

# 6. Sistemes basats en regles

Models d'intel·ligència artificial

# IA simbólica

# Inteligència artificial simbòlica

- **IA simbólica o IA basada en coneixement:**
  - Extraiem coneixement d'experts i el representem d'una forma que les màquines puguin entendre.
  - Utilitzem aquest coneixement per a:
    - Resoldre problemes automàticament.
    - Explicar el raonament de la màquina.
    - Aprendre noves coses.
    - Millorar el coneixement existent.

# Representació del coneixement

- Coneixement vs dades vs informació:
  - **Dades:** Fets o valors.
  - **Informació:** Dades amb significat.
  - **Coneixement:** Informació amb significat i estructura.
- “ El coneixement és un conjunt d'informació estructurada i interrelacionada que permet a un agent realitzar tasques. ”

# **Jerarquia del coneixement (I)**

- Moltes vegades definim el coneixement en relació a conceptes similars.
- La jerarquia del coneixement o jerarquia de DIKW és un model que mostra la relació entre *dades, informació, coneixement i saviesa*.



# Jerarquia del coneixement (II)

- **Dades (Data):** Fets o valors registrats en un suport físic. És independent de l'agent i pot ser interpretat de diferents maneres.
  - Exemple: "*Un smartwatch registra la temperatura corporal de la persona.*"
- **Informació (Information):** És com les dades són interpretades per un agent. És subjectiva i depèn de l'agent.
  - Exemple: "*La temperatura corporal de la persona és de 37°C*"

# Jerarquia del coneixement (III)

- **Coneixement** (Knowledge): És informació integrada en el nostre model del mon. Depèn de l'agent i dels seus coneixements previs.
  - Exemple: *"Si la temperatura és superior a 37°C, llavors la persona té febre"*
- **Saviesa** (Wisdom): Representa el meta-coneixement: coneixement sobre com i quan aplicar el coneixement.
  - Exemple: *"Si la persona té febre, llavors ha de prendre paracetamol"*

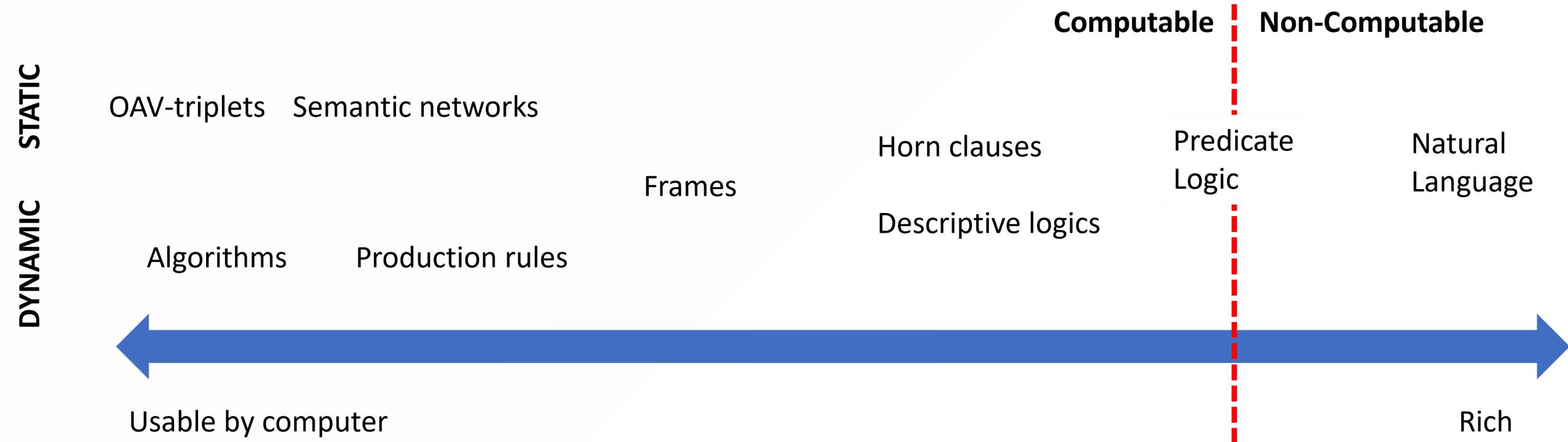
# Representació del coneixement (I)

- És la forma en la que representem el coneixement per a que les màquines puguin entendre'l.
- És un dels problemes fonamentals de la intel·ligència artificial.
- S'ha de representar de forma que:
  - Sigui **entendible** per a les màquines.
  - Sigui **útil** per a resoldre problemes.
  - Sigui **eficient** per a ser processat per les màquines.

# Representació del coneixement (II)

- Podem veure les diferents representacions com un **continuum**:
  - A l'esquerra tenim les representacions més **simples** (algorismes); utilitzables per els ordinadors de forma eficient però molt poc flexibles.
  - A la dreta tenim les representacions més **flexibles** (text natural); molt potents però no utilitzables diréctament per les màquines.

# Continuum del coneixement



# Representació del coneixement (III)

- **Representacions de xarxa:**
  - En la ment humana el coneixement es representa com una xarxa de conceptes interrelacionats.
  - Les representacions de xarxa intentem fer el mateix en un graf dins dels ordinadors.
    - Les anomenem **xarxes semàntiques**.
  - Hi ha diferents tipus: Parells d'atributs i valors, representacions jeràrquiques, representacions procedurals, lògica, etc.

