University of the Basque Country

Final year project

About Tree Depth

Author:
Asier Mujika

Supervisor:
Dr. Hubert Chen

January 15, 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.

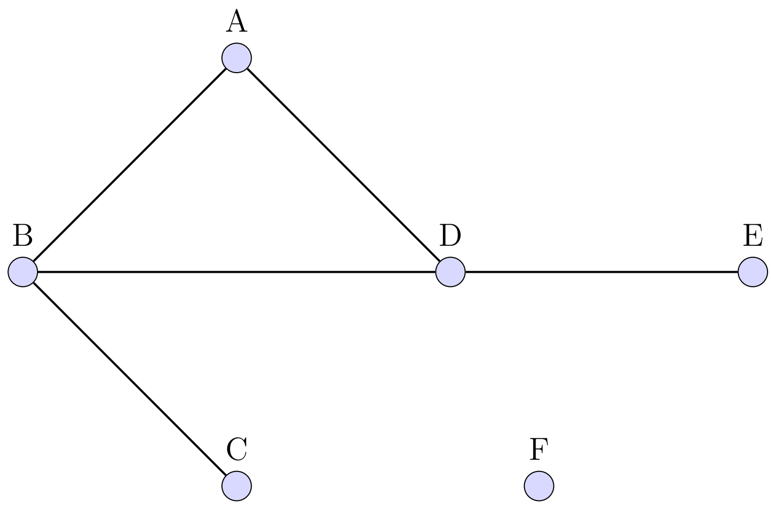

Figure 1: A graph with V = {A, B, C, D, E, F} and E = {{A, B}, {A, D}, {B, D}, {B, C}, {D, E}}

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, $x \neq y\}$

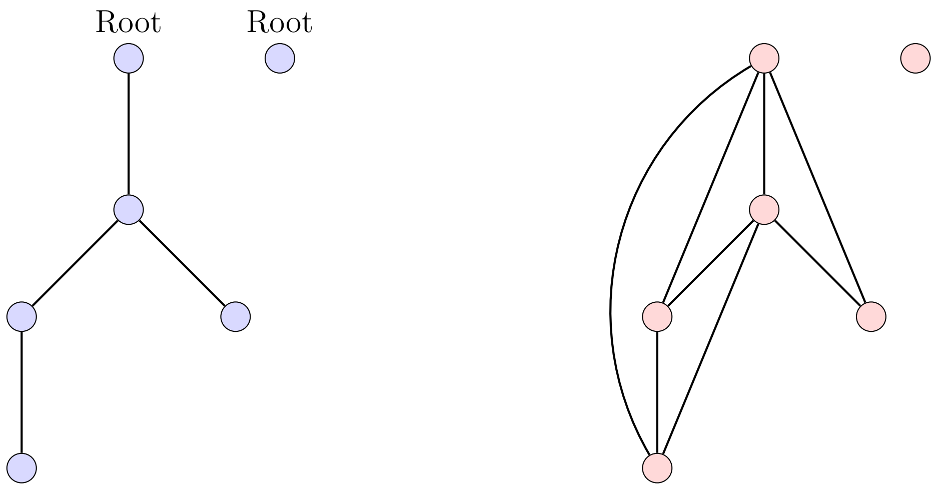

Figure 2: The blue graph at the left is a rooted forest F, the red graph at the right represents clos(F).

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)

