Exercise 4.1: Fuzzy Inference (5 Points)
Consider the membership functions for the linguistic terms of the linguistic variable mark. Notice that outside the given range their values are zero!

\[
\begin{align*}
\text{very_good}(x) &= \exp(-2x^2) \text{ for } x \geq 1 \\
good(x) &= -(x-1)(x-3) \text{ for } x \in (1, 3) \\
\text{fair}(x) &= -(x-2)(x-4) \text{ for } x \in (2, 4) \\
\text{bad}(x) &= \min\{1, \frac{1}{2}(x-3)\} \text{ for } x > 3
\end{align*}
\]

Below you can find the membership functions for the linguistic terms of the linguistic variable learning_time. Again, outside the given range their values are zero!

\[
\begin{align*}
\text{huge}(x) &= \min\{x-5, 1\} \text{ for } x \geq 5 \\
\text{big}(x) &= -\frac{4}{9}(x-3)(x-6) \text{ for } x \in (3, 6) \\
\text{low}(x) &= \min\{2-x, 1\} \text{ for } x < 2
\end{align*}
\]
Based on the fuzzy proposition

$$\text{if learning\_time is big then mark is good},$$

the Łukaciewicz implication $\text{Imp}(a, b) = \min\{1, 1 - a + b\}$ and the max-prod composition deduce the resulting fuzzy set over learning time for the given fuzzy fact

$$\text{mark is fair}.$$

Sketch the membership function. Hint: Discretize the function and use a table of values.

Exercise 4.2: Fuzzy Implication (5 Points)

a) Use the increasing generator $g(x) = \sqrt{x}$ to derive a fuzzy implication. Does the resulting implication fulfill the axiom of contraposition?

b) Check for all fuzzy implications below if they fulfill the axiom of contraposition:

- Reichenbach $\text{Imp}(a, b) = 1 - a + a b$
- Łukaciewicz $\text{Imp}(a, b) = \min\{1, 1 - a + b\}$
- Gödel $\text{Imp}(a, b) = \begin{cases} 1 & a \leq b \\ b & \text{otherwise} \end{cases}$

Exercise 4.3: Fuzzy Controller (10 Points)

Implement in R a Mamdani controller for the speed of a train. The train shall drive a given distance as precisely and fast as possible (assume, e.g., that the track ends in a dead-end railway station).

The train shall be “simulated” by the following simple function that takes the current speed and the controller’s output for the applied driving or braking force (for simplicity we combine throttle and brake in one controller). The function returns the speed of the train one second later in the simulation.
trainSpeed <- function(currentSpeed, drivingOrBrakingForce) {
    drivingOrBrakingForce <- max(min(drivingOrBrakingForce, 25), -200)
    trainMass <- 120 * 1000
    rollingDrag <- 20 * 0.001 * 9.81
    airDrag <- 1.3 / 2 * 1.5 * 14 * currentSpeed ** 2
    dragForce <- rollingDrag + airDrag
    propellingForce <- drivingOrBrakingForce * 1000 - dragForce
    acceleration <- propellingForce / trainMass
    newSpeed <- max(0, currentSpeed + acceleration)
    return(newSpeed)
}

The range for the driving/braking force shall be \([-200, 25]\) (negative values mean braking). In combination with the other parameters, this means that the maximum speed of the train is 42.8 m/s (≈ 154 km/h) and the braking distance is 697 m. The traveled distance in meter can be obtained by simply summing the returned speeds, because the unit is m/s and we use time steps of one second in the simulation.

a) Model the membership functions for the linguistic variables speed, remaining distance, and driving/braking force.
b) Setup an appropriate fuzzy rule system for a Mamdani controller.
c) Implement the control loop for the Mamdani controller using the center of gravity method for defuzzification.
d) Give a short documentation of your source code.
e) Start the train for a desired travel distance of 1000 m and 10 km with initial speeds of 0, 20, and 42.8 m/s. Report for each of the six settings the traveled distance and the required time. Also plot the train’s speed over time.