Knowledge Representation and Uncertain Reasoning

COMP3411/9814: Artificial Intelligence

How many rabbits are there?

How many rabbits are there?

- Perception isn't all in the eye.
- Knowledge is usually needed to understand the world

Lecture Overview

- Knowledge representation
- Reasoning
- Uncertainty
- Bayesian inference
- Fuzzy logic

Lecture Overview

- Knowledge representation
- Reasoning
- Uncertainty
- Bayesian inference
- Fuzzy logic

Levels of Abstraction

Qualitative, symbolic representation
Contains relational information
Easy to reason about
Easy to make generalisations
Memory efficient
Not suited to “low-level” perception

Feature-based Representations

Abstraction

Numeric (statistical) representation
No relational information
Difficult to reason about
Hard to make generalisations
Memory intensive
Well-suited for vision, sequential data

Iconic Representations

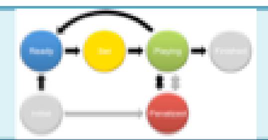
Iconic Representations

- Analogues to real world
  - Pixel representations like first layer of ANN
  - Maps
- Memory-based (requires a lot of memory)
- Fast, but ...
  - Do not generalise well
  - Difficult to perform inferences beyond pattern-response

Symbolic Representations

- State is represented by a set of abstract features and relations
  - For instance: expressions on logic, entity-relation graphs
- Can do complex reasoning over knowledge base
- Not well-suited to "low-level" perception

RoboCup Architecture

Game – Controller 

Roles (Robot-behaviour, eg Goalie, Striker, Supporter) 3

Skills (localise, position, find-ball, track-ball, get-behind-ball, kick-ball, getup) 2

 **State Estimation** Field-State + Robot-State 

Actuation (head-movement, leg/arm-movement, camera-switching, speakers, LEDs, wireless) 1

Sensing (cameras, IR, hall-effect, joint-angles, foot-sensors, IMU, bumpers, sonar, wireless)

Environment (this robot, ball, other robots)

Rule-Based Systems

- A production rule has the form

```
if <condition> then <conclusion>
```

- Production rule for dealing with the payroll of ACME, Inc., might be

```
rule r1_1
if the employer of Person is acme
then the salary of Person becomes large.
```

Rule-Based Systems

```
rule r1_1
if the employer of Person is acme
then the salary of Person becomes large.
```

- Production rules can often be written to closely resemble natural language

```
/* fact f1_1 */
the employer of joe_bloggs is acme.
```

- Capitalisation (like Prolog) indicates that “Person” is a variable that can be replaced by a constant, such as “joe_bloggs” or “mary_smith”, through pattern matching.

Rule-Based Systems

```
rule r1_1
if the employer of Person is acme
then the salary of Person becomes large.
```

```
rule r1_2
if the salary of Person is large
or the job_satisfaction of Person is true
then the professional_contentment of Person becomes true.
```

- Executing a rule may generate a new derived fact.
- There is a *dependency* between rules r1_1 and r1_2 since the conclusion of one can satisfies the condition of the other.

Rule-Based Systems

```
/* fact f1_1 */
the employer of joe_bloggs is acme.
```

```
rule r1_1
if the employer of Person is acme
then the salary of Person becomes large.
```

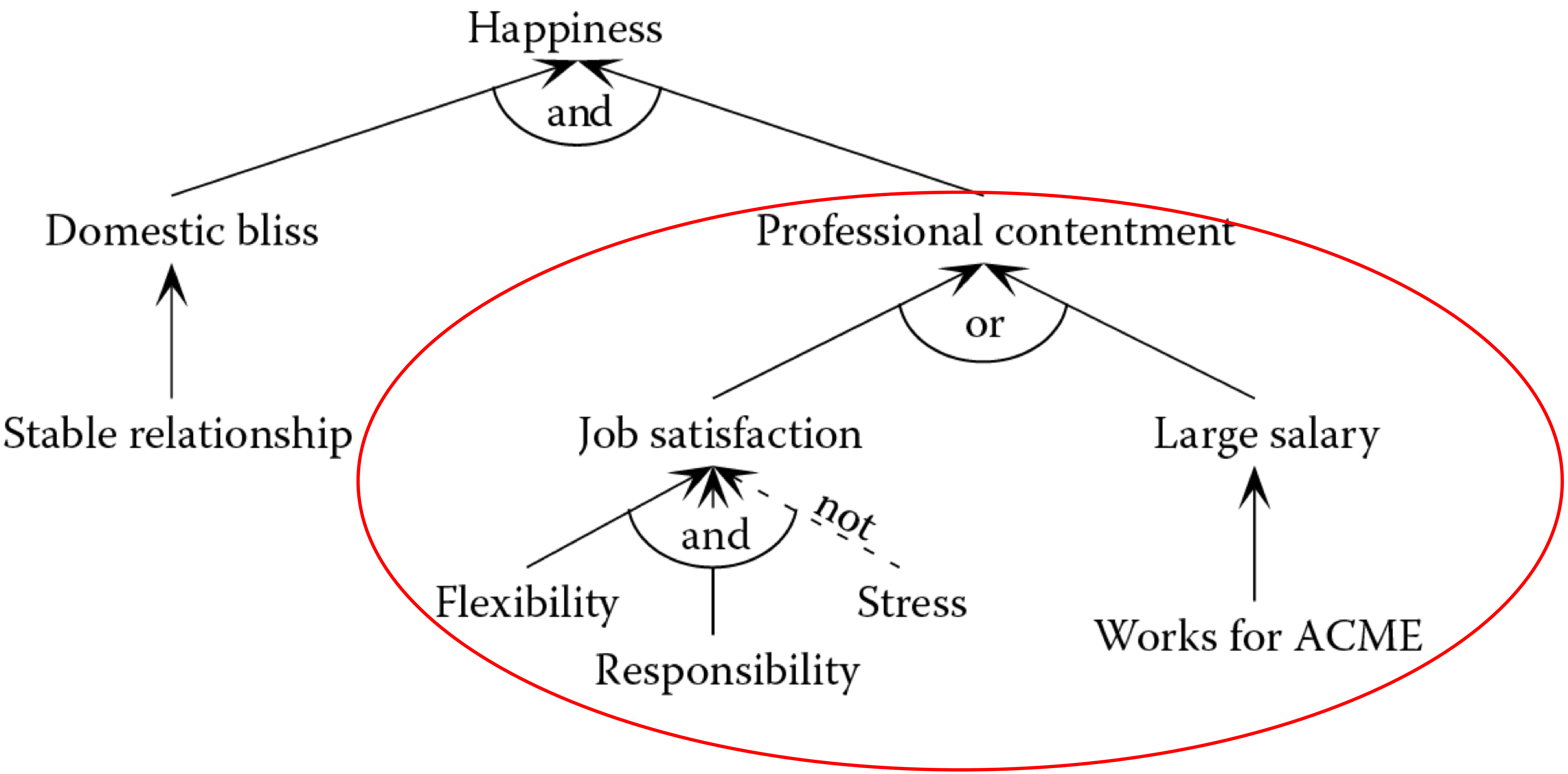
```
/* derived fact f1_2 */
the salary of joe_bloggs is large.
```

```
rule r1_2
if the salary of Person is large
or the job_satisfaction of Person is true
then the professional_contentment of Person becomes true.
```

Inference Networks

- The interdependencies among rules, such as r1_1 and r1_2 define a network
- Inference network shows which facts can be logically combined to form new facts or conclusions
- The facts can be combined using “and”, “or” and “not”.
    - Professional contentment is true if either job satisfaction or large salary is true (or both are true).
    - Job satisfaction is achieved through flexibility, responsibility, and the absence of stress.

An Inference Network



Professional contentment is true if either job satisfaction or large salary is true (or both are true).

Deduction, Abduction and Induction

Rules that make up inference network can be used to link cause and effect:

if <cause> then <effect>

For example:

if

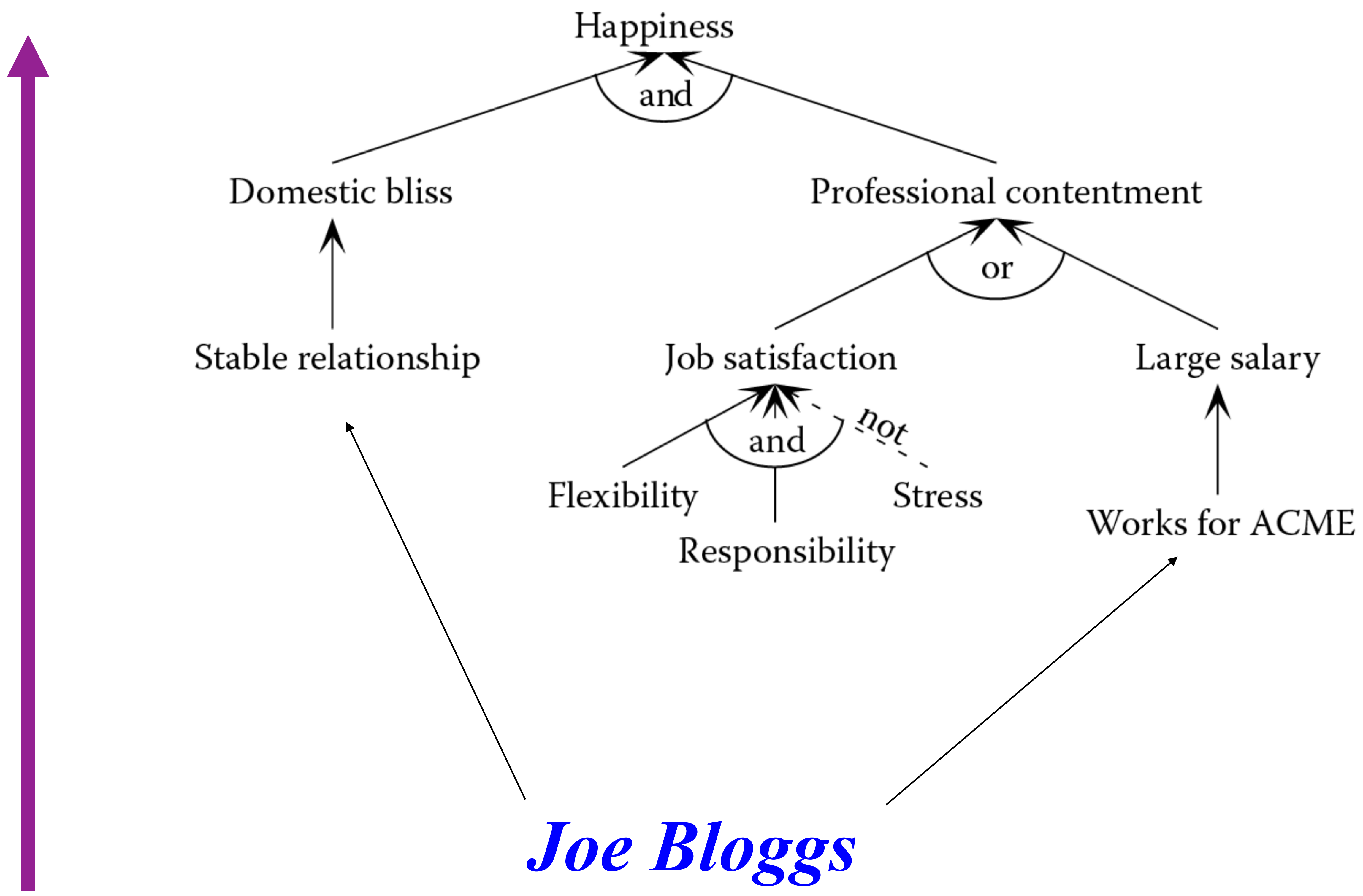
Joe Bloggs works for ACME and is in a stable relationship (the causes),

then

he is happy (the effect).

Deduction, Abduction and Induction

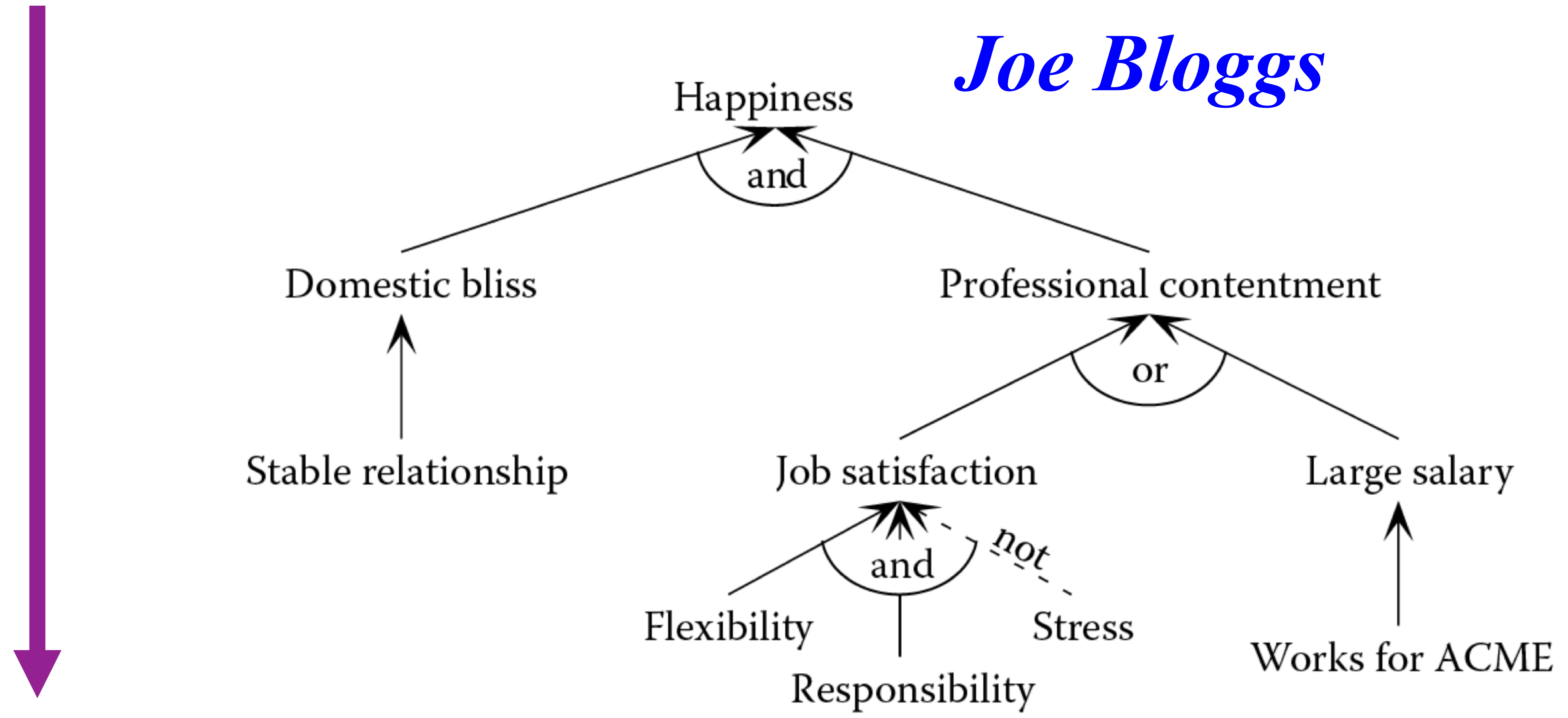
if <cause> then <effect>



if *Joe Bloggs* works for ACME and is in a stable relationship (causes), then he is happy (effect).

Deduction, Abduction and Induction

- Abduction - Many problems, such as diagnosis, involve reasoning in reverse, i.e, find a cause, given an effect.
- Given observation Joe Bloggs is happy, infer by abduction Joe Bloggs enjoys domestic bliss and professional contentment.



Deduction, Abduction and Induction

- If we have many examples of cause and effect, infer the rule that links them.
- For instance: if every employee of ACME earns a large salary, infer:

```
rule r1_1
  if the employer of Person is acme
  then the salary of Person becomes large.
```

- Inferring a rule from a set of examples of cause and effect is induction.

Deduction, Abduction and Induction

- deduction: cause + rule ⇒ effect
- abduction: effect + rule ⇒ cause
- induction: cause + effect ⇒ rule

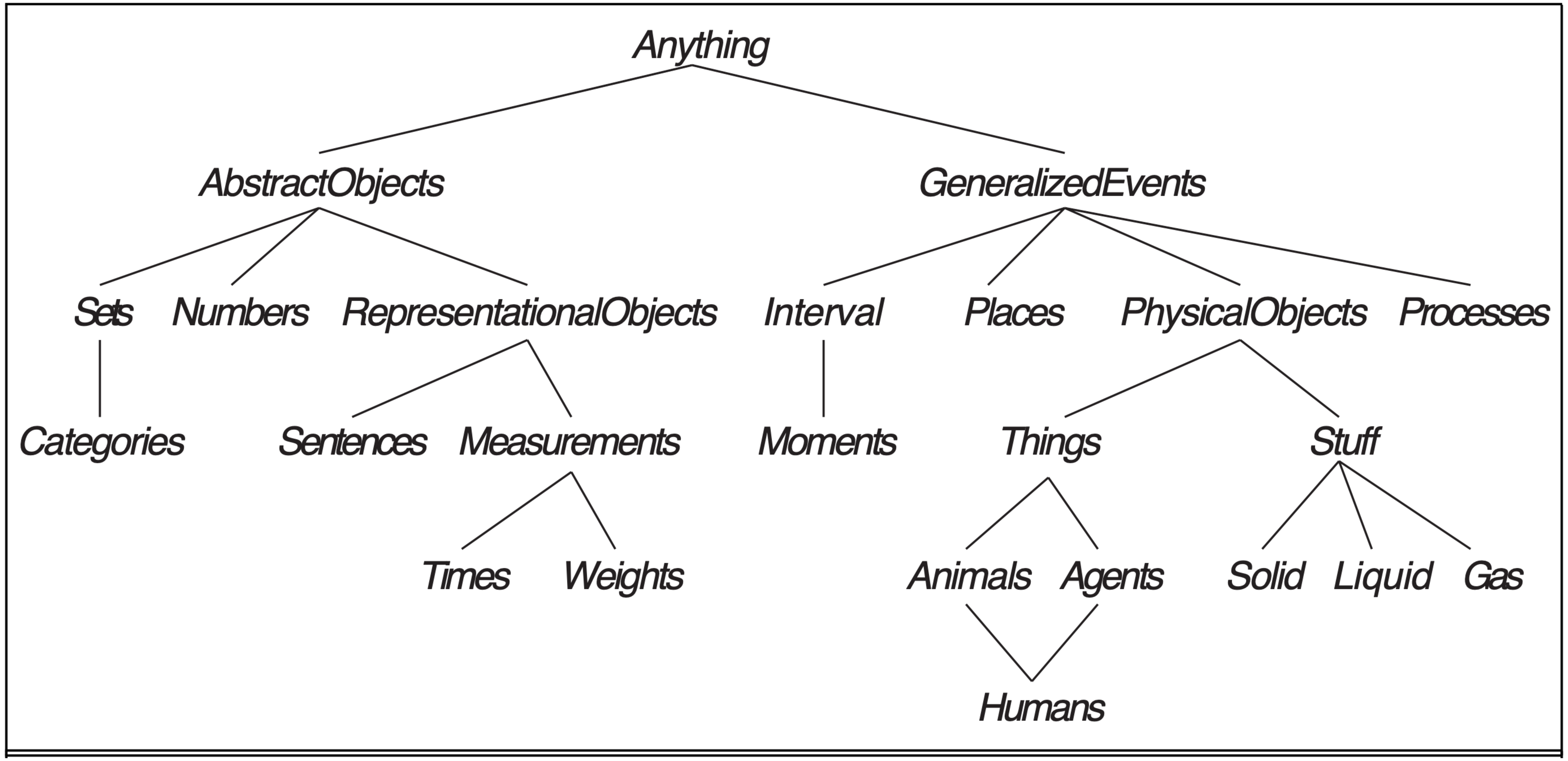
Closed-World Assumption

- Only facts that are in the knowledge base or that can be derived from rules are assumed to be true
- Everything else is false, i.e., if we don’t know it, it’s assumed to be false.
- That’s why it’s more accurate to say:
  - “a proof fails”, instead of “it’s false”
  - “a proof succeeds” instead of, “it’s true”

Ontologies and Ontology Engineering

- An ontology organises everything into a hierarchy of categories.
- Can’t actually write a complete description of everything
  - far too much
  - can leave placeholders where new knowledge can fit in.
  - e.g., define what it means to be a physical object
  - details of different types of objects (robots, televisions, books, …) filled in later
- Similar to object oriented (OO) programming framework

Ontology Example



- Child concept is a specialisation of parent
- Specialisations are not necessarily disjoint (a human is both an animal and an agent)

Categories and Objects

- Organising objects into categories is vital for knowledge representation.
- Interaction with world takes place at level of individual objects, but …
    - much **reasoning takes place at level of categories**
- Categories help make predictions about objects once they are classified

Categories and Objects

- Categories organise and simplify knowledge base through inheritance.
  - if all instances of category Food are edible, and
  - if Fruit is a subclass of Food and Apples is a subclass of Fruit, then
  - infer that every apple is edible.
- Individual apples inherit property of edibility
  - in this case, from membership in the Food category.
- Subclass relations organise categories into a taxonomy, or taxonomic hierarchy

Lecture Overview

- Knowledge representation
- Reasoning
- Uncertainty
- Bayesian inference
- Fuzzy logic

Knowledge Bases

- A knowledge base is a set of sentences in a formal language.
- Declarative approach to building an agent:
  - Tell the system what it needs to know,
    then it can ask itself what it needs to do
  - Answers should follow from the knowledge based.
- How do you formally specify how to answer questions?

