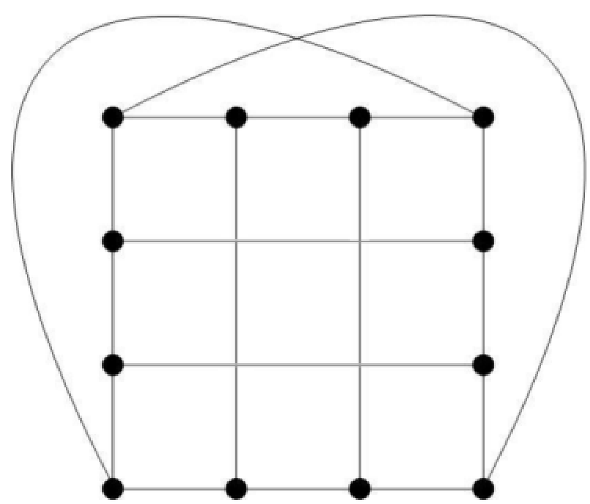
Planarność grafu

Paweł Sokołowski
Michał Kaszlej

17 grudnia 2013

1 Cel projektu

Celem projektu jest napisanie algorytmu heurystycznego sprawdzającego czy zadany graf jest planarny. Według twierdzenia podanego w książce (Wilson, 1998):



Rysunek 1: Zadany graf

Dany graf jest planarny wtedy i tylko wtedy, gdy nie zawiera podgrafu ściągalnego do grafu $K_{3.3}$ lub do grafu K_5

2 Język programowania

Algorytm zostanie zaimplementowany za pomocą języka C#.

3 Algorytm

Algorytm rozwiązujący zadany problem zostanie wybrany w dalszej fazie projektu.

4 Opis algorytmu

Algorytmem, który zostanie zaimplementowany w celu sprawdzenia czy zadany graf jest planarny jest test planarności przy użyciu twierdzena Wagnera:

Twierdzenie 1. *Dany graf jest planarny wtedy i tylko wtedy, gdy nie zawiera podgrafu ściągalnego do grafu* $K_{3.3}$ *lub do grafu* K_5

Implementacja algorytmu wywodzi się wprost z powyższego twierdzenia. Polegała będzie ona na dokonywaniu operacji ściągania grafu do podgrafu o 5 lub 6 wierzchołkach. Jeżeli uda się w ten sposób utworzyć graf K_5 lub $K_{3.3}$ oznaczało to będzie, że wejściowy graf nie jest planarny. Poniżej znajduje się pseudokod algorytmu.

```
1 JestPlanarny(graf g) begin
      Data: Testowany graf
      Result: Czy testowany graf jest planarny
2     if g posiada 6 wierzchołków then
3         Sprawdź czy graf jest K3.3;
4     if g posiada 5 wierzchołków then
5         Sprawdź czy jest K5;
6     if graf posiada więcej niż 5 wierzchołków then
7         foreach krawędź e grafu g do
8             h ← zbuduj graf poprzez ściągnięcie krawędzi e z grafu g;
9             JestPlanarny(h);
```

Algorytm 1: Pseudokod algorytmu

Linie 1-6 algorytmu wydają się dosyć oczywiste. Jeżeli mamy doczynienia z grafem pięcio lub sześcio wierzchołkowym sprawdzane jest odpowiednio czy nie jest to odpowiednio K_5 lub $K_{3.3}$. W linii 8 następuje iteracja po wszystkich krawędziach grafu. Najistotniejszym fragmentem algorytmu wydaje się być linia 9. Budowany jest tam nowy graf poprzez "ściągnięcie" danej krawędzi. Sama operacja zostanie opisana w dalszej części pracy.

4.1 Operacja ściągnięcia

Ściąganie wzdłuż krawędzi polega na utożsamianiu wierzchołków, które łączy dana krawędź i pomijaniu ewentualnych pętli.

Definicja 1. *Mówimy, że graf* G_1 *jest ściągalny do* G_2*, jeśli* G_2 *można otrzymać z* G_1 *poprzez skończony ciąg operacji ściągania wzdłuż krawędzi.*

Z **Twierdzenia 1** wiemy, że jeżeli uda nam się ściągnąć graf wejściowy do K_5 lub $K_{3,3}$ to nie jest planarny.

4.2 Struktury danych

Graf reprezentowany będzie za pomocą listy sąsiedztwa. W reprezentacji tej, struktura przedstawiająca wierzchołek będzie składała się z jego identyfikatora oraz listy jego sąsiadów. Sam graf będzie zbudowany z listy wierzchołków. Reprezentacja ta pozwoli na budowanie grafu poprzez ściągnięcie krawędzi w czasie $O(m)$.

4.3 Historia operacji

Celem zadania jest próba znalezienia podgrafu ściągalnego do K_5 lub $K_{3,3}$, aby odpowiedzieć na pytanie czy dany graf jest planarny. Nie wystarczy więc aby projektowany algorytm zwracał jedynie binarną odpowiedź, tak lub nie, musi w przypadku stwierdzenia nieplanarności zwrócić podgraf ściągalny do K_5 lub $K_{3,3}$. Aby mieć możliwość odtworzenia takiego podgrafu (w trakcie działania algorytmu zmieniana jest struktura grafu, usuwane są krawędzie i wierzchołki) podczas każdego zejścia rekurencyjnego zapamiętywana będzie historia ściągniętych krawędzi. Umożliwi to odtworzenie szukanego podgrafu w zadanym grafie.

4.4 Złożoność

Istnieje, szereg algorytmów pozwalących na sprawdzenie planarności grafu. Od roku 1974 istnieją algorytmy o złożoności liniowej $O(n)$ w tym np. metody Hopcroft, Tarjan czy Boye, Myrvold. Są to metody bardzo skomplikowane, wymagające żmudnej, czasochłonnej implementacji.

Prostszy w implementacji jest algorytm korzystający z obserwacji dotyczących cyklów i segmentów w grafach planarnych. Jest on mniej wydajny niż wspominane powyżej algorytmy, ale jego złożoność jest również wielomianowa - rzędu $O(n^3)$. Niestety, algorytm ten nie mógł być zastosowany, ponieważ w przypadku stwierdzenia nieplanarności nie zwraca on podgrafu ściągalnego do K_5 lub $K_{3,3}$.

Jako, że zadanie projektowe zdefiniowane zostało dla grafu o 12 wierzchołkach - zdecydowaliśmy, że zastosowanie tak skomplikowanych algorytmów nie jest tu zasadne. Zaprezentowane w ninejszej pracy rozwiązanie osiąga wydajność na poziomie $O(m!)$.

5 Bibliografia

Wilson R. J., Wprowadzenie do teorii grafów, Warszawa 1998, Wydawnictwo Naukowe PWN.

`www.win.tue.nl/~nikhil/courses/2012/2WO08/lecture5.pdf`, ost. wiz. 17.12.2013.