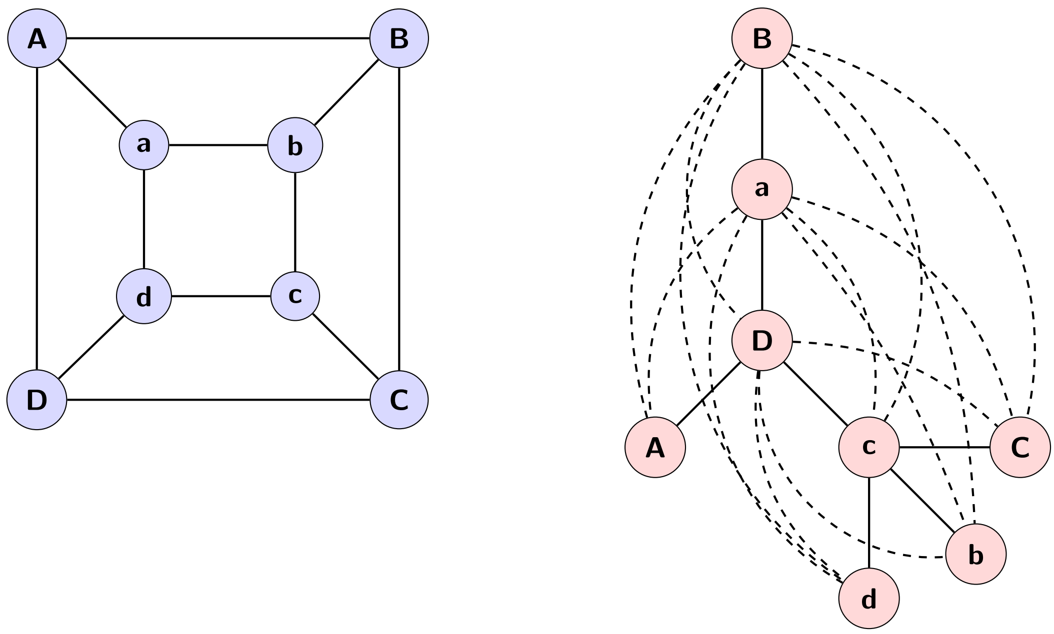

Figure 3: The graph G and tree T are in the left and right respectively. The doted edges in T, represent the clos(T). Because G $\subseteq$ clos(T), we know that td(G) is at most 4.

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 $\square$

3 Game Theoretic approach to Tree-Depth

3.1 Defining 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.

3.2 Bob's winning strategy

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

Proof. Because of lemma 2.1 we know an elimination forest Y exists such that height(Y) $\leq$ height(F). We will proof this by induction over 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 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 an elimination tree belonging to Y, $G_i \subseteq clos(Y_i)$ and obviously $height(Y_i) \leq height(Y) \leq t$. Bob will delete v, the root of Y_i. This will leave us with $G' = G_i - v$ as the new graph. If we consider the sons of v the new roots in $Y' = Y_i - v$, then $G' \subseteq clos(Y')$ because of how the elimination trees are built. As $height(Y') \leq t - 1$, we can assume by induction that Bob has a winning strategy in t rounds for G', which together with the strategy for the first round we have just defined makes a winning strategy for Bob in the (t+1)-step selection-deletion game on the graph G.

□

3.3 Alice's winning strategy

Definition A shelter S in a graph G is a partially ordered set with the next properties:

- $\forall H \in S$, $H \subseteq G$ and H is connected.
- $H < V$ if and only if $H \subset V$.
- If $H \in S$ and H is not minimal, then $\forall v \in V(H), \exists H'$ such that $H' = H - v$ and H covers H'.

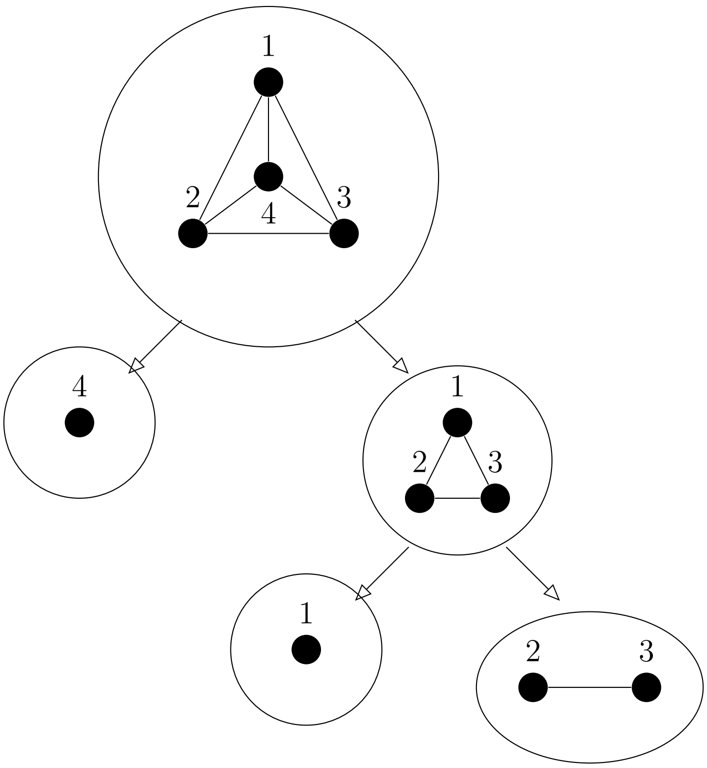

Figure 4: An example of a shelter. The arrows represent the covering relation.

The thickness of a shelter S is the shortest chain from a maximal element in S to a minimal element in S. The thickness of the shelter in figure 4 is 1.

Lemma 3.2. *Let G be a graph, S a shelter in G, and t the thickness of S. Then, S encodes a winning strategy for Alice in the t-step selection-deletion game.*

Proof. We will proof this by induction over t.

- **Base case**: If t = 0, then clearly Alice wins the 0-step selection-deletion game.

- **Induction**: Let H be a maximal element in S. Then, Alice picks the connected component G_i of G, such that $H \subseteq G_i$. Because t > 0, H is not minimal, so for any vertex v that Bob removes, if $v \in H$ there exists $H' \in S$ that is covered by H and $v \notin H'$. Otherwise, H is still a subset of $G_i - v$. Let $S' = \{X \mid X \in S \, and \, X \subseteq G_i - v\}$. It is clear that S' is a shelter for $G_i - v$ and that the thickness of S' is greater than or equal to t-1. By induction we can assume S' encodes a winning strategy in t-1 steps.

□