Formal Languages
(why not English, or other natural language)?

- Natural languages are ambiguous, for instance:
  - “The fisherman went to the bank”
  - “The boy saw a girl with a telescope”
- Ambiguity makes it difficult to interpret meaning of phrases/sentences
  - But also makes inference harder to define and compute
- Symbolic logic is a syntactically unambiguous language

Reasoning system for categories

- Categories are the building blocks of knowledge representation schemes
- Two closely related families of systems:
  - semantic networks:
    - graphical aids for visualizing a knowledge base
    - efficient algorithms for inferring properties of object based in category membership
  - description logics:
    - formal language for constructing and combining category definitions
    - efficient algorithms for deciding subset and superset relationships between categories

Semantic Networks

- Facts, Objects, Attributes and Relationships
    - Relationships exist among instances of objects and classes of objects.
- Attributes and relationships can be represented as a network, known as an associative network or semantic network
- We can build a model of the subject area of interest

Example – A simple set of statements

- <u>My car</u> is a car
- A car is a vehicle
- A car has four wheels
- A car’s speed is 0 mph
- <u>My car</u> is red
- <u>My car</u> is in <u>my garage</u>
- <u>My garage</u> is a garage
- A garage is a building
- <u>My garage</u> is made from brick
- <u>My car</u> is in <u>High Street</u>
- <u>High Street</u> is a street
- A street is a road

Underline = object (instance)

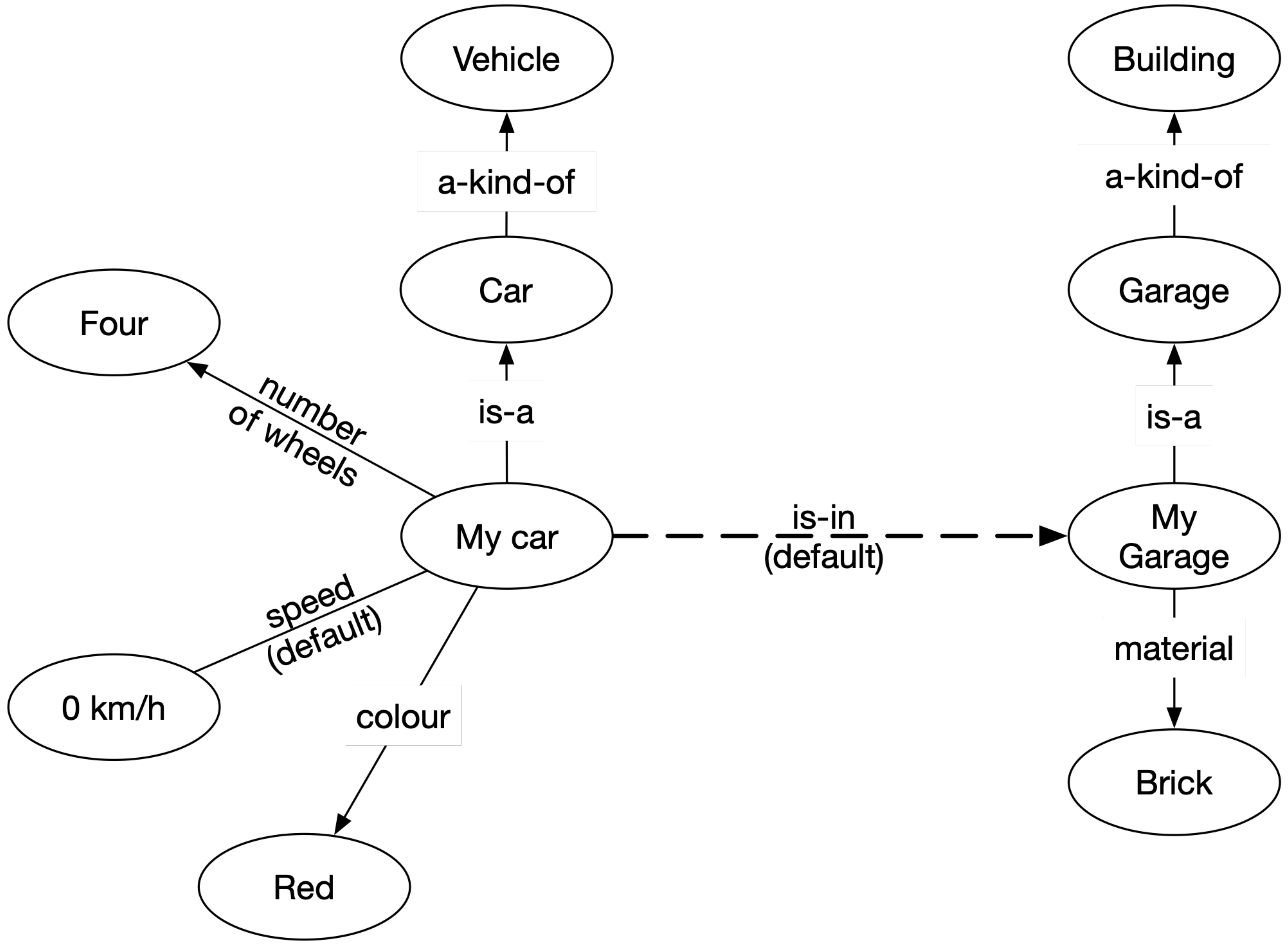
Example – facts, objects and relations

- My car is a **car**
- A car is a **vehicle**
- A car has four wheels
- A car's speed is 0 mph
- My car is red
- My car is in my garage
- My garage is a **garage**
- A garage is a **building**
- My garage is made from brick
- My car is in High Street
- High Street is a **street**
- A street is a **road**

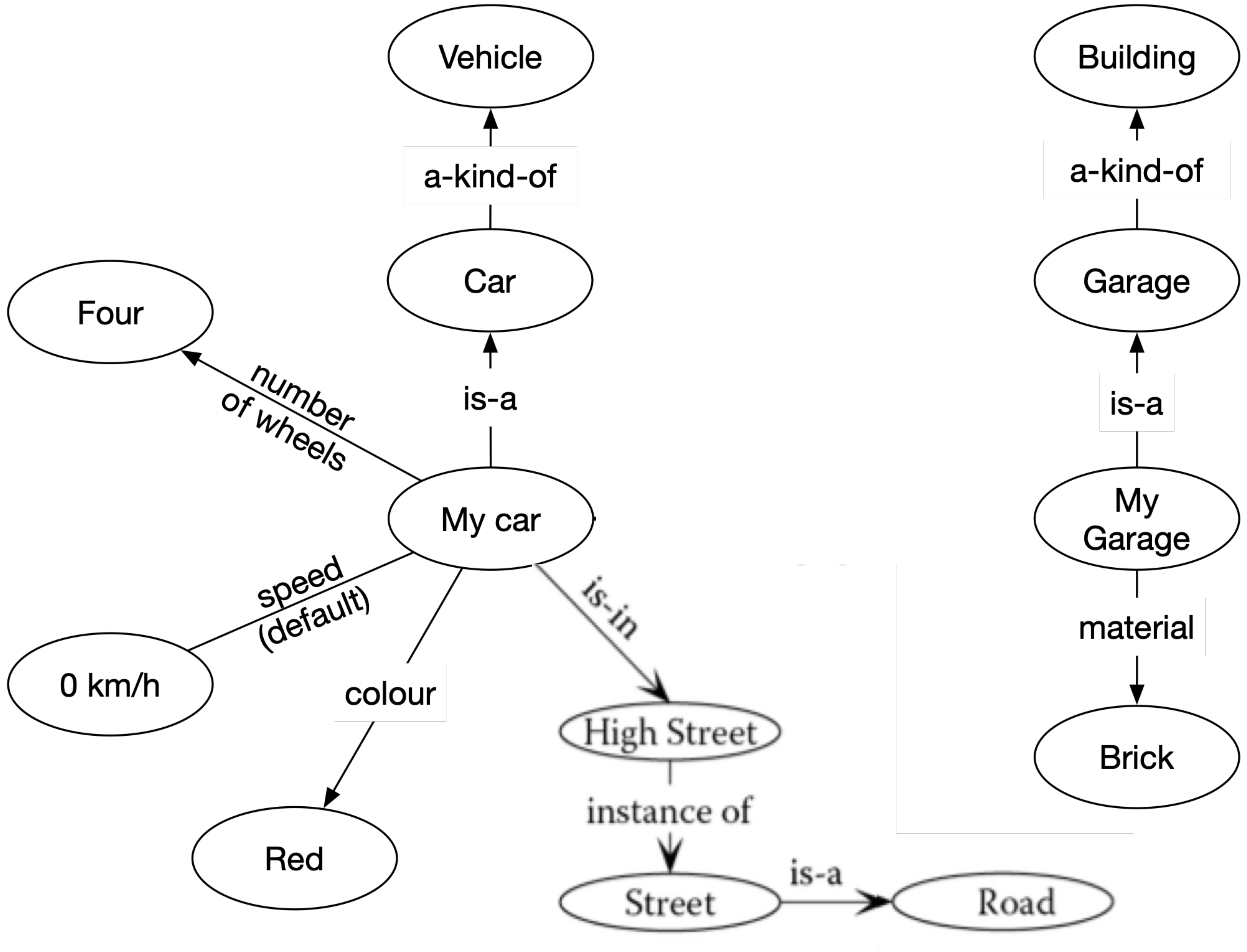
Example – facts, objects and relations

- My car is a **car** (static relationship)
- A car is a **vehicle** (static relationship)
- A car has four wheels (static attribute)
- A car's speed is 0 mph (default attribute)
- My car is red (static attribute)
- My car is in my garage (default relationship)
- My garage is a **garage** (static relationship)
- A garage is a **building** (static relationship)
- My garage is made from brick (static attribute)
- My car is in High Street (transient relationship)
- High Street is a **street** (static relationship)
- A street is a **road** (static relationship)

A semantic network (with a default)



A semantic network (with a default)



Classes and Instances

- Distinction between object instances and classes of objects:
  - Car and vehicle are both classes of objects
    - Linked by “ako” relation (a-kind-of)
    - Direction of arrow indicates “car is a vehicle” and not “vehicle is a car”
- *My car* is a unique entity.
  - Relationship between *my car* and *car* is “isa” (is an instance of)

Propositional Logic

- Propositions are entities (facts or non-facts) that can be true or false

Examples:

  - “The sky is blue” - the sky is blue (here and now).
  - “Socrates is bald” (assumes ‘Socrates’, ‘bald’ are well defined)
    “The car is red” (requires ‘the car’ to be identified)
  - “Socrates is bald and the car is red” (complex proposition)

- Use single letters to represent propositions, e.g. *P*: Socrates is bald
- Reasoning is independent of definitions of propositions

Propositional Logic

- Letters stand for "basic" propositions
- Combine into more complex sentences using operators not, and, or, implies, iff
- Propositional connectives:

$\neg$	negation	$\neg P$	"not P"
$\wedge$	conjunction	$P \wedge Q$	"P and Q"
$\vee$	disjunction	$P \vee Q$	"P or Q"
$\rightarrow$	implication	$P \rightarrow Q$	"If P then Q"
$\leftrightarrow$	bi-implication	$P \leftrightarrow Q$	"P if and only if Q"

From English to Propositional Logic

- "It is not the case that the sky is blue": $\neg B$
(alternatively "the sky is not blue")
- "The sky is blue and the grass is green": $B \wedge G$
- "Either the sky is blue or the grass is green": $B \vee G$
- "If the sky is blue, then the grass is not green": $B \rightarrow \neg G$
- "The sky is blue if and only if the grass is green": $B \leftrightarrow G$
- "If the sky is blue, then if the grass is not green, the plants will not grow":
$B \rightarrow (\neg G \rightarrow \neg P)$

First-Order Logic

- Propositional logic has limited expressive power
  - Cannot express relations and ontologies
- First-Order Logic can express knowledge about objects, properties and relationships between objects
- Syntax
  - Constant symbols: a, b, . . . , *Mary* (objects)
  - Variables: x, y, . . .
  - Function symbols: f, *mother_of*, *sine*, . . .
  - Predicate symbols: *Mother*, *likes*, . . .
  - Quantifiers: $\forall$ (universal); $\exists$ (existential)

Language of First-Order Logic

- Terms: constants, variables, functions applied to terms (refer to objects)
  - e.g. a, $f(a)$, $mother\_of(Mary)$, $\ldots$
- Atomic formulae: predicates applied to tuples of terms
  - e.g. $likes(Mary, mother\_of(Mary))$, $likes(x, a)$
- Quantified formulae:
  - e.g. $\forall x\ likes(x, a)$, $\exists x\ likes(x, mother\_of(y))$
  - Second occurrences of x are bound by quantifier ($\forall$ in first case, $\exists$ in second) and y in the second formula is free

From English to First-Order Logic

- Everyone likes lying on the beach — $\forall x\ likes\_lying\_on\_beach(x)$
- Someone likes Fido — $\exists x\ likes(x, Fido)$
- No one likes Fido — $\neg(\exists x\ likes(x, Fido))$ (or $\forall x\ \neg likes(x, Fido)$)
- Fido doesn't like everyone — $\neg\forall x\ likes(Fido, x)$
- All cats are mammals — $\forall x(cat(x) \rightarrow mammal(x))$
- Some mammals are carnivorous — $\exists x(mammal(x) \wedge carnivorous(x))$
- Note: $\forall x A(x) \Leftrightarrow \neg\exists x \neg A(x)$, $\exists x A(x) \Leftrightarrow \neg\forall x \neg A(x)$

Lecture Overview

- Knowledge representation
- Reasoning
- Uncertainty
- Bayesian inference
- Fuzzy logic

Uncertainty

- Classic logic and reasoning assume events either occur or do not occur, but not necessarily the same in real world.
- For instance, in weather prediction, probabilities can be used.
- People can cope well with uncertainty in complex situations assuming default or common values when precise data is unknown.
- However, the more data is missing, the more guesses have to be made, the lower the quality of the final reasoning.

Uncertainty

- A system designed to emulate the human reasoning process needs to generate and rank several potential solutions.
- Uncertainty might be expressed as confidence in which continuous values are allowed.
- Uncertainty can be expressed both in the information and in the reasoning process.
- The conclusion of a rule cannot be guaranteed.
  - For example: high blood pressure (HBP) increases the chance of a heart attack, but not everyone having HBP will have a heart attack.

Uncertainty

- Rules can express many types of knowledge
- But how can *uncertainty* be handled?
- Uncertainty may arise from:
  - Uncertain evidence (Not certain that Joe Bloggs works for ACME.)
  - Uncertain link between evidence and conclusion.
    (Cannot be certain that ACME employee earns a large salary, just likely.)
  - Vague rule. (What is a "large"?)

Confidence factors

Fuzzy Logic

Bayesian inference

Confidence Factors

- Uncertainty in antecedents: based on user information and deduced from another rule.
- Uncertainty in a rule: based on expert's rule confidence and propagated to the conclusion.
- For example, a simple rule: if A is true, then B is true (A => B)
- If we are uncertain about A, we are uncertain about B:

A	=>	B
0.8		0.8

Confidence Factors

- However, there might be uncertainty about the validity of the rule. For instance:

0.8

A => B

0.8 ?

- Reasoning with confident factors: two independent pieces of evidence should increase confidence. Then the rule is inverted, for instance:

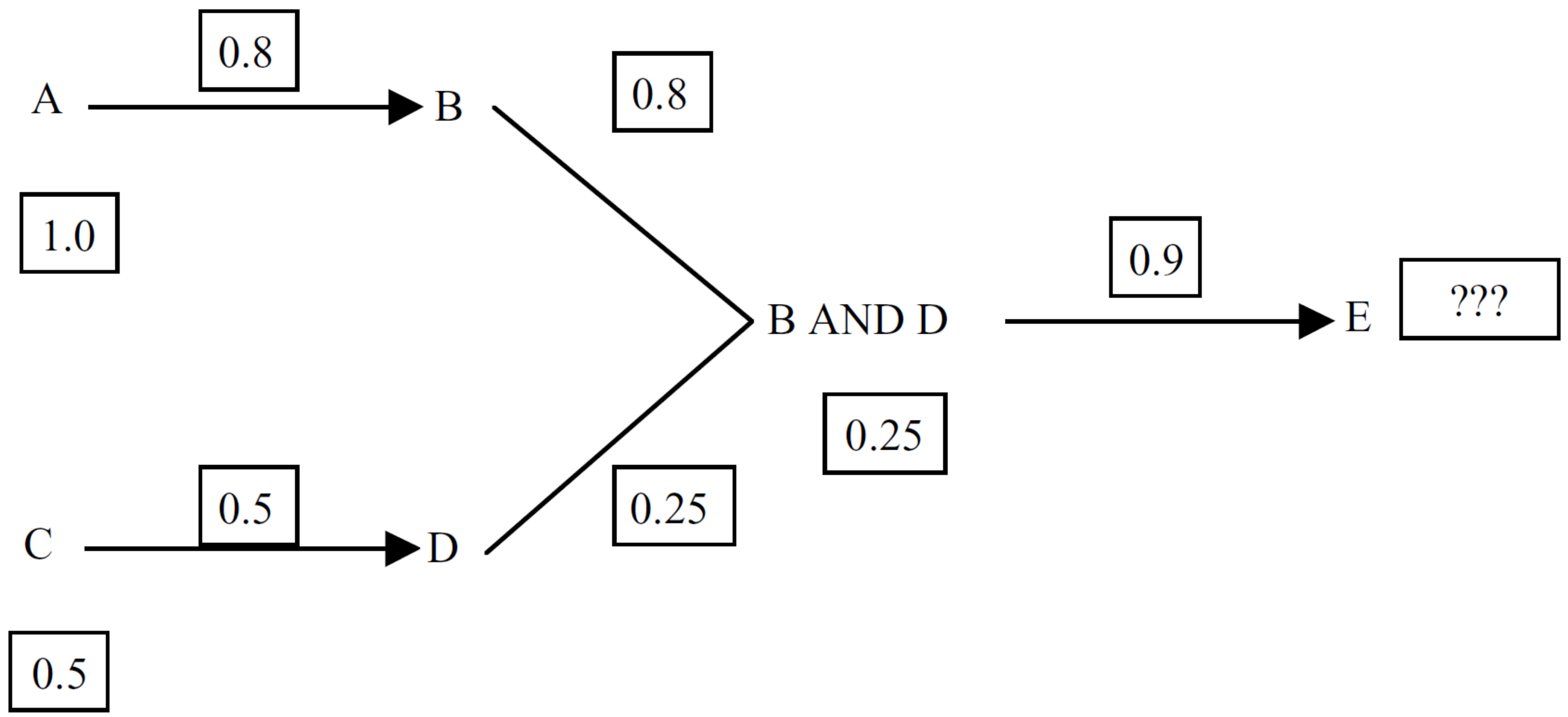
0.8 0.8

A => C B => C

(1 – 0.8)x(1 – 0.8) = 0.04 then C is true with 96% confidence.