# Parells d'atributs i valors o triplets objecte-atribut-valor

- Aprofitem que un graf es pot representar com una llista de nodes i arestes per a representar el coneixement.
- El coneixement es representa com una llista de parells d'atributs i valors.
  - "*El gos és un animal, el gos té quatre potes, el gos té pèl, el gos té cua, etc.*"
  - "*El colom és un animal, el colom és un ocell, el colom té dues potes, etc.*"
  - "*El cotxe és un vehicle, el cotxe té quatre rodes, el cotxe té un motor, etc.*"

# **Representacions jeràrquiques**

- El coneixement es representa com un arbre.
- Els nodes de l'arbre representen conceptes.
- Les arestes representen relacions entre conceptes.
  - Animals → Vertebrats → Mamífers → Gossos → Caniche
  - Animals → Vertebrats → Ocells → Coloms → Colom comú
  - Objectes → Vehicles → Cotxes → Cotxe de gasolina

# Representacions procedurals

- El coneixement es representa com un conjunt d'accions que es poden realitzar quan es donen certes condicions.
- Anomenem **regles de producció** a les **declaracions** que ens permeten obtindre conclusions a partir de certes premisses.
- Són de la forma: **IF** (premissa) **THEN** (conclusió)
  - **IF** (la temperatura és superior a 37°C) **THEN** (la persona té febre)
  - **IF** (la persona té febre) **THEN** (la persona ha de prendre paracetamol)

# Lògica

- La lògica és un sistema formal que ens permet representar el coneixement i raonar sobre ell proposada per Aristòtil fa més de 2000 anys com a eina per a la **deducció**.
- La lògica proposicional és un sistema formal que ens permet representar el coneixement i raonar sobre ell. Molt potenta a nivell teòric però no directament utilitzable per les màquines; un subconjunt es utilitzable en sistemes com prolog.
- Ex:  $p$ : "La persona té febre",  $q$ : "La persona ha de prendre paracetamol"
  - $p \rightarrow q$ : "Si la persona té febre, llavors la persona ha de prendre paracetamol"
  - $p \wedge q$ : "La persona té febre i la persona ha de prendre paracetamol"

# Sistemes experts

# Aprofitament del coneixement humà

## Definició

- **Sistemes basats en el coneixement** (SBC) o **sistemes experts** (SE):
  - sistemes que utilitzen el coneixement humà per resoldre problemes.
- El coneixement humà s'expressa en forma de **regles**.
  - La majoria de sistemes experts utilitzen **regles de producció**.
- Els SBC són un **subconjunt** de la intel·ligència artificial.
  - Actualment es prefereixen els sistemes basats en dades.
  - Així i tot, compleixen un paper important en la IA.

# Característiques

- **Coneixement:**

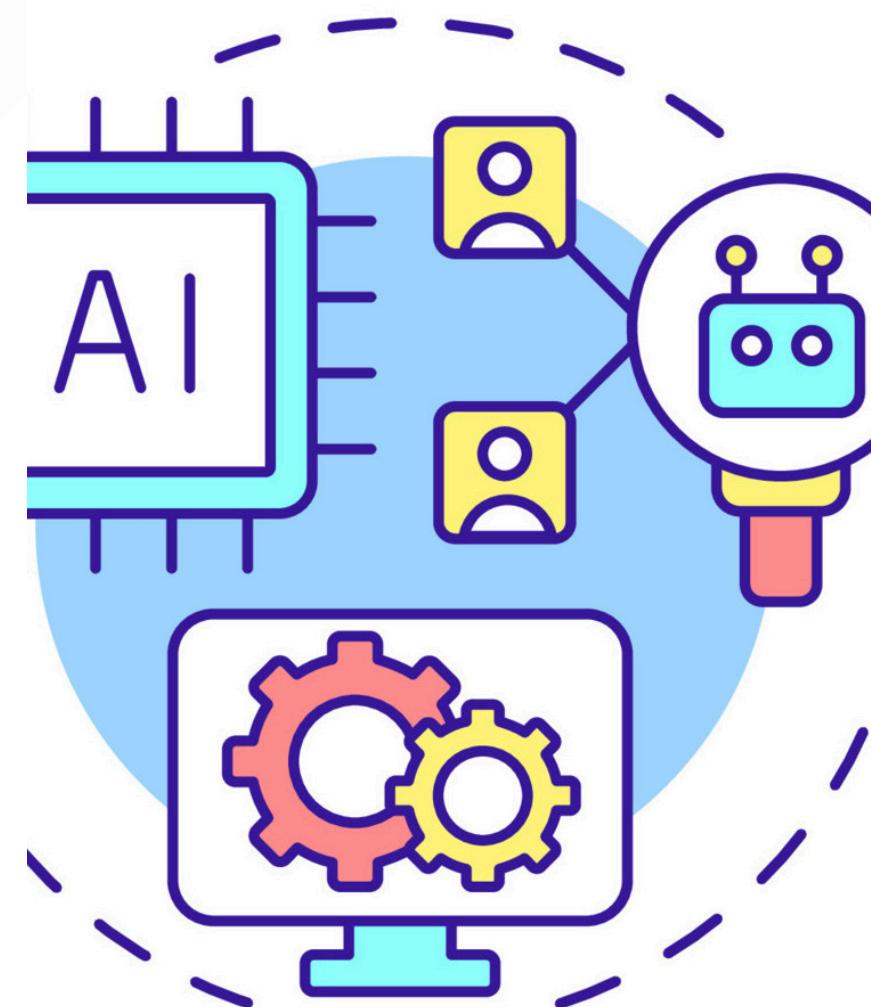
- El coneixement humà s'expressa en forma de regles.
- El coneixement és **declaratiu**.

- **Rendiment:**

- Els SBC són **especialistes** en un domini concret i **eficients** en ell.

- **Explicació:**

- Els SBC poden explicar el seu raonament i les seves conclusions.



# RULE-BASED EXPERT SYSTEMS

THE **MYCIN** EXPERIMENTS OF THE STANFORD  
HEURISTIC PROGRAMMING PROJECT

Bruce G. Buchanan  
Edward H. Shortliffe



# Sistemes basats en regles

# Definició

- És un tipus de sistema basat en el coneixement.
- Utilitza un conjunt de regles explícites per realitzar un raonament.
- Les regles són de la forma:
  - **IF** (premissa) **THEN** (conclusió)
- Es determinista
  - Per a una mateixa premissa sempre es dedueix la mateixa conclusió.
- Es pot representar en forma d'arbre de decisió.
  - Facilita la comprensió del raonament.

