Concepts and Models of Knowledge Engineering

# Car Proposes Actions Based on the Current State

Group 5: Bobko Yuliia, Ligori Armela, Farkhondeh Mehran, Mühlbauer Thomas, Tekhov Oleksii

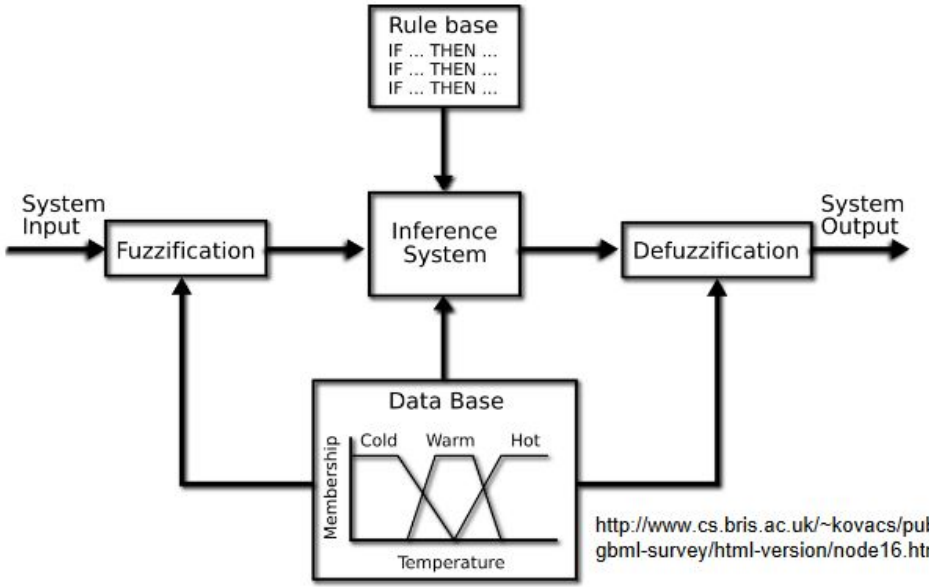
universität wien

# Problem: How to choose car actions?

- Multiple variables to consider
- Which action is more important?
- Client’s input to consider
- Regular logic cannot properly handle ambiguity

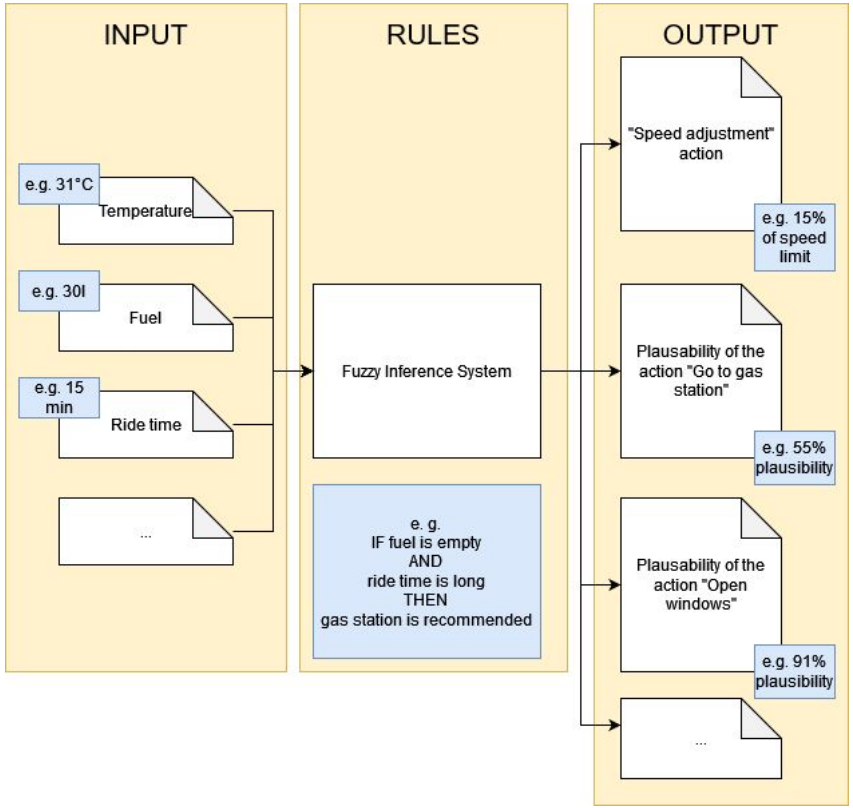
# Solution: Fuzzy Inference System

- Input: environmental values
- Output: proposed actions
- Rules that map input to output




