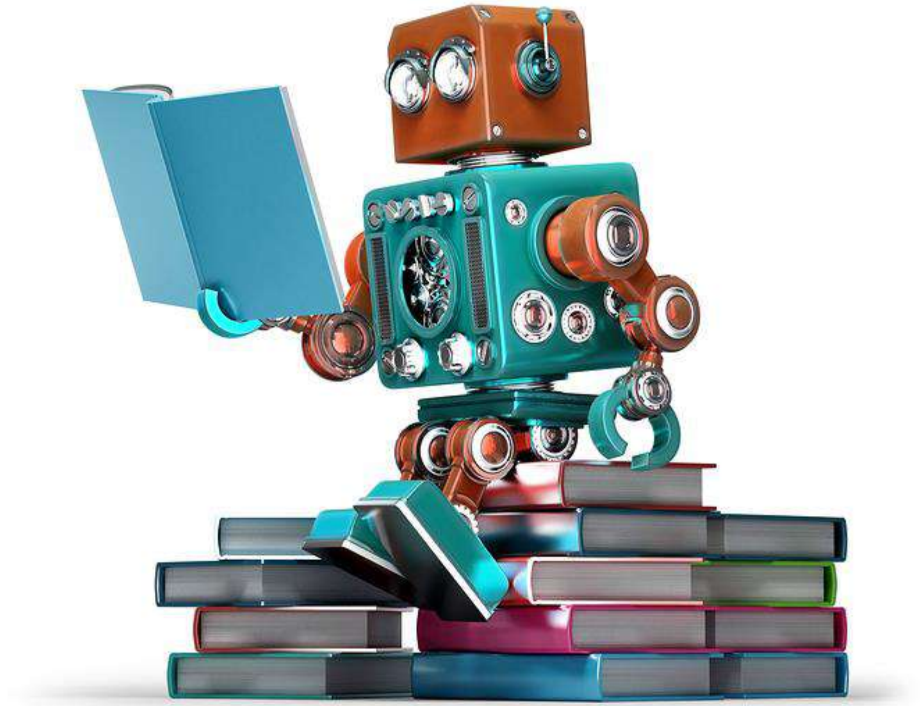
MACHINE REASONING

DAY 3



https://robohub.org/wp-content/uploads/2016/11/bigstock-Retro-Robot-Reading-A-Book-ls-110707406.jpg

DAY 3 AGENDA

3.1 Machine Inference (part 2)

{ Course **Assessment 1** }

3.2 Inference under Uncertainty

3.3 Knowledge Discovery by Machine Learning

3.4 Knowledge Discovery **Workshop**

DAY 3 TIMETABLE

No	Time	Topic	By Whom	Where
1	9 am	3.1 Machine Inference (2/2)	GU Zhan (Sam)	Class
2	10.10 am	Morning Break		
3	10.30 am	{ Course Assessment 1 } 3.2 Inference under Uncertainty	GU Zhan (Sam)	Class
4	12.10 pm	Lunch Break		
5	1.30 pm	3.3 Knowledge Discovery by Machine Learning	GU Zhan (Sam)	Class
6	3.10 pm	Afternoon Break		
7	3.30 pm	3.4 Knowledge Discovery Workshop	All	Class
8	4.50 pm	Summary and Review	All	Class
9	5 pm	End		

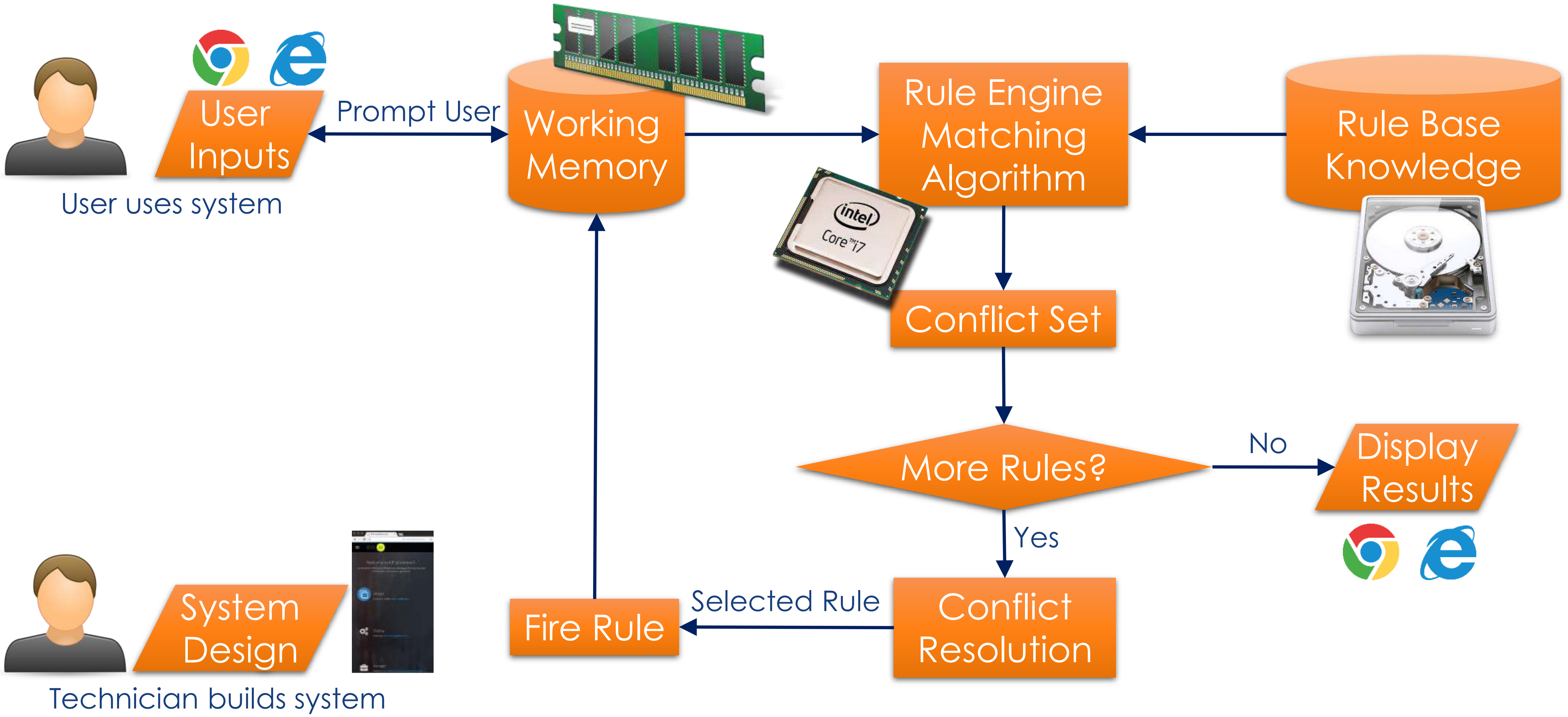
3.1

MACHINE INFERENCE

(PART 2)

3.1 MACHINE INFERENCE

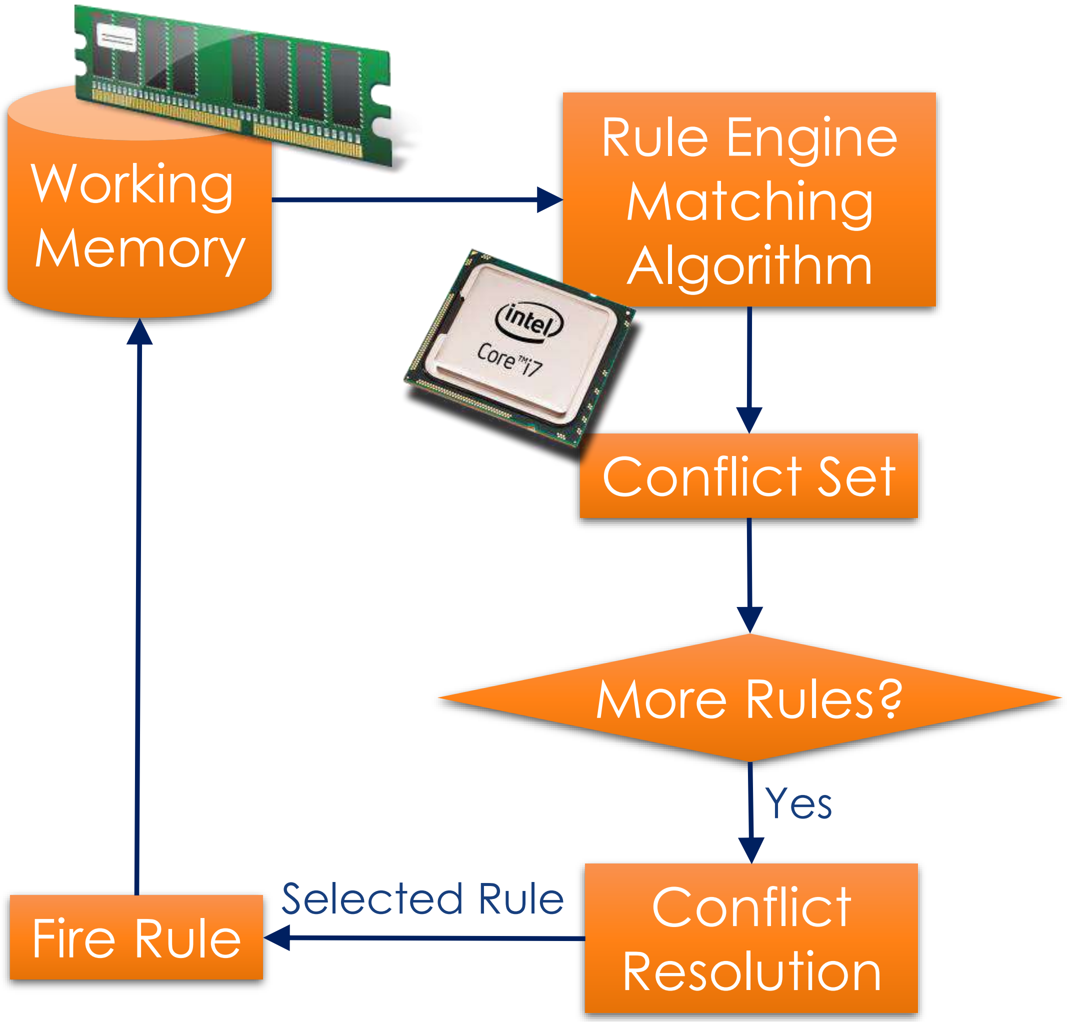
[AI] Recognise-Act Control Cycle



3.1 MACHINE INFERENCE

Conflict Resolution

- **Conflict Set**
  - More than one rule can fire based on the facts in WM. That is, the fact in WM can match more than one rule at a specific time. The matched activated rules represents a **conflict set**.
- **Conflict Resolution**
  - A method for choosing a rule to fire when more than one rule can be fired in a given cycle.



3.1 MACHINE INFERENCE

Conflict Resolution Example

Rule 1: **IF** I have at least $20 AND Man-United is playing today
THEN I should go to the football game.

Rule 2: **IF** it is raining today AND I don't have school
THEN I should stay home

Rule 3: **IF** I have at least $20
THEN I should go to the cinema.

Rule 4: **IF** I should go to the cinema
THEN I should call my friends

Initial Facts: **Fact-1**: I have $20
Fact-2: Man-United is playing today

Conflict Set: <R1: Fact-1, Fact-2>, <R3: Fact-1>

3.1 MACHINE INFERENCE
Conflict Resolution Strategy

- **The order in which the rule fires depends on facts in Working Memory, (in general) not the order of rules.**
- **Rule firing priority:**
  - Default is last in first out (**LIFO**) based on facts in WM
  - Attach priority to rules, and select the rule with the highest priority (**Salience**)
  - Order the facts by the length of time they have been in working memory, and select the most recent. (**Recency**)
  - Select rules which required the lowest/highest number of facts/rule-conditions (**Specificity**).

3.1 MACHINE INFERENCE

Conflict Resolution Strategy – KIE Drools

- **Salience / Specificity / Recency / LIFO / FIFO**

  Individual rule's priority **in Agenda**

- **AgendaGroup**

  It allow you to place rules into groups, and to place those groups onto a stack (**rule set's priority**). The stack has push/pop behaviour.

- **ActivationGroup**

  It is a set of rules bound together by the same "activation-group" rule attribute. In this group **only one rule can fire**, and after that rule has fired all the other rules are cancelled from the agenda.

- **RuleFlowGroup**

  It is a group of rules associated by the "ruleflow-group" rule attribute. These rules can only fire when the group is activated. (**jBPM Business Rule Task**)

```
rule "Print balance for AccountPeriod"
        salience -50
    when
        ap : AccountPeriod()
        acc : Account()
    then
        System.out.println( acc.accountNo + " : " + acc.balance );
end
```

```
rule "increase balance for credits"
  agenda-group "calculation"
when
  ap : AccountPeriod()
  acc : Account( $accountNo : accountNo )
  CashFlow( type == CREDIT,
            accountNo == $accountNo,
            date >= ap.start && <= ap.end,
            $amount : amount )
then
  acc.balance  += $amount;
end
```

```
rule "Print balance for AccountPeriod"
  agenda-group "report"
when
  ap : AccountPeriod()
  acc : Account()
then
  System.out.println( acc.accountNo +
                      " : " + acc.balance );
end
```

Source: https://docs.jboss.org/drools/release/latest/drools-docs/html_single/index.html#_conflict_resolution_2

3.1 MACHINE INFERENCE

Exercise 3.1

- **Using the Rules from previous Exercise 2.1, work out the following problems:**
- **Q1: Given these facts in Working Memory:**

  the animal gives milk, the animal eats grass, the animal has long legs, the animal has a long neck

  Goal: To establish by **forward chaining** that the animal is a giraffe. If you are not able to establish this, what is the rule(s) that you can add into the Knowledge/Rule Base to successfully perform the chaining (inference)?
- **Q2: Given these facts in Working Memory:**

  the animal has hair, the animal has claws, the animal has pointed teeth, the animal's eyes point forward, the animal has a tawny color, the animal has dark spots

  Goal: To establish by **backward chaining** that the animal is a cheetah

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: propositional logic

1. WHEN BodyCoverType = HasHair THEN AnimalType = Mammal
2. WHEN FeedType = FeedMilk THEN AnimalType = Mammal
3. WHEN BodyCoverType = HasFeather THEN AnimalType = Bird
4. WHEN MoveType = CanFly AND ReproduceType = LayEgg THEN AnimalType = Bird
5. WHEN AnimalType = Mammal AND FeedType = EatMeat THEN AnimalType = Carnivore
6. WHEN AnimalType = Mammal AND ToothType = HasPointedTeeth AND FootType = HasClaws AND EyeType = HasForwardEyes THEN AnimalType = Carnivore
7. WHEN AnimalType = Mammal AND FeedType = EatGrass THEN AnimalType = Herbivore
8. WHEN AnimalType = Mammal AND FootType = HasHooves THEN AnimalType = Herbivore
9. WHEN AnimalType = Carnivore AND ColorType = HasColorTawny AND MarkingType = HasDarkSpots THEN AnimalName = Cheetah
10. WHEN AnimalType = Carnivore AND ColorType = HasColorTawny AND MarkingType = HasDarkStripes THEN AnimalName = Tiger
11. WHEN AnimalType = Herbivore AND ColorType = HasColorTawny AND MarkingType = HasDarkSpots AND NeckType = HasLongNeck THEN AnimalName = Giraffe
12. WHEN AnimalType = Herbivore AND ColorType = HasColorBlackWhite THEN AnimalName = Zebra
13. WHEN AnimalType = Bird AND MoveType = CanWalk AND ColorType = HasColorBlackWhite AND NeckType = HasLongNeck THEN AnimalName = Ostrich
14. WHEN AnimalType = Bird AND MoveType = CanWwim AND ColorType = HasColorBlackWhite THEN AnimalName = Penguin
15. WHEN AnimalType = Bird AND MoveType = CanFly AND ColorType = HasColorBlackWhite THEN AnimalName = Albatross

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

X is an (instance of) animal (class).

