Raven Notes 2

CSCI 321

Based on *Programming Game AI by Example*, Buckland

May 29, 2015

Planning

- Involves more than one step at a time.
- *Simple soccer* planning was absorbed in the logic:
    - "Move to support position"
        - not a goal in itself
        - part of an underlying plan
- *Raven* requires plans:
    - None of these is a simple action by itself:
        - Get health
        - Get weapon
        - Attack target

Hierarchical planning

- Buy sword
  - Get gold
    - Plan path to goldmine
    - Follow path
      - Follow edge #1
      - Follow edge #2
      - Follow edge #3
    - Pick up nugget
  - Go to smithy

Use an abstract class that can be either a simple or a composite goal.

Raven Goal Interface

Similar to State interface

- Activate
  - can be called more than once to replan
- Process. Returns one of
  - inactive
  - active
  - completed
  - failed
- Terminate
- HandleMessage
- AddSubgoal

Raven Goals

Composite goals	Atomic Goals
Goal_Think	Goal_Wander
Goal_GetItem	Goal_SeekToPosition
Goal_MoveToPosition	Goal_TraverseEdge
Goal_FollowPath	Goal_DodgeSideToSide
Goal_AttackTarget	
Goal_Explore	
Goal_HuntTarget	

Atomic Steering Behavior Goals

- Goal_Wander
- Goal_SeekToPosition
- Goal_TraverseEdge
  - upon activation checks for special edge behavior:
    - open door
    - swim
    - *etc.*
  - uses **seek** for most edges, **arrive** for last edge
  - monitors to check for stuck bot

Goal_FollowPath

- Iterate through edges.
- Edge type determines subgoal:
  - Goal_TraverseEdge
  - Goal_NegotiateDoor
  - Goal_Jump
  - Goal_Swim
  - *etc.*

Goal_MoveToPosition

- Activate:
  - RequestPathToTarget (sent to Path Planner)
  - SeekToPosition
- HandleMessage:
  - FollowPath (received from Path Planner)

Goal_AttackTarget

- If target gone:
    - Add subgoal: hunt target
- If target is shootable:
    - If room:
        - Add subgoal: dodge side to side
    - else:
        - Add subgoal: seek to target position
- Alternatives:
    - move to best range for best weapon
    - move to sniping position
- Weapon system is completely separate
    - constantly selects best weapon, aims, shoots
    - regardless of other goals

Goal_Think

- Top level goal
- Decides between:
    - Explore
        - pick a random point and follow path there
    - Get Health
    - Get Weapon
        - Rocket Launcher
        - Shotgun
        - Railgun
    - Attack Target
- Uses four feature functions (all scaled 0-1):
    - Health
    - Distance to item
    - Individual weapon strength
    - Total weapon strength

Calculate Desirability and Choose Best

$$\begin{aligned}
Desirability(Health) &= k\left(\frac{1 - Health}{DistToHealth}\right) \\
Desirability(Weapon) &= k\left(\frac{Health(1 - WeaponStrength)}{DistToWeapon}\right) \\
Desirability_2(Weapon) &= k\left(\frac{Health(1 - WeaponStrength)}{DistToWeapon^2}\right) \\
Desirability(Attack) &= k(TotalWeaponStrength)(Health) \\
Desirability(Explore) &= 0.05
\end{aligned}$$

Using Empathy

- You observe a player low on health break off a battle and run
- You run your Goal_Think algorithm on the player's data
- The player's best option is to find health
- You plan a path to the health to intercept the player

Personalities

- Desirability scores can be weighted.
- Conservative player weighs health and weapons heavier than attack
- Aggressive player weighs attack heavier
- In a full RTS game you could:
  - Create opponent that favors exploration and research
  - Create opponent that favors massive armies quickly
  - Create opponent that favors city defenses

State Memory

- Use a goal stack to resume interrupted goals.
- Goal FollowPath could be interrupted by DefendAgainstAttacker and then resumed.
- Goal FollowPath interrupted by NegotiateDoor and then resumed.

