Universal Graphs
(& Implicit Representation)

Cyril Gavoille

LABORATOIRE BORDELAIS DE RECHERCHE EN INFORMATIQUE LaBRI - *University of Bordeaux*

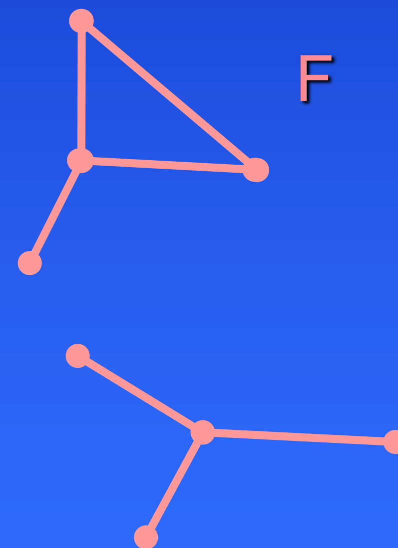
JCB
@Home on BBB
Feb. 4, 2021

Induced-Universal Graphs

[Babai,Chung,Erdös,Graham,Spencer ’82]

A graph U is an **induced-universal** graph for the family F if every graph of F is isomorphic to an *induced* subgraph of U.

U



F

Goal: Find universal graph for a given F(n) with few vertices, w.r.t. to *n*

Representation of a Graph

adjacency list

1	2,5
2	1,3
3	2,4,5
4	3
5	1,3

matrix

$$\begin{array}{c|ccccc} & 1 & 2 & \mathbf{3} & 4 & 5 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & 1 & 0 & 0 \\ \mathbf{3} & 0 & 1 & 0 & 1 & 1 \\ 4 & 0 & 0 & 1 & 0 & 0 \\ 5 & 1 & 0 & 1 & 0 & 0 \end{array}$$

1 node = 1 pointer in the data-structure

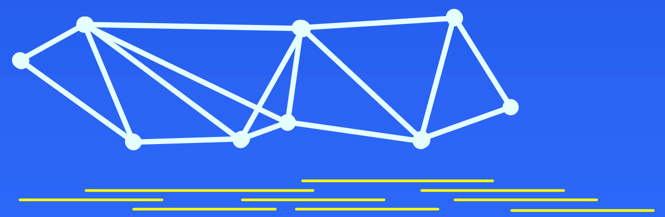
(it does not carry any specific information)

Implicit Representation

To associate with the nodes more information, typically adjacency, and to remove the data-structure.

Interval graphs: $u \mapsto \mathrm{I}(u) \subseteq [1,2n]$

Edges: $u - v \;\Leftrightarrow\; \mathrm{I}(u) \cap \mathrm{I}(v) \neq \emptyset$



Compact representation: $O(\log n)$ bits/node

Possibly time $O(n)$ algorithms vs. $O(n+m)$

Labelling Schemes

P = a graph property defined on pairs of nodes
F = a graph family

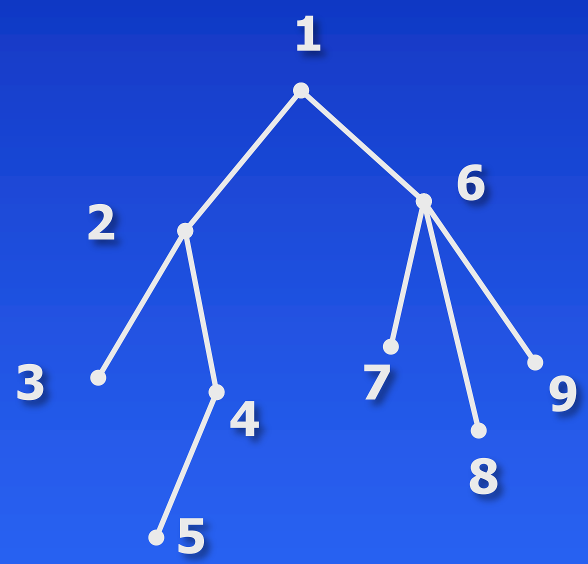
A P-labelling scheme for F is a pair ($\boldsymbol{\lambda}$,**f**) such that $\forall$ G $\in$ F, $\forall$ $u,v \in$ V(G):

- [labelling] $\boldsymbol{\lambda}(u,G)$ is a binary string
- [decoder] $\mathbf{f}(\boldsymbol{\lambda}(u,G),\boldsymbol{\lambda}(v,G))=P(u,v,G)$

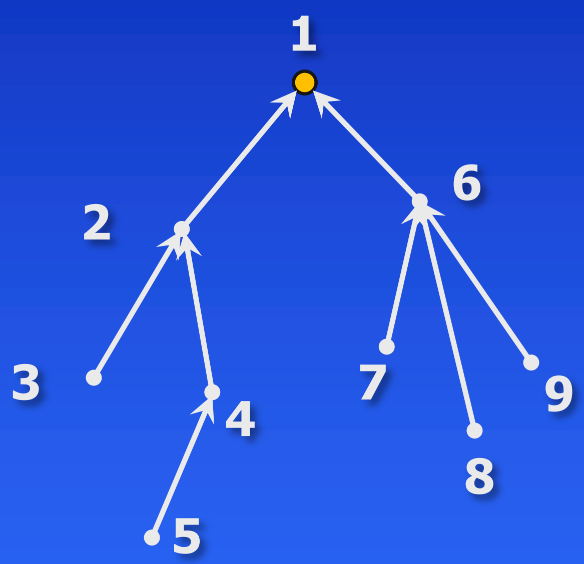
Goal: to minimize the maximum label size
In this talk: $P(u,v,G)$ is TRUE $\Leftrightarrow$ $uv \in E(G)$

