CS354R

DR SARAH ABRAHAM

AI BEHAVIOR TREES AND FUZZY LOGIC

FSMS AND DECISION TREES

- FSMs become unwieldy in larger, more complex cases
  - Need a simpler, more understandable solution to create compelling, believable AI
- Decision trees become difficult to construct in larger, more complex cases
  - Learned decision trees can help but learning is unpredictable
- Is it possible to combine the expressiveness and nuance of an FSM with a tree structure?

BEHAVIOR TREES

- Widely adopted AI system for NPCs in scenarios that work for FSMs and decision trees
- Tree structure allows for more efficient debugging and engineering
- Stateful representation allows for a range of emergent behaviors

BEHAVIOR TREE STRUCTURE

- Tree nodes represent behaviors
- Nodes can have children that represent subtasks
  - Children assigned priority
- Nodes evaluated from root
- At each level, node given a priority based on conditions
- Selected behavior node runs any subtasks
- On next tree evaluation, starts at highest priority node

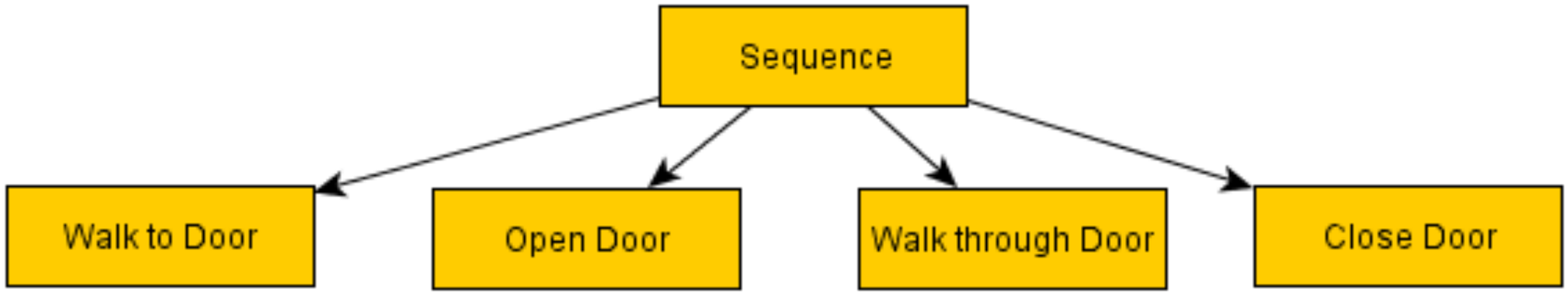
BEHAVIOR TREES IN ACTION

1. Traversal starts from root on start
2. Children evaluated in order of priority
3. If all conditions are met:
    1. Node set to “Running”
    2. Node runs its subtasks
4. If conditions not met:
    1. Node set to “Failed”
5. Upon completion of task:
    1. Node set to “Completed”
6. Upon completion of all tasks:
    1. Tree evaluation restarted

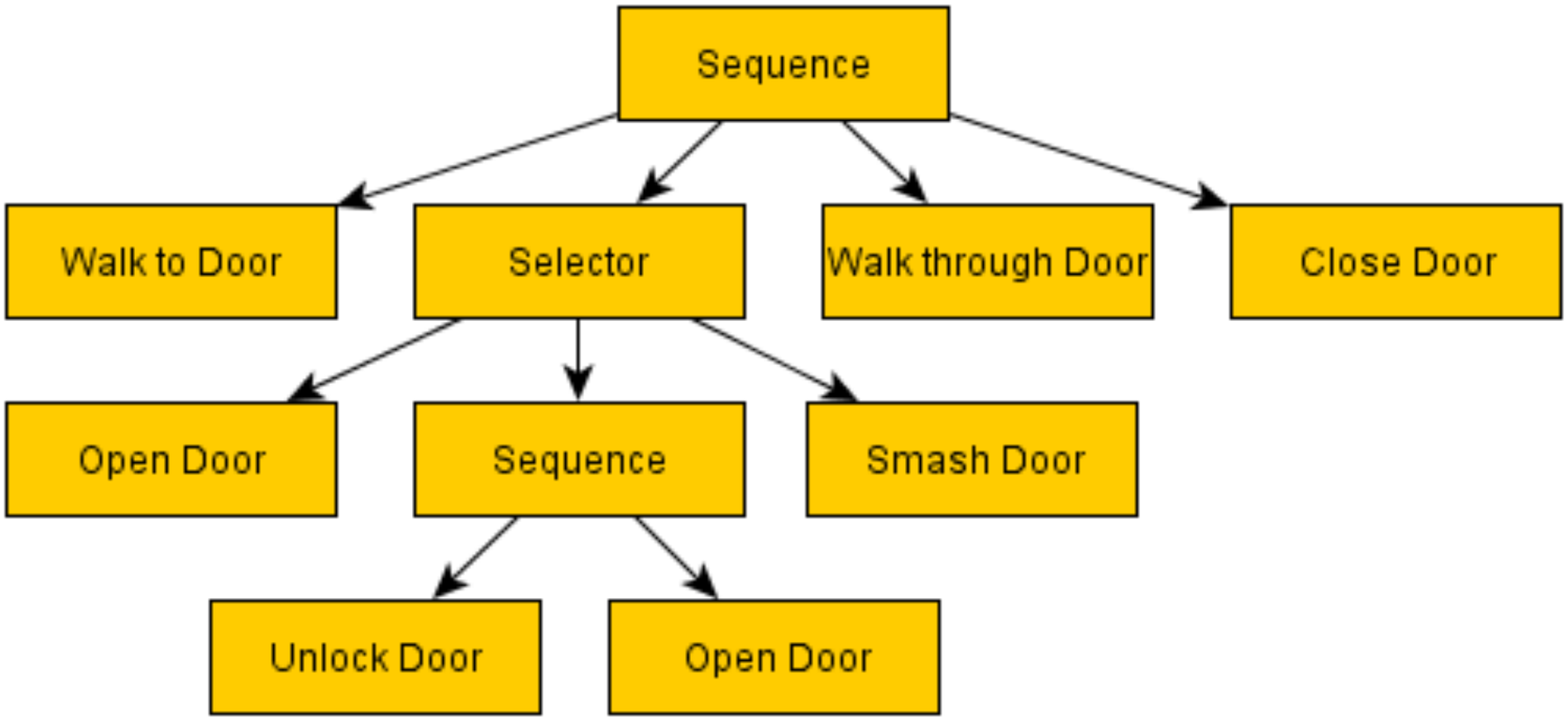
SEQUENCES AND SELECTORS

- Nodes used to organize behaviors
- Sequence nodes will find and execute child nodes that are not “complete” in order
  - Children will fail or run sequentially
  - If a child fails, the parent sequence will fail
- Selector nodes are like sequence nodes but will continue to check child nodes if previous children have failed
  - Parent selector succeeds if any child succeeds

