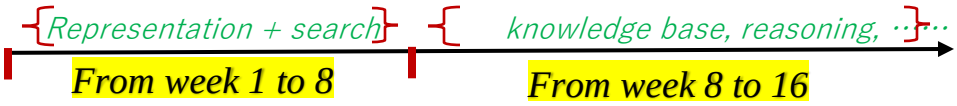
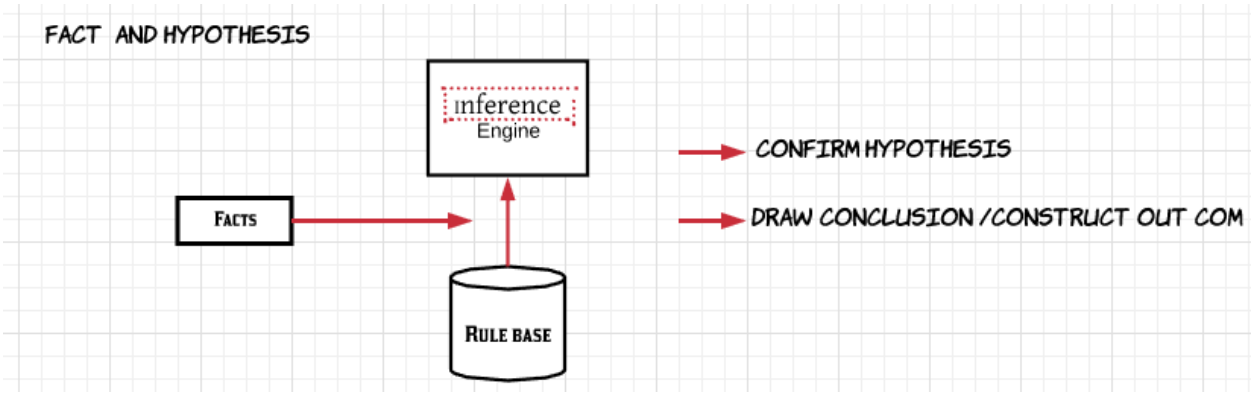
Time line of course



Expert system introduction

What is the reason like to do this "expert system"?

The big goal of AI in 70s'and 80s'was to build system called knowledge based system that would allow or provide decision support for humans in term of guiding them what to do!



1. Rule base
   - A set of productions { if < condition > then < action > / < Premise > then < conclusion >

   Example

   If it flies and has wings then it is bird

   It could be bird or airplane there is some degree of certainty in this and some degree of uncertainty.

2. *Facts*
   - *Observable data*

3. *Inference Engine*
   - *Module that uses facts, working memory and rules to reason draw conclusion or to construct an outcome.*

Example

- *R1 if green then produce*
- *R2 if package is small then it is delicacy*
- *R3 if refrigerator or produce then it is perishable*
- *R4 if it is weighs >=5 Ibs and it is not perishable then it is stable*
- *R5 if perishable and >=5 Ibs then it is turkey*
- *R6 if >=5 Ibs and produce then it is watermelon*

