University of the Basque Country

Final year project

About Tree Depth

Author:
Asier Mujika

Supervisor:
Dr. Hubert Chen

February 3, 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.

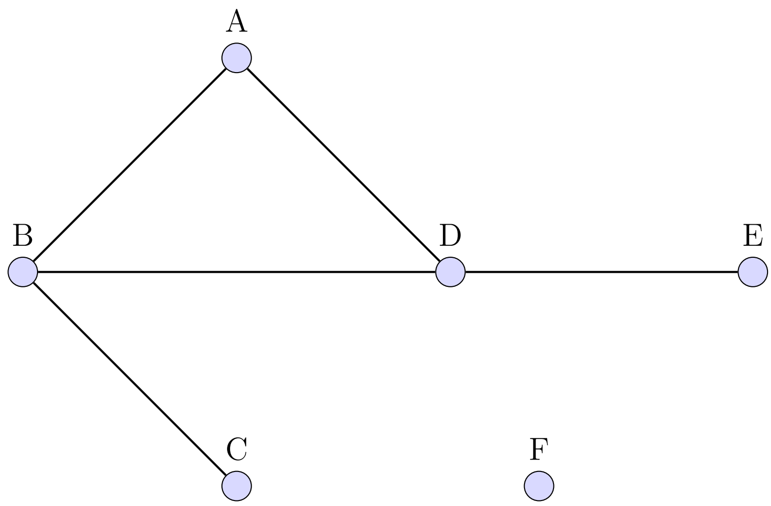


Figure 1: A graph with V = {A, B, C, D, E, F} and E = {{A, B}, {A, D}, {B, D}, {B, C}, {D, E}}

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 $\}$

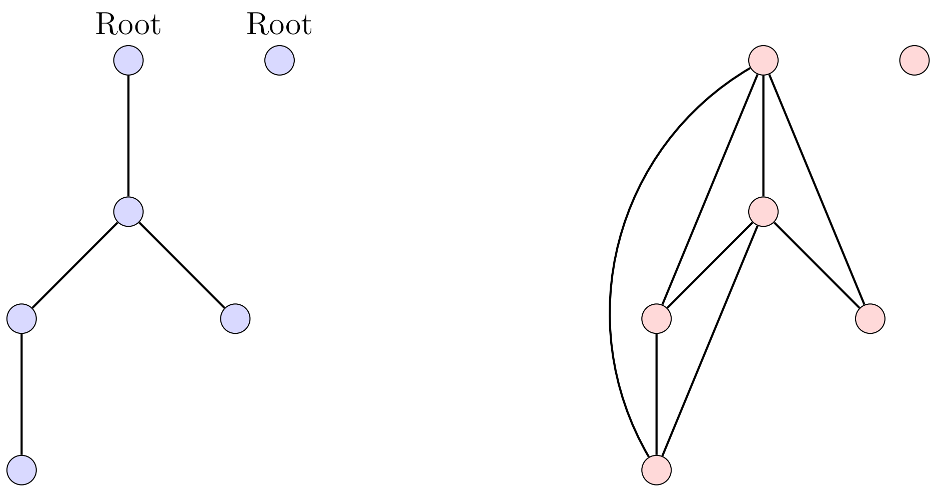
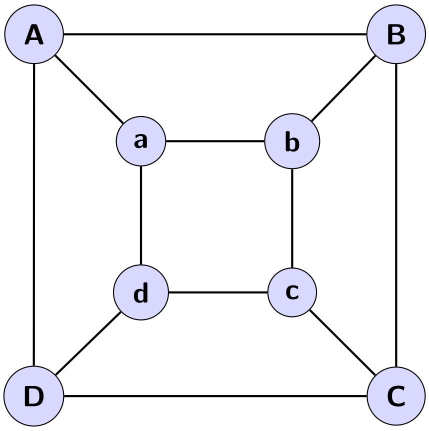


Figure 2: The blue graph at the left is a rooted forest F, the red graph at the right represents clos(F).

