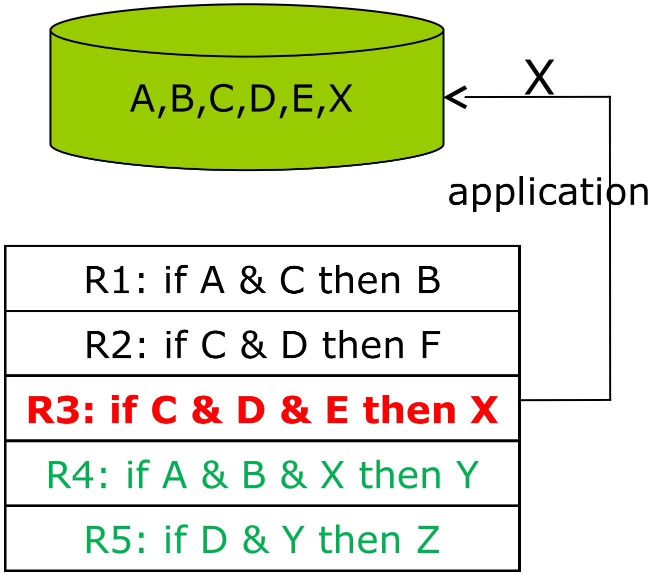
Z,Y,X

$X \notin BC$ → X = sub-goal

Intelligent systems – KBS – RBS

Architecture – inference engine – example

- Iteration 2.1



R1: if A & C then B

R2: if C & D then F

R3: if C & D & E then X

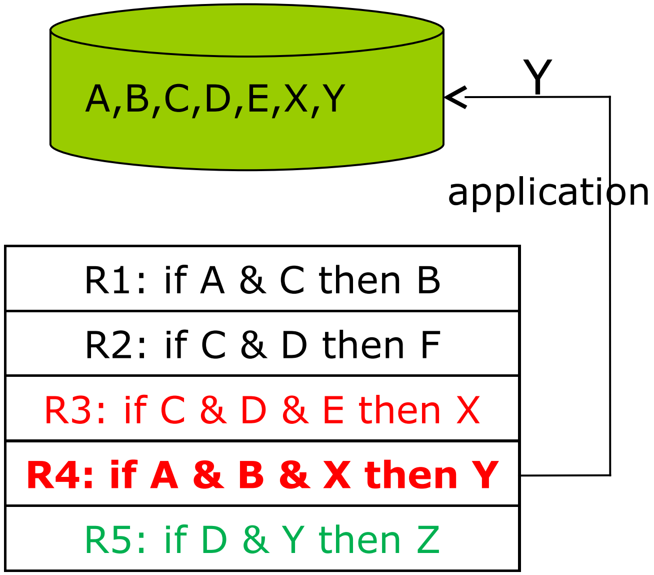
R4: if A & B & X then Y

R5: if D & Y then Z

Intelligent systems – KBS – RBS

Architecture – inference engine – example

- Iteration 2.1



A,B,C,D,E,X,Y

Y

application

R1: if A & C then B

R2: if C & D then F

R3: if C & D & E then X

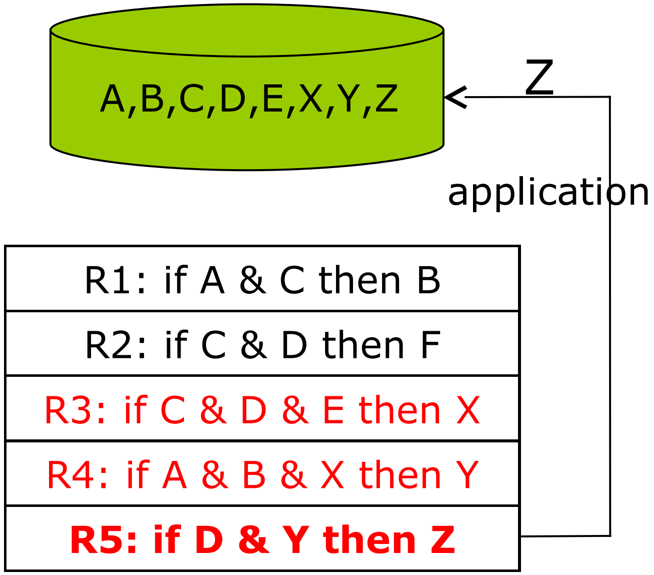
R4: if A & B & X then Y

R5: if D & Y then Z

Intelligent systems – KBS – RBS

Architecture – inference engine – example

- Iteration 2.1



R1: if A & C then B

R2: if C & D then F

R3: if C & D & E then X

R4: if A & B & X then Y

R5: if D & Y then Z

Intelligent systems – KBS – RBS

Architecture – inference engine

- Difficulties
  - *Forward Chaining* (FC) or *Backward chaining* (BC)?
  - Conflict resolution

