

In [8]: *# importing Libraries*

```
from skfuzzy import control as ctrl
import skfuzzy as fuzz
import numpy as np
```

In [20]: **class** washing_machine:

Defining Antecedents and Consequent:

degree_dirt: Represents the degree of dirtiness of the laundry.

type_dirt: Represents the type of dirt in the laundry.

wash_time: Represents the wash time required.

```
degree_dirt = ctrl.Antecedent(np.arange(0, 101, 1), 'degree_dirt')
type_dirt = ctrl.Antecedent(np.arange(0, 101, 1), 'type_dirt')
wash_time = ctrl.Consequent(np.arange(0, 61, 1), 'wash_time')
```

Linguistic Term Definitions:

degree_names: Linguistic terms for the degree of dirtiness (e.g., 'Low', 'Medium'

type_names: Linguistic terms for the type of dirt (e.g., 'NonFat', 'Medium', 'Fat'

```
degree_names = ['Low', 'Medium', 'High']
type_names = ['NonFat', 'Medium', 'Fat']
```

#Outputing them into auto-membership functions

```
degree_dirt.automf(names=degree_names)
type_dirt.automf(names=type_names)
```

Washing Time Universe

```
wash_time['very_short'] = fuzz.trimf(wash_time.universe, [0, 8, 12])
wash_time['short'] = fuzz.trimf(wash_time.universe, [8, 12, 20])
wash_time['medium'] = fuzz.trimf(wash_time.universe, [12, 20, 40])
wash_time['long'] = fuzz.trimf(wash_time.universe, [20, 40, 60])
wash_time['VeryLong'] = fuzz.trimf(wash_time.universe, [40, 60, 60])
```

Rule Application

A set of fuzzy rules (rule1 to rule9) that map the linguistic terms of degree

```
rule1 = ctrl.Rule(degree_dirt['High'] | type_dirt['Fat'], wash_time['VeryLong'])
rule2 = ctrl.Rule(degree_dirt['Medium'] | type_dirt['Fat'], wash_time['long'])
rule3 = ctrl.Rule(degree_dirt['Low'] | type_dirt['Fat'], wash_time['long'])
rule4 = ctrl.Rule(degree_dirt['High'] | type_dirt['Medium'], wash_time['long'])
rule5 = ctrl.Rule(degree_dirt['Medium'] | type_dirt['Medium'], wash_time['mediu
rule6 = ctrl.Rule(degree_dirt['Low'] | type_dirt['Medium'], wash_time['medium'])
rule7 = ctrl.Rule(degree_dirt['High'] | type_dirt['NonFat'], wash_time['medium'])
rule8 = ctrl.Rule(degree_dirt['Medium'] | type_dirt['NonFat'], wash_time['short
rule9 = ctrl.Rule(degree_dirt['Low'] | type_dirt['NonFat'], wash_time['very_sho
```

Washing Control Simulation

washing_ctrl: The control system that includes all the defined rules.

washing: The control system simulation object that allows you to set input va

```

washing_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6, ru
washing = ctrl.ControlSystemSimulation(washing_ctrl)

def fuzzify_laundry(fuzz_type,fuzz_degree):

    washing_machine.washing.input['type_dirt'] = fuzz_type
    washing_machine.washing.input['degree_dirt'] = fuzz_degree

    washing_machine.washing.compute()

    washing_machine.wash_time.view(sim=washing_machine.washing)

    return washing_machine.washing.output['wash_time']

```

```

In [19]: def compute_washing_parameters(type_of_dirt,degree_of_dirt):

    if type_of_dirt < 0.0 or type_of_dirt > 100.0:
        raise Exception("Invalid Type of Dirtiness: %lf" %type_of_dirt)
    if degree_of_dirt < 0.0 or type_of_dirt > 100.0:
        raise Exception("Invalid Degree of Dirtiness: %lf" %degree_of_dirt)

    type_fuzzy = fuzzify_laundry(type_of_dirt,degree_of_dirt)

    return type_fuzzy

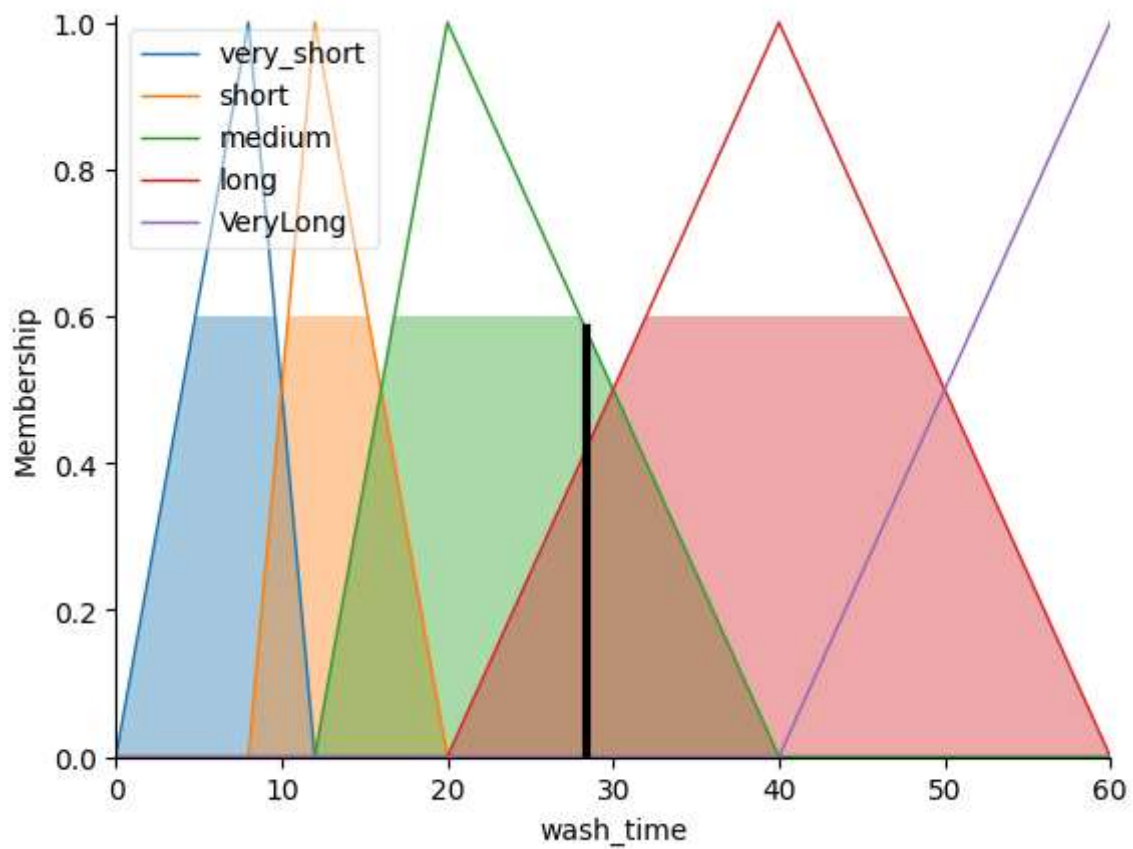
```

```

In [21]: if __name__ == "__main__":
    type_of_dirt = float(input("Enter Type of Dirtiness [0-100]"))
    degree_of_dirt = float(input("Enter Degree of Dirtiness [0-100]"))
    washing_parameters = compute_washing_parameters(type_of_dirt,degree_of_dirt)
    print(washing_parameters)

```

28.342558746736277



In []: