

Universal Graph and Implicit Representation of Planar Graphs

Cyril Gavoille

LABORATOIRE
BORDELAIS
DE RECHERCHE
EN INFORMATIQUE

LaBRI - *University of Bordeaux*

Séminaire du LIGM
Université Paris-Est Marne-la-Vallée
Nov. 16, 2021

Induced-Universal Graphs

[Babai, Chung, Erdős, 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 .



Goal: to find such a universal graph for a given family F_n with few nodes (w.r.t. n)

Induced-Universal Graphs

[Babai, Chung, Erdős, 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 .



Goal: to find such a universal graph for a given family F_n with few nodes (w.r.t. n)

Induced-Universal Graphs

[Babai, Chung, Erdős, 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 .



Goal: to find such a universal graph for a given family F_n with few nodes (w.r.t. n)

Representation of a Graph



adjacency list

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

matrix

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	0	0
3	0	1	0	1	1
4	0	0	1	0	0
5	1	0	1	0	0

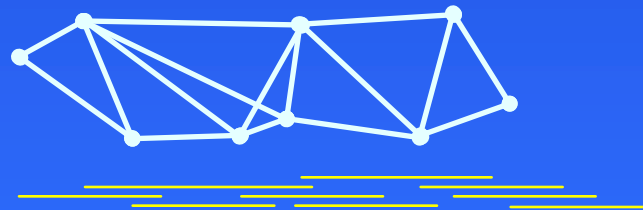
1 node = 1 pointer in the data-structure
(it does not carry any specific information)

Implicit Representation

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

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

Edges: $u-v \Leftrightarrow I(u) \cap I(v) \neq \emptyset$



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 (λ, f) such that $\forall G \in F, \forall u, v \in V(G)$:

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

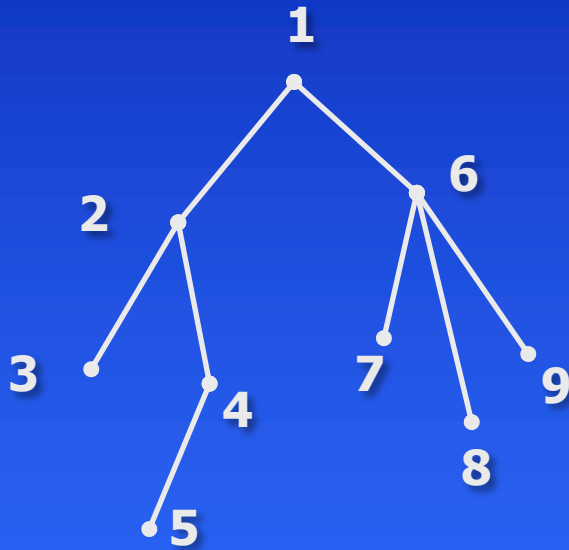
Goal: to minimize the maximum label size