There is two way of doing reasoning

1) *Forward chaining*

2) *Backward chaining*

How to work

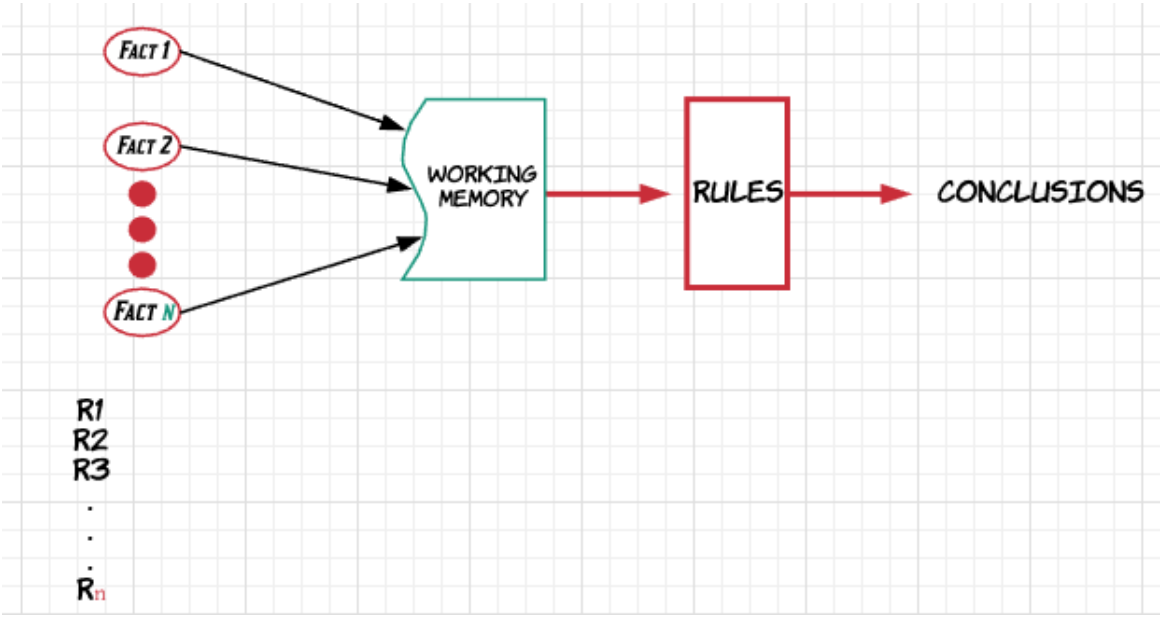
Looking for set of rules and check to see which sets of premise in rule base are logical true

What would this imply?

There are some facts supports the conditions of rule "support mean making them true" then if the condition of rules is met "true" then draw conclusion. When draw that conclusion, essential adding to working memory "fire or activity"

What working memory! ⇢ *It has fact and conclusion*

Now, having more knowledge and premises and keep going until no rule draw to conclusion. In the end of the processes whatever rule tells, this is what expert system recommend



There is control strategy involve with this and will exam rule in this order and if there is more than one to apply, will take the rule will low index.(just for the discussion in the class)

Working Memory (WM)

< Green >=5 Ibs >

Forward chaining

< Green >=5 Ibs, produce, perishable, turkey, watermelon >

Triger (condition is met)	*Fire (add conclusion to working memory)*
R1	*R1*
R1, R3, R6	*R3*
R1, R3, R5, R6	*R5*
R1, R3, R5, R6	*R6*

Note

- *Repeating rules ignor*
- *If having more than one rule that can apply then choose the lower index.*

Backward chaining

Assume, walking in the store and see this object

Question

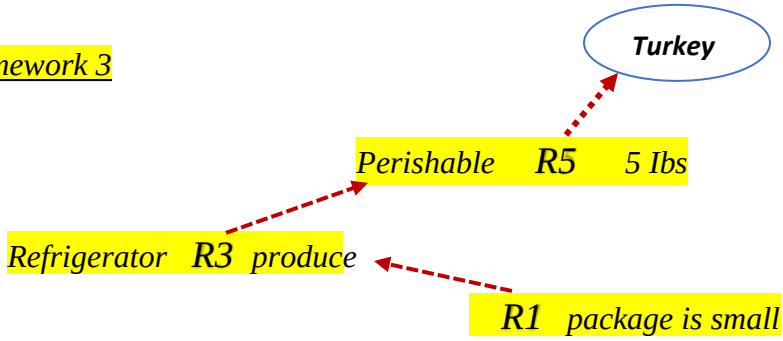
- *Can conform this hypothesis knowing the fact?*

What do have to know in order to conclude this is turkey which rule port this conclusion!

Scan the rules and check to see which the rule support this hypothesis.

In the backward changing scanning the conclusion of the rules and check to see which rules support this

Discussion of Homework 3



Homework #3
Due: March 30, 2018 (11:59pm) Please note: no late submission will be accepted.

Consider a sliding block puzzle with the following initial configuration:

B	B	B	W	W	W	E

There are three black tiles (B), three white tiles (W), and an empty cell (E). The puzzle has the following moves:

a. A tile may move into an adjacent empty cell with unit cost.
b. A tile may hop over at most two other tiles into an empty cell with a cost equal to the number of tiles hopped over.