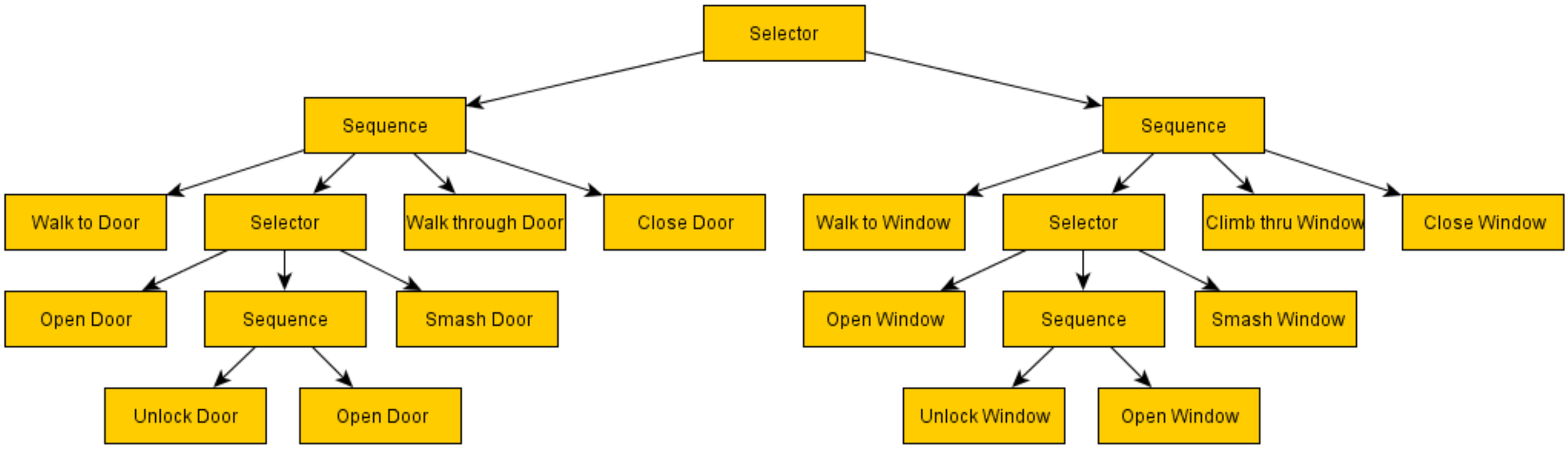
BEHAVIOR TREE EXAMPLE



BEHAVIOR TREE EXAMPLE

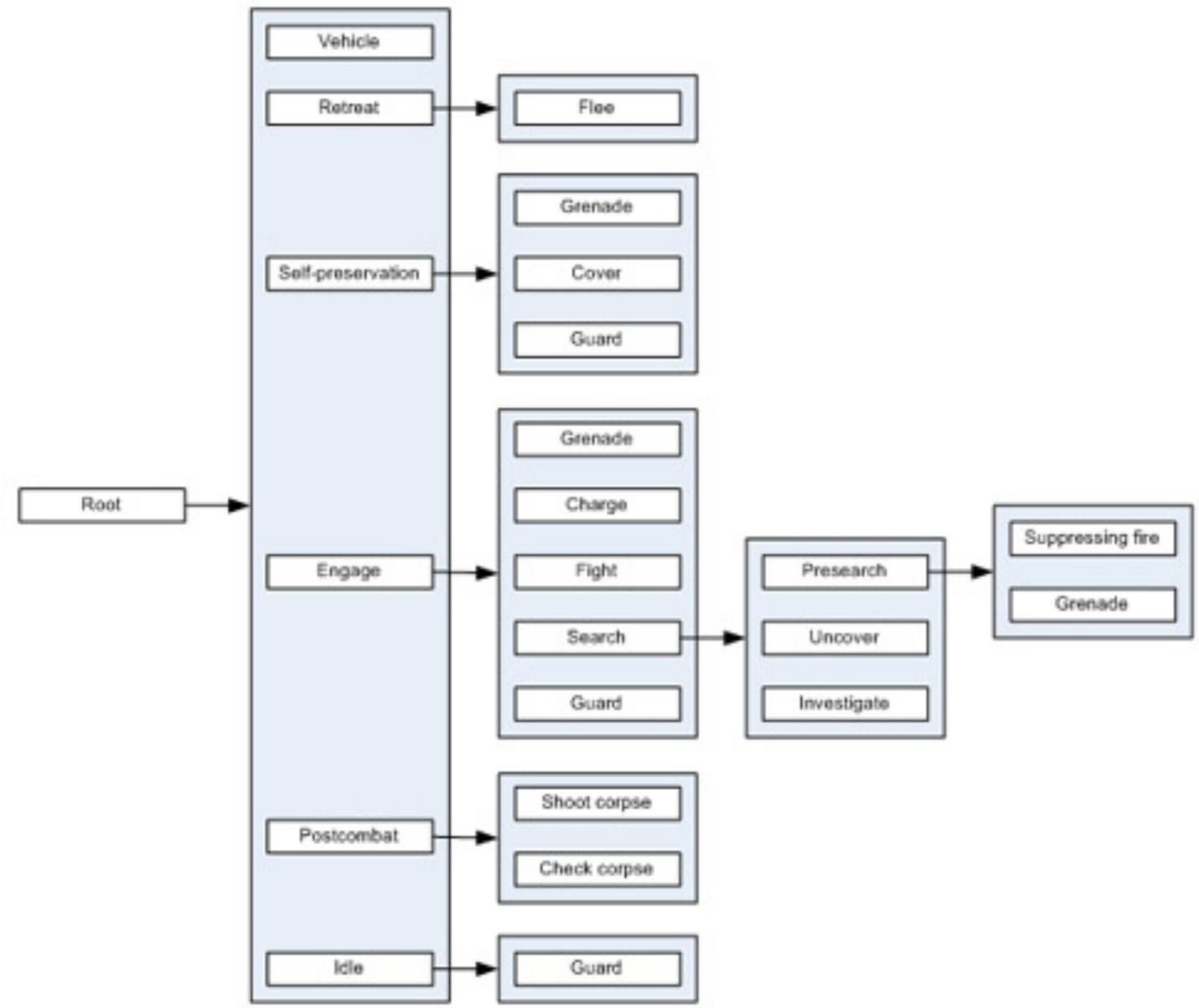


BEHAVIOR TREE EXAMPLE



BEHAVIOR TREE EXAMPLE: HALO

- Halo 2 first game to discuss use of behavior trees at GDC 2005



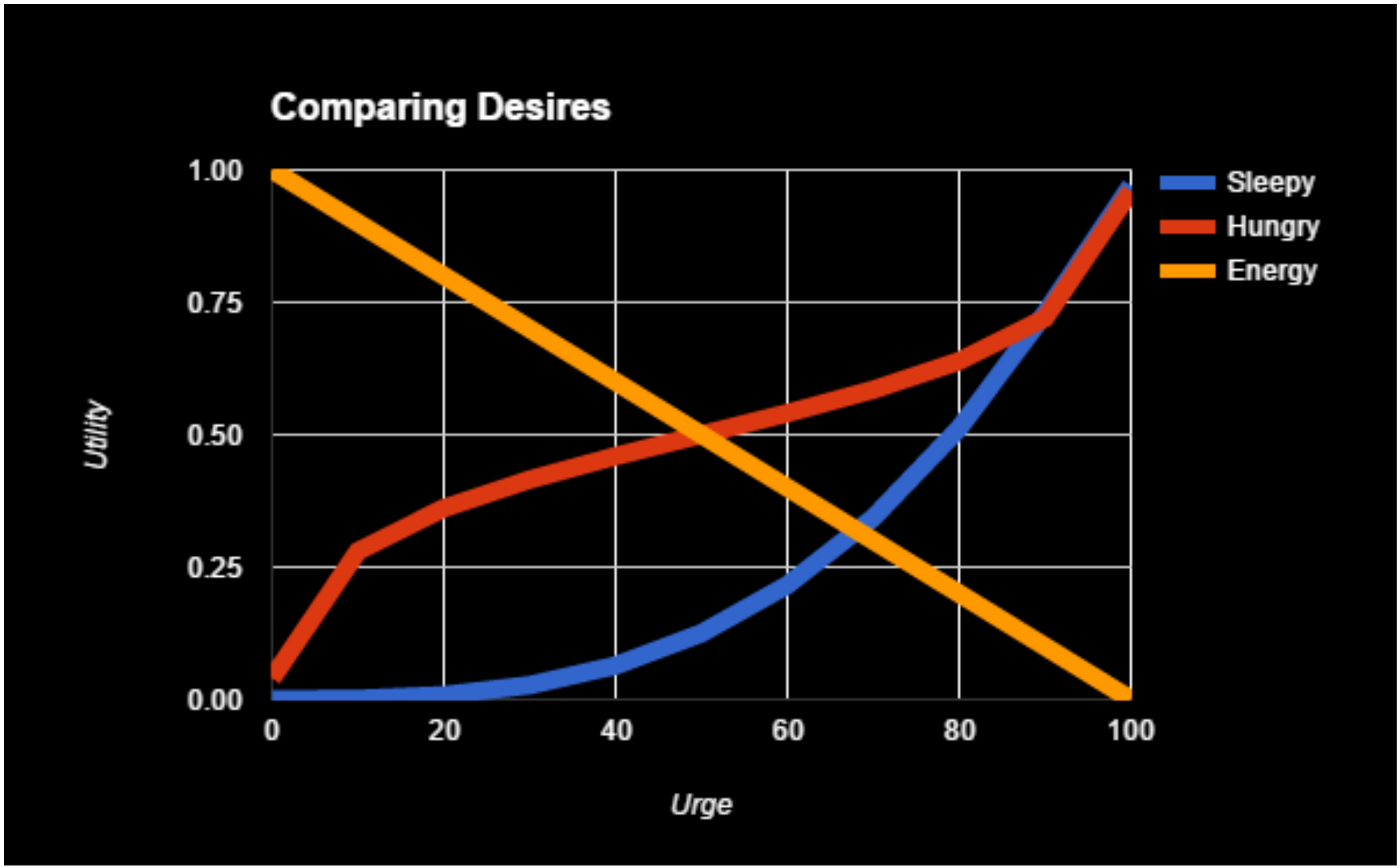
BEHAVIOR TREES PROPERTIES

- Behavior persistent across multiple AI loops
- Uses ordering of conditions and responses to emulate human reason
- Allows complex behavior with simpler designer control than a FSM
- Idea extensible to more complex systems such as “Utility Trees” (what Killzone 2 and F.E.A.R use)

UTILITY THEORY

- Realistic and varied AI is too complex for highly designed systems like FSMs and behavior trees
- Utility Theory provides simpler way to design and extend AI
  - Uses a “utility” score to determine “best” action
    - Score based on value of action (desire) and probability of achieving it
    - Model game values along a function to determine score
  - Provides “fuzzy logic” like solution with more potential for emergent behavior
    - AI is making a “best guess”