Intelligent systems – KBS – RBS

Architecture – inference engine – dificulties

- *Forward Chaining* (FC) or *Backward chaining* (BC)?
  - FC should be used when:
    - All or most of the data is given in the problem statement;
    - There exist a large number of potential goals but only a few of them are achievable in a particular problem instance.
    - It is difficult to formulate a goal or hypothesis.
  - BC should be used when:
    - A goal or hypothesis is given in the problem statement or can be easily formulated
    - There are a large number of rules that match the facts, producing a large number of conclusions - choosing a goal prunes the search space.
    - Problem data are not given (or easily available)

Intelligent systems – KBS – RBS

Architecture – inference engine – dificulties

- Conflict resolution
  - When more than one rule is matched on the facts asserted.
    - Eg.
      - R1: if color is yellow, then the fruit is apple
      - R2: if color is yellow and shape is longish then the fruit is banana
      - R3: if shape is round then the fruit is apple
  - Approaches
    - **first** rule
    - **random** rule
    - **the most specific** rule
    - **the oldest** rule
    - **the best** rule

Intelligent systems – KBS – RBS

Architecture – inference engine – dificulties

- Conflict resolution
  - ***First applicable*** (***First in first serve***)
    - If the rules are in a specified order, firing the first applicable one is the easiest way to control the order in which rules fire. From a practical perspective the order can be established by ordering the rules in the knowledge base by placing them in the preferred order
    - Example
      - R1: if color is yellow, then the fruit is apple
      - R2: if color is yellow and shape is longish then the fruit is banana
      - R3: if shape is round then the fruit is apple
    - Remarks
      - Ordering the rules only works for small systems (of up to 100 rules).
      - Rules are sorted only in small systems.

Intelligent systems – KBS – RBS

Architecture – inference engine – dificulties

- Conflict resolution
  - Random:
    - A random strategy simply chooses a single random rule to fire from the conflict set. It is also advantageous even though it doesn’t provide the predictability or control of the first applicable strategy.
    - Example
      - R1: if color is yellow, then the fruit is apple
      - R2: if color is yellow and shape is longish then the fruit is banana
      - R3: if shape is round then the fruit is apple
    - Remarks
      - Selection can be a good one or a bad one

Intelligent systems – KBS – RBS

Architecture – inference engine – dificulties

- Conflict resolution
  - Most Specific (***Specificity***)
    - This strategy is based on the number of conditions of the rules. From the conflict set, the rule with the most conditions is chosen. This is based on the assumption that if it has the most conditions then it has the most relevance to the existing data. It can also be called longest matching strategy and it is based on the assumption that a specific rule process more information than a general one.
    - Example
      - R1: if color is yellow, then the fruit is apple
      - R2: if color is yellow and shape is longish then the fruit is banana
      - R3: if shape is round then the fruit is apple
    - Remarks
      - A specific rule processes more information than a general one ➔*longest matching strategy*

Intelligent systems – KBS – RBS

Architecture – inference engine – dificulties

- Conflict resolution

  - **Least Recently Used** (***Recent***):
    - **Each of the rules has a time or step** stamp (or time and date) associated, which marks the last time it was used.
    - Example
      - R1: if color is yellow, then the fruit is apple [12.01.2012 – 13.45]
      - R2: if color is yellow and shape is longish then the fruit is banana [7.02.2012 – 21.10]
      - R3: if shape is round then the fruit is apple [10.01.2012 – 10.25]
    - Remarks
      - the new rules have been added by an expert whose opinion is less trusted than that of the expert who added the earlier rules. In this case,, it clearly makes more sense to allow the earlier rules priority.