Basic Example: Trees

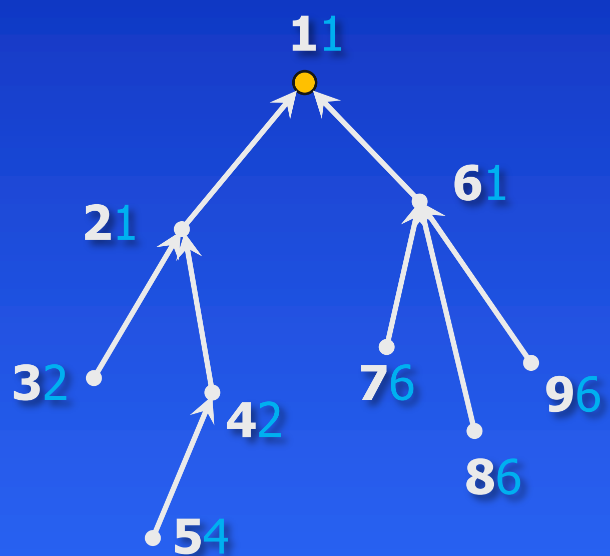
Basic Example: Trees



Basic Example: Trees



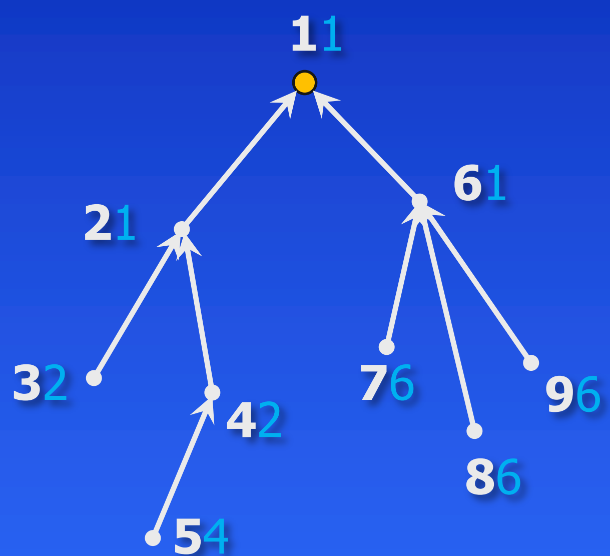
Basic Example: Trees



$\lambda(u,T)=(u,\text{parent}(u))$ or (u,u)

$f(uv,xy)=(v=x$ or $u=y)$

Basic Example: Trees



$\lambda(u,\mathrm{T})=(u,\mathrm{parent}(u))$ or (u,u)

$\mathbf{f}(uv,xy)=(v=x$ or $u=y)$

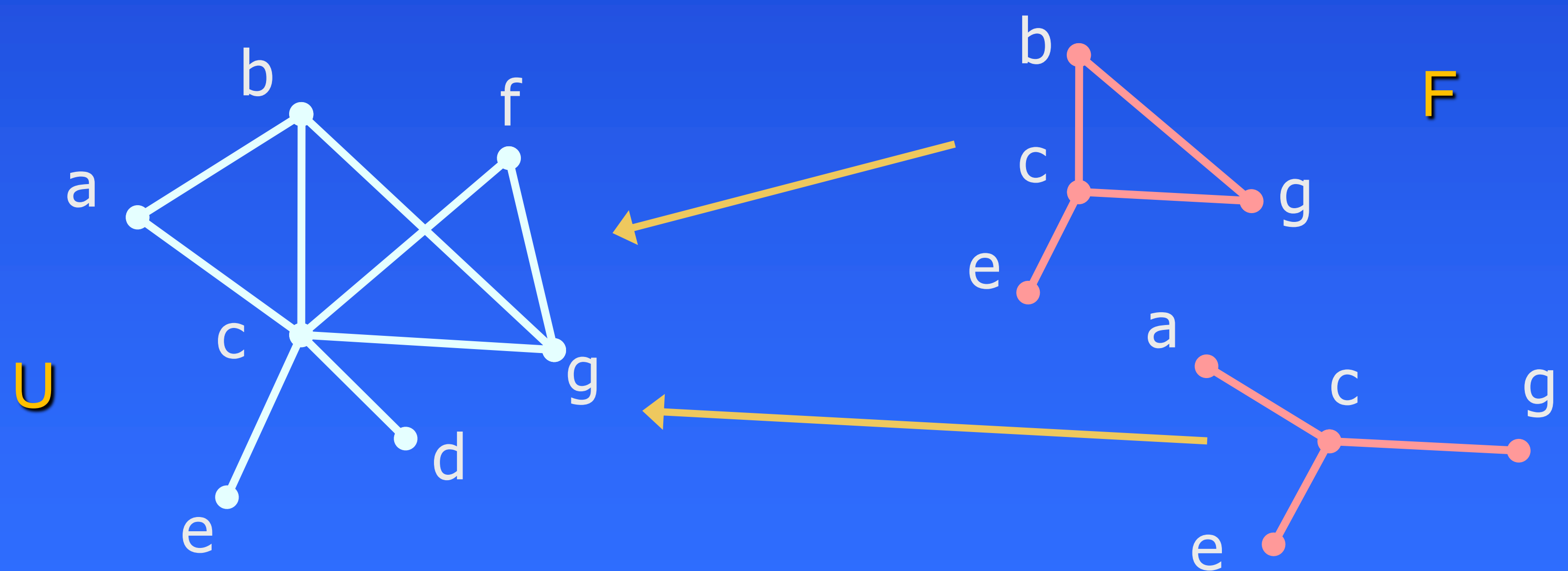
For trees with n nodes: ~ **2**log$n$ bits/node

(the constant does matter [Abiteboul et al. - SICOMP '06])

Induced-Universal Graphs

[Babai,Chung,Erdös,Graham,Spencer ’82]

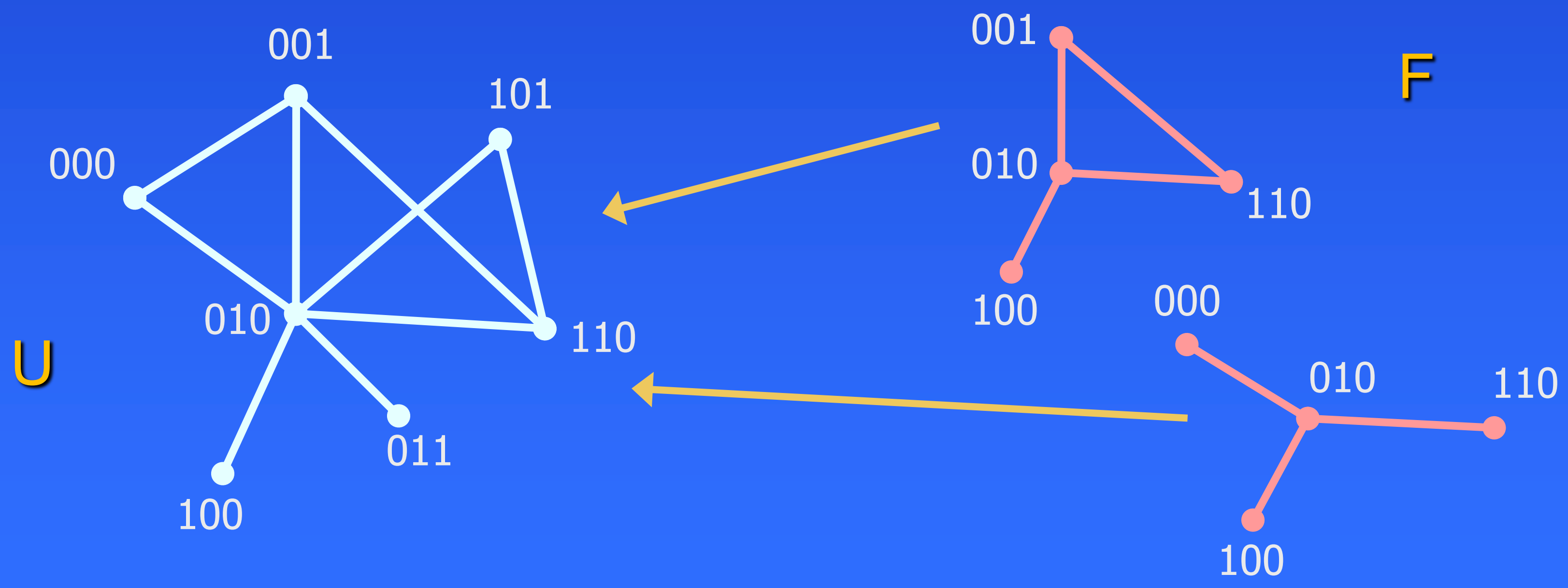
A graph U is an **induced-universal** graph for the family F if every graph of F is isomorphic to an *induced* subgraph of U.