# Parts d'un sistema basat en regles (I)

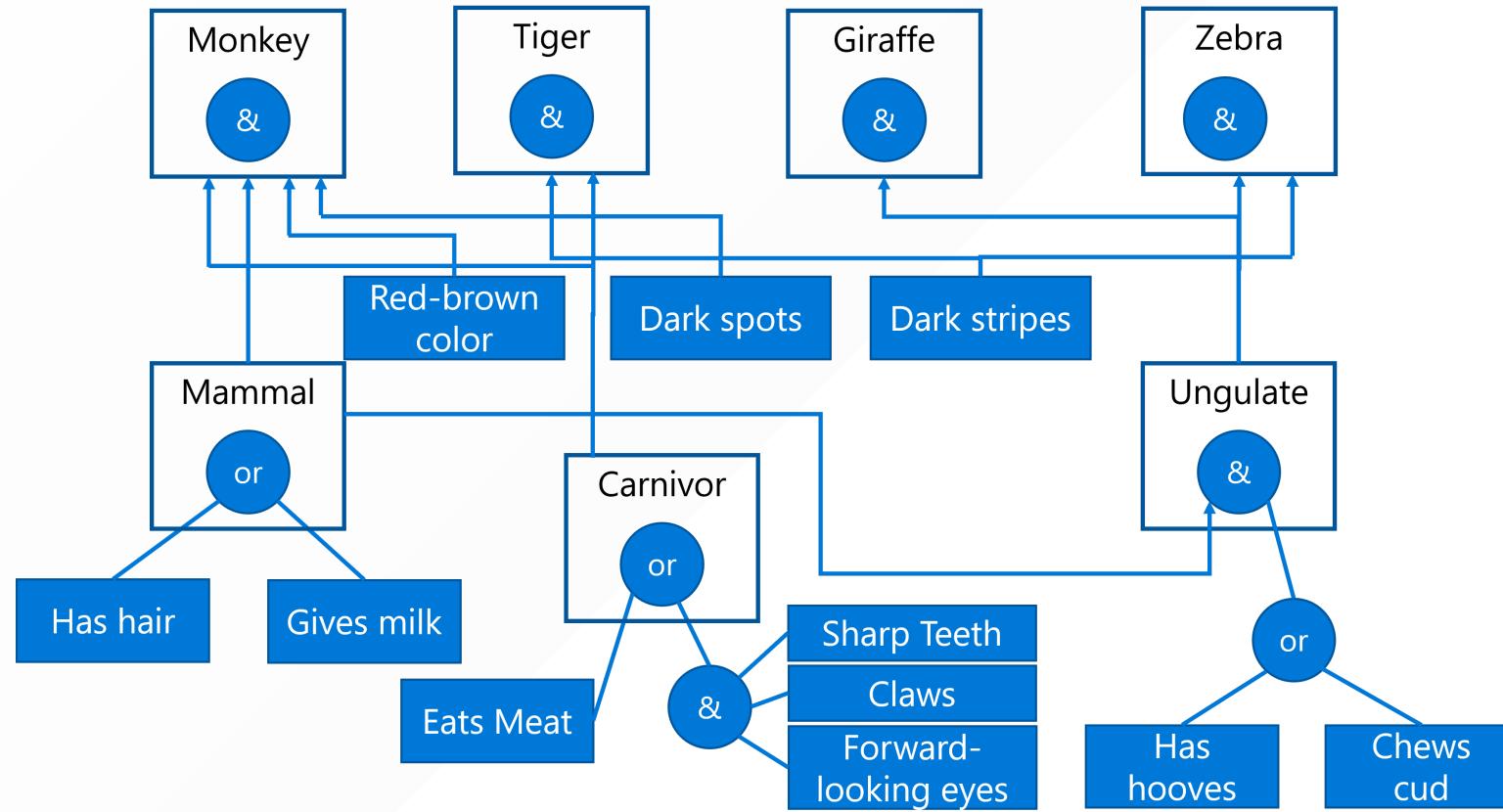
- **Memòria activa** (base de fets):
  - Base de dades de fets que descriuen la situació actual.  
*Volàtil.*
- **Base de coneixement:**
  - Conjunt de regles que descriuen el coneixement dels experts. *Persistent.*
- **Motor d'inferència:**
  - Busca les regles que s'apliquen a la situació actual (**conjunt conflictiu**) i les executa per ordre.



# **Parts d'un sistema basat en regles (II)**

- **Mitjans d'explicació:**
  - Permeten explicar el raonament del sistema a l'usuari.
- **Mitjans d'adquisició de coneixement:**
  - Permeten a l'usuari afegir noves regles al sistema. També es pot fer automàticament.
- **Interfície d'usuari:**
  - Permet a l'usuari interactuar amb el sistema.

# Exemple: Classificar un animal (I)



## Exemple: Classificar un animal (II)

**IF** (animal has hair **OR** animal gives milk) **THEN** the animal is a mammal

**IF** the animal eats meat **OR** (animal has sharp teeth **AND** animal has claws **AND** animal has forward-looking eyes ) **THEN** the animal is a carnivore

- Els elements de la premissa són **antecedents**, semblants a triplets objecte-atribut-valor.
- La **memòria activa** conté els fets que descriuen la situació actual.
- El **sistema de regles** mira que antecedents es compleixen i aplica les seves conclusions, agregant-les a la **memòria activa**.

# **Exemple: Classificar un animal (III)**

- *Observacions:*
  - Son necessàries moltes regles per cobrir tots els casos.
  - Les regles són difícils de mantenir.
  - Es fàcil que les regles entrin en contradicció.
    - Com representariem l'ornitorrinc?. És un mamífer o un ocell?

## **Exemple: Dolor de queixal (I)**

- **IF** (tinc dolor de queixal) **THEN** (tinc càries)
- **IF** (tinc dolor de queixal) **THEN** (tinc una infecció)
- **IF** (tinc dolor de queixal) **THEN** (tinc una sensibilitat)
- **IF** (tinc dolor de queixal) **THEN** (tinc una fractura)
- **IF** (tinc dolor de queixal) **THEN** (tinc una maloclusió)
- **IF** (tinc dolor de queixal) **THEN** (tinc una sinusitis)
- **IF** (la geniva està *més roja*) **THEN** (tinc una infecció)

# Exemple: Dolor de queixal (II)

- *Observacions:*
  - El dolor de queixal pot ser causat per moltes raons.
  - No es pot determinar la causa amb una única regla.
    - Ens falten eines per gestionar la **incertesa**.
  - No podem representar la regla "*si la geniva està més roja*".
    - Ens falten eines per representar el **coneixement imprecís**.

