PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

ARTIFICIAL INTELLIGENCE

The Very Idea

Vasant G. Honavar

Dorothy Foehr Huck and J. Lloyd Huck Chair in Biomedical Data Sciences and Artificial Intelligence
Professor of Data Sciences, Informatics, Computer Science, Bioinformatics & Genomics and Neuroscience
Director, Artificial Intelligence Research Laboratory
Director, Center for Artificial Intelligence Foundations and Scientific Applications
Associate Director, Institute for Computational and Data Sciences
Pennsylvania State University

vhonavar@psu.edu
http://faculty.ist.psu.edu/vhonavar
http://ailab.ist.psu.edu

Mid-term Topics

- What is AI?
- Theories of intelligence
- Tests of intelligence
- Intelligence in nature
- Computing machines
- Computational theory of mind
- Relation of AI to other disciplines
- Goals of AI
- Intelligent agents Agents and their environments
- Knowledge representation
- Problem-solving as search

Reasoning with Knowledge

"Knowledge is power" – Francis Bacon

- The approach to automating problem-solving using state-space search is general, and hence broadly applicable
- Here we focus on a special case of knowledge-based systems or expert systems where problem domain specific knowledge is encoded in IF-THEN rules
- Expert systems
  - Encode human knowledge into IF-THEN rules to solve problems in specific domains, e.g., medicine, chemistry, etc.
  - Do not seek to exhibit general intelligence, but rather, to solve problems that require expertise – typically in settings where such experts are scarce.

Reasoning with Knowledge

- Consider a company that manufactures automobiles.
- As part of its customer support efforts, the company has set up call center to help customers experiencing problems with their automobiles.
- For example, the call center may receive a call from a customer complaining that his car does not start.
- The call center worker must help the customer by diagnosing the cause of the problem.

Reasoning with Knowledge

- Suppose the call is answered by an expert mechanic who knows
  - automobile engines will not start if either the ignition or the battery or both are faulty
- To pinpoint the problem, the mechanic might use her knowledge
  - if the battery is faulty, the headlights won't work.
- To keep things simple, we are ignoring other possible causes like faulty spark-plugs, carburetor, fuel injector, and so on
- The operator might ask the caller to try turning on the headlights.
  - If she is told that the headlights work just fine, she can diagnose the problem to be faulty ignition.
  - If on the other hand, if she is told that the lights don't work, she may suggest that the caller might need to get the battery replaced.

Reasoning with Knowledge

- Human expertise is scarce – as are knowledgeable mechanics
- Human experts are prone to stress, fatigue, etc. that impacts their performance
- Can AI help? How?
- The automobile company could use an expert system that plays the role of the mechanic on call

How can we design an expert system for auto diagnosis?

- Suppose we limit ourselves to malfunctions of only the Battery, Bulbs, Wipers, and Ignition
- Suppose the observable symptoms are limited to:
  - Headlights work/don't work;
  - Engine starts/doesn't start;
  - Wipers work/don't work
- Example: If the engine does not start, we can conclude that either the battery or the ignition are to blame.
- Real-world settings can involve hundreds of possible observable symptoms involving large numbers of components.
- While this increases the complexity of the problem, it does not necessarily impact the overall approach to the problem.

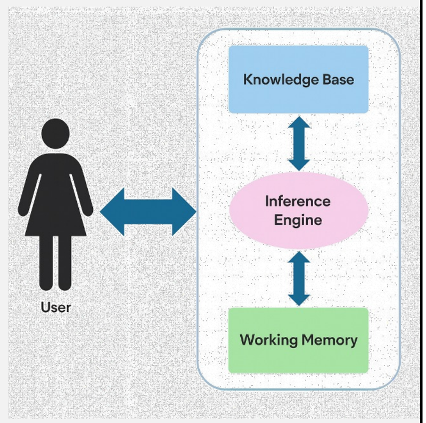
How can we design an expert system for auto diagnosis?

- Develop a set of rules that capture the mechanic's knowledge
- Knowledge engineers need to pick the brains of expert mechanics to elucidate the knowledge needed

Rule1 : IF Headlights don't work THEN faulty Bulbs or/and Battery
Rule 2: IF faulty Bulbs or/and Battery AND Engine does not start THEN faulty Battery
Rule 3: IF faulty Bulbs or/and Battery AND Engine starts THEN faulty Bulbs

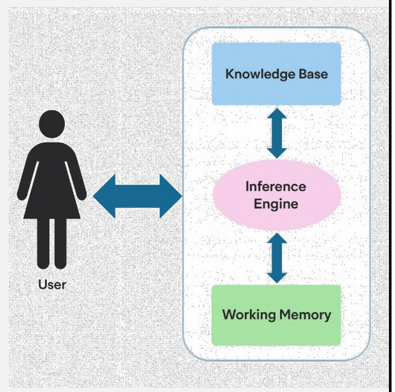
How does an expert system work?

- A knowledge base encodes relevant knowledge, e.g., rules of the form IF antecedents THEN consequences
- An inference engine draws conclusions by applying the rules in the knowledge base to the assertions supplied by the user.
- A working memory is used as a scratch pad to store user-supplied assertions and any assertions derived during the inference process.