In this talk: $P(u, v, G)$ is TRUE $\Leftrightarrow uv \in E(G)$

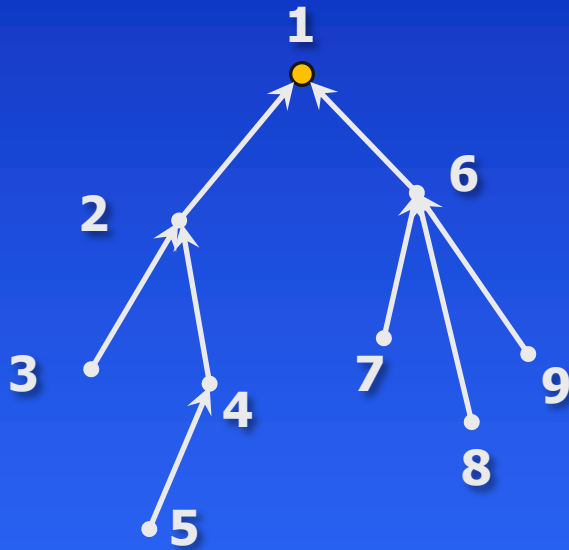
Basic Example: Trees



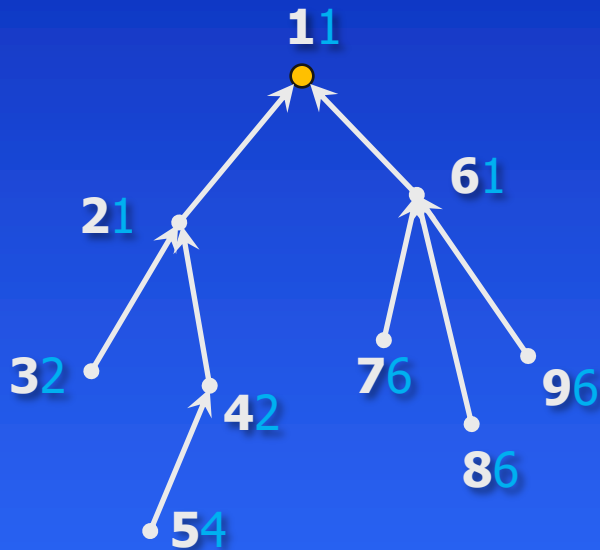
Basic Example: Trees



Basic Example: Trees



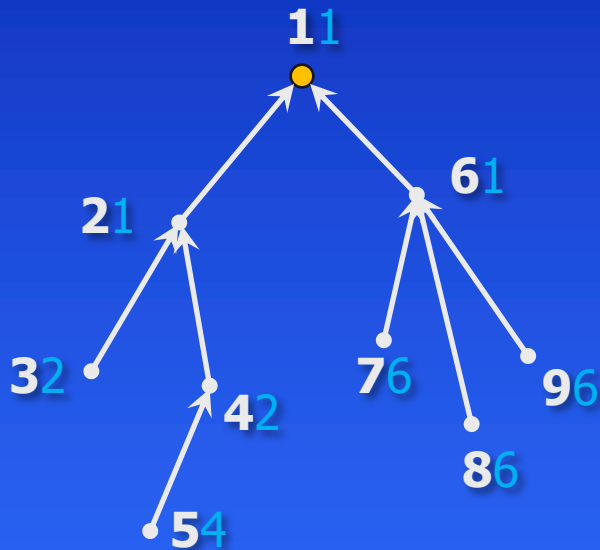
Basic Example: Trees



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

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

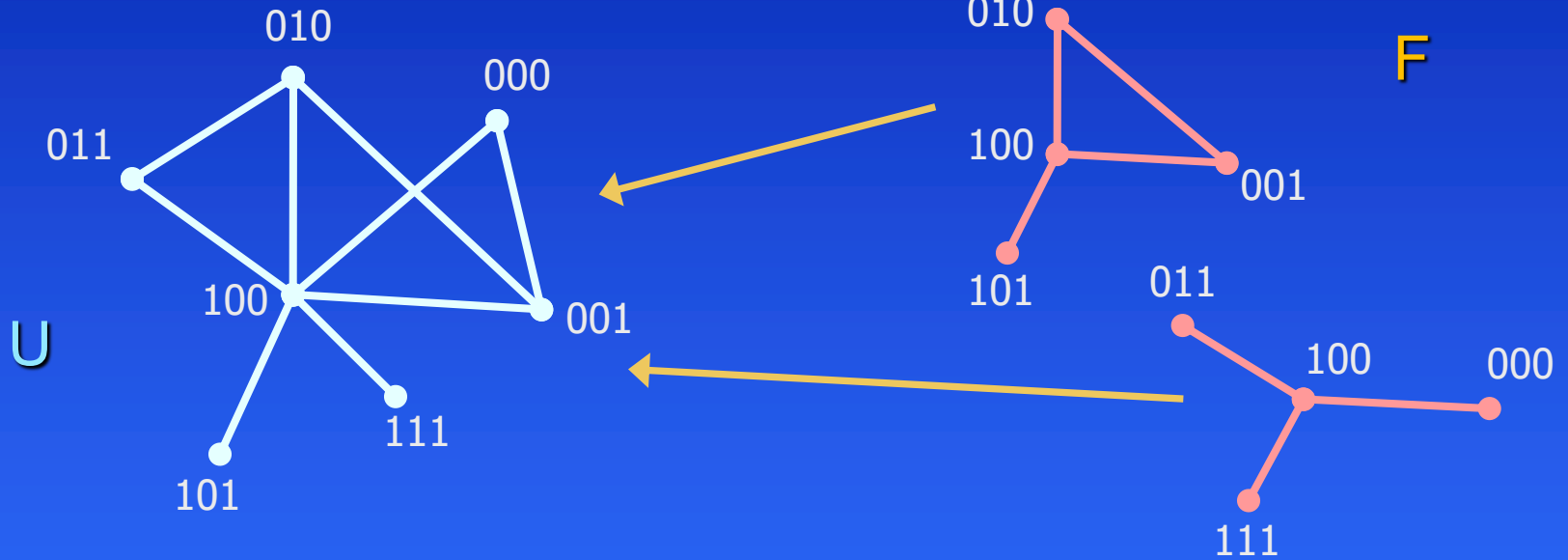
Basic Example: Trees



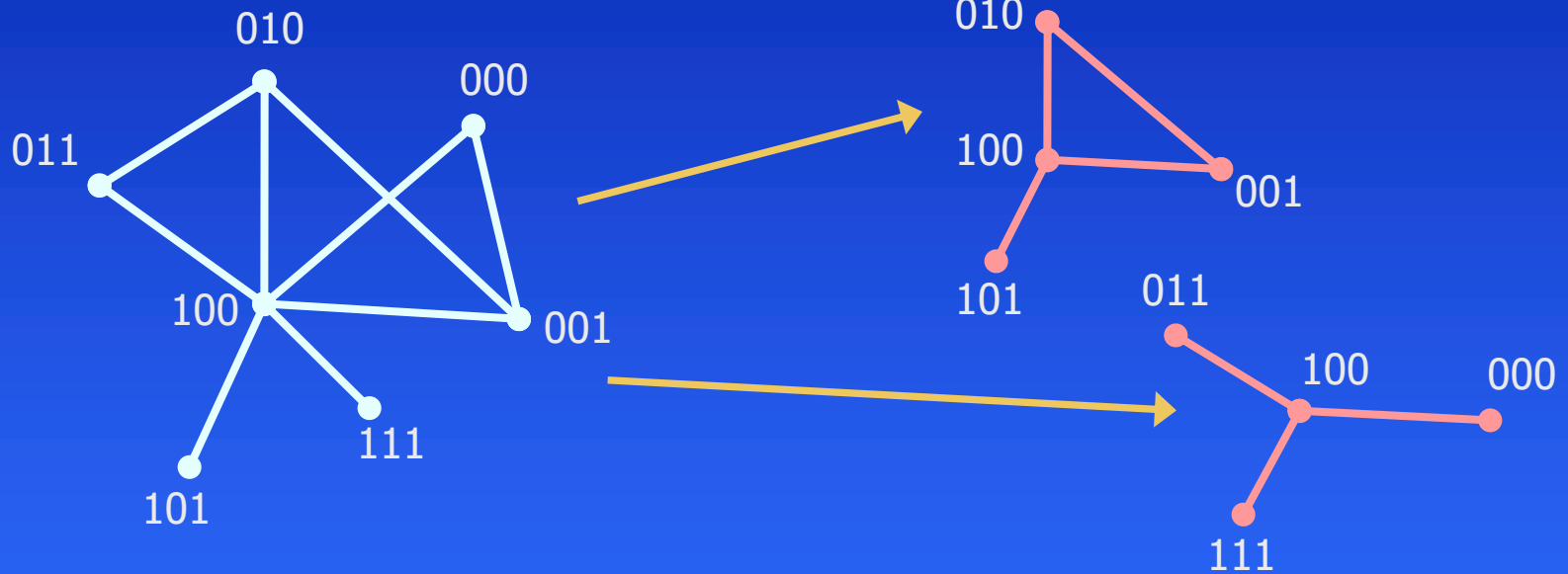
$$\lambda(u, T) = (u, \text{parent}(u)) \text{ or } (u, u)$$
$$f(uv, xy) = (v=x \text{ or } u=y)$$