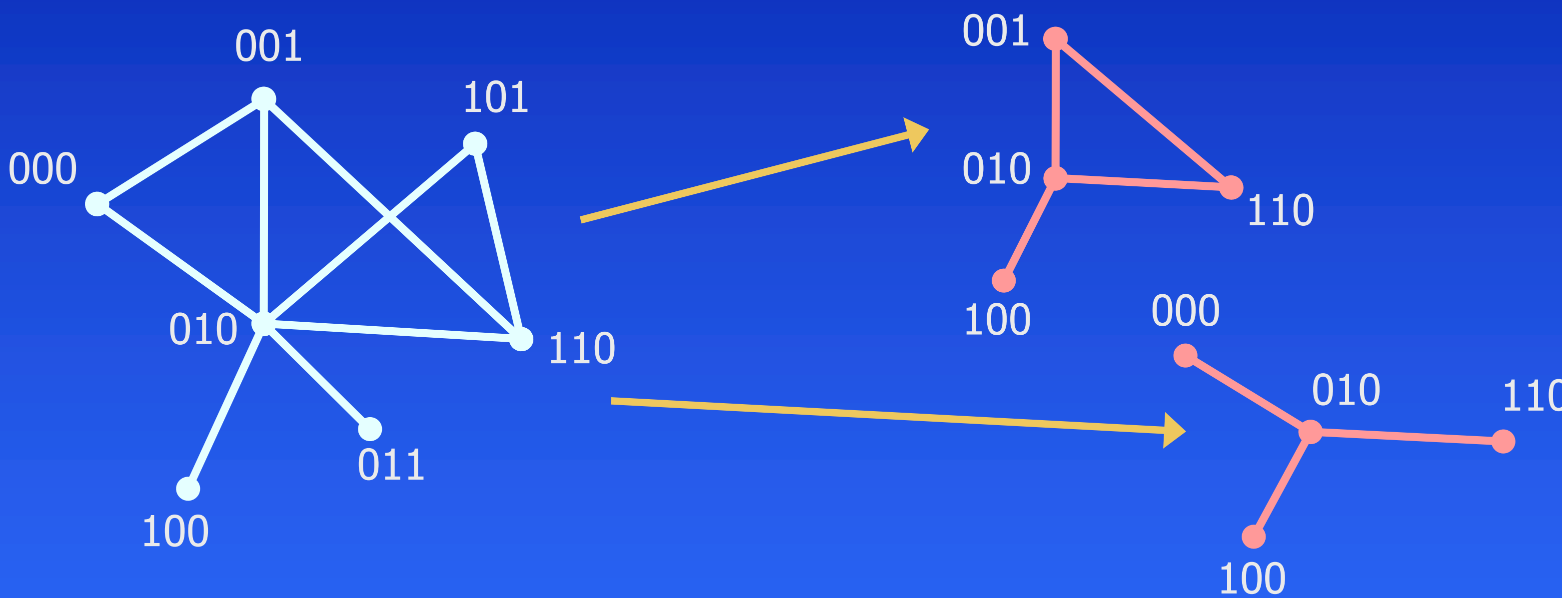
Induced-Universal Graphs

[Babai,Chung,Erdös,Graham,Spencer '82]

A graph U is an **induced-universal** graph for the family F if every graph of F is isomorphic to an *induced* subgraph of U.



Induced-Universal Graphs



induced-universal graph U   graphs of F

$c\log n$-bit labelling ⇔ induced-universal graph of n^c nodes

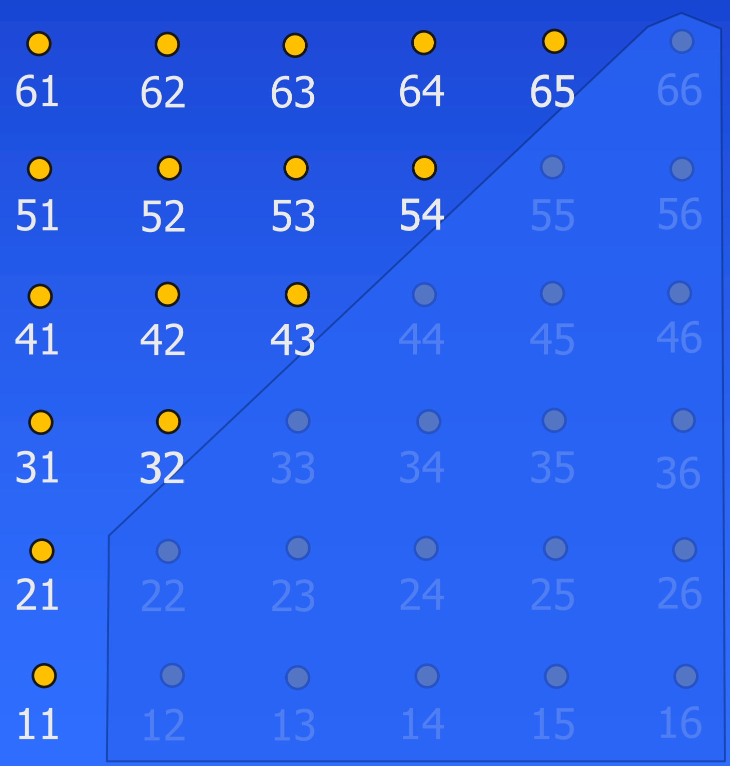
Universal Graphs for Trees
(for n=6 nodes)

From the (u,parent(u)) labelling
$\Rightarrow$ induced-universal graph of n^2=36 nodes

Using DFS for T: (u,v)
$\Rightarrow$ u>v or u=v=1

Universal Graphs for Trees
(for n=6 nodes)

