

---

# Laboratorium

## Kontroler rozmyty

---

### Zadanie 1

Użyjemy paczki `simpful`, by stworzyć prosty kontroler rozmyty do obliczania napiwków (przykład był na wykładzie).

a) Poczytaj o paczce `simpful` np. tutaj:

- <https://pypi.org/project/simpful/>
- [https://pythonhosted.org/scikit-fuzzy/auto\\_examples/plot\\_tipping\\_problem\\_newapi.html](https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem_newapi.html)
- [https://www.researchgate.net/publication/346395808\\_Simpful\\_A\\_User-Friendly\\_Python\\_Library\\_for\\_Fuzzy\\_Logic](https://www.researchgate.net/publication/346395808_Simpful_A_User-Friendly_Python_Library_for_Fuzzy_Logic)

Zwróć uwagę na wstawki kodu i sposób tworzenia kontrolerów (zmienne, reguły, wyostrzanie).

- b) Zainstaluj paczkę i skopiuj z wybranej strony kod tworzący system do dawania napiwków (3 zmienne lingwistyczne, 3 reguły).
- c) Wyświetl wykresy zmiennych lingwistycznych.
- d) Przetestuj działanie kontrolera. Daj kilka danych (liczby dla jedzenia i obsługi) i wyświetl jaki napiwek (0-30%) proponuje system dla tych inputów.

### Zadanie 2

Wykorzystaj paczkę `simpful` do stworzenia kontrolera rozmytego dla wybranej gry z Gym. Preferowane gry, gdzie sterujemy jakimś prostym robotem np.

- LunarLander, albo:
- Bipedal Walker, albo:
- Pendulum (lub inne)

Rozwiązanie powinno się składać z następujących kroków:

- a) Zdefiniowanie sensownych zmiennych lingwistycznych dla każdej zmiennej. Wyświetl wykresy.
- b) Zdefiniowanie zestawu sensownych reguł wnioskowania rozmytego. Zastanów się czy lepszym operatorem będą „or” czy „and”.
- c) Odpalenie systemu i obliczanie wyostrzonych zmiennych sterujących (consequents). Zmienne powinny być obliczane w każdej klatce symulacji. W każdej klatce animacji fuzzy controller oblicza co trzeba zrobić i wykonuje tę akcję.

Postaraj się, by system działał w miarę dobrze (nie musi działać idealnie). W razie czego zmodyfikuj zmienne lingwistyczne lub reguły rozmyte.