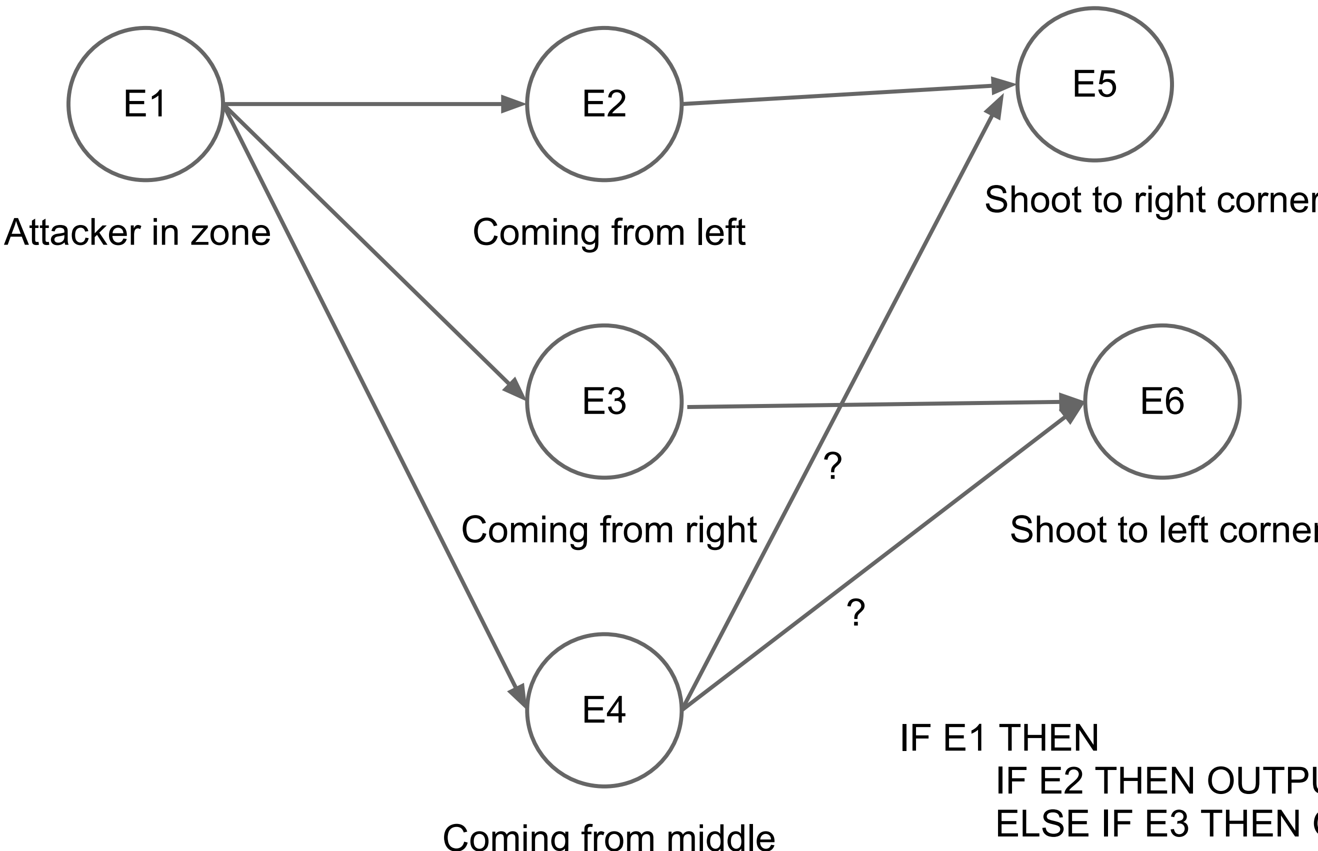
Game Artificial Intelligence by example

Evalds Urtans

Finite-state machine

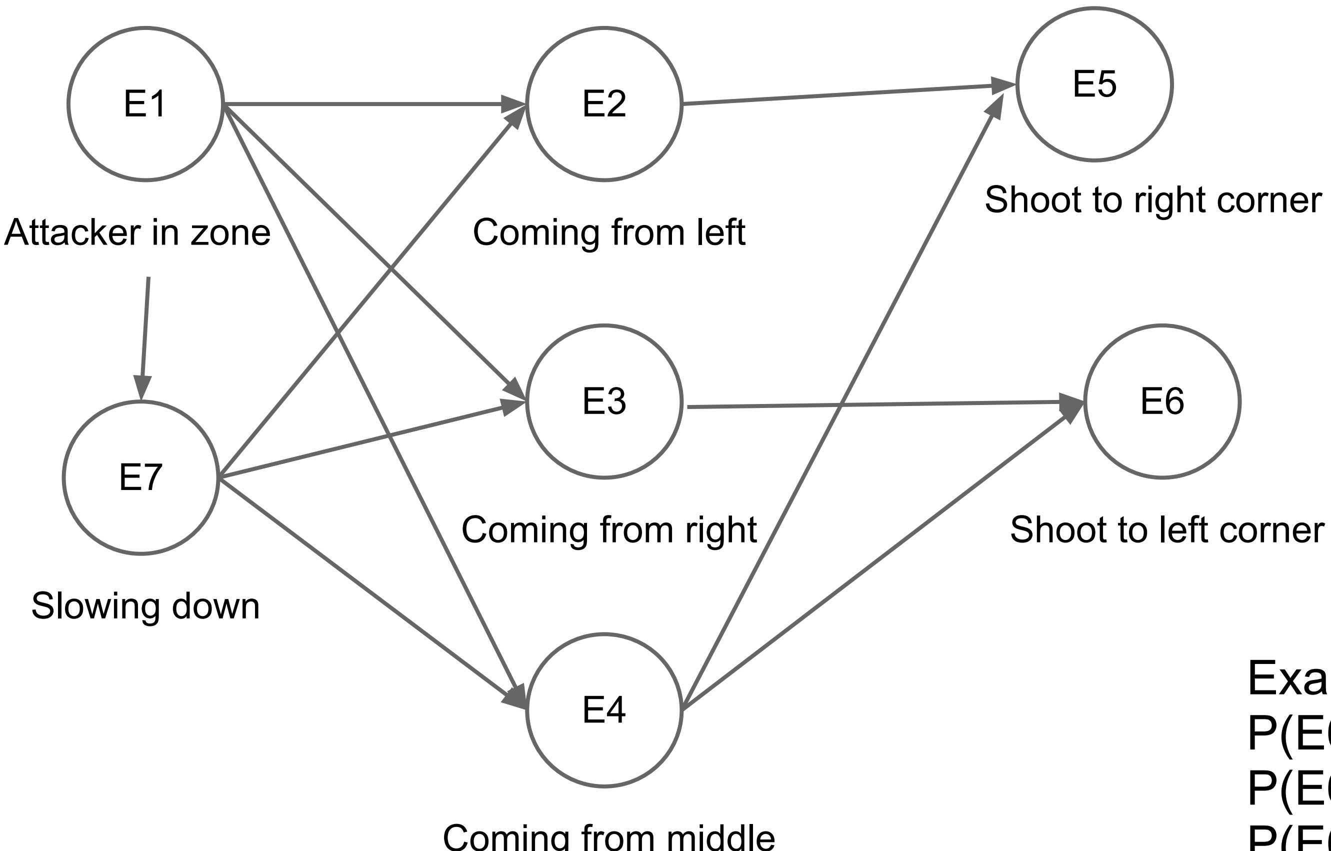
Ice Hockey game - what should goalie do?




```
IF E1 THEN
        IF E2 THEN OUTPUT = E5
        ELSE IF E3 THEN OUTPUT = E6
        ELSE IF E4 THEN OUTPUT = RAND(E5,E6)
END IF
```

Bayesian belief networks

Ice Hockey game - what should goalie do?



Example questions:
P(E6|E3) = ?
P(E6|E7) = ?
P(E6|E1) = ?

Bayesian belief networks

Ice Hockey game - what should goalie do?

Statistics (can & should be adaptive)

P(E1)	0.8
P(E1, E2)	0.4
P(E1, E2, E3)	0.0
P(E1, E2, E3, E4, ..)	..
P(E2)	0.7
P(E2, E1)	0.4