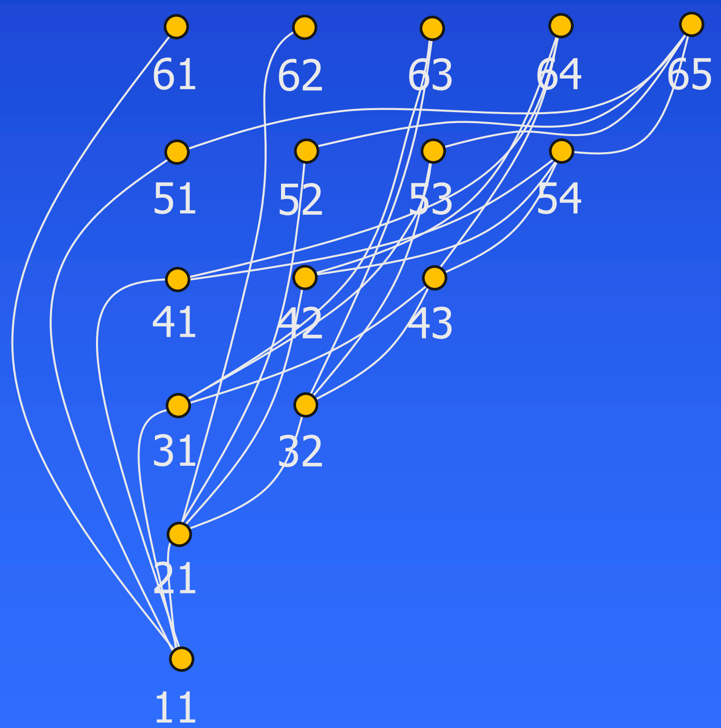
From the (u,parent(u)) labelling
$\Rightarrow$ induced-universal graph of n^2=36 nodes



Using DFS for T: (u,v)
$\Rightarrow$ $u>v$ or $u=v=1$
$\Rightarrow$ $n(n\text{-}1)/2+1=16$ nodes

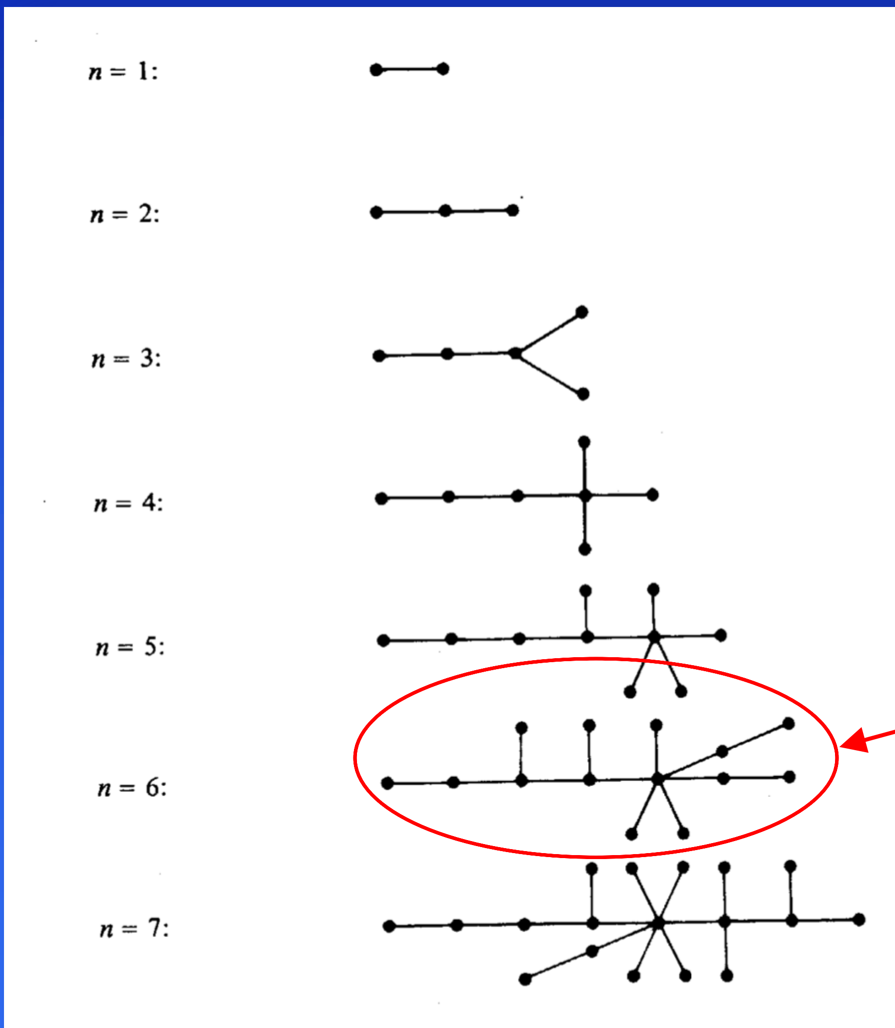
Universal Graphs for Trees
(for n=6 nodes)

From the (u,parent(u)) labelling
$\Rightarrow$ induced-universal graph of n^2=36 nodes



Using DFS for T: (u,v)
$\Rightarrow$ u>v or u=v=1
$\Rightarrow$ n(n-1)/2+1=16 nodes

Universal Graphs for Trees
(universal trees)



ON GRAPHS WHICH CONTAIN ALL SMALL TREES, II.

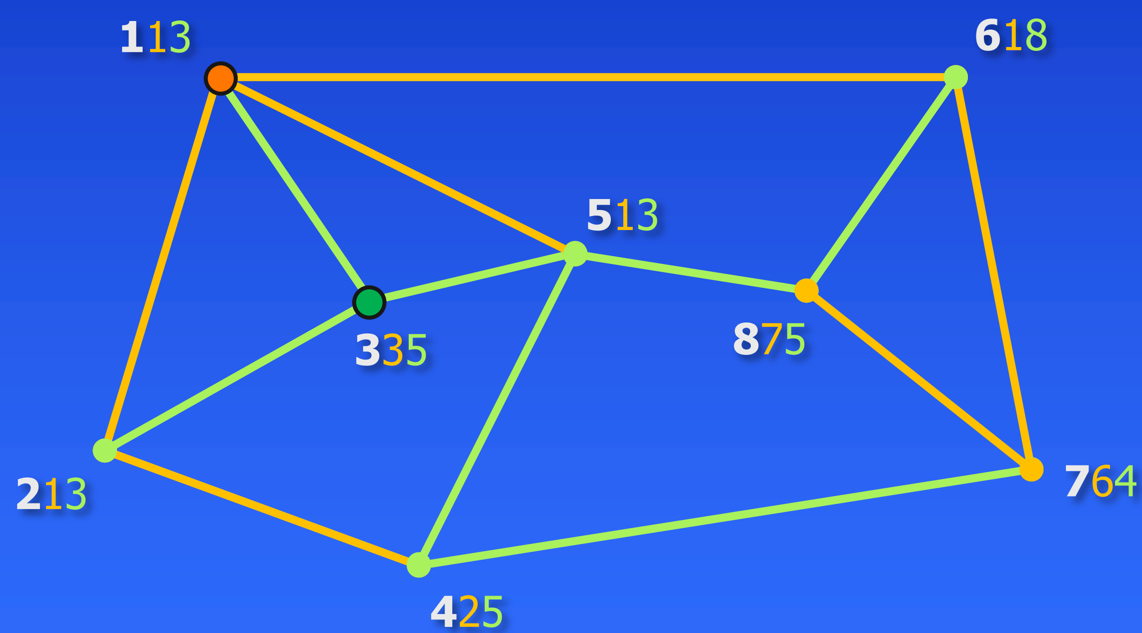
F.R.K. CHUNG — R.L. GRAHAM — N. PIPPENGER