Initial	Next	Cost
BBBWWEW	BBBW**EW**W	1
BBBWWEW	BBB**E**W**W**W	1
BBBWWEW	BB**E**WW**B**W	2

The goal of the puzzle is to have all of the white tiles to the left of all of the black tiles (regardless of the position of the blank cell).

You are to:

1. Specify a heuristic function, h, for this problem and show the search tree produced by the algorithm A* using this heuristic function (show the explicit portion of the search tree).
2. Determine and show that your function never overestimates the true cost of reaching a goal or otherwise.
3. Implement a program in C, C++ or Java to solve the sliding block puzzle using your heuristic function.

Expert System

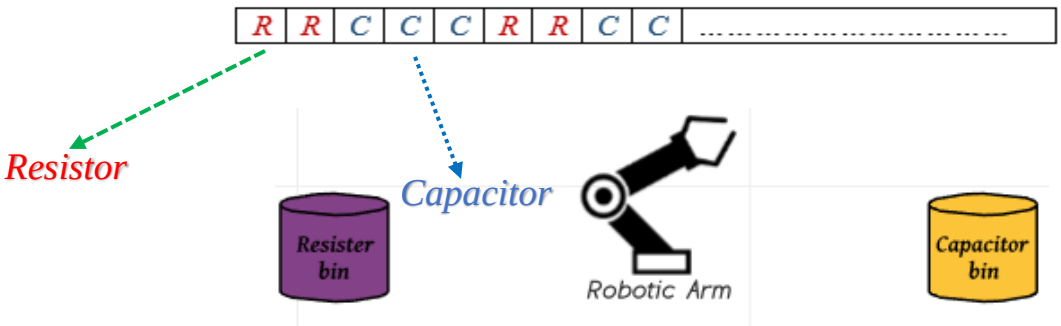
It uses two context

1. *confirm hypothesis* *(Hypothesis checking)*
2. *Assist in configuration* *(Configuration)*

In pure engineering context configuration can be intuitively appealing

Let explaining what this is by example

Let suppose having array and want to sort some part by using robotic arm and having two bins one is resistor bin and another is capacitor bin. The robotic arm picks up the resistor and capacitor and sort them in resister bin and capacitor bin.



Let assume the robot has a sensor and camera that can recognize resistor and capacitor

Writing set of rules (production rules) and writing a little expert system that would do it in (forward chain fashion)

Forward chain means match the state working memory fact with precondition and execute the action which is the conclusion of the rule.

Rule Base

- *R1 if there is a resistor then pick it up and place in "R" bin*
- *R2 if there is a resistor and another resistor next to it then pick up both and place them in "R" bin*
- *R3 if there is a capacitor then pick it up and place in "C" bin*
- *R4 if there is a capacitor and another capacitor next to it then pick up both and place them in "C" bin*
- *R5 if there is no resistor and no capacitor left then stop*

Conflict resolution because with one robotic arm, it can do one thing at time. In this cause if having more than one rule trigger, having something is called "Conflict resolution "

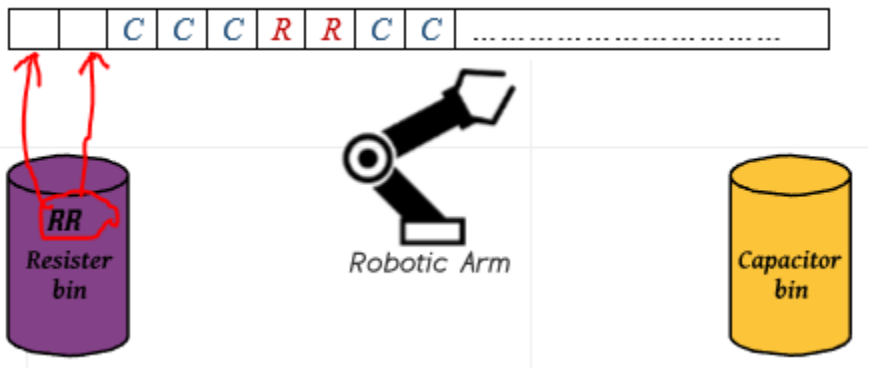
Conflict Resolution "if more than one rule trigger which one to fire"