P(E2, E3)	0.0
P(E2, E4)	..
..	..

Bayesian belief networks

Ice Hockey game - what should goalie do?

$P(E6|E3) = ?$

Conditional probability (top-bottom, cause-to-effect):
$P(E6|E3) = P(E6,E3) / P(E3)$

Conditional probability (bottom-top, effect-to-cause):
$P(E3|E6) = P(E6,E3) / P(E6)$

out of this comes

Bayes rule (bottom-top, effect-to-cause):
$P(E3|E6) = P(E6|E3) * P(E3) / P(E6)$

Useful in problems where you have different statistical data, for example, only $P(E6|E3)$, $P(E3)$, $P(E6)$

Bayesian belief networks

Ice Hockey game - what should goalie do?

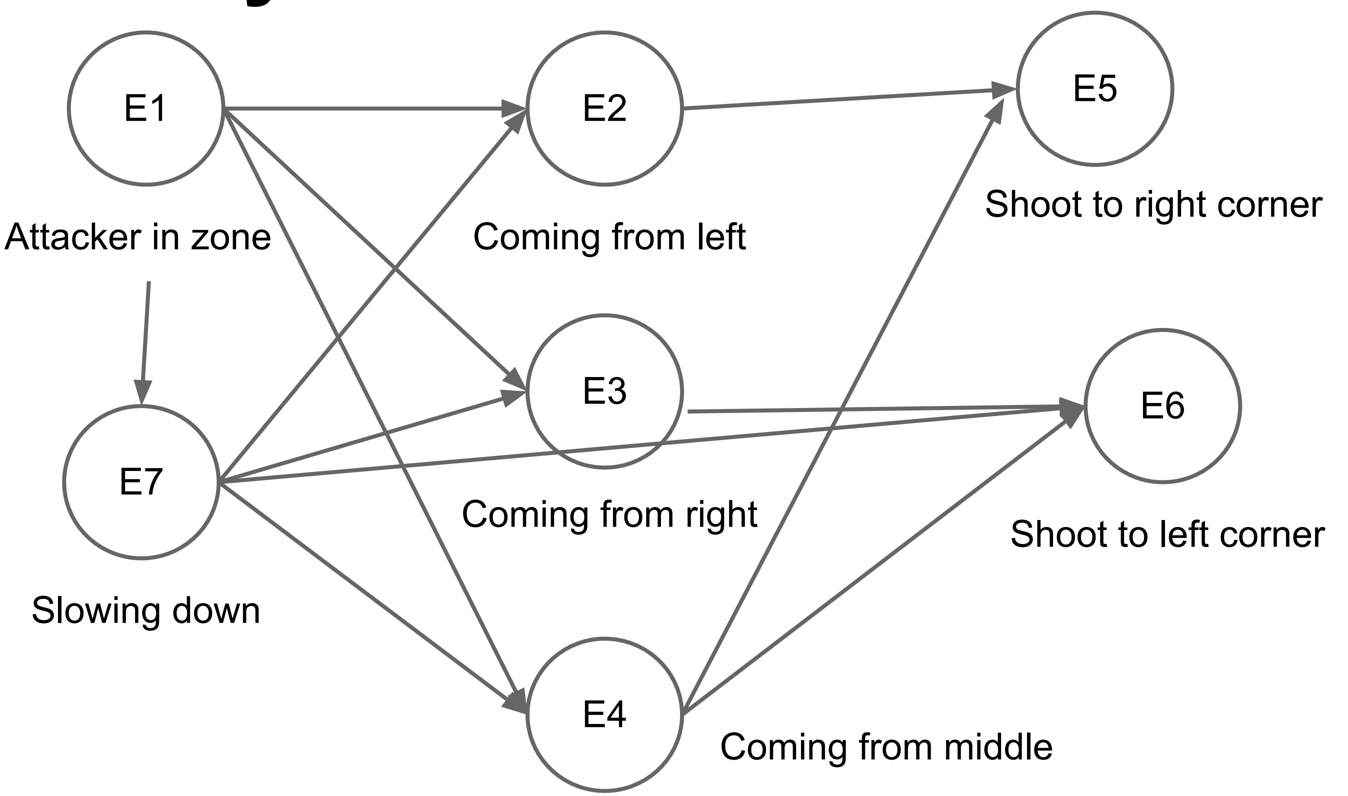
Network based statistics:
P(E6,E3) = 0.5
P(E3) = 0.3