For trees with n nodes: $\sim 2 \log n$ bits/node
(the constant does matter [Abiteboul et al. - SICOMP '06])

Universal & Label Graphs



Universal & Label Graphs



induced-universal graph U

graphs of F

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

Universal Graphs for Trees (for $n=6$ nodes)

From the $(u, \text{parent}(u))$ 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)

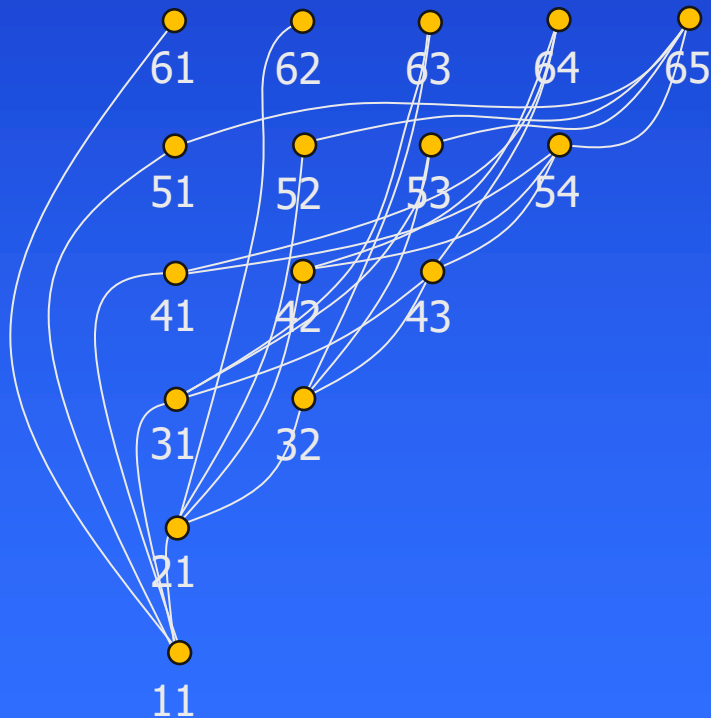
From the $(u, \text{parent}(u))$ labelling
 \Rightarrow induced-universal graph of $n^2=36$ nodes



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 (for $n=6$ nodes)