universität wien

# Overview of the proposed system




universität wien

# Input variables

| Variable name | Description |
|---|---|
| **Temperature** | Current temperature in Celsius |
| **Ride time** | Length of the current ride in minutes |
| **Fuel level** | Current quantity of fuel in liters |
| **Traffic congestion** | Mean average speed of the selected route in km/h |
| **Car's maintenance history** | Condition of the car presented as a range (very poorly maintained – very well maintained) |
| **Points of interest (POI)** | Number of POI in the area presented as a range (scarce – abundant) |
| **Visibility** | Clarity of sight calculated by car's sensors |
| **Daytime** | Time of the day in hours |

# Output variables

| Variable name | Description |
|---|---|
| **Gas station** | Refuel the car and provide intensive to the rider |
| **Open windows** | Open the windows based on POI and visibility |
| **Supply customer with a snack and water** | Provide snacks and water based on ride time and temperature |
| **Adjust speed** | Adjust speed of the car based on visibility and traffic congestion |
| **Go to the maintenance** | Fix the car based on ride time and maintenance history |
| **Recommend a stop** | Advise stop if area has POIs |
| **Headlight intensity** | Adjust headlight intensity based on visibility and daytime |
| **Recommend car-sharing** | Recommend  to split ride with other rides based on daytime and traffic |

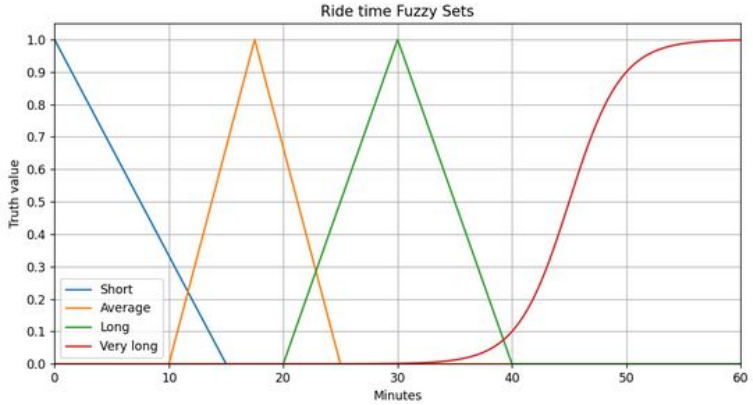
# Example input variable

- Ranges = Fuzzy Sets
- Describe variable with human language: “short ride time”
- Overlapping sets provide ambiguity

**Ride time**

Total time of the trip, in minutes.

| Linguistic Values | Ranges | Description |
|---|---|---|
| **Short** | 0-15 | minutes |
| **Average** | 10-30 | |
| **Long** | 20-40 | |
| **Very Long** | 30-60 | |




universität wien

# Example output variable (recommendation)

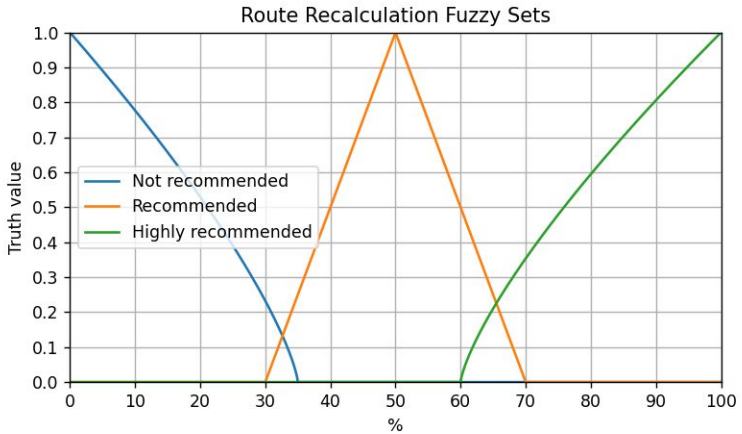
**Route recalculation**

| Linguistic Values | Ranges | Description |
|---|---|---|
| **Not Recommended** | 0-35 | |
| **Recommended** | 30-70 | |
| **Highly Recommended** | 60-100 | |

**Rules:**

| Ride time/ Traffic congestion | Uncongested | Average | Highly congested |
|---|---|---|---|
| **Short** | No | No | Rec |
| **Average** | No | No | Highly |
| **Long** | No | Rec | Highly |
| **Very Long** | No | Rec | Highly |