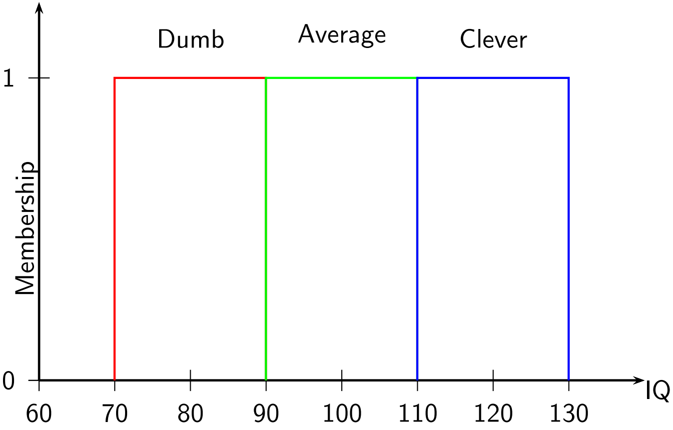
Command Queuing

- Used in modern RTS games
- Can click many waypoints, Bot navigates to each in turn
- Can establish patrols by making waypoints into a loop
- Can queue multiple commands of any sort:
  - Build a barracks *and then*
  - Move to this spot *and then*
  - Build a turret
- Only change needed is adding subgoals to the back of the queue instead of the front.

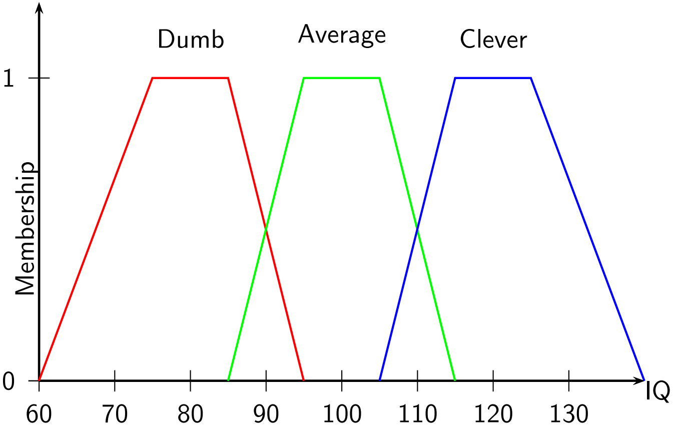
Fuzzy Logic

- A more intuitive alternative to mathematical formulas for making decisions.
- Examples of fuzzy quantities:
  - A large piece of pie
  - A fairly strong wind
  - Low health
  - Hit the ball very hard
  - Far away

