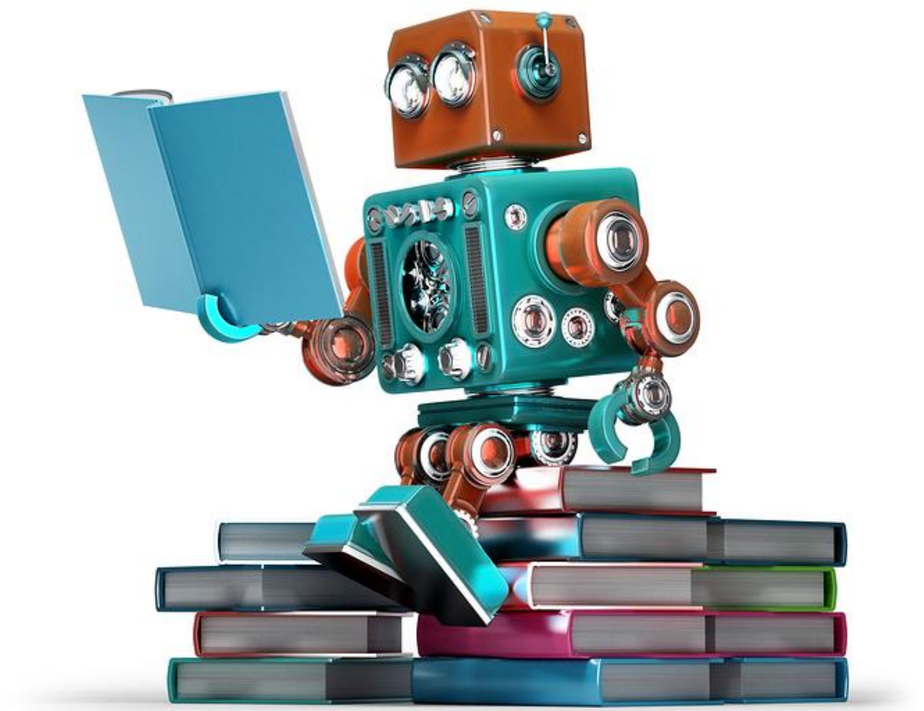
MACHINE REASONING

DAY 3



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

DAY 3 AGENDA

3.1 Technical Machine Inference

3.2 Inference under Uncertainty

3.3 Knowledge Discovery by Machine Learning

3.4 Knowledge Discovery **Workshop**

DAY 3 TIMETABLE

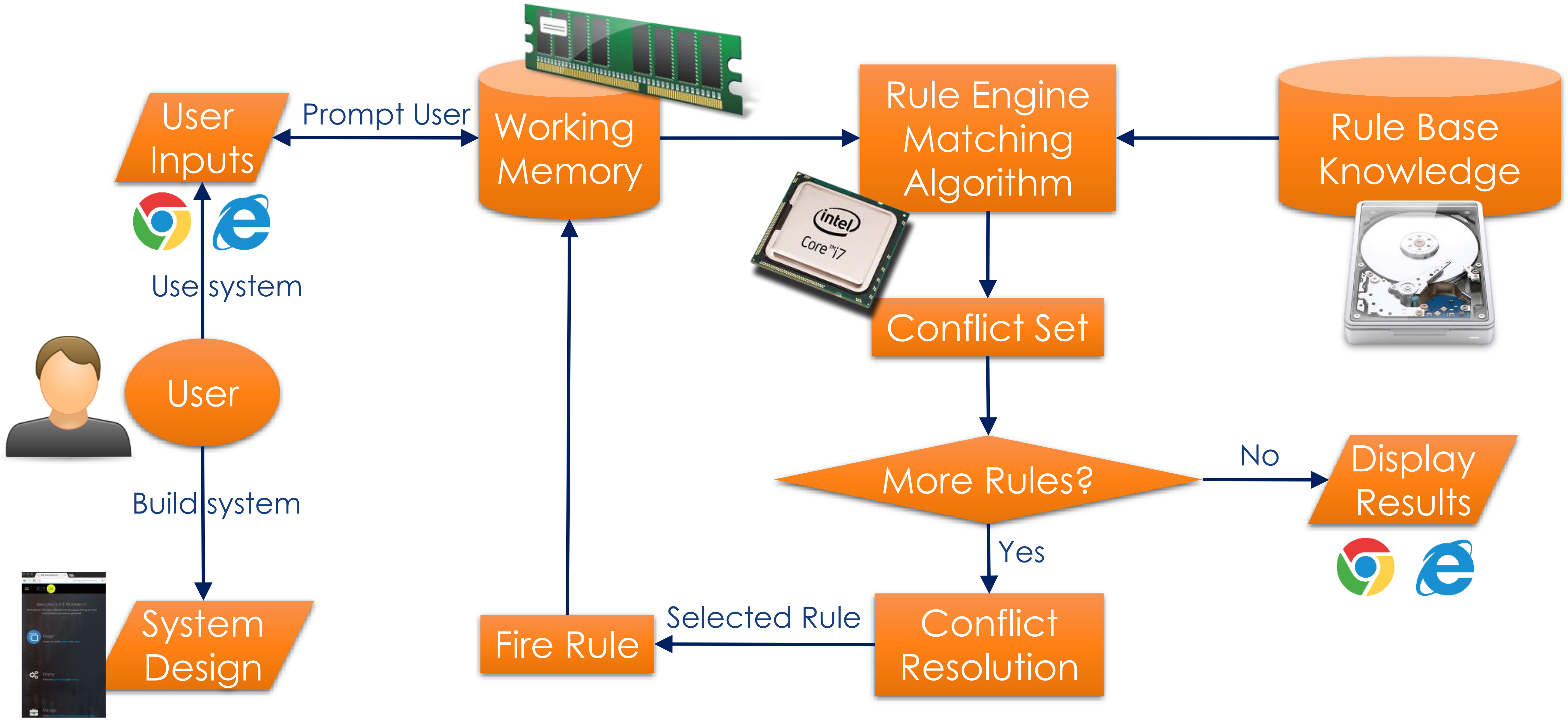
No	Time	Topic	By Whom	Where
1	9 am	3.1 Technical Machine Inference	GU Zhan (Sam)	Class
2	10.10 am	Morning Break		
3	10.30 am	3.2 Inference under Uncertainty	GU Zhan (Sam)	Class
4	12.10 pm	Lunch Break		
5	1.30 pm	3.3 Knowledge Discovery by Machine Learning	GU Zhan (Sam)	Class
6	3.10 pm	Afternoon Break		
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

TECHNICAL MACHINE INFERENCE

3.1 TECHNICAL MACHINE INFERENCE

Recognise-Act Control Cycle



3.1 TECHNICAL MACHINE INFERENCE

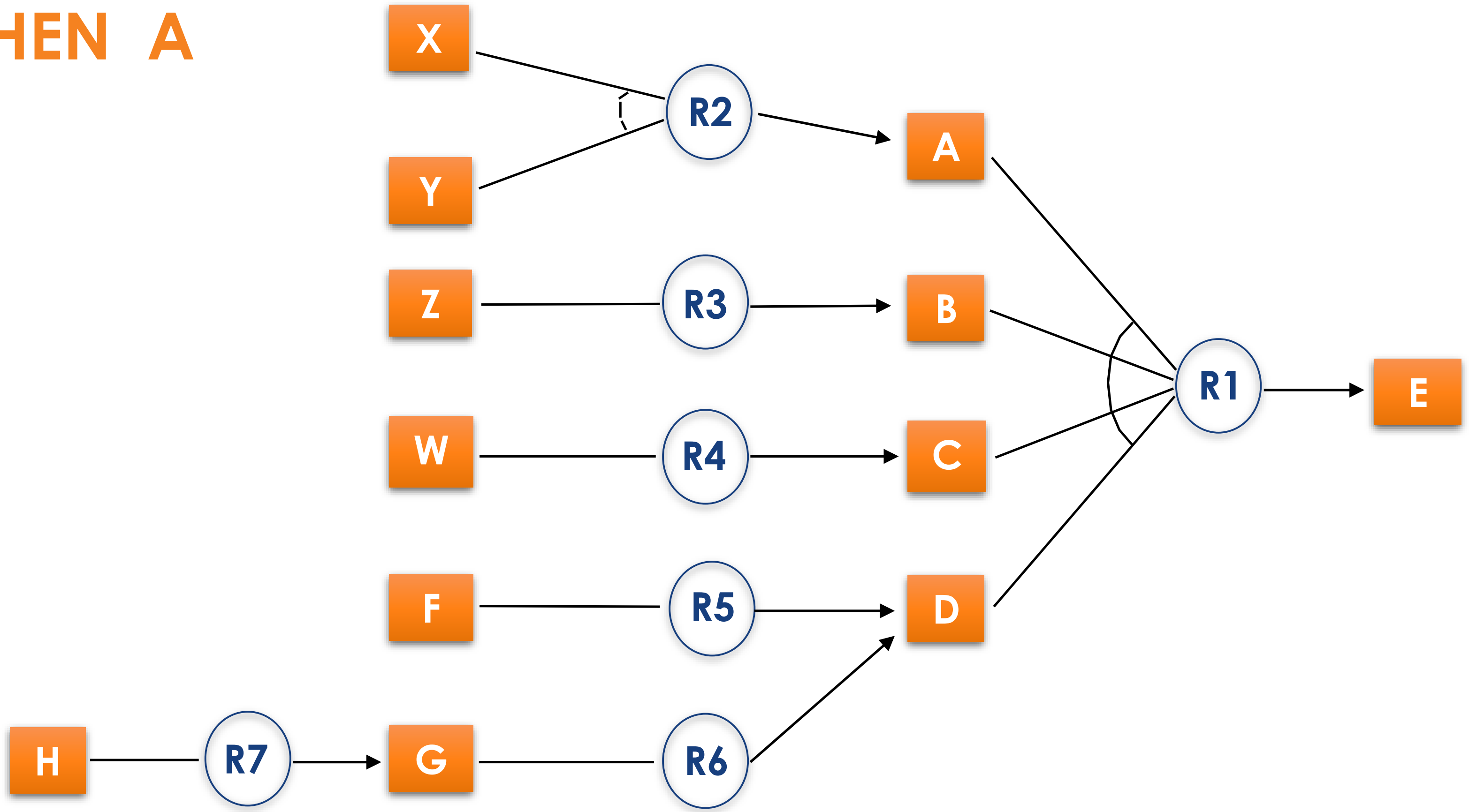
Three main components of Rule Based System (RBS)

- **A set of business rules (Rule Base)**
  - A rule represents a single chunk of problem-solving knowledge
- **A working memory (WM)**
  - Contains Rules and Data (current state of program execution)
- **A rule engine**
  - A computational system that implements the control strategy and applies (fire) the rules
  - The patterns in WM are matched against the conditions of the rules
  - Matched rules are called the conflict set
  - The control strategy determines the order in which the rules are fired and resolves any rule conflicts
  - Uses the Recognise-Act cycle