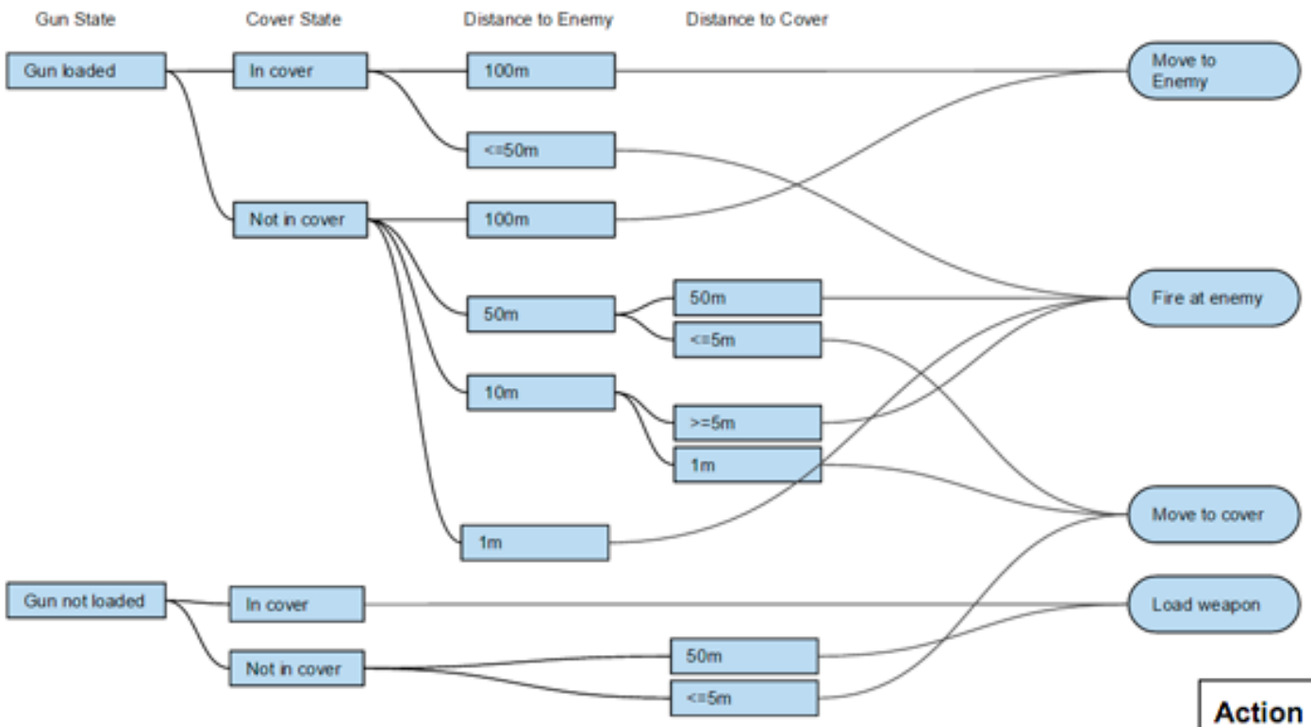
APPLYING UTILITIES



UTILITY AI IN GAMES

- Sims
- Killzone 2
- F.E.A.R
- Civilization
- Quake (modern)
- etc…

BEHAVIOR TREE VS UTILITY



Action	Scorer	Score
Move to Enemy	Distance to Enemy	0-100
	Gun is not loaded	-100
Fire at Enemy	Proximity to Enemy < 50	75
	Cannot make it to cover	50
	Gun is not loaded	-125
Move to Cover	Is not in cover	50
	Proximity to Cover < 50	50
Load	Gun is not loaded	75
	Is in cover	50
	Gun is loaded	-125

HOW TO POPULATE TREES?

- Data-driven or code-driven
  - Determines whether behaviors encoding into actual data structures or built out of imported data
- Useful tools in both pipelines: standard file format (i.e. XML), custom data parser, standard scripting language (e.g. Lua or Python)

FUZZY LOGIC

- Philosophical approach
  - Decisions based on “degree of truth”
  - Is not a method for reasoning under uncertainty – that’s probability
- Crisp Facts – distinct boundaries
- Fuzzy Facts – imprecise boundaries
- Probability - incomplete facts
- Example – Scout reporting an enemy
  - “Two tanks at grid NV 54“ (Crisp)
  - “A few tanks at grid NV 54“ (Fuzzy)
  - “There might be 2 tanks at grid NV 54 (Probabilistic)

APPLY TO COMPUTER GAMES

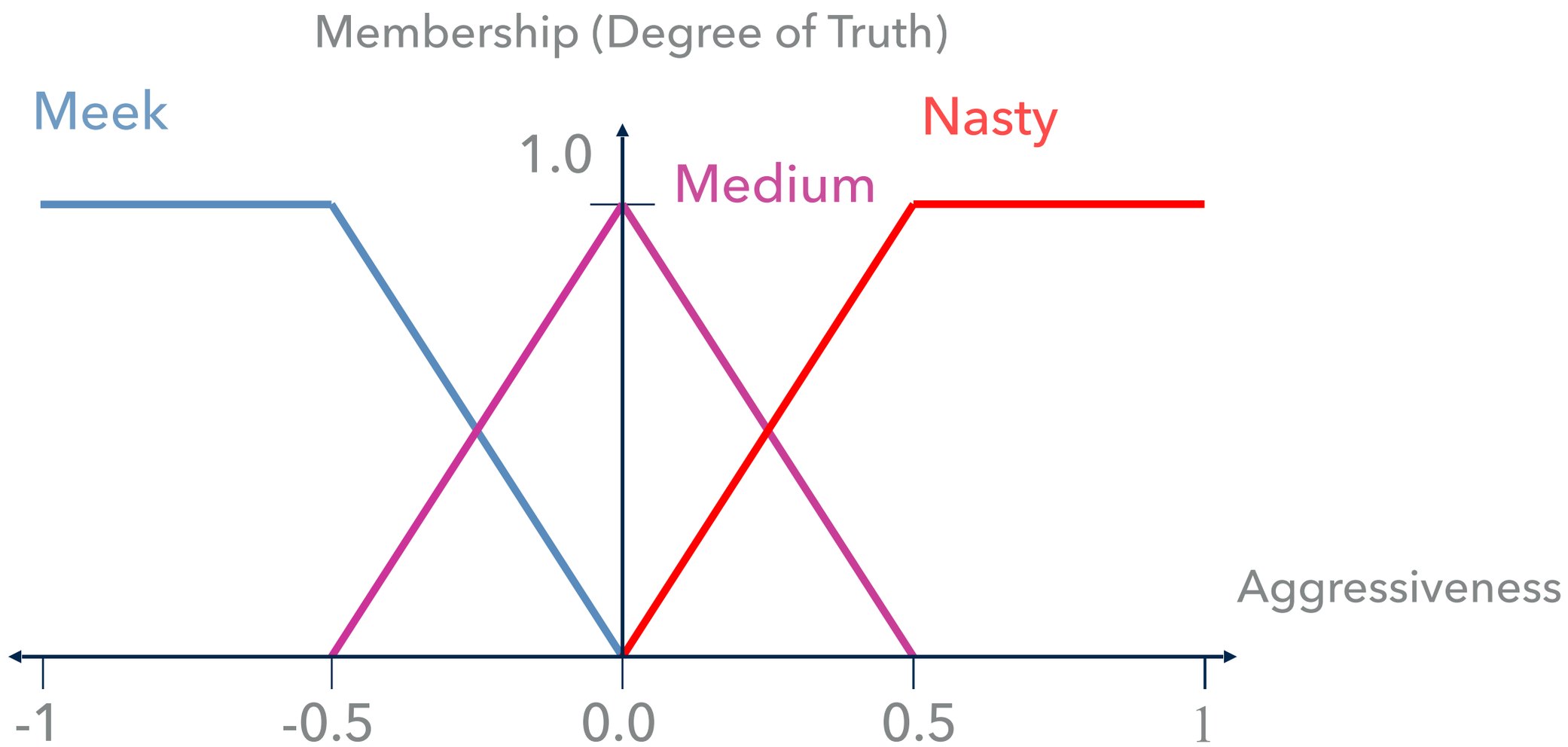
- Can have different characteristics of players
  - Strength: strong, medium, weak
  - Aggressiveness: meek, medium, nasty
    - If meek and attacked, run away fast
    - If medium and attacked, run away slowly
    - If nasty and strong and attacked, attack back
- Control of a vehicle
  - Should slow down when close to car in front
  - Should speed up when far behind car in front
- Provides smoother transitions – not a sharp boundary

FUZZY SETS

- Provides a way to write symbolic rules with terms like “medium” but evaluate them in a quantified way
- Classical set theory: An object is either in or not in the set
- Fuzzy sets have a smooth boundary
  - Not completely in or out – somebody 6ft is “80% in the tall set” tall
- Fuzzy set theory
  - An object is in a set by matter of degree
  - 1.0 => in the set
  - 0.0 => not in the set
  - 0.0 < object < 1.0 => partially in the set