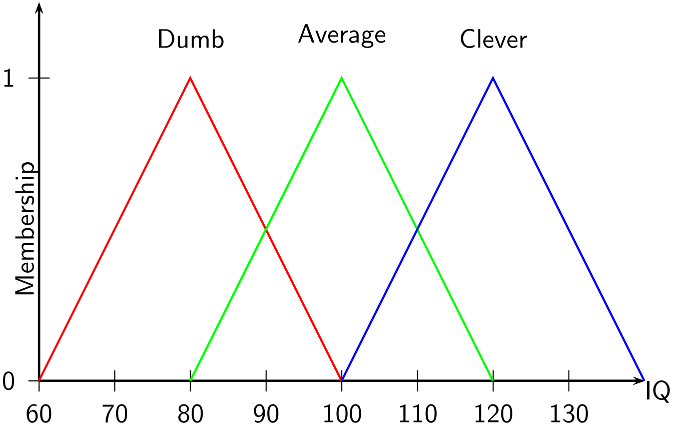
Crisp sets


Dumb
Average
Clever
1
0
Membership
IQ
60
70
80
90
100
110
120
130

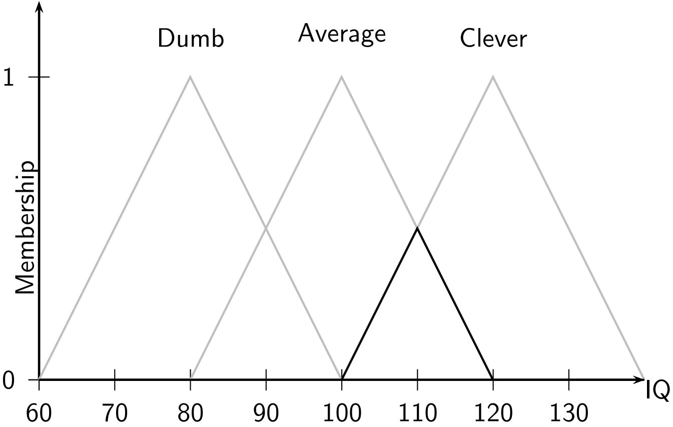
Fuzzy sets


Dumb
Average
Clever
1
0
Membership
60
70
80
90
100
110
120
130
IQ

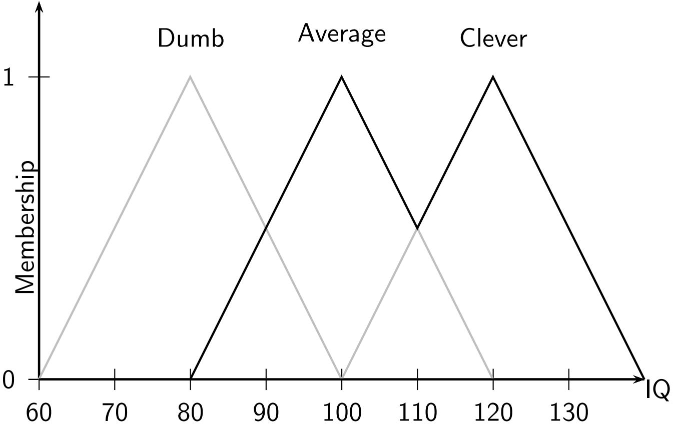
Fuzzy sets


Dumb
Average
Clever
1
0
Membership
IQ
60
70
80
90
100
110
120
130

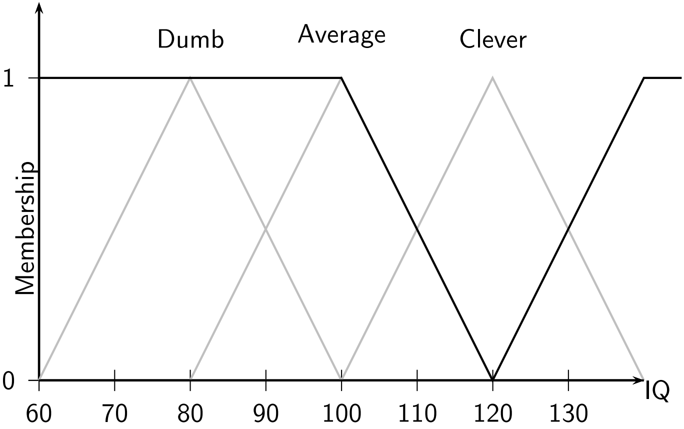