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

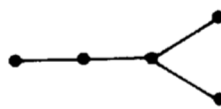


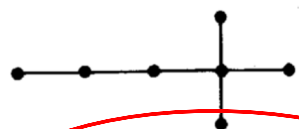
Using DFS for $T: (u, v)$
 $\Rightarrow u > v$ or $u = v = 1$
 $\Rightarrow n(n-1)/2 + 1 = 16$ nodes

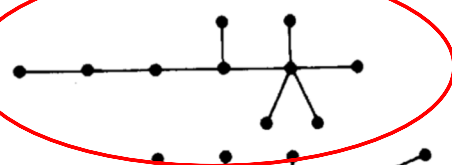
Universal Graphs for Trees (universal trees)

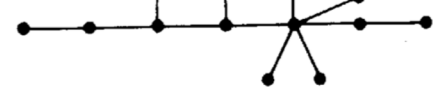
$n = 1$:  $n = \# \text{edges}$


$n = 2$: 

$n = 3$: 

$n = 4$: 

$n = 5$: 

$n = 6$: 

$n = 7$: 

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.

10 nodes

Universal Graphs for Trees (for $n=6$ nodes)



9 nodes is optimal because only
3 nodes of $K_{1,5}$ can be in P_6



Universal Graphs for Trees (for $n=6$ nodes)



9 nodes is optimal because only
3 nodes of $K_{1,5}$ can be in P_6

Universal Graphs for Trees (for $n=6$ nodes)



9 nodes is optimal because only
3 nodes of $K_{1,5}$ can be in P_6

Universal Graphs for Trees (for $n=6$ nodes)



9 nodes is optimal because only
3 nodes of $K_{1,5}$ can be in P_6

Universal Graphs for Trees (for $n=6$ nodes)



9 nodes is optimal because only
3 nodes of $K_{1,5}$ can be in P_6

Universal Graphs for Trees (for $n=6$ nodes)



9 nodes is optimal because only
3 nodes of $K_{1,5}$ can be in P_6

Universal Graphs for Trees (for $n=6$ nodes)



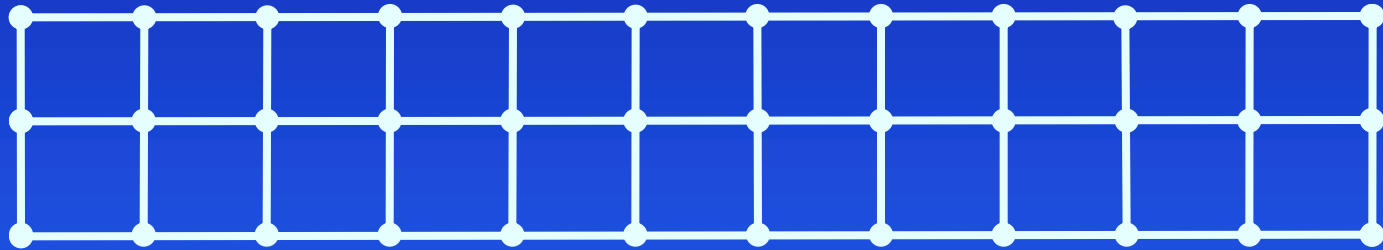
9 nodes is optimal because only
3 nodes of $K_{1,5}$ can be in P_6

Exercise: Degree at most 2

How to get $\log n + o(1)$ bit labels?

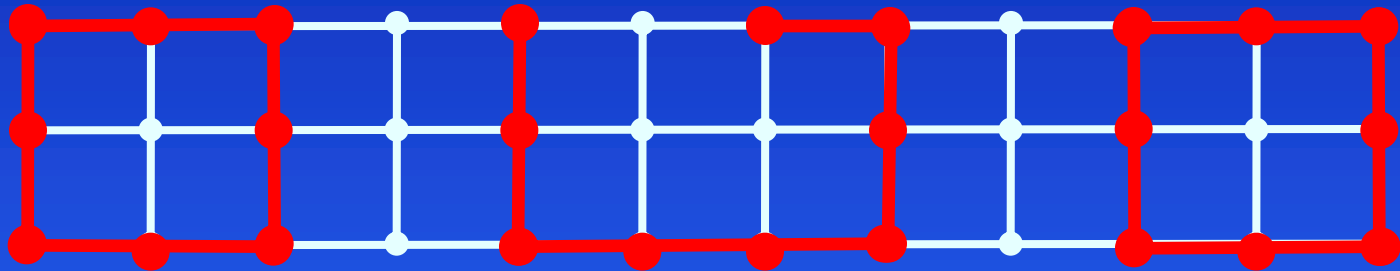


Exercise: Degree at most 2
How to get $\log n + o(1)$ bit labels?



