

Jak wiemy z logiki, formuły rachunku zdań możemy budować ze zmiennych zdaniowych i spójników logicznych.

Twoim zadaniem będzie napisanie funkcji `evaluate(formula, sigma)`, która sprawdzi wartość formuły `formula` (string) w wartościowaniu `sigma` (słownik).

W tym zadaniu wykorzystasz wbudowaną funkcję `eval`, przeczytaj więc jej dokumentację.

Wartościowanie `sigma` możemy reprezentować jako słownik, np.

```
sigma = {  
  'p': True,  
  'q': False  
}
```

Interesować nas będą jedynie formuły zbudowane ze zmiennych zdaniowych (pojedyncza litera) oraz spójników: negacja `~`, koniunkcja `&` i alternatywa `|` (pojedynczy znak). (Przykładowa formuła: $(p \mid q) \& (\sim p \mid \sim q)$)

Do każdego z podpunktów przygotuj kilka przykładów użycia, pokazujących, że twoje rozwiązanie jest poprawne.

Punkty sprawdzane będą po kolei, żeby dostać punkty za następny należy zrobić wszystkie poprzednie!

Happy Path (0.5p)

Z pewnych względów, w tym wariancie interesować nas będą jedynie formuły zbudowane ze zmiennych zdaniowych `p` i `q`. Będziemy też zakładać, że `sigma` jest zawsze poprawna, i zawiera zarówno `p` jak i `q`. Napisz funkcję `evaluate(formula, sigma)` która zwraca `True` jeśli w wartościowaniu `sigma` formuła `formula` jest spełniona, i `False` w przeciwnym przypadku.

Wskazówka (skopiuj i wklej!):

```
def evaluate(formula, sigma):  
    # Twój kod tutaj
```

Sprawdzanie poprawności sigmy (0.2p)

Jeśli `sigma` nie zawiera wszystkich zmiennych które są w formule, to z pewnością nie jest sigmą poprawną. Napisz funkcję, która sprawdzi, czy `sigma` jest poprawna (zakładając, że opisany warunek jest warunkiem wystarczającym). W tym celu możesz założyć, że zmienna zdaniowa to dowolny alfanumeryczny znak.

Użyj nowo napisanej funkcji w funkcji `evaluate`, w celu wyrzucenia wyjątku `Zła Sigma` jeśli `sigma` jest niepoprawna dla zadanej formuły.

Dowolność zmiennych zdaniowych (0.5p)

Jeśli w napisie $(p \mid r)$ zamienimy wszystkie wystąpienia zmiennej `p` na `True`, a potem wszystkie wystąpienia `r` na `False`, to otrzymamy napis $(True \mid False)$. Wszystkie litery które wchodzi w skład napisów `True` i `False` są więc zabronione. Zaproponuj rozwiązanie tego problemu tak, by można było używać wszystkich liter alfabetu.

Sprawdzanie formuły (0.6p)

Przypomnijmy, że formuła jest **tautologią** jeśli jest spełniona we wszystkich wartościowaniach, oraz **sprzeczną** jeśli nie jest spełniona w żadnym z wartościowań.

Napisz funkcję, która sprawdzi, czy formuła jest tautologią albo formułą sprzeczną. Jeśli jest tautologią, funkcja powinna wypisać `Tautologia`, jeśli jest sprzeczną, funkcja powinna wypisać `Sprzeczna`.

Uwaga: Zakładamy niepustość i poprawność formuły.

PS. W tym zadaniu **możesz** używać pomocniczych funkcji takich jak `product` z `itertools` oraz `zip` czy `dict`. Możesz też zrobić to bez użycia innych funkcji.

Wskazówka (skopiuj i wklej!):

```
def is_satisfiable(formula):
    # ...
    return True
```

Spełnialność formuł (0.2p)

Jeśli formuła nie jest formułą sprzeczną ani tautologią, to jest formułą spełnialną. Rozszerz poprzednią funkcję i zwróć `Spełnialna`, i wypisz jedno wartościowanie spełniające formułę i jedno które jej nie spełnia.