3.1 TECHNICAL MACHINE INFERENCE

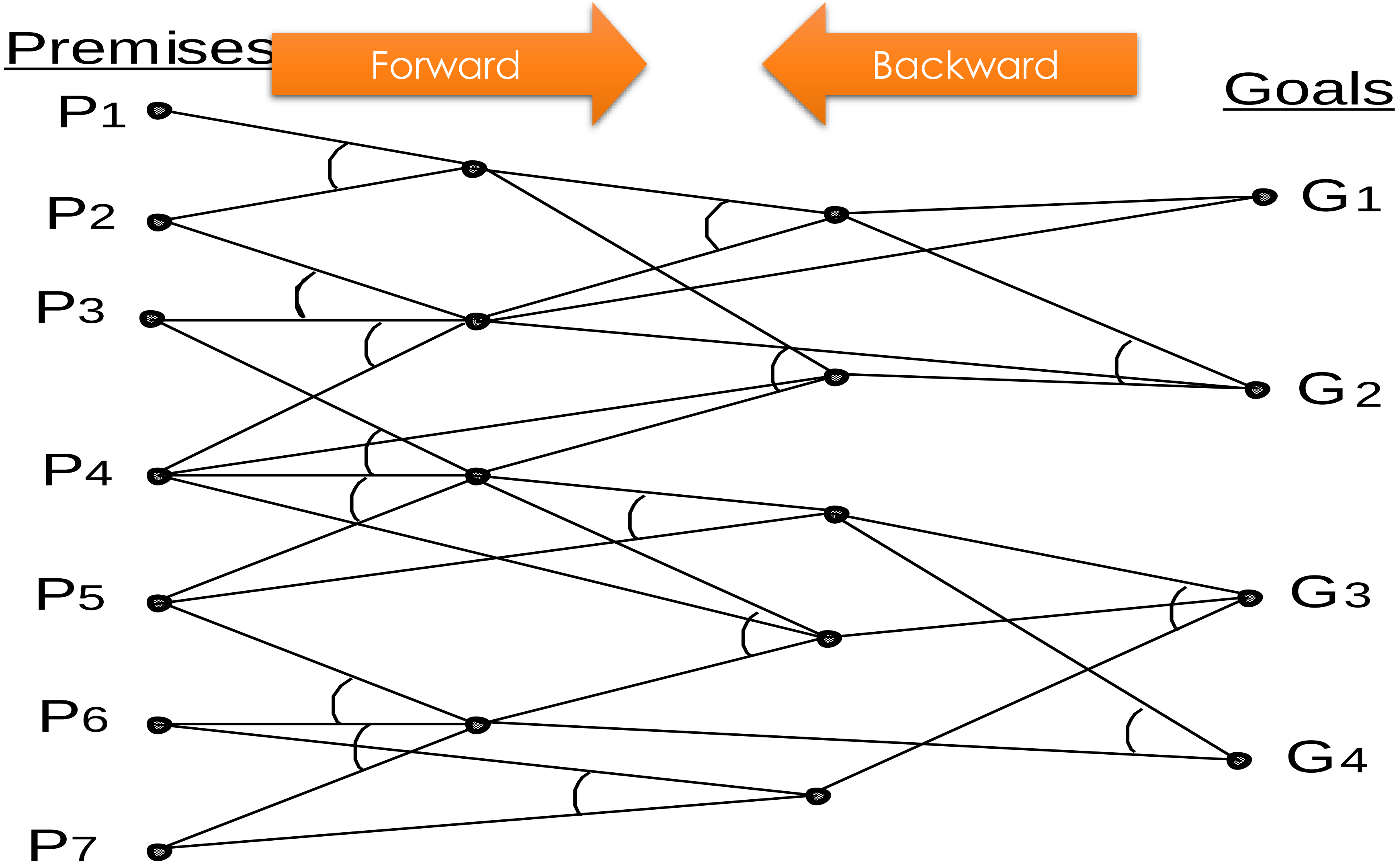
Rules form a Search Tree

- R1 IF A and B and C and D THEN E
- R2 IF X and Y THEN A
- R3 IF Z THEN B
- R4 IF W THEN C
- R5 IF F THEN D
- R6 IF G THEN D
- R7 IF H THEN G



- Inference strategy is known as “**chaining**”.

3.1 TECHNICAL MACHINE INFERENCE
Forward Chaining & Backward Chaining



3.1 TECHNICAL MACHINE INFERENCE

Forward Chaining Example

Rule 1:

IF	the patient has a sore throat
AND	we suspect a Bacterial infection
THEN	we believe the Illness is a strep throat

Rule 2:

IF	the patient's temperature is > 100
THEN	the patient has a fever

Rule 3:

IF	the patient has been sick for over a month
AND	the patient has a fever
THEN	we suspect a bacterial infection

Facts:

Temperature = 104°F (40°C)	F1
Patient has been sick for 2 months	F2
Patient has a sore throat	F3

3.1 TECHNICAL MACHINE INFERENCE

Forward Chaining Example

Rule 1:

IF	the patient has a sore throat
AND	we suspect a Bacterial infection
THEN	we believe the Illness is a strep throat

Rule 2:

IF	the patient's temperature is > 100
THEN	the patient has a fever

Rule 3:

IF	the patient has been sick for over a month
AND	the patient has a fever
THEN	we suspect a bacterial infection

Facts:

Temperature = 104°F (40°C)	F1 ←
Patient has been sick for 2 months	F2
Patient has a sore throat	F3
patient has a fever	→ F4

3.1 TECHNICAL MACHINE INFERENCE

Forward Chaining Example

Rule 1:

IF	the patient has a sore throat
AND	we suspect a Bacterial infection
THEN	we believe the Illness is a strep throat

Rule 2:

IF	the patient's temperature is > 100
THEN	the patient has a fever

Rule 3:

IF	the patient has been sick for over a month
AND	the patient has a fever
THEN	we suspect a bacterial infection

Facts:

Temperature = 104ºF (40ºC)	F1
Patient has been sick for 2 months	F2 ⬅
Patient has a sore throat	F3
patient has a fever	F4 ⬅
suspect a bacterial infection	➡ F5

3.1 TECHNICAL MACHINE INFERENCE

Forward Chaining Example

Rule 1:

IF	the patient has a sore throat
AND	we suspect a Bacterial infection
THEN	we believe the illness is a strep throat

Rule 2:

IF	the patient's temperature is > 100
THEN	the patient has a fever

Rule 3:

IF	the patient has been sick for over a month
AND	the patient has a fever
THEN	we suspect a bacterial infection

Facts:

Temperature = 104°F (40°C)	F1
Patient has been sick for 2 months	F2
Patient has a sore throat	F3
patient has a fever	F4
suspect a bacterial infection	F5
illness is a strep throat	F6

3.1 TECHNICAL MACHINE INFERENCE

Forward Chaining Example

Initial state of working memory WM: [F1, F2, F3]

Rule-2 [F1]

Add F4: “patient has a fever”

Rule-3 [F2, F4]

Add F5: “suspect a bacterial infection”

Rule-1 [F3, F5]

Add F6: “illness is a strep throat”

No more rules to fire → halt

Conclusion → illness is a strep throat

Forward Chaining Definition

- **A forward chaining system**
  - Begin with a set of facts in the Working Memory, then apply rules to generate new facts until the desired goal is reached.
- **Rules whose premise (IF..) is known to be true are fired, and their conclusions (THEN..) are declared true.**
  - This process continues until no more rules can be triggered/fired. The system then reports its conclusions.

3.1 TECHNICAL MACHINE INFERENCE
Backward Chaining Example

Rule 1:
IF the patient has a sore throat
AND we suspect a Bacterial infection
THEN we believe the illness is a strep throat

Rule 2:
IF the patient's temperature is > 100°F
THEN the patient has a fever

Rule 3:
IF the patient has been sick for over a month
AND the patient has a fever
THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)
F2: Patient has been sick for 2 months
F3: Patient has a sore throat
F4: Is the illness a strep throat?

goal / hypothesis to prove

3.1 TECHNICAL MACHINE INFERENCE

Backward Chaining Example

Rule 1:
- IF the patient has a sore throat
- AND we suspect a Bacterial infection
- THEN we believe the illness is a strep throat

Rule 2:
- IF the patient's temperature is > 100°F
- THEN the patient has a fever

Rule 3:
- IF the patient has been sick for over a month
- AND the patient has a fever
- THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)
F2: Patient has been sick for 2 months
F3: Patient has a sore throat
F4: Is the illness a strep throat?

3.1 TECHNICAL MACHINE INFERENCE

Backward Chaining Example

Rule 1:

IF the patient has a sore throat ☑
AND we suspect a Bacterial infection [?]
THEN we believe the illness is a strep throat

Rule 2:

IF the patient's temperature is > 100°F
THEN the patient has a fever

Rule 3:

IF the patient has been sick for over a month
AND the patient has a fever
THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)
F2: Patient has been sick for 2 months
F3: Patient has a sore throat
F4: Is the illness a strep throat?

3.1 TECHNICAL MACHINE INFERENCE

Backward Chaining Example

Rule 1:
- IF the patient has a sore throat ☑
- AND we suspect a Bacterial infection [?]
- THEN we believe the illness is a strep throat

Rule 2:
- IF the patient's temperature is > 100°F
- THEN the patient has a fever

Rule 3:
- IF the patient has been sick for over a month
- AND the patient has a fever
- THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)

F2: Patient has been sick for 2 months

F3: Patient has a sore throat

F4: Is the illness a strep throat?

F5: Do we suspect a bacterial infection?

3.1 TECHNICAL MACHINE INFERENCE

Backward Chaining Example

Rule 1:

IF the patient has a sore throat ☑

AND we suspect a Bacterial infection [?]

THEN we believe the illness is a strep throat

Rule 2:

IF the patient's temperature is > 100°F

THEN the patient has a fever

Rule 3:

IF the patient has been sick for over a month

AND the patient has a fever [?]

THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)

F2: Patient has been sick for 2 months

F3: Patient has a sore throat

F4: Is the illness a strep throat?

F5: Do we suspect a bacterial infection?

3.1 TECHNICAL MACHINE INFERENCE

Backward Chaining Example

Rule 1:

IF the patient has a sore throat ☑

AND we suspect a Bacterial infection [?]

THEN we believe the illness is a strep throat

Rule 2:

IF the patient's temperature is > 100°F

THEN the patient has a fever

Rule 3:

IF the patient has been sick for over a month ☑

AND the patient has a fever [?]

THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)

F2: Patient has been sick for 2 months

F3: Patient has a sore throat

F4: Is the illness a strep throat?

F5: Do we suspect a bacterial infection?

F6: Does the patient have a fever?

3.1 TECHNICAL MACHINE INFERENCE

Backward Chaining Example

Rule 1:
IF the patient has a sore throat ☑
AND we suspect a Bacterial infection [?]
THEN we believe the illness is a strep throat

Rule 2:
IF the patient's temperature is > 100°F ☑
THEN the patient has a fever

Rule 3:
IF the patient has been sick for over a month ☑
AND the patient has a fever [?]
THEN we suspect a bacterial infection

F1: Temperature = 104°F (40°C)
F2: Patient has been sick for 2 months
F3: Patient has a sore throat
F4: Is the illness a strep throat?
F5: Do we suspect a bacterial infection?
F6: Does the patient have a fever?

Proved: Patient has a strep throat.

3.1 TECHNICAL MACHINE INFERENCE

Backward Chaining Example

- **Initial state of working memory WM: [Facts + Hypothesis]**

Rule-1 [Hypothesis F4]

Goal: we suspect a strep throat? **{new goal to pursue Add F4}**

Check: patient has a sore throat? **{proved by F3}**

Check: we suspect bacterial infection? **{new goal to pursue Add F5}**

Rule-3 [F5]

Check: patient has been sick for over a week? **{proved by F2}**

Check: patient has a fever? **{new goal to pursue Add F6}**

Rule-2 [F6]

Check: patient's temperature is > 100°F (37.8°C)? **{proved by F1}**

No more rules to fire {all proved} → halt

Conclusion → yes, illness is a strep throat

3.1 TECHNICAL MACHINE INFERENCE

Backward Chaining Definition

- **A backward chaining inference engine starts from a goal or hypothesis.**
  - It works through the rules trying to match the goal with the action clauses (THEN part) of a rule.
  - When a match is found, the condition clauses (IF part) of the matching rule become "sub-goals".
  - The cycle is repeated until a verifiable set of condition clauses is found.