COLLOQUIA MATHEMATICA SOCIETATIS JÁNOS BOLYAI
18. COMBINATORICS, KESZTHELY (HUNGARY), 1976.

14 nodes

Labelling Schemes for Planar Graphs

Edge partition: combining schemes



Arboricity-k graphs: $(k+1)\log n$ bits

$\Rightarrow$ Planar ($k=3$): **4**$\log n$ bits [KNR – STOC'88]

Ex: Degree at most 2?

Better Labelling Schemes

For trees: log$n$ + O(log$^*n$), log$n$ + O(1)

[Alstrup,Rauhe – FOCS'02]

[Alstrup,Dahlgaard,B.T.Knudsen – FOCS'15 & JACM'17]

⇒ Arboricity-k: $k$log$n$ + O(1)

⇒ Planar: **3**log$n$ + O(1)

For treewidth-k: log$n$ + O($k$loglog$n$)

[G.,Labourel – ESA'07]

⇒ Planar & Minor-free: **2**log$n$ + O(loglog$n$)

Bonamy, G., Pilipczuk - SODA'20

For planar & bounded genus: $^4/_3 \log n + O(\log\log n)$
$\Rightarrow$

Induced-universal graph of $n^{4/3+o(1)}$ nodes for n-node planar graphs
(and bounded genus graphs)

Labelling the nodes is polynomial
Decoding adjacency takes constant time

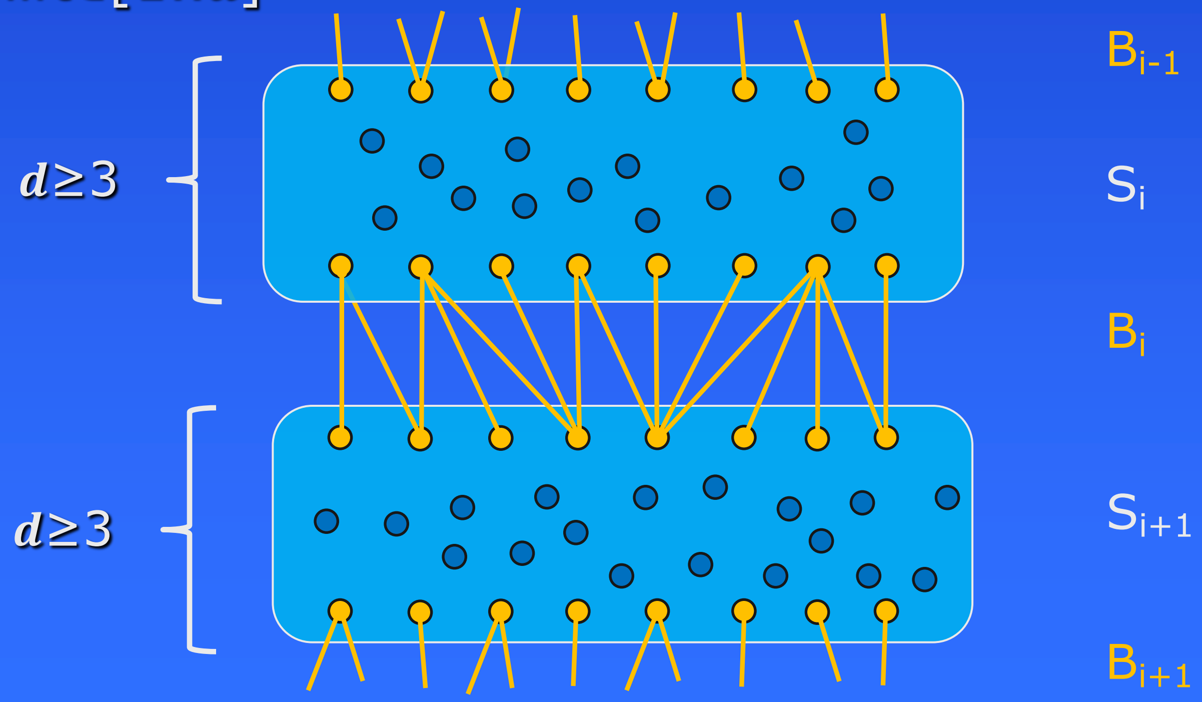
Sketch of Proof (1/2)

Edge partition: G = S ∪ B (Strips & Border)

S: components have d layers $\sim n^{1/3}$

B: has treewidth≤5 and $n/d \sim n^{2/3}$ nodes

BFS & shift∈$[1..d]$



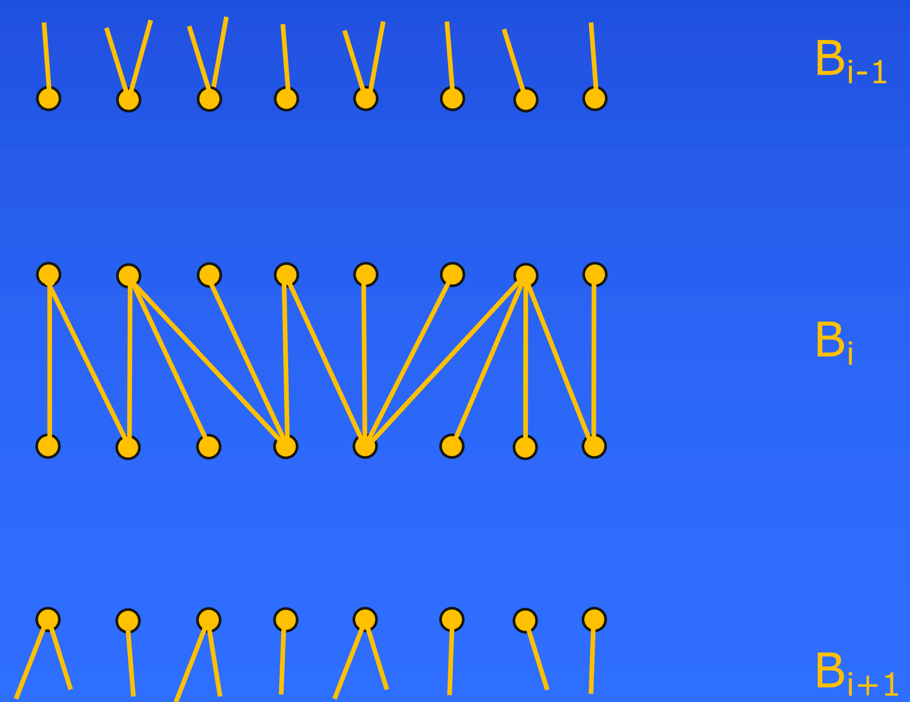
Sketch of Proof (1/2)