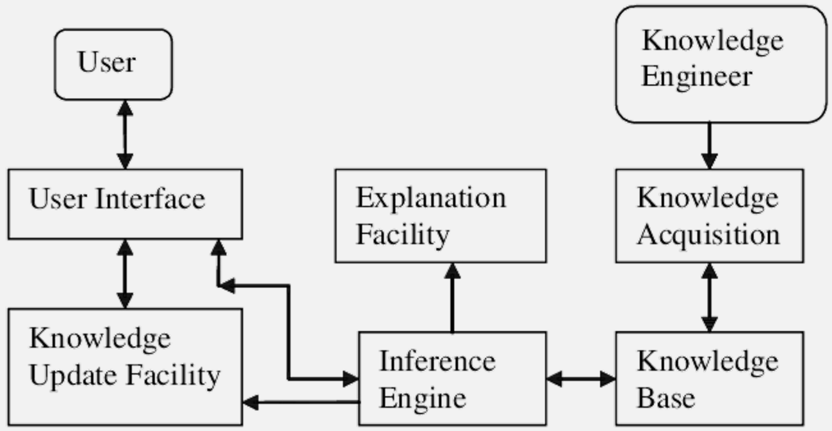


How does an expert system work?

- During a consultation with an expert system the user enters the facts relevant for the problem to be solved.
- The system matches the user-supplied facts in its working memory with the knowledge contained in the knowledge base to infer additional facts which go into the working memory.
- The process continues until the system either completes the task presented to it or runs into a dead end.



Design of an expert system

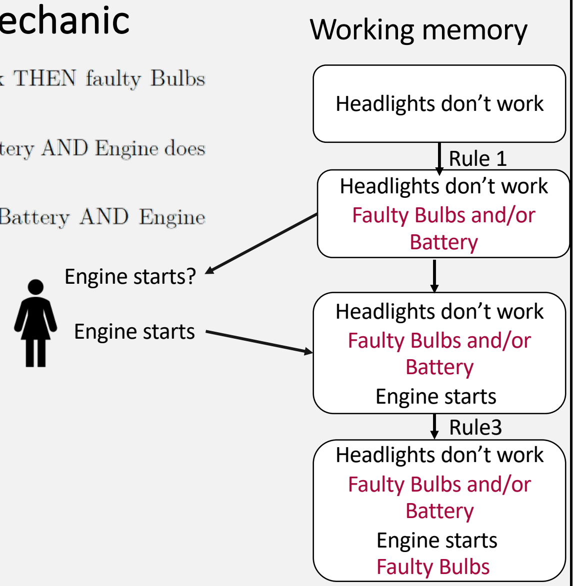


Rule-based auto mechanic

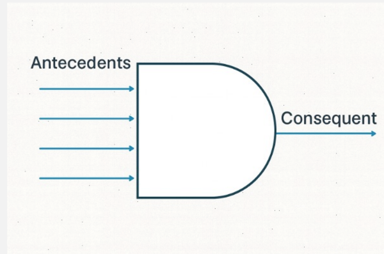
Rule1 : IF Headlights don't work THEN faulty Bulbs or/and Battery

Rule 2: IF faulty Bulbs or/and Battery AND Engine does not start THEN faulty Battery

Rule 3: IF faulty Bulbs or/and Battery AND Engine starts THEN faulty Bulbs



Rule-based systems



- Rule based systems use IF-THEN rules
  - IF – antecedents
  - THEN – consequent

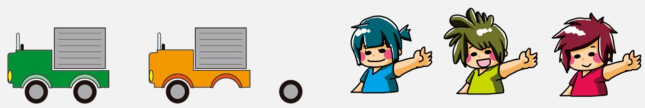
Forward chaining

- If all the antecedents match the working memory, then the rule is triggered
- Triggered rules get on an Agenda with some priority relative to each other
- The triggered rule with the highest priority fires
- When a rule fires, the consequents are added to the working memory
- The process repeats

Rule-based systems

- Deductive systems
  - Rule firing simply adds new assertions to the working memory
- Reactive systems
  - Rule firing results in an action
  - Action may result in some working memory assertions being deleted and new ones added

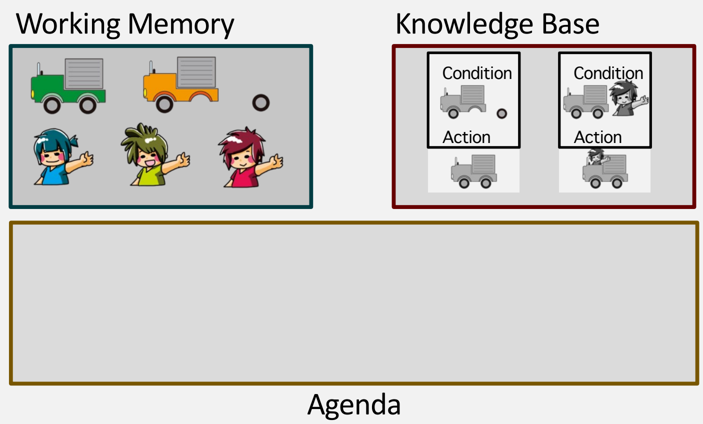
Example: Expert system for managing a fleet of trucks