# Problemes

- Necesiten una **gran quantitat de coneixement expert**.
- El coneixement expert és difícil d'**obtenir, representar i mantenir**.
- Raonament en **incertesa**:
  - Els sistemes basats en regles solament poden treballar amb les **regles definides**
  - Dificultat per treballar amb dades **imprecises o incompletes**.
- Flexibilitat:
  - Dificultat per **adaptar-se** a nous problemes.
  - Dificultat per **aprendre** nous coneixements.

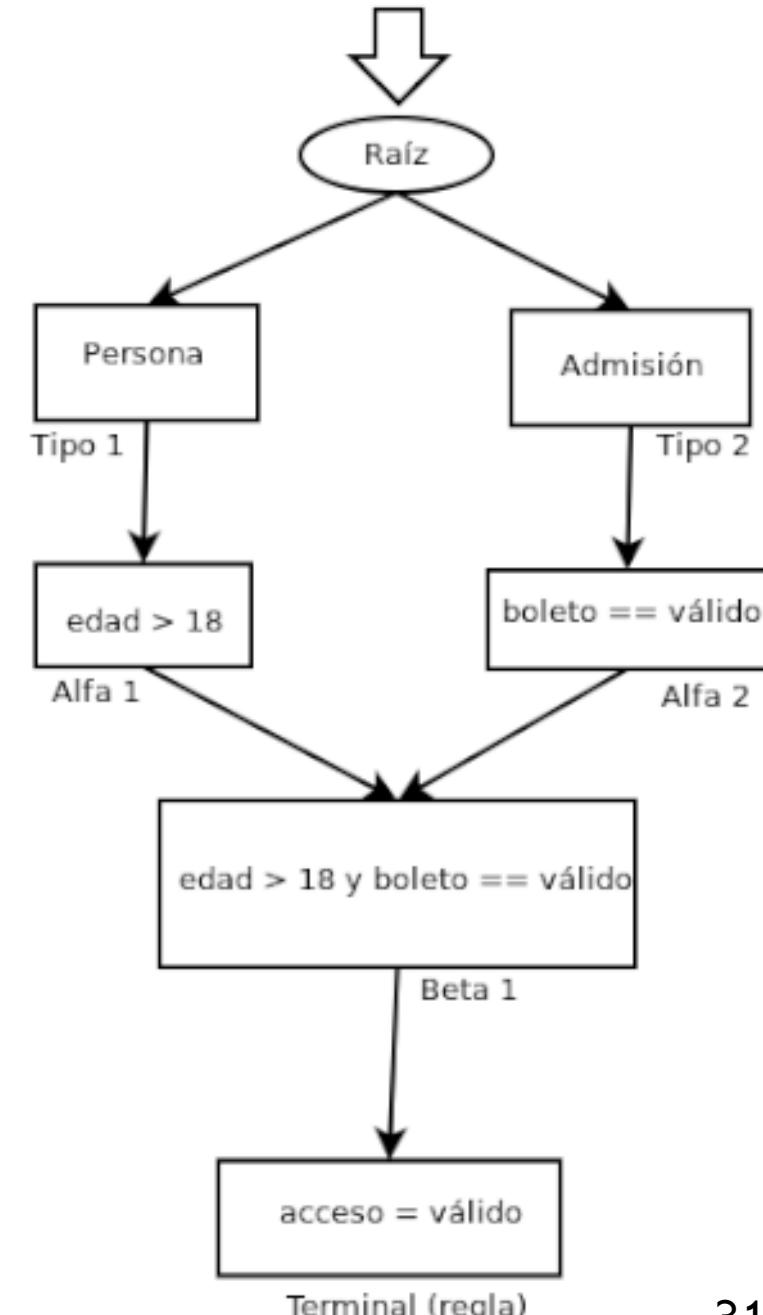
# Estratègies d'inferència

## Encadenament cap endavant: *forward chaining* (I)

- Revisa els antecedents de les regles per buscar coincidències en els fets i inferir noves conclusions.
  - Raonament **deductiu**.
- Basat en la **lògica proposicional**.
  - Modus ponens:  $p \rightarrow q, p \vdash q$
- **Problema**: Pot no trobar la conclusió.
  - Exemple: "*Si el gos està malalt, doncs el gos està malalt*"

## **Forward chaining (II)**

- El raonament **no ve guiat per la conclusió**.
  - Es troben totes les conclusions possibles (relevants o no).
  - Es fa més treball del necessari.
- L'**estratègia de resolució** de conflictes és fonamental
  - Es pot millorar en heuristiques
- **Problema:** Detecció de les regles que s'han de disparar.
  - Algorisme RETE (Forgy, 1979)



# Estratègies d'inferència

## Encadenament cap enrere: *backward chaining* (I)

- Comença amb la conclusió i busca els antecedents que la justifiquen.
  - Raonament **inductiu**.
  - Utilitza el **modus ponens a l'inrevés**
  - $p \rightarrow q, q \vdash p$
- Els objectius determinen les regles a aplicar.
- El raonament ve **guiat per la conclusió**.

[ croaks and eats flies – Then X is  
[ chirps and sings – Then X is a canary  
[ is a frog – Then X is green  
[ is a canary – Then X is yellow

Looking for what color your pet is there are two options.

[ croaks and eats flies – Then X is  
[ chirps and sings – Then X is a canary  
[ is a frog – Then X is green  
[ is a canary – Then X is yellow

First option.

[ croaks and eats flies – Then X is  
[ chirps and sings – Then X is a canary  
[ is a frog – Then X is green  
[ is a canary – Then X is yellow

Through the list and see if you can find if X is a frog.