2.2 Tree Depth

Definition 2.1. *The tree-depth td(G) of a graph G is the minimum height of a rooted forest F such that G $\subseteq$ clos(F)*



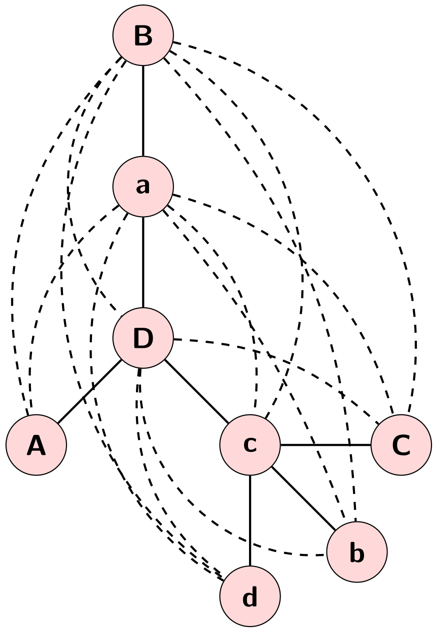


Figure 3: The graph G and tree T are in the left and right respectively. The doted edges in T, represent the clos(T). Because G $\subseteq$ clos(T), height(T) = 5 and the definition of tree depth we just gave, we know that td(G) is at most 5.

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 on its representation.

2.3 Elimination Forest

An elimination forest F of a connected graph G = (V, E) is defined recursively as follows:

- If V = {x} then F is just {x}.
- If G is not connected, then F is the union of the elimination forests of each component of G.
- Otherwise, r $\in$ V is chosen as the root of F and an elimination forest is created for G $-$ r. The roots of this elimination forest will be the children of r in F.

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

Lemma 2.2. *Let G be a graph and F a rooted forest such that G $\subseteq$ clos(F). Then, Y exists, where Y is an elimination forest of G and height(Y) $\leq$ height(F).*

Proof.

Base case: If V(G) = {v}, then V(Y) = {v} and height(Y) = 1. Beware that F can have nodes that are not in G but it must contain v, so height(Y) $\leq$ height(F).

Induction: If G is connected, set the root of F, v, as the root of Y. Clearly, G-v $\subseteq$ clos(F-v), so by induction an elimination forest Y' exists such that G-v $\subseteq$ clos(Y') and height(Y') $\leq$ height(F-v). The roots of Y' will be the children of v in Y and as G-v $\subseteq$ clos(Y'), then G $\subseteq$ clos(Y) and Y is an elimination forest. With that we can prove the lemma like this: height(Y) = 1 + height(Y') $\leq$ 1 + height(F-v) = height(F), so height(Y) $\leq$ height(F).

If G is not connected, then every component G_i in G is contained in the closure of a component F_i in F. Otherwise, the edge between two adjacent nodes in G that are both in G_i but in two diferent components of F wouldn't be in clos(F) and that can't happen. By induction we can assume that for every component G_i, there exists an elimination forest Y_i such that $G_i \subseteq$ clos(Y_i) and height(Y_i) $\leq$ height(F_i). Y will be the union of all these Y_i which is clearly an elimination forest and because for every component of F there exists a component in Y with smaller or equal height, then height(Y) $\leq$ height(F). $\square$

From this lemma we can say that the tree depth of a graph is the minimal height of an elimination forest for that graph. We can now recursively define the tree depth of a graph using the definition of an elimination forest:

Definition 2.3. *The tree depth of a graph G with $G_1, \ldots, G_k$ components is the following:*

$$td(G) = \begin{cases} 1 & \textit{if } |G| = 1 \\ max_{i=1}^{p} td(G_i) & \textit{if G is not connected} \\ 1 + min_{v \in V(G)} td(G - v) & \textit{otherwise} \end{cases}$$

3 Game Theoretic approach to Tree-Depth

3.1 Defining the game

For k $\geq$ 0, the k-step selection-deletion game is played by Alice and Bob on a graph. 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 and the next round is played with this graph.

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

From this definition we can observe that if Bob has a strategy to win in k rounds that strategy will also guarantee a win in any game that lasts more than k rounds. Conversely, if Alice has a winning strategy in k-rounds, that same strategy will also win any game with less than k rounds.

3.2 Bob's winning strategy

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

Proof. Because of lemma 2.2 we know an elimination forest Y exists such that height(Y) $\leq$ height(F). Consider h = height(Y), we will prove that a winning strategy exists in h rounds which is also a winning strategy in the t-step selection-deletion game because h $\leq$ t.

- **Base case**: If h = 1, 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 $height(Y_i) \leq h$. 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 children of v the new roots in $Y' = Y_i - v$, then $G' \subseteq clos(Y')$ because of how the elimination trees are built. As $height(Y') \leq h - 1$, we can assume by induction that Bob has a winning strategy in h-1 rounds for G', which together with the strategy for the first round we have just defined makes a winning strategy for Bob in the h-step selection-deletion game on the graph G.