Suggestion the following

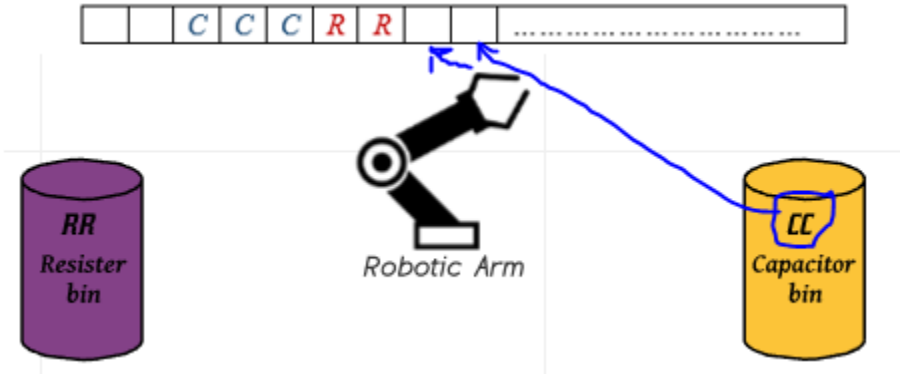
What are they?

1. *Order index of the rule*
2. *Specificity if the set of the precondition of rule Rx is subset of set precondition of rule Ry execute the rule that is superset of precondition. " in other word do something more specific"*
3. *context switching explicit rules " if there is no more resistor left switch and pick up capacitor only"*

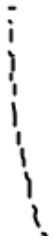
In the expert system using this technique to decide which rule to pick up



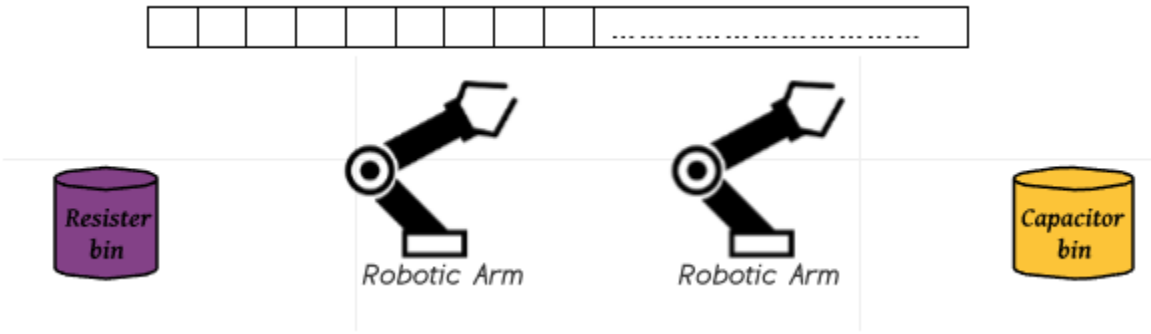
Scanning the rule again



And repeating until no resistor and capacitor left.

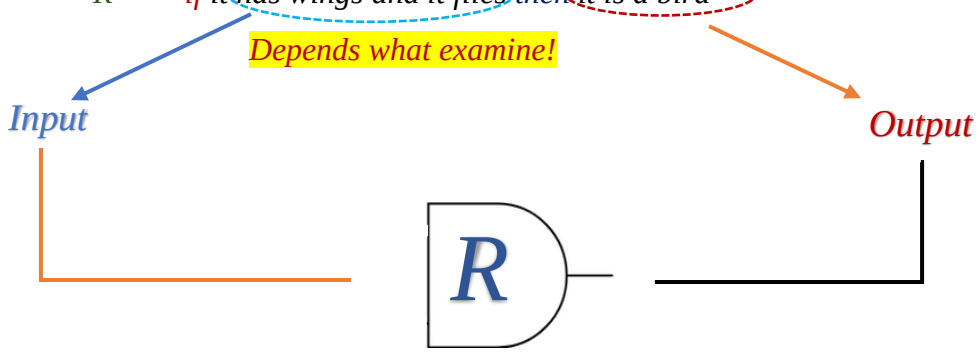


What if have two robotic arms



Imprecise Reasoning

- *R* *if it has wings and it flies then it is a bird*



Looking to this problem from 3 angles

1. *How much confident have in the input to reasoning rule*
   - *The confident level determine by confident factor "CF" and measure between 0 and 1*
   - *It is a degree of confident or trust or measure of believe in something*
   - *Reliability of rule chain of reasoning how rational to*