$P_a(E6|E3) = P(E6,E3) / P(E3)$ [prediction with reasoning]

Independent statistics:
$P_b(E6|E3) = 0.2$ [prediction without reasoning]
* Investigate Markov Chains model (Finite state machine /w probabilites)

$P_a(E6|E3)$ is better prediction than $P_b(E6|E3)$

Reasoning based Dempster-Shafer theory



E4

$P(E5) = 0.4$
$P(E6_1) = 0.6$

E7

$P(E2) = 0.2$
$P(E3) = 0.2$
$P(E4) = 0.5$
$P(E6_2) = 0.1$

$P(E6_1) \mathrel{!=} P(E6_2) \mathrel{!=} P(E6)$

Reasoning based Dempster-Shafer theory

Credibility (based on DS), Plausibility, Belief

If E4 & E7 are observed

Calculate mass of interferences between co-dependent events
$m(E6) = P(E6_1)*P(E3) + P(E6_1)*P(E4) + P(E6_1)*P(E6_2) = 0.6*0.2 + 0.6*0.5 + 0.6*0.1 = 0.48$

m(not E6) =
$1 - m(E6) =$
$P(E5)*P(E2) + P(E5)*P(E3) + P(E5)*P(E4) + P(E5)*P(E6_2) + P(E6_1)*P(E2) = 0.4*0.2 + 0.4*0.2 + 0.4*0.5 + 0.4 * 0.1 + 0.6*0.2 = 0.52$ (double check, not necessary)

$Cedr(E6) = m(E6) / (1 - m(E6)) = 0.92$

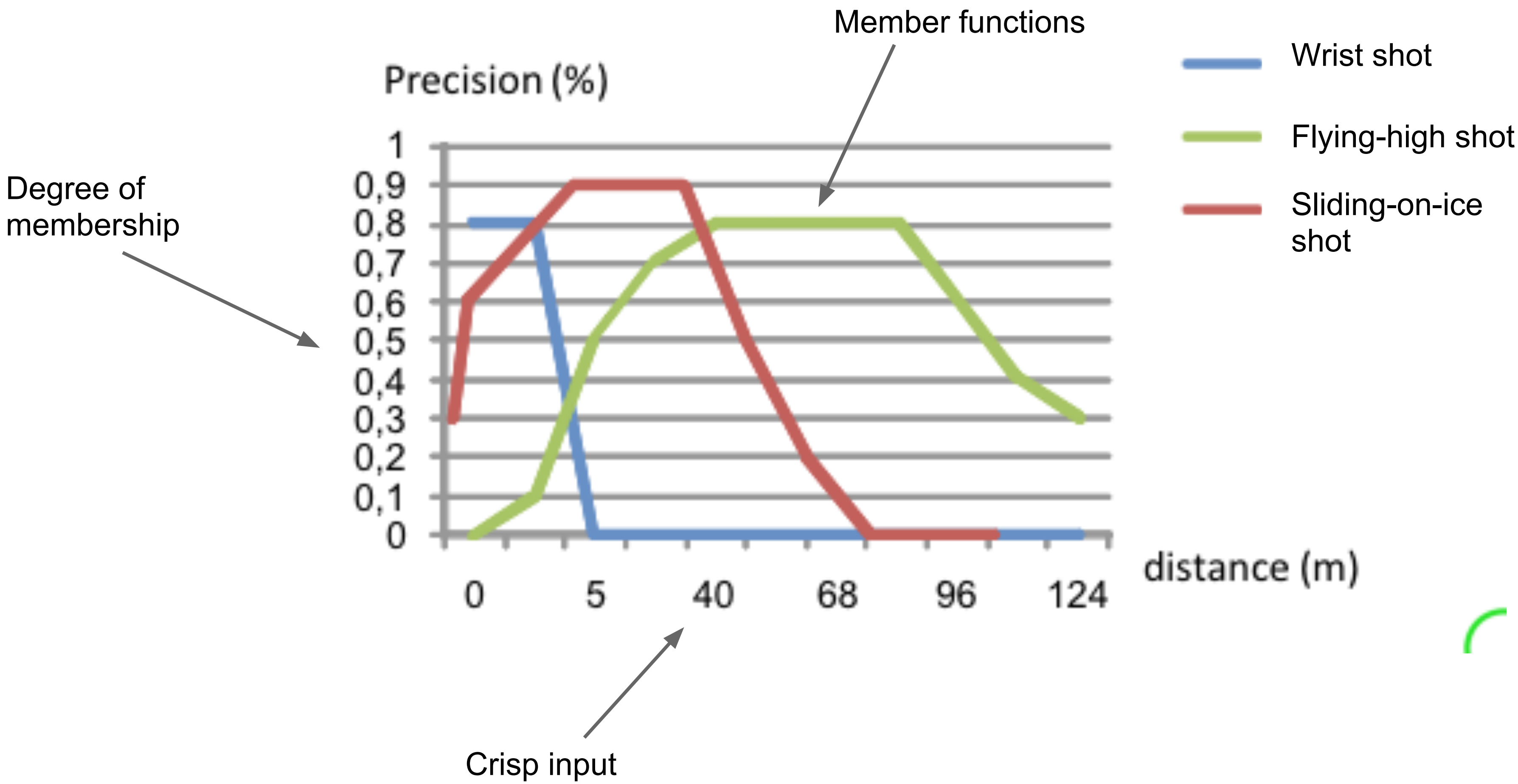
$Plaus(E6) = (P(E6_1) + P(E6_2)) / 2 = (0.6 + 0.1) / 2 = 0.35$
[average of all known probabilities P(E6)]