Exercise: Degree at most 2

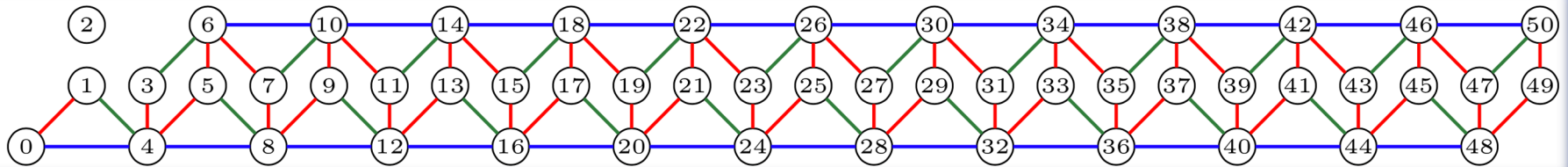
How to get $\log n + o(1)$ bit labels?



$3n$ nodes $\Rightarrow \log n + o(1)$ bits

Exercise: Degree at most 2

How to get $\log n + o(1)$ bit labels?



$2n$ nodes $\Rightarrow \log n + 1$ bits

(Lower Bound: $11n/6$ nodes)

Exercise II: Caterpillar

How to get $\log n + o(1)$ bit labels?

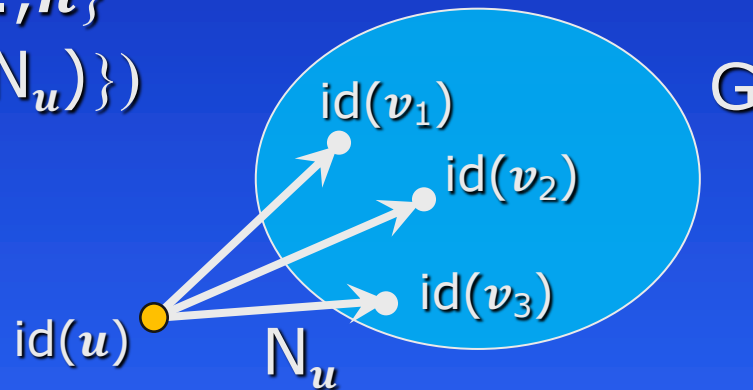


(Lower Bound: $3n/2$ nodes)

Labelling Schemes for Planar Graphs

Basic: Truncated List

$$u \mapsto \text{id}(u) \in \{1, \dots, n\}$$
$$\lambda(u) = (\text{id}(u), \{\text{id}(N_u)\})$$



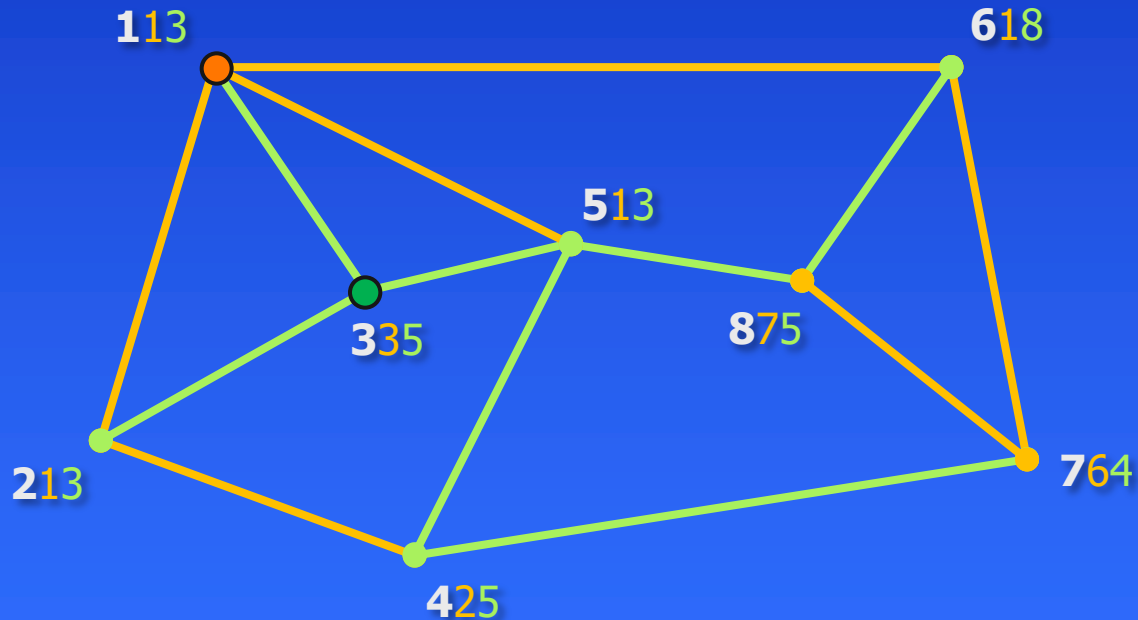
$$u-v \Leftrightarrow \text{id}(v) \in \{\text{id}(N_u)\} \text{ or } \text{id}(u) \in \{\text{id}(N_v)\}$$

