University of the Basque Country

Final year project

About Tree Depth

Author:
Asier Mujika

Supervisor:
Dr. Hubert Chen

January 13, 2015

1 Introduction to Graph Theory

1.1 Definition of a graph

A graph is defined as a pair of sets G = (V, E), such that E $\subseteq V^2$. The members of V are called vertices and the ones of E edges. Take into account, that the vertices can be anything, they can even be sets themselves. The usual way to draw a graph is by representing the vertices as individual points and for each edge, draw a link between both elements of that edge. The shape in which a graph is drawn is irrelevant, it will contain the same information.

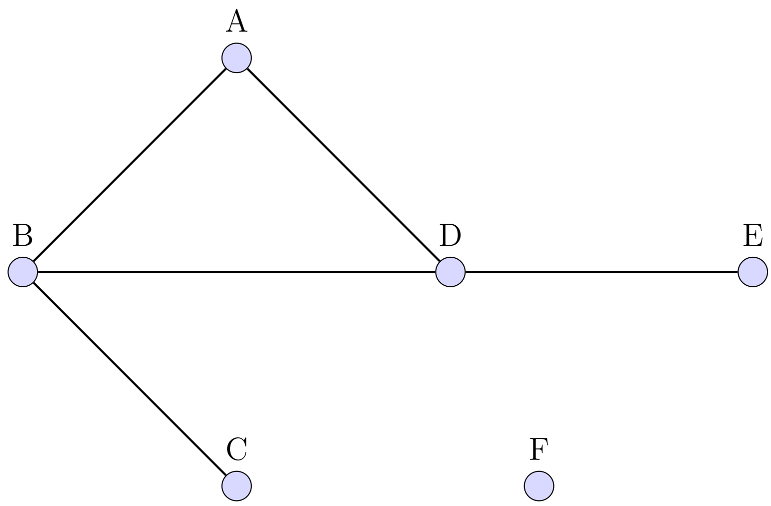

Figure 1: A graph with V = {A, B, C, D, E, F} and E = {{A, B}, {A, D}, {B, D}, {B, C}, {D, E}}

1.2 Connectivity

An essential concept in graph theory is adjacency. Two vertices x, y $\in$ V are said to be adjacent in G if and only if {x, y} $\in$ E.

2 Introduction to Tree Depth

2.1 Basic definitions

Vertex x is said to be the ancestor of y in a rooted forest F, if and only if x belongs to the path between y and the root of the component to which x belongs, y included.

The closure of a rooted forest F, expressed as C = clos(F), is defined as follows:

- V(C) = V(F)
- E(C) = $\{\{x, y\}$: x is an ancestor of y in F, $x \neq y$ $\}$

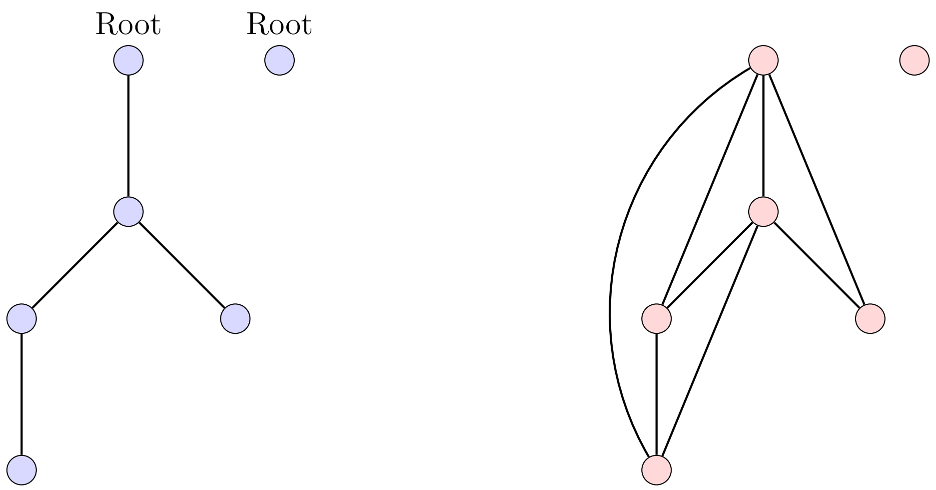

Figure 2: The blue graph at the left is a rooted forest F, the red graph at the right represents clos(F).