1. HasHair(X) → Mammal(X)
2. FeedMilk(X) → Mammal(X)
3. HasFeather(X) → Bird(X)
4. CanFly(X) ∧ LayEgg(X) → Bird(X)
5. Mammal(X) ∧ EatMeat(X) → Carnivore(X)
6. Mammal(X) ∧ HasPointedTeeth(X) ∧ HasClaws(X) ∧ HasForwardEyes(X) → Carnivore(X)
7. Mammal(X) ∧ EatGrass(X) → Herbivore(X)
8. Mammal(X) ∧ HasHooves(X) → Herbivore(X)
9. Carnivore(X) ∧ HasColorTawny(X) ∧ HasDarkSpots(X) → Cheetah(X)
10. Carnivore(X) ∧ HasColorTawny(X) ∧ HasDarkStripes(X) → Tiger(X)
11. Herbivore(X) ∧ HasColorTawny(X) ∧ HasDarkSpots(X) ∧ HasLongNeck(X) → Giraffe(X)
12. Herbivore(X) ∧ HasColorBlackWhite(X) → Zebra(X)
13. Bird(X) ∧ CanWalk(X) ∧ HasColorBlackWhite(X) ∧ HasLongNeck(X) → Ostrich(X)
14. Bird(X) ∧ CanSwim(X) ∧ HasColorBlackWhite(X) → Penguin(X)
15. Bird(X) ∧ CanFly(X) ∧ HasColorBlackWhite(X) → Albatross(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

- **Knowledge Base (KB)**
  - Rule sets plus below Facts:
  - HasHair(X)
  - HasClaws(X)
  - HasPointedTeeth(X)
  - HasForwardEyes(X)
  - HasColorTawny(X)
  - HasDarkSpots(X)
- **Clause form conversion :** $p \rightarrow q \equiv \neg p \vee q$
- **Hypothesis to prove is :** a = Cheetah(X)
- **Refutation of hypothesis is :** $\neg a = \neg$Cheetah(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. HasHair(X) → Mammal(X)
2. FeedMilk(X) → Mammal(X)
3. HasFeather(X) → Bird(X)
4. CanFly(X) ∧ LayEgg(X) → Bird(X)
5. Mammal(X) ∧ EatMeat(X) → Carnivore(X)
6. Mammal(X) ∧ HasPointedTeeth(X) ∧ HasClaws(X) ∧ HasForwardEyes(X) → Carnivore(X)
7. Mammal(X) ∧ EatGrass(X) → Herbivore(X)
8. Mammal(X) ∧ HasHooves(X) → Herbivore(X)
9. Carnivore(X) ∧ HasColorTawny(X) ∧ HasDarkSpots(X) → Cheetah(X)
10. Carnivore(X) ∧ HasColorTawny(X) ∧ HasDarkStripes(X) → Tiger(X)
11. Herbivore(X) ∧ HasColorTawny(X) ∧ HasDarkSpots(X) ∧ HasLongNeck(X) → Giraffe(X)
12. Herbivore(X) ∧ HasColorBlackWhite(X) → Zebra(X)
13. Bird(X) ∧ CanWalk(X) ∧ HasColorBlackWhite(X) ∧ HasLongNeck(X) → Ostrich(X)
14. Bird(X) ∧ CanSwim(X) ∧ HasColorBlackWhite(X) → Penguin(X)
15. Bird(X) ∧ CanFly(X) ∧ HasColorBlackWhite(X) → Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v Carnivore(X)
6. ¬ Mammal(X) v ¬ HasPointedTeeth(X) v ¬ HasClaws(X) v ¬ HasForwardEyes(X) v Carnivore(X)
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v Cheetah(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v Carnivore(X)
6. ¬ Mammal(X) v ¬ HasPointedTeeth(X) v ¬ HasClaws(X) v ¬ HasForwardEyes(X) v Carnivore(X)
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v Cheetah(X) ^ ¬ Cheetah(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v Carnivore(X)
6. ¬ Mammal(X) v ¬ HasPointedTeeth(X) v ¬ HasClaws(X) v ¬ HasForwardEyes(X) v Carnivore(X)
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v {}
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v Carnivore(X)
6. ¬ Mammal(X) v ¬ HasPointedTeeth(X) v ¬ HasClaws(X) v ¬ HasForwardEyes(X) v Carnivore(X)
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v Carnivore(X)
6. ¬ Mammal(X) v ¬ HasPointedTeeth(X) v ¬ HasClaws(X) v ¬ HasForwardEyes(X) v Carnivore(X)
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) ^ HasDarkSpots(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v Carnivore(X)
6. ¬ Mammal(X) v ¬ HasPointedTeeth(X) v ¬ HasClaws(X) v ¬ HasForwardEyes(X) v Carnivore(X)
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X) v ¬ HasColorTawny(X) v {}
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v Carnivore(X)
6. ¬ Mammal(X) v ¬ HasPointedTeeth(X) v ¬ HasClaws(X) v ¬ HasForwardEyes(X) v Carnivore(X)
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X) v ¬ HasColorTawny(X) ^ HasColorTawny(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v Carnivore(X)
6. ¬ Mammal(X) v ¬ HasPointedTeeth(X) v ¬ HasClaws(X) v ¬ HasForwardEyes(X) v Carnivore(X)
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X) v {}
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v Carnivore(X)
6. ¬ Mammal(X) v ¬ HasPointedTeeth(X) v ¬ HasClaws(X) v ¬ HasForwardEyes(X) v Carnivore(X)
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)
¬ Carnivore(X)

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v Carnivore(X) ^ ¬ Carnivore(X)
6. ¬ Mammal(X) v ¬ HasPointedTeeth(X) v ¬ HasClaws(X) v ¬ HasForwardEyes(X) v Carnivore(X) ^ ¬ Carnivore(X)
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)
¬ Carnivore(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v {} ← branch 2
6. ¬ Mammal(X) v ¬ HasPointedTeeth(X) v ¬ HasClaws(X) v ¬ HasForwardEyes(X) v {} ← branch 1
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)
¬ Carnivore(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v {} ← branch 2
6. ¬ Mammal(X) v ¬ HasPointedTeeth(X) v ¬ HasClaws(X) v ¬ HasForwardEyes(X) v {} ← branch 1
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)
¬ Carnivore(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v {} ← branch 2
6. ¬ Mammal(X) v {} ← branch 1
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Which branch to search/check next? Use conflict resolution strategy, ***e.g. Less No. Sub-goals***

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)
¬ Carnivore(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X)
2. ¬ FeedMilk(X) v Mammal(X)
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v {} ← branch 2
6. ¬ Mammal(X) ← branch 1
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)
¬ Carnivore(X)
¬ Mammal(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v Mammal(X) ^ ¬ Mammal(X) ← branch 1.1
2. ¬ FeedMilk(X) v Mammal(X) ^ ¬ Mammal(X) ← branch 1.2
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v {} ← branch 2
6. ¬ Mammal(X) ← branch 1
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)
¬ Carnivore(X)
¬ Mammal(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) v {} ← branch 1.1
2. ¬ FeedMilk(X) v {} ← branch 1.2
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v {} ← branch 2
6. ¬ Mammal(X) ← branch 1
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)
¬ Carnivore(X)
¬ Mammal(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. ¬ HasHair(X) ^ HasHair(X) ← branch 1.1
2. ¬ FeedMilk(X) v {} ← branch 1.2
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v {} ← branch 2
6. ¬ Mammal(X) ← branch 1
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)
¬ Carnivore(X)
¬ Mammal(X)

Exercise 3.1 – A possible solution: first order logic

1. {} ← branch 1.1
2. ¬ FeedMilk(X) v {} ← branch 1.2
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v {} ← branch 2
6. ¬ Mammal(X) ← branch 1
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)
¬ Carnivore(X)
¬ Mammal(X)

3.1 MACHINE INFERENCE

Exercise 3.1 – A possible solution: first order logic

1. {} ← a = Cheetah(X) proved by refutation ← branch 1.1
2. ¬ FeedMilk(X) v {} ← branch 1.2
3. ¬ HasFeather(X) v Bird(X)
4. ¬ CanFly(X) v ¬ LayEgg(X) v Bird(X)
5. ¬ Mammal(X) v ¬ EatMeat(X) v {} ← branch 2
6. ¬ Mammal(X) ← branch 1
7. ¬ Mammal(X) v ¬ EatGrass(X) v Herbivore(X)
8. ¬ Mammal(X) v ¬ HasHooves(X) v Herbivore(X)
9. ¬ Carnivore(X)
10. ¬ Carnivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkStripes(X) v Tiger(X)
11. ¬ Herbivore(X) v ¬ HasColorTawny(X) v ¬ HasDarkSpots(X) v ¬ HasLongNeck(X) v Giraffe(X)
12. ¬ Herbivore(X) v ¬ HasColorBlackWhite(X) v Zebra(X)
13. ¬ Bird(X) v ¬ CanWalk(X) v ¬ HasColorBlackWhite(X) v ¬ HasLongNeck(X) v Ostrich(X)
14. ¬ Bird(X) v ¬ CanSwim(X) v ¬ HasColorBlackWhite(X) v Penguin(X)
15. ¬ Bird(X) v ¬ CanFly(X) v ¬ HasColorBlackWhite(X) v Albatross(X)

Facts:
HasHair(X)
HasClaws(X)
HasPointedTeeth(X)
HasForwardEyes(X)
HasColorTawny(X)
HasDarkSpots(X)
¬ Cheetah(X)
¬ Carnivore(X)
¬ Mammal(X)

COURSE ASSESSMENT 1

LumiNUS

MY MODULES MODULE SEARCH CONTENT BANKS RESEARCH RECRUITMENT STUDENT FEEDBACK

ISY5001

Grad Cert in Intelligent Reasoning Systems (IRS-MR, IRS-RS, ...

[1920] 2019/2020 Semester 2

Owner

GENERAL

- Module Overview
- Module Settings
- Module Details
- Class & Groups
- Attendance
- Task Report

TOOLS

- Announcements
- Chat
- Conferencing
- Consultation
- Files
- Forum

Files > IRS-MR: Machine Reasoning

Search files

Folder Name	Opening Date	Expiry Date	Status
IRS-MR: Machine Reasoning Courseware			Open
IRS-MR: Machine Reasoning In-Class Workshop			Open
IRS-MR: Machine Reasoning In-Class Workshop Submission Submission			Open
IRS-MR: Machine Reasoning In-Class Assessment			Closed
IRS-MR: Machine Reasoning In-Class Assessment Submission			Open

Upload Files

- Rearrange
- Create Folder
- Folder Settings
- Bulk Create Folders
- Activity Report

01-S-AssessmentQuestion A1234567B_GuZhan(Sam).docx

Upload word, pdf or zip file to LumiNUS (one single file per participant)

3.2

INFERENCE UNDER UNCERTAINTY

3.2 INFERENCE UNDER UNCERTAINTY

"WHEN the lecture(r) is very boring THEN we feel sleepy."

- **Certainty Factor (CF)**
    - Certainty Factors are **measures of belief** or how much **confidence** we have in the knowledge/rule/process/data
- **Fuzzy Logic (FL)**
    - Fuzzy Logic are **measures of inclination (degree of belonging)** towards a **linguistic** concept/word, which lacks a **rigorous definition**.

3.2.1

CERTAINTY FACTOR (CF)

Certainty Factor (CF)

- It allows experts to fairly easily express their personal "probability" and, it also allows analyst to easily incorporate them in machine reasoning systems
- Certainty Factors are **measures of belief** or how much **confidence** we have in the data/information
- Certainty Factors can be incorporated into Rules and Facts.
- Typically CF range: $-1.0 \leq CF \leq +1.0$
  - CF = +1.0 The rule/fact is certainly true.
  - CF = 0.0 We don't know whether it is true or not.
  - CF = -1.0 The rule/fact is certainly false.

3.2.1 INFERENCE UNDER UNCERTAINTY

Certainty Factors in Rules

- CF in a rule represents the expert's confidence or belief in that chunk of knowledge.
- Rules with CF has the following structure:

```
IF good_earnings THEN share_up {cf 0.7}
IF win_contract THEN share_up {cf 0.9}
```

- If the condition is true then the conclusion is known to be true (proportional to the strength of the CF).
- CF can be elicited by "How confident are you that good earnings will cause the share price to go up?".

3.2.1 INFERENCE UNDER UNCERTAINTY

Certainty Factors in Facts

- CF in facts represents the expert's or user's belief in that piece of information:

```
good_earnings {cf -0.7}
win_contract {cf 0.8}
```

- Facts can consist of evidence, observations, intuition, therefore fact CF can be subjective.
- It can also be based on probability or obtained through statistical analysis and surveys.
- CF can be elicited by "What is the chance of the company winning the contract?".