- Same idea as input variables
- Rules based on two inputs
- Output is the plausibility of action
- Can be compared to other actions




universität wien

# Example output variable (value)

**Speed adjustment**

Determining the appropriate speed category, assuming real-time data on traffic congestion and weather conditions that affect visibility. The legal speed limit for urban areas in Vienna is 50 km/h.

| Linguistic Values | Ranges | Description |
|---|---|---|
| **Very slow** | 0-25% of the legal speed limit | Significantly below the usual speed, prioritizes safety |
| **Slow** | 20-55% of the legal speed limit | Below the average speed |
| **Moderate** | 50 - 85% of the legal speed limit | Around the average speed for the given area |
| **Fast** | 80 - 100% of the legal speed limit | Close to the maximum legal speed for the current area |

- Different interpretation of the output value
- Output is specific to the action
- Cannot be compared to other actions

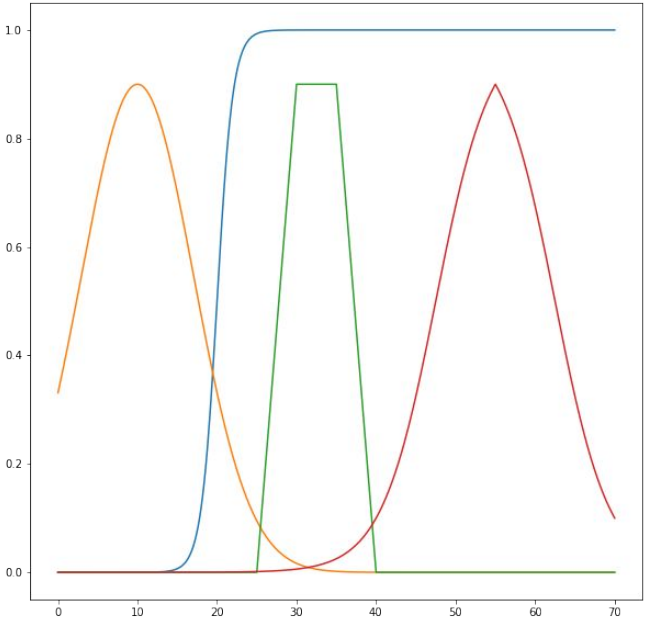
**Rules:**

| Traffic Congestion/ Visibility | Clear | Hazy | Drizzle | Rainy | Foggy |
|---|---|---|---|---|---|
| **Uncongested** | Fast | Fast | Moderate | Moderate | Slow |
| **Average** | Moderate | Moderate | Moderate | Slow | Very slow |
| **Highly Congested** | Slow | Slow | Slow | Very slow | Very slow |

# Fuzzy Logic for Python

- Handles fuzzification
- Handles defuzzification using the COG (center of gravity) method
- Provides functions for fuzzy sets: linear, triangular, trapezoid, sigmoid…
- Provides hedges (very) and combinators (and, or)

```
# define the domain for Ride time
ride_time = Domain(DOMAIN_NAME, X_FROM, X_TO, res=0.1)
# define sets
ride_time.short = S(0,15)
ride_time.average = triangular(10,25)
ride_time.long = triangular(20,40)
ride_time.very = bounded_sigmoid(40, 50) # range is 30-60
```