$\Rightarrow (k+1)\log n$ bits if G is k -degenerated

$\Rightarrow 6\log n$ bits for planar [Muller'88]

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]

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 & J.ACM'17]

⇒ Arboricity- k : $k \log n + o(1)$

⇒ Planar: **3** $\log n + o(1)$

For treewidth- k : $\log n + o(k \log \log n)$

[G.,Labourel – ESA'07]

⇒ Planar & Minor-free: **2** $\log n + o(\log \log n)$

Better: Bonamy, G., Pilipczuk – SODA'20

For planar & bounded genus: $\frac{4}{3} \log n + O(\log \log n)$

⇒

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

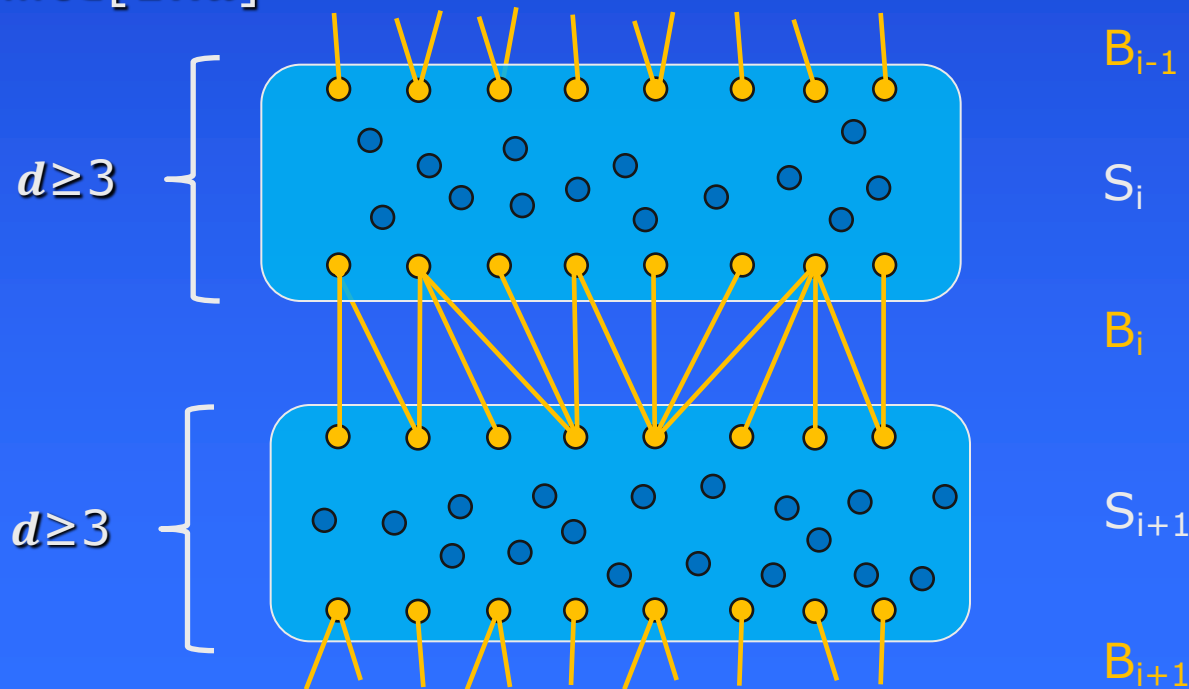
Sketch of Proof (1/2)

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

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

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

BFS & $\text{shift} \in [1..d]$



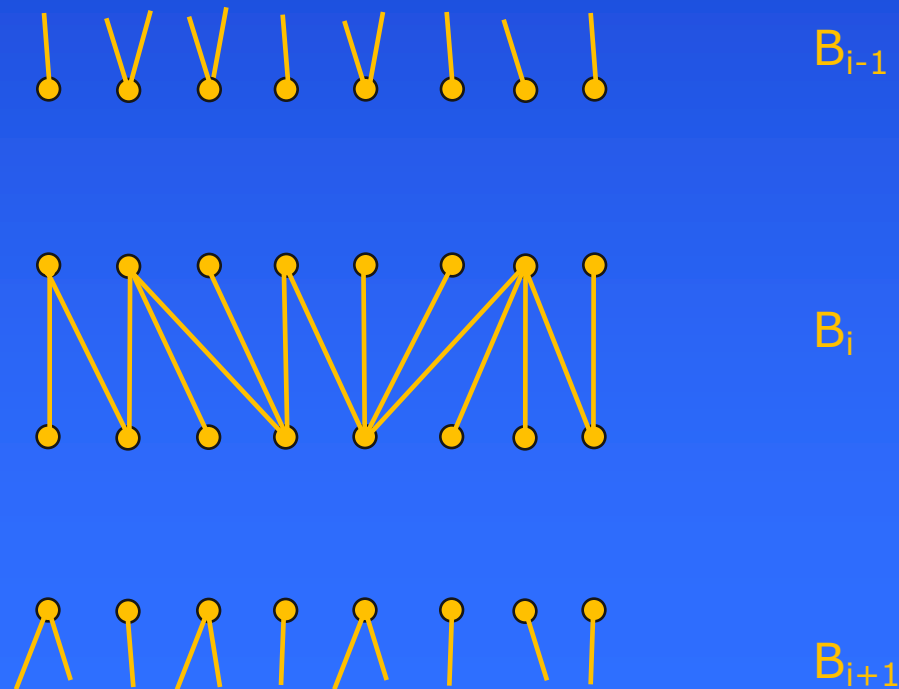
Sketch of Proof (1/2)