3.2.1 INFERENCE UNDER UNCERTAINTY

Uncertain Terms Interpretation

Definitely NOT	-1.0
Almost Certainly NOT	-0.8
Probably NOT	-0.6
Maybe NOT	-0.4
UNKNOWN	-0.2 to +0.2
Maybe	+0.4
Probably	+0.6
Almost Certainly	+0.8
Definitely	+1.0

- **Certainty factors are propagated (calculated) through the reasoning chain when rules are fired.**
- **The following is a typical sequence of CF propagation:**
  1. User inputs a fact with a certainty value.
  2. All applicable rules are activated, ready to fire.
  3. When a rule is fired, the net rule certainty is calculated.
  4. When many rules are fired, their combined net certainty value is calculated.
  5. Final rule conclusion is then given with a merged single certainty value.

3.2.1 INFERENCE UNDER UNCERTAINTY

Finding the Net Certainty of a Rule

- When a rule is fired, the net certainty of the rule conclusion is calculated as follows:

$$cf(H,E) = cf(E) * cf(R)$$

cf(H,E) – Net Certainty of the rule conclusion
cf(E) – Certainty of the fact (rule input)
cf(R) – Certainty of the rule

For example:

IF earnings=good THEN shares=up {cf 0.7}

and the current certainty of earnings=good is 0.8, then

$$cf\ (H,E) = 0.8 \times 0.7 = 0.56$$

This result can be interpreted as “shares will probably go up”.

3.2.1 INFERENCE UNDER UNCERTAINTY

Conjunctive Evidences

- For rules with conjunctive evidences the certainty of the hypothesis *H* is calculated as follows:

$$cf\,(H, E_1 \cap E_2 \cap \ldots \cap E_n) = min\,[cf\,(E_1), cf\,(E_2), \ldots, cf\,(E_n)] \times cf$$

For example:

IF earnings=good AND contract=big THEN shares=up {cf 0.9}

current certainty of earnings=good is 0.8, and contract=big is 0.1 then

$$cf\,(H, E_1 \cap E_2) = min[0.8, 0.1] \times 0.9 = 0.1 \times 0.9 = 0.09$$

This result can be interpreted as "it is unknown if shares will go up"

3.2.1 INFERENCE UNDER UNCERTAINTY

Disjunctive Evidences

- For rules with disjunctive evidences the certainty of the hypothesis *H* is calculated as follows:

$$cf\,(H, E1 \cup E2 \cup \ldots \cup E_n) = max\,[cf\,(E_1), cf\,(E_2),\ldots, cf\,(E_n)] \text{ x } cf$$

For example:

IF earnings=good ***OR*** *contract=big THEN shares=up {cf* ***0.9****}*

current certainty of earnings=good is **0.8**, and contract=big is **0.1** then

$$cf\,(H, E_1 \cup E_2) = max[0.8, 0.1] \text{ x } 0.9 = 0.8 \text{ x } 0.9 = 0.72$$

This result can be interpreted as "**shares will most probably go up**"

3.2.1 INFERENCE UNDER UNCERTAINTY

Combining Multiple Conclusions

- When rules are fired, they insert/assert their respective *cf(H,E)* into working memory
- When the same Hypothesis *H* is asserted by two or more rules, e.g. $cf(H,E_1)$ ... $cf(H,E_n)$, all *cfs* are combined to a single *cf(H)*

For example:

IF earnings=good THEN shares=up {cf 0.7}
IF contract=big THEN shares=up {cf 0.9}

and earnings=good is 0.8, and contract=big is 0.1 then

$cf(H,E_1) = 0.56$ $cf(H,E_2) = 0.09$

- What will be the advice? "share will probably go up" or "it is unknown"

3.2.1 INFERENCE UNDER UNCERTAINTY

Combining Multiple Conclusions

- When several rules are fired that lead to the same conclusion, we combine them as follows:

$$cf(cf_1, cf_2) = \begin{cases} cf_1 + cf_2 \times (1 - cf_1) & \text{if } cf_1 > 0 \text{ and } cf_2 > 0 \\ \dfrac{cf_1 + cf_2}{1 - min\left[|cf_1|, |cf_2|\right]} & \text{if } cf_1 < 0 \text{ or } cf_2 < 0 \\ cf_1 + cf_2 \times (1 + cf_1) & \text{if } cf_1 < 0 \text{ and } cf_2 < 0 \end{cases}$$

$cf_1 = cf(H,E_1)$ is the net certainty of rule 1 conclusion

$cf_2 = cf(H,E_2)$ is the net certainty of rule 2 conclusion

3.2.1 INFERENCE UNDER UNCERTAINTY

Certainty Factor Exercise

R1:	**IF**	dividends=yes **AND** mgnt=good **AND** earnings=positive
	THEN	**buy=yes (0.6)**
R2:	**IF**	contract=large
	THEN	**buy=yes (1.0)**
R3:	**IF**	stock=penny
	THEN	**buy=yes (-0.7)**

Inputs: dividends=yes (cf 0.9)
mgnt=good (cf 0.7)
earnings=positive (cf 0.5)
contract=large (cf 0.8)
stock=penny (cf 1.0)

- Fire R1: buy=yes = min (0.9, 0.7, 0.5) * 0.6 = 0.3
- Fire R2: buy=yes = 0.8 * 1.0 = 0.8
- <u>Fire RX: buy=yes = 0.3 + 0.8 * (1.0 – 0.3) = 0.86</u>
- Fire R3: buy=yes = 1.0 * -0.7 = -0.7
- <u>Fire RX: buy=yes = (-0.7 + 0.86) / (1.0 – 0.7) = 0.53</u>
- Therefore, final recommendation: buy=yes (0.53)

3.2.1 INFERENCE UNDER UNCERTAINTY

Certainty Factor Summary

- Certainty factors theory provides a practical alternative to probability calculation.

- Certainty Factor approach mimics the thinking process of a human expert.

- Certainty Factor approach provides better intuitive explanations to users.

3.2.2

FUZZY LOGIC (FL)

3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic (FL)

- Fuzzy logic is an approach to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based.
- Fuzzy logic is close to the way our brains work. We aggregate data and form a number of partial truths which we aggregate further into higher truths (higher confidence) which in turn, when certain “latent thresholds” are exceeded, cause certain further results such as motor reaction. A similar kind of process is used in neural networks, expert systems and other artificial intelligence applications.

Source: https://www.youtube.com/watch?v=J_Q5X0nTmrA

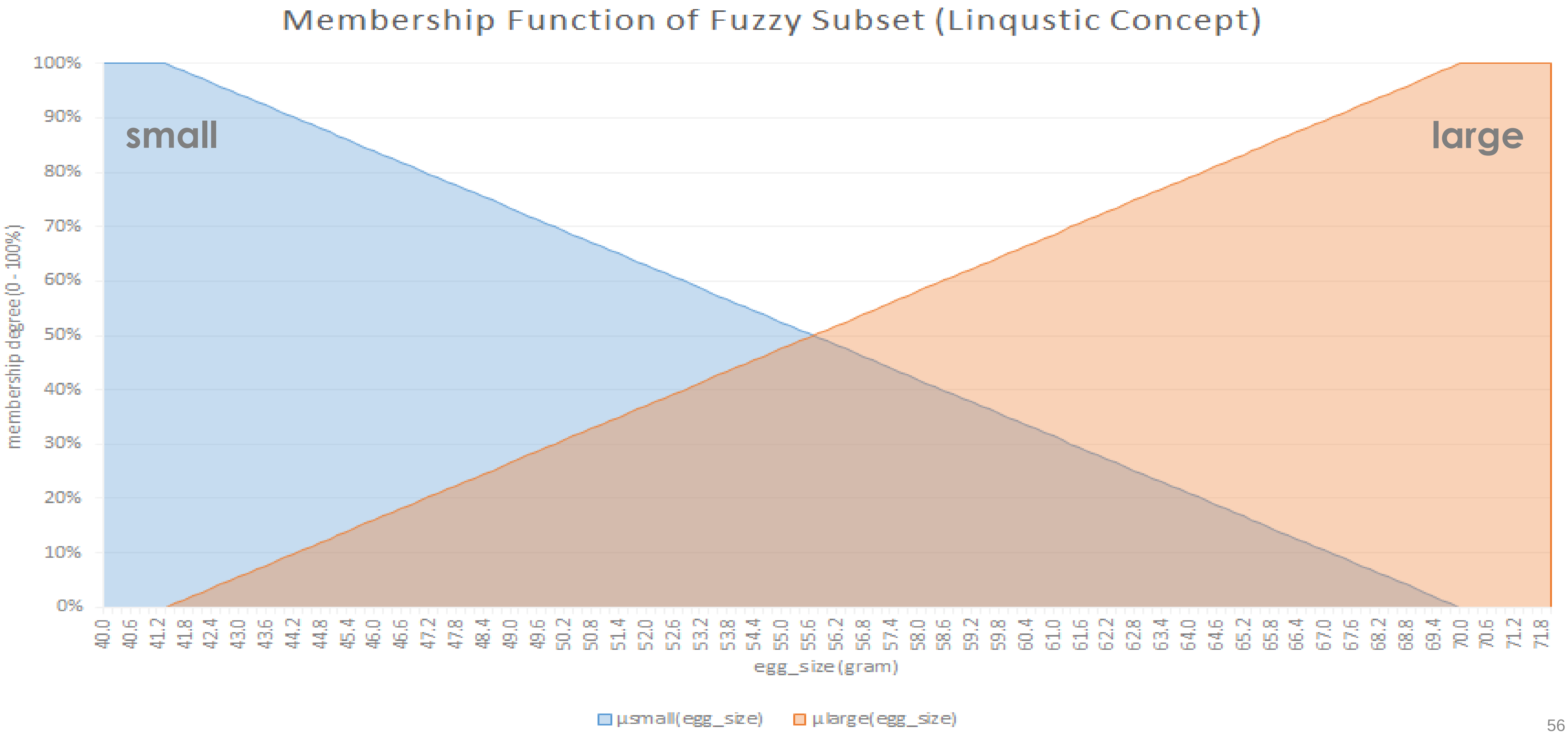
3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic in Rules

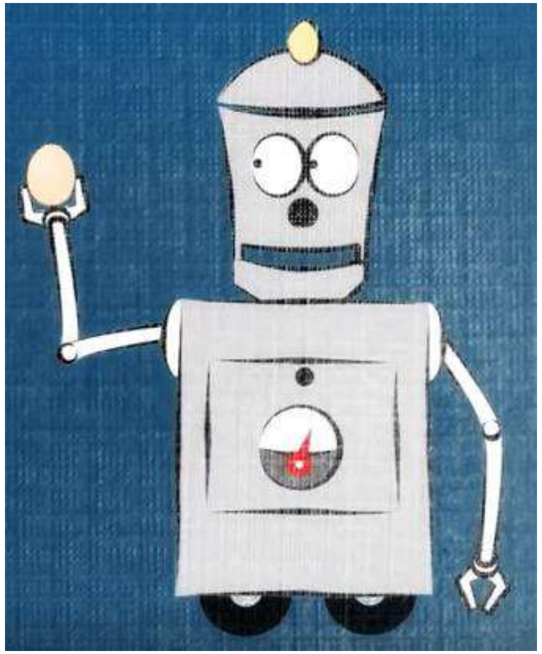
- **Egg-boiling Fuzzy Ruleset**
  - **IF** egg size is **small** **THEN** boil **less than 5 minutes**
  - **IF** egg size is **large** **THEN** boil **more than 5 minutes**
- **3 Steps of Fuzzy Reasoning**
  - Fuzzification
  - Inference
  - Defuzzificaiton

3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic in Rules : Fuzzification



Fuzzy Logic in Rules : Inference



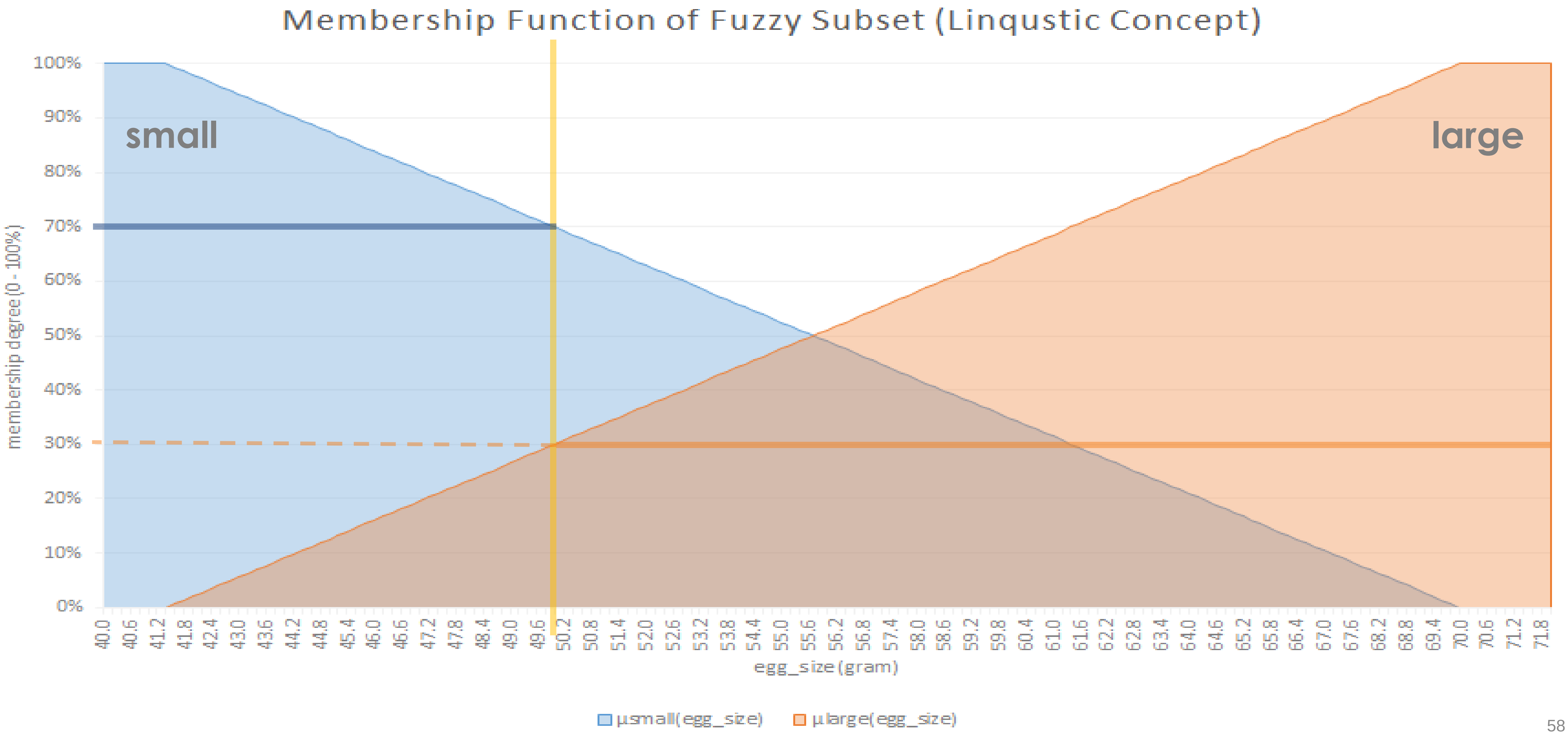
Link https://www.youtube.com/watch?v=J_Q5X0nTmrA

