

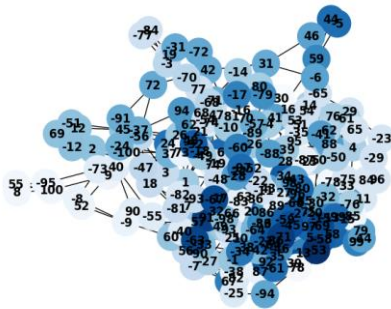
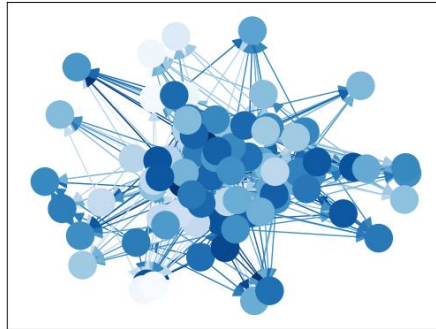
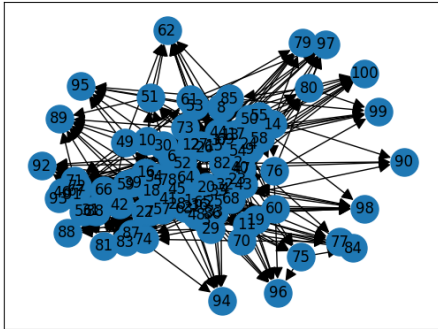
Studio Projektowe 1

Wizualizacja formuł logicznych

Jakub Banach, Karol Błaszczak

Raport z wykonanych prac na dzień 12.12.2023

## Hipergrafy



Utworzony kod dość dobrze realizuje podstawowe zagadnienie wizualizacji hipergrafów, jednakże w celu zwiększenia czytelności należy go wzbogacić o kolejne funkcje. Podstawowym pomysłem nasuwających się po zajęciach z Panem Profesorem jest wyeliminowanie węzłów o najniższej wartości (czyli o najjaśniejszym odcieniu niebieskiego) w celu zwiększenia przejrzystości bardziej istotnych danych.

Kolejne pomysły dotyczące rozbudowy funkcjonalności kodu to, np.:

1. Wizualizacja interaktywna, stworzona przy użyciu biblioteki takiej jak Plotly, pozwalająca użytkownikom interaktywnie eksplorować hipergraf, wraz z jego osadzeniami grafowymi i grafem skierowanym
2. Algorytmy detekcji społeczności, pozwalające dostarczyć wgląd w strukturę hipergrafu i sposób grupowania węzłów
3. Analiza osadzeń grafowych (np. przez połączenie obu tworzonych sposobów)
4. Modyfikacja hipergrafu w czasie rzeczywistym, pozwalająca użytkownikowi dynamicznie dodawać/usuwać klauzule/węzły w hipergrafie i wizualizować te zmiany w czasie rzeczywistym. Ta funkcjonalność może być przydatna do eksploracji wpływu modyfikacji na osadzenia węzłów i graf skierowany.
5. Wizualizacja podgrafów. To może być przydatne do skupienia się na konkretnych częściach hipergrafu.

**Z komentarzem [KB1]:** Punkt "0": zrealizowany przy użyciu dodania parametru threshold w funkcjach draw\_hypergraph oraz interactive\_hypergraph w pliku hypergraphs v3.0

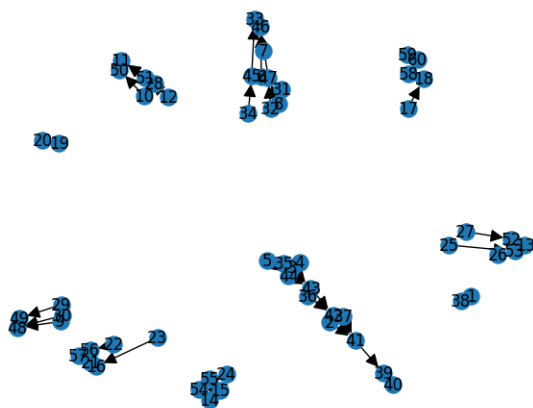
**Z komentarzem [KB2R1]:**  
`nodes_to_remove = [node for node, degree in dict(G.degree()).items() if degree <= threshold]`

`G.remove_nodes_from(nodes_to_remove)`

**Z komentarzem [KB3]:** Punkt 1: zrealizowany przy użyciu plotly.graph\_objs w funkcji interactive\_hypergraph w pliku hypergraphs v3.0

**Z komentarzem [KB4]:** Punkty 2-5 w trakcie realizacji, aktualnie planuję rozwinąć przede wszystkim punkty 4-5, wykorzystując między innymi funkcjonalności stworzone podczas realizacji punktów "0" oraz 1. Realizacja punktu 3 wykorzysta fragmenty tworzonego przez kolegę Jakuba Banacha kodu dotyczącego osadzeń grafowych

## Graph Embeddings



Kod stworzony dla graph embeddings pokazuje grupy najbardziej powiązanych ze sobą wartości (dla każdej zmiennej strzałka określa wskazuje zmienną najbardziej do niej podobną). Wykres nie jest jednak czytelny, ale aby móc uzyskać wartościowe można zacząć bazować na określonych koncepcjach:

1. Wydzielenie najmniejszej/największej grupy zależności zmiennych.
2. Wyznaczanie zmiennych które są najbardziej podobne do innych (mają najwięcej strzałek)