Confidence Factors

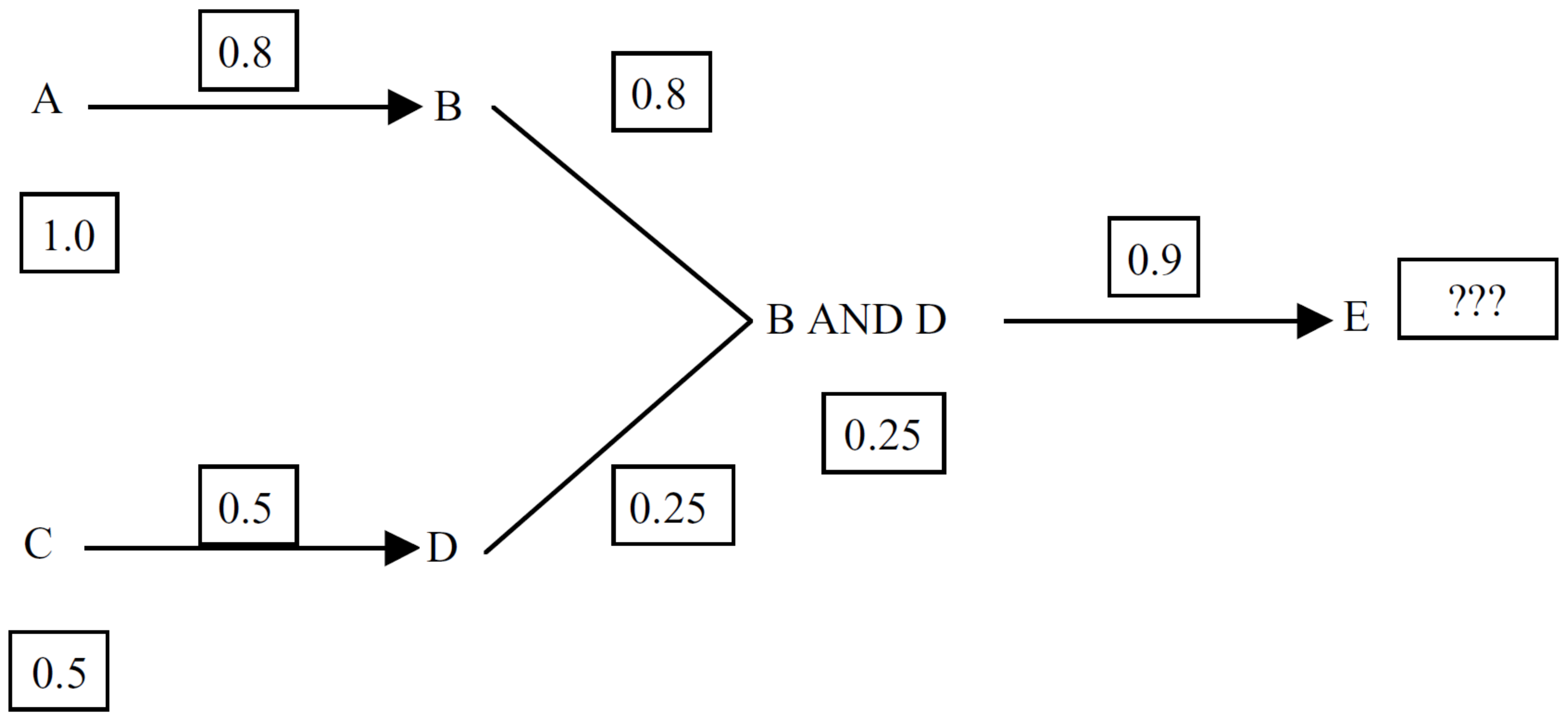
- Inference network: sequence of relationships between facts and confidence factors.



- CF(B) = CF(A) × CF(IF A THEN B) = 1 × 0.8 = 0.8
- CF(D) = CF(C) × CF(IF C THEN D) = 0.5 × 0.5 = 0.25
- CF(B&D) = min (CF(B),CF(D)) = min (0.8, 0.25) = 0.25
- CF(E) = CF(B&D) × CF (IF B&D THEN E) = ?

Confidence Factors

- Inference network: sequence of relationships between facts and confidence factors.



- CF(B) = CF(A) × CF(IF A THEN B) = 1 × 0.8 = 0.8
- CF(D) = CF(C) × CF(IF C THEN D) = 0.5 × 0.5 = 0.25
- CF(B&D) = min (CF(B),CF(D)) = min (0.8, 0.25) = 0.25
- CF(E) = CF(B&D) × CF (IF B&D THEN E) = 0.225

Lecture Overview

- Knowledge representation
- Reasoning
- Uncertainty
- Bayesian inference
- Fuzzy logic

Probabilistic Inference

- It is not always possible to create a complete, consistent model of the world.
- Probability theory will serve as the formal language for representing and reasoning with uncertain knowledge.
- Bayes theorem allows computing a conditional probability.
- It is useful to calculate the probability of events where intuition often fails.
- Widely used in machine learning. For instance, classification predictive modelling problems such as the Bayes Optimal Classifier and Naive Bayes.

Probabilistic Inference

- For each primitive proposition or event, attach a degree of belief to the sentence.
- For proposition A, probability $0 \leq P(A) \leq 1$.
  - If A is true, $P(A) = 1$.
  - If A is false, $P(A) = 0$.
- Proposition A is either true or false. But $P(A)$ is the degree of belief in A being true or false.

Probabilistic Inference

- Boolean variables abbreviated, P(A=true) = P(A) and P(A=false) = P(~A)
- Examples:
- P(Weather = sunny) = 0.6 (we believe weather will be sunny with 60% certainty – from {sunny, rainy, snowy, cloudy}).
- P(Cavity = true) = 0.05 (we believe there is a 5% possibility that a person has a cavity).
- P(A=a ^ B=b) = P(A=a, B=b) = 0.2, with A=Mood, a=happy, B=Weather, and b=rainy (we believe there is a 20% chance that when it is raining my mood is happy).

Probabilistic Inference

- Axioms:
  - $0 \leq P(A=a) \leq 1$
  - P(True) = 1, P(False) = 0
  - P(A v B) = P(A) + P(B) - P(A ^ B)
- Properties:
  - P(~A) = 1 - P(A)
  - P(A) = P(A ^ B) + P(A ^ ~B)
  - Sum{P(A=a)} = 1, where the sum is over all possible values a in the sample space of A

Joint Probability Distribution

- Probability of two (or more) simultaneous events, often described in terms of events A and B from two dependent random variables, e.g., X and Y.

  P(A and B) or P(A ^ B) or P(A, B).

- The joint probability is symmetrical: P(A, B) = P(B, A).
- Full joint probability distribution assigns probabilities to **all possible combinations**.

Joint Probability Distribution

- For n Boolean variables, rows in the table 2^n.
- For k possible values, rows in the table k^n.
- Sum in the right column must be 1. For n Booleans, we need to know 2^n-1 other values.
- If we know all the values, we can compute any probability in the domain:
  - P(Bird=T) = P(B) = 0.0 + 0.2 + 0.04 + 0.01 = 0.25
  - P(Bird=T, Flier=F) = P(B, ~F) = P(B, ~F, Y) + F(B, ~F, ~Y) = 0.04 + 0.01 = 0.05

Bird	Flier	Young	Probability
T	T	T	0.0
T	T	F	0.2
T	F	T	0.04
T	F	F	0.01
F	T	T	0.01
F	T	F	0.01
F	F	T	0.23
F	F	F	0.5

Conditional Probability

- Probability of one (or more) event given the occurrence of another event, denoted P(A | B).
- Conditional probabilities are key for reasoning as they accumulate evidence.
- P(A | B) = 1 is equivalent to B => A in propositional logic. Thus, P(A | B) = 0.9 is B => A with 90% certainty.
- The conditional probability is defined as:

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

- Joint probability is symmetrical: P(A, B) = P(B, A).
    - However, conditional probability is not: P(A | B) != P(B | A)
- Chain Rule: P(A,B,C,D) = P(A|B,C,D)P(B|C,D)P(C|D)P(D)

Conditional Probability

- Example: P(~Bird | Flier)

  $$P(\sim B|F) = P(\sim B,F) / P(F)$$
  $$= (P(\sim B,F,Y) + P(\sim B,F,\sim Y)) / P(F)$$
  $$= (.01 + .01)/P(F)$$

- $P(\sim B|F) = 0.02/0.22 = 0.091$
- This is intractable as it means that we must compute and store the full joint probability distribution table.

Bird	**Flier**	**Young**	**Probability**
T	T	T	0.0
T	T	F	0.2
T	F	T	0.04
T	F	F	0.01
F	T	T	0.01
F	T	F	0.01
F	F	T	0.23
F	F	F	0.5

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

Bayes' Rule

- Bayes theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\tilde{}A)P(\tilde{}A)}$$

- P(A|B) is referred to as the posterior probability
- P(A) is referred to as the prior probability.
- P(B|A) is referred to as the likelihood.
- P(B) is referred to as the evidence.

Bayes' Rule

- Often, we want to know P(A|B) but we only have access to P(B|A).
- Example: If S represents a given patient has a stiff neck and M the patient has meningitis.
- The doctor and patient may like to know P(M|S), but from the general population is difficult.
- Doctors may be able to accumulate statistics that define P(S|M).
- Then, if P(M) = 1/50,000, P(S) = 1/20, and P(S|M) =1/2. Using Bayes' Rule P(M|S) = 1/5000 = 0.0002 or 0.02%

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayes' Rule

- Base rates of women having breast cancer and having no breast cancer are 0.02% and 99.98% respectively. The true positive rate or sensitivity P(positive mammography | breast cancer) = 85% and the true negative or specificity P(negative mammography | ~breast cancer) = 95%. Compute P(C | M).

P(C | M) = P(M | C) * P(C) / P(M)

P(C | M) = 0.85 * 0.0002 / P(M)

From Bayes' rule: P(B) = P(B|A) * P(A) + P(B|~A) * P(~A)

P(M) = P(M|C) * P(C) + P(M|~C) * P(~C)

P(M) = 0.85 * 0.0002 + 0.05 * 0.9998 = 0.05016

P(C | M) = 0.85 * 0.0002 / 0.05016 = 0.00339 or 0.34%

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\tilde{\ }A)P(\tilde{\ }A)}$$

Belief Networks

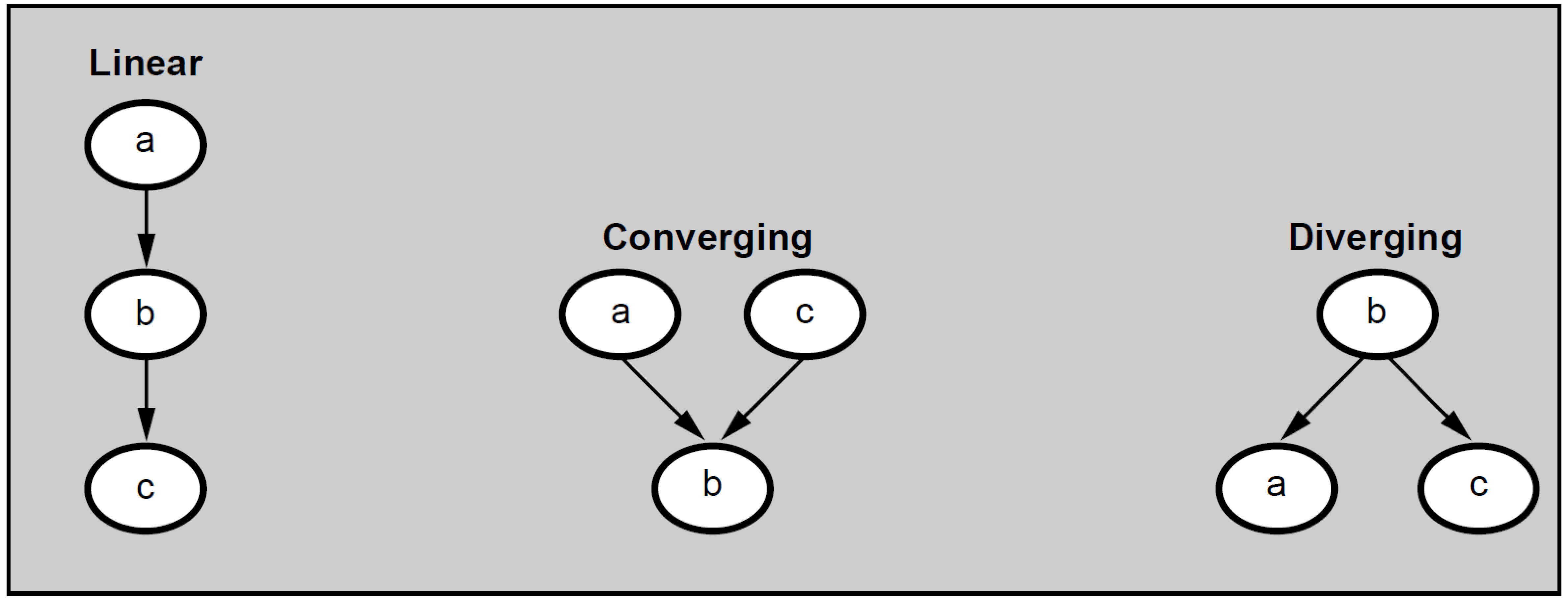
- Bayesian network, Bayes nets, causal nets.
- Space-efficient data structure for full joint probability distribution.
- Represent causal relations: arcs from cause variables to immediate effects.
    - Predictive reasoning from causes to effects (top-down).
    - Diagnostic reasoning from effects to causes (bottom-up).

Belief Networks

- Useful in situations in which causality plays a role but our understanding is incomplete.
- Formally, it's a directed, acyclic graph (DAG)
  - Include a node for each variable and a directed arc from A to B if A is a direct causal influence on B.
- For a probability distribution, we need only to give the prior probabilities for root nodes and conditional probabilities for non-root nodes considering all combinations from the direct predecessors.

Belief Networks

- Topology: Three connection types.
- Variables are true or false and certain independence assumptions hold.
- For instance in diverging: A and C are conditionally independent given A → P(C|B,A) = P(C|B) and symmetrically P(A|B,C) = P(A|B).



Belief Network Example

- Consider the problem domain in which I go home, and I want to know if someone from my family is home before I go in. Let's say I know the following information:

    (1) When my wife leaves the house, she often (but not always) turns on the outside light.

    (2) When nobody is home, the dog is often left outside.

    (3) If the dog has bowel troubles, it is also often left outside.

    (4) If the dog is outside, I will probably hear it barking (though it might not bark, or I might hear a different dog barking and think it's my dog).

Belief Network Example

- Given the previous information, we can consider the following five Boolean random variables:
  - family-out (*fo*): everyone is out of the house.
  - light-on (*lo*): the light is on.
  - dog-out (*do*): the dog is outside.
  - bowel-problem (*bp*): the dog has bowel troubles.
  - hear-bark (*hb*): I can hear the dog barking.

Belief Network Example

- From this information, the following direct causal influences seem appropriate:

1. *hb* is only directly influenced by *do*. Hence *hb* is conditionally independent of *lo*, *fo* and *bp* given *do*.

2. *do* is only directly influenced by *fo* and *bp*. Hence *do* is conditionally independent of *lo* given *fo* and *bp*.

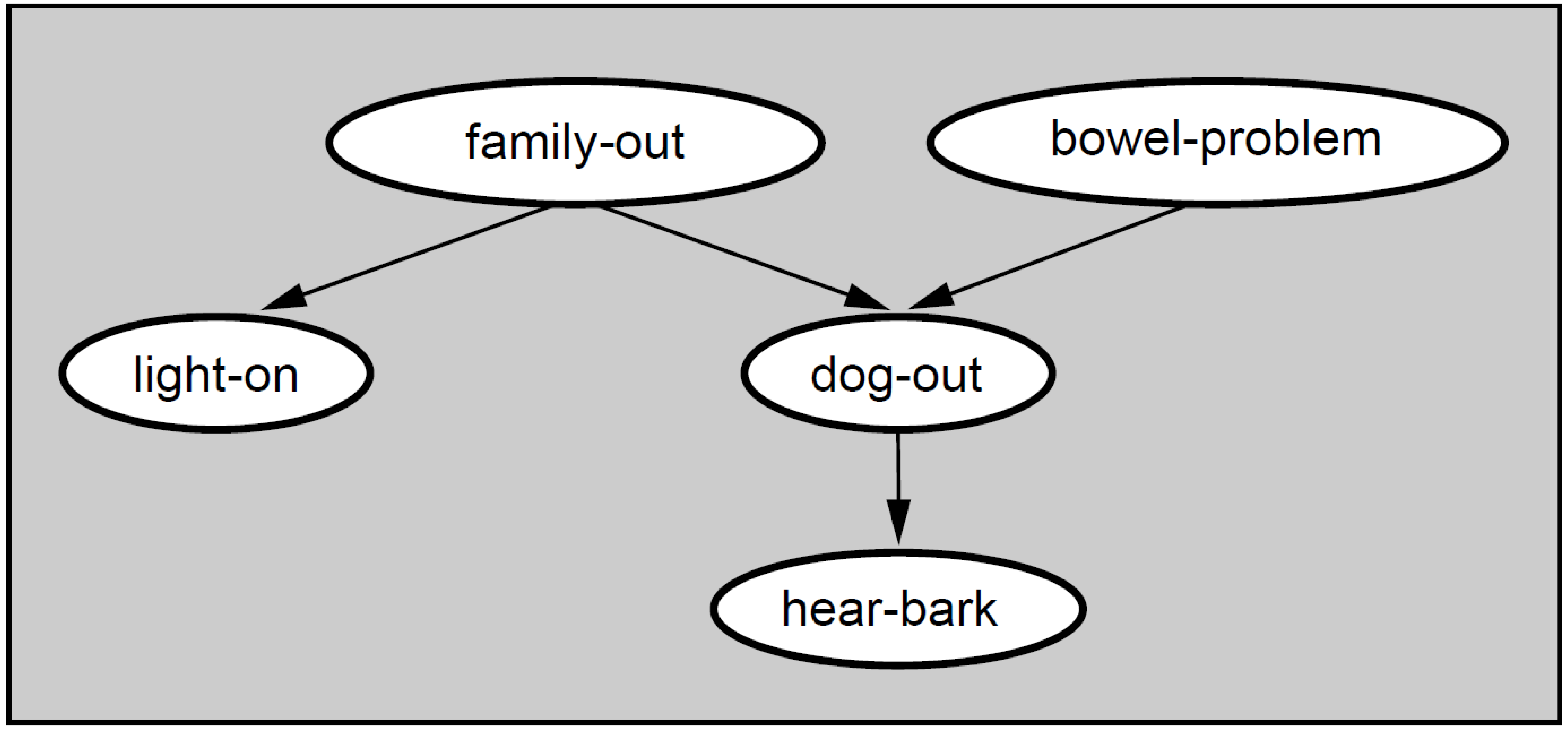
3. *lo* is only directly influenced by *fo*. Hence *lo* is conditionally independent of *do*, *hb* and *bp* given *fo*.

4. *fo* and *bp* are independent.

- family-out (*fo*): everyone is out of the house.
- light-on (*lo*): the light is on.
- dog-out (*do*): the dog is outside.
- bowel-problem (*bp*): the dog has bowel troubles.
- hear-bark (*hb*): I can hear the dog barking.