- **New fact: The egg is 50 grams**

- **Egg-boiling Fuzzy Ruleset**
  - **IF** egg size is **small (%) THEN** boil **less than 5 minutes (%)**
  - **IF** egg size is **large (%) THEN** boil **more than 5 minutes (%)**

3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic in Rules : Inference



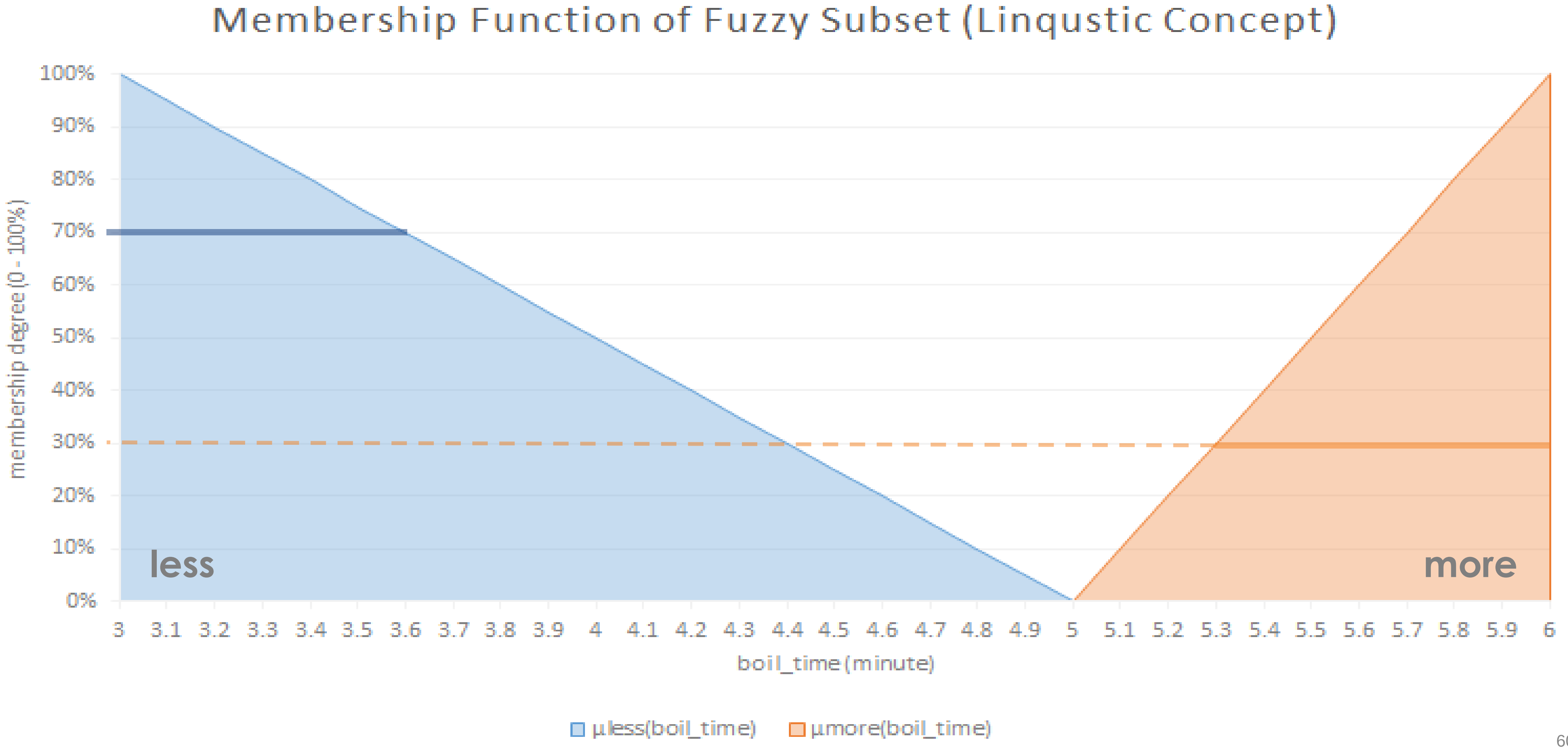
Fuzzy Logic in Rules : Inference

- **Egg-boiling Fuzzy Ruleset**

  - **IF** egg size is **small (70%) THEN** boil **less than 5 minutes (70%)**
  - **IF** egg size is **large (30%) THEN** boil **more than 5 minutes (30%)**

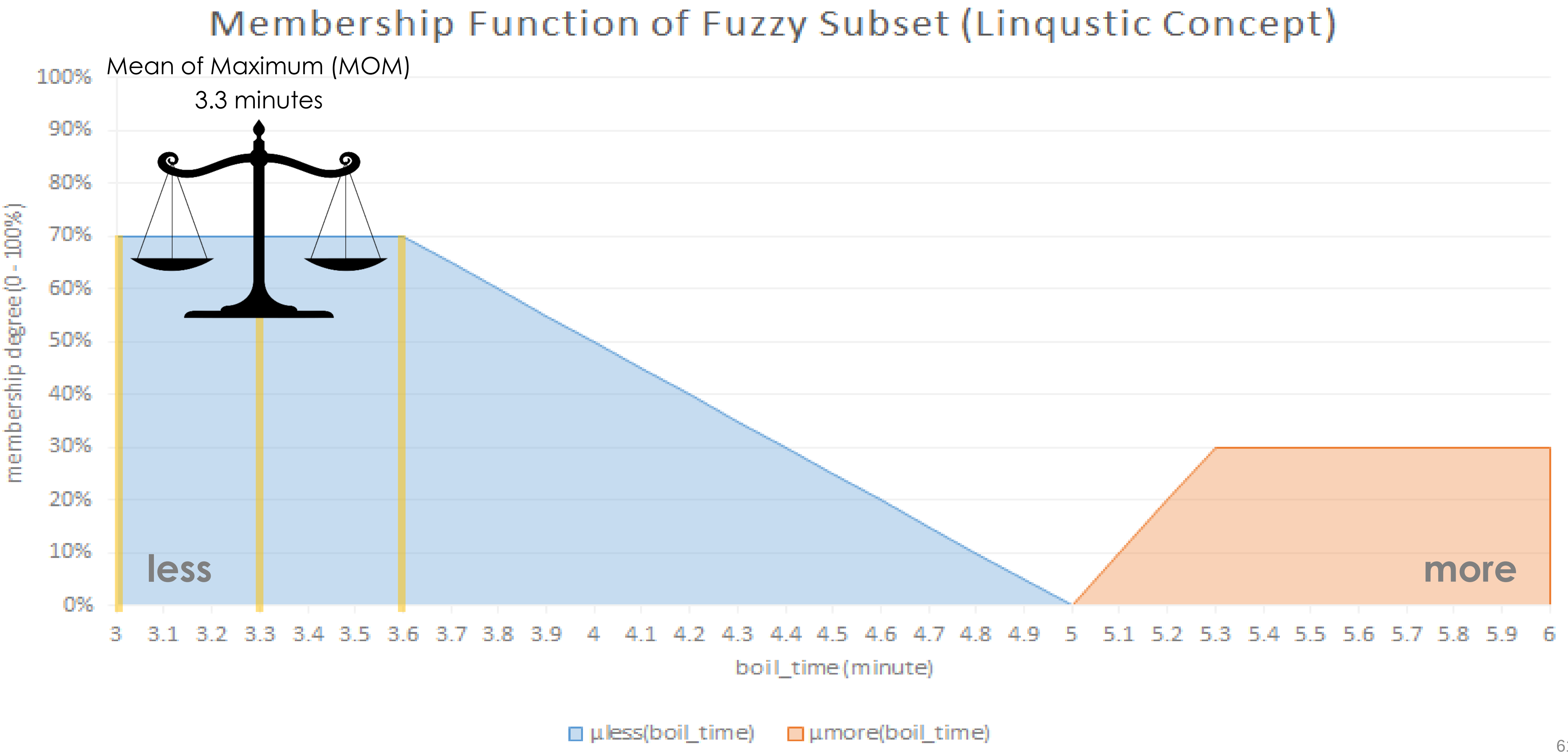
3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic in Rules : Inference



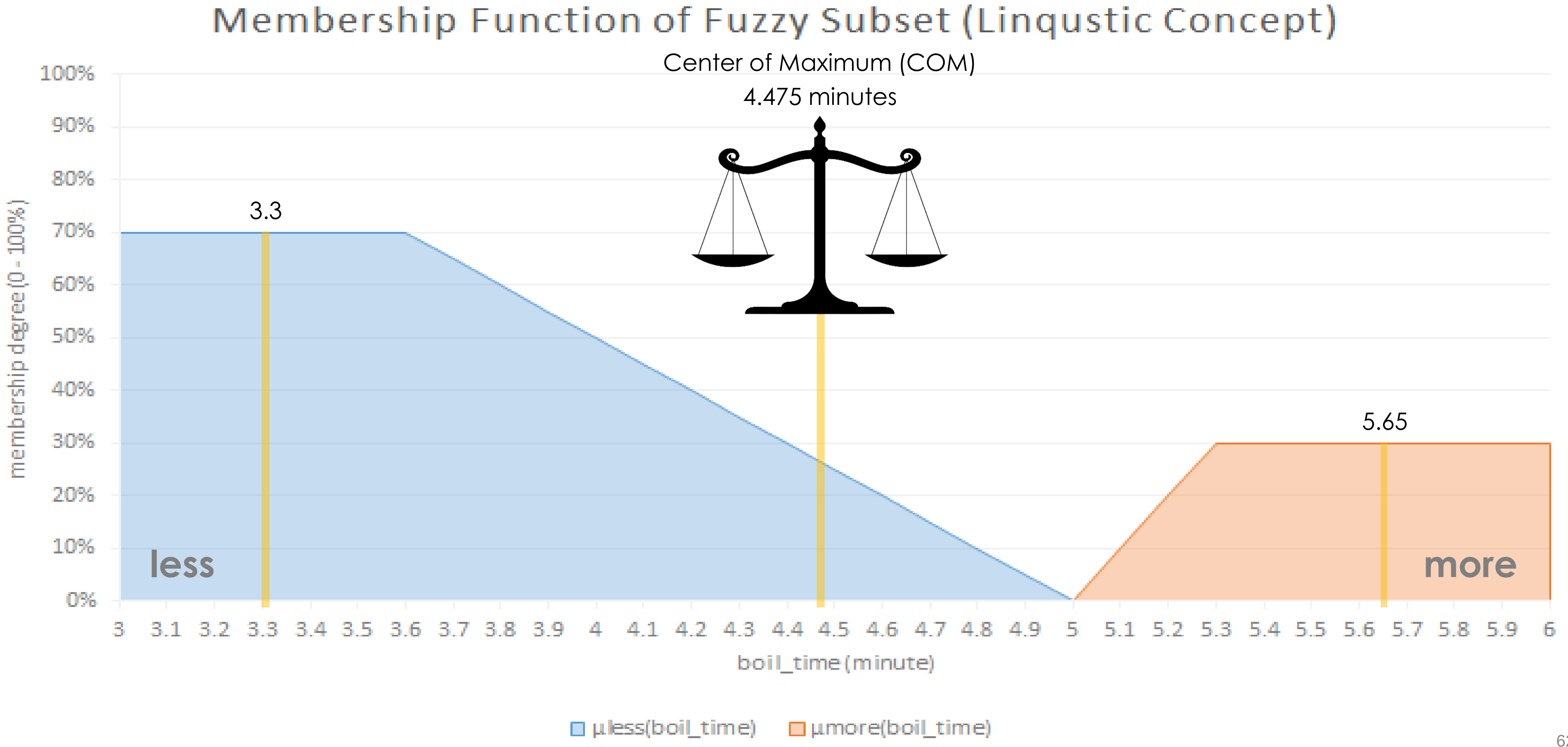
3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic in Rules : Defuzzification