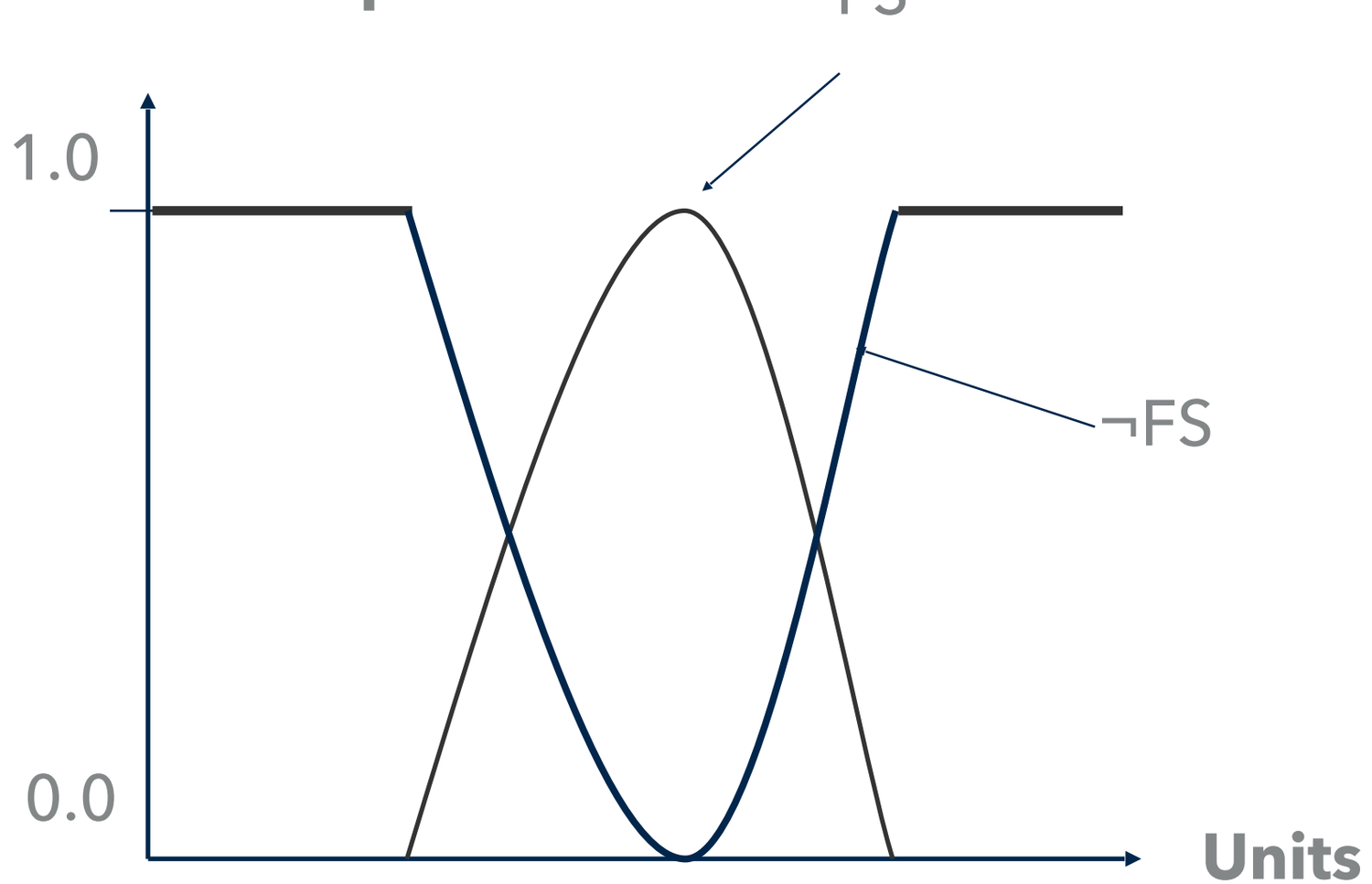
EXAMPLE FUZZY VARIABLE

- Each function tells us how much we consider a character in the set if it has a particular aggressiveness value
- Or, how much truth to attribute to the statement: “The character is nasty (or meek, or neither)?”

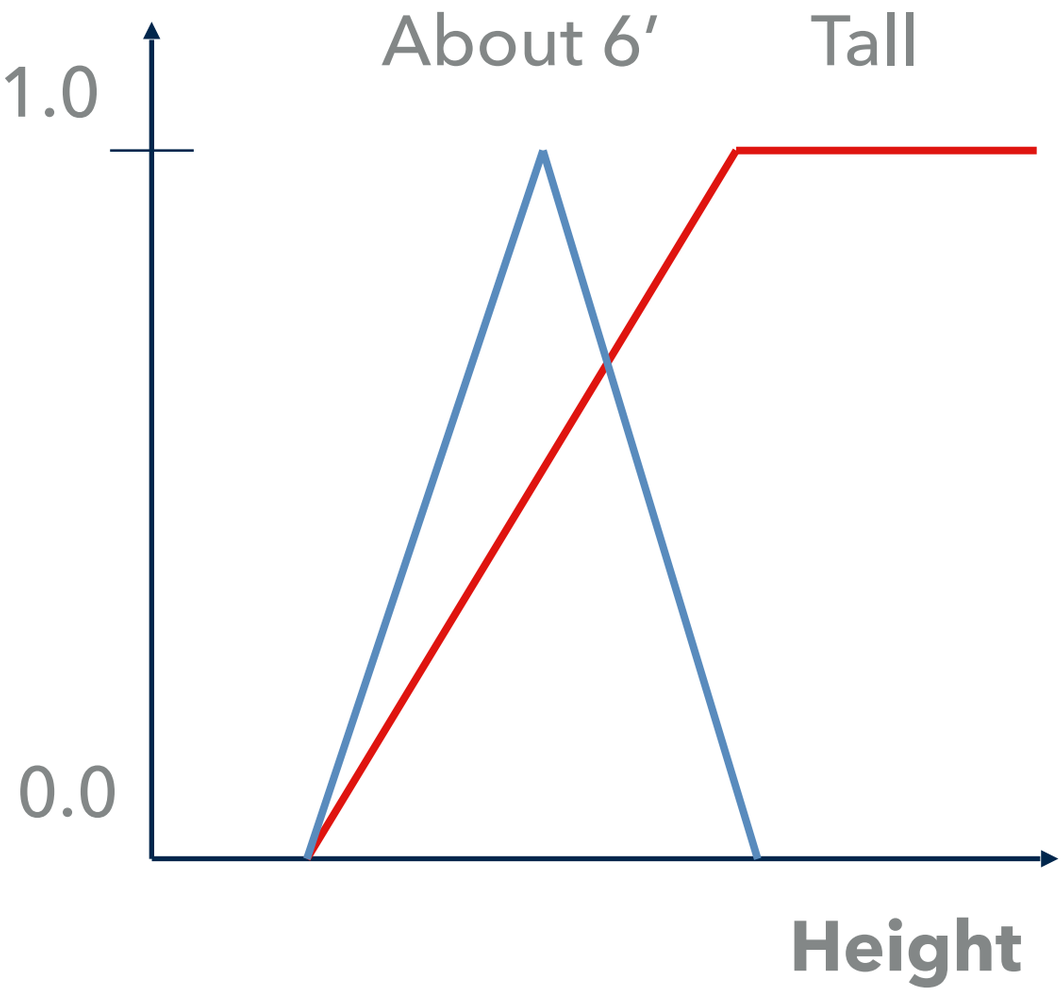


FUZZY SET OPERATIONS: COMPLEMENT

- The degree to which you believe something is **not** in the set is 1.0 minus the degree to which you believe it is in the set

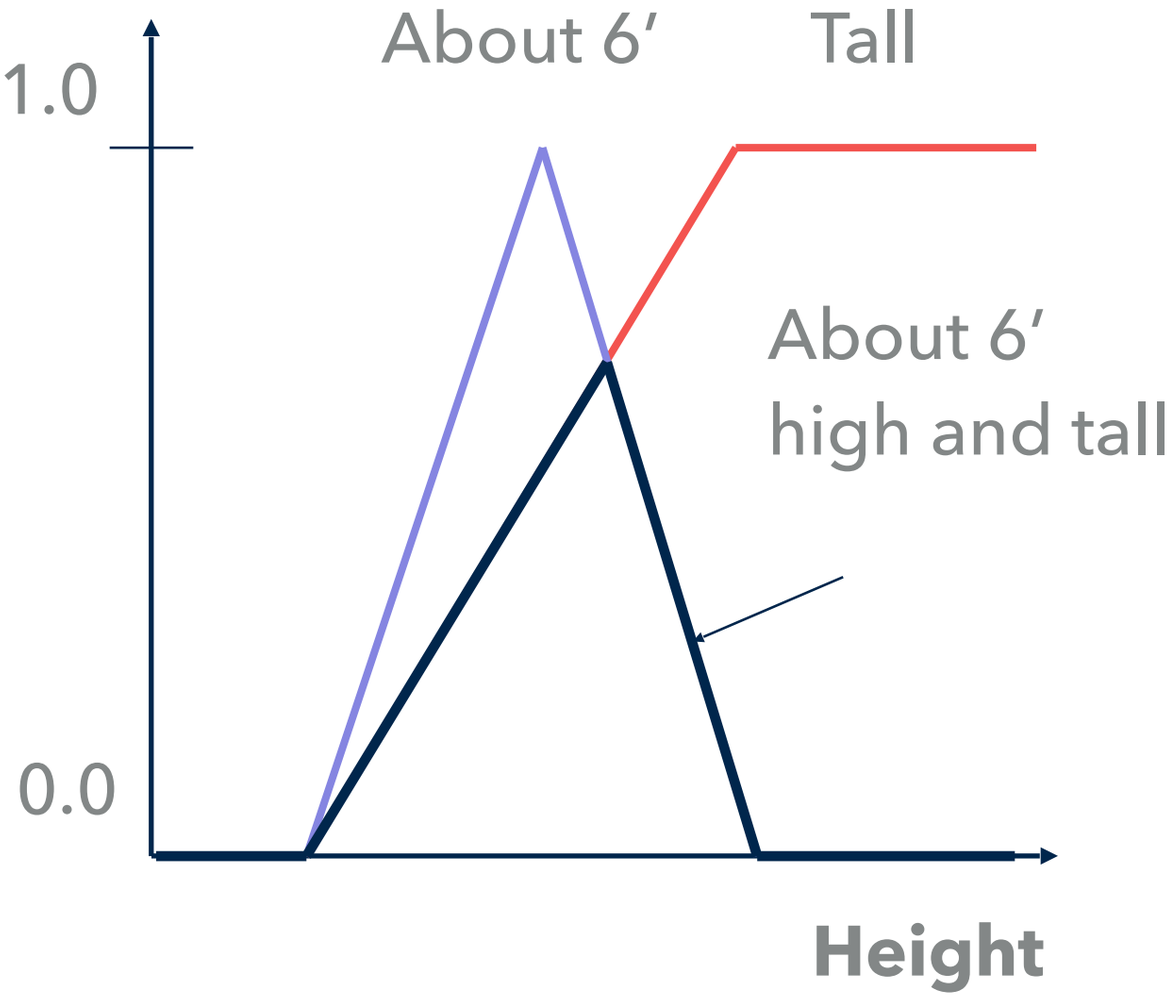


FUZZY SET: INTERSECTION (AND)