$\square$

3.3 Alice's winning strategy

Definition 3.2. *A shelter S in a graph G is a set of graphs with the next properties:*

- $\forall H \in S$, $H \subseteq G$ *and* H *is connected.*
- $H \leq M$ *if and only if* $H \subseteq M$.
- *If* $H \in S$ *and* H *is not minimal, then* $\forall v \in V(H)$, *there exists* $H' \subseteq H - v$ *such that* H *covers* H'. *In a partially ordered set* a *covers* b *means that* $a \neq b$, $a \geq b$, *and no* c *exists such that* $c \neq a \neq b$ *and* $a \geq c \geq b$.

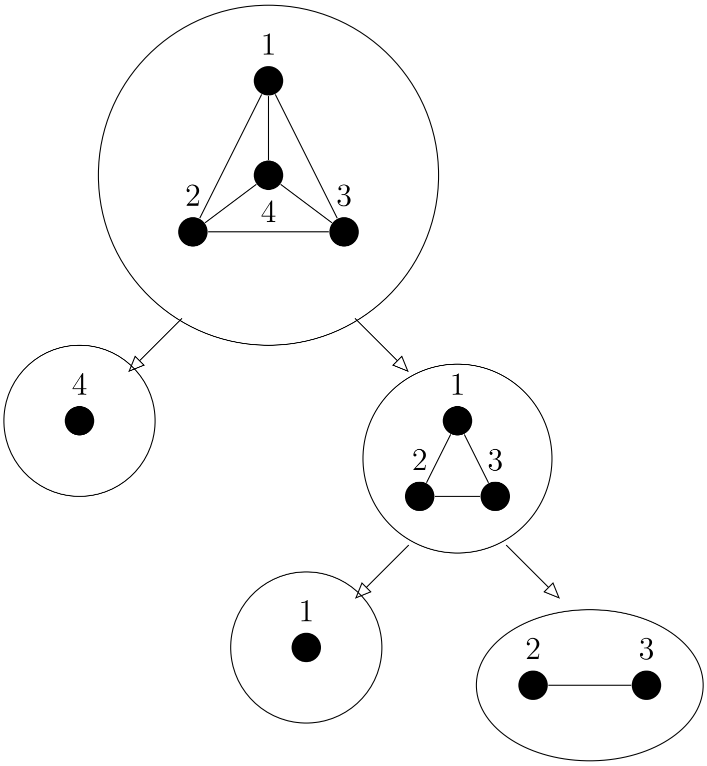


Figure 4: An example of a shelter. The arrows represent the covering relation.

The thickness of a shelter S is the shortest chain from a maximal element in S to a minimal element in S. The length of a chain is defined as the number of elements in it minus one. The thickness of the shelter in figure 4 is 1, because of the chain $\{1, 2, 3, 4\} \geq \{4\}$.

Lemma 3.3. *Let G be a graph, S a shelter in G, and t the thickness of S. Then, there exists 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 subgraph of $G_i - v$. Let $S' = \{X \mid X \in S \wedge X \subseteq G_i - v\}$. 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 Alice has a winning strategy in t-1 steps in $G_i - v$, which together with the strategy for the first round we have just defined is a winning strategy for the t-step selection-deletion game.

□

3.4 Relation to Tree-Depth

It is clear that if Alice has a winning strategy in the t-step selection deletion game, Bob can't have a winning strategy in that same game. Because of this and lemmas 3.1 and 3.3 we can state the following:

Theorem 3.4. *Let G be a graph, S a shelter in G of thickness x and F a rooted forest of height y such that G ⊆ clos(F). Then the following is true.*

1. *Alice has a winning strategy in the t-step selection-deletion game, for any t smaller than or equal to x.*

2. *Bob has a winning strategy in the (t+1)-step selection-deletion game, for any t greater than or equal to y.*

3. *Every rooted forest who's closure contains G has an height higher than or equal to x. Otherwise, Bob would have a winning strategy in the x-step selection-deletion game, which contradicts statement 1.*

4. *Every shelter in G has a thickness smaller than or equal to y. Otherwise, Alice would have a winning strategy in the (y+1)-step selection-deletion game, which contradicts statement 2.*

5. *Because we have F, td(G) $\leq y$. Also, from statement 3 it is clear that $x \leq td(G)$. So we can say that $x \leq td(G) \leq y$.*

With this theorem we can now give an upper-bound and a lower-bound to a graphs tree depth.