[ croaks and eats flies – Then X is  
[ chirps and sings – Then X is a canary  
[ is a frog – Then X is green  
[ is a canary – Then X is yellow

In step 1. X croaks and eats flies is given as true. Since X croaks and eats flies, X is a frog. Since X is a frog, X is green.

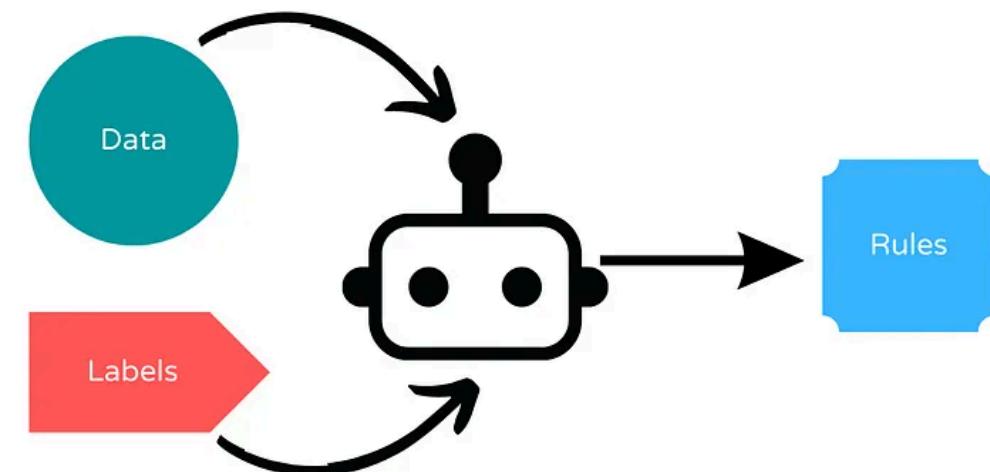
# Plataformes per a sistemes basats en regles

- **CLIPS** (C Language Integrated Production System):  
Llenguatge de programació i motor d'inferència.
- **Drools**: Llenguatge de programació i motor d'inferència.
- **Prolog**: Llenguatge de programació lògica.
- **Python**
  - Llibreries *PyKnow*, *PyKE* i *Expert*a.
  - Llibreria *PyCLIPS*.
  - Llibreria *PyDrools*.

# Sistemes híbrids

## Regles/Dades (I)

- Dos enfocaments:
  - Deducció de regles a partir de dades. - Facilita la **interpretació** del raonament.
  - Integració de regles definides per l'usuari i Aprendentatge Automàtic.
    - Permet definir unes regles que es poden **millorar** amb l'aprenentatge automàtic.



# Sistemes híbrids Regles/Dades (II)

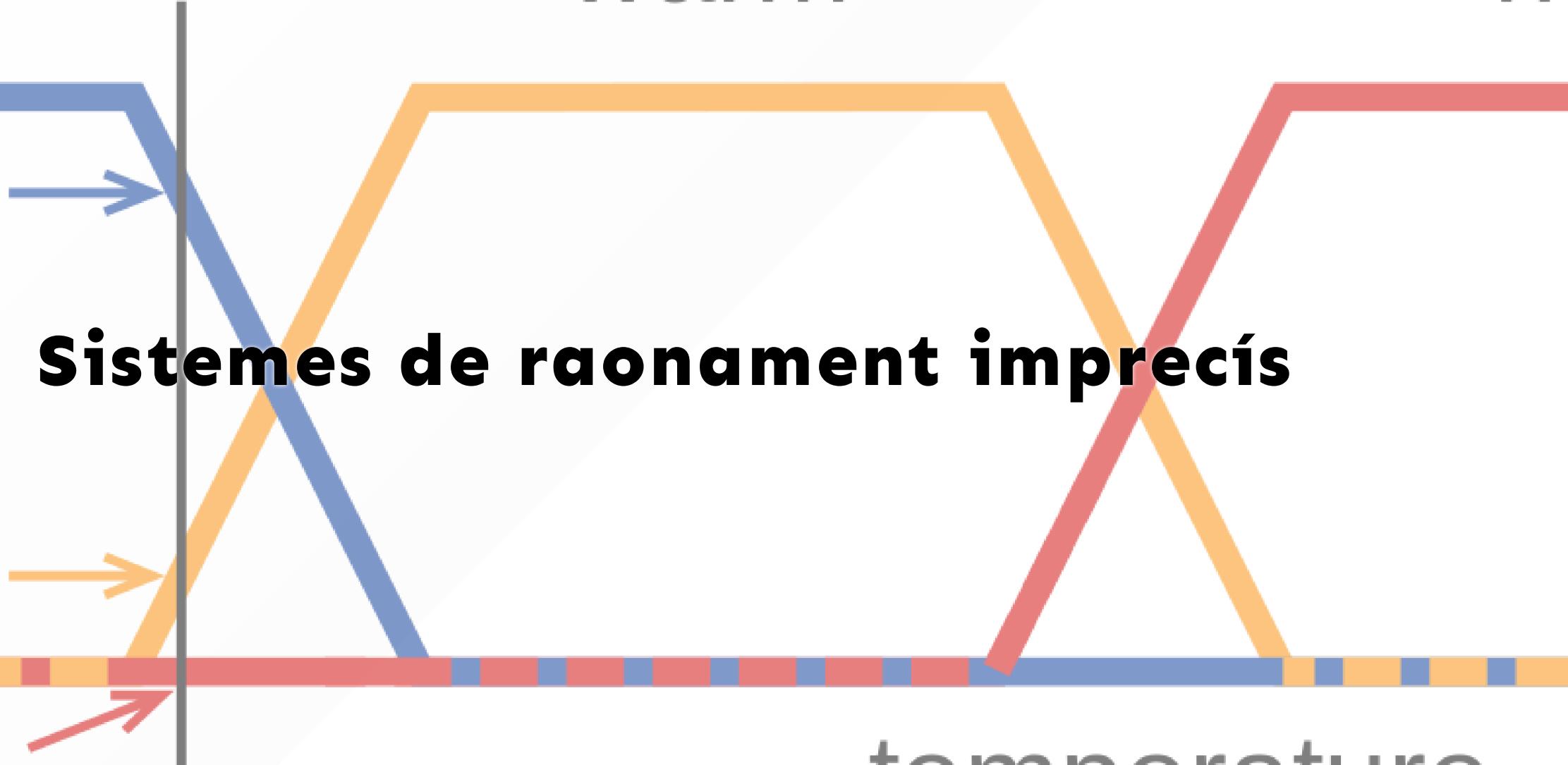
## Libreries

- [Human-Learn](#): Permet definir i dibuixar regles que es poden millorar amb l'aprenentatge automàtic.
- [skope-rules](#): Analitza les dades i dedueix regles per a classificar.
  - Permet analitzar les regles per millorar-les i interpretar-les.
- [SpaCy](#):
  - Permet definir regles per a l'extracció d'informació per a textos.
  - Útil en casos on no es disposa de prou dades etiquetades o per casos específics.