Average AND Clever



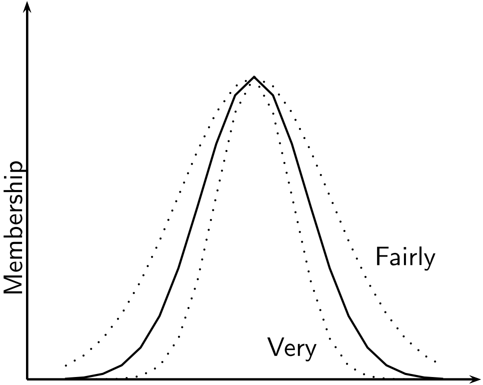
Average OR Clever



NOT Clever

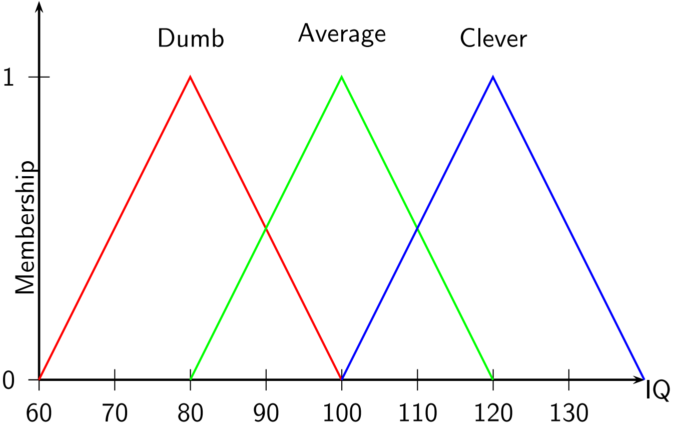

Dumb
Average
Clever
1
0
Membership
60
70
80
90
100
110
120
130
IQ

Hedges



Fairly = square root
Very = square

Fuzzy Linguistic Variable

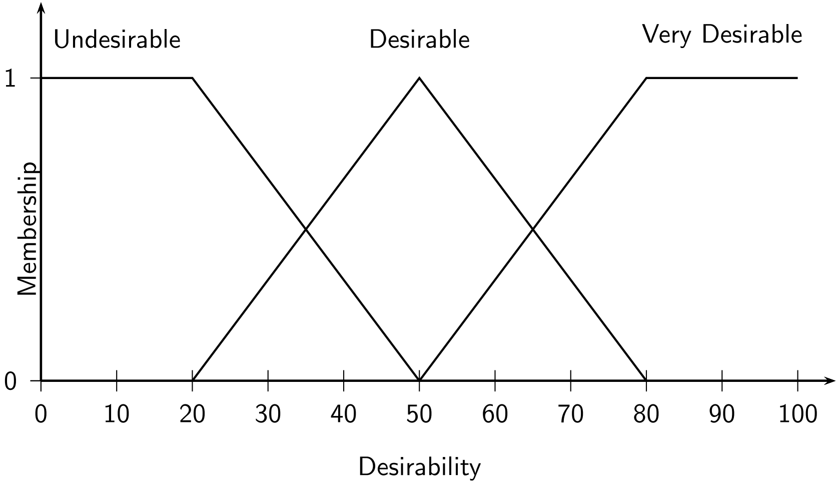


IQ = {Dumb, Average, Clever}
A set of fuzzy sets.