$Belief(E6) = (Cedr(E6) + Plaus(E6)) / 2 = (0.92 + 0.35) / 2 = 0.635$

* You can choose to be optimistic, pessimistic or rational

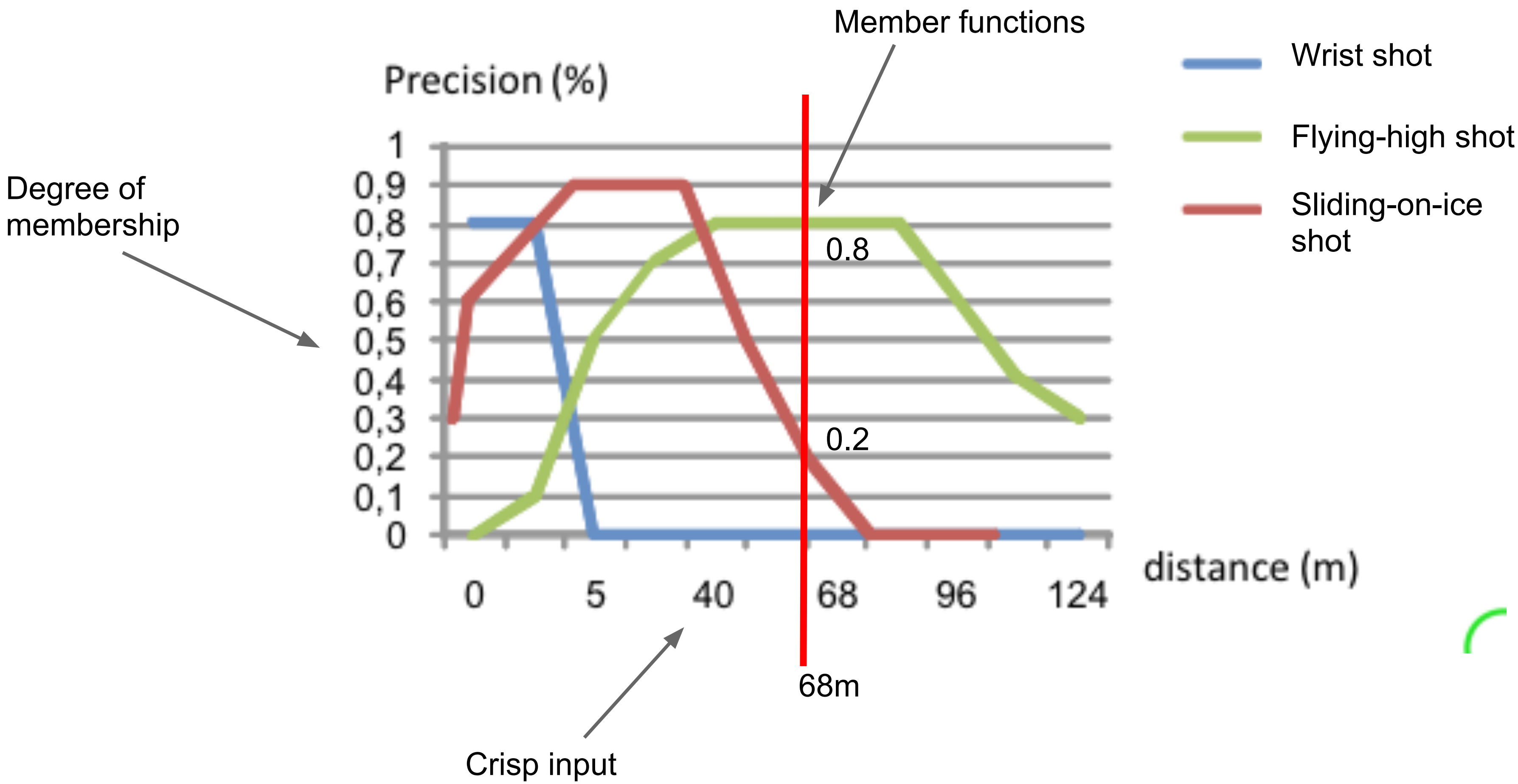
Fuzzy sets

Input = crisp value, output = degree of membership



Fuzzy sets

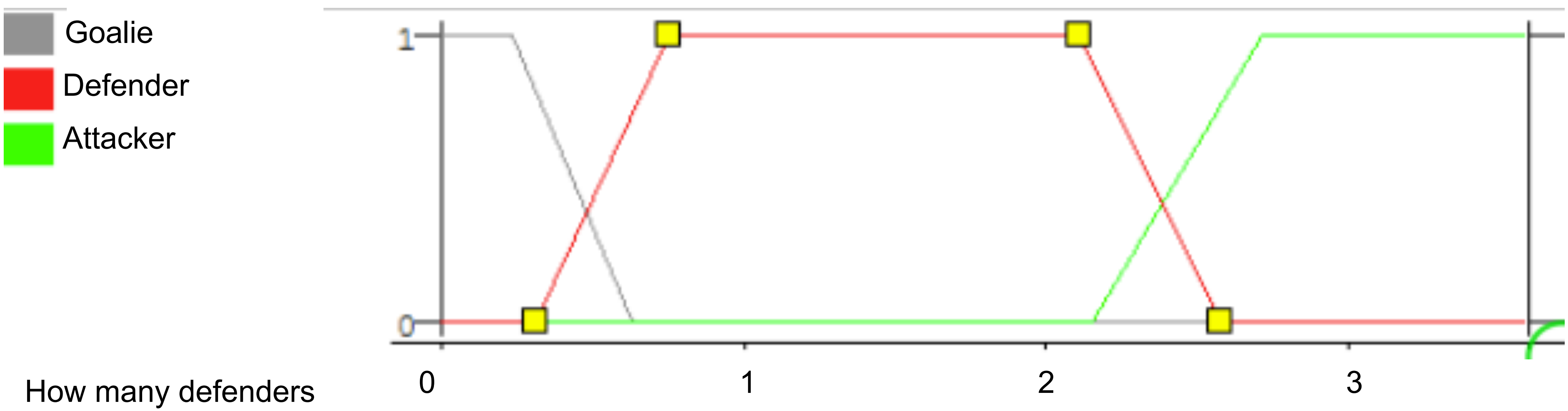