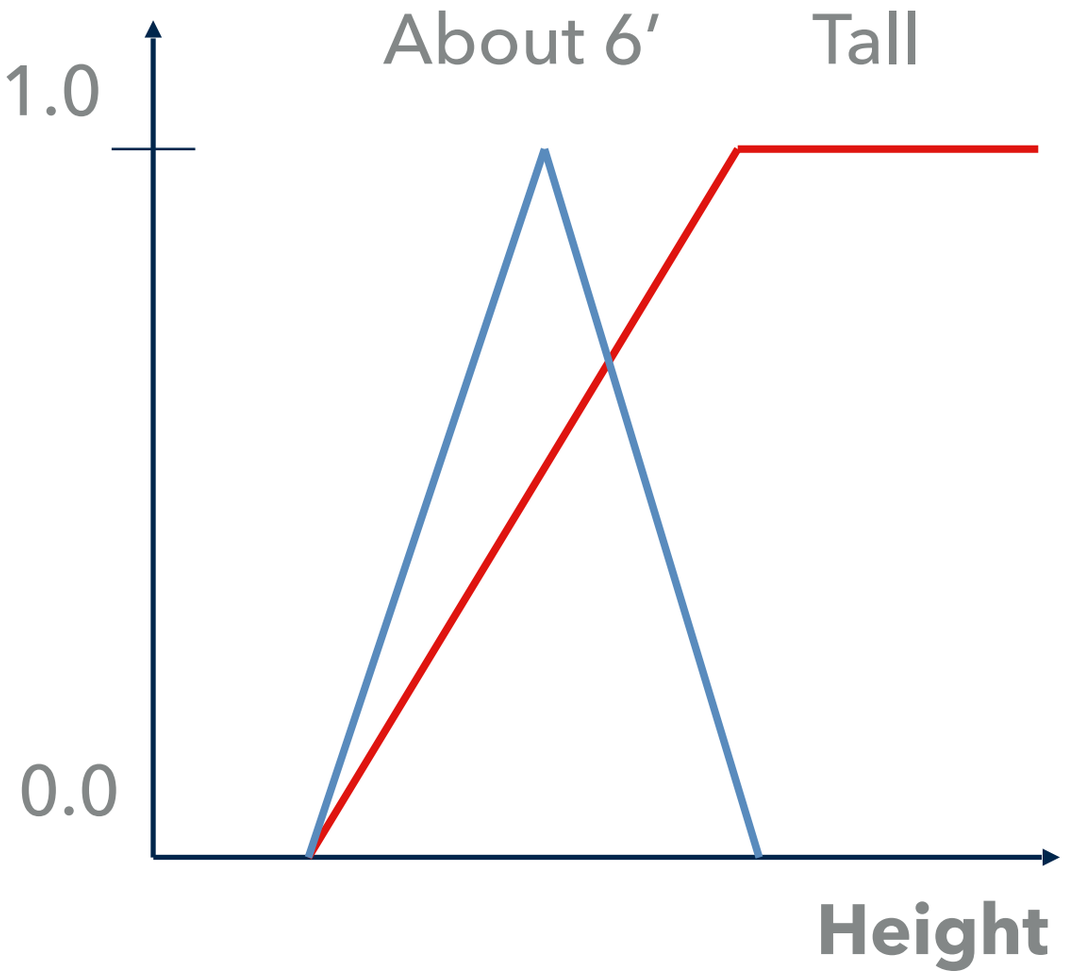
- If you have *x* degree of faith in statement A, and *y* degree of faith in statement B, how much faith do you have in the statement A and B?
  - e.g how much faith in "that person is about 6' high **and** tall"

FUZZY SET: INTERSECTION (AND)



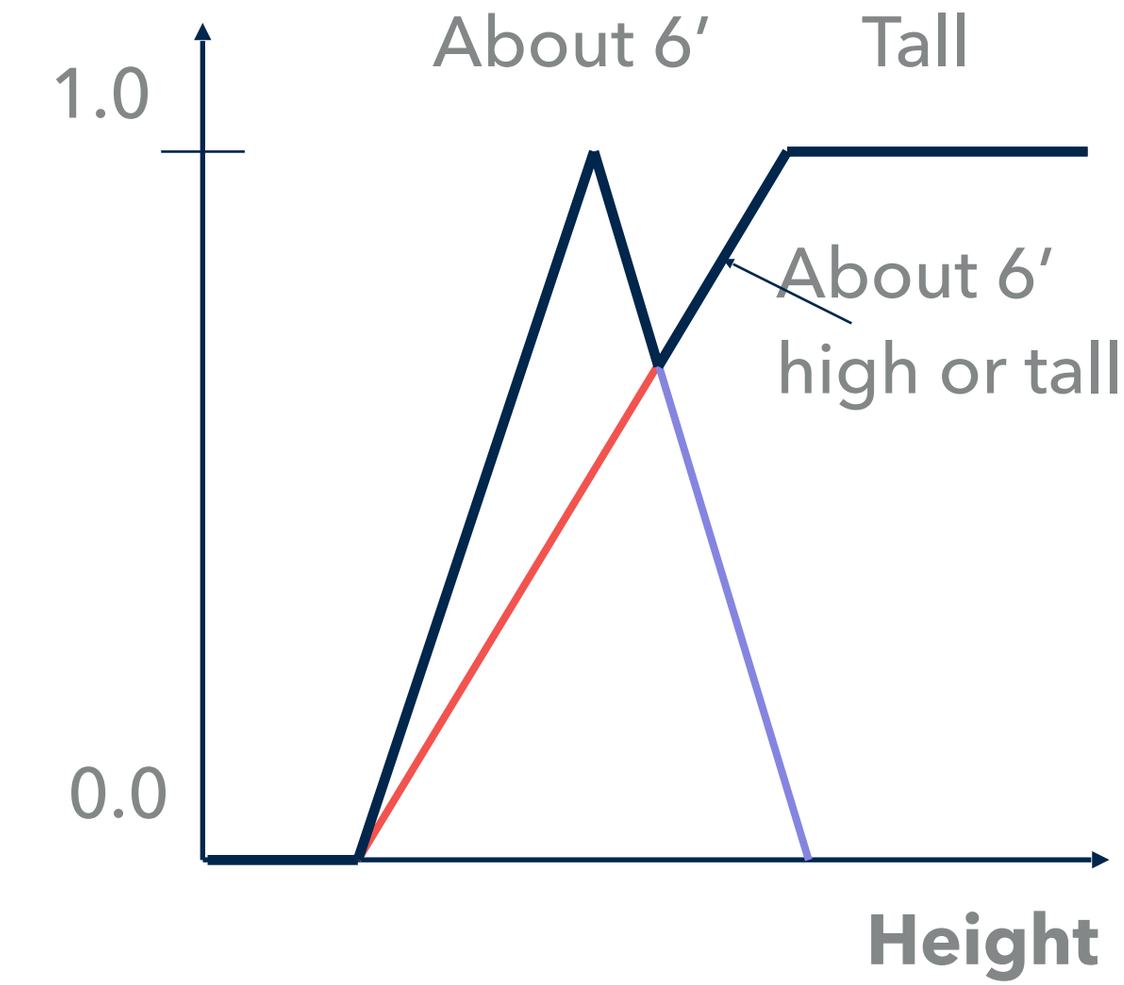
- Assumption: Membership in one set does not affect membership in another
- Take the *min* of your beliefs in each individual statement
- Also works if statements are about different variables
  - Dangerous **and** injured - belief is the *min* of the degree to which you believe they are dangerous and injured

FUZZY SET: UNION (OR)



- If you have *x* degree of faith in statement A, and *y* degree of faith in statement B, how much faith do you have in the statement A or B?
  - e.g. how much faith in “that person is about 6’ high **or** tall”

FUZZY SET: UNION (OR)



- Take the *max* of your beliefs in each individual statement
- Also works if statements are about different variables
  - Dangerous **or** injured - belief is the *max* of the degree to which you believe they are dangerous or injured