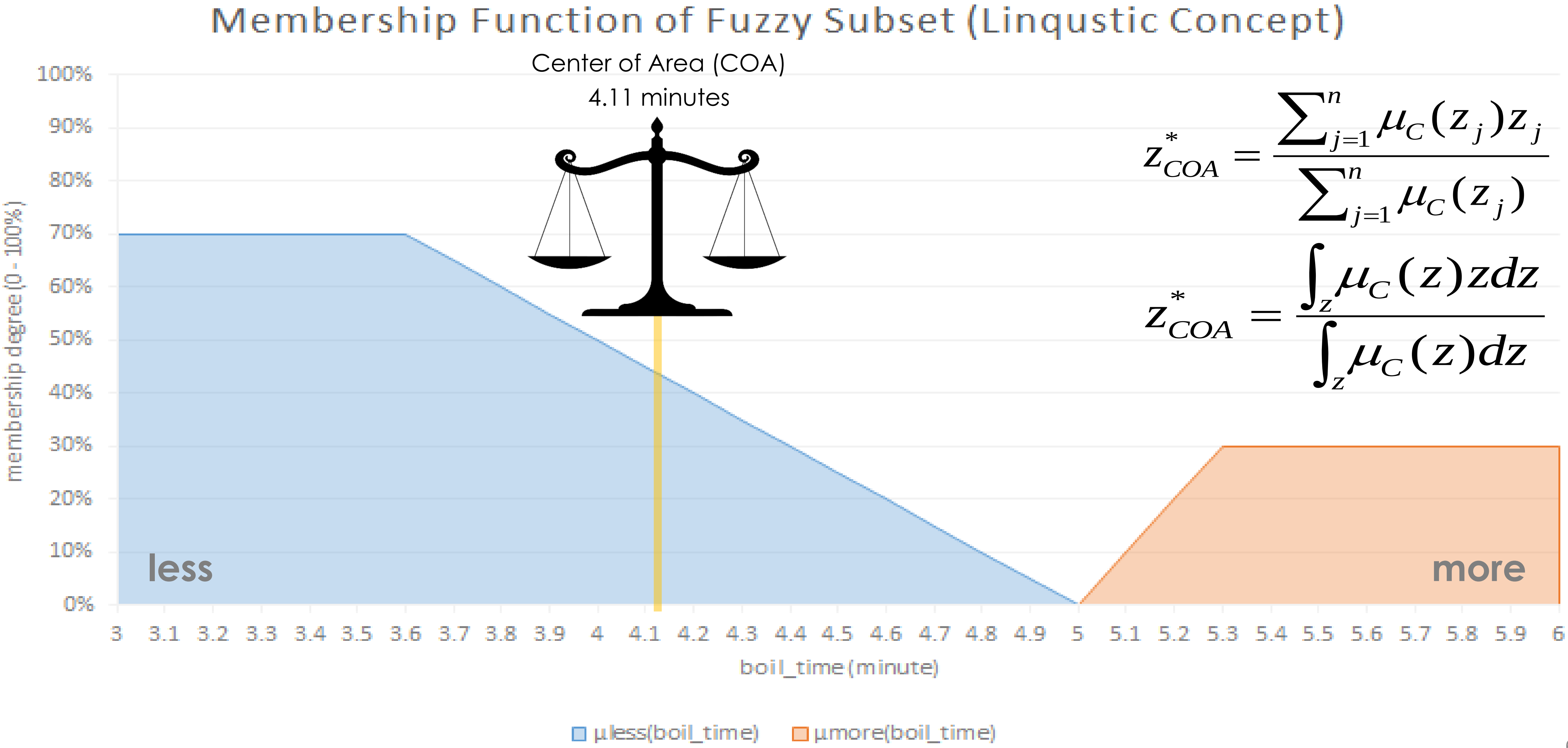
3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic in Rules : Defuzzification



3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic in Rules : Defuzzification



$$z^*_{COA} = \frac{\sum_{j=1}^{n} \mu_C(z_j) z_j}{\sum_{j=1}^{n} \mu_C(z_j)}$$

$$z^*_{COA} = \frac{\int_z \mu_C(z) z dz}{\int_z \mu_C(z) dz}$$

3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic in Rules : Defuzzification Exercise

boil_time (minute)	μ_{less}(boil_time)	μ_{more}(boil_time)	μ(boil_time) * boil_time (minute)	$\sum_{j=1}^{n} \mu_C(z_j) z_j$	$\sum_{j=1}^{n} \mu_C(z_j)$	Z_{COA}
3	0.7	0				
3.1	0.7	0				
3.2	0.7	0				
3.3	0.7	0				
3.4	0.7	0				
3.5	0.7	0				
3.6	0.7	0				
3.7	0.65	0				
3.8	0.6	0				
3.9	0.55	0				
4	0.5	0				
4.1	0.45	0				
4.2	0.4	0				
4.3	0.35	0				
4.4	0.3	0				
4.5	0.25	0				
4.6	0.2	0				
4.7	0.15	0				
4.8	0.1	0				
4.9	0.05	0				
5	0	0				
5.1	0	0.1				
5.2	0	0.2				
5.3	0	0.3				
5.4	0	0.3				
5.5	0	0.3				
5.6	0	0.3				
5.7	0	0.3				
5.8	0	0.3				
5.9	0	0.3				
6	0	0.3				

3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic in Rules : Extension

- More than one conditions
  - **IF** egg size is **small (70%)** **OR** **very** hungry **(90%)** **THEN** boil **less than 5 minutes (%)**
  - **IF** egg size is **large (45%)** **AND** **slightly** hungry **(75%)** **THEN** boil **more than 5 minutes (%)**
- Composition operation
  - **AND   Min()**
  - **OR   Max()**

3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic Applications

Automatic Washing Machine

- **Using fuzzy rules in the form of:**

  IF **few** clothes and they are soft THEN gentle flow and **short** washing time

  (where **few**, **soft** , … are based on measure (fuzzy values) from sensor, **gentle**, **short**, … are fuzzy concepts for control)

Fuzzy Cleaner

- **Fuzzy control of absorbing power based on the material & the dirty degree of the floor**

  If the sucking power is too strong, the nozzle will stuck on floor (difficult to operate); if too weak, the corner dust cannot be absorbed well.

3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic vs. Probability

Fuzziness and randomness deal with different types of uncertainty in our life

- **Is it a raining day now?**
  - To describe some existing situation
  - It is more subjective (different people may have different ideas)
  - Uncertainty of classification
- **Is it going to rain tomorrow?**
  - The event may or may not happen
  - It is objective (determined by natural law)
  - Uncertainty of occurrence

https://us.123rf.com/450wm/spawn83/spawn831809/spawn83180900051/108908180-rain-outside-the-window-raindrops-on-the-windowpane-on-a-cloudy-day.jpg?ver=6

3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic Summary

The theory of fuzziness is to build models for entities which lack a rigorous definition.

- The concept of "graded membership" belongs to a class which could be subjective in different business context.
- It is not compatible with a concept suitable for the lack of information, which is with probability.

3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic Summary

Long Hair Group ← Hair length ≥ 10 cm

Hair length < 10 cm → Short Hair Group

Long Hair Group ← Hair length is long

Hair length is short → Short Hair Group

What if the hair length is both long and short → Which Group?

3.3

KNOWLEDGE DISCOVERY BY MACHINE LEARNING

3.3 KNOWLEDGE DISCOVERY BY MACHINE LEARNING

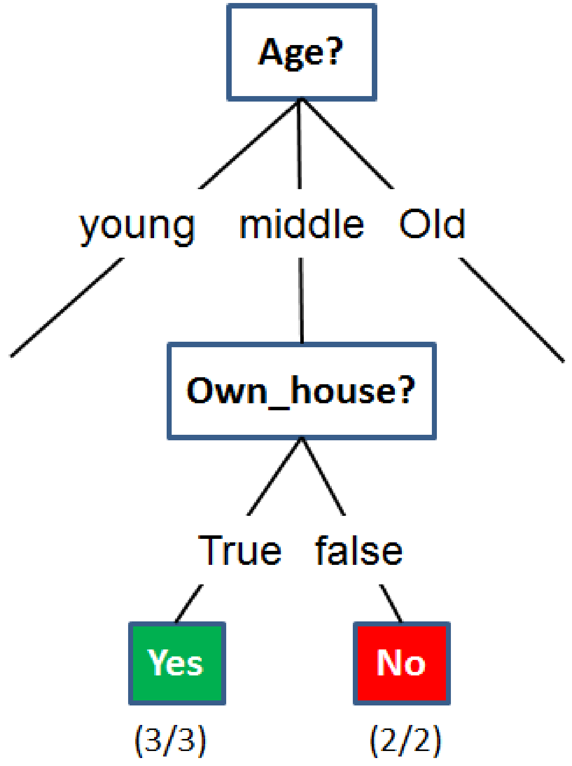
Bank Loan Example – Business Background

- Banks receive many loan applications that has to be assessed for approval.
- Each application consists of many factors such as **Age**, **Job status**, **Housing**, **Credit history**.
- Some applications are approved, others are not; Some debtors default, others don't.
- Banks dislike defaulters. Banks want to approve only applicants who are unlikely to default.
- Bank's task is to predict if a new applicant will default or not.
- This a classification problem: **Approve** projected non defaulter or **Reject** projected defaulter during loan application.

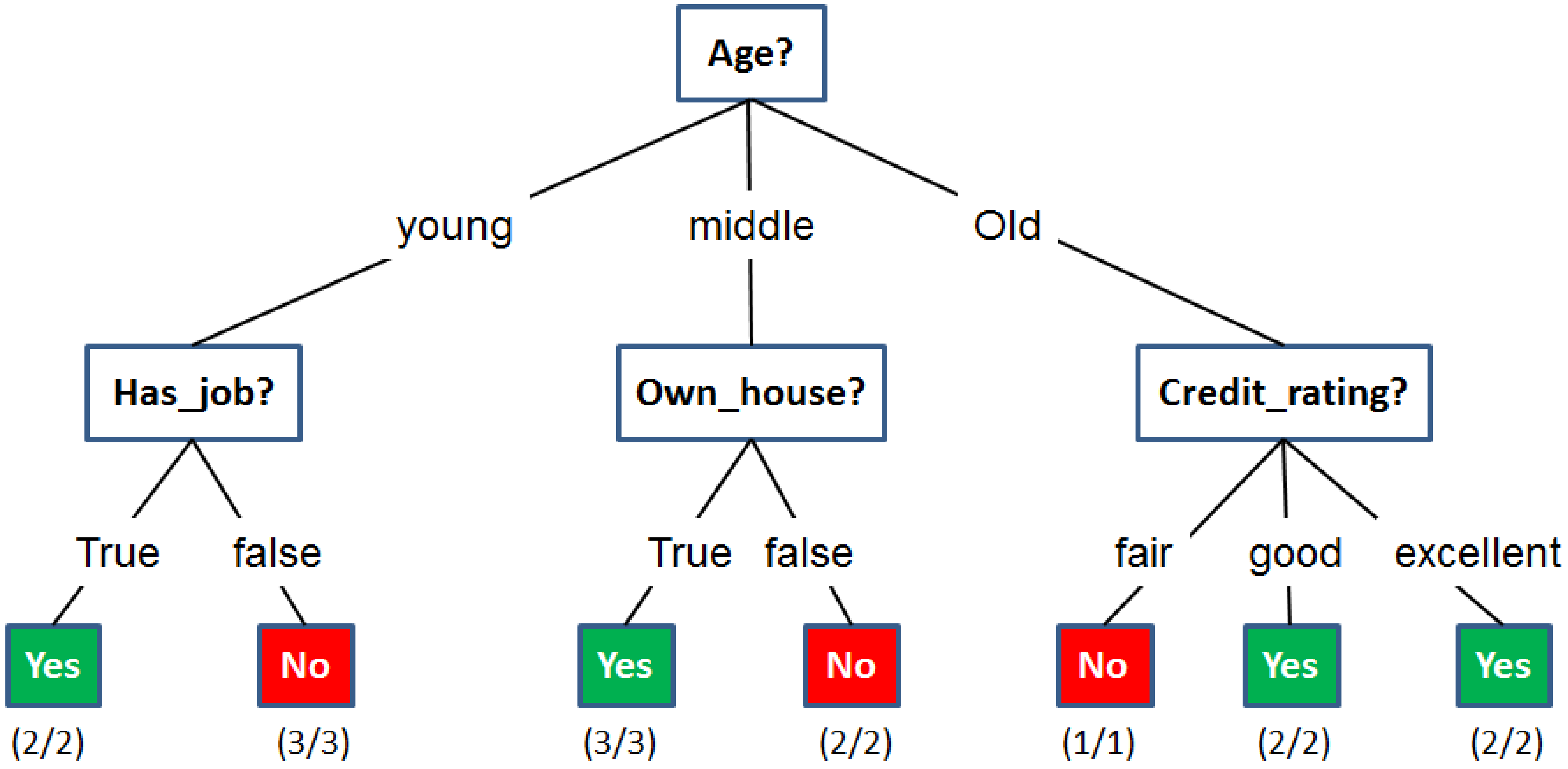
3.3 KNOWLEDGE DISCOVERY BY MACHINE LEARNING

Bank Loan Example – Rule Induction Data Science

ID	Age	Has_job	Own_house	Credit_rating	Outcome
1	young	False	False	fair	No
2	young	False	False	good	No
3	young	True	False	good	Yes
4	young	True	True	fair	Yes
5	young	False	False	fair	No
6	middle	False	False	fair	No
7	middle	False	False	good	No
8	middle	True	True	good	Yes
9	middle	False	True	excellent	Yes
10	middle	False	True	excellent	Yes
11	old	False	True	excellent	Yes
12	old	False	True	good	Yes
13	old	True	False	good	Yes
14	old	True	False	excellent	Yes
15	old	False	False	fair	No