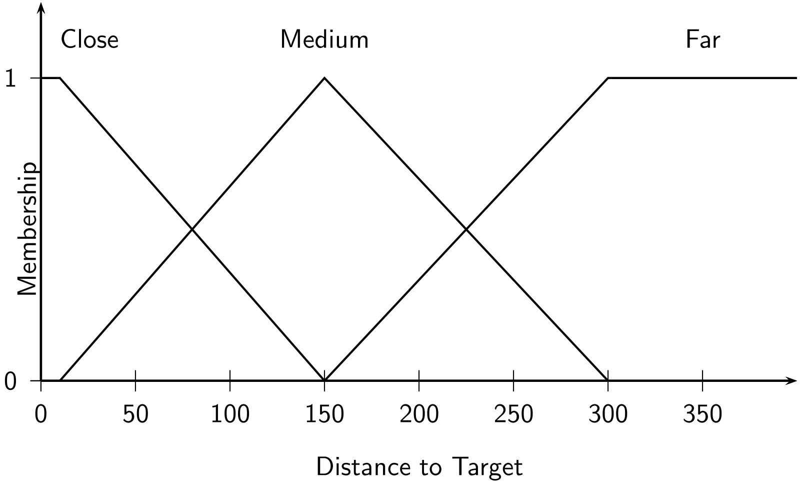
Reasoning with FLV's

- Define FLV's for all quantities of interest:
  - Distance: close, medium, far
  - Ammo: low, okay, loads
  - Desirability: undesirable, desirable, very desirable
- Define fuzzy rules (*e.g* desirability of using rocket launcher):
  - If target is far and ammo is loads then desirable
  - If target is close then undesirable
  - *etc.*
- Start with crisp values (measurements) of distance and ammo
- Fuzzify crisp values to fuzzy values
- Reason with fuzzy values
- Come to fuzzy conclusion
- Defuzzify conclusion

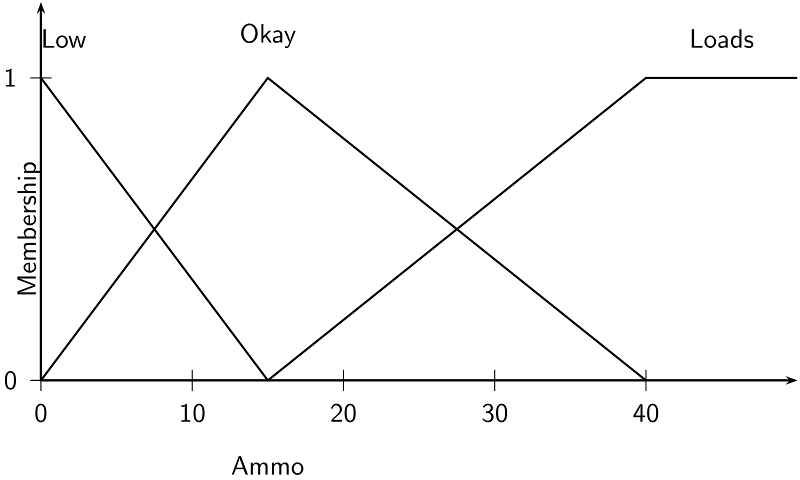
Fuzzy Desirability



Fuzzy Distance to Target



Fuzzy Ammo Status



Fuzzy Rules for Desirability of Rocket Launcher

1. If target is far and ammo is loads then desirable
2. If target is far and ammo is okay then undesirable
3. If target is far and ammo is low then undesirable
4. If target is medium and ammo is loads then very desirable
5. If target is medium and ammo is okay then very desirable
6. If target is medium and ammo is low then desirable
7. If target is close and ammo is loads then undesirable
8. If target is close and ammo is okay then undesirable
9. If target is close and ammo is low then undesirable

Fuzzy Inference

1. For each rule,
    1.1 For each antecedant, calculate the degree of membership of the input data
    1.2 Calculate the rule's inferred conclusion based on these values
2. Combine all the inferred conclusions into a single fuzzy conclusion
3. Defuzzify the conclusion