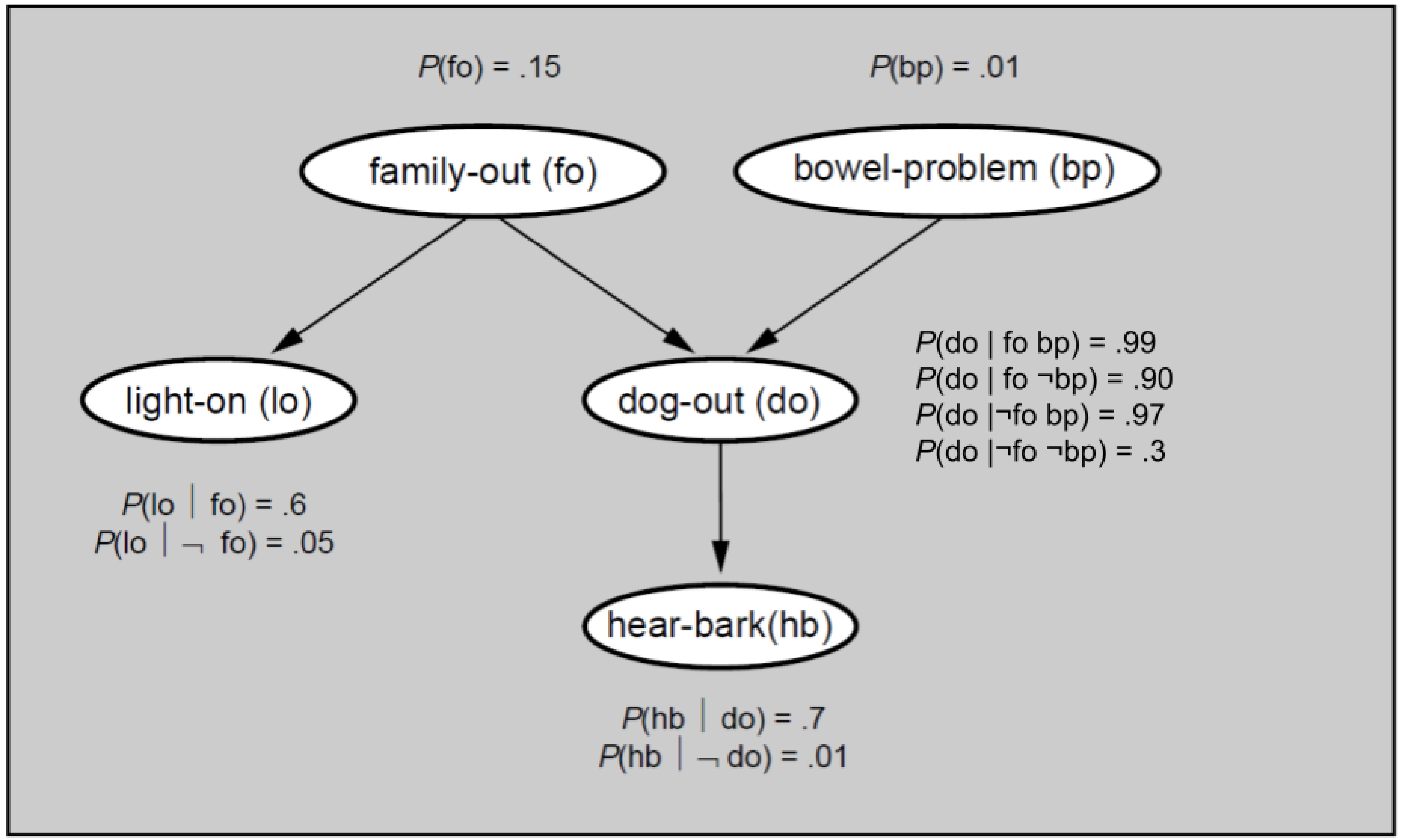
Belief Network Example

- Belief network representing these direct causal relationships (though these causal connections are not absolute, i.e., they are not implications):

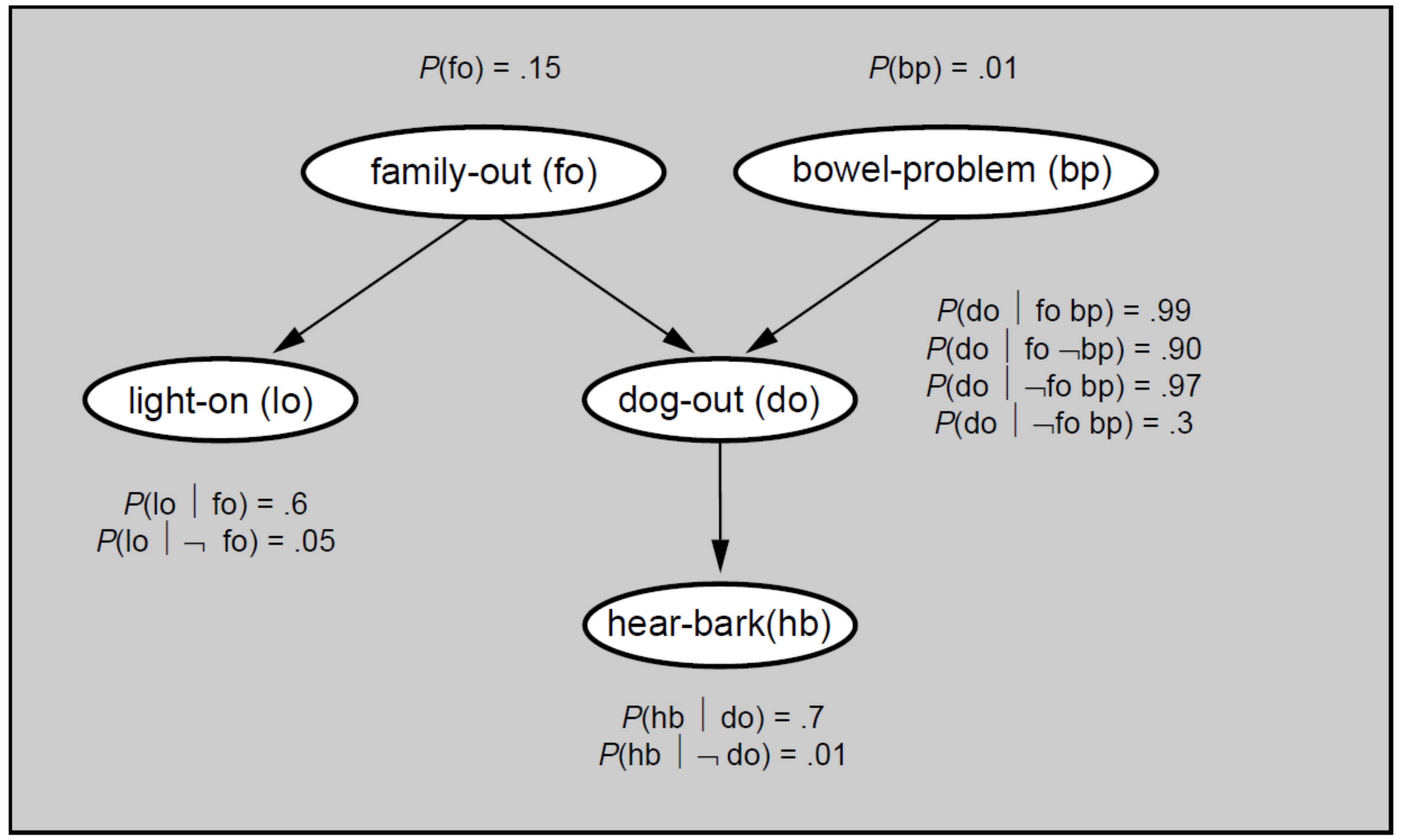


Belief Network Example

- For each root node, the prior probability of the random variable needs to be determined.
- For each non-root node, the conditional probabilities of the node's variable given all possible combinations of its immediate parent nodes need to be determined.
- In this example, a total of 10 probabilities are computed and stored in the net (not 32!)
    - The reduction is due to the conditional independence of many variables.



Computing Joint Probability



- Example: Compute P(BP,~FO,DO,~LO,HB)

P(BP,~FO,DO,~LO,HB) = P(HB,~LO,DO,~FO,BP)

= P(HB | ~LO,DO,~FO,BP) * P(~LO,DO,~FO,BP) by Product Rule

= P(HB|DO) * P(~LO,DO,~FO,BP) by Conditional Independence of HB and LO,FO, and BP given DO

= P(HB|DO) P(~LO | DO,~FO,BP) P(DO,~FO,BP) by Product Rule

= P(HB|DO) P(~LO|~FO) P(DO,~FO,BP) by Conditional Independence of LO and DO, and LO and BP, given FO

= P(HB|DO) P(~LO|~FO) P(DO | ~FO,BP) P(~FO,BP) by Product Rule

= P(HB|DO) P(~LO|~FO) P(DO|~FO,BP) P(~FO | BP) P(BP) by Product Rule

= P(HB|DO) P(~LO|~FO) P(DO|~FO,BP) P(~FO) P(BP) by Independence of FO and BP

= (.7)(1 - .05)(.97)(1 - .15)(.01) = 0.005483

All values are available directly in the network (since P(~A|B) = 1 - P(A|B))

Lecture Overview

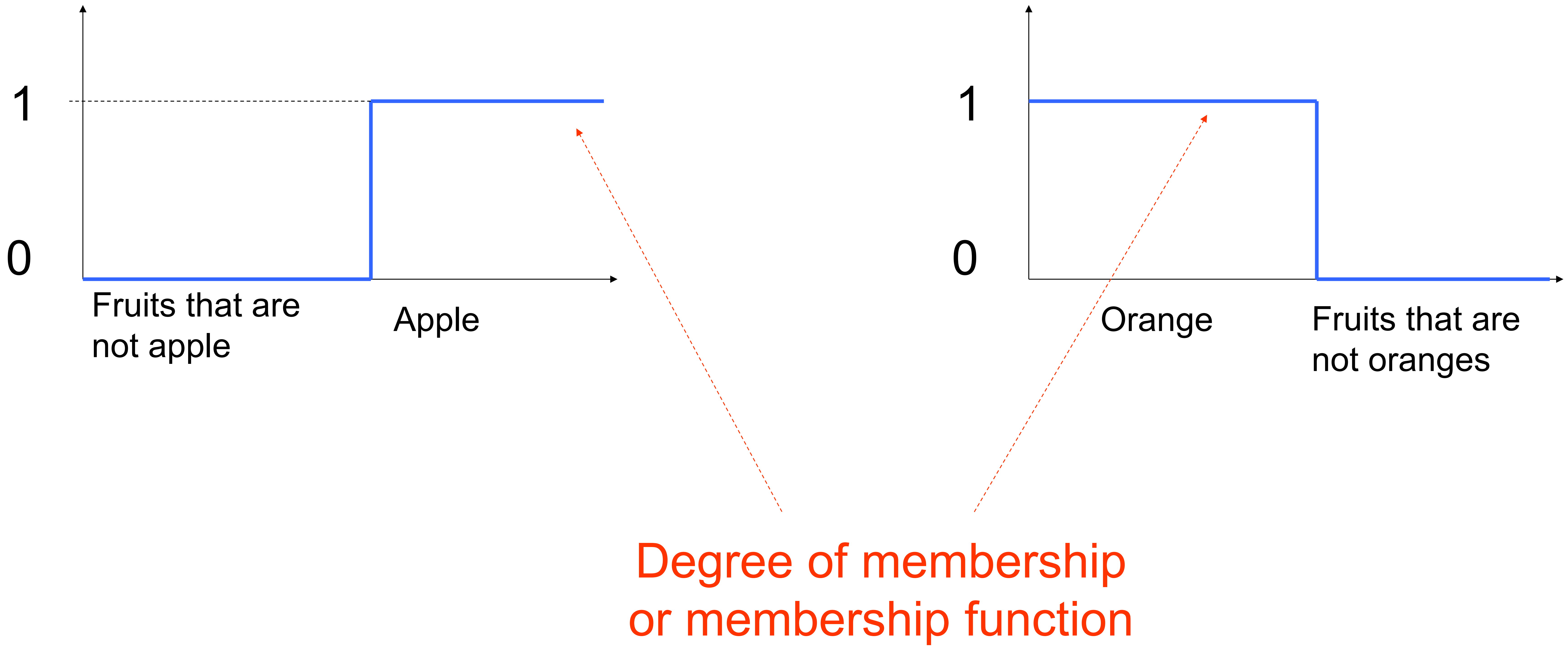
- Knowledge representation
- Reasoning
- Uncertainty
- Bayesian inference
- Fuzzy logic

Fuzzy Logic

- Crisp or Boolean logic uses only two binary values: true, false.
- Fuzzy logic is an extension of classic logic and is based on the idea that at a given moment, it is not possible to precisely determine the value of a variable X.
- In fuzzy logic, one can only know the degree of membership in each of the sets that have participated in defining the range of variation of the variable.
- For example: low, medium, or high temperature.

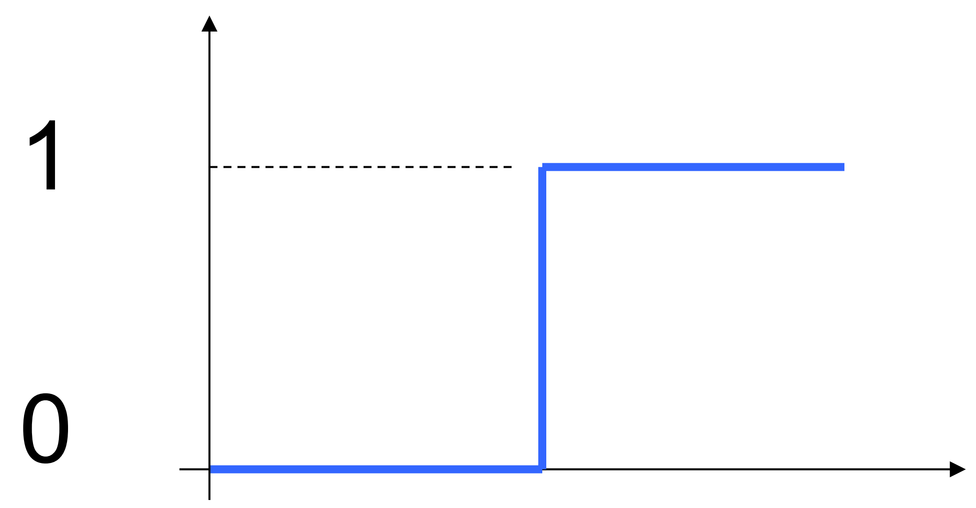
Fuzzy Logic

- Boolean logic. Set of fruits: Apple|Fruit, Orange|Fruit

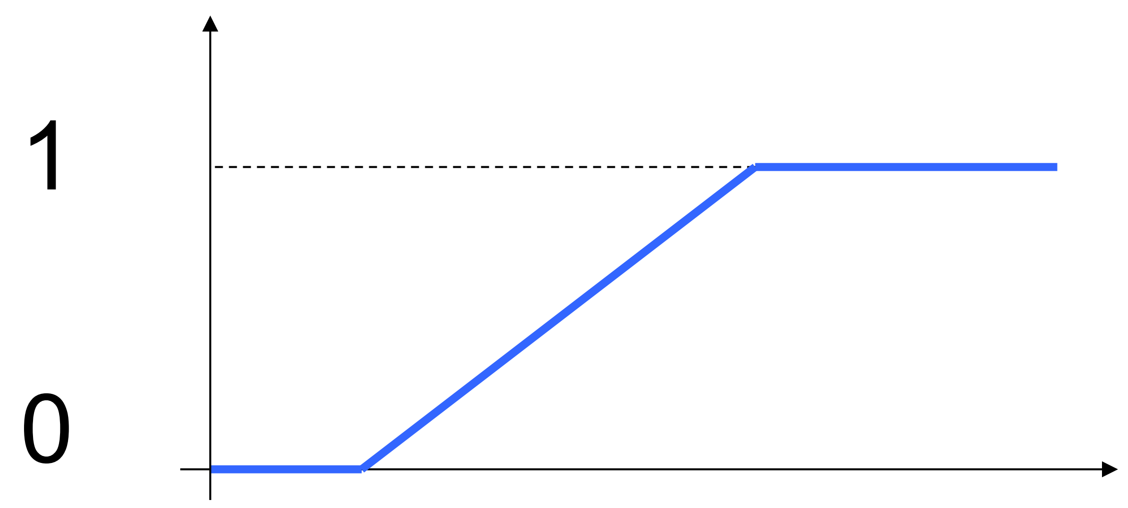


Fuzzy Logic

- Fuzzy logic is an extension of Boolean logic to handle the concept of partial truth when truth values lie between "absolutely true" and "absolutely false."

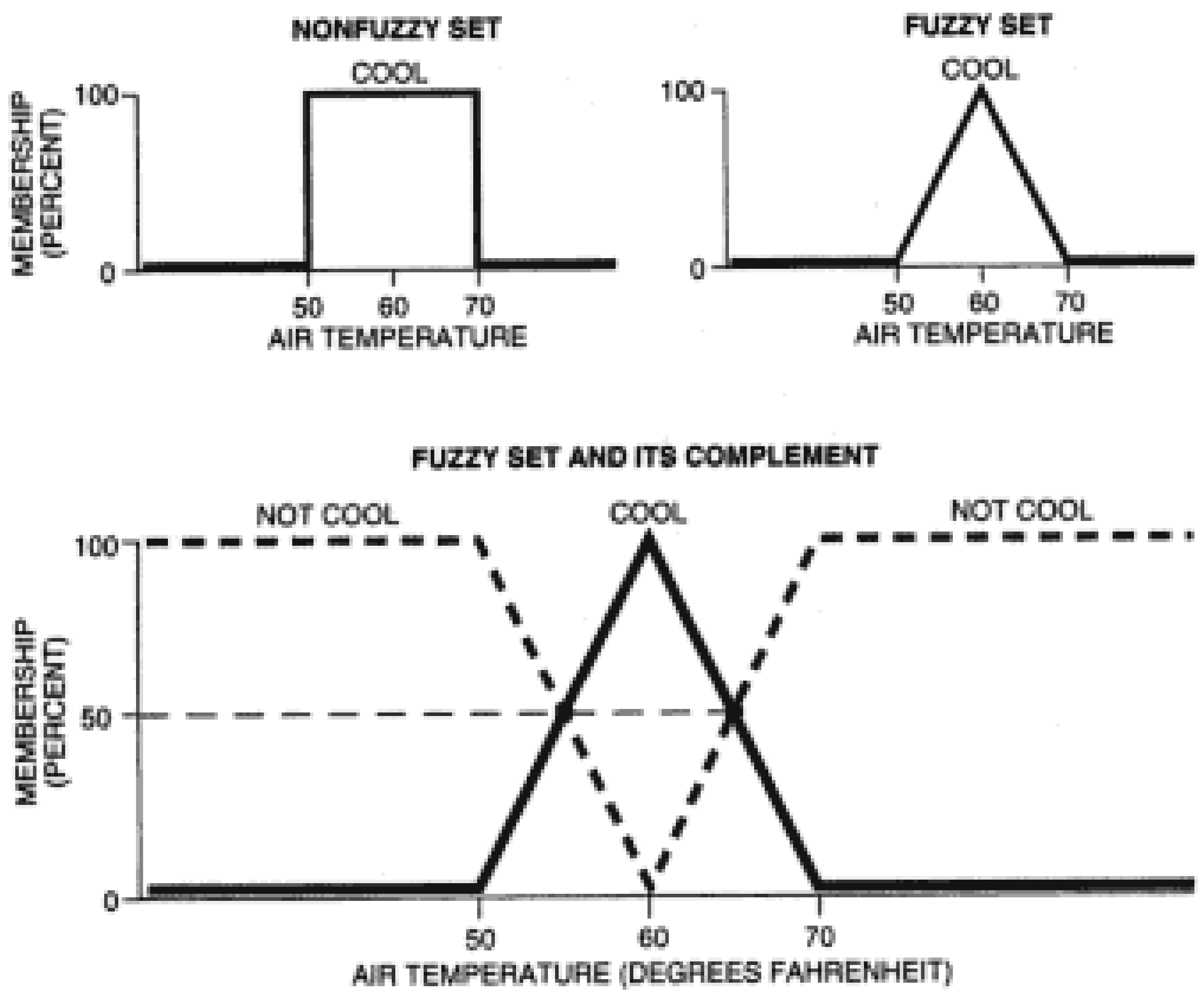


Boolean logic



Fuzzy logic

Fuzzy Logic


NONFUZZY SET
COOL
MEMBERSHIP (PERCENT)
100
0
50 60 70
AIR TEMPERATURE
FUZZY SET
COOL
100
0
50 60 70
AIR TEMPERATURE
FUZZY SET AND ITS COMPLEMENT
NOT COOL
COOL
NOT COOL
100
50
0
MEMBERSHIP (PERCENT)
50 60 70
AIR TEMPERATURE (DEGREES FAHRENHEIT)

Ambiguity

- Ambiguity is related to the degree to which events occur regardless of the probability of their occurrence.
- For example, the degree of youth in a person is a fuzzy event regardless of being a random variable.
- Probabilistic uncertainty dissipates with an increase in the number of occurrences, whereas fuzziness does not.
- If an event occurs, it is random. The degree to which it occurs is fuzzy.

Ambiguity

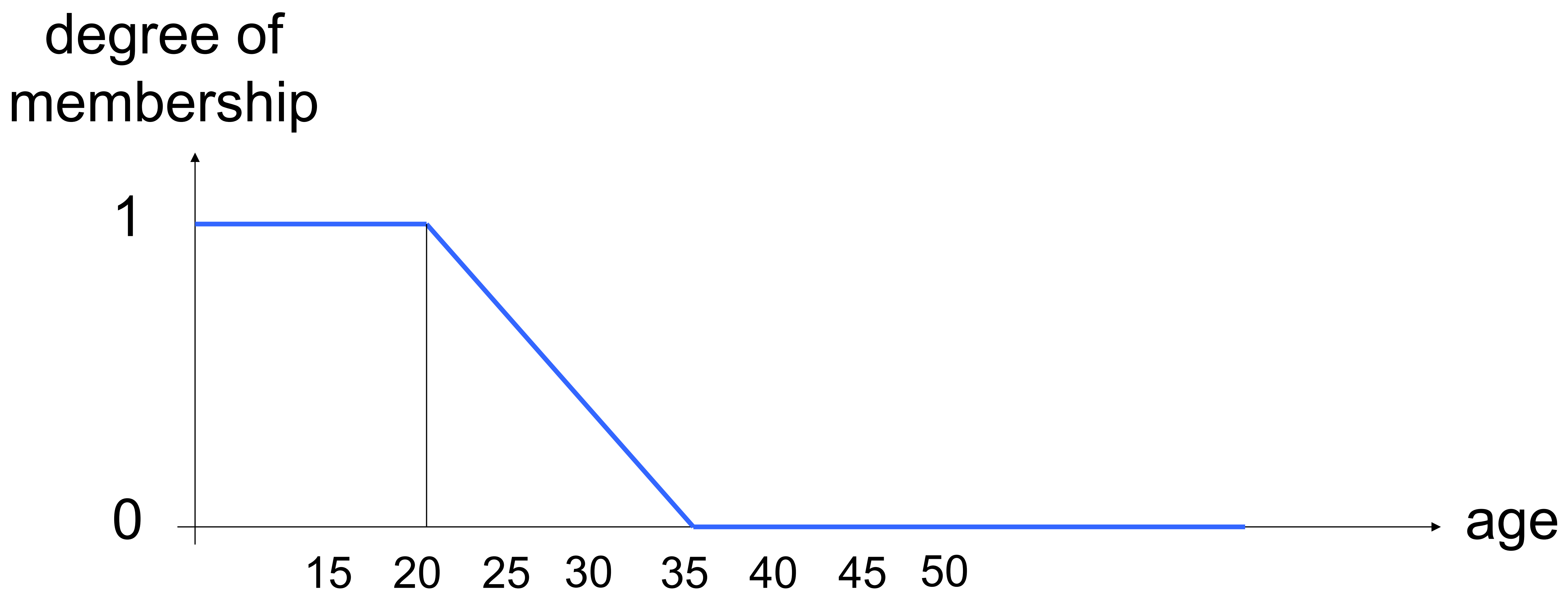
- Ambiguity is a characteristic of human language. For example:
  - If you study hard, then you will get good grades.
  - The second AI assignment is progressing strongly.
  - Students put effort into their projects.
  - Easy-going lecturer.
  - Difficult exam.
  - If the teacher is kind-hearted, then the exam will be easy.