Edge partition: $G = S \cup B$ (Strips & Border)

S: components have d layers $\sim n^{1/3}$

B: has treewidth$\leq$5 and $n/d \sim n^{2/3}$ nodes

BFS & shift$\in[1..d]$



Sketch of Proof (2/2)

Labelling for B: $\log(n/d)$ ✔

Labelling for S: $\log n + \log d$ ***new!***

[up to $+O(\log\log n)$ terms]

Problem: nodes in V(B) pay both labels

$\Rightarrow \log(n/d) + \log n + \log d = 2\log n$ ☹

Improved labelling for S: nodes in V(B) pay only $\log|B| = \log(n/d)$ bits!

$\Rightarrow$ nodes in S\V(B): $\log n + \log d = \frac{4}{3}\log n$ ☺

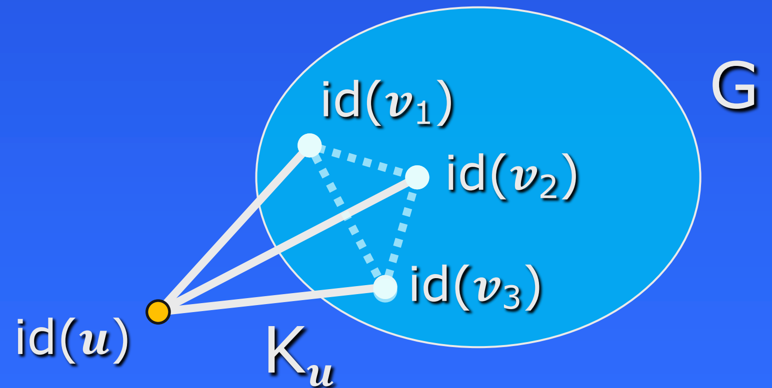
$\Rightarrow$ nodes in V(B): $\log(n/d) + \log(n/d) = \frac{4}{3}\log n$ ☺

Improved Scheme for Treewidth-k

G = treewidth-k, $V(G) = V_1 \cup V_2$, $K_u = N[u]$ = simplicial complex of u, $|K_u| \le k+1$.

Lemma. G has a scheme providing, for each u, $id(u)$ and $\lambda(u)$ st. $\forall v \in K_u$ $id(v)$ can be extracted from $\lambda(u)$. Moreover, for $u \in V_i$

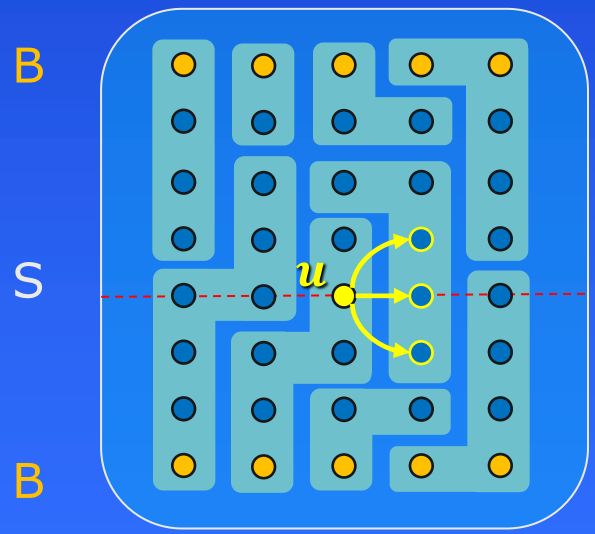
$$|\lambda(u)| = \log|V_i| + O(k \log\log|V(G)|).$$



$$u - v \Leftrightarrow id(v) \in \{id(K_u)\} \text{ or } id(u) \in \{id(K_v)\}$$

Labelling Scheme for S

Key lemma. [2018,2019] If G is planar, there is a node partition into monotone paths taken from any given BFS such that contracting each one leads to a treewidth-8 graph.



Label $\lambda(u)$ consists of:

- the treewidth-8 scheme
- the depth of u in S (unless $u \in$B)
- 3 bits/path in the treewidth-8 scheme, i.e., 3x8 = 24 extra bits

[Extend to genus-g graphs]

Improved Bound: Dujmovic, Esperet, G., Joret, Micek, Morin

Theorem. The family of n-vertex subgraphs of H$\boxtimes$P has a labelling scheme with log$n$ + o(log$n$) bit labels, where H has bounded treewidth and P is a path.

$\Rightarrow$ **Induced-universal graph of $n^{1+o(1)}$ nodes for n-node planar graphs**

(and many other families of graphs)