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

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

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

BFS & $\text{shift} \in [1..d]$



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 labellings

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

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

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

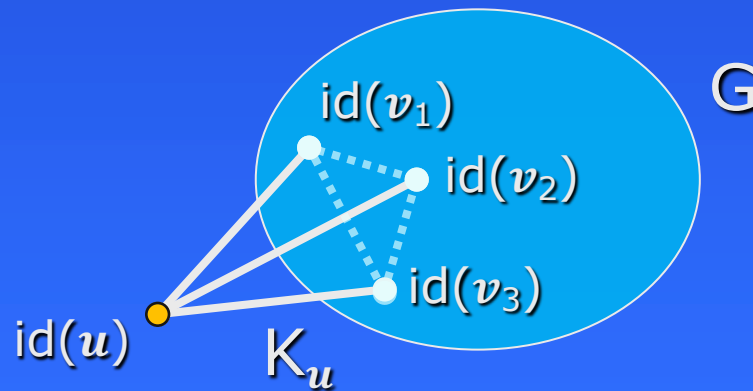
\Rightarrow nodes in $V(B)$: $\log(n/d) + \log(n/d) = \frac{4}{3} \log n$ 😊

Improved Scheme for Treewidth- k

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

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

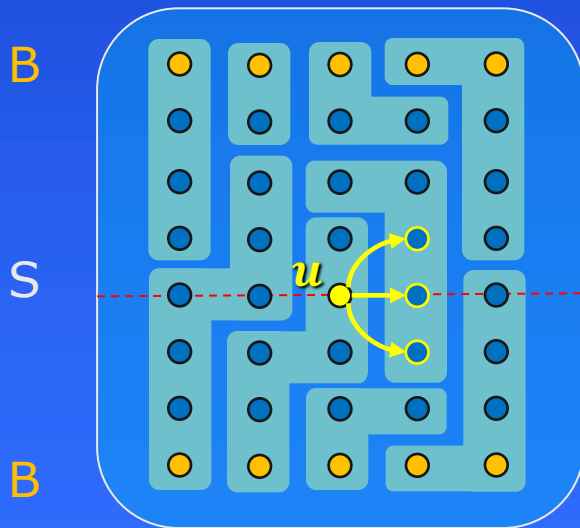
$$|\lambda(u)| = \log |V_i| + O(k \log \log |V(G)|).$$



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

Labeling Scheme for S

Key lemma. [2018,2019,2021] 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-6 graph.



Label $\lambda(u)$ consists of:

- the treewidth-6 scheme
- the depth of u in S (unless $u \in B$)
- 3 bits/path in the treewidth-6 scheme, i.e., $3 \times 6 = 18$ extra bits

[Extend to genus- g graphs]

Improved Bound: Dujmovic, Esperet, G., Joret, Micek, Morin – J.ACM'21

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

⇒

**Induced-universal graph of $n^{1+o(1)}$
nodes for n -node planar graphs
(and many other graph families)**

Labelling the nodes takes $O(n \log n)$ time
Decoding adjacency takes $\sim \sqrt{\log n}$ time