FUZZY RULES

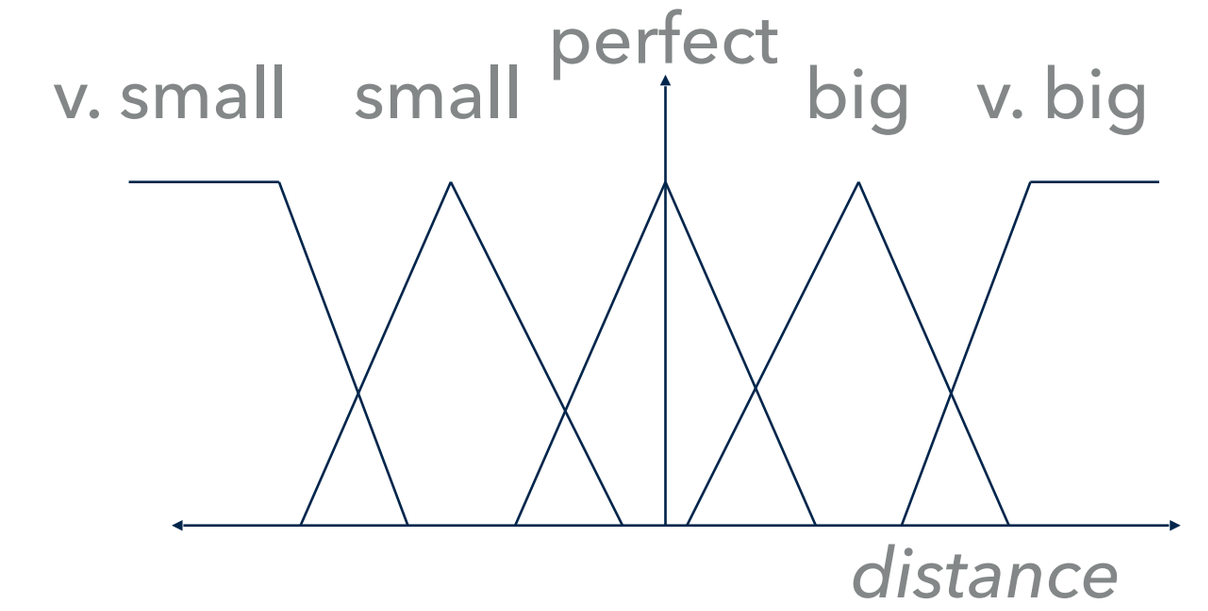
- "If our distance to the car in front is small, and the distance is decreasing slowly, then decelerate quite hard"
  - Fuzzy variables in blue
  - Fuzzy sets in red
- We have a certain belief in the truth of the condition, and hence a certain strength of desire for the outcome
- Multiple rules may match to **some degree**, so we require a means to arbitrate and choose a particular goal -- **defuzzification**

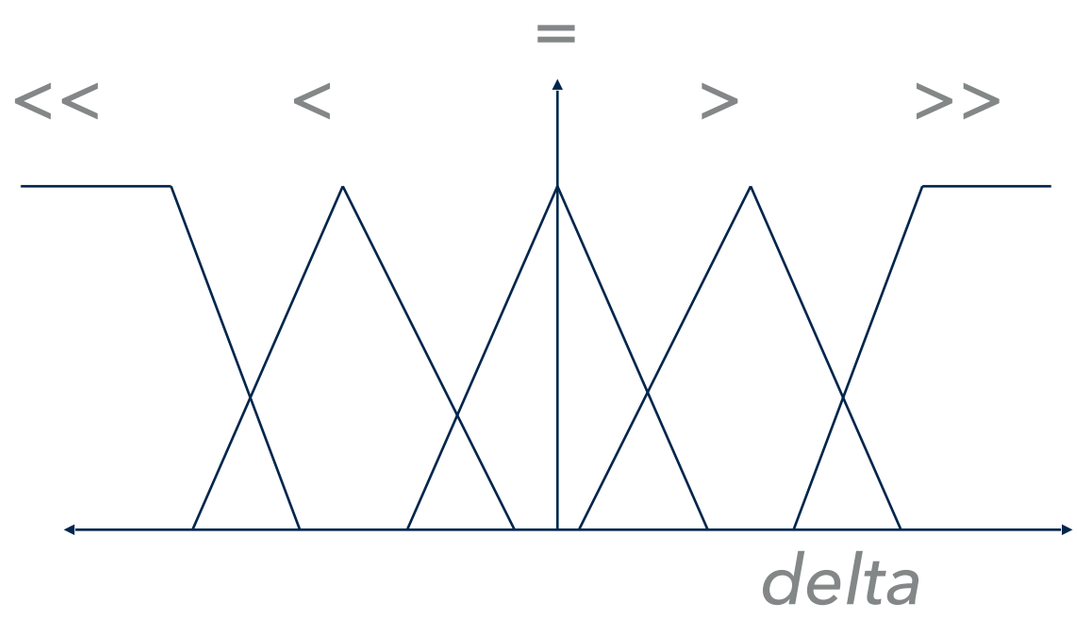
FUZZY RULES EXAMPLE

- Rules for controlling a car:
  - Variables are *distance* to car in front, *delta* is how fast it's changing, and *acceleration* is how to apply it
  - Sets are:
    - Very small, small, perfect, big, very big (distance)
    - Shrinking fast, shrinking, stable, growing, growing fast (delta)
    - Brake hard, slow down, none, speed up, floor it (acceleration)
    - Rules for every combination of distance and delta sets define acceleration set

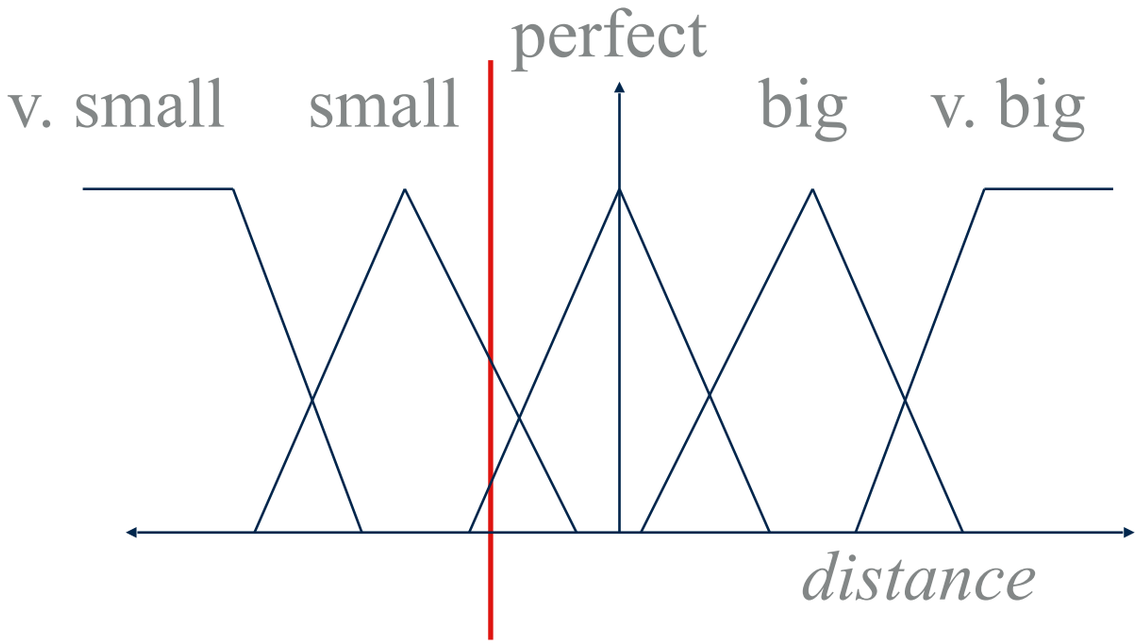
(from Game Programming Gems)