- *R1* *if it has wings, feather and it flies then it is a bird*

.99 confident

.89 Rational

Level of confident

Confident level to input

Confident level in how rational this particular rule is

- *R2* *if it has wings, feather and it swims then it is a bird*

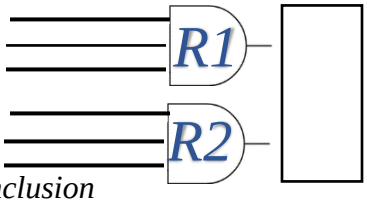
➢ Imagine this chain of reasoning where having two rule whose conclusion are the same
It is called "multiple argue evidence"

Question

If know the reliability of input and the reliability of

Output then can compute the reliability of output

Then can compute the reliability of multiple argue conclusion



Quiz topics

1. *AND /OR graph*
2. *Alpha Beta prune*
3. *Expert system (until lecture 20)*

Understand

- *what graph is*
- *what use for*
- *how to compute the cost*

Imprecise Reasoning

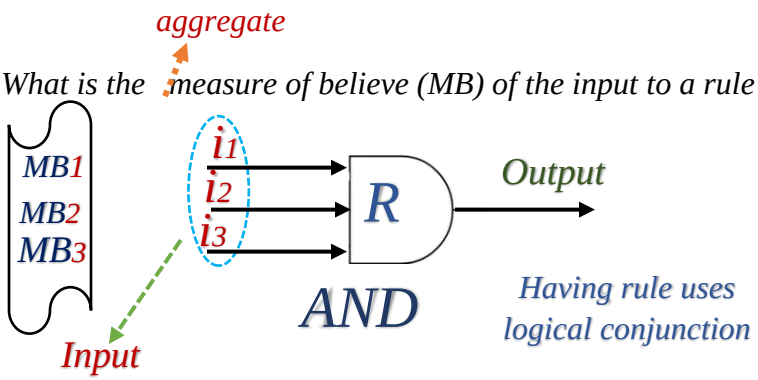
- *How to compute none probabilistically confidence level that certain searching using expert system.*
- *For giving more background posting work Stanford university using the concept to write one of the first expert system to help physicians prescribe particular type of antibiotic*

Confidence Factor (CF) or Measure Of Believe (MB)

- *Constrain* $0 \leq MB \leq 1$

Posing three questions

1. *What is the measure of believe (MB) of the input to a rule?*



What is the aggregate measure of believe?

Formal $MB(i_1 \& i_2 \& \ldots\ldots \& i_n) = Min\{MB_1, MB_2 \ldots\ldots\ldots MB_n\}$

- *Taking the aggregate to be the minimum the measure of confidence in the input*
- *Why taking the Min!*

- *R1* *if* 0.95 MB *X drives a Volvo* *AND* 0.45 MB *X reads the New York Times (NYT)* *then* *X will vote Democrat*

What is the aggregate input?

- *Min {0.95 , 0.45} = 0.45*

Why taking the Min!

Question 2 is

2. *1.a if having to compute the measure of believe of*

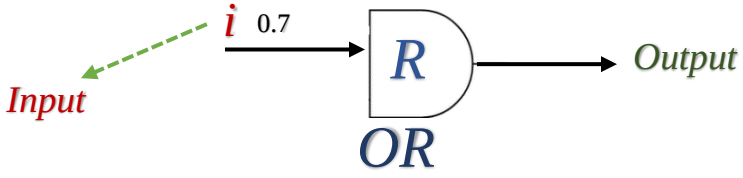
$$MB\ (\ i_1\ OR\ i_2\ OR\ \ldots\ldots OR\ i_n\ ) = Max\ \{\ MB_1\ ,\ MB_2\ldots\ldots\ldots\ MB_n\ \}$$