Fuzzy Sets

- They relax the degree of membership restriction, $A: X \rightarrow [0,1]$, interval
- A fuzzy set in the universe U is characterized by the membership function A(x), which takes values in the interval [0,1], unlike classical sets that take either the value zero or one {0, 1}.
- The fuzzy set A can be represented as:
  - $A = \{ (\mu_A(x), x) \mid x \in U \}$
  - $A = \{ (\mu_A(x) / x) \mid x \in U \}$
- Where $\mu_A(x)$ is the degree of membership.

Fuzzy Sets

- Define the fuzzy set young:

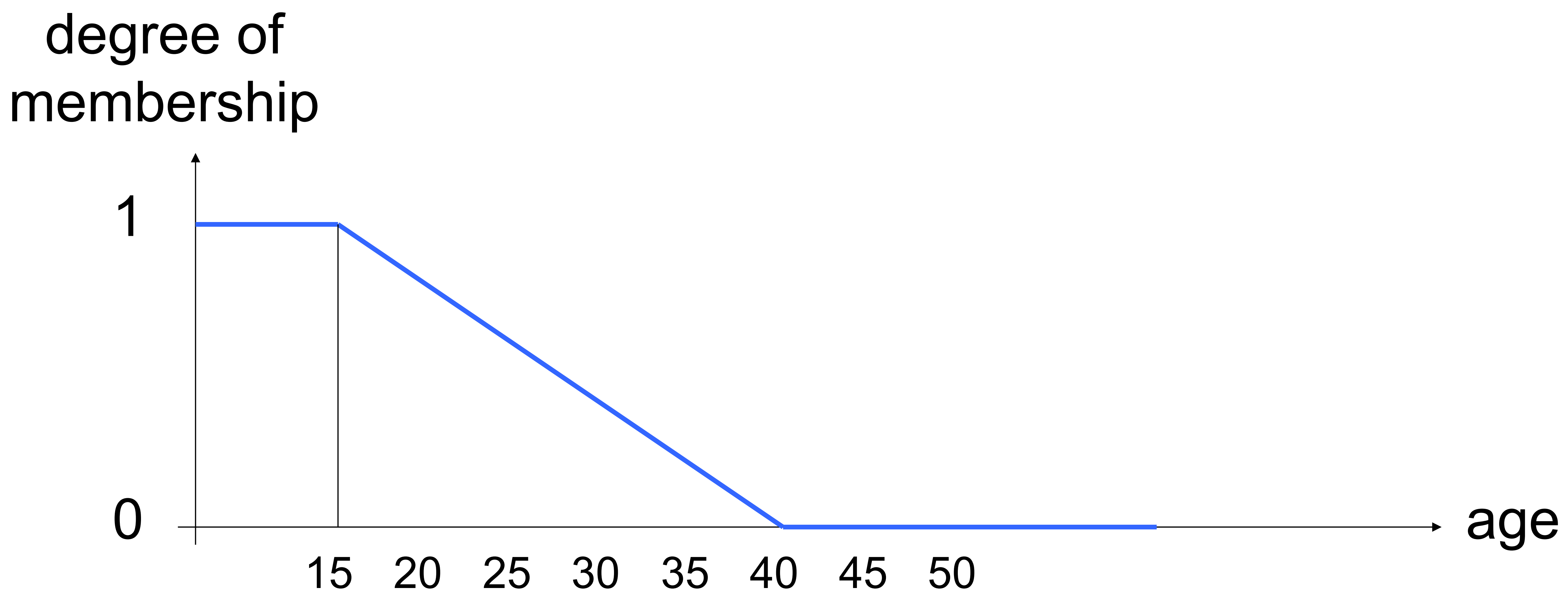


A = {1/10, 1/15, 1/20, 0.75/25, 0.25/30, 0/35}

A = {(1,10), (1,15), (1,20), (0.75,25), (0.25,30), (0,35)}

Fuzzy Sets

- Define the fuzzy set young:

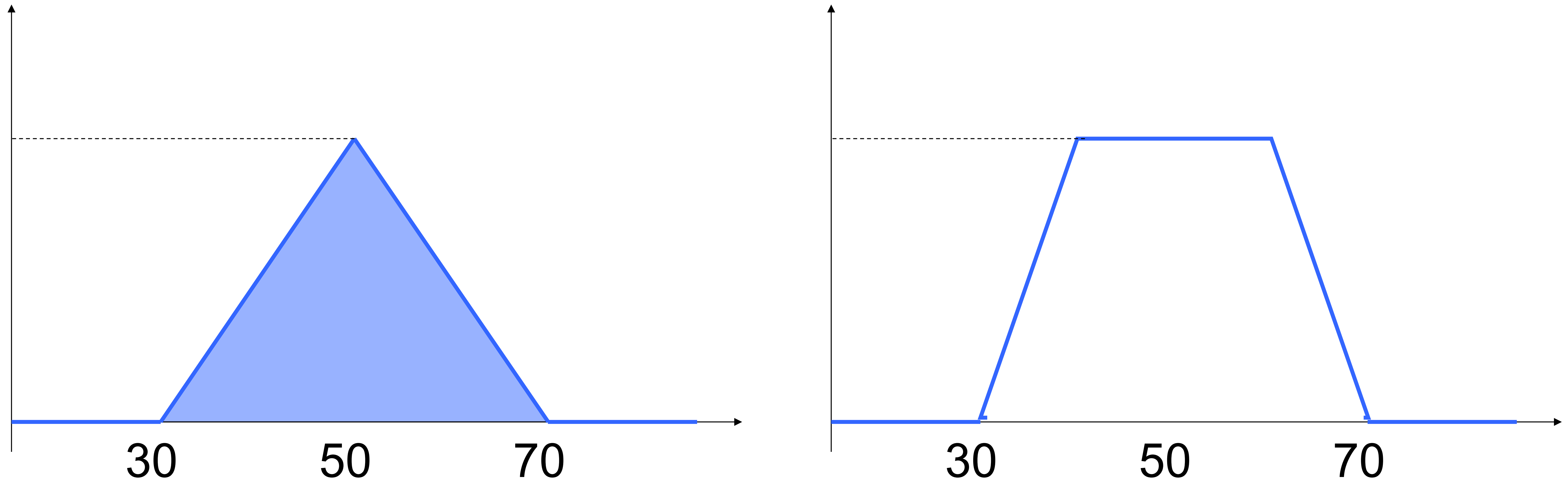


A = {1/10, 1/15, 0.80/20, 0.60/25, 0.40/30, 0.20/35, 0.0/40}

A = {(1,10), (1,15), (0.80,20), (0.60,25), (0.40,30), (0.20,35), (0.0,40)}

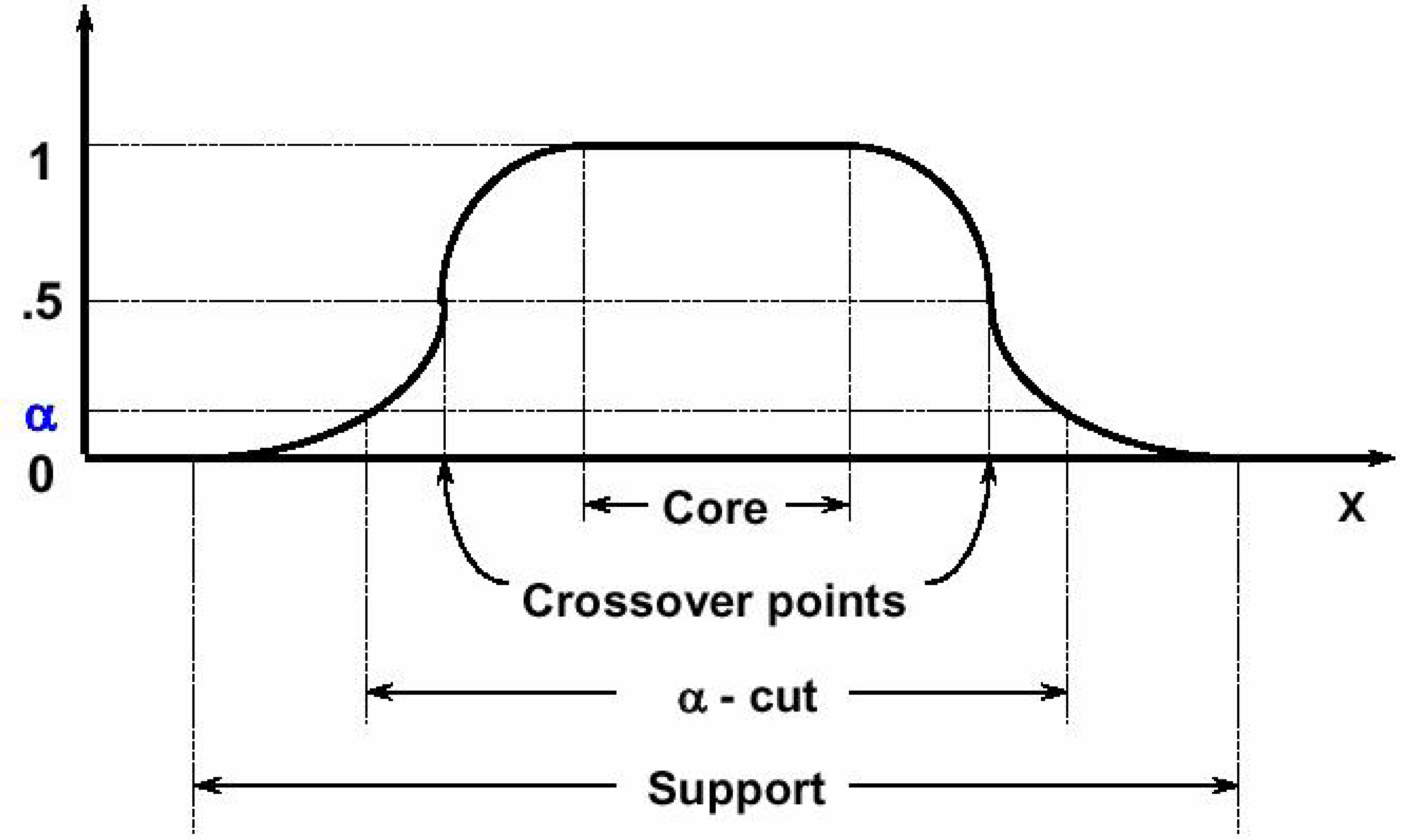
Fuzzy Sets

- Plot the fuzzy set near 50 years old:



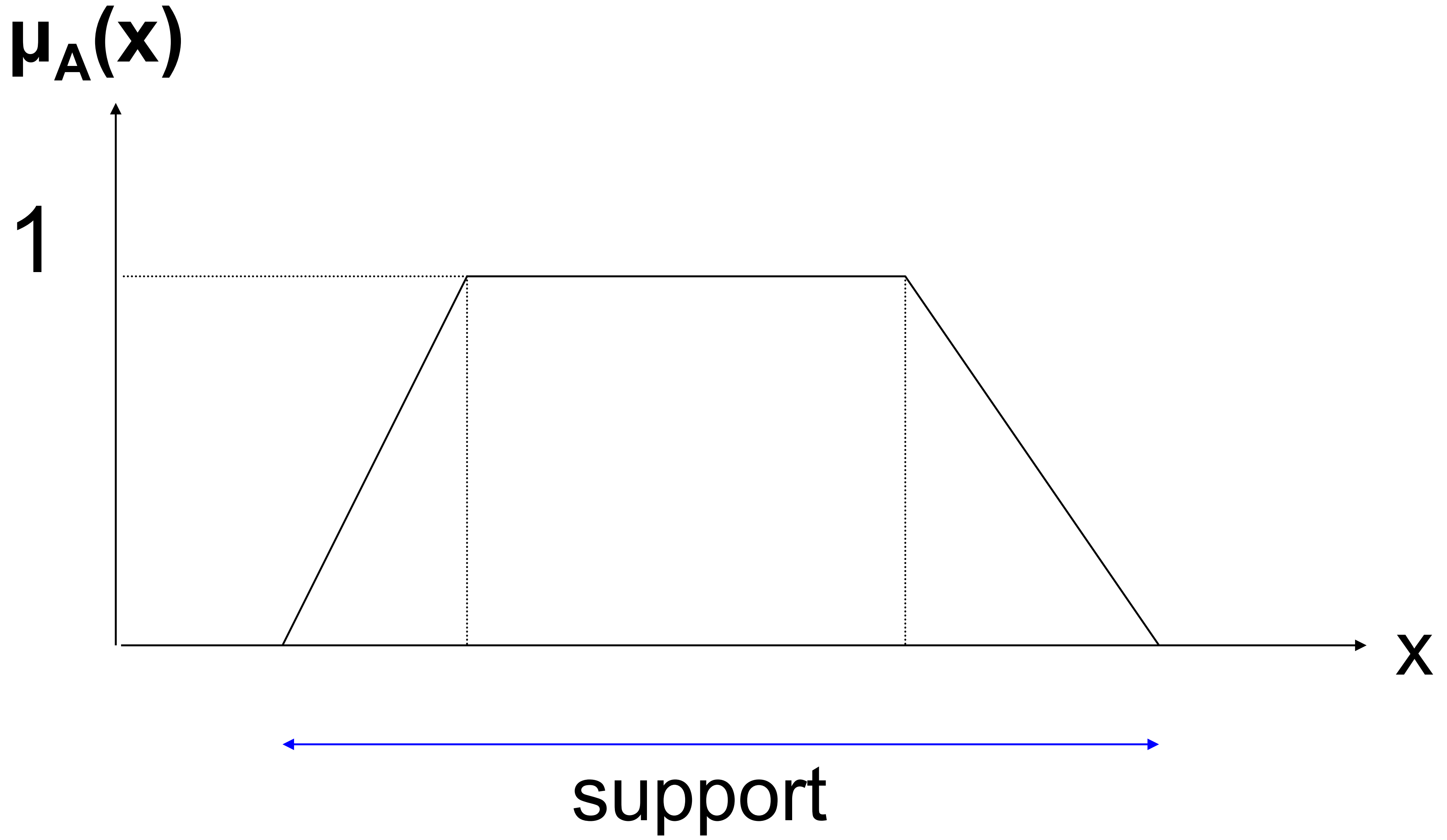
Fuzzy Sets

- Some basic concepts include: support, crossover points, core, height, centre value, and α-cut.



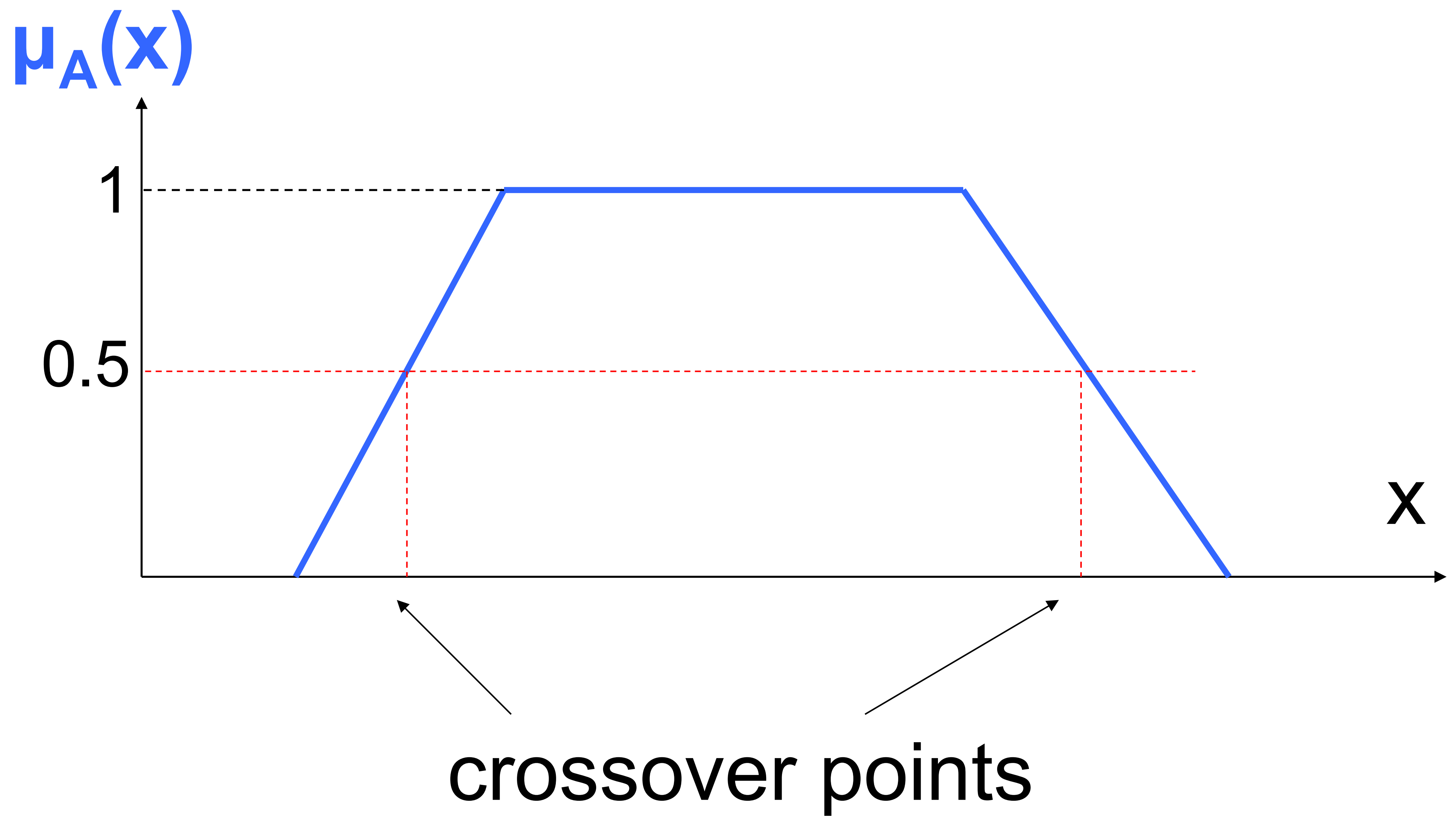
Fuzzy Sets: Support

- The support of a fuzzy set A in the universe of discourse U is a crisp set that contains all the elements of U that have non-zero membership values in A.
- If the support of a fuzzy set is empty, it is called an empty fuzzy set.
- If the support of the fuzzy set is represented by a single point in U, it is called a fuzzy singleton.
- Support(A) = $\{x \in U \,/\, \mu_A(x) > 0\}$



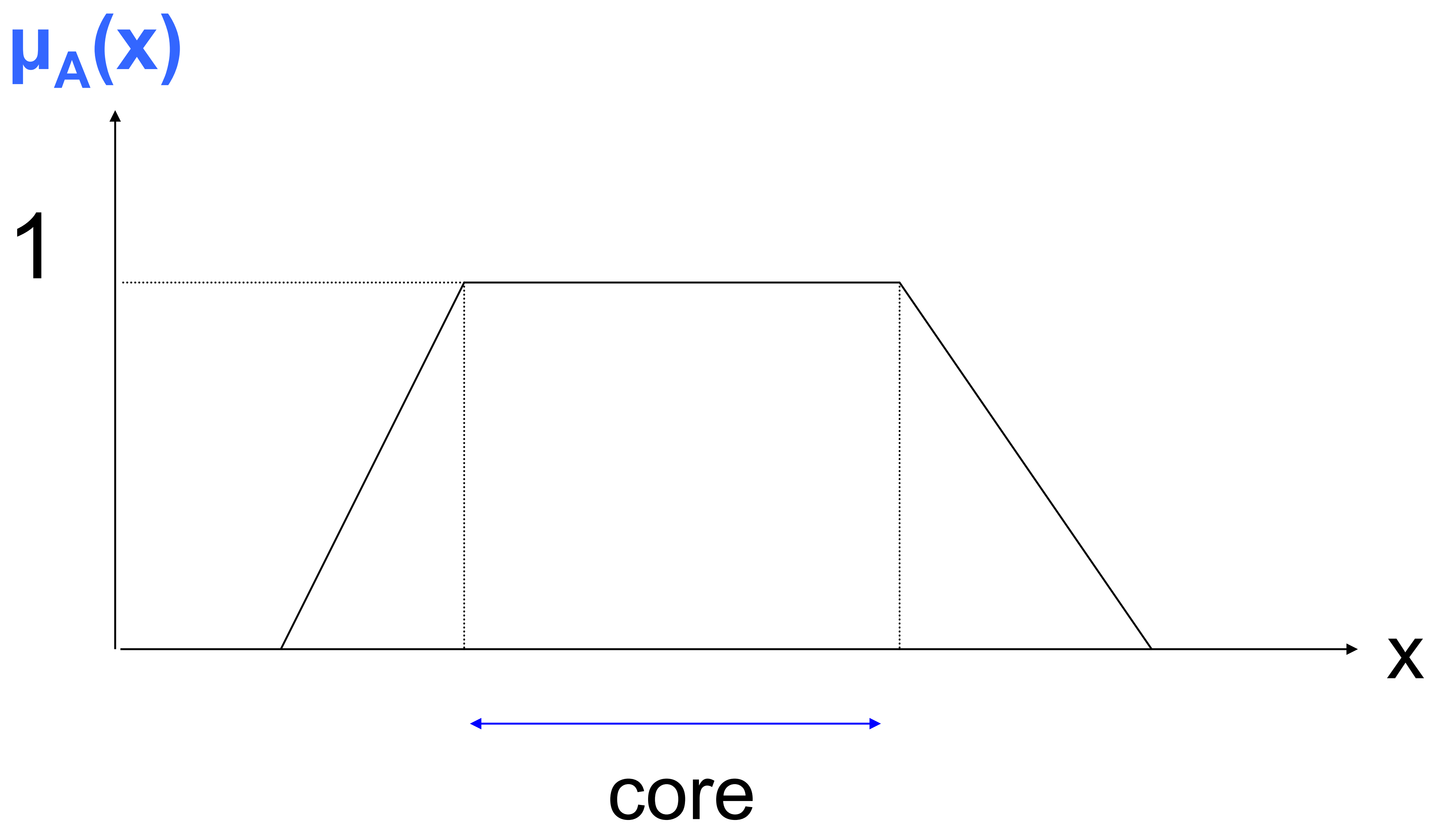
Fuzzy Sets: Crossover Point

- The crossover point of a fuzzy set is the point in U where the membership value in A is 0.5.