Example

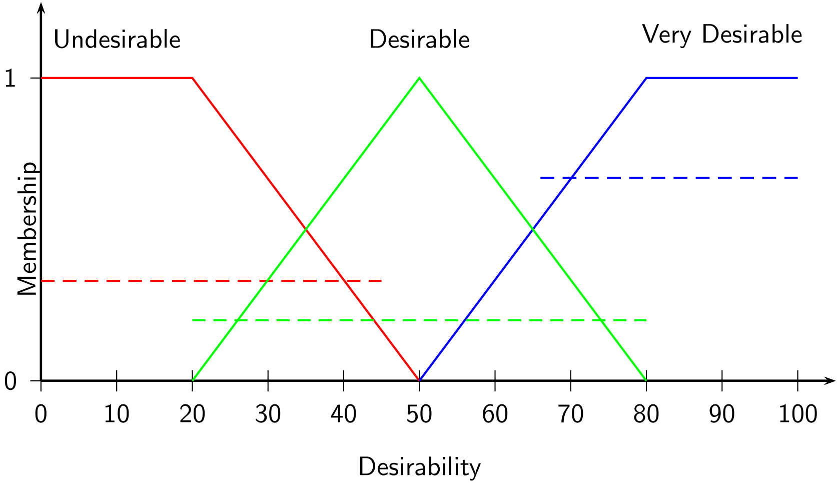
- Suppose target is at distance 200 and ammo is 8 rockets remaining.
- **Rule one:** If target is far and ammo is loads then desirable
  - Distance 200 means target far has DOM 0.33
  - Ammo 8 means ammo loads has DOM 0.0
  - ANDing these together means desirable is 0.0
- **Rule two:** If target is far and ammo is okay then undesirable
  - Distance 200 means target far has DOM 0.33
  - Ammo 8 means ammo okay has DOM 0.78
  - ANDing these together means undesirable is 0.33
- **Rule three:** If target is far and ammo is low then undesirable
  - Distance 200 means target far has DOM 0.33
  - Ammo 8 means ammo low has DOM 0.2
  - ANDing these together means undesirable is 0.2

FAM: Fuzzy Associative Matrix

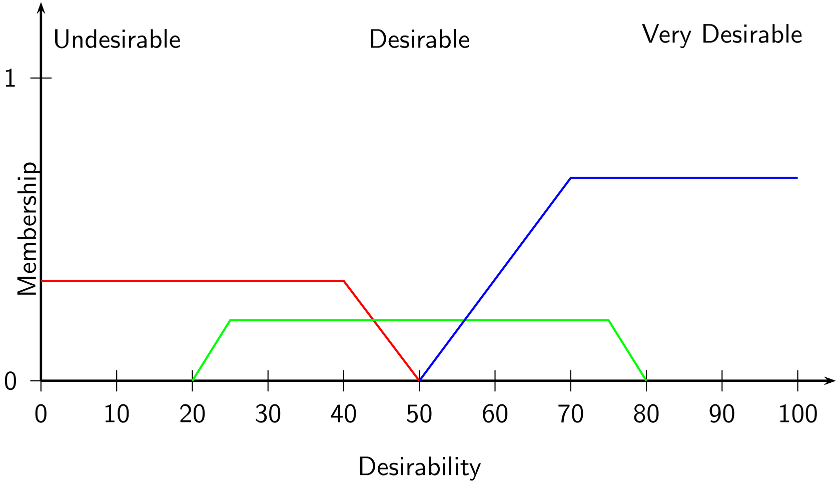
	Target close	Target medium	Target far
Ammo low	Undesirable 0	Desirable 0.2	Undesirable 0.2
Ammo okay	Undesirable 0	VeryDesirable 0.67	Undesirable 0.33
Ammo loads	Undesirable 0	VeryDesirable 0	Desirable 0