Strong Product

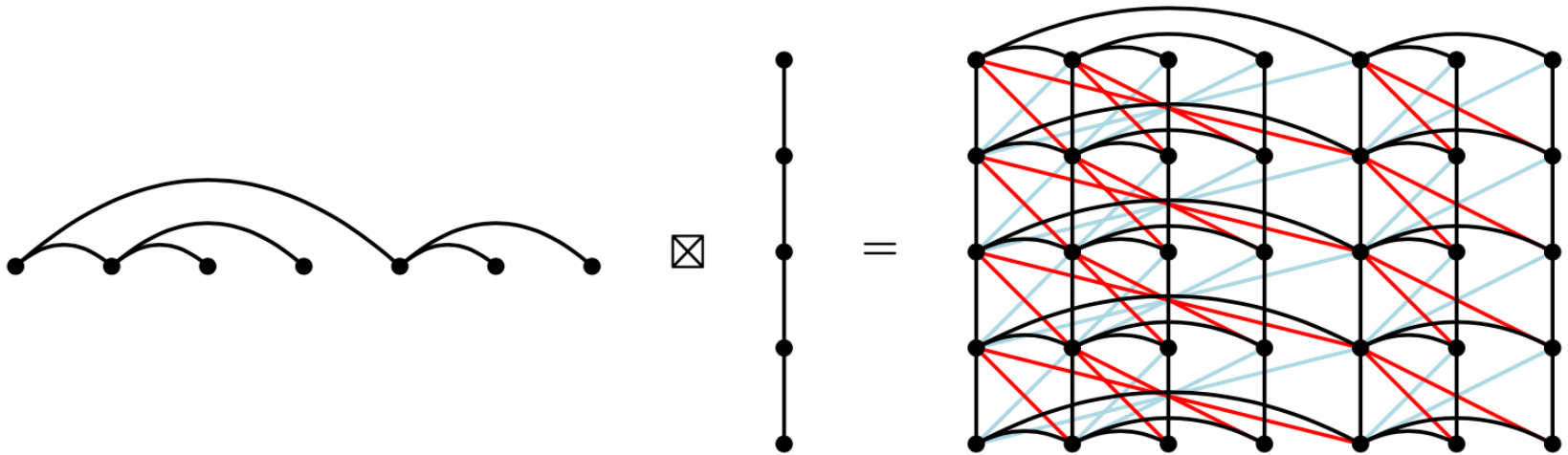


Figure 1: The strong product $H \boxtimes P$ of a tree H and a path P .

Special Case

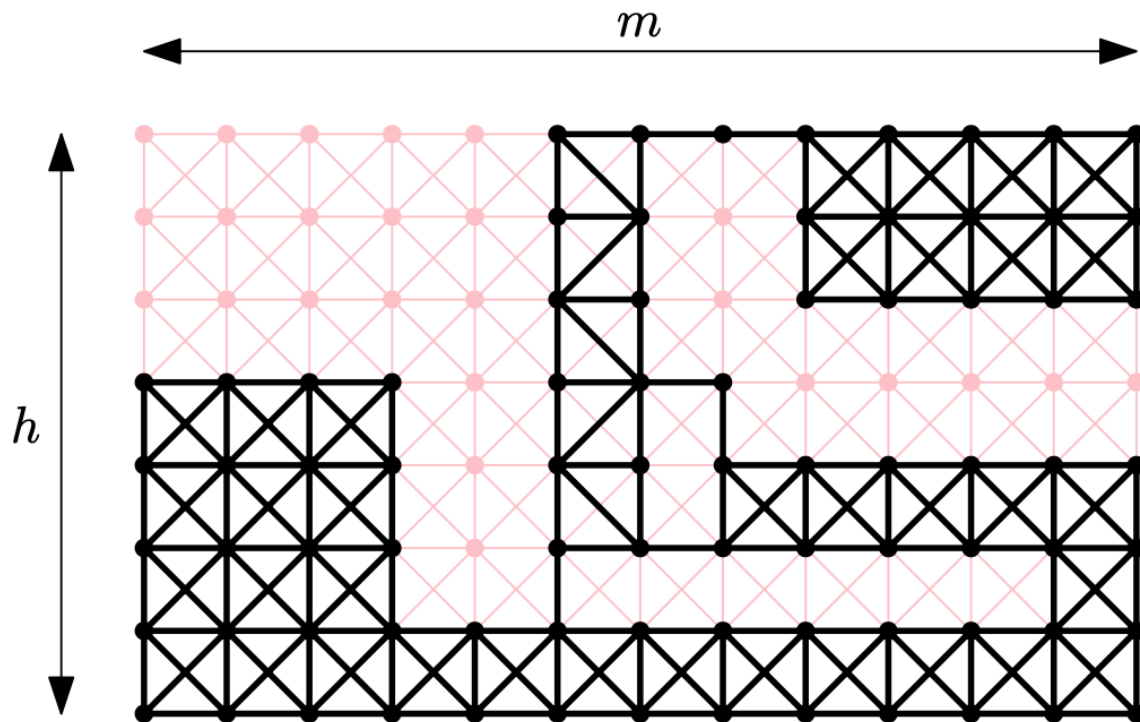


Figure 5: The special case where G is a subgraph of $P_1 \boxtimes P_2$.

Other Recent Results

1. Universal graph of $8n$ nodes for caterpillars
2. $\log n + \theta(p)$ bits for pathwidth- p
3. $n^{1/2+o(1)}$ bits for graphs of VC-dimension 2

\Rightarrow

At most $\exp(n^{3/2})$ such graphs, whereas only $\exp(n^{5/3})$ was previously known, $\exp(n^{2-1/(d+1)})$ for VC-dim. d by [Alon, Bollobás et al.'11]

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
5. Prove the “Implicit Graph Conjecture”

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, ... is known to require labels of $\log n + \omega(1)$ bits.