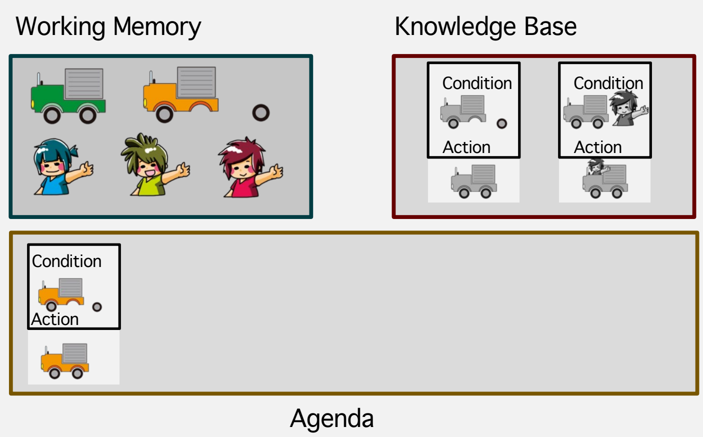
Rules

- IF a truck is is broken and parts are available, THEN fix the truck
- IF a truck is without a driver and a driver is available THEN assign it the driver

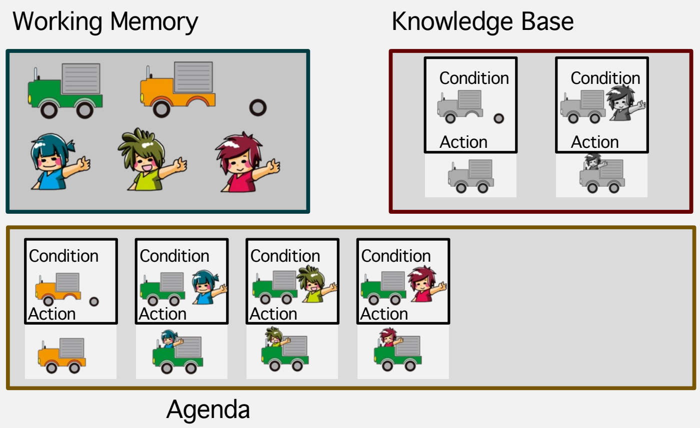
Example Expert system for managing a fleet of trucks



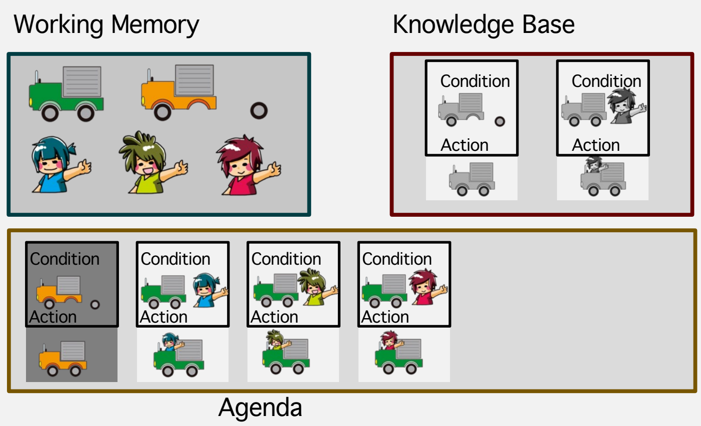
Example: Expert system for managing a fleet of trucks



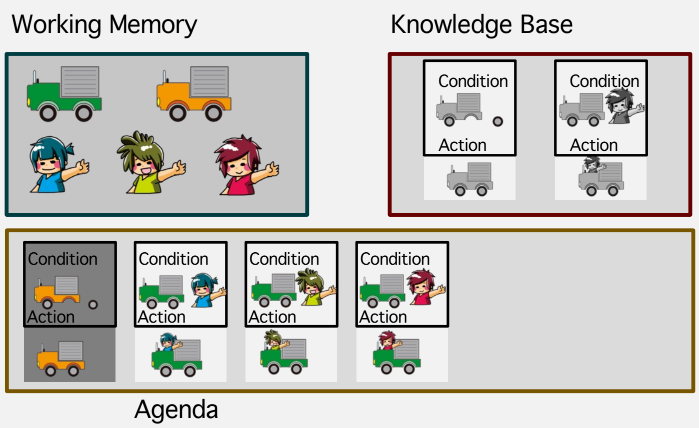
Example: Expert system for managing a fleet of trucks



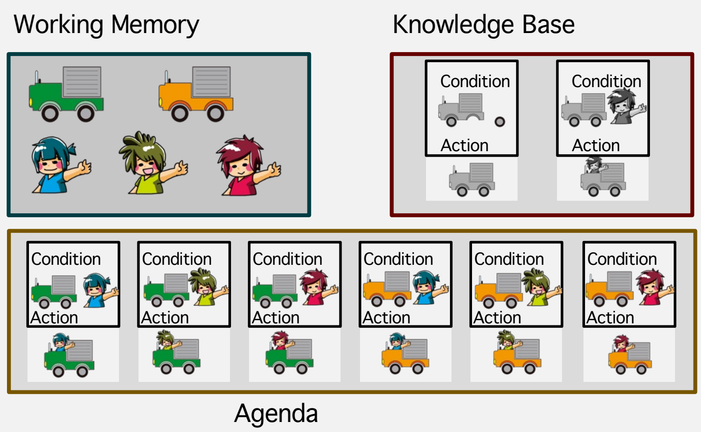
Example: Expert system for managing a fleet of trucks