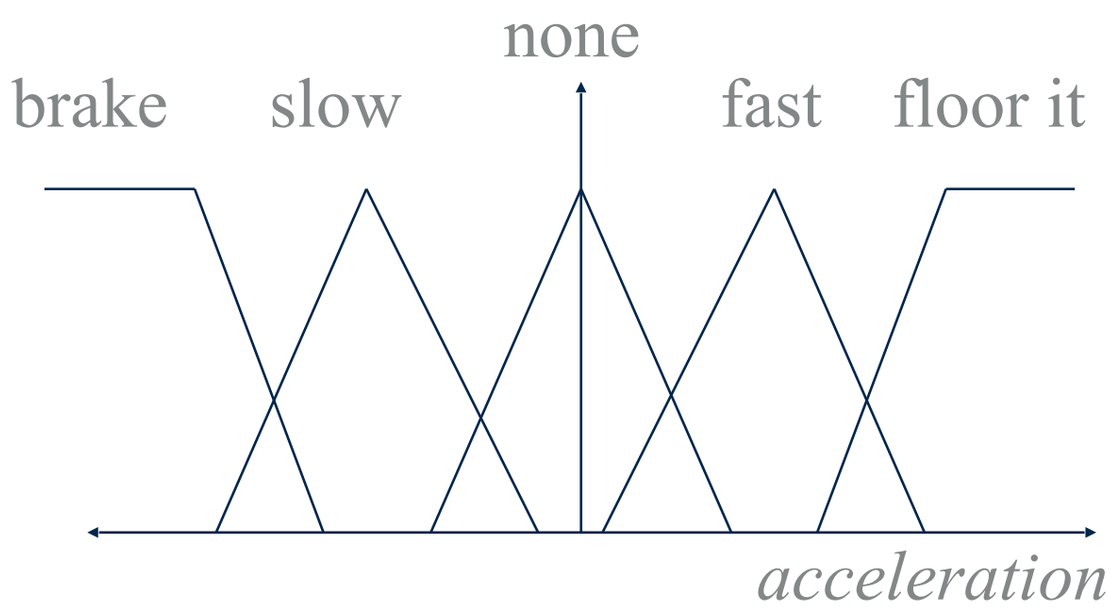
SET DEFINITIONS

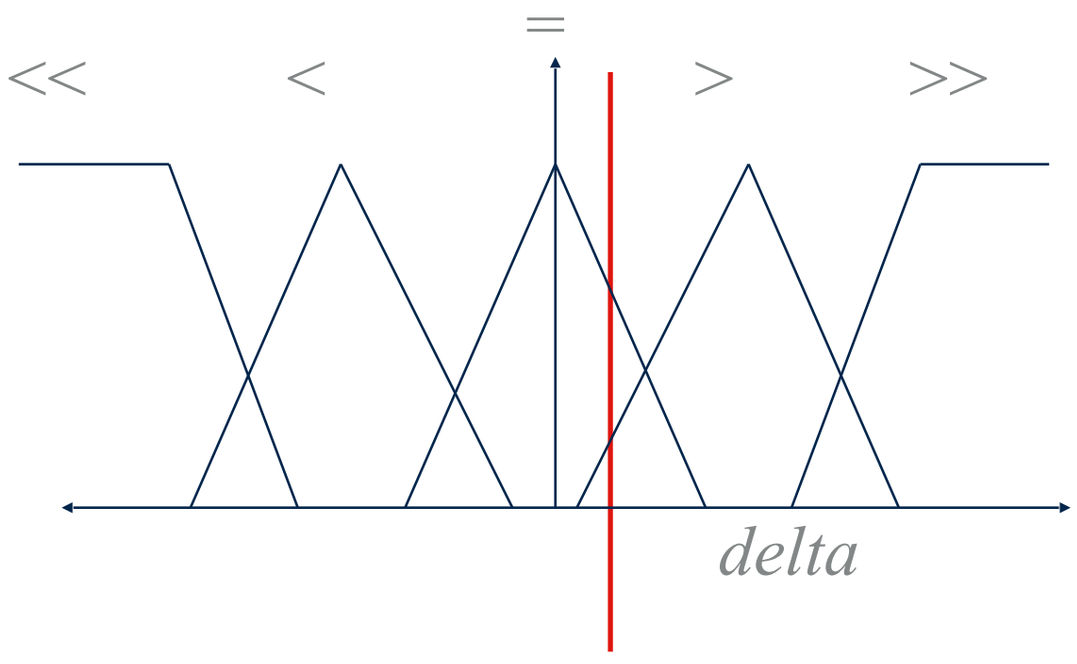




INSTANCE







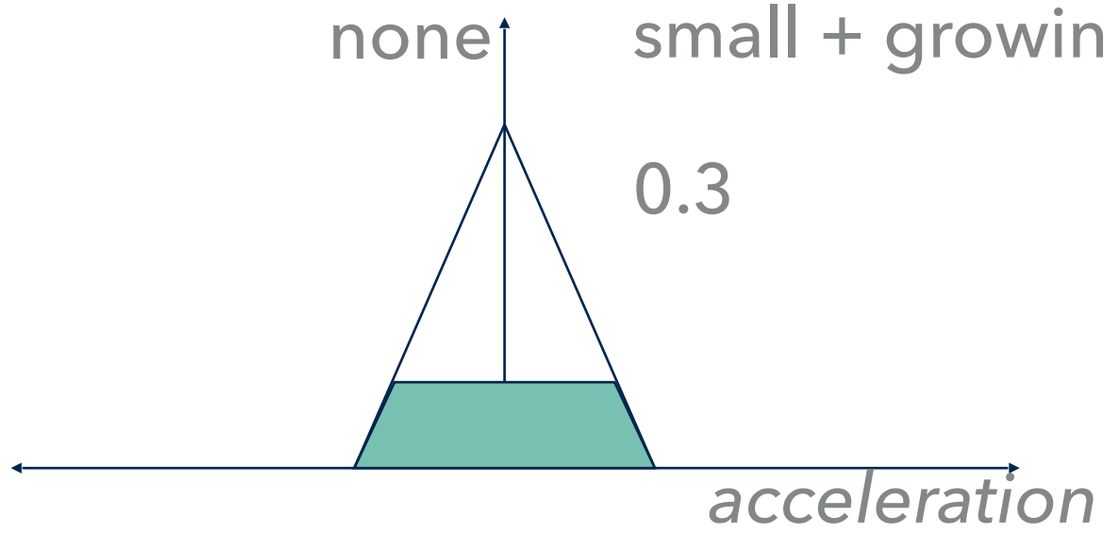
- Distance could be considered small or perfect
- Delta could be stable or growing
- What is acceleration?

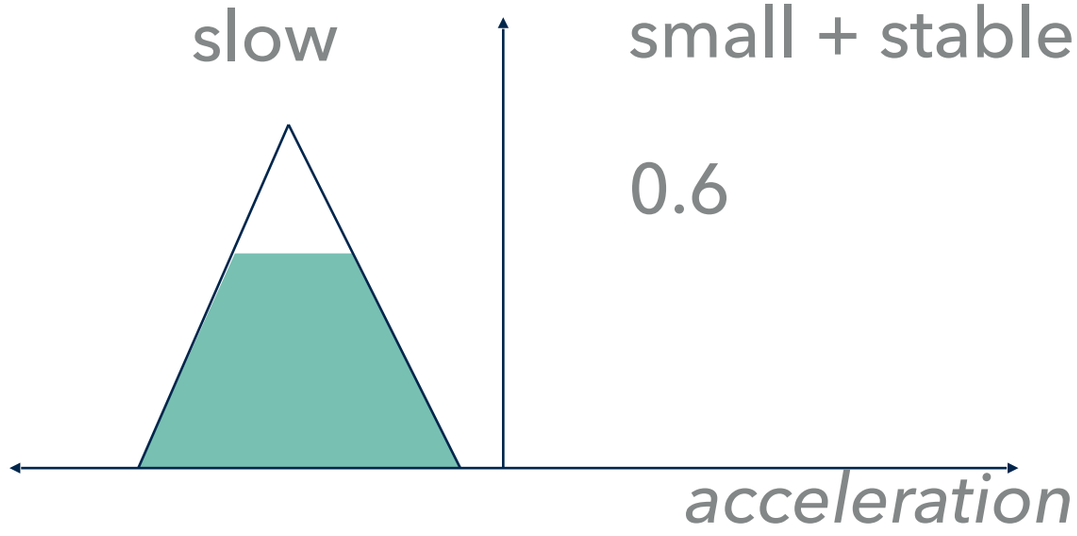
MATCHING

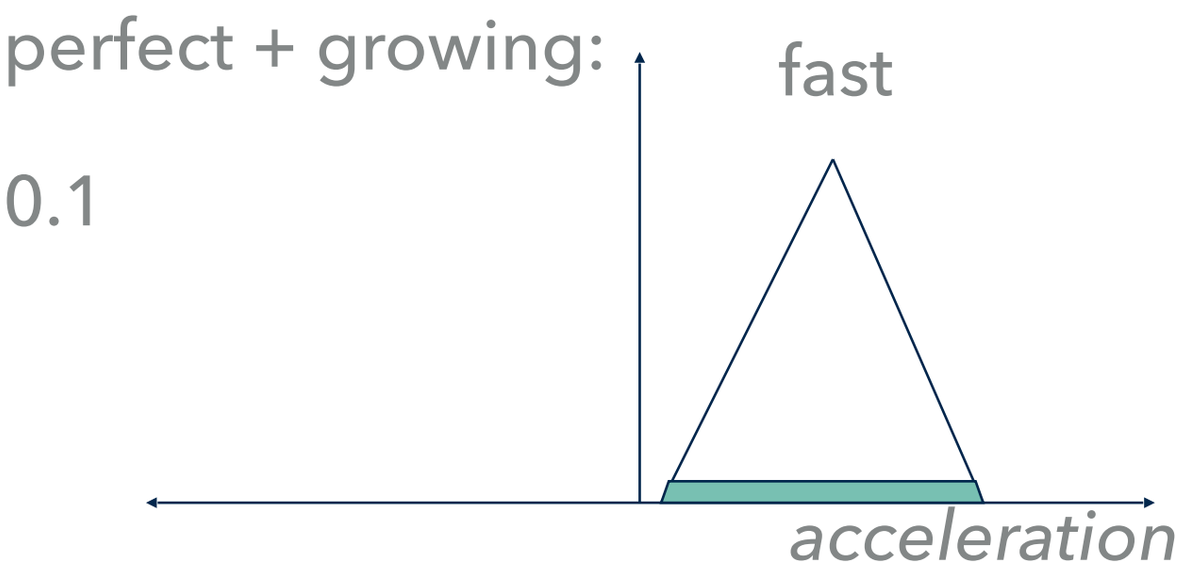
- Relevant rules are:
  - If distance is small and delta is growing, maintain speed
  - If distance is small and delta is stable, slow down
  - If distance is perfect and delta is growing, speed up
  - If distance is perfect and delta is stable, maintain speed
- For first rule, “distance is small” has 0.75 truth, and “delta is growing” has 0.3 truth
  - So the truth of the AND is 0.3
- Other rule strengths are 0.6, 0.1 and 0.1