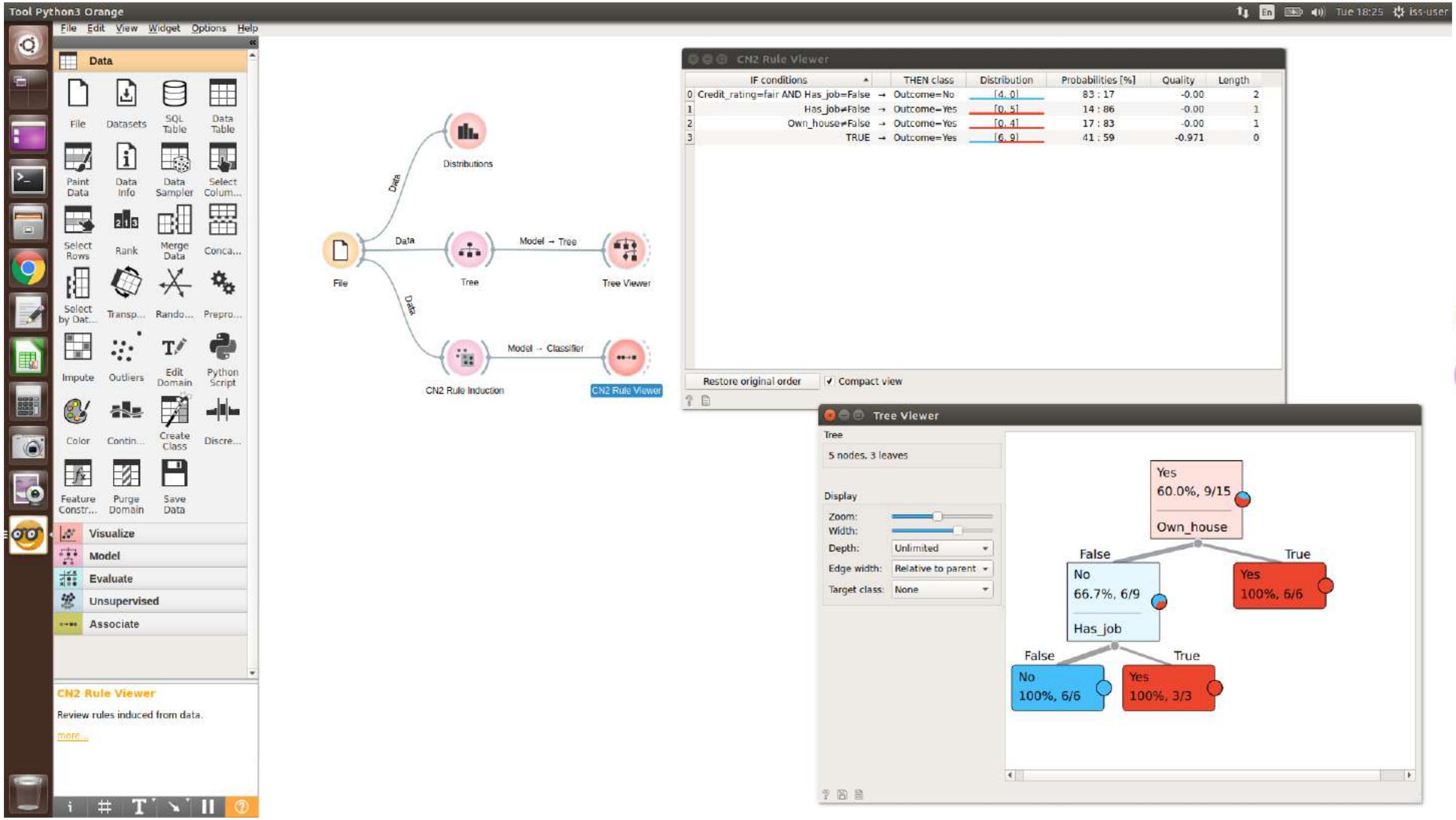


Bank Loan Example – Decision Tree



3.3 KNOWLEDGE DISCOVERY BY MACHINE LEARNING

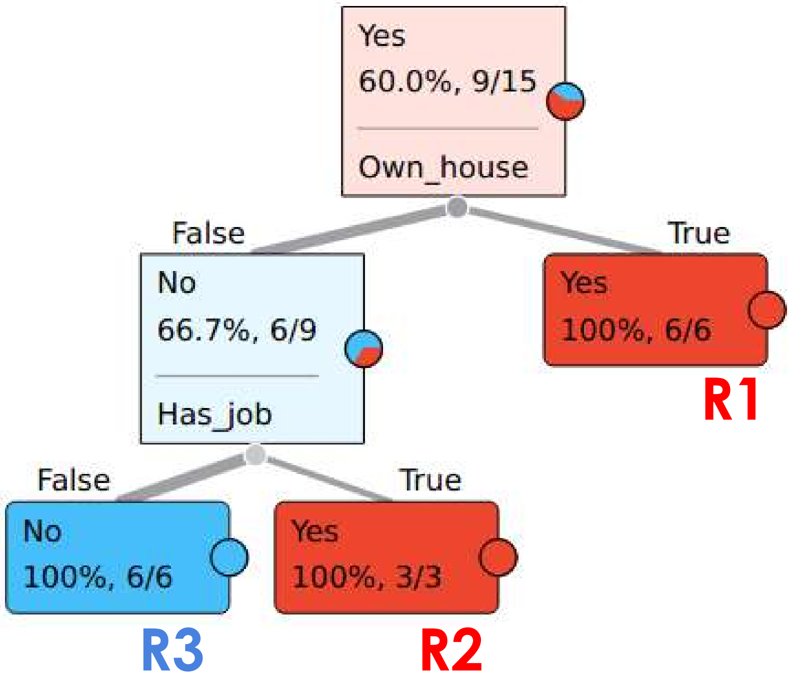
Data Mining Tool: Orange3 (python)



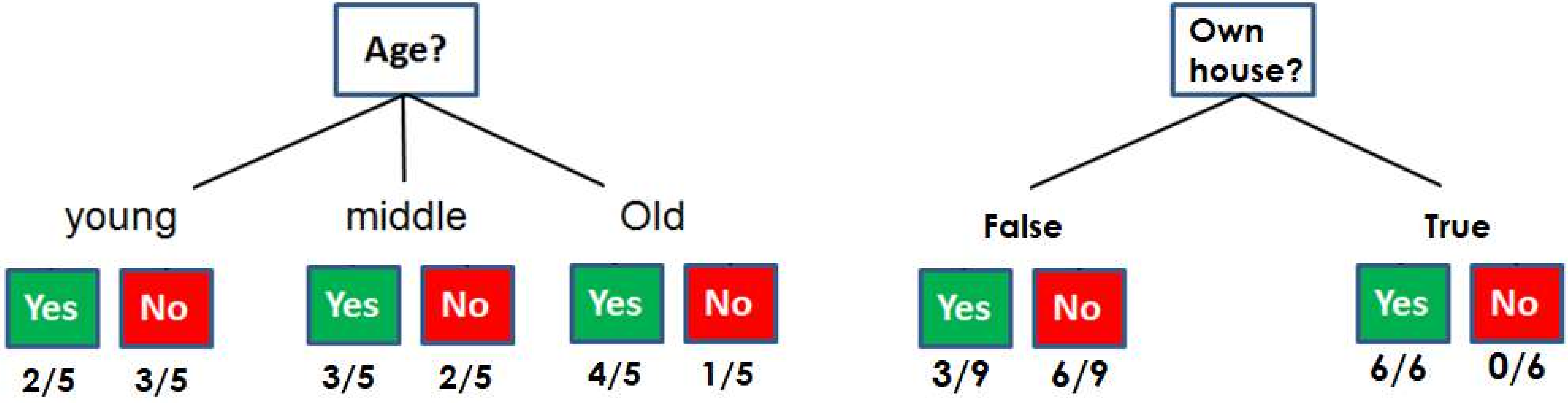
3.3 KNOWLEDGE DISCOVERY BY MACHINE LEARNING

Orange3 Bank Loan Example – Decision Tree

ID	Age	Has_job	Own_house	Credit_rating	Outcome	
1	young	False	False	fair	No	R3
2	young	False	False	good	No	
3	young	True	False	good	Yes	R2
4	young	True	True	fair	Yes	R1
5	young	False	False	fair	No	
6	middle	False	False	fair	No	R3
7	middle	False	False	good	No	
8	middle	True	True	good	Yes	
9	middle	False	True	excellent	Yes	
10	middle	False	True	excellent	Yes	R1
11	old	False	True	excellent	Yes	
12	old	False	True	good	Yes	
13	old	True	False	good	Yes	R2
14	old	True	False	excellent	Yes	
15	old	False	False	fair	No	R3



Decision Tree Algorithm – Which feature to select for split?



- **Which attribute is more intuitively for your to better/easier decision making?**

ID3 algorithm

- In decision tree learning, ID3 (Iterative Dichotomiser 3) is an algorithm invented by Ross Quinlan used to generate a decision tree from a dataset.

- Information Gain formula:

IG(DataSubsets | Attribute) = Imp(Initial Dataset) – Imp(DataSubsets | Attribute)

3.3 KNOWLEDGE DISCOVERY BY MACHINE LEARNING

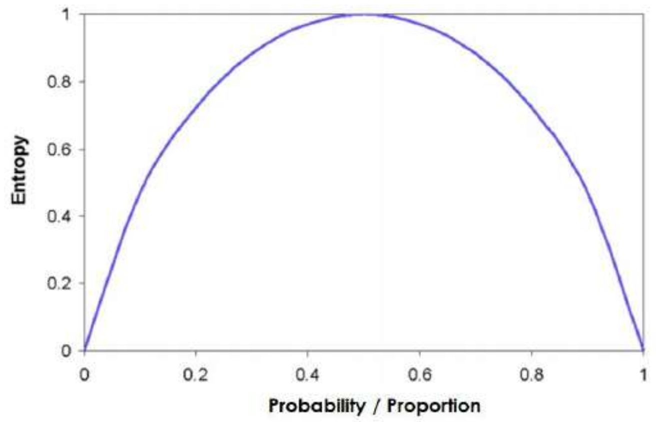
Decision Tree Algorithm – Initial Dataset Impurity

ID	Age	Has_job	Own_house	Credit_rating	Outcome
1	young	False	False	fair	No
2	young	False	False	good	No
3	young	True	False	good	Yes
4	young	True	True	fair	Yes
5	young	False	False	fair	No
6	middle	False	False	fair	No
7	middle	False	False	good	No
8	middle	True	True	good	Yes
9	middle	False	True	excellent	Yes
10	middle	False	True	excellent	Yes
11	old	False	True	excellent	Yes
12	old	False	True	good	Yes
13	old	True	False	good	Yes
14	old	True	False	excellent	Yes
15	old	False	False	fair	No

Yes	No
9	6

$P_1(Yes) = 9/15$

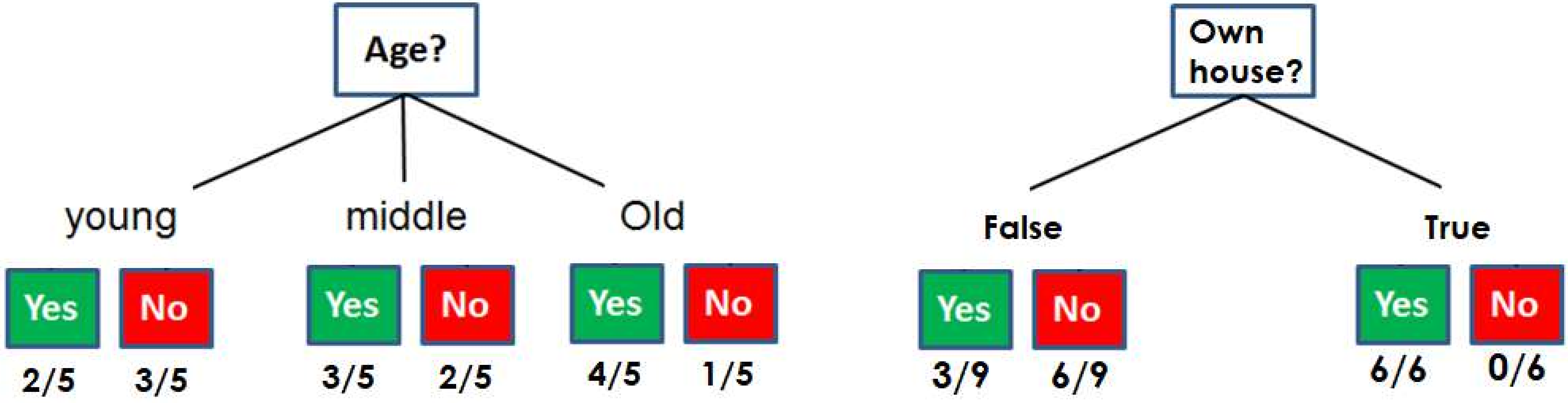
$P_2(No) = 6/15$



$$Imp(D_j) = Entropy(p) = \mathrm{I}_E(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

Imp(Initial Dataset) = - 9/15 log(9/15) - 6/15 log(6/15) = 0.5288 + 0.442 = 0.971

Decision Tree Algorithm – Data Subset Impurity by attribute split

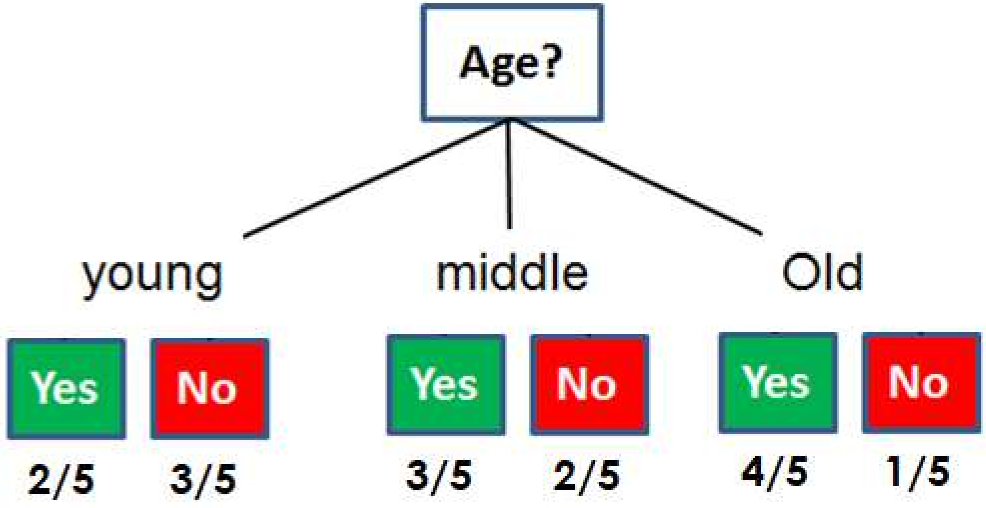


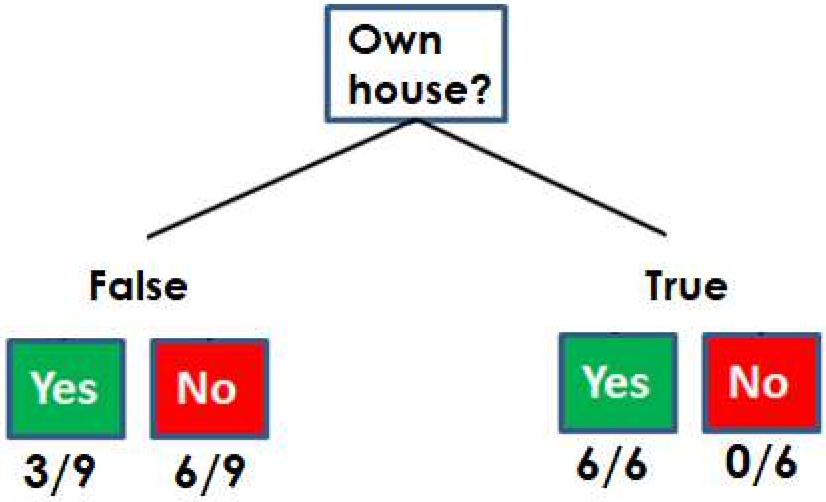
$$Imp(D_j) = Entropy(p) = \mathrm{I}_E(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

$$Imp(\{D_1, \ldots, D_l\}) = \sum_{j=1}^{l} \frac{|D_j|}{|D|} Imp(D_j)$$

3.3 KNOWLEDGE DISCOVERY BY MACHINE LEARNING

Decision Tree Algorithm – Which attribute for conditional split?