Intelligent systems – KBS – RBS

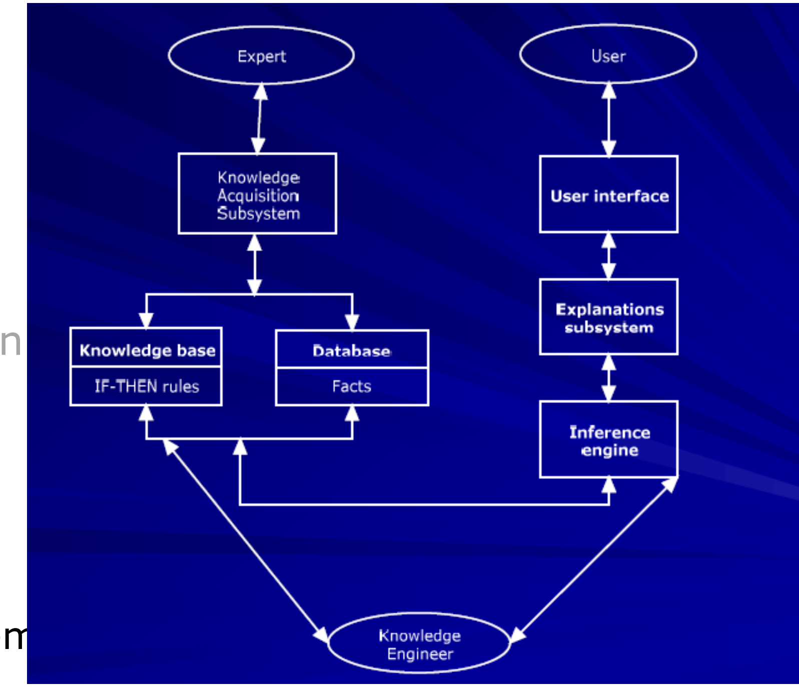
Architecture – inference engine – dificulties

- Conflict resolution
  - **"Best" rule (*Prioritization*):**
    - each rule is given a "weight" which specifies how much it should be considered over the alternatives. The rule with the most preferable outcomes is chosen based on this weight
    - Example
      - R1: if color is yellow, then the fruit is apple [30%]
      - R2: if color is yellow and shape is longish then the fruit is banana [30%]
      - R3: if shape is round then the fruit is apple [40%]
    - Remarks
      - Require human expertise for prioritization

Intelligent systems – KBS – RBS

Architecture

- Knowledge base (KB)
  - Specific information
- Inference engine (IE)
  - Rules utilized for generating new information
- User interface
  - Allows dialog with users and accessing facts and knowledge from KB (to be added or updated)
  - A facility for the user to interact with the system
- Knowledge Acquisition Facility
  - An automatic way to acquire knowledge
- Explanation Facility
  - Explains reasoning of the system to the user



Intelligent systems – KBS – RBS

Architecture

- User interface
  - Allows dialog with users and accessing facts and knowledge from KB (to be added or updated)
  - A facility for the user to interact with the system
    - Language processing methods
    - Menus
    - Graphical elements
- Knowledge Acquisition Facility
  - An automatic way to acquire knowledge

Intelligent systems – KBS – RBS

Architecture

- Explanation Facility
  - Explain to users
    - Knowledge of the system
    - Reasoning of the system for obtaining solutions
  - provides the user with a means of understanding the system behavior.
  - This is important because a consultation with a human expert will often require some explanation.
  - Many people would not always accept the answers of an expert without some form of justification.
  - ex. a medical expert providing a diagnosis/treatment of a patient is expected to explain the reasoning behind his/her conclusions: the uncertain nature of this type of decision may demand a detailed explanation so that the patient concerned is aware of any risks, alternative treatments ,etc.

Intelligent systems – KBS

- Rule-based systems (RBS)
  - Content and objectives
  - Design
  - Architecture
  - Tools and example
  - Advantages and limits

Intelligent systems – KBS – RBS

Tools

- PROLOG
  - Programming language based on *backward chaining*
- ART (Inference Corporation)
  - In 1984, Inference Corporation developed the Automated Reasoning Tool (ART), a forward chaining system.
- ART-IM (Inference Corporation)
  - Following the distribution of NASA's CLIPS, Inference Corporation implemented a forward-chaining only derivative of ART/CLIPS called ART-IM.
- CLIPS –
  - NASA took the forward chaining capabilities and syntax of ART and introduced the "C Language Integrated Production System" (i.e., CLIPS) into the public domain.
- OPS5 (Carnegie Mellon University)
  - OPS5 (Carnegie Mellon University) – First AI language used for Production System (XCON)
- Eclipse (The Haley Enterprise, Inc.)
  - Eclipse is the only C/C++ inference engine that supports both Forward and Backward chaining.