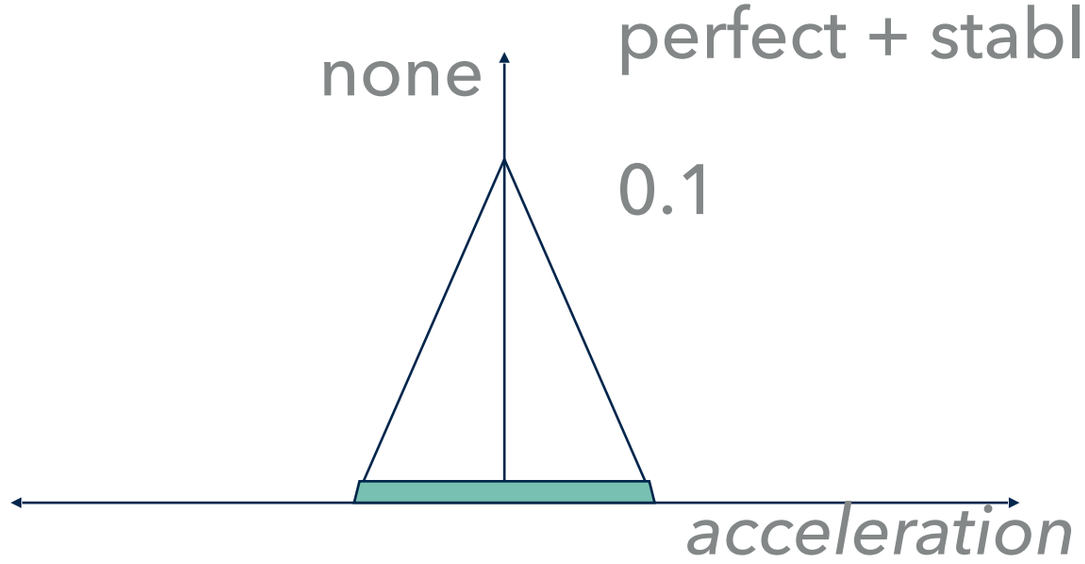
FUZZY INFERENCE

- For each *rule*, clip *action* fuzzy set by belief in rule









DEFUZZIFICATION EXAMPLE

- Three actions (*sets*) we have reason to believe we should take, and each action covers a range of values (*accelerations*)
- Two options in going from current state to a single value:
  - Mean of Max: Take the rule we believe most strongly, and take the (weighted) average of its possible values
  - Center of Mass: Take all the rules we partially believe, and take their weighted average
- In this example, we slow down either way, but we slow down more with Mean of Max
  - Mean of max is cheaper, but center of mass exploits more information

EVALUATION OF FUZZY LOGIC

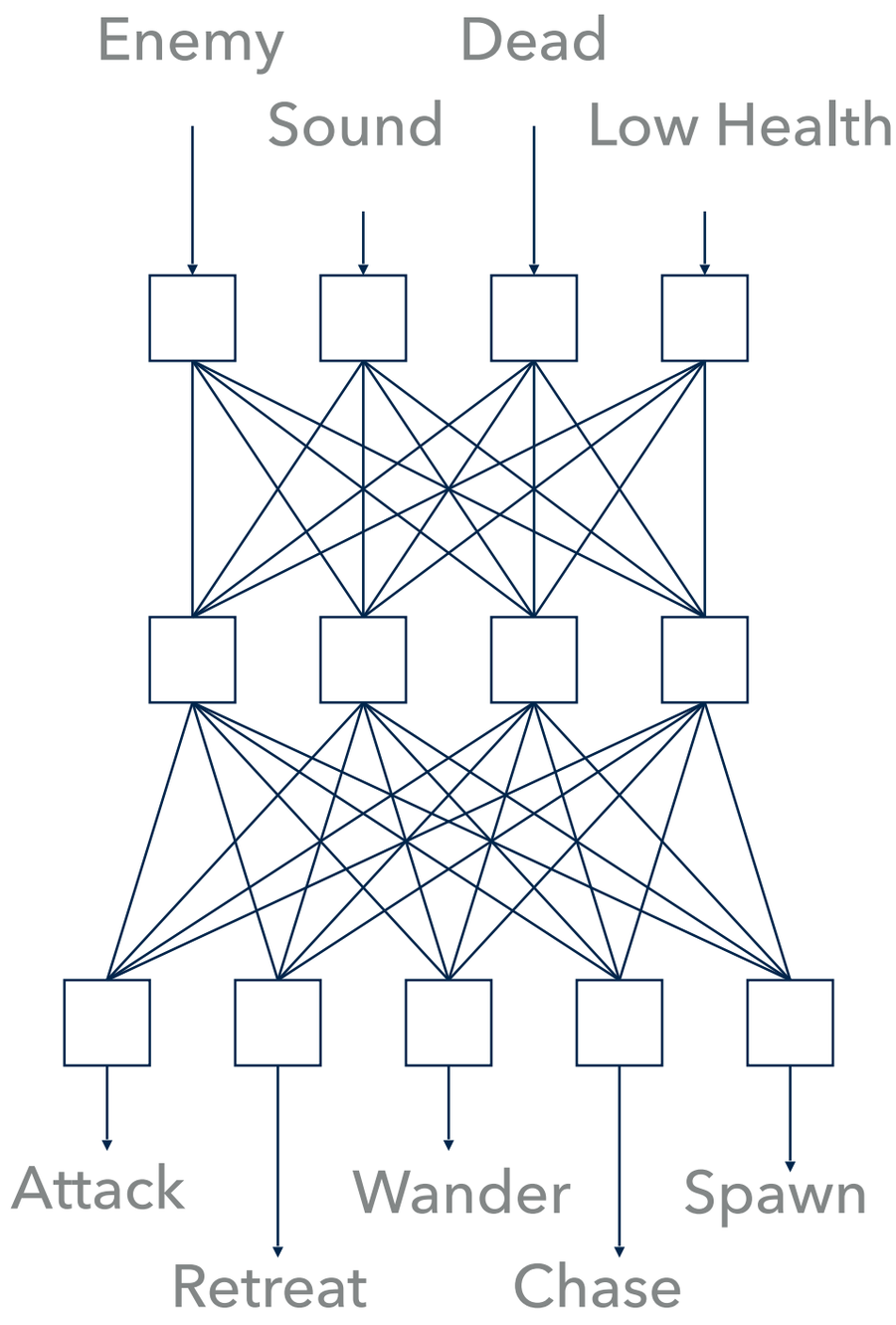
- Does not necessarily lead to non-determinism
- Advantages
  - Allows use of continuous valued actions while still writing “crisp” rules – can accelerate to different degrees
  - Allows use of “fuzzy” concepts such as medium
  - Biggest impact is for control problems
    - Can avoid discontinuities (but not always)
- Disadvantages
  - Sometimes results are unexpected and hard to debug
  - Additional computational overhead
  - Other ways to get continuous functions

SIDE NOTE: NEURAL NETWORKS

- Inspired by natural decision making structures (real nervous systems and brains)
- If you connect lots of simple decision making pieces together, they can make more complex decisions
  - Compose simple functions to produce complex functions
- Take multiple numeric input variables
- Produce multiple numeric output values
- Threshold outputs to turn them into discrete values
- Map discrete values onto classes, and you have a classifier!
- Also work as approximation functions

NEURAL NETWORK FOR QUAKE

- Four input perceptron
  - One input for each condition
- Four perceptron hidden layer
  - Fully connected
- Five output perceptron
  - One output for each action
  - Choose action with highest output
  - Or, probabilistic action selection
    - Choose at random weighted by output



NEURAL NETWORKS IN GAMES AS AI

- Forza
- Supreme Commander 2
- Black & White

NEURAL NETWORKS IN GAMES: THE FUTURE

- Deep learning may not be ideal for the AI in games, but it can assist with streamlining the asset pipeline
  - Animations
  - Generated assets
  - etc...
- Example: Deep Learning for Combat Animations:
  - https://80.lv/articles/ea-studies-the-use-of-deep-learning-for-combat-animations/
- Example: Deep Reinforcement Learning for Racing:
  - https://www.nature.com/articles/s41586-021-04357-7

NEURAL NETWORKS EVALUATION

- Advantages
  - Handle errors well
  - Graceful degradation
  - Can learn novel solutions
  - Learning during play might be possible
- Disadvantages
  - Can’t understand how or why the learned network works
  - Examples must match real problems
  - Need many examples
  - Learning takes lots of processing

REFERENCES

- Nguyen, H. T. and Walker, E. A. A First Course in Fuzzy Logic, CRC Press, 1999.
- Rao, V. B. and Rao, H. Y. C++ Neural Networks and Fuzzy Logic, IGD Books Worldwide, 1995.
- McCuskey, M. Fuzzy Logic for Video Games, in Game Programming Gems, Ed. Deloura, Charles River Media, 2000, Section 3, pp. 319-329.
- <https://www.gamasutra.com/blogs/ChrisSimpson/20140717/221339/Behavior_trees_for_AI_How_they_work.php>
- <http://www.gamasutra.com/blogs/JakobRasmussen/20160427/271188/Are_Behavior_Trees_a_Thing_of_the_Past.php>
- <http://www.gameaipro.com/GameAIPro/GameAIPro_Chapter09_An_Introduction_to_Utility_Theory.pdf>