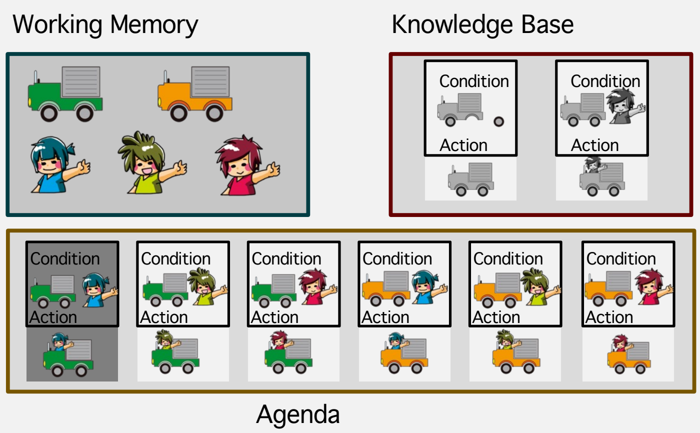
Example: Expert system for managing a fleet of trucks



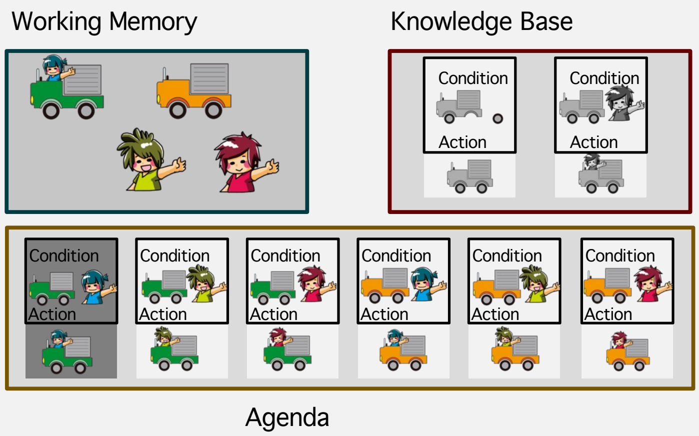
Expert system for managing a fleet of trucks



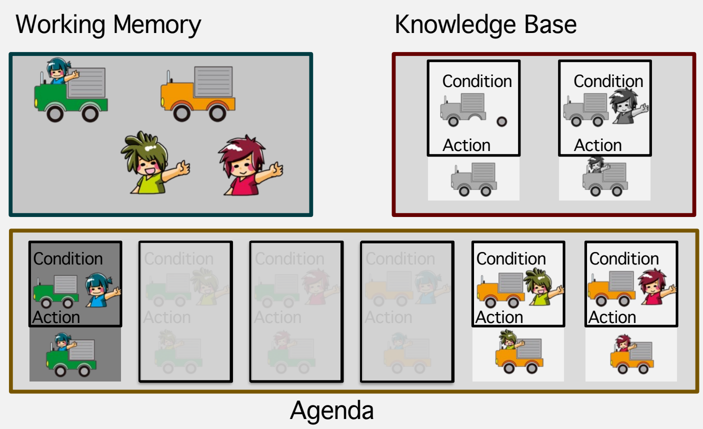
Example: Expert system for managing a fleet of trucks



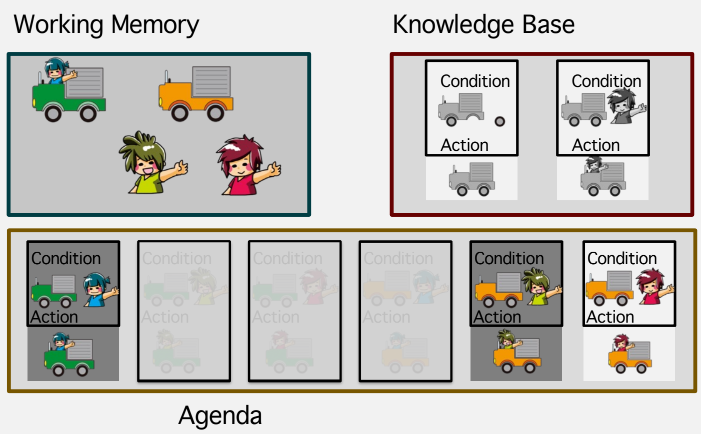
Example: Expert system for managing a fleet of trucks



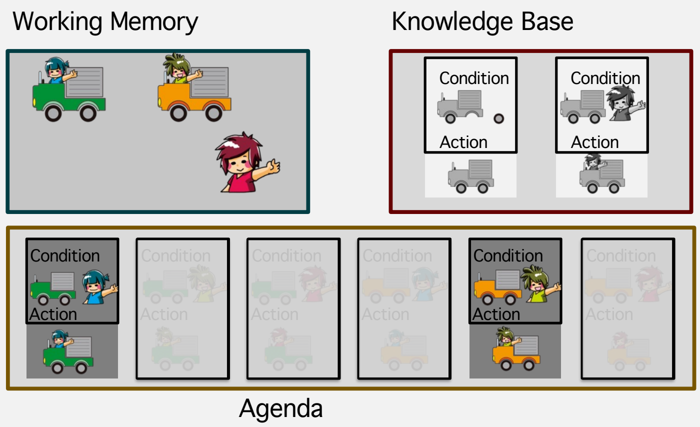
Expert system for managing a fleet of trucks