3.1 TECHNICAL MACHINE INFERENCE

Forward Chaining vs. Backward Chaining

Forward Chaining	Backward Chaining
Planning, monitoring, surveillance, control, decision	Diagnosis, trouble-shooting
Present to future Antecedent to consequent	Present to past Consequent to antecedent
Data driven, bottom-up reasoning	Goal driven, top-down reasoning
Work forward to find what solutions that follow from the facts	Work backwards to find facts that support a given hypothesis
Facilitates Breadth-First-Search	Facilitates Depth-First-Search
Does not facilitate Explanation	Facilitates Explanation
CLIPS, KIE Drools	PROLOG, KIE Drools

3.1 TECHNICAL MACHINE INFERENCE

Forward Chaining vs. Backward Chaining

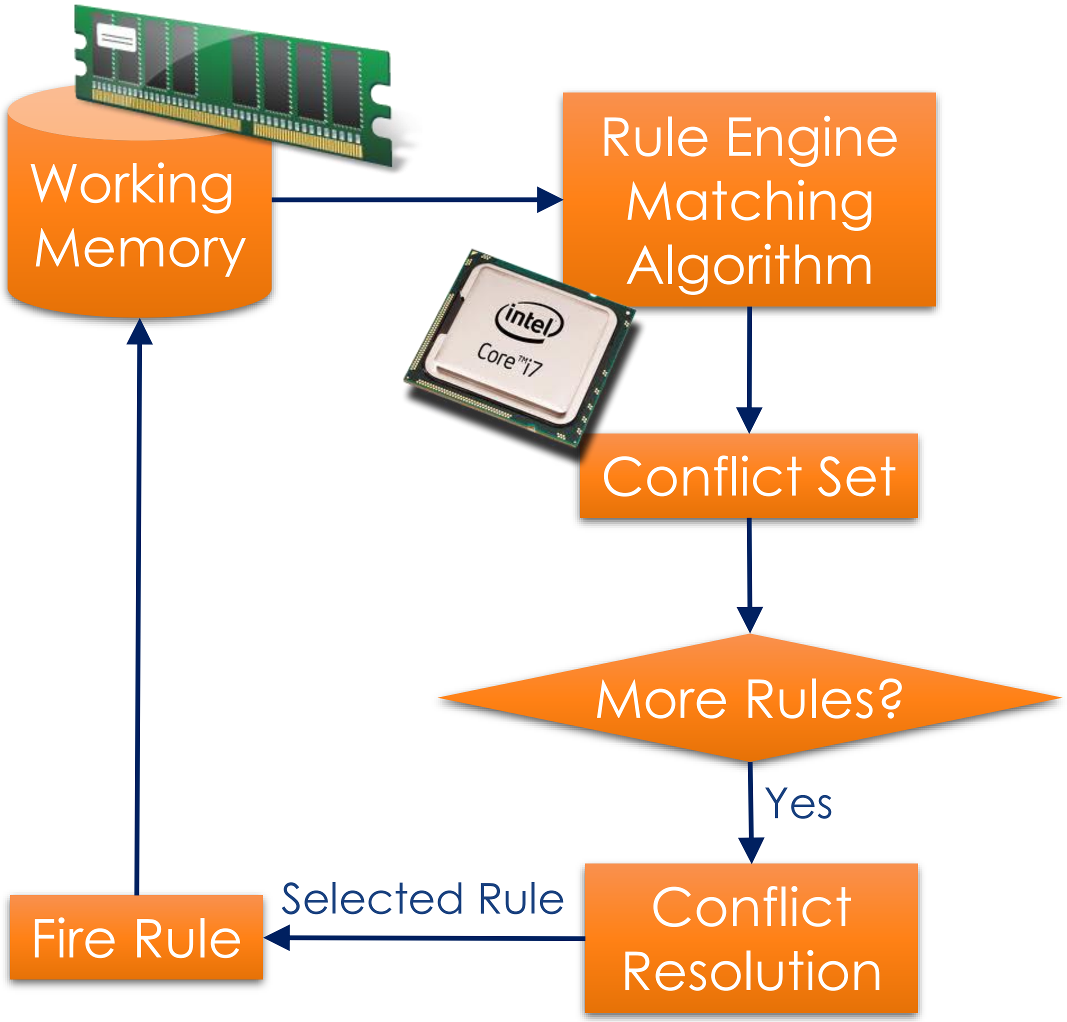
- **FC is data-driven**
  - The focus of attention starts form known data & business rules
    - e.g., object recognition, routine decisions
    - May do lots of work that is irrelevant to the goal
- **BC is goal-driven**
  - Appropriate for problem-solving & investigation
    - e.g., Where are my keys? How do I get into a PhD program?

☺ Computer memory consumption of BC can be much less than FC when knowledge base is large.

3.1 TECHNICAL 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 TECHNICAL 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 TECHNICAL 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 TECHNICAL 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 TECHNICAL 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

3.2

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)

3.2.1 INFERENCE UNDER UNCERTAINTY

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: $-1.0 \le CF \le +1.0$
  - 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

3.2.1 INFERENCE UNDER UNCERTAINTY

Reasoning with Certainty Factors

- **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

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

- 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)

Fuzzy Logic (FL)

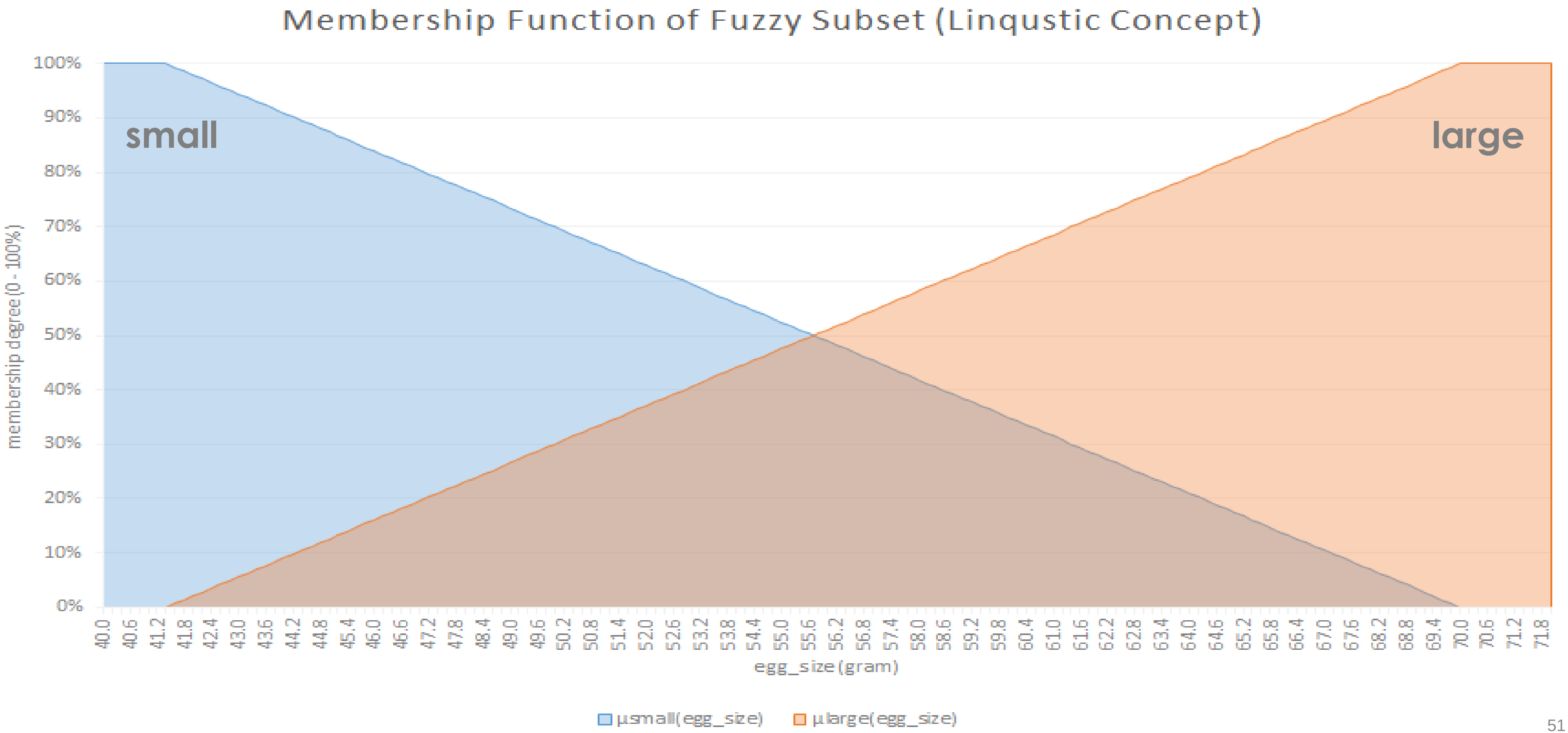
- Fuzzy logic is an approach to computing based on "**degrees of truth**" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based.
- Fuzzy logic is close to the way our brains work. We aggregate data and form **a number of partial truths** which we aggregate 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