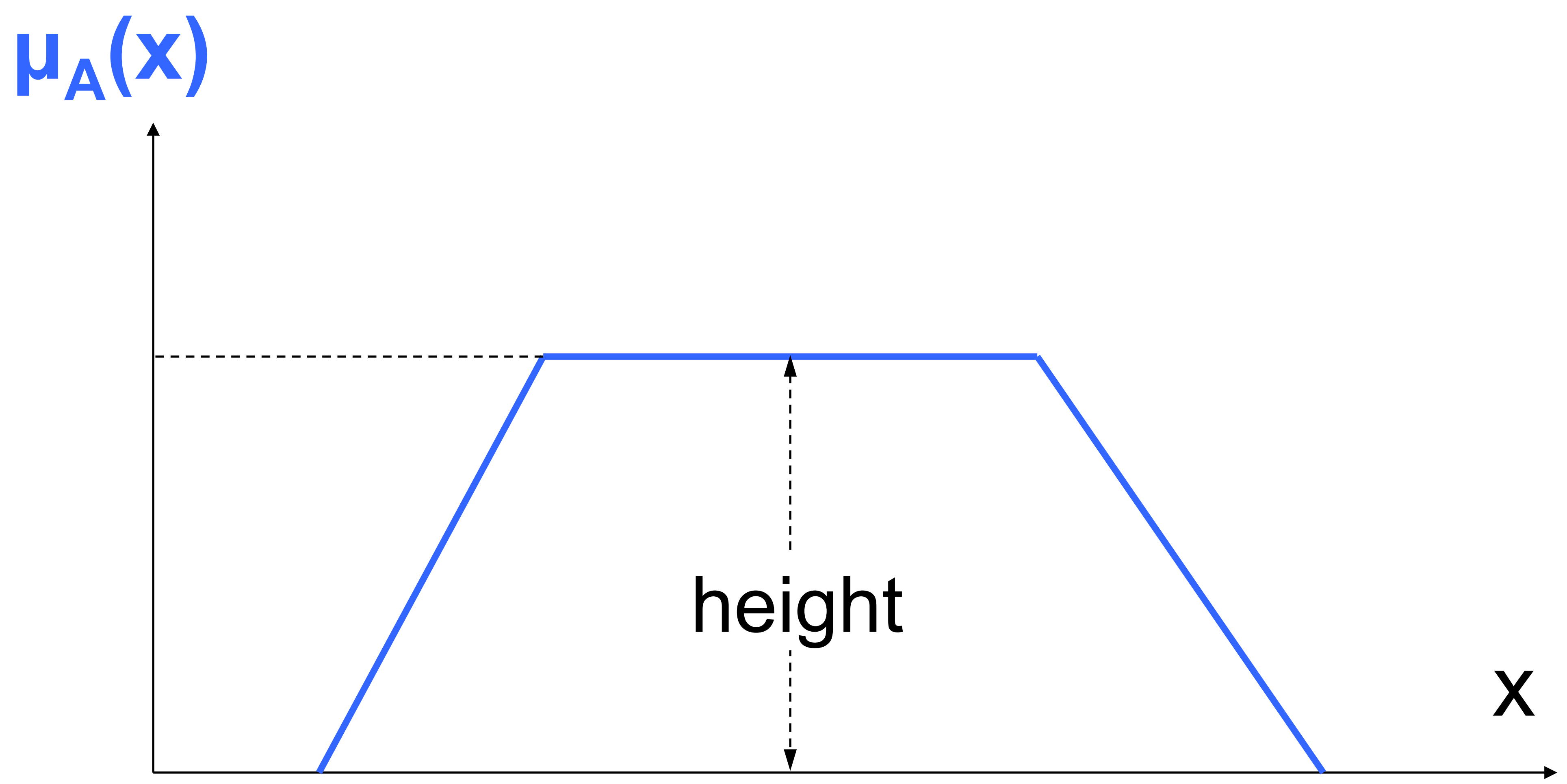
Fuzzy Sets: Core

- The set of x values where $\mu_A(x)$ reaches the value of 1 is called the core of the fuzzy set A.



Fuzzy Sets: Height

- The height of a fuzzy set is the highest membership value achieved by any point.
- In a normal fuzzy set, the height is 1.
  - normal: $\mu_A(x) = 1$
  - subnormal: $\mu_A(x) < 1$



Fuzzy Sets: Centre Value

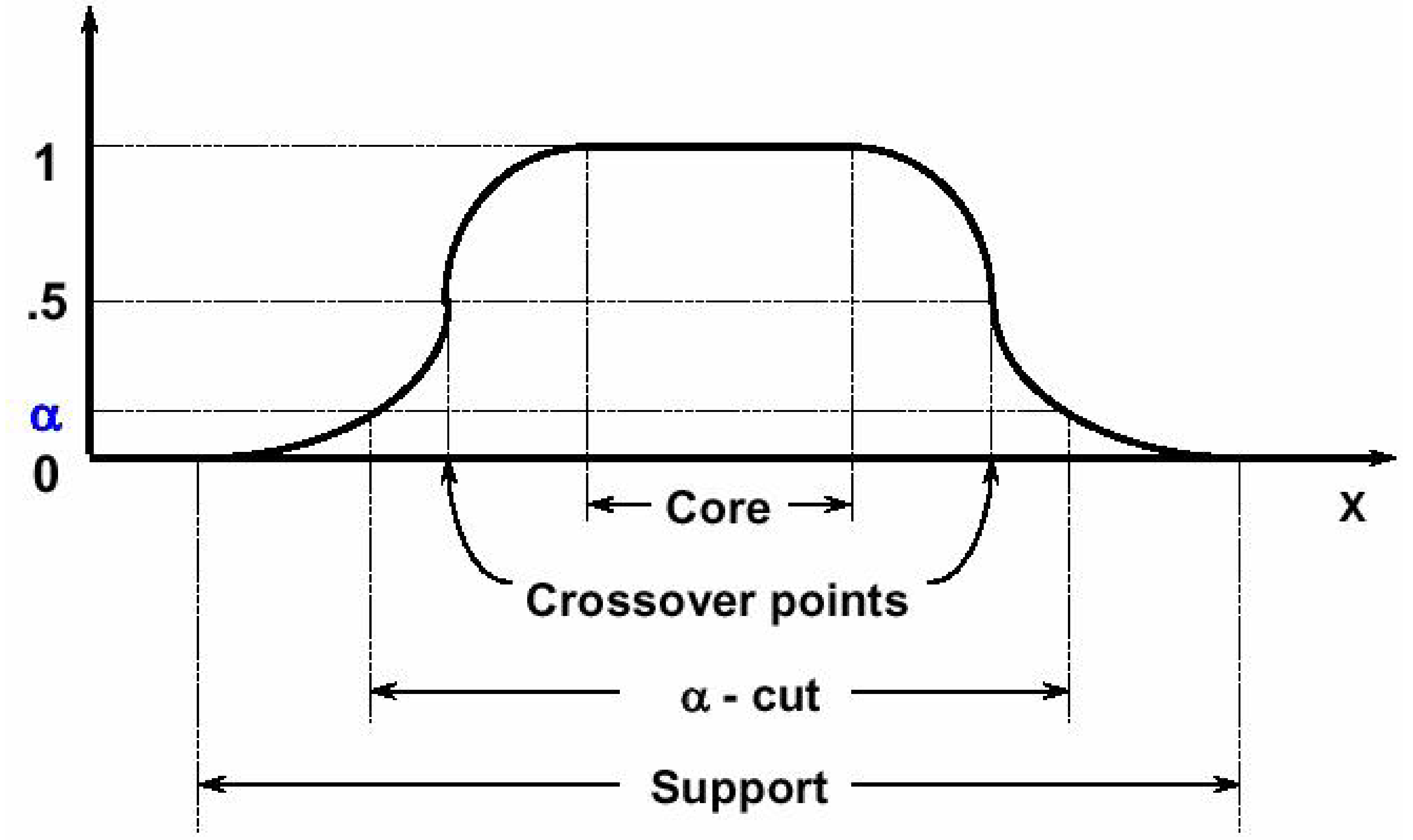
- If the mean value of all the points at which the membership function of a fuzzy set achieves its maximum value is finite, then the centre of the fuzzy set is the average of these values.

- If the mean value is infinite, then the centre is defined as the smallest among all the points that achieve the maximum membership value.

Fuzzy Sets: α-cut

- Given a fuzzy set A defined in X and a number $\alpha \in [0; 1]$, an α-cut is a crisp set that contains all the elements in U that have membership values in A greater than or equal to α, defined by:
  - $A_\alpha = \{ x \in U / \mu_A(x) \geq \alpha\}$
  - $A_{\alpha +} = \{ x \in U / \mu_A(x) > \alpha\}$ strong α-cut

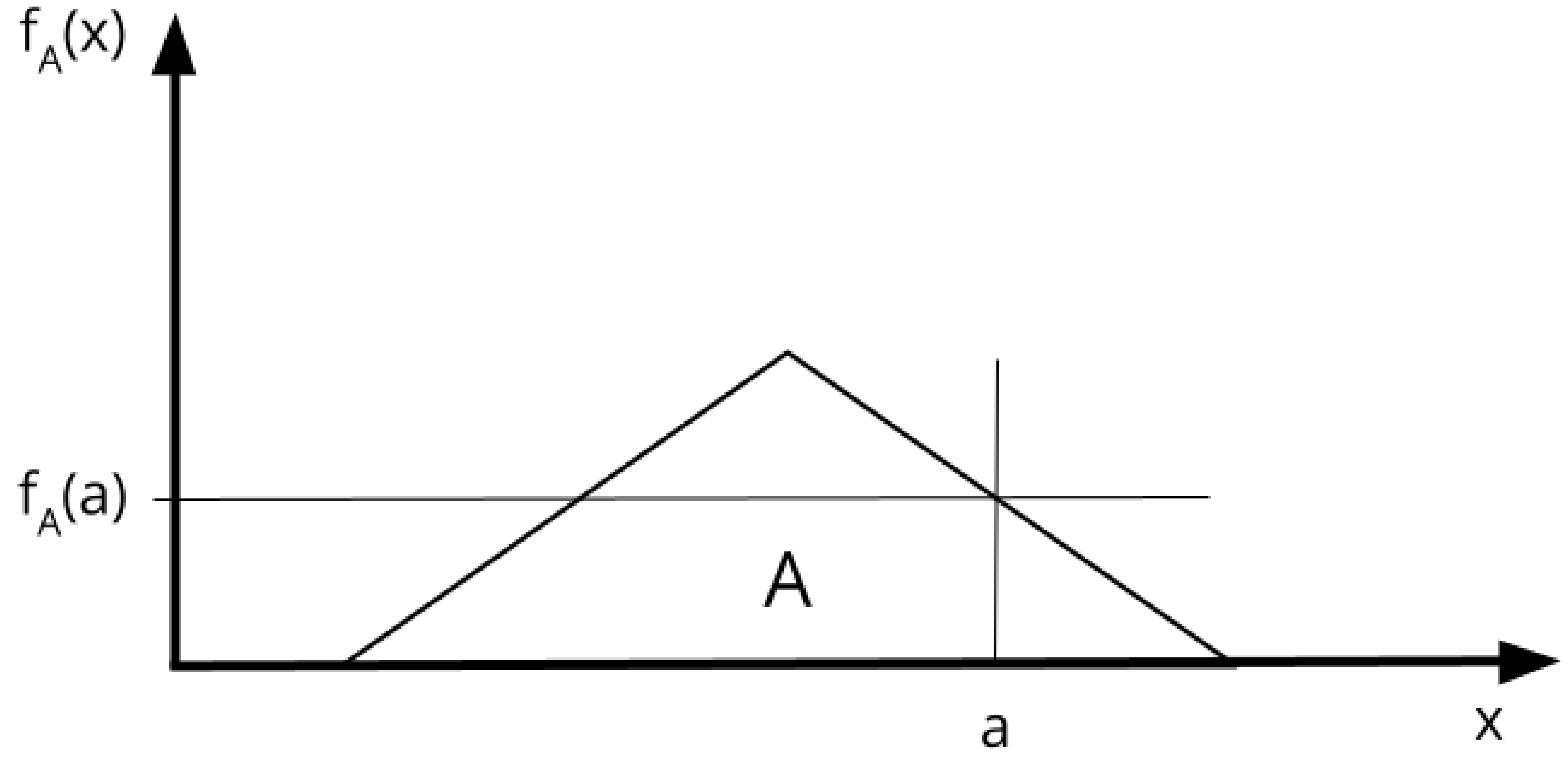
- Properties. Given a fuzzy set A defined in X and two values α1 and α2 ∈ [0; 1] such that α1 > α2, then:
  - $A_{\alpha 1} \subset A_{\alpha 2}$ and $A_{\alpha 1+} \subset A_{\alpha 2+}$
  - $(A_{\alpha 1} \cap A_{\alpha 2}) = A_{\alpha 1}$ and $(A_{\alpha 1+} \cap A_{\alpha 2+}) = A_{\alpha 1+}$
  - $(A_{\alpha 1} \cup A_{\alpha 2}) = A_{\alpha 2}$ and $(A_{\alpha 1+} \cup A_{\alpha 2+}) = A_{\alpha 2+}$



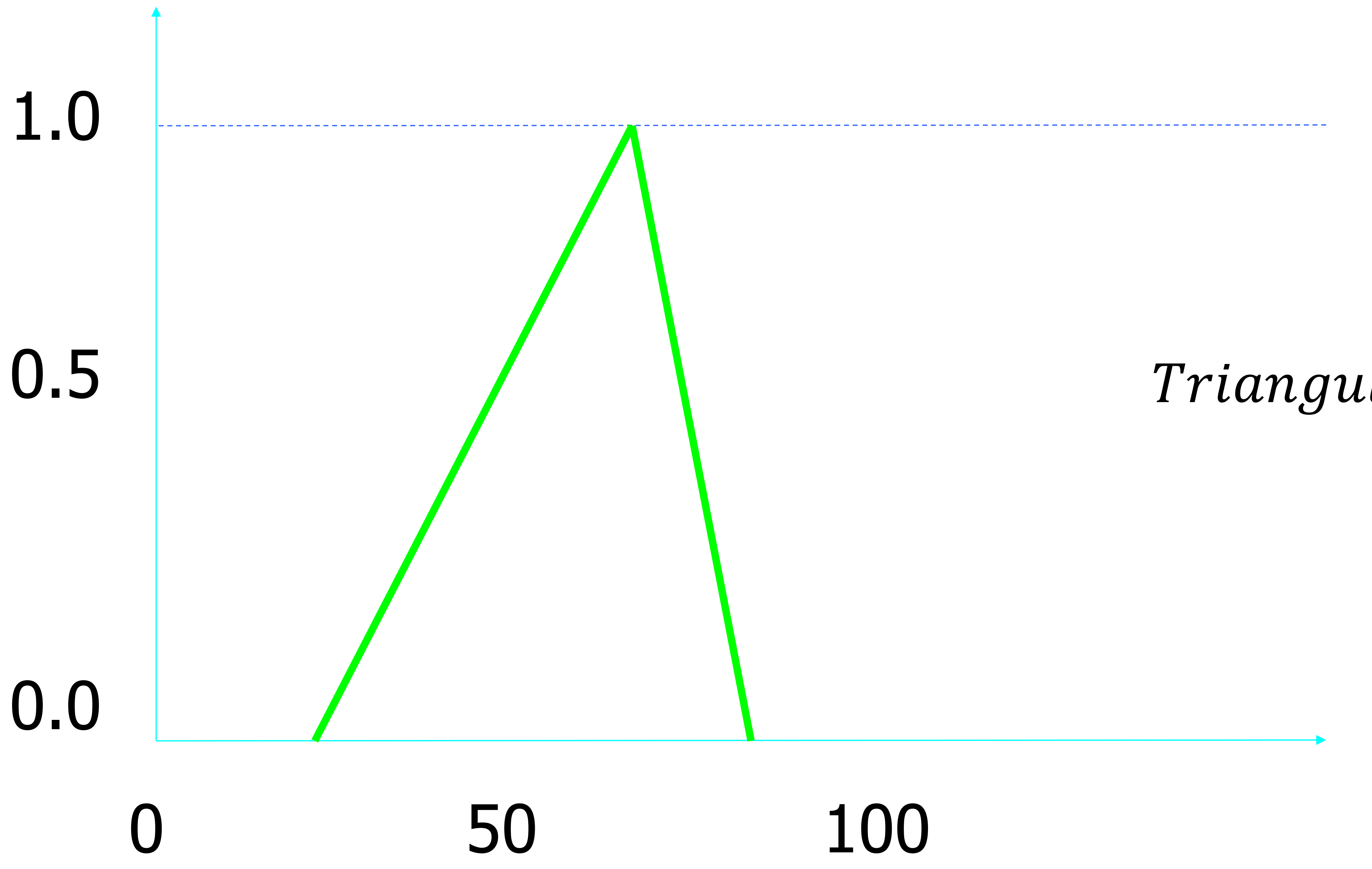
Membership functions

- If $f_A(x)$ denotes the membership function of x to set A, then:
    - $f_A(x)$ is a value between 0 and 1.
    - if $f_A(x) = 1$, x belongs completely to A.
    - if $f_A(x) = 0$, x does not belong to A.