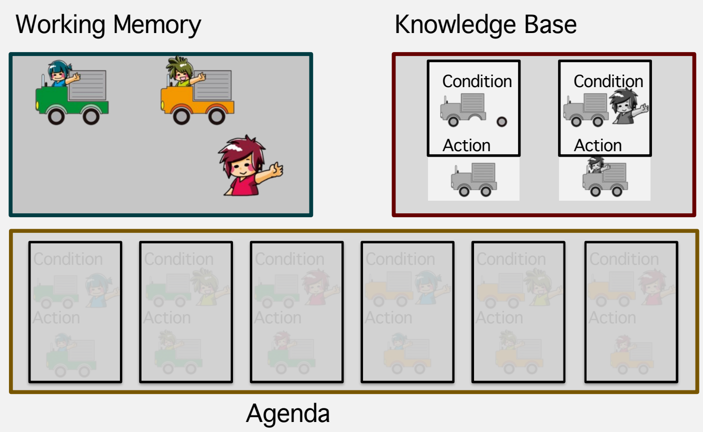
Example: Expert system for managing a fleet of trucks



Example: Expert system for managing a fleet of trucks



Example: Expert system for managing a fleet of trucks



Rules come in many flavors

Relations

IF the 'fuel tank' is empty
THEN the car is dead

IF the 'mass' is m
AND 'acceleration' is a
THEN the 'force' is ma

Recommendations

IF the season is autumn
AND the sky is cloudy
AND the forecast is drizzle
THEN the advice is 'take an umbrella'

Rules come in many flavors

Directive

IF the car is dead
AND the 'fuel tank' is empty
THEN the action is 'refuel the car'

Strategy

IF the car is dead
THEN the action is 'check the fuel tank';
use fuel rules

IF use fuel rules
AND the 'fuel tank' is full
THEN the action is 'check the battery';
use engine rules

Rules come in many flavors

Guidance

IF	the spill is liquid
AND	the 'spill pH' < 6
AND	the 'spill smell' is vinegar
THEN	the 'spill material' is 'acetic acid'

Law

IF	the traffic light is red
THEN	the driver must stop

IF	the traffic light is green
THEN	the driver may go

Inference in expert systems: Forward Chaining

- Data driven way of deriving a conclusion from given facts and using rules in the knowledge base
- When the antecedents of the rules match the facts in the working memory the consequents of the rules are added to the working memory

Inference in Expert Systems: Backward Chaining

- Goal driven matching of rules to derive antecedents that need to hold
  - Inference rules are applied by matching the goal to the consequents of rules in the knowledge base
  - When the goal matches the consequent of a rule, the antecedents of the rule are added to the list of goals

Forward Chaining

Rules

1. IF A AND B THEN C
2. IF A THEN D
3. IF C AND D THEN E
4. IF B AND E AND F THEN G
5. IF A AND E THEN H
6. IF D AND E AND H THEN I

Facts: A, B, F

Goal: H

Facts	Rules triggered	Rule fired
A,B,F	1,2	1
A,B,C,F	2	2
A,B,C,D,F	3	3
A,B,C,D,E,F	4,5	4
A,B,C,D,E,F,G	5	
A,B,C,D,E,F,G,H		

		Triggers: 1,2	Fires: 1
Add: C	Triggers: 2		Fires: 2
Add: D	Triggers: 3		Fires: 3
Add: E	Triggers: 4,5	Fires: 4	
Add: G	Triggers: 5		Fires: 5
Add: H	Triggers: 6		STOP

Backward Chaining

Rules
1. IF A AND B THEN C
2. IF A THEN D
3. IF C AND D THEN E
4. IF B AND E AND F THEN G
5. IF A AND E THEN H
6. IF D AND E AND H THEN I

Facts: A, B, F

Goal: H

Facts	Goals	Matching Rules
A,B,F	H	5
A,B,F	H,E	3
A,B,F	H,E,C,D	1,2
A,B,C,D,F	H,E	3
A,B,C,D,E,F	H	5
A,B,C,D,E,,F,H		

Add:		Goal: H	Match: 5
Add:		Goal: E	Match: 3
Add:		Goal: C,D	Match: 1
Add: C	Goal: D	Match: 2	
Add: D			STOP

Example: Expert System for Patient Monitoring

- Rule 1:
  - If Patient has cough
  - AND Patient has Infection
  - Then Patient has Pneumonia
- Rule 2:
  - If Temperature exceeds 100 degrees Fahrenheit
  - Then Patient has Fever
- Rule 3:
  - If Patient has been sick
  - AND Patient has fever
  - Then Patient has Infection

Example: Expert System for Patient Monitoring

Facts

- Patient temperature is 103 degrees Fahrenheit
- Patient has been sick
- Patient has severe cough

Forward Chaining in Action

Query: Does the patient have Pneumonia?