- **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

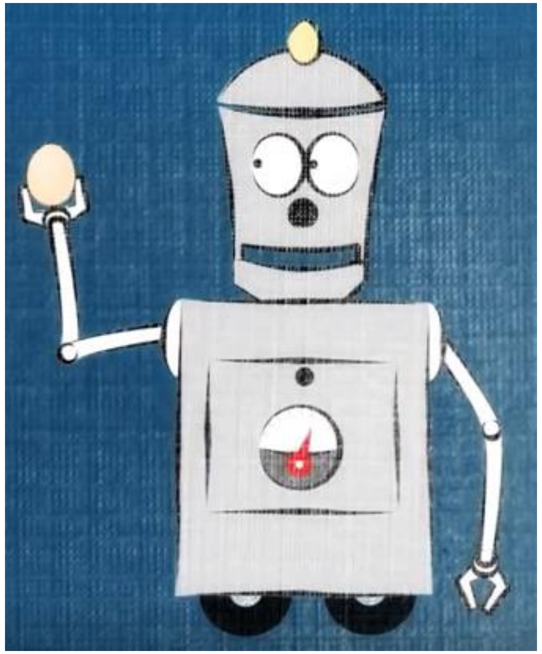
Fuzzy Logic in Rules : Fuzzification



3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic in Rules : Inference

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

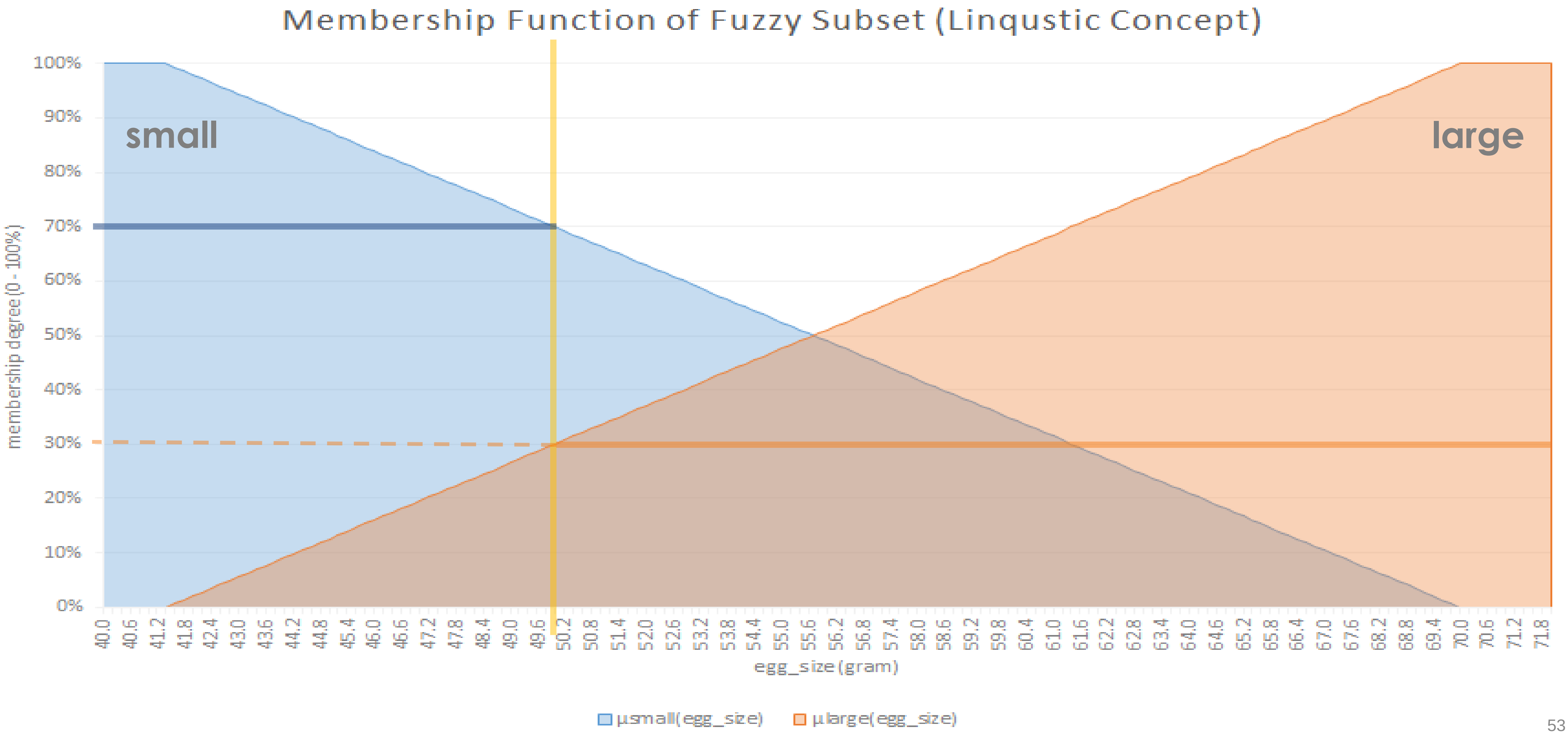


Link https://www.youtube.com/watch?v=J_Q5X0nTmrA

- **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

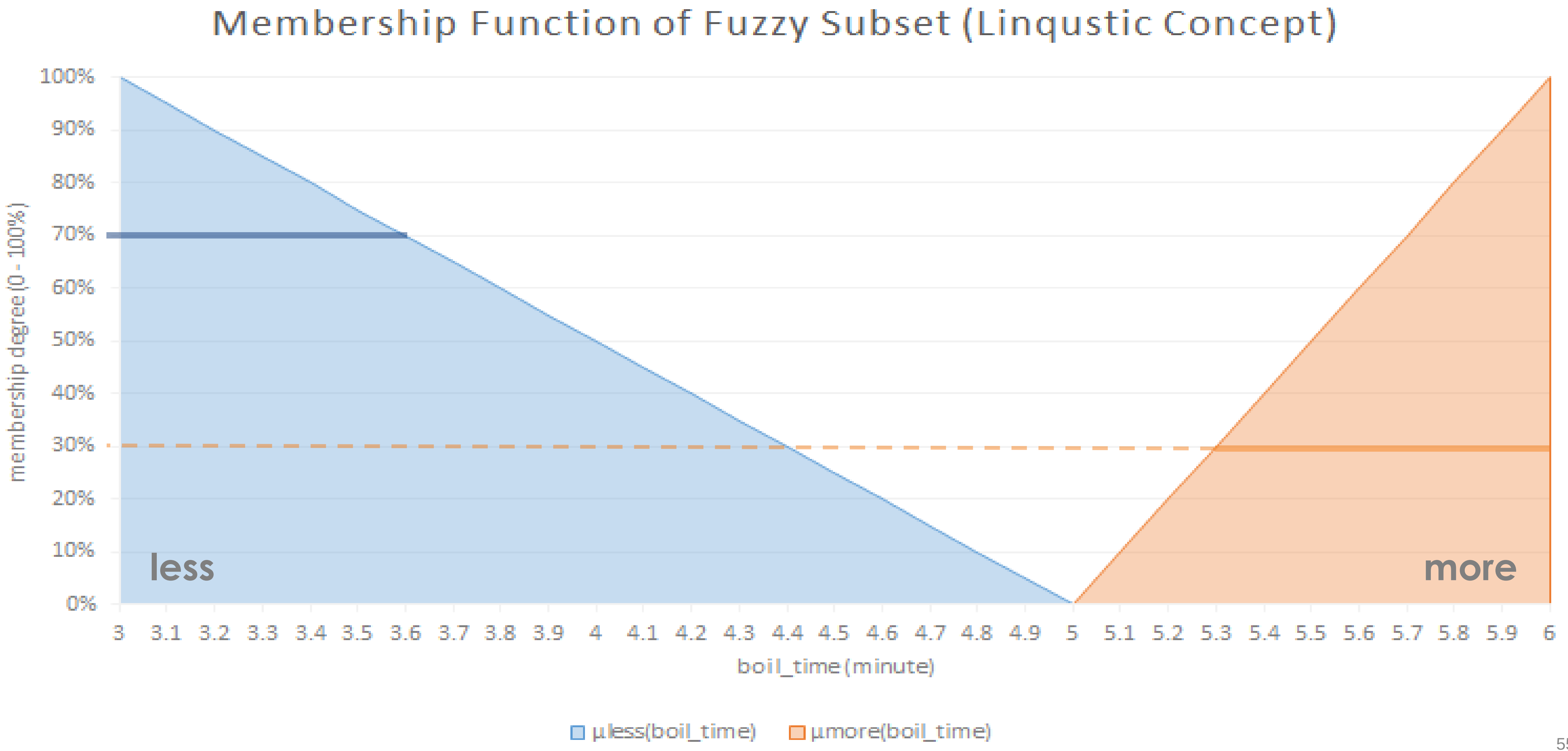
- Egg-boiling Fuzzy Ruleset

  - **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%)**

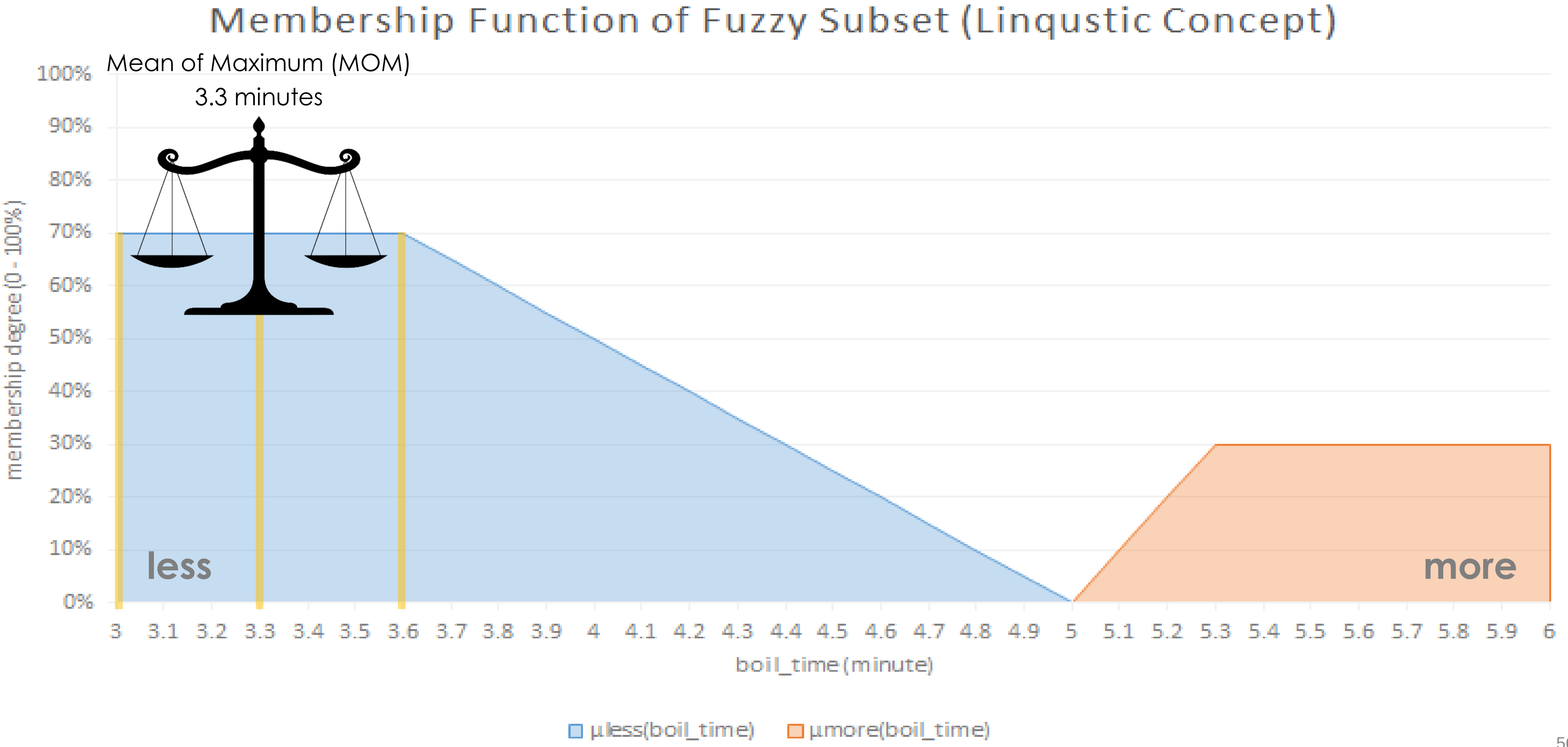
3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic in Rules : Inference



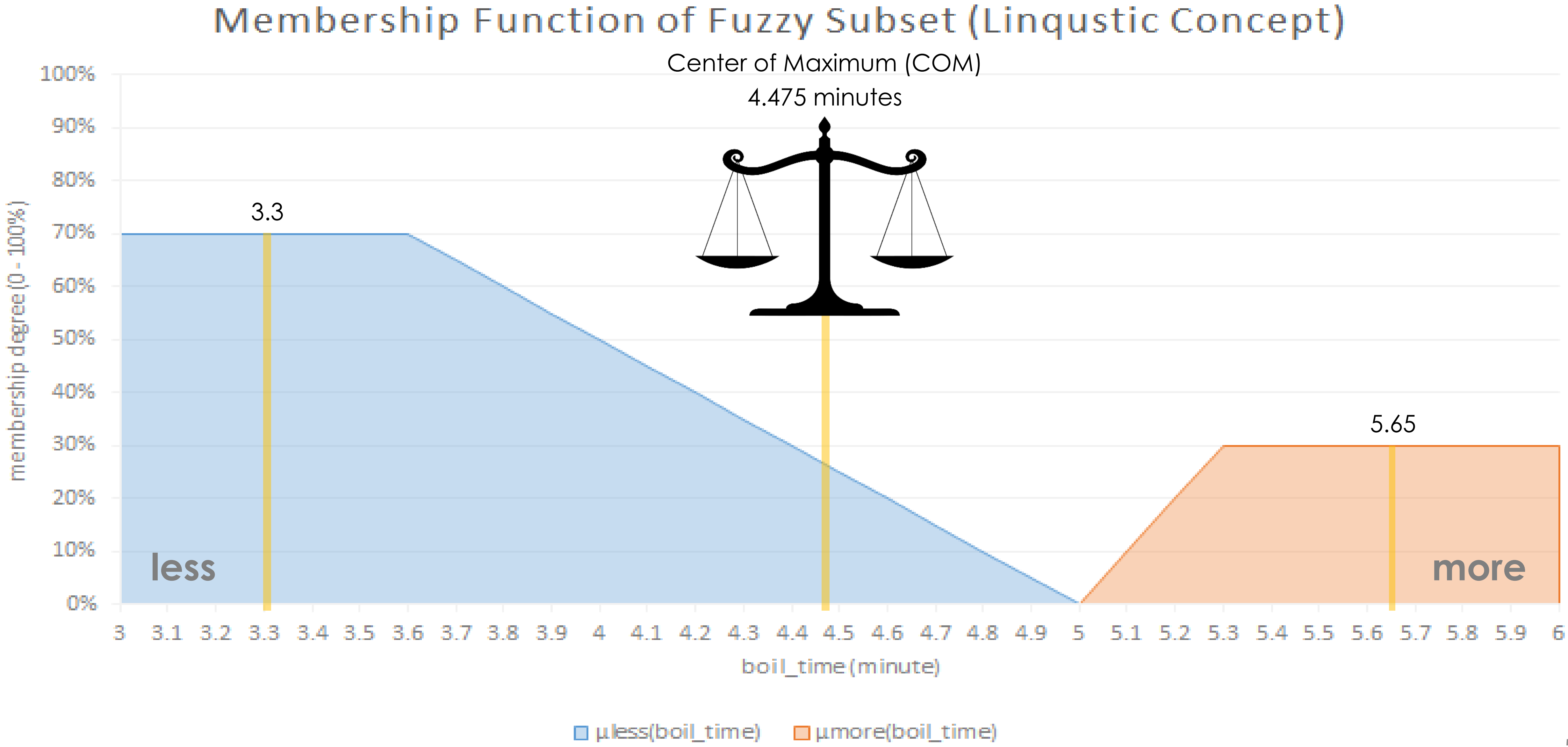
3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic in Rules : Defuzzification