Input = crisp value, output = degree of membership



Fuzzy sets

Input = degree of membership, output = crisp value

Breakaway situation (attack close to goals):

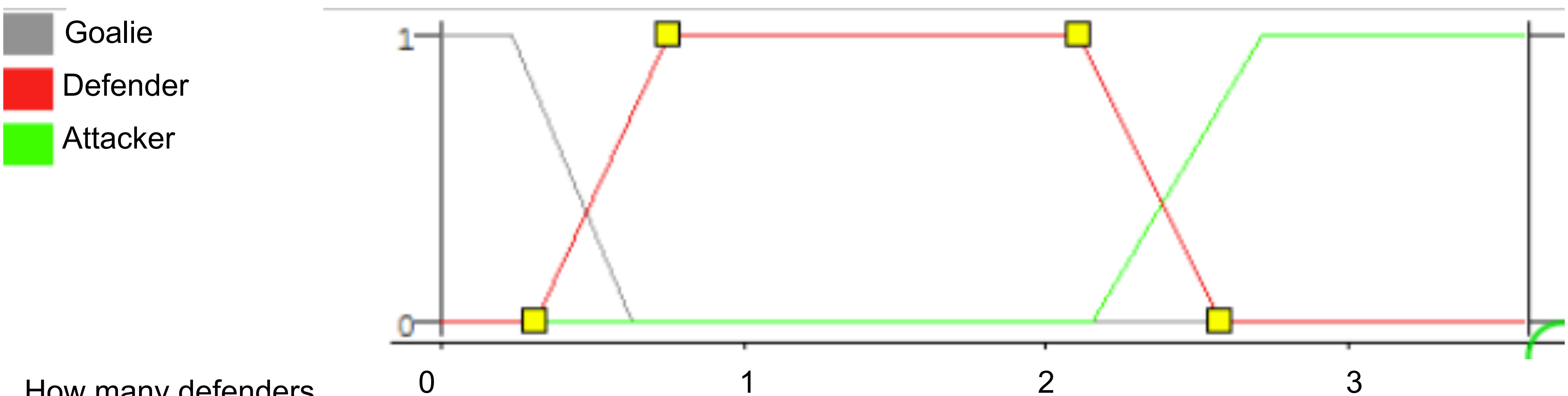


How many defenders needed to be allocated to opponent?