Consequent	Confidence
Undesirable	0.33
Desirable	0.2
VeryDesirable	0.67

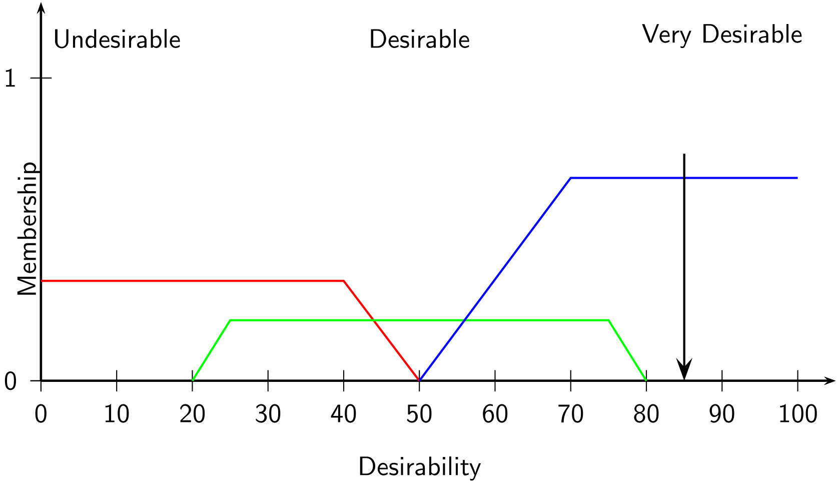
Fuzzy Conclusion: Truncate each set



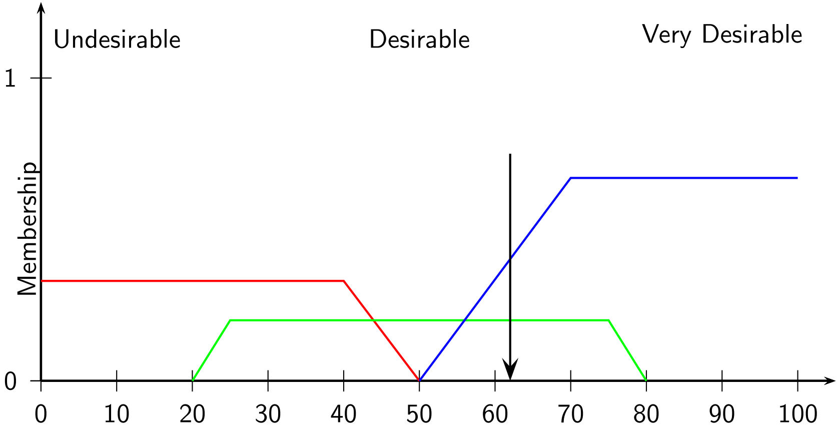
Fuzzy Conclusion: Truncate each set



Defuzzify: Mean of Maximum

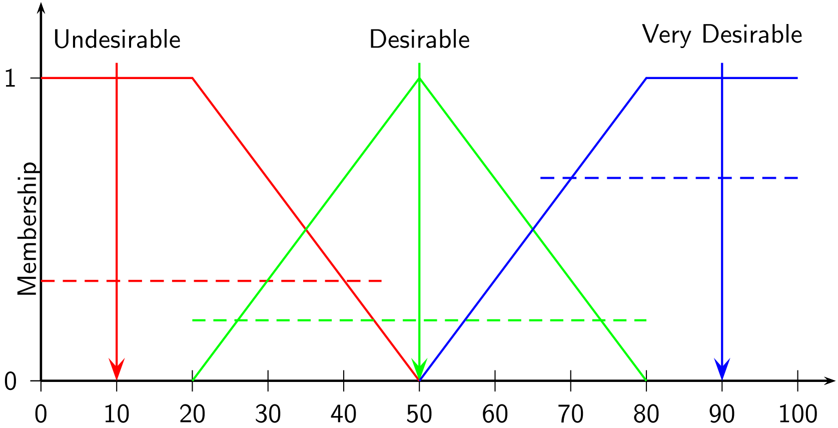


Defuzzify: Centroid



$$\sum_i \frac{i(membership_i)}{\sum_i membership_i}$$

Defuzzify: Average of Maxima



$$\frac{\sum_i (value_i)(confidence_i)}{(\sum_i confidence_i)}$$

Combinatorial explosion of rules

If A_i and B_i and C_i and D_i then Consequent

If each FLV has 5 possible values, then we have $5^4 = 625$ rules.

If we had 6 LFV's instead of 4, there would be $5^6 = 15625$ rules.

Rewrite rules for ease of computation

If A_i and B_i and C_i and D_i then Consequent

is equivalent to

If A_i then Consequent
OR
If B_i then Consequent
OR
If C_i then Consequent
OR
If D_i then Consequent

Using equivalences like this we can usually greatly reduce the number of rules. For example, the three rules for desirability of using the rocket launcher when the target is close can all be reduced to one:

If target is close then undesirable

This simplification can be done automatically.