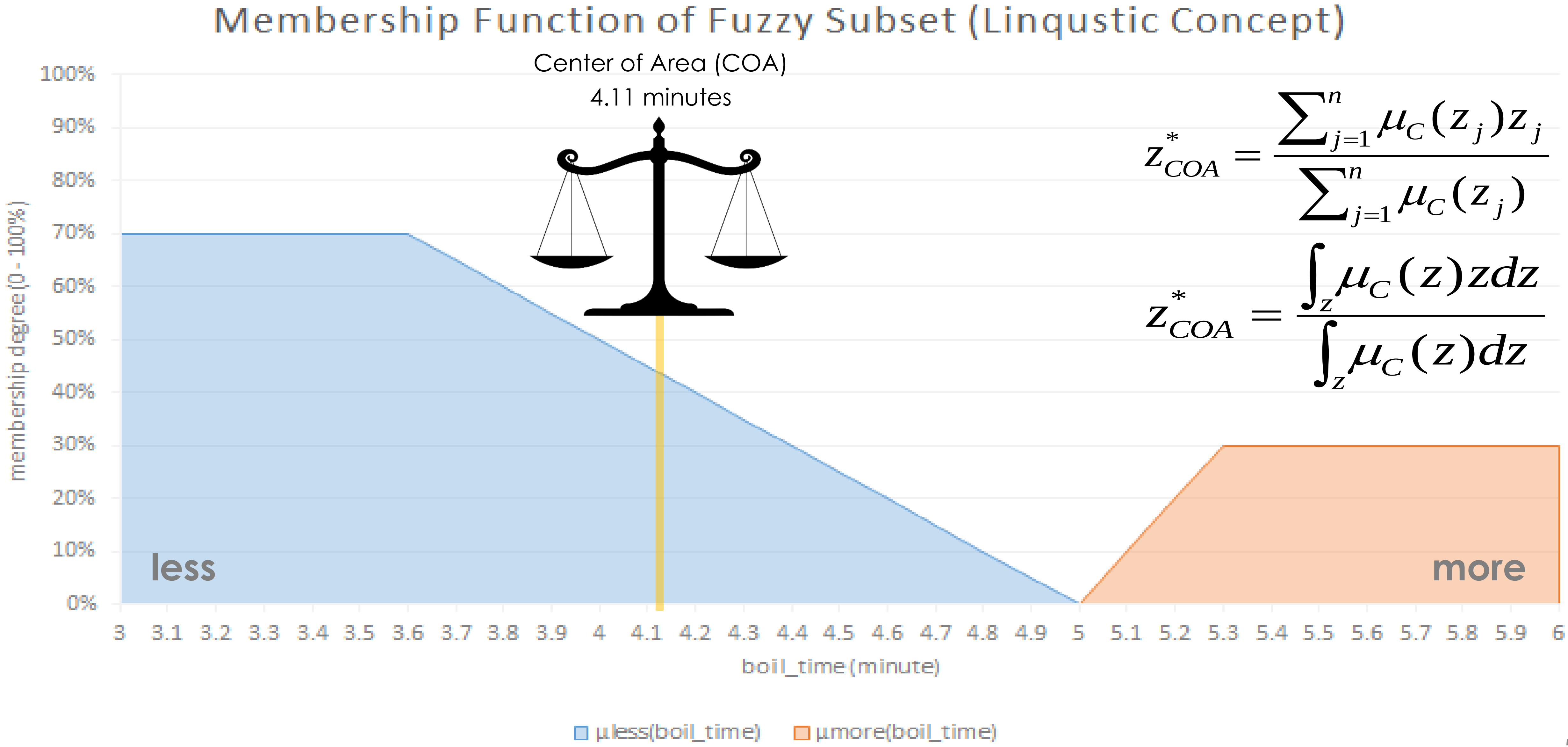
3.2.2 INFERENCE UNDER UNCERTAINTY

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)	μless(boil_time)	μmore(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()**

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 Applications

Chem. Tech	Polymer production,
Computer Tech.	Fuzzy Neural Networks
Entertainment industry	TVs, Camcorders
Household appliances	Cookers, Dishwashers, Wash machines
Industrial plants	Blast furnaces, Cement Kilns
Medicine	Disease diagnosis, Pacemakers
Optical equipment	Cameras, light sensors
Physics	Fuzzy Chaos, fuzzy simulation
Pollution control	Oil spill monitors
Robotics	Process controllers, Cranes
Stock market	Fund mgmt., trend prediction
Food Industry	Electronic Nose non-destructive detector

3.2.2 INFERENCE UNDER UNCERTAINTY

Fuzzy Logic Applications

Transportation	Cars, Buses, Trains
Mathematics	Fuzzy Integral, Fuzzy Metric Spaces
Operations Research	Fuzzy Optimization, Fuzzy Games
Economics	Fuzzy Supply-fuzzy Demand models
Social sciences	Modeling Fuzzy behaviors
Management	Fuzzy Decision Making models
Statistics	Fuzzy Cluster Analysis, Fuzzy Regression Analysis
Financial Engineering	Modeling fuzzy behavior of customers
Reliability Engineering	Fuzzy Reliability Analysis
Nuclear Science/Engg.	Fuzzy Safety Analysis of Nuclear Reactors
Data Mining	Fuzzy Data Mining; Algorithmic Trading

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

Supplementary Bayesian Reasoning (Probabilistic Inference)

P(man with long hair) = P(long hair) * P(man | long hair)

P(long hair and man) = P(man) * P(long hair | man)

Because P(man and long hair) = P(long hair and man)

P(long hair) * P(man | long hair) = P(man) * P(long hair | man)

P(man | long hair) = P(man) * P(long hair | man) / P(long hair)

P(A | B) = P(B | A) * P(A) / P(B)

Bayesian inference

https://en.wikipedia.org/wiki/Bayesian_inference

How Bayesian inference works

https://brohrer.github.io/how_bayesian_inference_works.html

P(man | long hair) = [P(man) * P(long hair | man)] / [P(woman with_and long hair) + P(man with_and long hair)]

= (0.5 * 0.04) / (0.25 + 0.02) = 0.02 / 0.27 = 0.07 (7%)

3.2.2 INFERENCE UNDER UNCERTAINTY

Supplementary Test suggested dengue virus in my blood!

Table 7 **Overall accuracy of physician's dengue diagnosis with NS1 rapid test result.**

Physician's diagnosis (Medical test result: positive/negative?)	Confirmed diagnosis (Virus in blood?)		
	Dengue (virus in blood)	Non-dengue (no-virus in blood)	Total
Dengue (Medical test: positive)	137	44	181
Non-dengue (Medical test: negative)	46	170	216
Total	183	214	397

Sensitivity 75%; Specificity 79%

A sample test to estimate the medical test performance, which can be considered as a (Data-Driven Bayesian Machine Learning) predictive model. It's hard to blood test every person in Singapore, again and again.

https://doi.org/10.1371/journal.pntd.0006573.t007

P(Hypothesis: **virus** | Evidence: **positive**)

Expected positive proportion if all 5 mil people went testing.

= [P(virus) * P(positive | virus)] / P(positive)

Prior probability: around 250 / 5 million people = 0.00005 = 0.005% (Based on NEA weekly data)

= [P(virus) * P(positive | virus)] / [P(positive and no-virus) + P(positive and virus)]

= Sensitivity

= [P(virus) * P(positive | virus)] / [P(positive | no-virus) * P(no-virus) + P(positive | virus) * P(virus)]

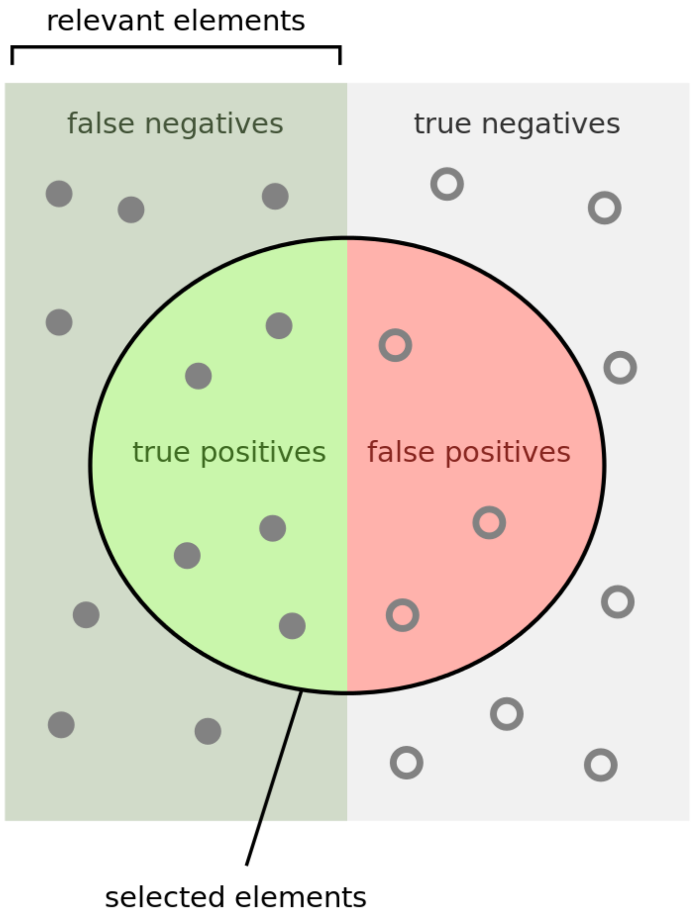
= 1 - Specificity

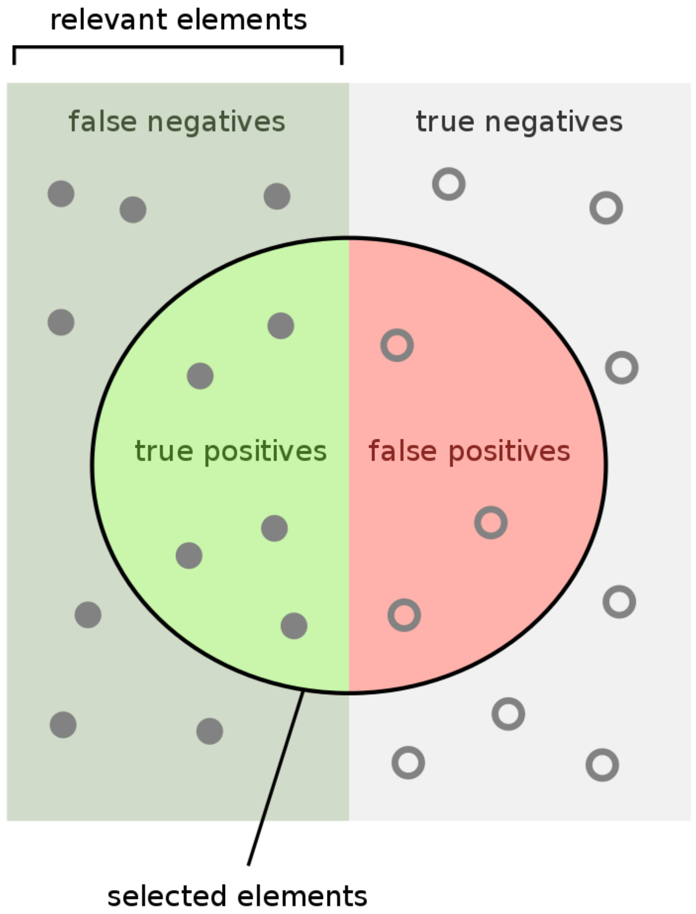
= (0.00005 * 137/183) / [(44/214) * (1-0.00005) + (137/183) * 0.00005]

= (0.00005 * 0.7486) / (0.2055597 + 0.0000374) = 0.000182 ≈ 0.02%

Accuracy of dengue clinical diagnosis with and without NS1 antigen rapid test: Comparison between human and Bayesian network model decision, Sa-ngamuang, Chaitawat; Haddawy, Peter; Luvira, Viravarn; Piyaphanee, Watcharapong; Iamsirithaworn, Sopon; et al. PLoS Neglected Tropical Diseases; San Francisco Vol. 12, Iss. 6, (Jun 2018): e0006573. DOI:10.1371/journal.pntd.0006573
https://journals.plos.org/plosntds/article?id=10.1371/journal.pntd.0006573

Singapore The National Environment Agency NEA: Dengue Cases:
https://www.nea.gov.sg/dengue-zika/dengue/dengue-cases





How many selected items are relevant?

