CSL851: Algorithmic Graph Theory **Fall 2013**

Lecture 8: November 13

Lecturer: Amit Kumar *Scribes: Sushant Saxena*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

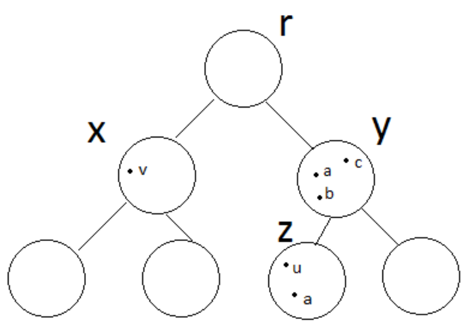

Figure 8.1: Example Tree Decomposition

8.1 Independent Set

After removing y, there is no path between $u \in B_z$ and $v \in B_x$.

Definition 8.1 G_x *: graph induced by* $\bigcup_{y \text{ is descendant of } x} B_y$

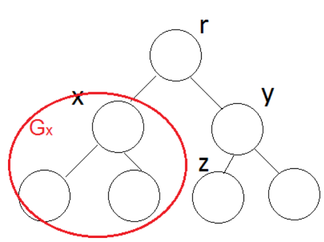

Figure 8.2: Example of G_x

Definition 8.2 $A(x, S)$ *: max independent set* I *in in* G_x *st* $I \cap B_x = S$

Possibilities of $x = \#nodes = O(2^n)$
Possibilities of $S \leq 2^{k+1}$
where $k =$ tree width

8.1.1 Dynamic Programming

Vertices appearing in r and x doesn't appear in G_y

Definition 8.3 *(r, T) and (x, S) are consistent if for every $v \in B_T \cap B_x$, $v \in T$ iff $v \in S$.*

$A(r, T) = |T| + max_{(r,T)\ \&\ (x,S)\ \ consistent}\{A(x, S) - |T \cap S|\} + max_{(r,T)\ \&\ (y,U)\ \ consistent}\{A(y, U) - |T \cap U|\}$

8.2 Chromatic Numbering

Theorem 8.4 *Any graph with tree width p has a vertex of degree $\leq p$*

Proof:
- There is a leaf z st $B_z \nsubseteq B_y$, y : parent of z(If $B_z \subseteq B_y$ then z can be removed and still it will be a valid tree decomposition)
- $u \in B_z \setminus B_y$
- z is the only node st B_z contains $u \Rightarrow$ all neighbors of u are in B_z
- Therefore, degree of u is $\leq p$ ■

Theorem 8.5 *Any graph with tree width p can be colored with $\leq p + 1$ colors*

Proof:
- Take a vertex v with degree $\leq p$ (Using theorem 8.4, such v exists) and remove it.
- New graph has tree width $\leq p$
- Color this graph with $\leq p + 1$ colors
- Color v(only p neighbors, so one color must be free) ■

8.2.1 Find a way of coloring with k colors

Definition 8.6 *$A(x, \chi)$ = Yes if there is a k coloring of G_x st $\forall v \in B_x$, v gets the colour $\chi(v)$*

Possibilities of χ : k^{p+1}, $k \leq p$
In the example tree decomposition 8.1, G_x and G_y can be colored separately as there is no edge between them.

Definition 8.7 *(r, ϕ) and (x, χ) are consistent if $\forall v \in B_x \cap B_r, \chi(v) = \phi(v)$*

8.2.1.1 Dynamic Programming

$A(r, \phi)$ = true if $\exists\chi$ st (r, ϕ) and (x, χ) are consistent and $A(x, \chi)$ is true and $\exists\chi'$ st (y, χ') and (r, ϕ) are consistent and $A(y, \chi')$ is true

8.3 Connectivity

Consider any tree T.

Definition 8.8 *T_v : sub-tree rooted at v.*

Theorem 8.9 *$\exists v \in T$, removing v each component has $\leq \frac{n}{2}$ vertices.*

Proof: *Proof by finding a vertex v st $|T_v| > \frac{n}{2}$ and $|T_w| \leq \frac{n}{2}$ for each child w of v.*
- Try removing r(Initially, r is root)
- If it doesn't work $\rightarrow$ there is a child w having $|T_w| > \frac{n}{2}$ (Note that there will only be one such child)
- Set r to be v and repeat the procedure.
Above algorithm will terminate as on each step it moves to the child of current vertex. So, its output will be the vertex v with $|T_v| > \frac{n}{2}$ and $|T_w| \leq \frac{n}{2}$ for each child w of v. On removing this vertex v, each component will have $\leq \frac{n}{2}$ vertices. ■

Theorem 8.10 *Suppose we give a weight w_v for each vertex v st $\Sigma_v w_v = 1$, then there is a vertex in the tree st after removing it, total weight of each component is $\leq \frac{1}{2}$*

Proof: *Proof follows the same idea, by finding a vertex v st $w(T_v) > \frac{1}{2}$ and $w(T_u) \leq \frac{1}{2}$ for each child u of v*
- Try removing r(Initially, r is root)
- If it doesn't work $\rightarrow$ there is a child u having $w(T_u) > \frac{1}{2}$ (Note that there will only be one such child)
- Set r to be v and repeat the procedure.

Above algorithm will terminate as on each step it moves to the child of current vertex. So, its output will be the vertex v with $w(T_v) > \frac{1}{2}$ and $w(T_u) \leq \frac{1}{2}$ for each child u of v. On removing this vertex v, each component will have weight $\leq \frac{1}{2}$. ■

Now, consider a graph with tree width $= p$.

Theorem 8.11 *G has tree width p and $\Sigma_v w_v = 1$, then there is a set of $p+1$ vertices st after removing it, total weight of each component $\leq \frac{1}{2}$*

Proposition 8.12 *Graph with small tree width(say p) is easy to disconnect. Removing $p+1$ vertices disconnects graph with each component being not very big.*

Proposition 8.13 *Grid graph is not easy to disconnect.*

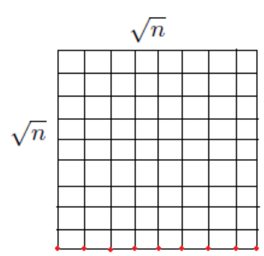

Figure 8.3: Grid graph

Consider any grid graph G, which has weights as $\frac{1}{\sqrt{n}}$ for bottom row vertices(red vertices in fig 8.3) and 0 weight for every other vertex.

Theorem 8.14 *Removing $< \frac{\sqrt{n}}{2}$ vertices in G, we can't get components where each component has weight $\leq \frac{1}{2}$*

Proof: *Proof by contradiction. Suppose $\exists S$ of $\frac{\sqrt{n}}{2}$ vertices st after removing it, every component has weight $\leq \frac{1}{2}$*
A column(or row) is free if no vertex in that column(or row) is removed.
There are atleast $\frac{\sqrt{n}}{2}$ free columns(marked with green).
There is a free row(marked with green).
So, $\frac{\sqrt{n}}{2}$ free columns will be in same connected component and hence, its weight will be atleast $\frac{1}{2}$.

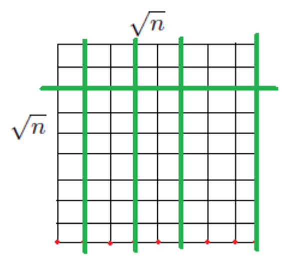

Figure 8.4: Free columns/row

■

Theorem 8.15 *Planar Separator Theorem : For any planar graph removing at most $O(\sqrt{n})$ vertices will split it into small pieces($\leq \frac{2n}{3}$)*

Proof: *Done in Next Class* ■