2.2 Tree Depth

Definition The tree-depth td(G) of a graph G is the minimum height of a rooted forest F such that $G \subseteq$ clos(F)

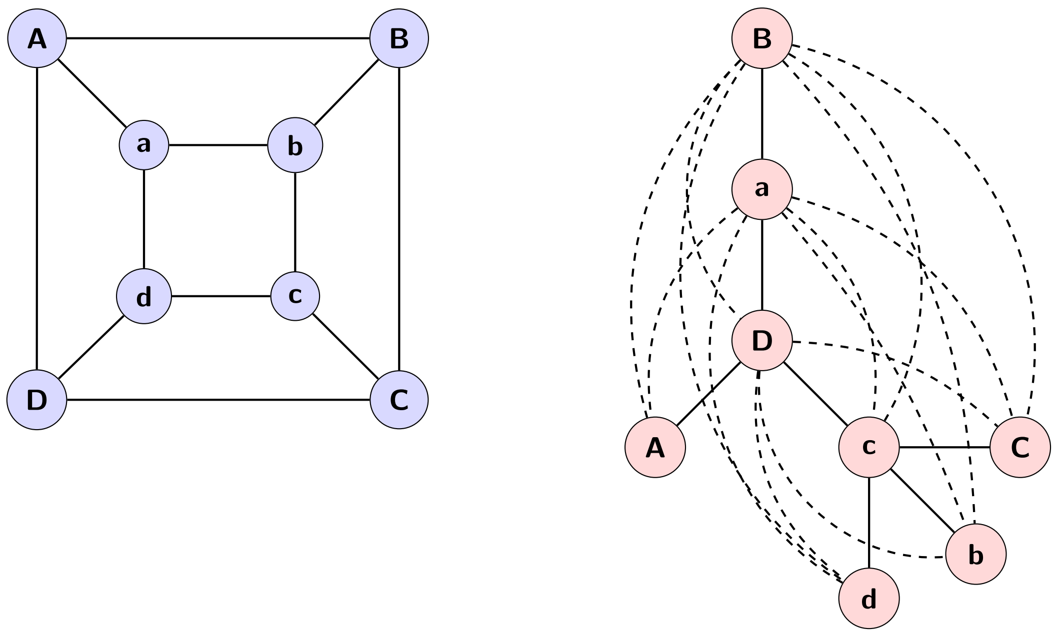

Figure 3: The graph G and tree T are in the left and right respectively. The doted edges in T, represent the clos(T). Because G $\subseteq$ clos(T), we know that td(G) is at most 4.

The tree depth of a graph G is a numerical invariant of a graph. In other words, the tree depth is a property that depends only on the abstract structure of a graph, not in its representation.

2.3 Elimination tree

An elimination tree Y of a connected graph G is defined recursively as follows:

- If V(G) = {x} then Y is just {x}.
- Otherwise, r $\in$ V(G) is chosen as the root of Y and an elimination tree is created for each component of G $-$ r. r will be connected to the root of each of these trees.

The tree T in Figure 3 is an elimination tree for the graph G.

Lemma 2.1 *Let G be a graph and T a tree such that G $\subseteq$ clos(T). Then, Y exists, where Y is an elimination tree of G and height(Y) $\leq$ height(T).*

Proof @TODO

3 Game Theoretic approach to Tree-Depth

3.1 Definition of the game

The k-step selection-deletion game is played by Alice and Bob on a graph G. The game is played by turns as follows:

- First, Alice selects a connected component of the graph, and the rest of the components are deleted.
- Then, Bob deletes a node from the remaining graph.

If Bob deletes the last node at the k-th round or earlier, he is said to win the k-step selection-deletion game. Otherwise, Alice wins.

Lemma 3.1 *Let G be a graph and let Y be a rooted forest of height at most t such that $G \subseteq clos(Y)$. Then Bob has a winning strategy for the (t+1)-step selection-deletion game.*

Proof Because of lemma 2.1 we can assume that Y is an elimination tree without loss of generality. We will proof this by induction on the height of Y.

- **Base case**: If height(Y) = 0, then every component of G will have a single vertex, so it's clear that Bob will win in the 1-step selection-deletion game.
- **Induction**: Let $G_i \subseteq G$ be the component Alice chooses, then Y_i exists such that Y_i is a component of Y, $G_i \subseteq clos(Y_i)$ and obviously $height(Y_i) \leq height(Y)$.

  Assume we have a winning strategy for Bob in t steps for every graph that is contained in a rooted forest of height t-1.