

## Lab1. Wyrażenie logiczne

Poprawne wyrażenie logiczne (wersja podstawowa) może zawierać:

- zmienne logiczne oznaczane literami **a-z**,
- operatory: koniunkcji '&', alternatywy '|' i implikacji '>'
- nawiasy: ( )

Wersja rozszerzona może dodatkowo wyrażenie zawierać:

- stałe oznaczające logiczną prawdę i fałsz oznaczone literami **T** , **F**
- operatory: negacji '~', różnicy symetrycznej (xor) '^'

Priorytety operatorów logicznych na potrzeby zadania przedstawia poniższa tabela.

Operator	Priorytet
~	4
^	3
&	2
>	1

1. Proszę napisać funkcję `check(expr)`, sprawdzającą poprawność wyrażenia logicznego.  
Na przykład:  
`'a>b>c&a' -> True`  
`'(a>(b&c)' -> False`
2. Proszę napisać funkcję `bracket(expr)` usuwającą w wyrażeniu zewnętrzne nawiasy.  
Na przykład:  
`'(a|b)' -> 'a|b'`  
`'((a>b)&(c>d))' -> '(a>b)&(c>d)'`  
`'(((a)))' -> 'a'`
3. Proszę napisać funkcję `bal(exp)`, zwracającą najbardziej prawą pozycję danego operatora niezagnieżdżonego w nawiasach.  
Na przykład:  
`bal('a>(b>c)', '>') -> 1`  
`bal('a|b&c', '&') -> 3`  
`bal('a|(b&c)', '&') -> None`
4. Proszę napisać funkcję `onp(expr)` zamieniającą wyrażenie z postaci algebraicznej (infiksowej) na postać ONP (postfiksową).  
Na przykład:  
`'a&b' -> 'ab&'`  
`'a>b>c' -> 'ab>c>'`  
`'(a>(b|c))' -> 'abc|>'`
5. Proszę napisać funkcję `map(expr, vec)` mapującą zmienne w wyrażeniu ONP na wartości z wektora `vec`.  
Na przykład:  
`map('ab&c|', '101') -> '10&1|'`  
`map('ac>b>', '101') -> '11>0>|'`
6. Napisać funkcję `gen(n)`, generującą wszystkie ciągi zero-jedynkowe o długości `n`.
7. Napisać funkcję `val(expr)` obliczającą wartość wyrażenia logicznego `expr`.  
Na przykład:  
`val('10>') -> 0`  
`val('101>&') -> 1`
8. Napisać funkcję `tautology(expr)`, sprawdzającą czy wyrażenie `expr` jest tautologią.

### **Zadania do samodzielnego wykonania**

9. Proszę napisać funkcję generującą przypadkowe wyrażenia logiczne, a następnie przypadkowe tautologie.
10. Proszę napisać funkcję  $\text{alg}(\text{expr})$  zamieniającą wyrażenie z postaci ONP na postać algebraiczną
11. Proszę napisać funkcję sprawdzającą tożsamość dwóch wyrażeń logicznych.
12. Proszę napisać funkcję redukującą długość wyrażenia logicznego.