- *R1* *if* 0.6 MB *X does not like real estate developers*

OR

0.7 MB

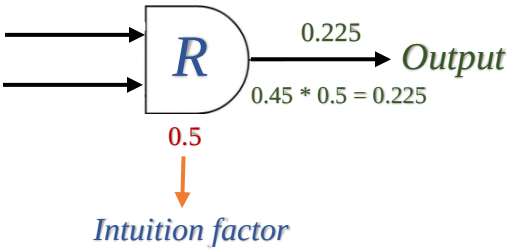
If *X likes* National Public Radio *(NPR)* *then* *X will vote Democrat*



- *Max {0.7 , 0.6} = 0.6*

- *What is the measure of believe MB in the rule itself?*

Taking confidence input which is 0.45



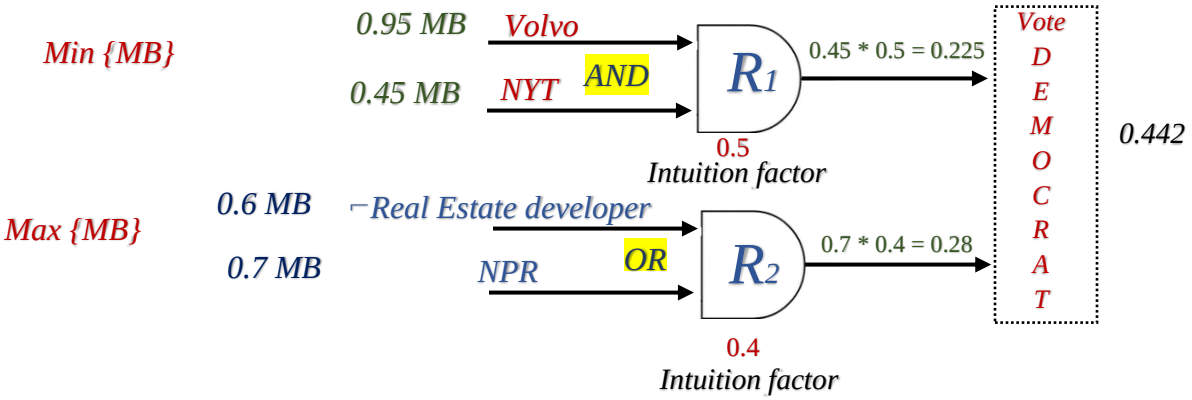
The confidence in the output is half of the confidence of input

*Output = Min (MB) * intuition factor*

Question 3 is

How to combine this?

- *Multiple facts supporting the same conclusion what is my measure of believe in that conclusion?*



*Output = Min (MB) * intuition factor*

*Output = Max (MB) * intuition factor*

What is measure of believe MB (Democrat, R1 & R2)

Referring to system is called MYCIN

MB (H, R1&R2) = MB (R1) + MB (R2) * (1 – MB (R1))

Hypothesis

*0.225 + 0.28 * (1 – 0.225) = 0.442*

- Different rules or independent rules supporting same conclusion.

Introduction to predicate logic

- Logic as representation method
  - What is representation know
  - *How to describe UML!*
    - Structured representation method because it is lean object class and physical entities and it has hierarchy class, subclass and decomposition.

Logic is called unstructured representation

- *What do we know about logic?*
- *Vocabulary*
  - Each time using capital letters A , B ,….K → label for object
  - Lower case x logic variable

<X, A> means X ← A

Logic world	*Real world*
A----------	----------A
B----------	----------B
x----------	---------
X← A	

- Predicates $O_n(A, B)$ → *True* / *OR* / *False*
  - *Predicate is expression whose value is true or false*
  - Dealing with binary logic not fuzzy logic

- *Function* $y=O_n(x)$

  To express the relationship between object

- *Basic operators*
  - & , ^ *AND* $A \wedge B$ *Conjunction*
  - *V OR* $A \vee B$ *Disjunction*

$A \Rightarrow B$

A	B	$A \Rightarrow B$	$\neg A$	$\neg A \vee B$
T	T	T	F	T
T	F	F	F	F
F	T	T	T	T
F	T	T	T	T

  - ¬ , ~ *negation* $\neg A$ *It is true or false depends on A*
  - ⟹ *implication* $A \Rightarrow B \overset{=}{=} \neg A \vee B$ (= *equal*)