- Rule 1:
- If Patient has cough
- AND Patient has Infection
- Then Patient has Pneumonia
- Rule 2:
- If Temperature exceeds 100 degrees Fahrenheit
- Then Patient has Fever
- Rule 3:
- If Patient has been sick
- AND Patient has fever
- Then Patient has Infection

Matching	Status	Working Memory
Rule 1 Antecedent 1	True	Temp=103 Sick cough
Rule 1 Antecedent 2	Unknown	Temp=103 Sick cough
Rule 2 Antecedent 1	True	Temp=103 Sick cough Fever
Rule 3 Antecedent 1	True	Temp=103 Sick cough Fever
Rule 3 Antecedent 2	True	Temp=103 Sick cough Fever Infection

Forward Chaining in Action

Query: Does the patient have Pneumonia?

- Rule 1:
- If Patient has cough
- AND Patient has Infection
- Then Patient has Pneumonia
- Rule 2:
- If Temperature exceeds 100 degrees Fahrenheit
- Then Patient has Fever
- Rule 3:
- If Patient has been sick
- AND Patient has fever
- Then Patient has Infection

Matching	Status	Working Memory
Rule 1 Antecedent 1	True	Temp=103 Sick Cough Fever Infection
Rule 1 Antecedent 2	True	Temp=103 Sick Cough Fever Infection Pneumonia

With Pneumonia added to working memory, we can conclude that the patient has Pneumonia

Backward Chaining in Action

Query: Does the patient have Pneumonia?

- Rule 1:
- If Patient has cough
- AND Patient has Infection
- Then Patient has Pneumonia
- Rule 2:
- If Temperature exceeds 100 degrees Fahrenheit
- Then Patient has Fever
- Rule 3:
- If Patient has been sick
- AND Patient has fever
- Then Patient has Infection

Goal-driven backward chaining	Working Memory
Goal: The patient has pneumonia.	
Goal: Pneumonia; Matching rule: Rule1	
Premise 1 of Rule 1 is “Patient has cough”	
Cough? Response: Yes	Cough
Premise 2 of Rule 1 is “Patient has Infection”	Cough
Goal: Infection; Matching rule: Rule3	Cough
Premise 1of Ryle 3 is “Patient has been sick” Premise 2 of Rule 3 is “Patient has fever”	Cough
Sick? Response: Yes	Cough Sick
Goal: Fever: Matching rule: Rule 2	Cough Sick

Backward Chaining in Action

Query: Does the patient have Pneumonia?

- Rule 1:
- If Patient has cough
- AND Patient has Infection
- Then Patient has Pneumonia
- Rule 2:
- If Temperature exceeds 100 degrees Fahrenheit
- Then Patient has Fever
- Rule 3:
- If Patient has been sick
- AND Patient has fever
- Then Patient has Infection

Goal-driven backward-chaining	Working Memory
Premise of Rule 2 is “Temperature exceeds 100 degrees Fahrenheit	Cough Sick
Temp > 100 degrees Fahrenheit ? Response: Yes Rule 2 Fires	Cough Sick Fever
Rule 3 Fires	Cough Sick Fever Infection
Rule 1 Fires	Cough Sick Fever Infection Pneumonia

Rob, the robot Zoo-Guide: Rules for recognizing zoo animals

- Z_1: IF $?x$ has hair THEN $?x$ is a mammal
  - $?x$ is a variable that can be instantiated through a match with working memory
  - Suppose working memory contains Maya has hair
  - Then the antecedent of rule Z_1 match the working memory
  - Upon firing of rule Z_1 Maya is a mammal is added to the working memory
- When multiple rules match, they get on an Agenda with some mechanism for prioritization among the different matching rules

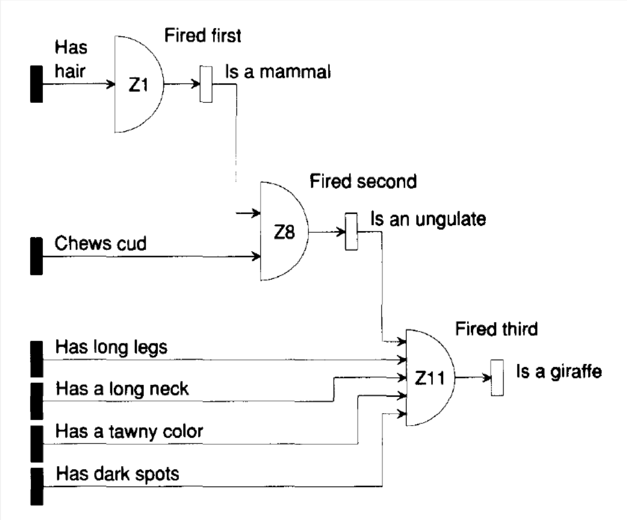
Rob, the robot Zoo-Guide: Rules for recognizing zoo animals