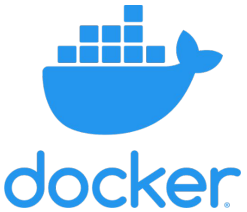

https://github.com/Python-Fuzzylogic/fuzzylogic/blob/master/docs/Showcase.ipynb

universität wien

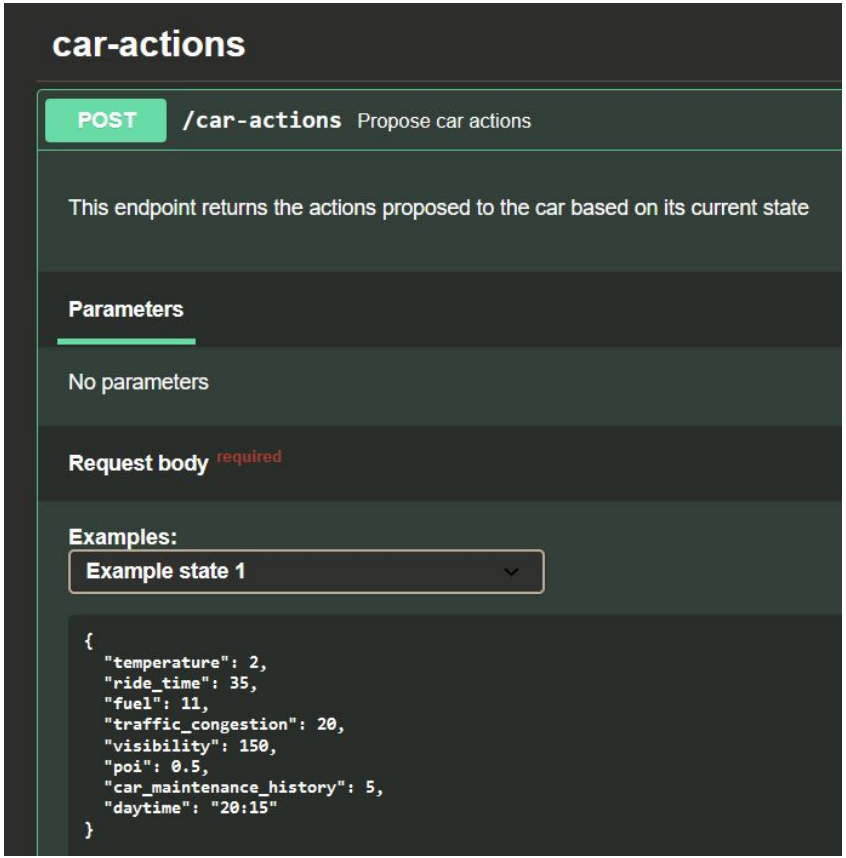
# Tools and technologies

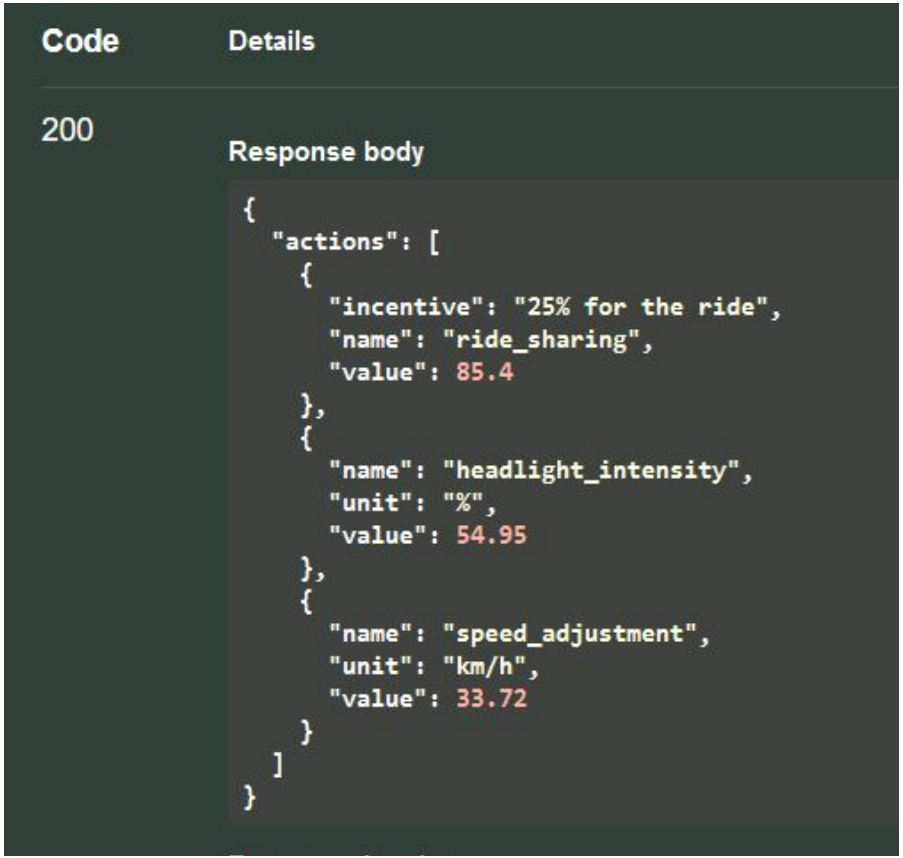
- Docker
- Python
- Flask Framework
- Swagger API
- fuzzylogic
- matplotlib






# Swagger API Endpoint



# DEMO

# Evaluation of the solution

**What worked well:**

- High ambiguity
- Meaningful output (including incentives)
- Easy to use
- Easy to modify (fine-tune variables and rules)

**What can be improved:**

- More fuzzy sets for output variables
- More complexity for the sets' functions
- Simulations can be done to analyse the results

Thank you!

Questions?