- From this definition, it is possible to verify the following properties:
    - $f_{A \text{ or } B}(x) = \max(f_A(x), f_B(x))$.
    - $f_{A \text{ and } B}(x) = \min(f_A(x), f_B(x))$.
    - $f_{\text{nor } A}(x) = 1 - f_A(x)$



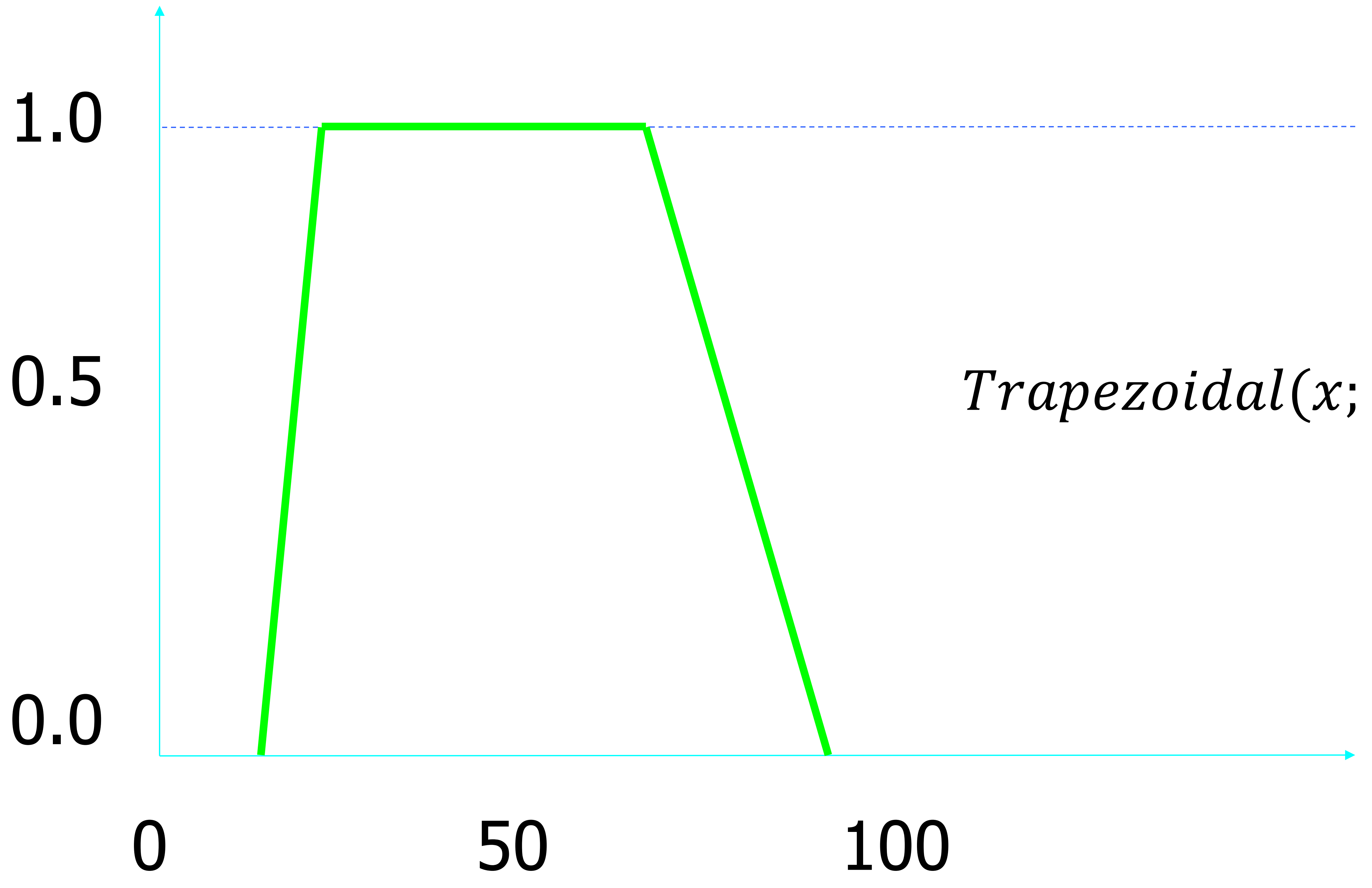
Membership functions: Triangular



$Triangular(x;\ 20,\ 60,\ 80)$

$$Triangular(x; a, b, c) = max\left(min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right),\ 0\right)$$

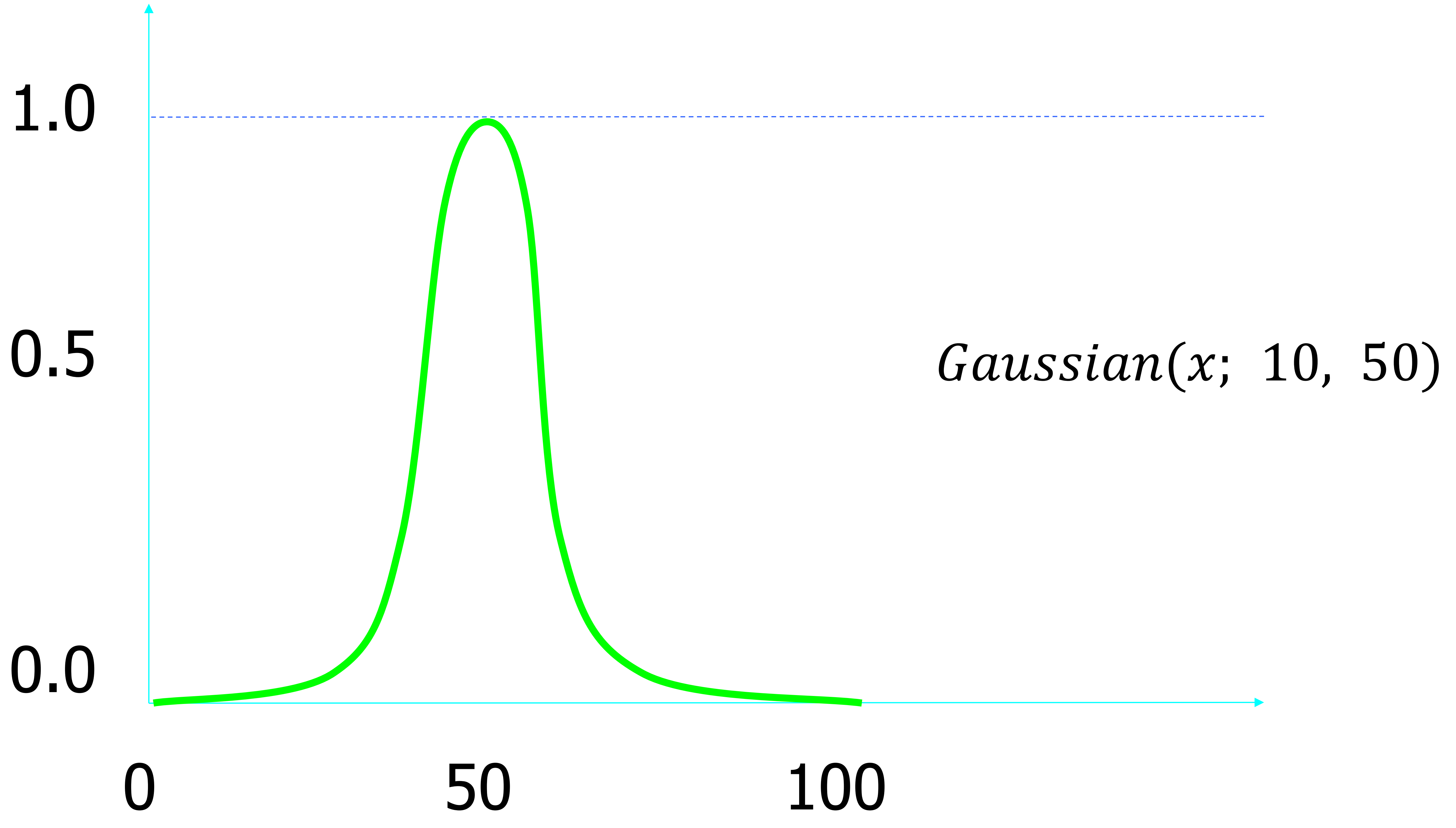
Membership functions: Trapezoidal



$$Trapezoidal(x;\ 10,\ 20,\ 60,\ 95)$$

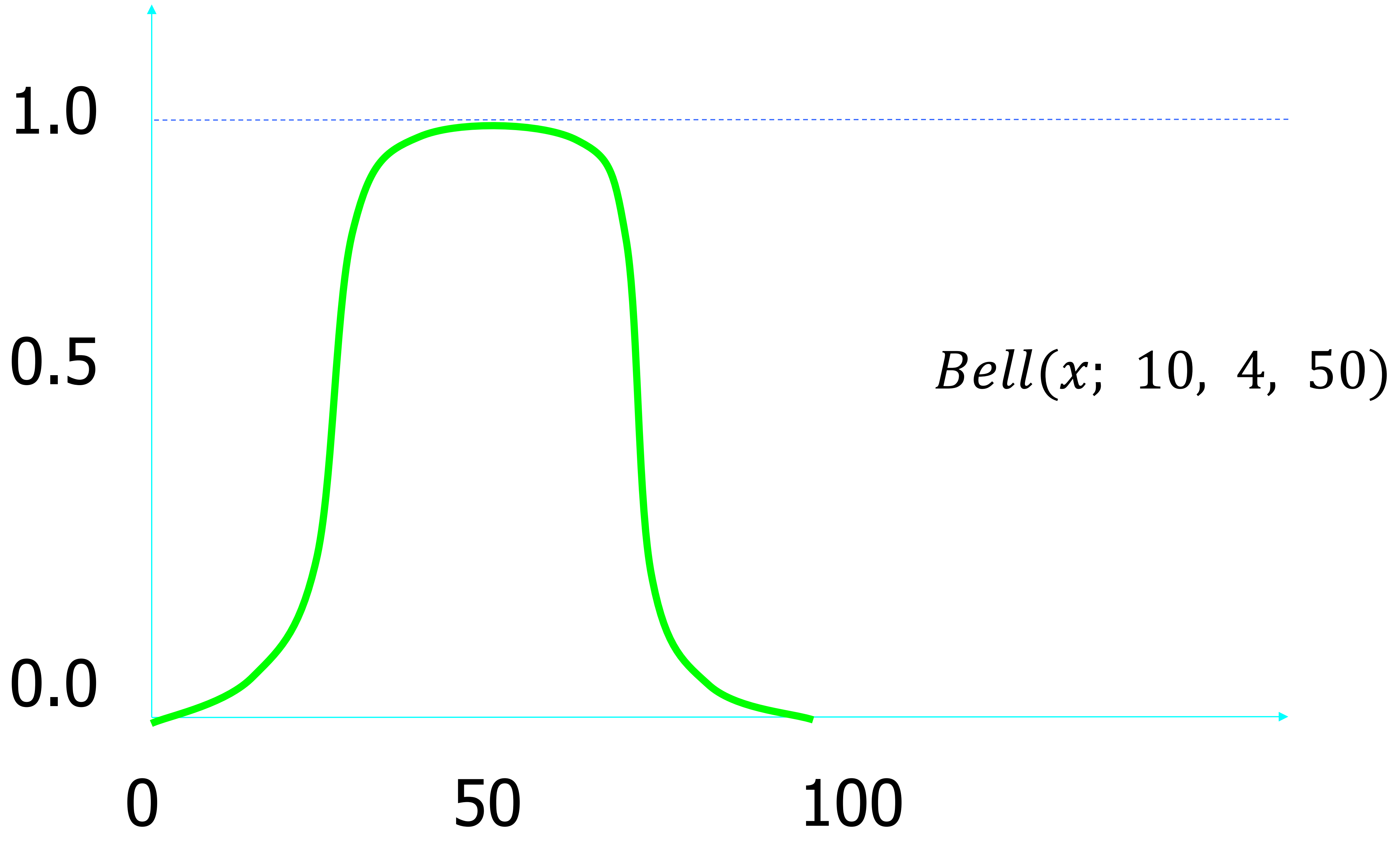
$$Trapezoidal(x; a, b, c, d) = max\left(min\left(\frac{x-a}{b-a},\ 1,\ \frac{d-x}{d-c}\right),\ 0\right)$$

Membership functions: Gaussian



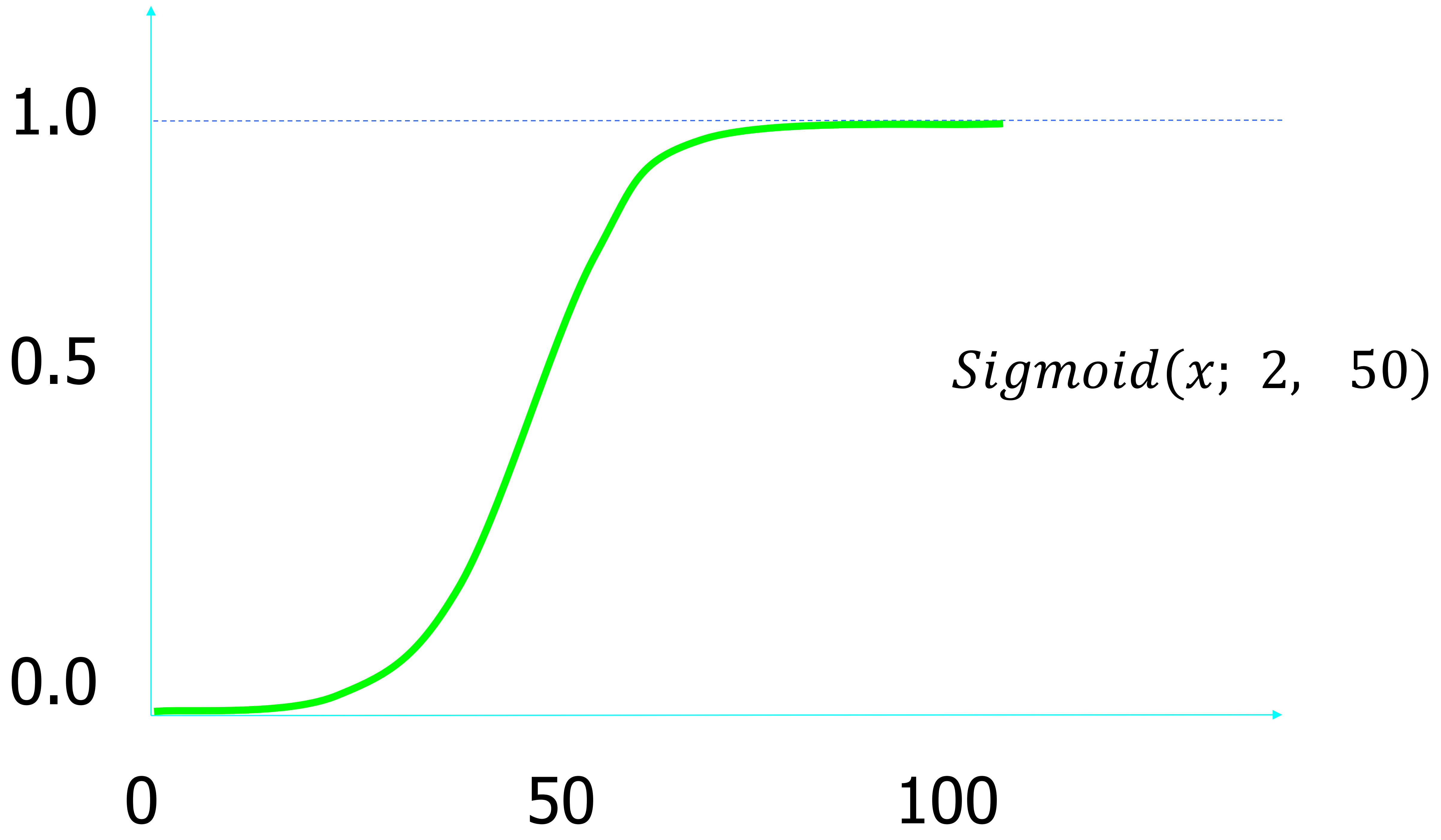
$$Gaussian(x; \sigma, c) = e^{-\left(\frac{x-c}{\sigma}\right)^2}$$

Membership functions: Bell-shaped



$$Bell(x; a, b, c) = \frac{1}{1 + \left(\frac{x-c}{a}\right)^{2b}}$$

Membership functions: Sigmoid



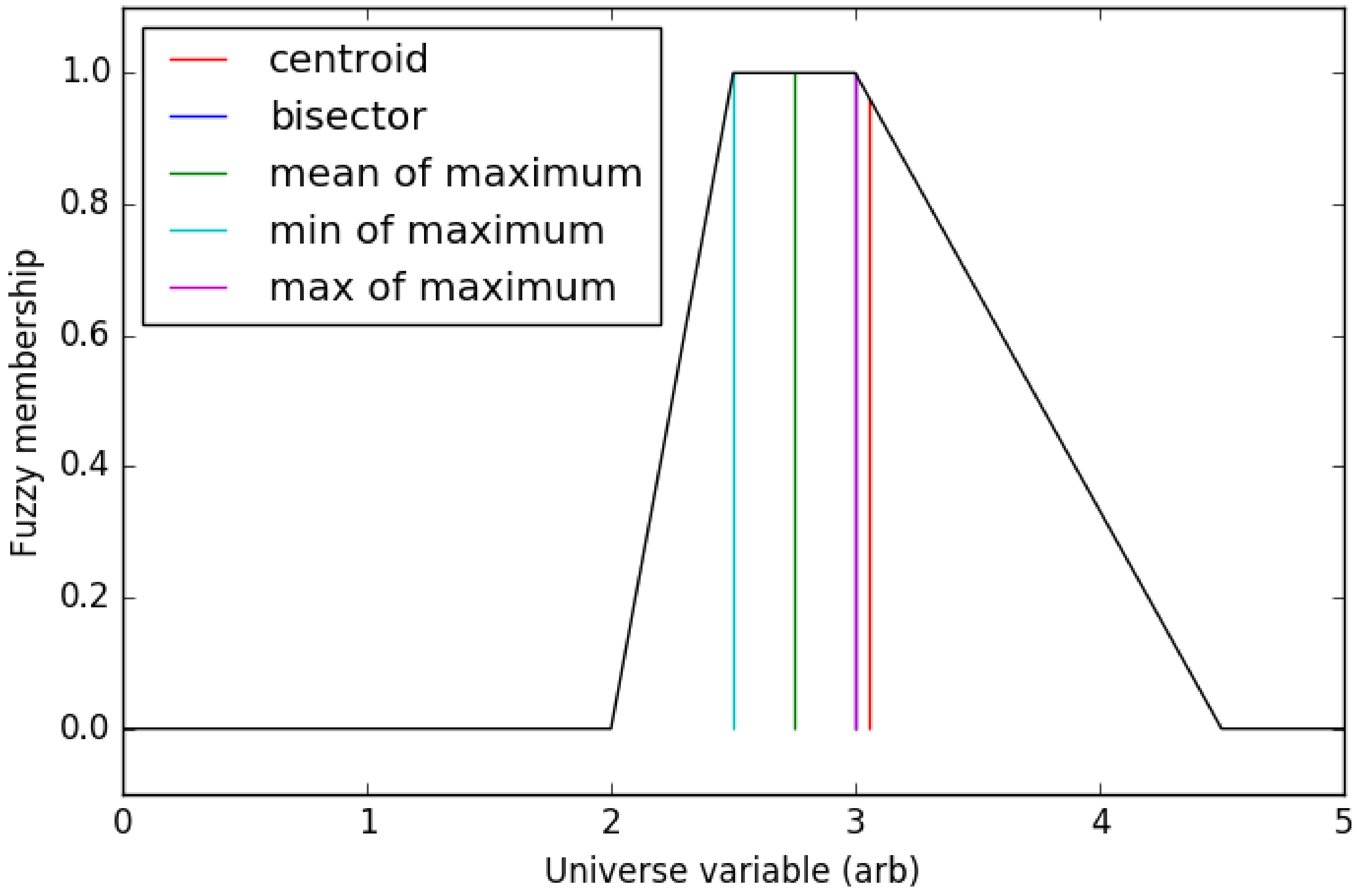
$$Sigmoid(x; a, c) = \frac{1}{1 + e^{-a(x-c)}}$$

Rules

- Rules are expressed in terms of linguistic variables as IF x is A THEN y is B
- Logic operators can be used, and the following properties apply:
  - $f_{A \text{ or } B}(x) = \max(f_A(x), f_B(x))$.
  - $f_{A \text{ and } B}(x) = \min(f_A(x), f_B(x))$.
  - $f_{\text{nor } A}(x) = 1 - f_A(x)$
- For example:
  - IF the speed is high and the temperature is high, THEN the fuel injection should be low.
  - IF the food is delicious or the service excellent, THEN the tip should be generous.