cold

warm

hot

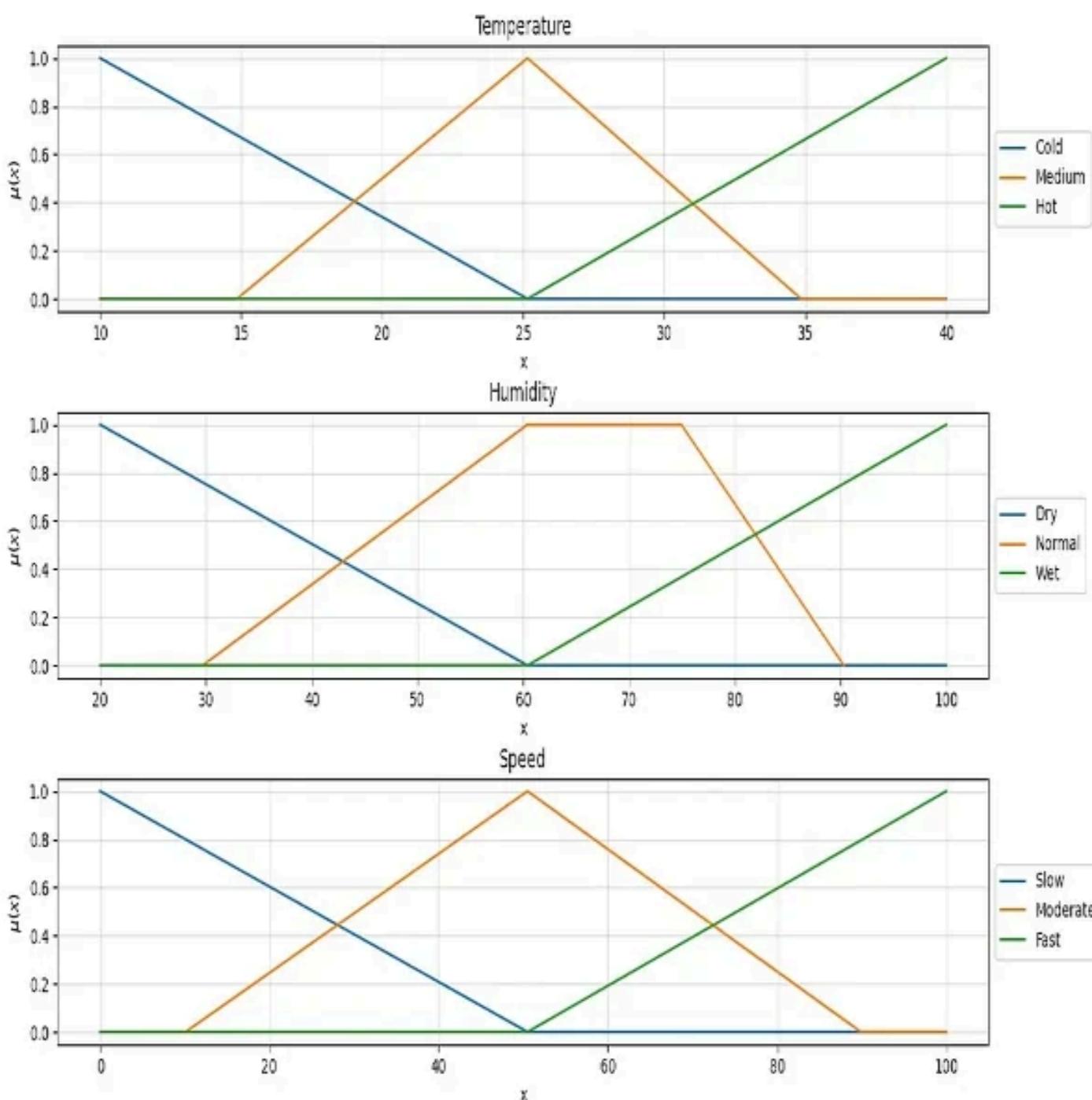


# Definició

- **Lògica difusa o lògica borrosa:**
  - Extensió de la lògica proposicional per a treballar amb la incertesa.
  - Permet treballar amb valors imprecisos.
- **Sistemes de raonament imprecís:**
  - Sistemes basats en regles que utilitzen la lògica difusa.
  - Permeten treballar amb valors **continus**.
  - Faciliten modelar el **coneixement humà**.
  - Molt apropiats per a **sistemes de control**
  - Ens permeten tindre una **bona solució**, si no la **millor**.

# Lògica difusa (I)

- La lògica proposicional és **binària**.
  - Un enunciat és **cert** o **fals**.
- La lògica difusa permet treballar amb valors **continus**.
  - Un enunciat pot ser **cert i fals** en un grau **parcial**.
- Els valors de veritat són **nombres reals** en l'interval  $[0, 1]$ .
  - $0 : Fals$ ,  $1 : Cert$ ,  $0.5 : Cert$  en un 50%
- La pertinença d'un element a un conjunt vindrà donada per una **funció de pertinença**.
  - $\mu_A(x)$ : Grau de pertinença d' $x$  al conjunt  $A$ .