- Z_2: IF $?x$ gives milk THEN $?x$ is a mammal
- Z_3 IF $?x$ has feathers THEN $?x$ is a bird
- Z_4 : IF $?x$ flies AND $?x$ lays eggs THEN $?x$ is a bird
- Z_5: IF $?x$ is a mammal AND $?x$ eats meat THEN $?x$ is a carnivore
- Z_6 : IF $?x$ is a mammal AND $?x$ has pointed teeth AND $?x$ has claws AND $?x$ has forward-pointing eyes THEN $?x$ is a carnivore
- Z_7 : IF $?x$ is a mammal AND $?x$ has hoofs THEN $?x$ is an ungulate
- Z_8 : IF $?x$ is a mammal AND $?x$ chews cud THEN $?x$ is an ungulate
- Z_9 : IF $?x$ is a carnivore AND $?x$ has tan color AND $?x$ has dark spots THEN $?x$ is a cheetah
- Z_{10} : IF $?x$ is a carnivore AND $?x$ has tan color AND $?x$ has black stripes THEN $?x$ is a tiger
- Z_{11} : IF $?x$ is an ungulate AND $?x$ has long legs AND $?x$ has long neck AND $?x$ has tan color AND $?x$ has dark spots THEN $?x$ is a giraffe
- ..

Rob, the robot Zoo-Guide: Forward chaining

- Maya has hair.
- Maya chews cud.
- Maya has long legs.
- Maya has a long neck.
- Maya has tan color.
- Maya has dark spots.
- Because Maya has hair, rule Z_1 fires, and asserts that Maya is a mammal.
- Because Maya is a mammal and chews cud, rule Z_8 fires and asserts that Maya is an ungulate.
- At this point, all the antecedents for rule Z_{11} match the working memory, allowing Z_{11} to fire and inform Rob that Maya is a giraffe.

Rob, the robot Zoo-Guide: Forward Chaining

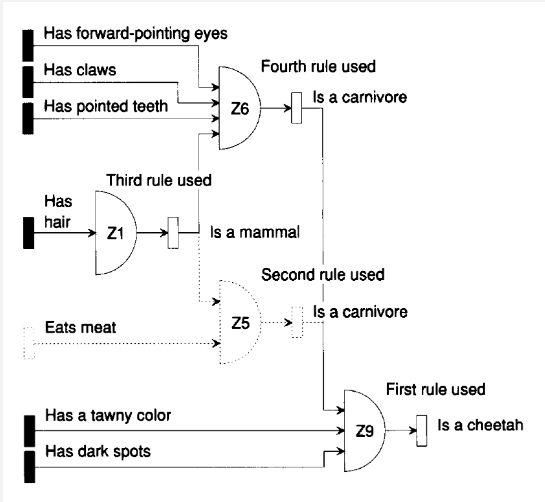


Forward chaining proceeds from facts to conclusions

Rob, the robot Zoo-Guide: Forward Chaining

- Until no rule produces a new assertion or the animal is identified,
  - For each rule,
    - Try to support each of the rule's antecedents by matching it to known facts.
    - If all the rule's antecedents are satisfied, assert the consequent unless there is an identical assertion already.
    - Repeat for all alternative matches.

Rob, the robot Zoo-Guide: Backward Chaining



Backward chaining proceeds from hypothesis to facts

Rob, the robot Zoo-Guide: Backward Chaining

- For each hypothesis,
  - For each rule whose consequent matches the current hypothesis,
    - Try to support each of the rule's antecedents by
      - matching it to assertions in working memory or by
      - backward chaining through another rule, creating new hypotheses.
    - If all the rule's antecedents are supported, announce success and conclude that the hypothesis is true.

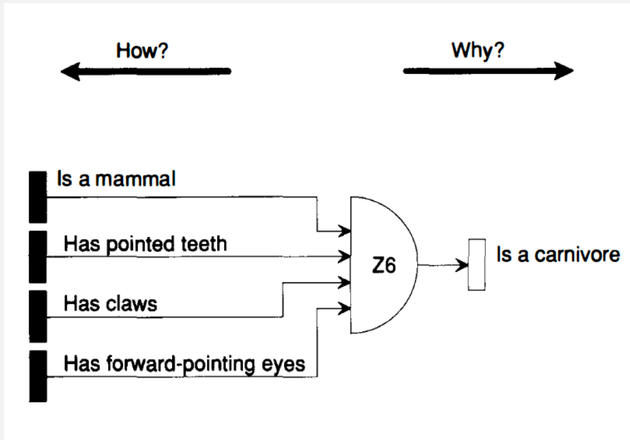
To Forward chain or backward chain?

- Whenever the rules are such that a typical set of assertions can lead to many conclusions, the rule system has a high degree of fan-out which argues for backward chaining.
- On the other hand, whenever the rules are such that a typical hypothesis can lead to many questions, rule system has a high degree of fan-in which argues for forward chaining.

To Forward chain or backward chain?

- In many situations, neither fan-out nor fan-in dominates
- If you have not yet gathered any facts, and you are interested in only whether one of many possible conclusions is true, use backward chaining.
  - Example: You do not care about the identity of an animal. All you care about is whether it is a carnivore.
- If you already have in hand all the facts you are ever going to get, and you want to know everything you can conclude from those facts, use forward chaining.
  - Example: You have just seen an animal run by. You want to know what you can deduce about it.