Precision = 

How many relevant items are selected?

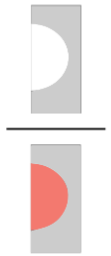
Recall = 

= (same as)

How many relevant items are selected? e.g. How many sick people are correctly identified as having the condition.

Sensitivity= 

How many negative selected elements are truly negative? e.g. How many healthy peple are identified as not having the condition.

Specificity = 

3.3

KNOWLEDGE DISCOVERY BY MACHINE LEARNING

3.3 KNOWLEDGE DISCOVERY BY MACHINE LEARNING

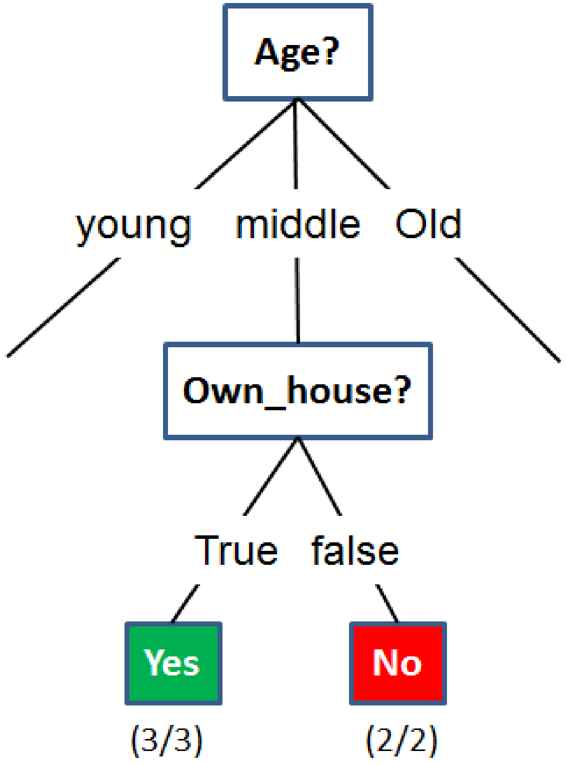
Bank Loan Example – Business Background

- 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.

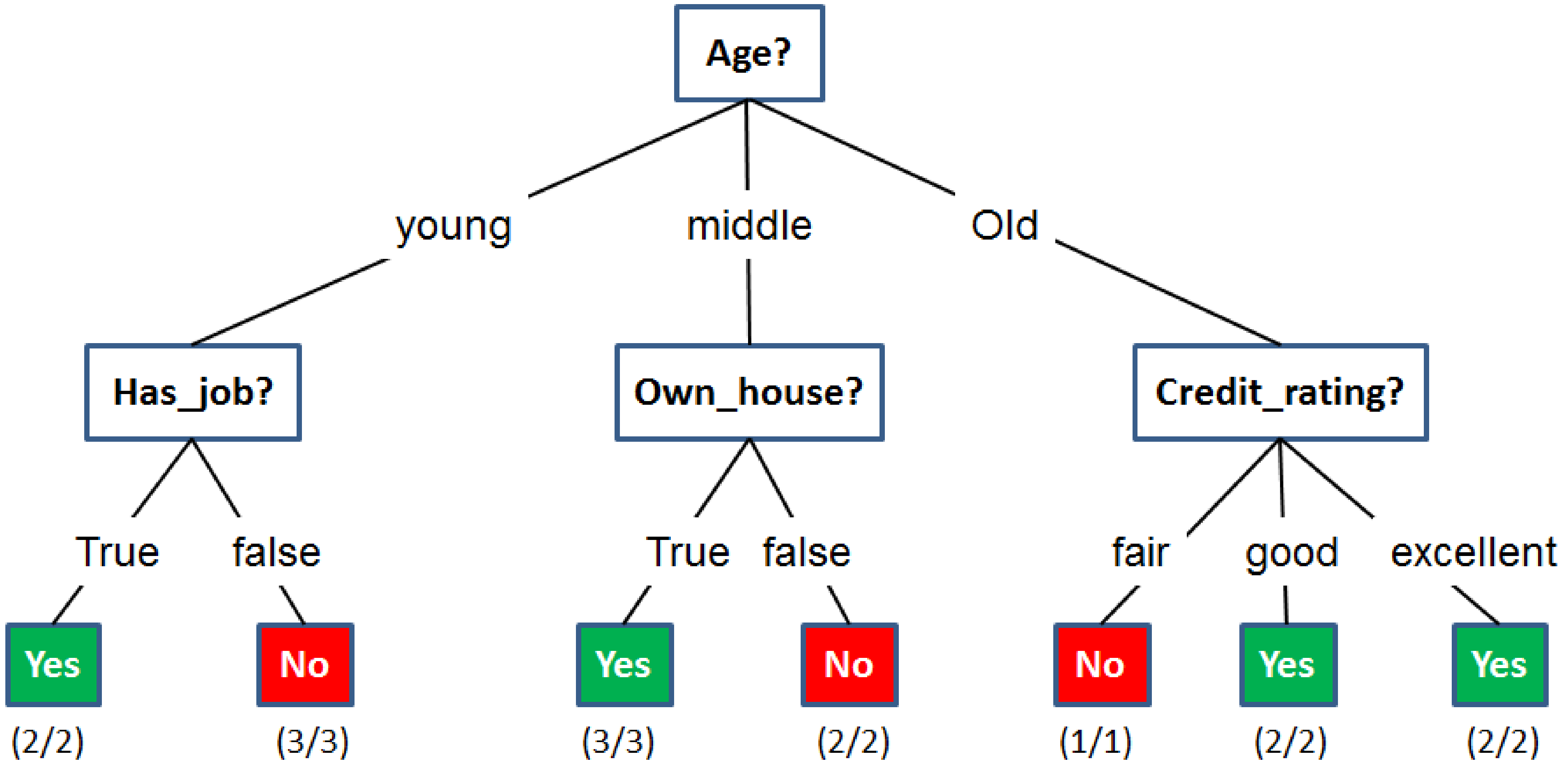
3.3 KNOWLEDGE DISCOVERY BY MACHINE LEARNING

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

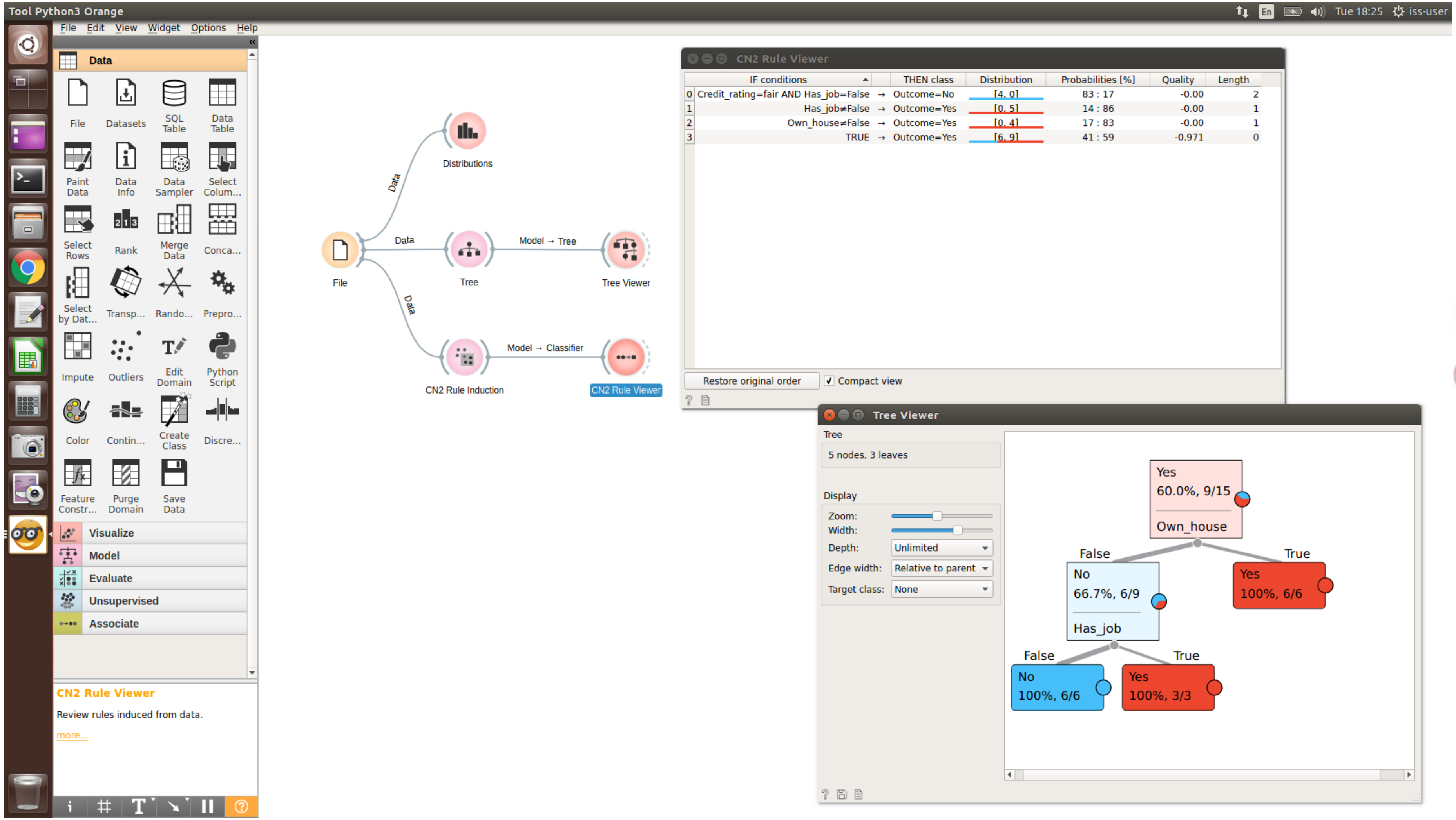


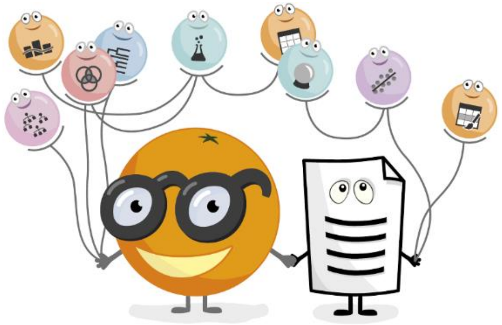
Bank Loan Example – Decision Tree



3.3 KNOWLEDGE DISCOVERY BY MACHINE LEARNING

Data Mining Tool: Orange3 (python)