# Lògica difusa (II)

- La lògica difusa facilita la **representació del coneixement humà**.
  - Els humans no raonem en termes binaris.
  - Els humans no tenim un coneixement precís ni complet.
- Conceptes com \$\$humit\$\$ o \$\$fred\$\$ són difícils de definir amb precisió.
  - La lògica difusa ens permet definir-los amb **funcions de pertinença**.
  - El poder treballar amb aquests conceptes facilita la creació de dispositius com **assecadors** o **termòstats**.
    - *"Si la temperatura és freda, llavors encén la calefacció"*

# Conceptes bàsics (I)

- **Variable lingüística:** Variable que pot prendre valors lingüístics.
  - Exemple: *Temperatura*
- **Valors lingüístics:** Valors que pot prendre una variable lingüística.
  - Exemple: *Fred, Calor*
- **Funció de pertinença:** Funció que assigna a cada valor d'una variable lingüística un grau de pertinença a un valor lingüístic.
  - Exemple:  
 $Temperatura = 27^\circ C \rightarrow Calor = 0.8, Molta Calor = 0.2$

# Conceptes bàsics (II)

- **Regla difusa:** Regla que utilitza valors difusos.
  - Exemple: "*Si la temperatura és freda, llavors calefacció alta*"
- **Funció d'agregació:** Funció que combina els valors difusos de les regles per a deduir la conclusió final.
  - Exemple:  
 $Calor = 0.8, Humit = 0.7 \rightarrow Sensacio desagradable = 0.8$
- **Sistema de raonament imprecís:**
  - Sistema basat en regles que utilitza la lògica difusa.
  - Exemple: Sistema de control de la temperatura d'un habitatge.

# Funcionament dels sistemes de raonament imprecís (I)

- *Fuzzyfication:*
  - Conversió de les dades d'entrada precises a valors difusos.
  - Passem de valors precisos a valors difusos.
  - Utilitza les **funcions de pertinença**.
    - Assignen a cada valor d'entrada un grau de pertinença a cada **variable lingüística**
    - $27^{\circ}C \rightarrow Calor = 0.8, Molta calor = 0.2$

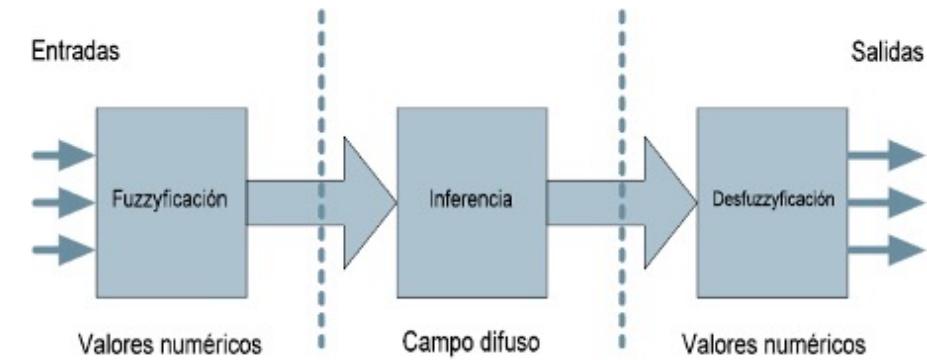
# Funcionament dels sistemes de raonament imprecís (II)

- *Evaluació de les regles:*
  - En aquest pas s'**apliquen les regles del sistema**.
  - S'estableix la relació entre les **variables d'entrada** i les **variables de sortida**.
  - "*Si la temperatura és **alta** i la humitat és **baixa**, llavors la velocitat del ventilador ha de ser **alta***"
  - Es combinen les **funcions de pertinença** de les variables **d'entrada**
    - per a deduir la **pertinença** de la variable de **sortida**.

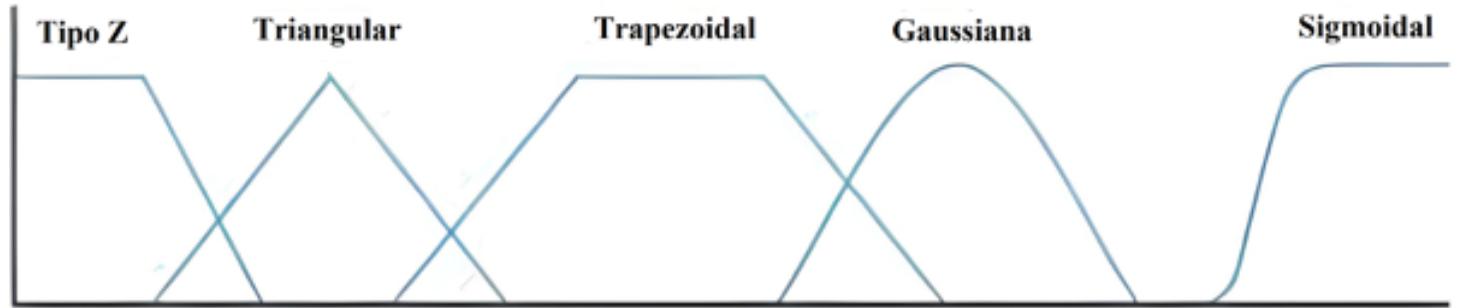
# Funcionament dels sistemes de raonament imprecís (III)

- *Defuzzyfication:*

- Conversió de les dades de sortida difuses a valors precisos.
- Passem de valors difusos a valors precisos.
- Utilitza les **funcions d'agregació**.
  - Combina les conclusions de les regles per a deduir la conclusió final.
  - Es sol utilitzar la funció de **centre de gravetat o màxim**.