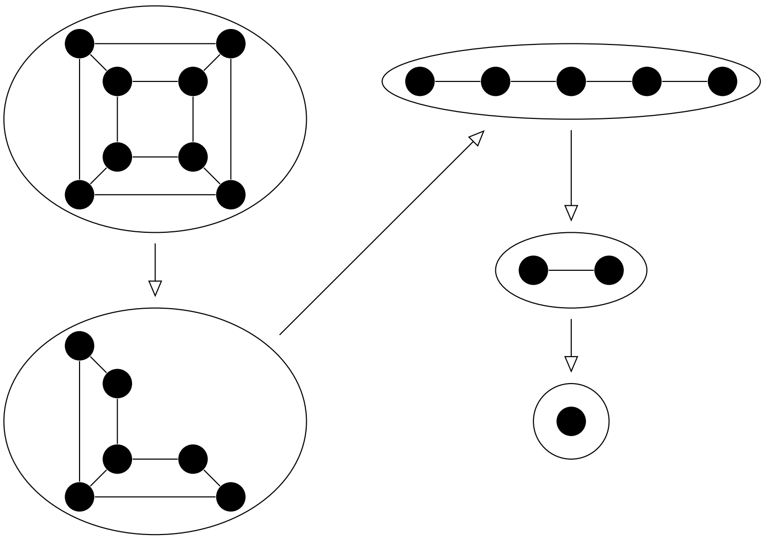


Figure 5: This is a shelter of thickness 4 for the graph in Figure 3. Beware that not all graphs in the shelter are drawn, but every graph in the shelter is isomorphic to these. With this and the rooted forest from Figure 3 we can say that td(G) = 4.

4 Cycle rank

4.1 Defining cycle rank

Cycle rank is a numerical invariant in a directed graph which is closely related to the tree depth of an undirected graph.

Definition 4.1. *The cycle rank of a digraph G, denoted by r(G) is defined as follows:*

- *If G is acyclic, then r(G) = 0.*
- *If G is nontrivially strongly connected, then r(G) = 1 + $min_{v \in V(G)} \{r(G - v)\}$.*
- *If G is not strongly connected and contains at least a cycle, then r(G) is the maximum cycle rank among all nontrivial strongly connected components of G.*

The graph with a single node and no edges is considered trivially strongly connected, while the graph with a single node and a loop is considered non-trivially strongly connected.

4.2 Directed elimination forest

Similar to the notion of elimination forests in undirected graphs, we have directed elimination forests on digraphs.

Definition 4.2. *A directed elimination forest for a digraph G is a rooted forest* $F = (\nabla, \xi)$. *F can be defined recursively as follows:*

- *For the* $k \geq 0$ *non trivially strongly connected components of G,* $Y_1, \ldots, Y_k$, *(*v_i*,* Y_i*) are the roots in F, where* $v_i \in Y_i$ *and* $1 \leq i \leq k$.
- *For each (*v_i*,* Y_i*), a directed elimination forest is created for G[*Y_i*] -* v_i *and the roots of that forest are the children of (*v_i*,* Y_i*) in F.*

Lemma 4.3. *Let F be directed elimination forests of minimum height for a digraph G = (V, E). Then, r(G) = 1 + height(F).*

Proof. We will proof this by induction on the number of vertices of G.

- **Base case**: If G is acyclic, then r(G) = 0, height(F) is -1 because we assume that the height of the empty tree is -1.
- **Induction**: If G is nontrivially strongly connected, then v $\in$ V exists, such that r(G) = 1 + r(G-v). Let (v, V) be the root of F, then height(F) = 1 + height(F') where F' is any directed elimination forest of G-v because of definition 4.2. If we consider F' to be the directed elimination forest of G-v of minimum height, by induction we can assume that r(G-v) = 1 + height(F'). So, r(G) = 1 + r(G-v) = 2 + height(F') = 1 + height(F).

  If G is not strongly connected but it has at least a cycle, then, for every X that is a strongly connected component of G by induction we can assume that r(G[X]) = 1 + height(F_X) where F_X is the directed elimination tree of minimum height for G[X]. Because r(G) is the maximum among all r(G[X]) and the height(F) is the maximum among all height(F_X), r(G) = 1 + height(F).

□