Labelling the nodes takes O($n$log$n$) time

Decoding adjacency takes ~ √log$n$ time

Strong Product

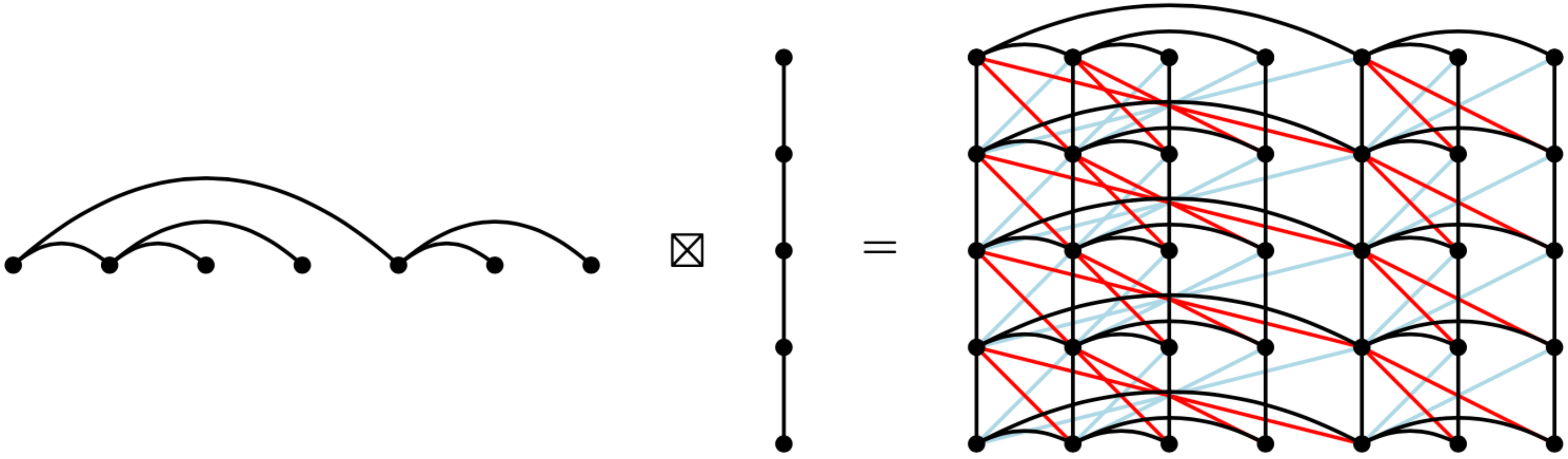


Figure 1: The strong product $H \boxtimes P$ of a tree H and a path P.

Special Case

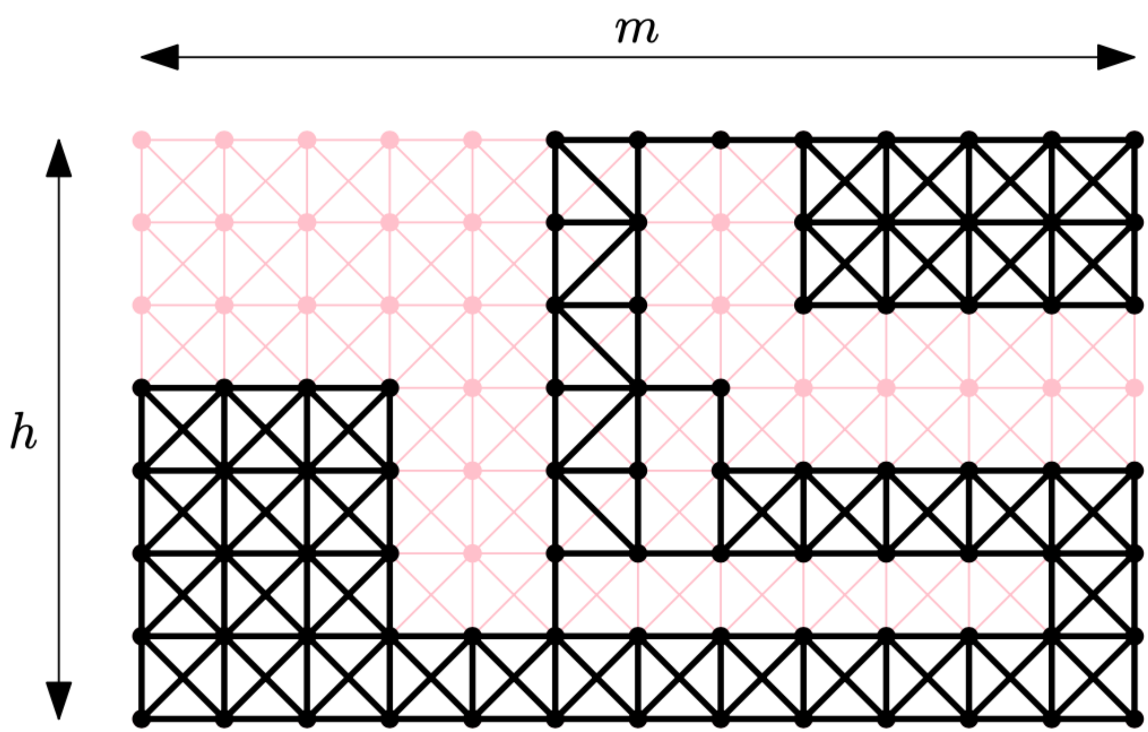
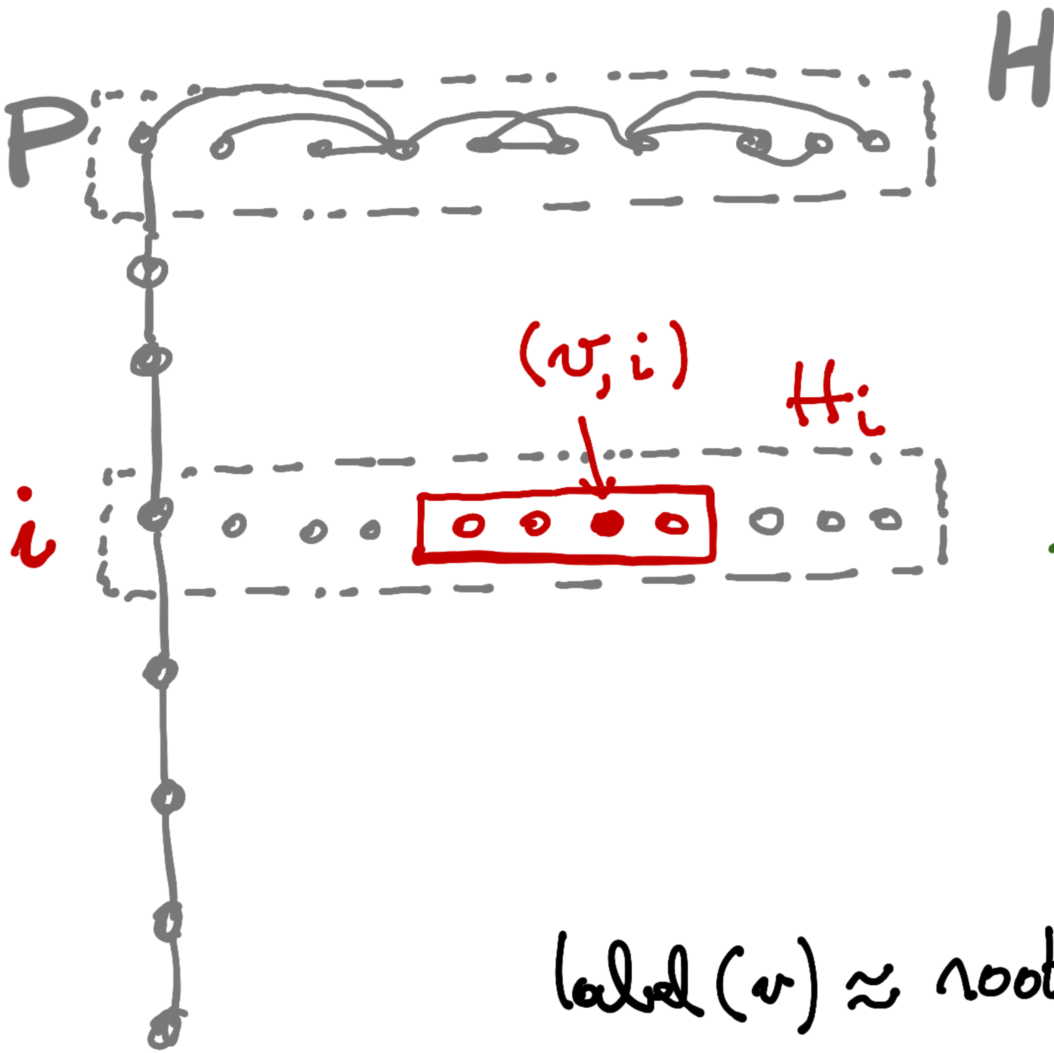
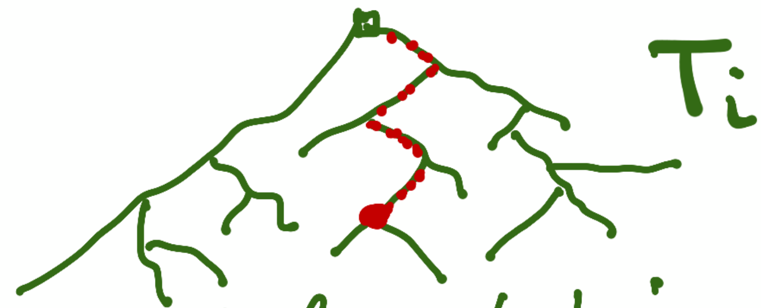