# Funcions de pertinença (I)



Funciones de pertenencia

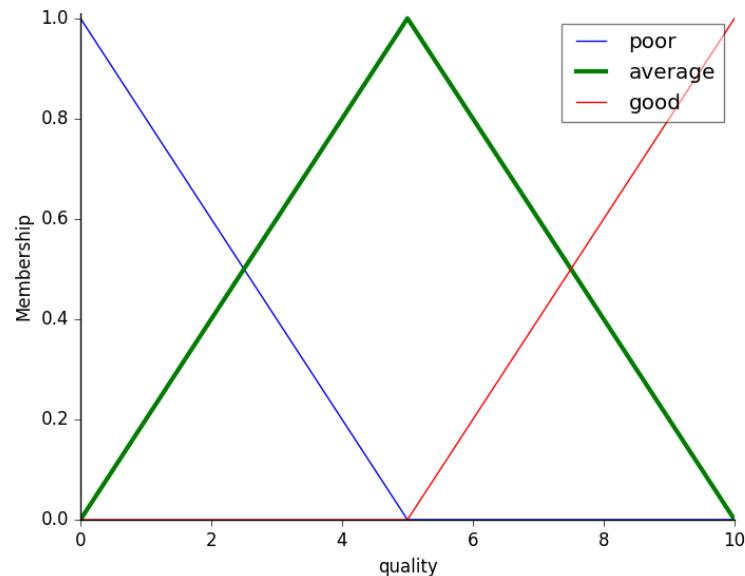
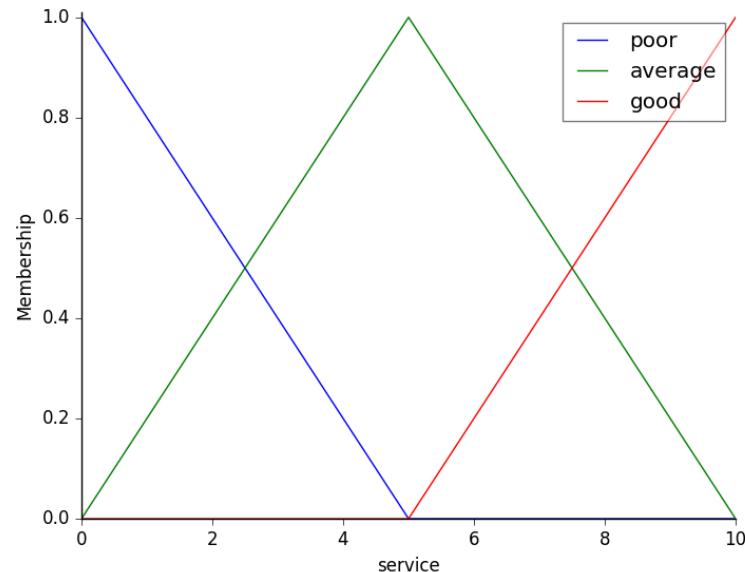
- Les més utilitzades són les **funcions trapezoïdals** i les **funcions triangulares**.
- Les sinusoïdals són útils per a representar **períodes**.
- Les sigmoidals són útils per a representar **probabilitats**.

# Exemple: Propines (I)

## Variables d'entrada

Utilitzarem funcions triangulars per a representar les variables d'entrada i sortida

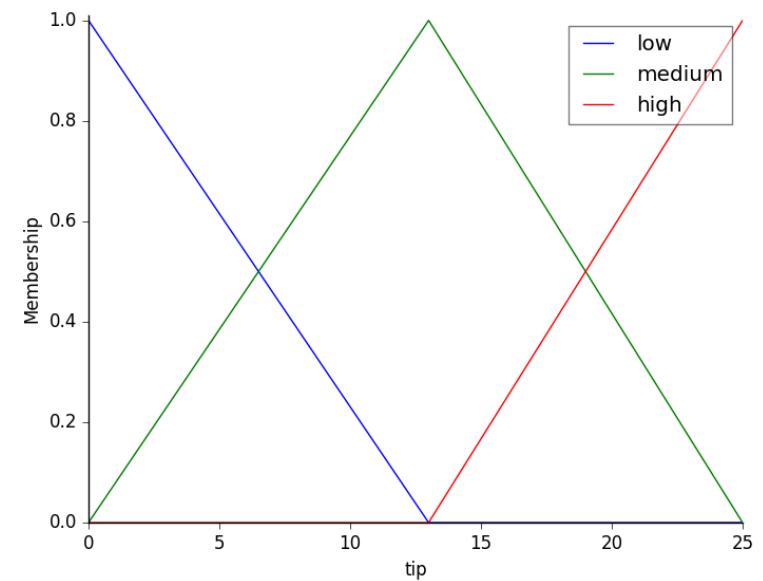
- **Servei:**
  - **Baixa:** [0, 5]
  - **Mitjana:** [0, 10]
  - **Alta:** [5, 10]
- **Qualitat del menjar:**
  - **Baix:** [0, 5]
  - **Mitjà:** [0, 10]
  - **Alt:** [5, 10]



# Exemple: Propines (II)

## Variables de sortida

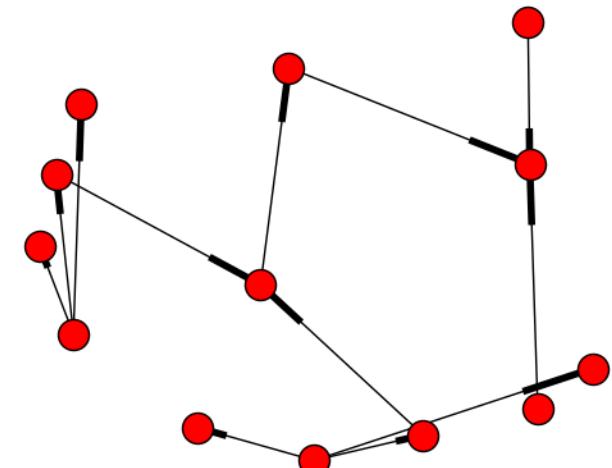
- **Propina:**
  - **Baixa:** [0, 13]
  - **Mitjana:** [0, 25]
  - **Alta:** [13, 25]



# Exemple: Propines (III)

## Regles

- **IF** (Qualitat del servei és **baixa** o Menjar és **baix**) **THEN** (Propina és **baixa**)
- **IF** (Qualitat del servei és **mitjana**) **THEN** (Propina és **mitjana**)
- **IF** (Qualitat del servei és **alta** o Menjar és **alt**) **THEN** (Propina és **alta**)



# Exemple: Propines

## (IV)

### Inferència

- Qualitat del servei: **9.8**
- Qualitat del menjar: **6.5**
- Propina: **20.24%**