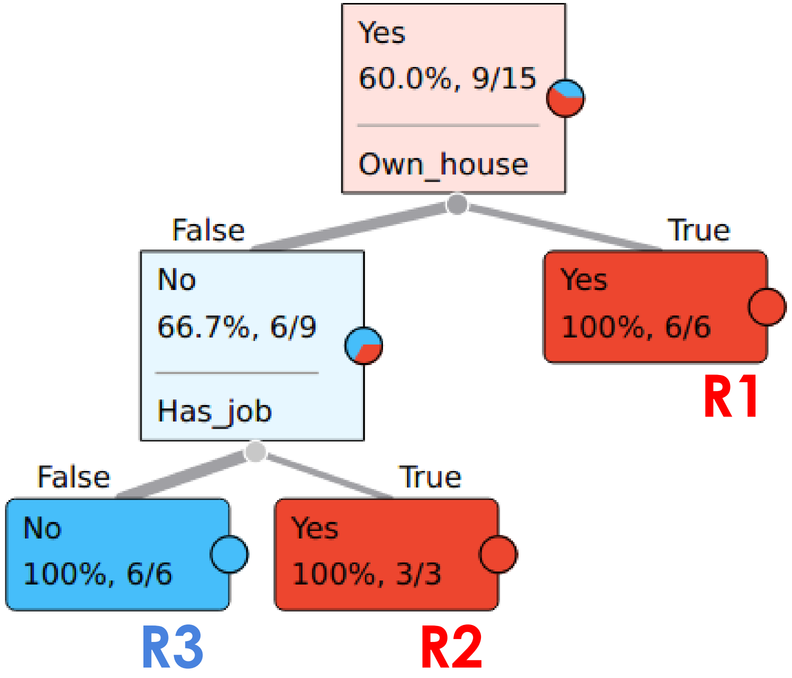




3.3 KNOWLEDGE DISCOVERY BY MACHINE LEARNING

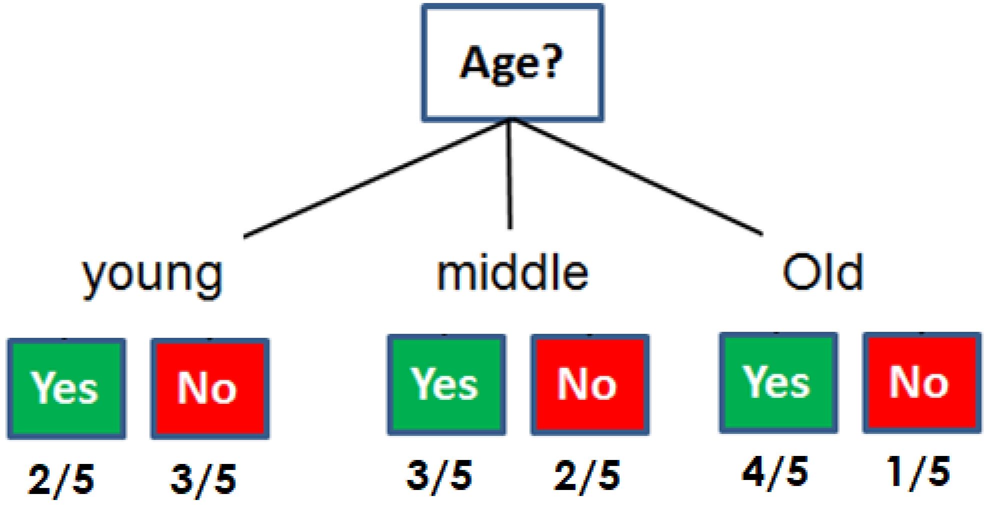
Orange3 Bank Loan Example – Decision Tree

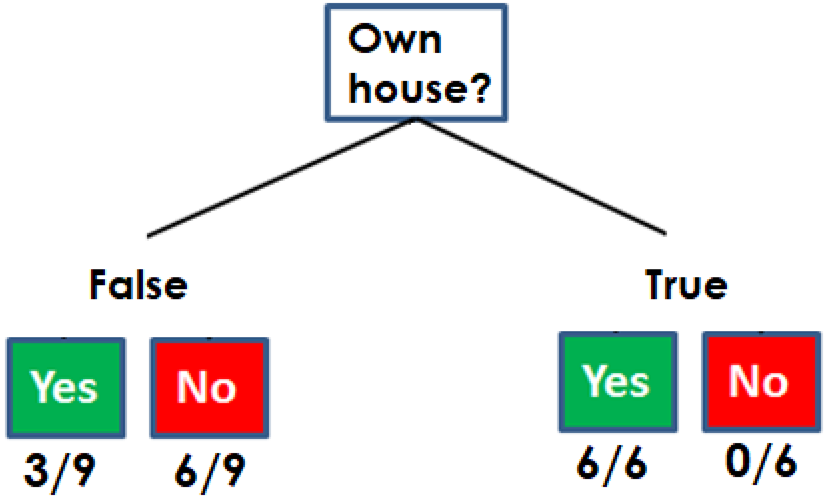
ID	Age	Has_job	Own_house	Credit_rating	Outcome	
1	young	False	False	fair	No	R3
2	young	False	False	good	No	
3	young	True	False	good	Yes	R2
4	young	True	True	fair	Yes	R1
5	young	False	False	fair	No	
6	middle	False	False	fair	No	R3
7	middle	False	False	good	No	
8	middle	True	True	good	Yes	
9	middle	False	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



3.3 KNOWLEDGE DISCOVERY BY MACHINE LEARNING

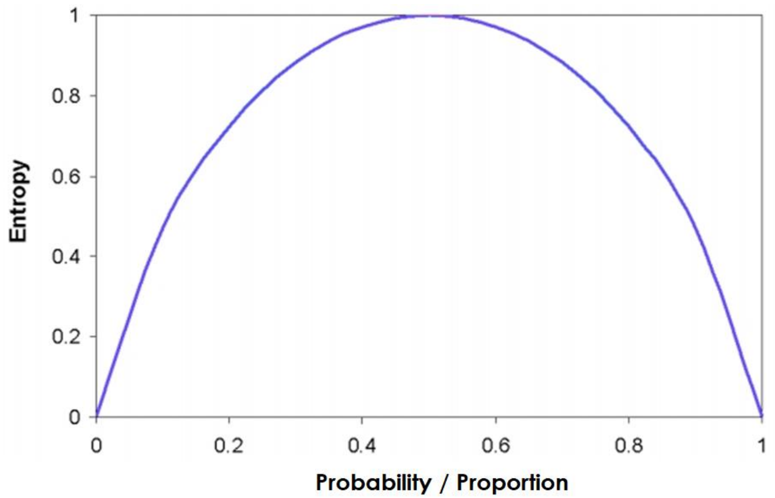
Decision Tree Algorithm – Which feature to select for split?





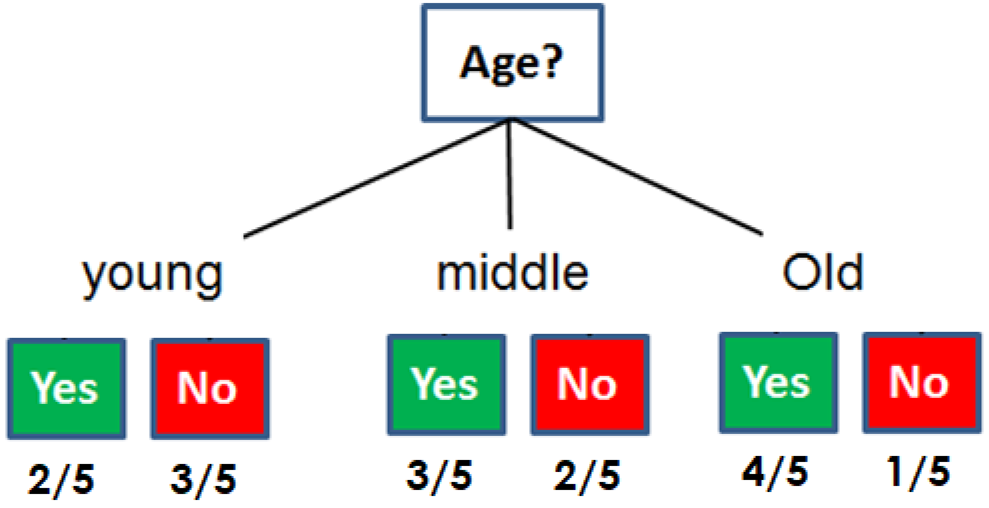
$$Imp(D_j) = Entropy(p) = -p \log p - (1-p)\log(1-p)$$

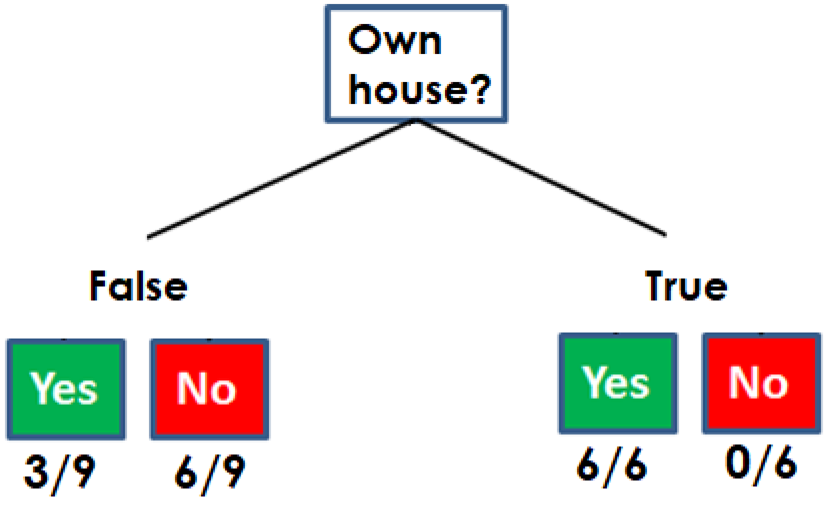
$$Imp(\{D_1, \ldots, D_l\}) = \sum_{j=1}^{l} \frac{|D_j|}{|D|} Imp(D_j)$$



3.3 KNOWLEDGE DISCOVERY BY MACHINE LEARNING

Decision Tree Algorithm – Which feature to select for 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

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

= 0.551 + 0

= 0.551 ☺ Smaller impurity, the better conditional 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