Figure 5: The special case where G is a subgraph of $P_1 \boxtimes P_2$.

1





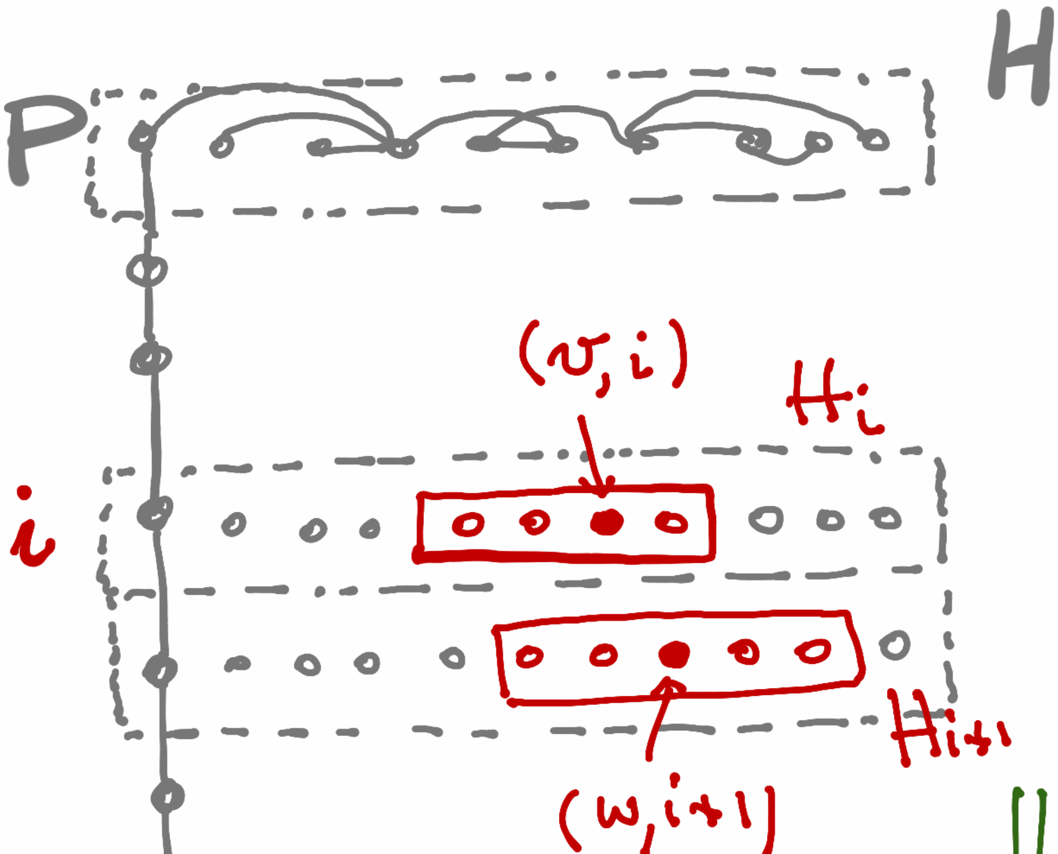
balanced binary search tree

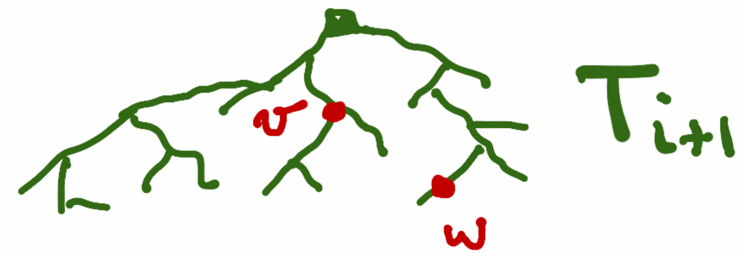
$v \to$ position in T_i

label(v) $\approx$ root-to-position(v) path

adj: $(v,i) - (v',i)$ ✓

2





T_{i+1} spans labels of H_{i+1} and H_i

PB: Need to know position of v in T_{i+1} from position of v in T_i ...

PB: Need to know position of v in T_{i+1} from position of v in T_i ...

Key obs.: T_{i+1} and T_i are not indep.

$\Rightarrow$ Given $o(\log n)$ bits we can extract position of v in T_{i+1} given its pos. in T_i.

Tools: $T_i \rightarrow T_{i+1}$

- bulk insertions
- bulk deletions
- rebalancing

Open Problems

1. Improve the 2nd order term, $\log n + O(1)$?
2. Extend to minor-free graphs
3. Improve to $\log n + \theta(k)$ for treewidth-k
4. Prove lower bounds for planar or minor-free

Best lower bound for planar: $\log n + \Omega(1)$ ☹

No families with $n!\,2^{O(n)}$ labelled graphs like trees, planar, bounded genus, bounded treewidth, minor-free (hereditary)… is known to require labels of $\log n + \omega(1)$ bits.


THANK YOU!
[0, 100) [100, 150)
logn + O(loglogn)
9th Slovenian Conference on
Graph Theory