Imp(Young) =

-2/5log(2/5)-3/5log(3/5)

= 0.5288 + 0.442

= 0.971

Imp(Middle)

= 0.971

Imp(Old) =

-4/5log(4/5)-1/5log(1/5)

= 0.2575 + 0.4644

= 0.722

Imp(False) =

-3/9log(3/9)-6/9log(6/9)

= 0.5283 + 0.39

= 0.918

Imp(True) =

-6/6log(6/6)-0/6log(0/6)

= 0 + 0

= 0

Total Imp(Age) = 5/15 x 0.971 + 5/15 x 0.971 + 5/15 x 0.722

= 0.3237 + 0.3237 + 0.2407 = 0.888

IG(D|Age) = Imp(Initial Dataset) - Imp(D|Age)

= 0.971 – 0.888 = 0.083

Total Imp(Own_house) = 9/15 x 0.918 + 6/15 x 0

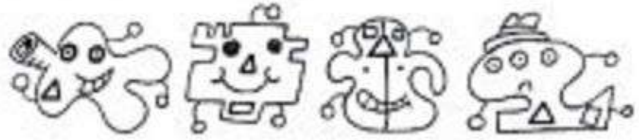
= 0.551 + 0 = 0.551

IG(D|Own_house) = Imp(Initial Dataset) - Imp(D|Own_house)

= 0.971 – 0.551 = 0.42 ☺ The larger IG, the better attribute split !

3.3 KNOWLEDGE DISCOVERY IDENTIFY ALIENS

Aliens



Not aliens

Training Data

SN	Triangle	Antenna	Teeth	Eyes	Alien
1	1	3	1	2	TRUE
2	1	3	0	2	TRUE
3	1	3	1	2	TRUE
4	1	3	0	3	TRUE
5	1	2	1	2	FALSE
6	0	3	0	3	FALSE
7	1	6	0	2	FALSE
8	0	3	0	2	FALSE

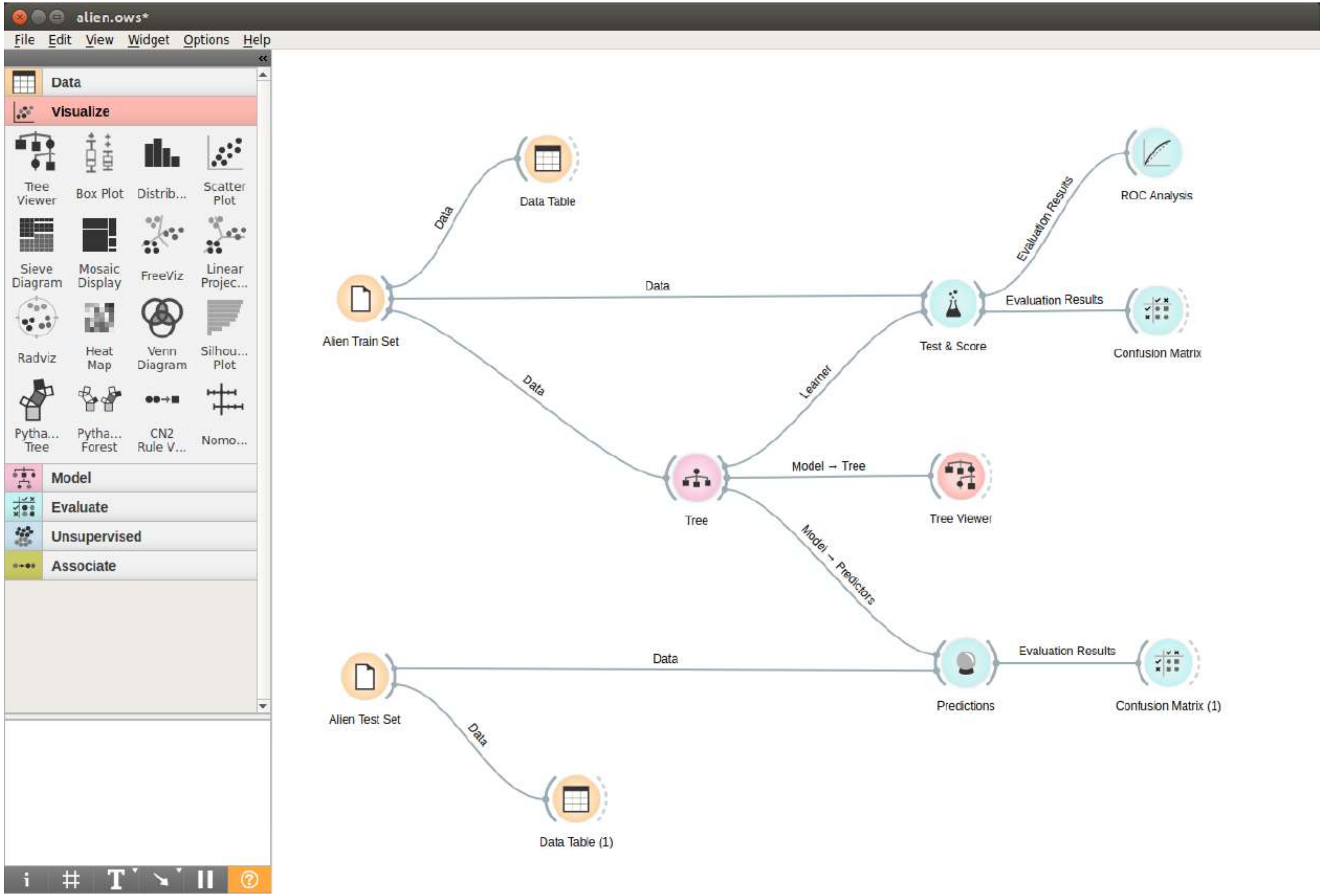
Which one is alien?

A B C D E

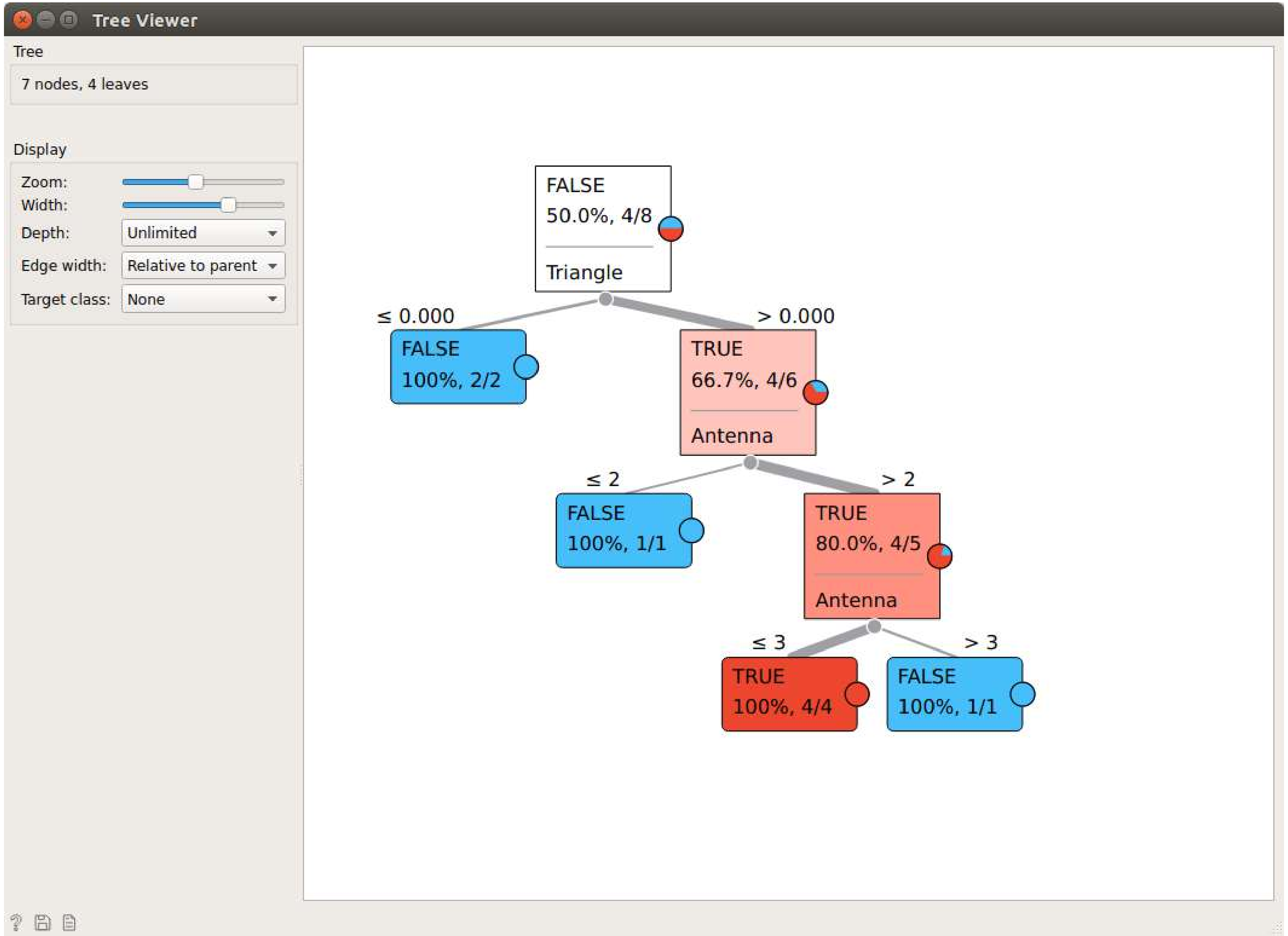
Test Data

SN	Triangle	Antenna	Teeth	Eyes	Alien
A	1	2	0	2	FALSE
B	3	2	1	2	FALSE
C	1	4	0	2	FALSE
D	1	3	0	2	TRUE
E	0	3	0	2	FALSE