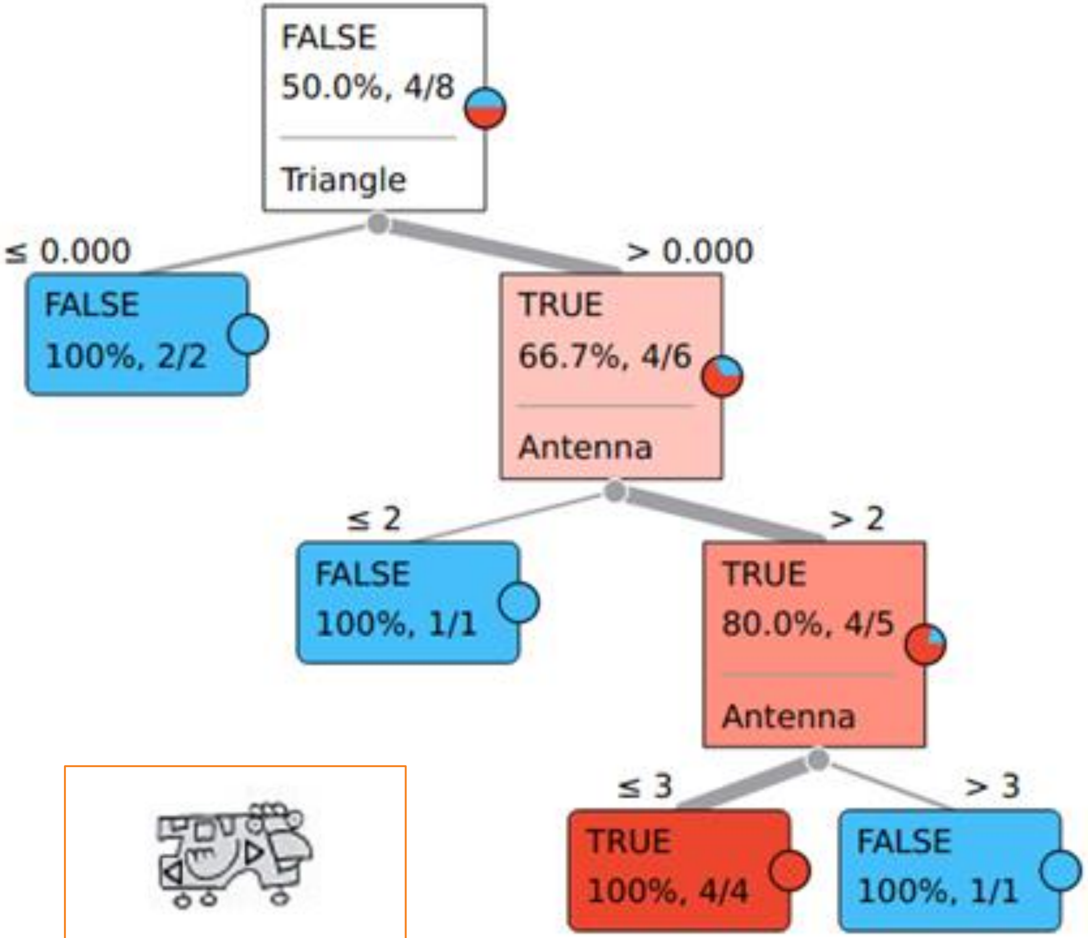
Aliens

Not aliens

Which one is alien?

A B C D E

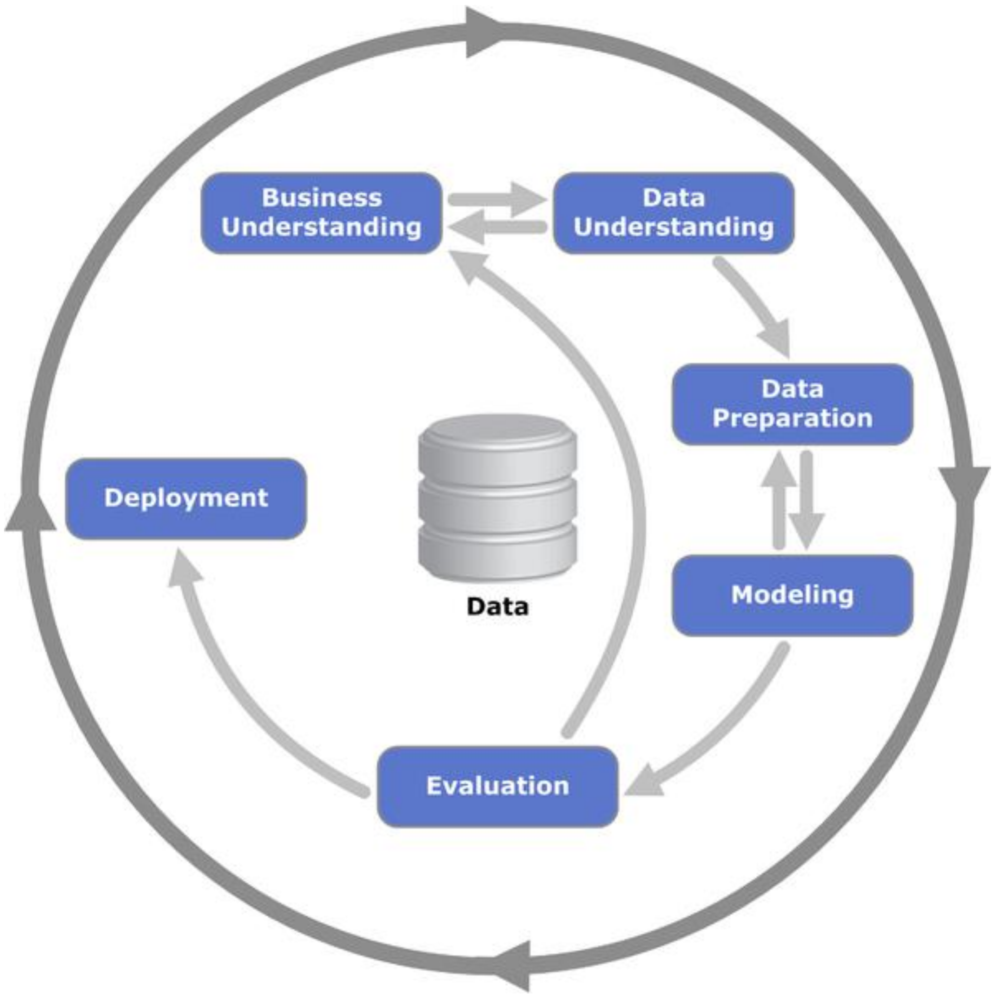
F (Black Swan)



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



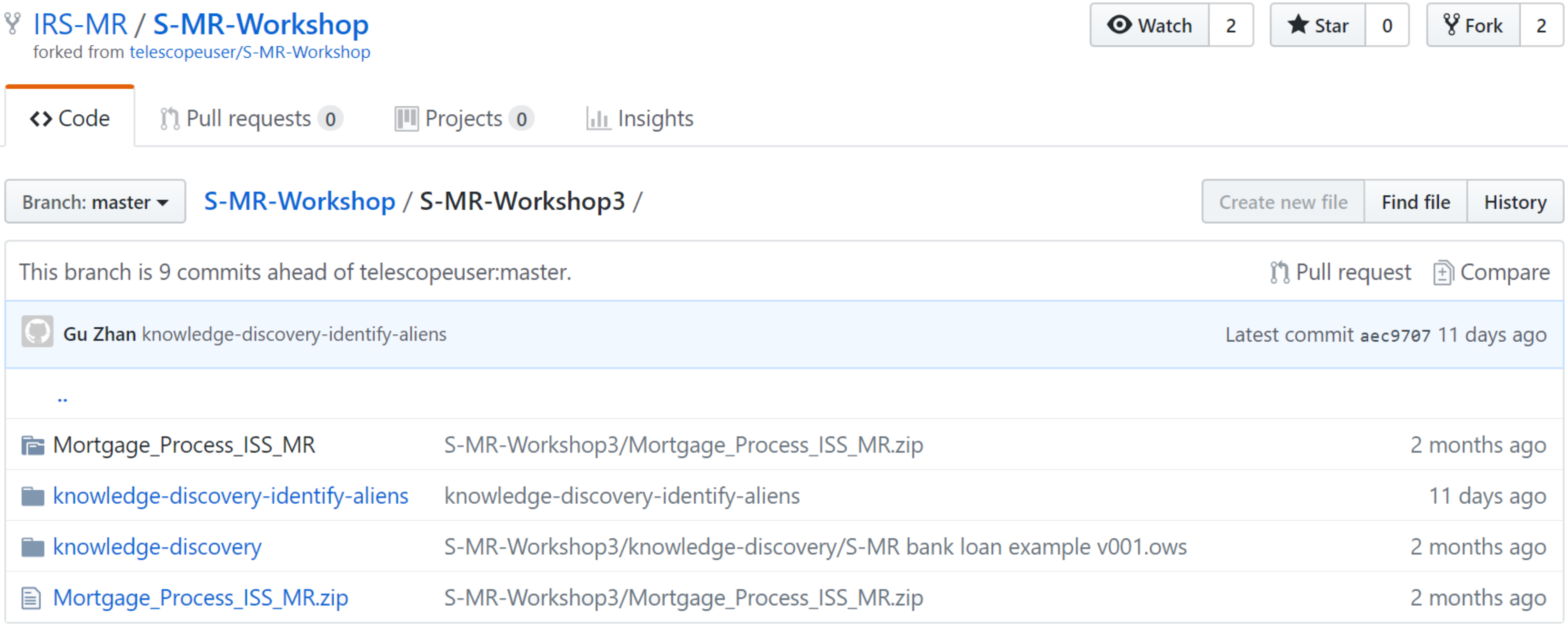
3.4 WORKSHOP KNOWLEDGE DISCOVERY

3.4 WORKSHOP KNOWLEDGE DISCOVERY
MTech Thru-Train

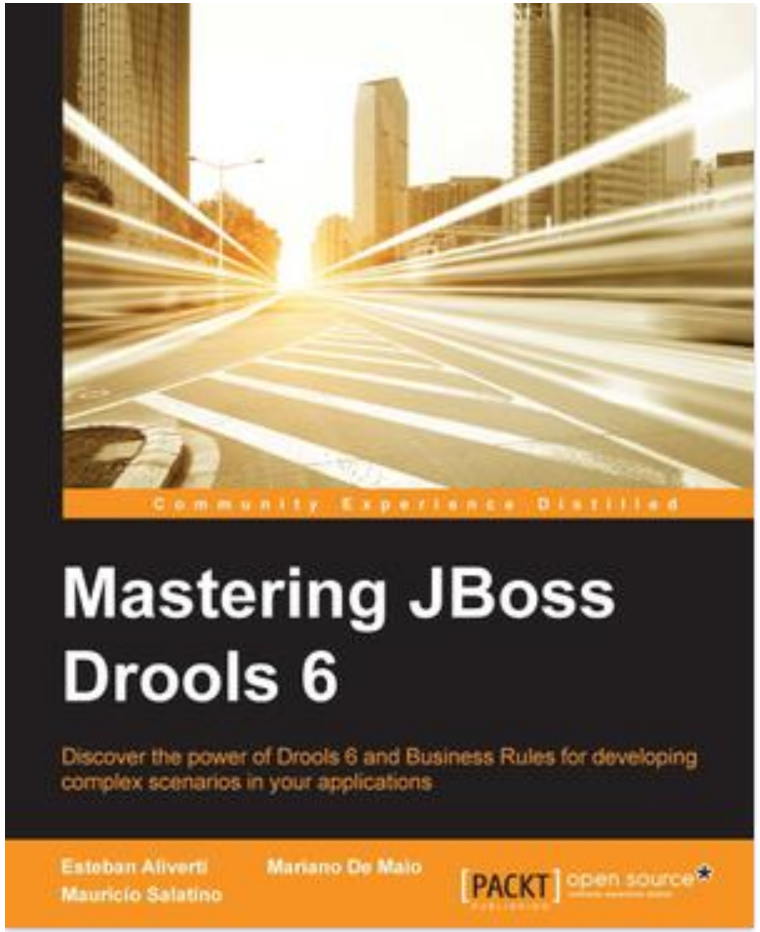
- **Knowledge Discovery – Individual Work**
  - Extract business rule from data using inductive reasoning, e.g. bank loan example
  - Enhance KIE home loan system using the discovered knowledge
  - Export enhanced KIE system and prepare for individual submission
- **KIE Development – Group Work**
  - Form a project team of 4-6 members, choose a team name, appoint a team leader.
  - Discuss within team each individual's business question & knowledge models derived from Day 2 workshop: **Knowledge Representation and Acquisition – Individual Work**
  - Select one business question/problem and extend the scope for group project
  - If there is an domain expert team member, conduct an interview with him/her
  - Extend knowledge models; Create business use/test case scenarios; Design system
  - Follow SDLC to start developing bespoke system using KIE tools
- **Project Submission Tutorial**
  - Refer to Project Submission Template

☺ Candidate Project: HDB BTO; Airport Gate Assignment System (AGAS); DoReMi

LINK HTTPS://GITHUB.COM/IRS-MR/S-MR-WORKSHOP/TREE/MASTER/S-MR-WORKSHOP3



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