

# **TEORIA WSPÓŁBIEŻNOŚCI**

## **LABORATORIUM 5**

Zadanie domowe FNF

Bartosz Konieczny

Wydział Informatyki  
Grupa 5  
Wtorek 8:00

data oddania:  
10.11.2025





# Ćwiczenie

Napisz program, który:

1. Wyznacza relację zależności  $D$
2. Wyznacza relację niezależności  $I$
3. Wyznacza postać normalną Foaty FNF([ $w$ ]) śladu  $[w]$
4. Rysuje graf zależności w postaci minimalnej dla słowa  $w$

## Instrukcja uruchomienia

W głównym katalogu projektu należy wpisać:

```
pip install .
```

Zostaną wtedy zainstalowane niezbędne biblioteki, dlatego zalecane jest wykonanie tej czynności po aktywacji wirtualnego środowiska.

Aby uruchomić program dla pliku o nazwie `file_name` znajdującego się w katalogu `data` w projekcie należy wpisać:

```
startapp file_name
```

Domyślnie użyty zostanie plik `example1.txt` po wpisaniu komendy `startapp`.

## Format pliku wejściowego

Program przyjmuje pliki w formacie:

```
(<litera>) <akcja>
...
(<litera>) <akcja>
A = {<alfabet>}
w = <słowo>
```

Gdzie:

- zmienna - pojedyncza litera
- akcja - wyrażenie w postaci `zmienna := <wyrażenie składające się ze zmiennych, cyfr oraz podstawowych operacji matematycznych zapisanych w poprawny sposób>`

Plik powinien znajdować się w folderze `data`, natomiast plik wynikowy będzie znajdował się w folderze `results` o nazwie, a jego nazwa to `res_filename`, gdzie `filename` to plik wejściowy.

Przykład poprawnego pliku wejściowego:

```
(a) x := x + 1
(b) y := y + 2z
(c) x := 3x + z
(d) w := w + v
(e) z := y - z
(f) v := x + v
A = {a, b, c, d, e, f}
w = acdcfbbe
```



## Opis programu

Program został napisany w języku Python. Do programu zostały dostarczone testy. Użyte biblioteki to: NumPy, Matplotlib, NetworkX oraz pytest.

Program składa się z 4 plików:

- `app.py` - główna część programu odpowiedzialna za wczytanie danych i zwrócenie wyniku,
- `relations.py` - część odpowiedzialna za wyznaczenie relacji zależności  $D$  i niezależności  $I$ ,
- `dependency_graph.py` - część odpowiedzialna za stworzenie i narysowanie grafu zależności,
- `fnf.py` - część odpowiedzialna za wyznaczenie FNF na podstawie grafu zależności.

## Wynik działania dla przykładowych danych

### 5.1 example1

Dla danego pliku `example1.txt`:

```
(a) x := x + y  
(b) y := y + 2z  
(c) x := 3x + z  
(d) z := y - z  
A = {a, b, c, d}  
w = baadcb
```

Otrzymamy plik `res_example1.txt`:

```
D = {(a, a),(a, c),(a, b),(b, a),(b, d),(b, b),(c, a),(c, d),(c, c),(d, d),(d, c),(d, b)}  
I = {(a, d),(d, a),(b, c),(c, b)}  
FNF([w]) = (b)(ad)(a)(bc)  
digraph g{  
    1 -> 2  
    1 -> 4  
    2 -> 3  
    3 -> 5  
    3 -> 6  
    4 -> 5  
    4 -> 6  
    1[label=b]  
    2[label=a]  
    3[label=a]  
    4[label=d]  
    5[label=c]  
    6[label=b]  
}
```



Graph for word "baadcb"



Rys. 1: Graf otrzymany dla przykładu 1 oraz słowa „baadcb”.

## 5.2 example2

Dla danego pliku example2.txt:

```
(a) x := x + 1
(b) y := y + 2z
(c) x := 3x + z
(d) w := w + v
(e) z := y - z
(f) v := x + v
A = {a, b, c, d, e, f}
w = acdcfbbe
```

Otrzymamy plik res\_example2.txt:

```
D = {(a, f), (a, a), (a, c), (c, f), (c, a), (c, c), (c, e), (f, f), (f, d), (f, a), (f, c), (b, b),
(b, e), (e, b), (e, e), (e, c), (d, f), (d, d)}
I = {(a, b), (a, d), (a, e), (b, f), (b, d), (b, a), (b, c), (d, b), (d, a), (d, c), (d, e), (e, f),
(e, d), (e, a), (c, b), (c, d), (f, b), (f, e)}
FNF([w]) = (abd)(bc)(c)(ef)
digraph g{
1 -> 2
2 -> 4
3 -> 5
4 -> 5
4 -> 8
6 -> 7
7 -> 8
1[label=a]
2[label=c]
3[label=d]
4[label=c]
5[label=f]
6[label=b]
7[label=b]
8[label=e]
}
```



Graph for word "acdcfbbe"



Rys. 2: Graf otrzymany dla przykładu 2 oraz słowa „acdcfbbe”.

### 5.3 example3

Dla danego pliku example3.txt:

```
(a) x := x + y
(b) y := z - v
(c) z := v * x
(d) v := x + 2y
(e) x := 3y + 2x
(f) v := v - 2z
A = {a,b,c,d,e,f}
w = afaeffbcd
```

Otrzymamy plik res\_example3.txt:

```
D = {(a, b), (a, e), (a, c), (a, a), (a, d), (b, e), (b, f), (b, b), (b, c), (b, a), (b, d), (c, b),
(c, e), (c, f), (c, c), (c, a), (c, d), (d, d), (d, b), (d, e), (d, c), (d, a), (d, f), (e, b), (e,
e), (e, c), (e, a), (e, d), (f, c), (f, d), (f, b), (f, f)}
I = {(a, f), (f, e), (f, a), (e, f)}
FNF([w]) = (af)(af)(ef)(b)(c)(d)
digraph g{
1 -> 3
2 -> 5
3 -> 4
4 -> 7
5 -> 6
6 -> 7
7 -> 8
8 -> 9
1[label=a]
2[label=f]
3[label=a]
4[label=e]
5[label=f]
6[label=f]
7[label=b]
8[label=c]
9[label=d]
}
```



Graph for word "afaeffbcd"



Rys. 3: Graf otrzymany dla przykładu 3 oraz słowa „afaeffbcd”.