Fuzzy sets

Input = degree of membership, output = crisp value

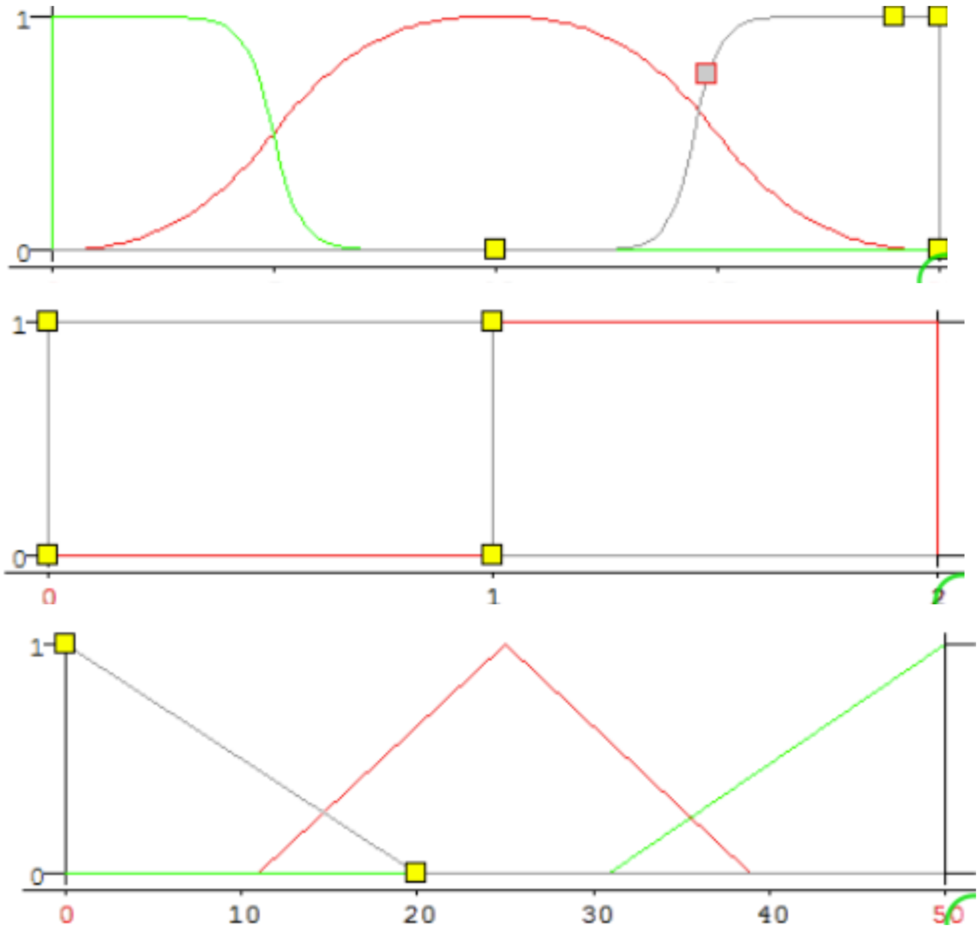
Breakaway situation (attack close to goals):



How many defenders needed to be allocated to opponent?

Fuzzy sets

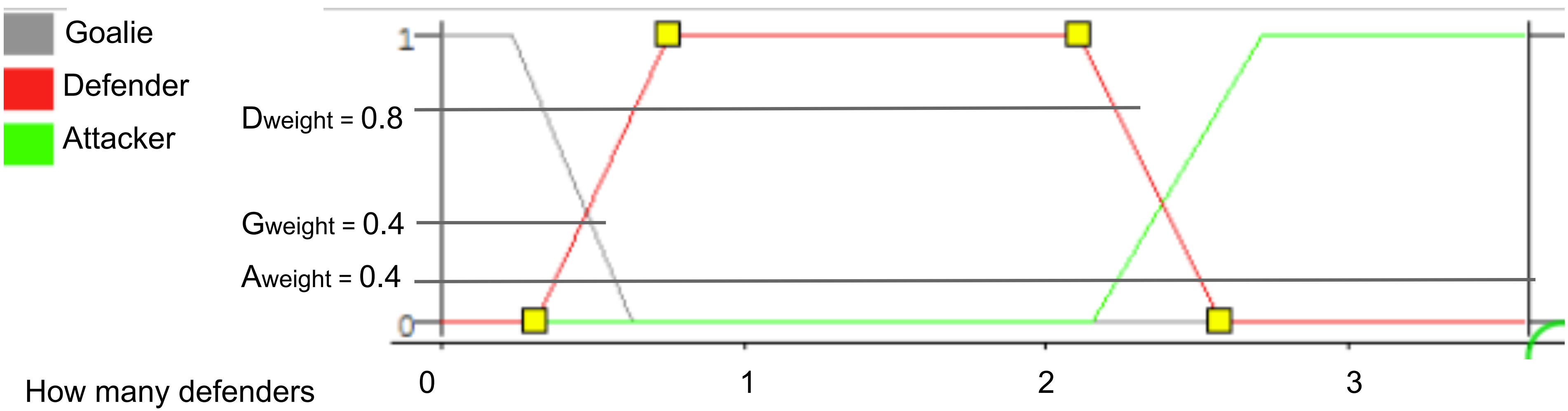
Member functions can be any geometrical function



Fuzzy sets

Input = degree of membership, output = crisp value

Breakaway situation (attack close to goals):

