# Transformacje i algorytmy grafowe – projekt 2020/2021

Autorzy:

Ernest Roszak

Filip Piwosz

#### Informacje wstępne

Wynikiem projektu jest program, umożliwiający wykonanie transformacji grafowych (produkcji wraz z transformacjami osadzenia) na grafie zapisanym w formacie dot. Program posiada prosty interfejs graficzny, oraz umożliwia wyświetlenie podstawowych statystyk dotyczących transformowanego grafu.

Program został napisany w języku python. Do jego uruchomienia wymagane są następujące biblioteki: graphviz, pydotplus, tkinter, PIL. Opcjonalnie program można otworzyć w środowisku Visual Studio Code, w którym pracowaliśmy, ale program uruchamia się również z konsoli.

Wykonywać można tylko produkcje z <u>jednym</u> wierzchołkiem lewej strony produkcji. Wyszukiwany jest tylko jeden wierzchołek lewej strony produkcji zgodny z etykietą i dla niego wykonywana jest produkcja. Jeżeli nie znaleziono wierzchołka lewej strony produkcji, produkcja jest pomijana i wykonuje się kolejna.

## Graf wejściowy

Graf wejściowy musi być zapisany w postaci pliku .dot. Oprócz tego każdy wierzchołek musi mieć swoje imię 'name' w postaci liczby, jak również posiadać atrybut "label", będący etykietą wierzchołka. Krawędzie są definiowane w standardowy sposób.

```
graph Start
{
    0 [label=A];
    1 [label=B];
    2 [label=C];
    0--1--2;
    2--0;
}
```

Przykład grafu wejściowego

### Wczytywanie produkcji

Produkcje wczytywane są z pliku .txt w następującym formacie:

```
Id_produkcji

#
Wierzchołek lewej strony produkcji

#
Graf prawej strony produkcji w takim samym formacie jak graf wejściowy

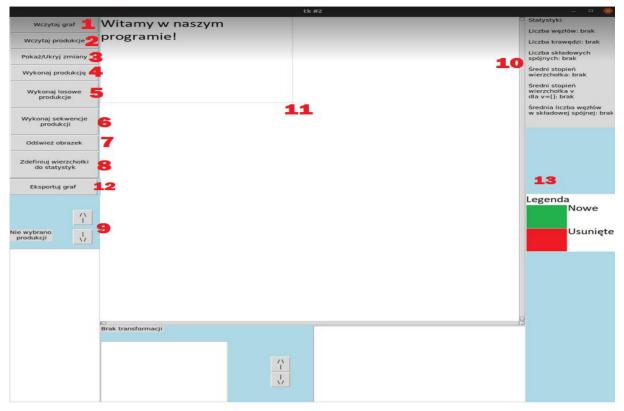
#
transformacje osadzenie w następującym formacie:
wierzchołek lewej strony [odstęp] wierzchołek prawej strony, gdzie "—,,
oznacza brak wierzchołka
...
```

# < -- obowiązkowy znak końca produkcji

```
1 1
2 #
3 A
4 #
5 graph RHS_1 {
6 1 [label=A];
7 2 [label=A];
8 3 [label=A];
9 1 -- 2;
10 2 -- 3;
11 3 -- 1;
12 }
13 #
14 A A
15 B -
16 C -
17 a A
18 b -
19 c -
```

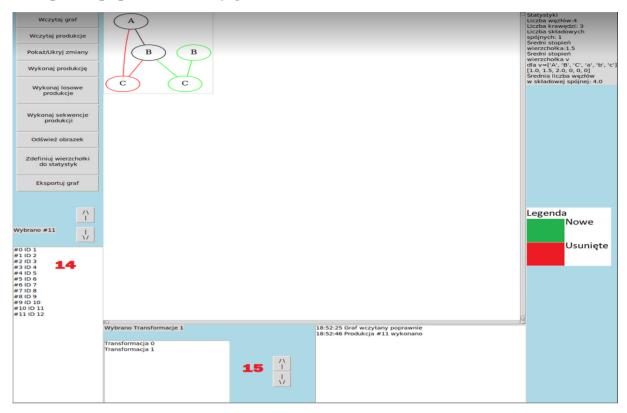
Przykładowa produkcja

### Opis interfejsu



- 1 Wczytanie grafu w postaci pliku .dot
- 2 Wczytanie produkcji w formacie .txt
- 3 Pokaż/ ukryj legendę
- 4 Wykonanie produkcji zaznaczonej poprzez przyciski 9
- 5 Wykonanie losowych produkcji
- 6 Wykonanie sekwencji produkcji
- 7 odświeżenie obrazka
- $\mathbf{8}$  Definiuje etykiety wierzchołków, dla których pokazują się szczegółowe statystyki w  $\mathbf{10}$
- 9 przyciski do zmiany aktualnie wybranej produkcji
- 10 statystyki dotyczące aktualnego grafu
- 11 ekran na którym wyświetlany jest graf
- 12 eksportuj graf do pliku .dot
- 13 legenda
- 14 lista transformacji

## 15 – pokaż poprzednie stany grafu



## Instrukcja korzystania z programu

Pierwszym krokiem jest wczytanie grafu w formacie.dot, zgodnego ze wspomnianymi wyżej specyfikacjami, przy pomocy przycisku 1. Następnie należy wczytać produkcje z pliku .txt przy pomocy przycisku 2. Możemy teraz wykonywać produkcję przy pomocy przycisków 4, 5 i 6. Aby zapisać graf do pliku .dot, użyj przycisku 12.

#### Kwestie techniczne

Program składa się z następujących plików:

main.py – główny plik z którego uruchamiamy program, zawiera również funkcje interfejsu

ProdClass.py - Klasy produkcji, oraz funkcje odpowiedzialne za produkcje

Parser.py – odpowiada za wczytywanie plików produkcji

AdditionalFunc.py – funkcje pomocnicze

graphProperties.py – funkcje obliczające właściwości grafu

DeleteCache.py - wyczyszczenie folderu Cache

Oprócz tego do poprawnego wyświetlania potrzebne są również pliki "legend.png" oraz "hello.png". Należy również stworzyć folder "cache". W folderze "default data" dane są przykładowe produkcje i graf.