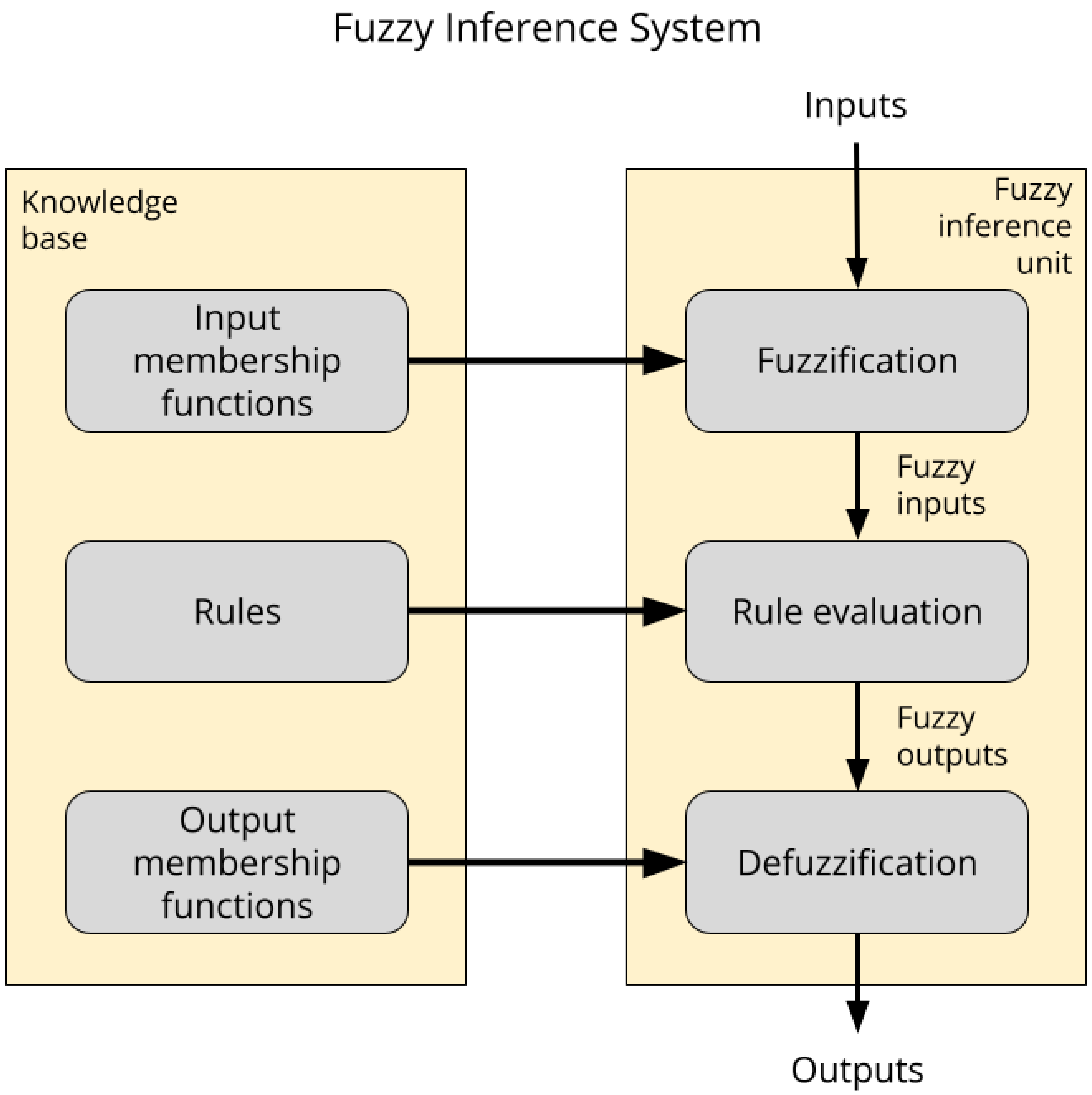
Defuzzification

- The area for the resultant consequent should be converted from fuzzy to crisp, i.e., a quantifiable result.
- There are different methods for defuzzification, for instance, the ones implemented in scikit-fuzzy (a.k.a. skfuzzy) are: centroid, bisector, mean of maximum, min of maximum, max of maximum.



Fuzzy Inference

- Mapping from given inputs to a crisp output using fuzzy logic is known as fuzzy inference.
- Inputs are crisp values and go through a fuzzification process (from input membership functions).
- Rules are evaluated as linguistic variables using IF-THEN structures.
- Output memberships functions establish the outputs area which is defuzzified using methods such as centroid.



Fuzzy Inference: The Tipping Problem*

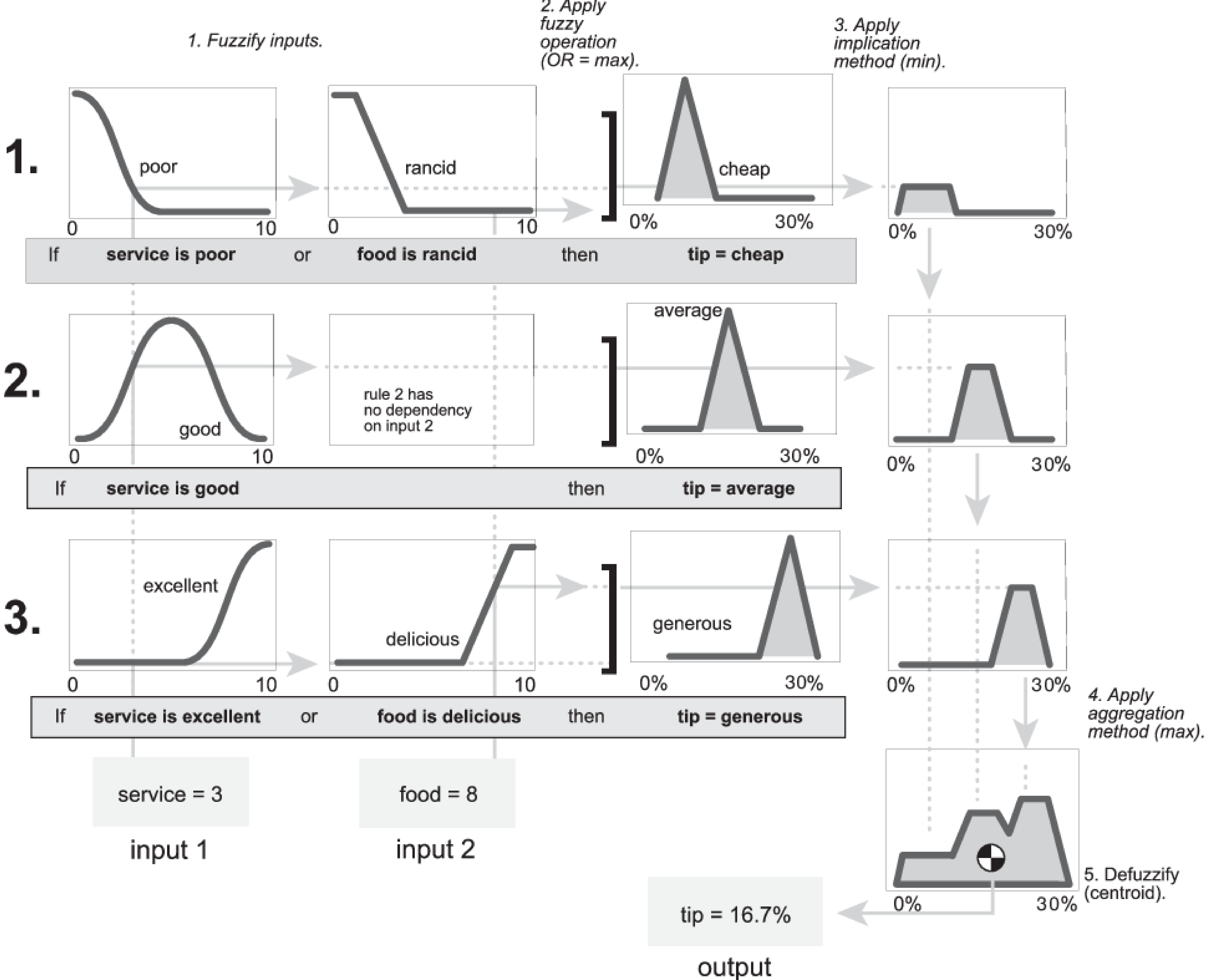
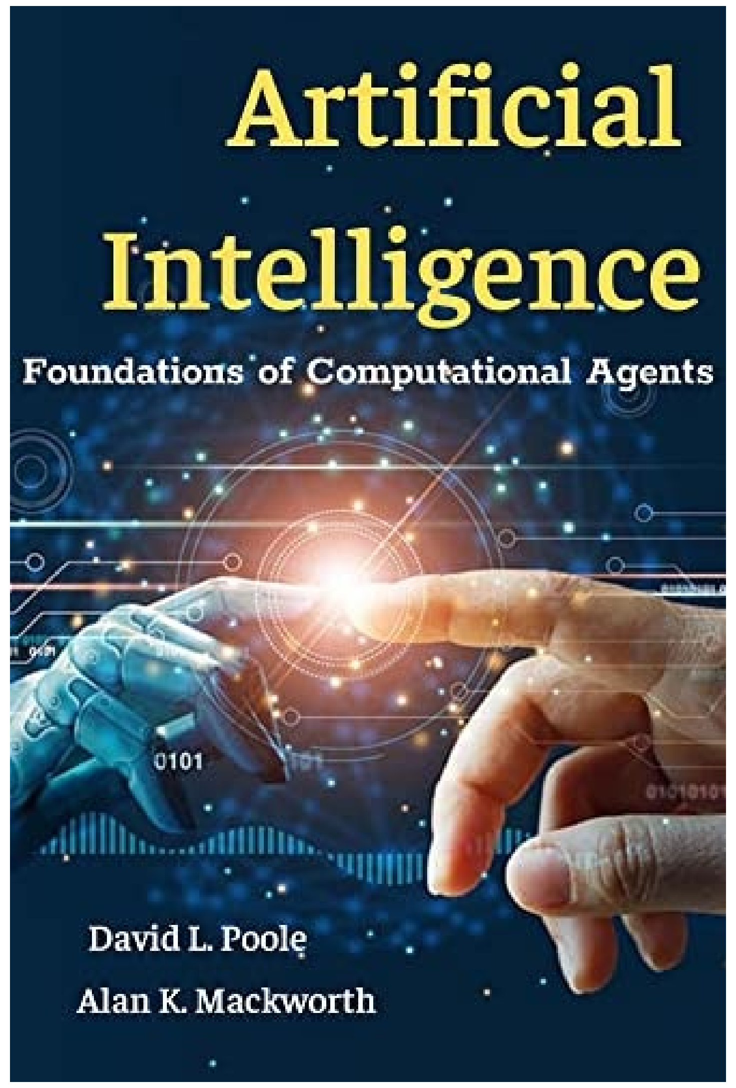
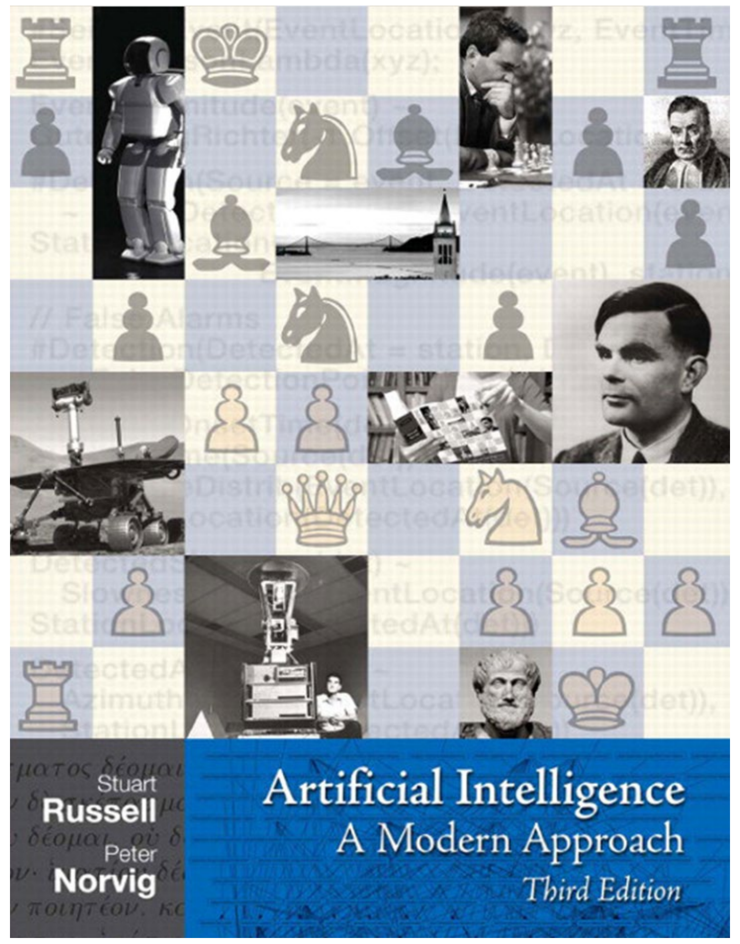


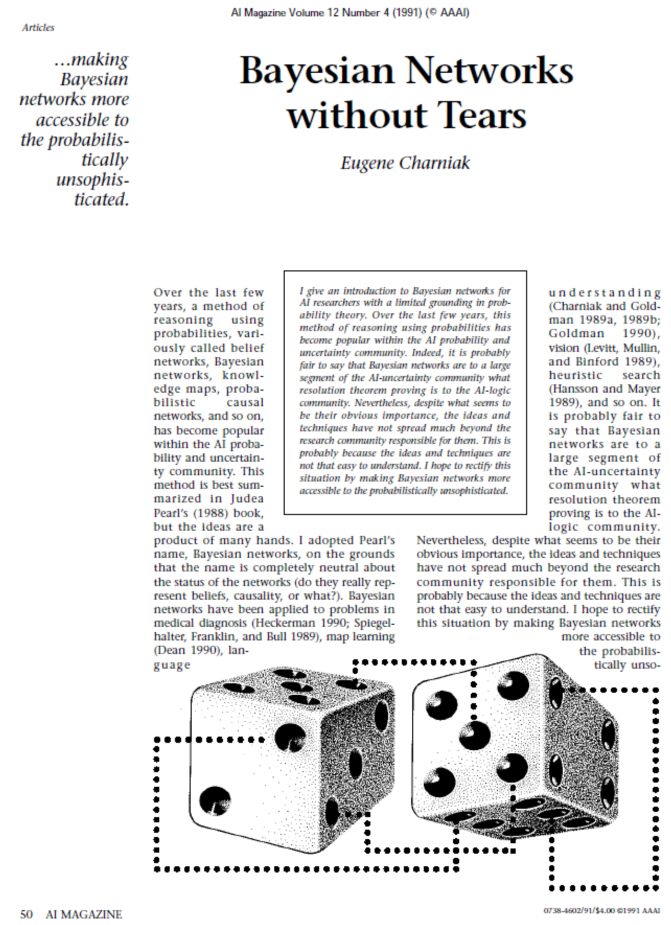
*Image from MathWorks

References

- Poole & Mackworth, Artificial Intelligence: Foundations of Computational Agents, chapter 5.
- Russell, S.J. & Norvig, P. (2016). Artificial Intelligence: A Modern Approach. Third Edition, *Pearson Education*, Hoboken, NJ. Chapters 7, 8, 12,13 and 14.
- Charniak, E. (1991). Bayesian networks without tears. *AI magazine*, 12(4), 50-50.
- Kendal, S. L., & Creen, M. (2007). An introduction to knowledge engineering. *Springer London*. Chapter 7.



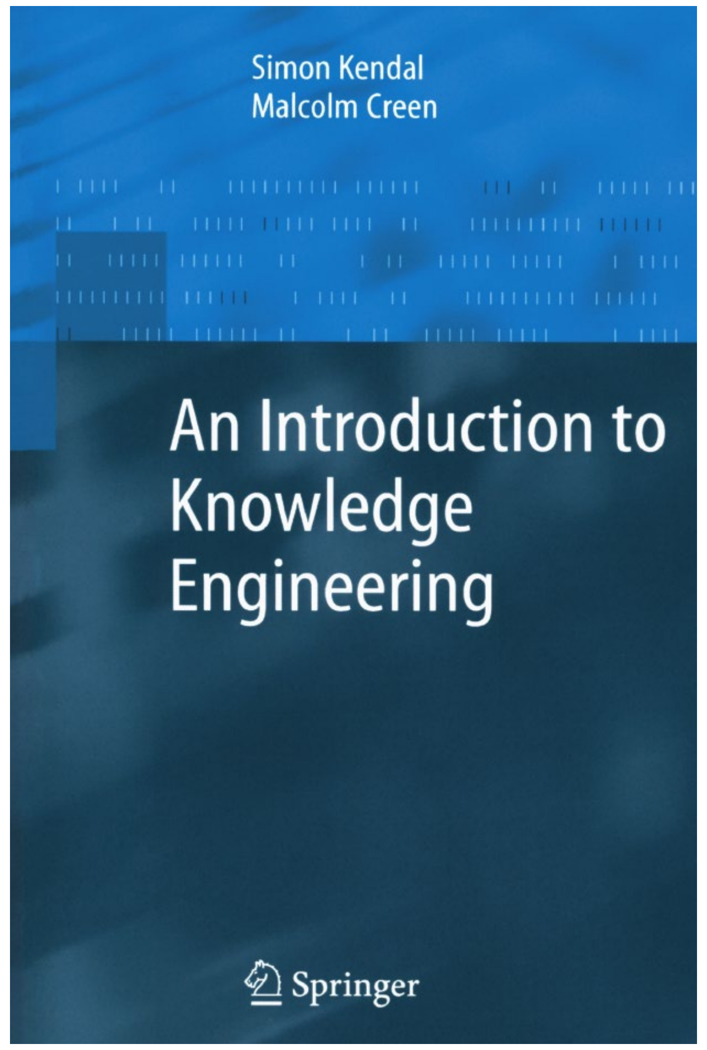




Bayesian Networks without Tears

Eugene Charniak

...making Bayesian networks more accessible to the probabilistically unsophisticated.

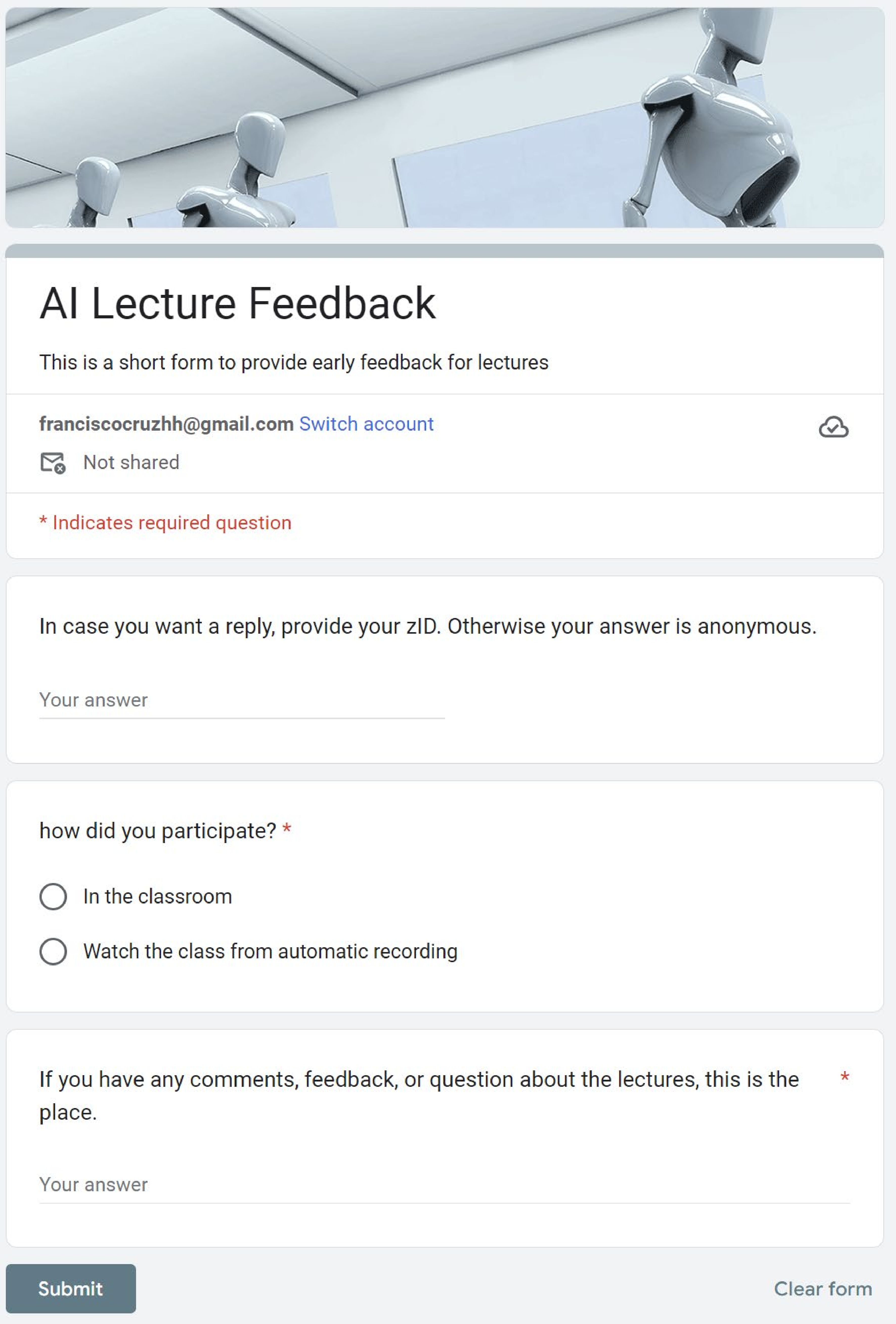


Feedback

- In case you want to provide anonymous feedback on these lectures, please visit:

- https://forms.gle/KBkN744QuffuAZLF8

Muchas gracias!



AI Lecture Feedback

This is a short form to provide early feedback for lectures

franciscocruzhh@gmail.com Switch account

Not shared

* Indicates required question

In case you want a reply, provide your zID. Otherwise your answer is anonymous.

Your answer

how did you participate? *

- In the classroom
- Watch the class from automatic recording

If you have any comments, feedback, or question about the lectures, this is the place. *

Your answer

Submit

Clear form