3.3 KNOWLEDGE DISCOVERY IDENTIFY ALIENS

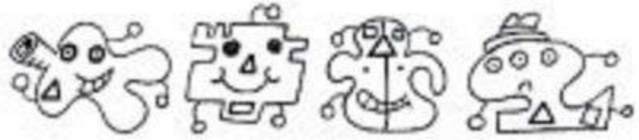


3.3 KNOWLEDGE DISCOVERY IDENTIFY ALIENS



3.3 KNOWLEDGE DISCOVERY IDENTIFY ALIENS

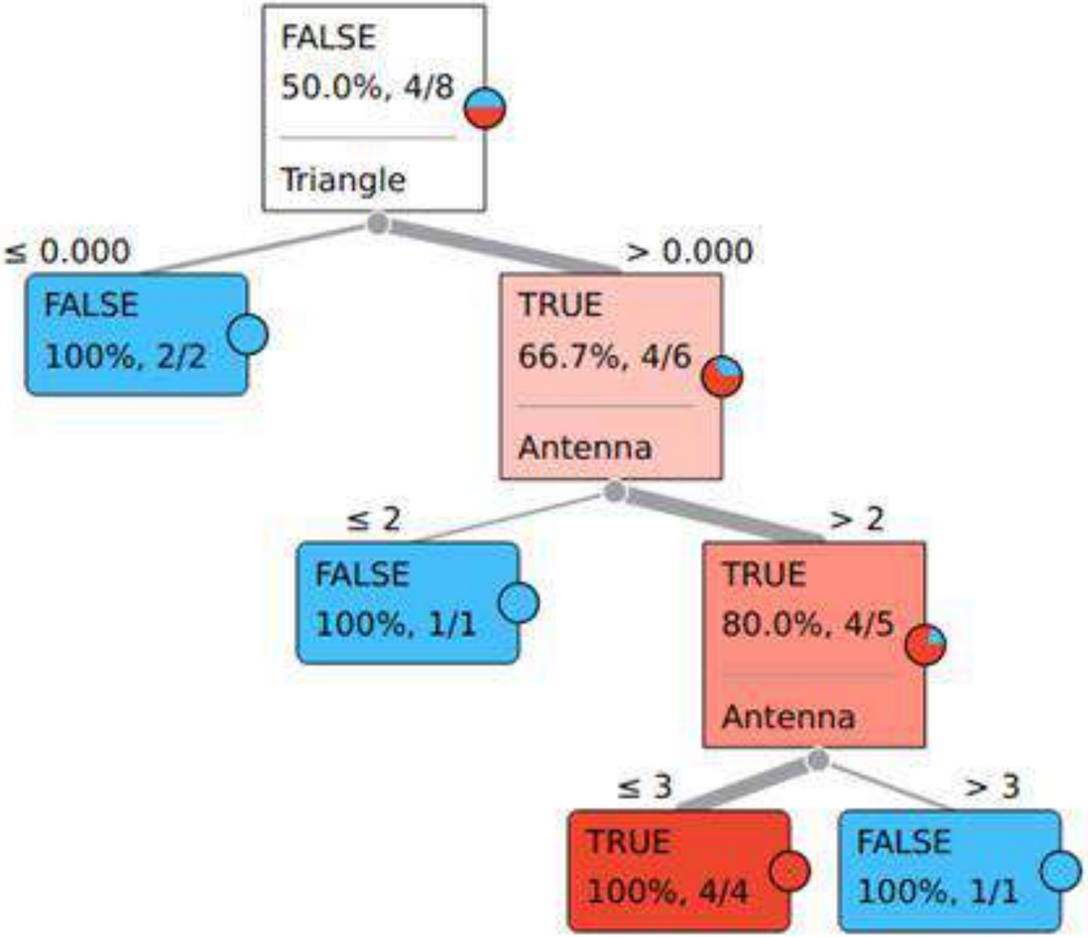
Aliens



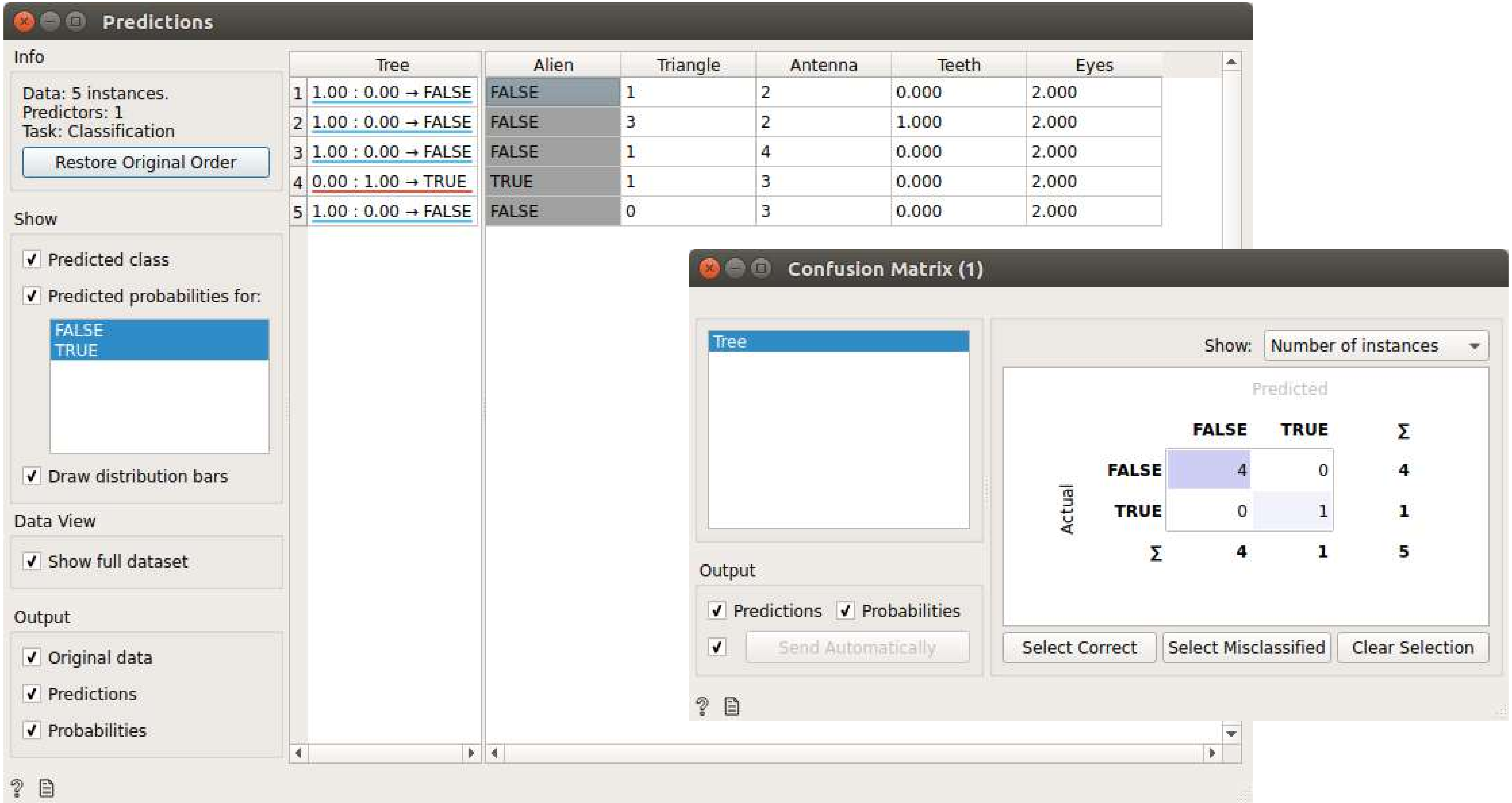
Not aliens

Which one is alien?

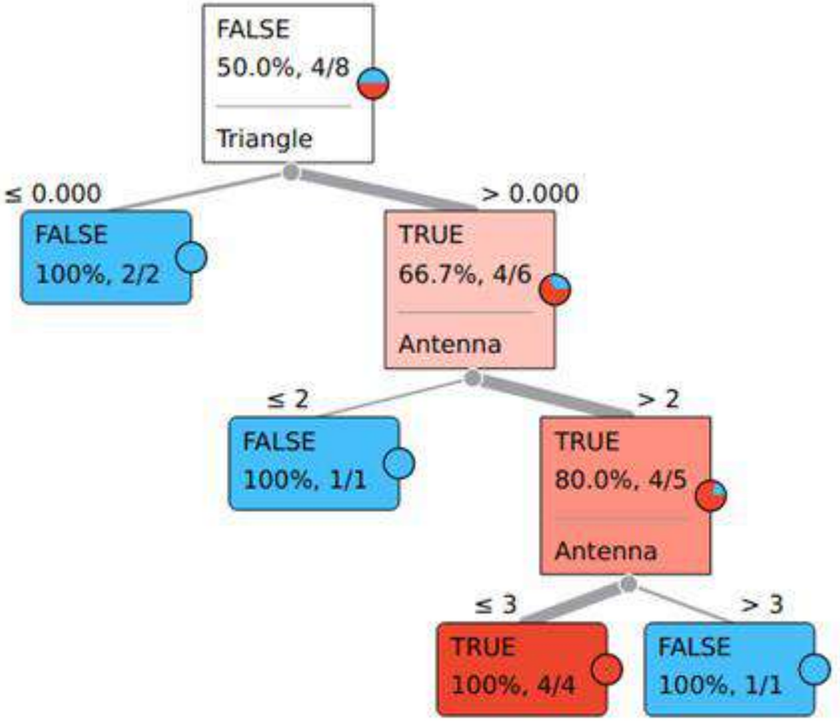
A B C D E



3.3 KNOWLEDGE DISCOVERY IDENTIFY ALIENS



3.3 KNOWLEDGE DISCOVERY IDENTIFY ALIENS



Test Data

SN	Triangle	Antenna	Teeth	Eyes	Alien
A	1	2	0	2	FALSE
B	3	2	1	2	FALSE
C	1	4	0	2	FALSE
D	1	3	0	2	TRUE
E	0	3	0	2	FALSE
F	2	3	0	2	FALSE

Actual \ Predicted	FALSE	TRUE	Σ
FALSE	4	1	5
TRUE	0	1	1
Σ	4	2	6

	Tree	Alien	Triangle	Antenna	Teeth	Eyes
1	1.00 : 0.00 → FALSE	FALSE	1	2	0.000	2.000
2	1.00 : 0.00 → FALSE	FALSE	3	2	1.000	2.000
3	1.00 : 0.00 → FALSE	FALSE	1	4	0.000	2.000
4	0.00 : 1.00 → TRUE	TRUE	1	3	0.000	2.000
5	1.00 : 0.00 → FALSE	FALSE	0	3	0.000	2.000
6	0.00 : 1.00 → TRUE	FALSE	2	3	0.000	2.000

Which one is alien?

A B C D E

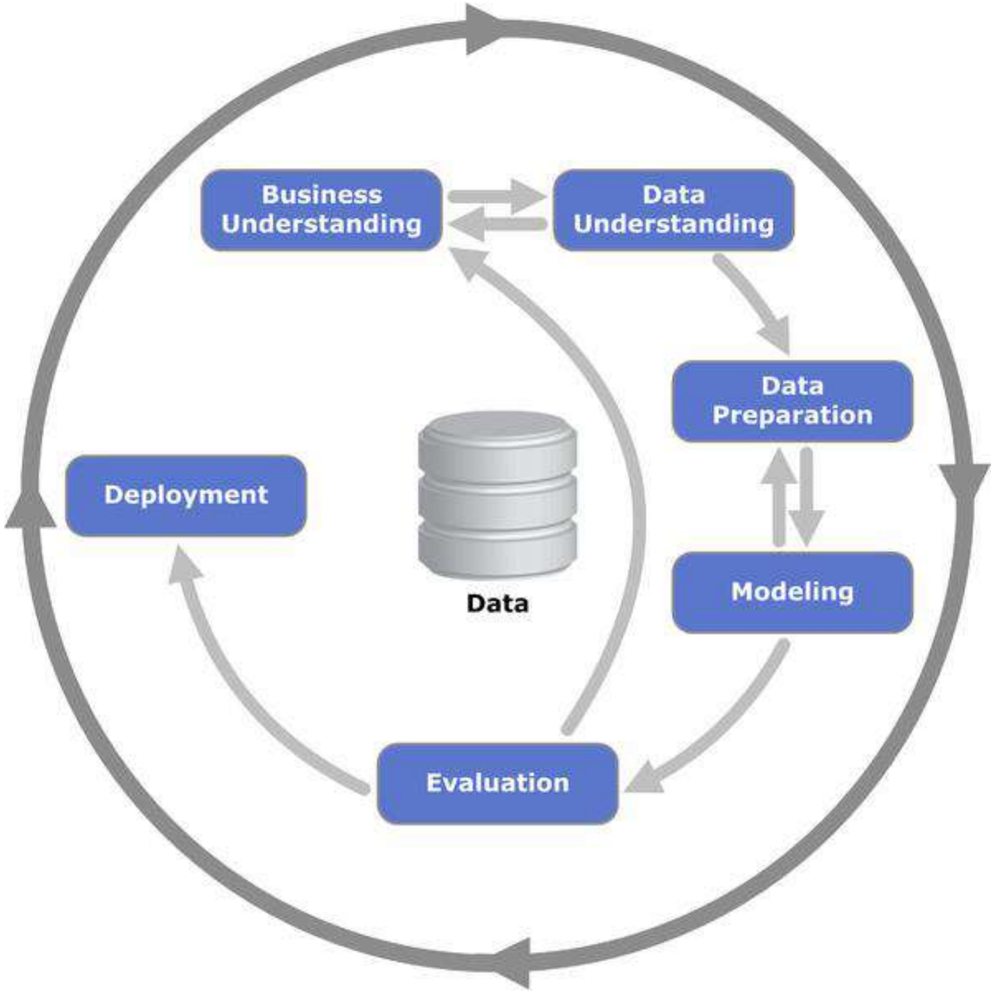
F

The 'black swan': unseen before in historical data (no scenario/representation in training data)

3.3 KNOWLEDGE DISCOVERY

Data Mining Framework: CRISP-DM

- Cross-Industry Standard Process for Data Mining
- Began life as a Data Mining methodology
- Non-proprietary and Application/Industry/Tool neutral
- Focus on business issues, as well as technical analysis
- Framework for guidance and aim is for a Process Model designed for use by anyone
- Experience base: Templates for Analysis
- Provides a complete blueprint that describes all steps in the process: Life cycle has six phases

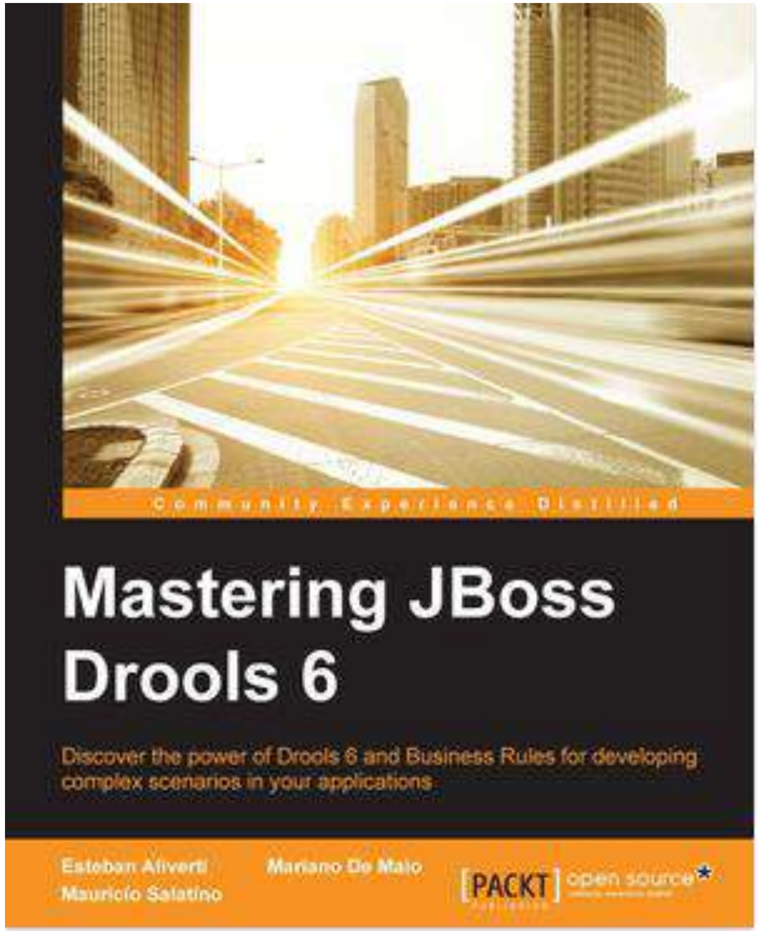


3.4 WORKSHOP
KNOWLEDGE DISCOVERY

3.4 WORKSHOP KNOWLEDGE DISCOVERY

- **Knowledge Discovery [Orange3]**
  - Extract business rule from data using inductive reasoning: bank loan example
- **KIE BPMS/BRMS Business System Enhancement**
  - Business system enhancement [Decision Table]
  - Business system enhancement [Deploy] v6.0.0

DAY 3 REFERENCE



1. Orange3 Tutorials

   https://orange.biolab.si/getting-started/

2. Designing a decision service using guided decision tables

   https://access.redhat.com/documentation/en-us/red_hat_decision_manager/7.2/html-single/designing_a_decision_service_using_guided_decision_tables/

3. Designing a decision service using uploaded decision tables (Excel)

   https://access.redhat.com/documentation/en-us/red_hat_decision_manager/7.2/html-single/designing_a_decision_service_using_uploaded_decision_tables/

4. Drools Using Rules from Excel Files by Sunil Mogadati

   https://www.baeldung.com/drools-excel

DAY 3 SUMMARY

3.1 Machine Inference (part 2)

- Conflict Set; Conflict Resolution

3.2 Inference under Uncertainty

- Certainty Factor: Multiple Conclusions CF Composition;
- Fuzzy Logic: Fuzzification, Inference, and Defuzzification

3.3 Knowledge Discovery by Machine Learning

- Data Mining; Decision Tree; Orange3
- CRISP-DM data mining framework

3.4 Knowledge Discovery Workshop

END OF LECTURE NOTES