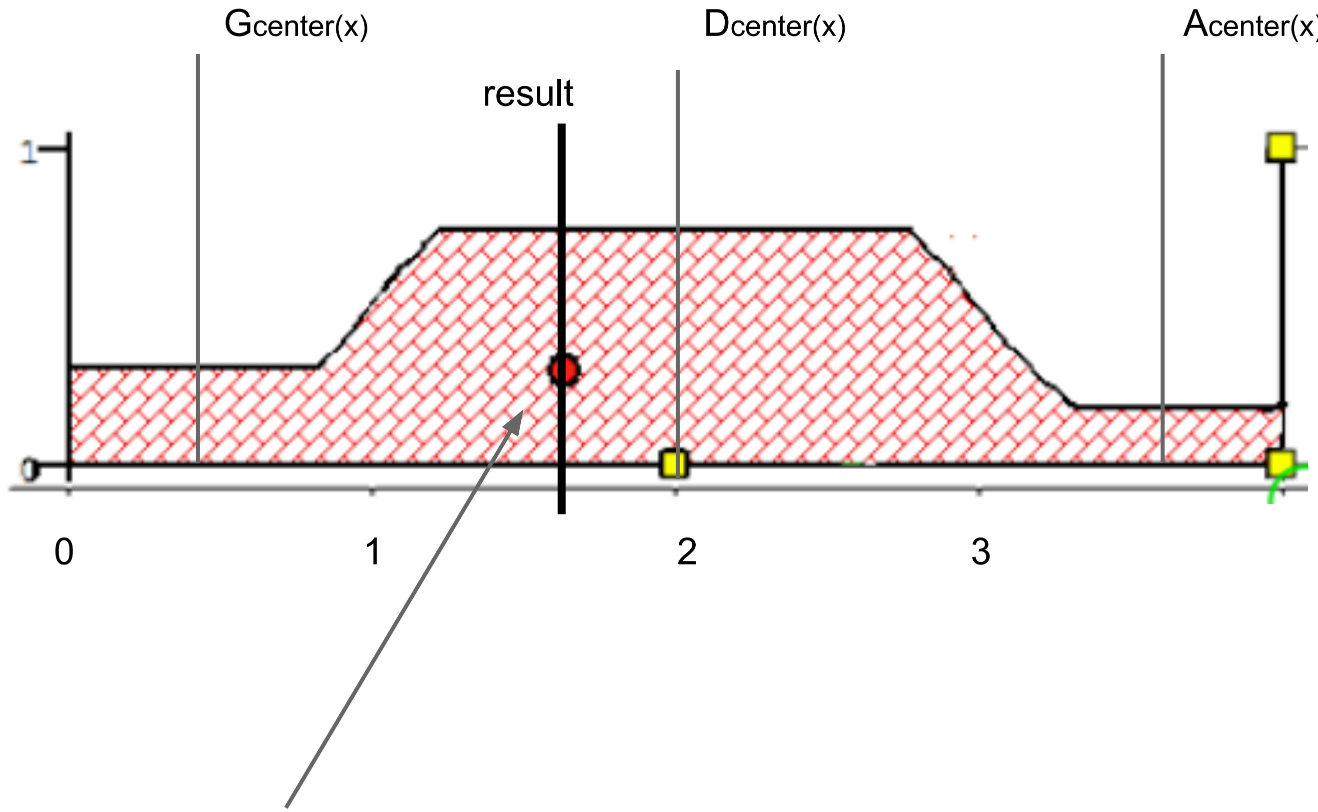


How many defenders needed to be allocated to opponent?

Fuzzification

Fuzzy sets

Goalie
Defender
Attacker



How many defenders needed to be allocated to opponent?

Defuzzification - calculated using centroid formula (centre of mass, singleton function)

$$result = (G_{weight} * G_{center}(x) + D_{weight} * D_{center}(x) + A_{weight} * A_{center}(x)) / (G_{weight} + D_{weight} + A_{weight})$$

Neural Networks

Phong game example

Figure 11 -- Ping-Pong Sample Application

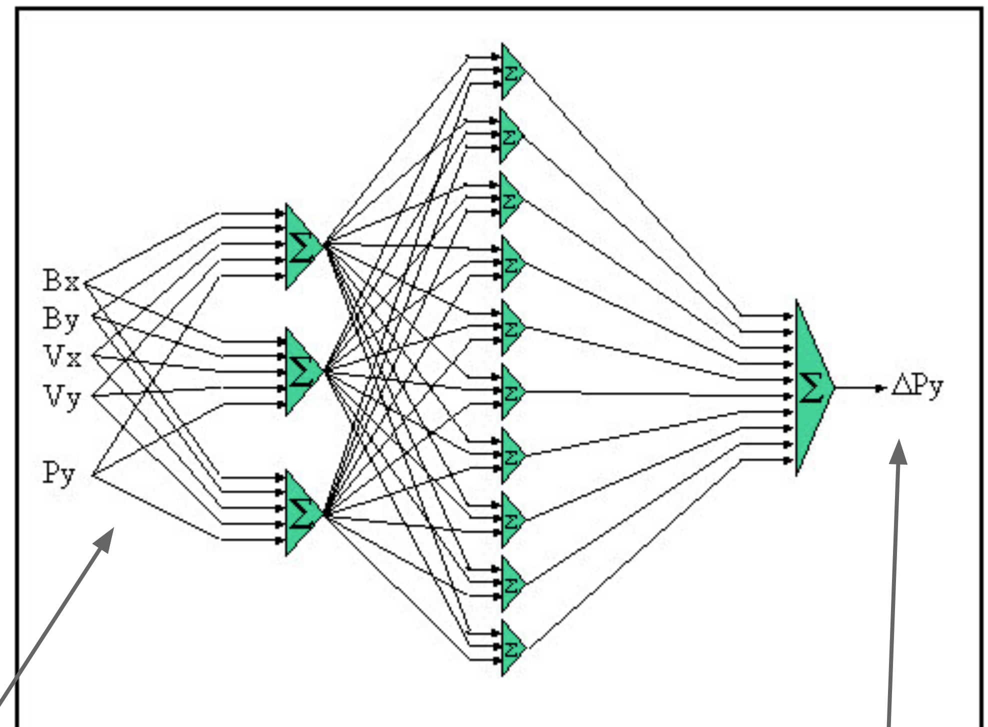


Figure 12 -- Neural Network for Ping Pong Game

Input:
Opponent's Y position,
Ball's X position,
Ball's Y position,
Ball's velocity