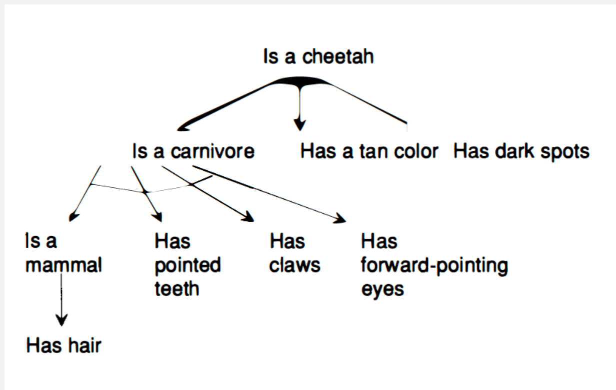
Explaining inferences: Forward chaining



Explaining inferences

- If the question is a how question, report the assertions connected to the IF side of the rule that established the assertion referenced in the question.
- If the question is a why question, report the assertions connected to the THEN sides of all rules that used the assertion referenced in the question

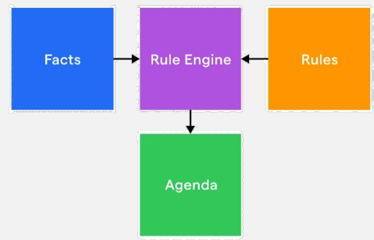
Explaining inferences: Backward chaining



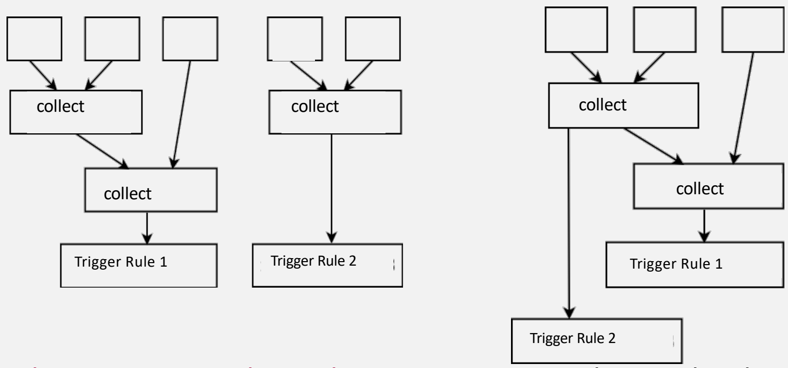
Making reasoning efficient: RETE network

- The naïve approach to rule matching repeats work because IF conditions can be shared by multiple rules
- This can be avoided by compiling the rules into a network so that once a particular IF condition is matched, the match is shared with all of the rules that depend on it
- A rule is triggered when all of its IF conditions are matched

Expert System



Making reasoning efficient: RETE network



The naïve approach to rule matching repeats work when IF conditions are shared by multiple rules

Rete network compiles the rules so that once a particular IF condition is matched, the match is shared with all of the rules that depend on it

Expert systems in practice: R1/XCON

- Expert systems were among the first commercial successes of AI
  - R1 – an expert system designed by John McDermott of CMU in collaboration with Digital Equipment Corporation (DEC) later named XCON (Expert configurer)
  - Configuring computer system orders
  - Started with about 750 rules in 1982 and grew to 2500 rules by 1984
  - By 1986, XCON had processed 80,000 orders, and achieved 95–98 percent accuracy
  - XCON saved DEC between 10 and 40 million dollars per year

Expert systems in practice: MYCIN

- MYCIN was an expert system for clinical consultation about diagnosis and treatment of bacterial infections designed by Ted Shortliffe of Stanford
- MYCIN informs itself about patients by asking questions symptoms, history, and laboratory-test results
- Question asked by MYCIN is determined by MYCIN's current hypothesis and the answers it has received so far
- MYCIN could also explain its reasoning in simplified English.
- MYCIN matched the performance of human specialists in blood infections and outperformed general practitioners.
- MYCIN never saw deployment in actual clinical setting due to legal and regulatory concerns

Expert systems in practice

- According to the results of a survey by Durkin, as of 1996, more than 2500 expert systems were deployed.
- Durkin speculated that this figure represents only about a fifth of systems that might have been developed.
- Rule- based expert systems are widely used in a broad range of applications.

Expert systems in practice

Knowledge Engineering Bottleneck

- The experts providing the domain knowledge must be able to express their knowledge in the form of simple rules.
- The knowledge engineer working with the experts should be able to grasp the expert's knowledge well enough to encode it in the knowledge base
- Knowledge engineering bottleneck is responsible for the exorbitant cost and effort involved in developing expert systems for real-world applications.
- Machine learning offers a way out -- by giving expert systems the ability to acquire the knowledge needed by observing experts or interacting with the world

Expert systems in practice

Engineering challenge

- The interaction between the knowledge base, the inference mechanism, the rule prioritization (conflict resolution) strategies and the order in which facts enter the working memory can be complex
- Real-world expert systems require painstaking debugging

Expert Systems: Summary

Advantages of expert systems

- In domains where the relevant expert knowledge can be codified in the form of IF-THEN rules, expert systems
  - offer a simple and straightforward approach to building systems that mimic experts
  - can use forward chaining or backward chaining for inference
  - can be deductive or reactive
  - can explain their inferences
  - Have been fielded across a broad range of applications

Drawbacks of expert systems

- Codifying knowledge in the form of a small number of rules can be challenging
- Knowledge elicitation can be challenging