Intelligent systems – KBS – RBS

Examples

- DENDRAL (1965-1983)
  - rule-based expert systems that analyzes molecular structure. Using a plan-generate-test search paradigm and data from mass spectrometers and other sources, DENDRAL proposes plausible candidate structures for new or unknown chemical compounds.
- MYCIN (1972-1980)
  - MYCIN is an interactive program that diagnoses certain infectious diseases, prescribes antimicrobial therapy, and can explain its reasoning in detail
- EMYCIN, HEADMED, CASNET și INTERNIST
  - For medical domain
- PROSPECTOR (1974-1983)
  - **Provides advice on mineral exploration**
- TEIRESIAS
  - For information retrieval
- XCON (1978-1999)
  - **configure computers**
- Financial RBSs
  - ExpertTAX, Risk Advisor (Coopers & Lybrand), Loan Probe, Peat/1040 (KPMG), VATIA, Flow Eval (Ernst & Young), Planet, Compas, Comet (Price Waterhouse), Rice (Arthur Andersen), Audit Planning Advisor, World Tax Planner (Deloitte Touche)

Intelligent systems – KBS – RBS

Advantages and limits

- Advantages
  - Provide advises to non experts
  - Allows the organizations to replicate their very best people. Expert systems carry the intelligence and information found in the intellect of experts and provides this knowledge to other members of the organization.
  - Reduce the error due to automation of tedious, repetitive or critical tasks
  - Reduce the manpower and time required for system testing and data analysis
  - Reduce the costs through acceleration of fault observations
  - Eliminate the work that people should not do (such as difficult, time-consuming or error
  - prone tasks, jobs where training needs are large or costly).
  - Eliminates work that people would rather not do (such as jobs involving decision making,
  - which does not satisfy everyone; expert systems ensure fair decisions without favoritism
  - in such cases).
  - Expert systems perform better than humans in certain situations.
  - Perform knowledge acquisition, process analysis, data analysis, system verification
  - Increased visibility into the state of the managed system
  - Develop functional system requirements
  - Coordinate software development
  - For simple domains, the rule-base might be simple and easy to verify and validate.
  - Expert system shells provide a means to build simple systems without programming.
  - Provide consistent answers for repetitive decisions, processes and tasks
  - Hold and maintain significant levels of information
  - Reduces creating entry barriers to competitors.

- Limits
  - Narrow domain
  - Limited focus
  - Instability to learn
  - Maintenance problems
  - Developmental cost

Review

- KBSs
  - Computational systems that
  - Overlap KB and IE
- Types of KBSs
  - LBSs
    - Explore knowledge by using methods of formal logic
    - Components
      - Language (syntax and semantics) and
      - Deduction method (reasoning)
  - RBSs
    - Explore knowledge by using methods of human logic
    - Can solve problems that do not have a deterministic solution
    - Try to simulate a human expert
    - Components
      - KB → facts and rules
      - IE → forward and backward chaining

Next lecture

A. Short introduction in Artificial Intelligence (AI)

A. Solving search problems
   A. Definition of search problems
   B. Search strategies
      A. Uninformed search strategies
      B. Informed search strategies
      C. Local search strategies (Hill Climbing, Simulated Annealing, Tabu Search, Evolutionary algorithms, PSO, ACO)
      D. Adversarial search strategies

C. **Intelligent systems**
   A. Rule-based systems in certain environments
   B. **Rule-based systems in uncertain environments (Bayes, Fuzzy)**
   C. Learning systems
      A. Decision Trees
      B. Artificial Neural Networks
      C. Support Vector Machines
      D. Evolutionary algorithms
   D. Hybrid systems

Next lecture – useful information

- Chapter V of *S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995*
- Chapter 3 of *Adrian A. Hopgood, Intelligent Systems for Engineers and Scientists, CRC Press, 2001*
- Chapters 8 and 9 of *C. Groşan, A. Abraham, Intelligent Systems: A Modern Approach, Springer, 2011*

- Presented information have been inspired from different bibliographic sources, but also from past AI lectures taught by:
  - PhD. Assoc. Prof. Mihai Oltean – www.cs.ubbcluj.ro/~moltean
  - PhD. Assoc. Prof. Crina Groşan - www.cs.ubbcluj.ro/~cgrosan
  - PhD. Prof. Horia F. Pop - www.cs.ubbcluj.ro/~hfpop