Output:
AI's Y position

Neural Networks

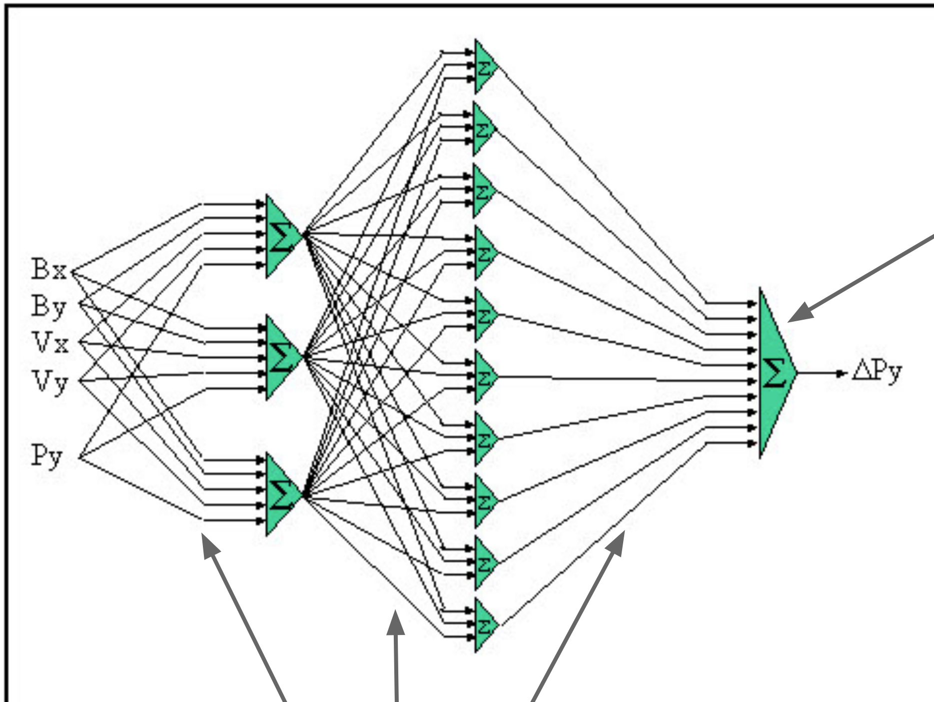
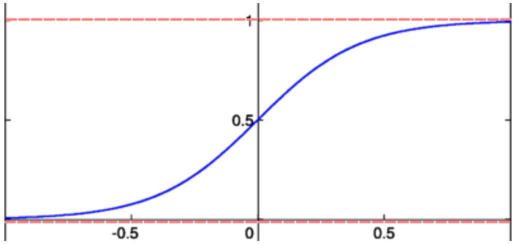


Figure 12 -- Neural Network for Ping Pong Game



Neuron resembled as sigmoid function

Weights in between connections
"connection strength between neurons"
These will change resembling a learning process (delta rule algorithm, back-propagation algorithm)

Further information

Bayesian networks

http://www.cs.indiana.edu/classes/b351-gass/Notes/uncertainty.html

https://controls.engin.umich.edu/wiki/index.php/Bayes_Rule,_conditional_probability,_independence

https://controls.engin.umich.edu/wiki/index.php/Bayesian_network_theory

http://staff.utia.cas.cz/vomlel/vomlel-ova-cze-jap-2009.pdf

http://opencourseware.kfupm.edu.sa/colleges/ccse/ics/ics381/files/2_Lectures%2030-31-Ch-14_Probabilistic%20Reasoning.pdf

http://ai.stanford.edu/~paskin/gm-short-course/lec1.pdf

https://controls.engin.umich.edu/wiki/index.php/Bayes_Rule,_conditional_probability,_independence

http://www.conradyscience.com/index.php/software
BayesiaLab - very interesting software for calculating probabilities using bayesian networks
Used in marketing research by most of the big brand companies

Further information

Dempster-Shafer theory

http://www.google.lv/books?id=4f5Gszjyb8EC&lpg=PR11&ots=9BTKlvOwsl&dq=dempster-shafer%20game%20ai&lr&hl=lv&pg=PA356#v=onepage&q&f=false

Neural Networks

http://www.cs.bham.ac.uk/~jxb/NN/nn.html

http://www.webpages.ttu.edu/dleverin/neural_network/neural_networks.html

http://software.intel.com/en-us/articles/an-introduction-to-neural-networks-with-an-application-to-games/

http://www.ibm.com/developerworks/library/l-neural/

http://takinginitiative.net/2008/04/03/basic-neural-network-tutorial-theory/

Further information

Fuzzy Logic

http://www.chebucto.ns.ca/Science/AIMET/archive/ddj/fuzzy_logic_in_C/

Interesting articles & research in Game-AI

http://aigamedev.com/