

# Smaller Universal Graphs for Caterpillars and Graphs of Bounded Path-Width

Cyril Gavoille and Arnaud Labourel

université  
de BORDEAUX

LaBRI

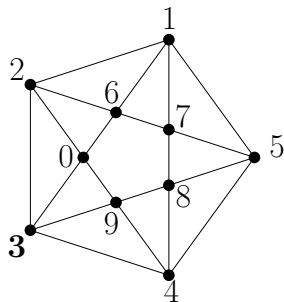
Aix-Marseille  
université  
*Socialement engagée*

L i S

LABORATOIRE  
D'INFORMATIQUE  
& SYSTÈMES

ICGT '22 – Montpellier – July 5th, 2022

# Representation of Graphs



adjacency lists

0	2	3	6	9
1	2	5	6	7
2	0	1	3	6
<b>3</b>	0	2	4	9
4	3	5	8	9
5	1	4	7	8
6	0	1	2	7
7	1	5	6	8
8	4	5	7	9
9	0	3	4	8

adjacency matrix

	0	1	2	<b>3</b>	4	5	6	7	8	9
0	0	0	1	1	0	0	1	0	0	1
1	0	0	1	0	0	1	1	1	0	0
2	1	1	0	1	0	0	1	0	0	0
<b>3</b>	1	0	1	0	1	0	0	0	0	1
4	0	0	0	1	0	1	0	0	1	1
5	0	1	0	0	1	0	0	1	1	0
6	1	1	1	0	0	0	0	1	0	0
7	0	1	0	0	0	1	1	0	1	0
8	0	0	0	0	1	1	0	1	0	1
9	1	0	0	1	1	0	0	0	1	0

1 node = 1 pointer in the data-structure

# Implicit Representation

**Idea:** Associate with each node some information and only use the information of any two nodes to determine if they are adjacent.

An **adjacency labeling scheme** for a graph family  $\mathcal{F}$  pair of functions  $(\ell, f)$  such that  $\forall G \in \mathcal{F}, \forall u, v \in V(G)$

- Encoder:  $\ell(u, G)$  is a label (binary string)
- Decoder:  $f(\ell(u, G), \ell(v, G))$  is true  $\Leftrightarrow uv \in E(G)$

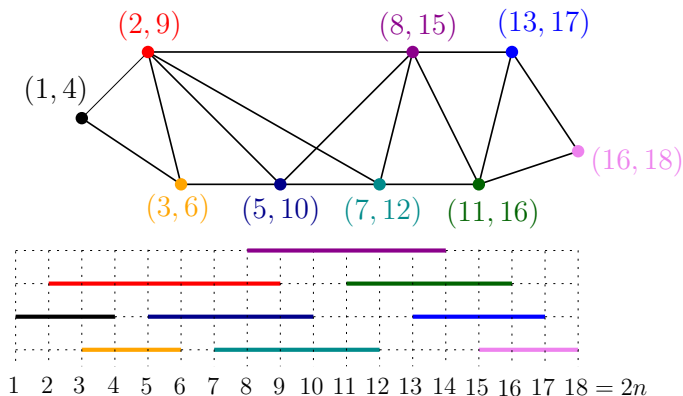
**Goal:** minimize label size (in bits)

... and time complexities of  $\ell, f$

# Ex: Interval Graphs

$\ell(u, G) := (l_u, r_u)$  with  $l_u, r_u \in \{1, \dots, 2n\}$

$f((l_u, r_u), (l_v, r_v)) := \text{true} \Leftrightarrow [l_u, r_u] \cap [l_v, r_v] \neq \emptyset$



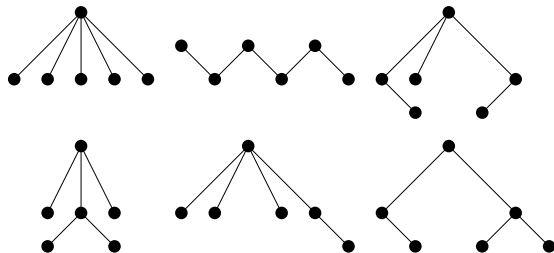
$\Rightarrow$  labels of  $2 \lceil \log(2n) \rceil$  bits

# Universal Graph

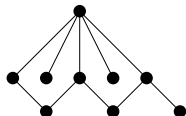
Babai, Chung, Erdős, Graham, Spencer 1982

A graph  $U$  is an **induced-universal graph** for  $\mathcal{F}$  if every graph of  $\mathcal{F}$  is isomorphic to some induced subgraph of  $U$ .

$\mathcal{F} = \{\text{trees with 6 nodes}\}$

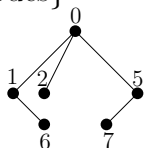
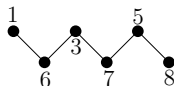
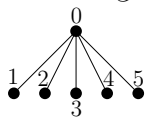


universal graph for  $\mathcal{F}$

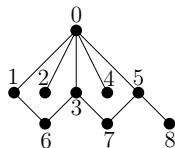
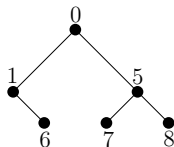
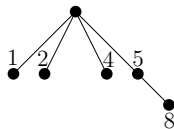
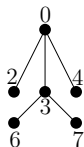


# Universal Graphs / Labeling Schemes

$\mathcal{F} = \{\text{trees with 6 nodes}\}$



universal graph for  $\mathcal{F}$



$k$ -bit labels  $\Leftrightarrow$  universal graph of  $2^k$  nodes

# Results

family	label size	U. graph size	refs.
treewidth $k$	$\log n + O(k \lg \lg n)$	$n \cdot (\log n)^{O(k)}$	[GL07]
planar, $H \boxtimes P$	$\log n + \tilde{O}(\sqrt{\lg n})$	$n^{1+o(1)}$	[DEGJx21]

# Results

family	label size	U. graph size	refs.
treewidth $k$	$\log n + O(k \lg \lg n)$	$n \cdot (\log n)^{O(k)}$	[GL07]
planar, $H \boxtimes P$	$\log n + \tilde{O}(\sqrt{\lg n})$	$n^{1+o(1)}$	[DEGJx21]
outerplanar $\Delta$	$\log n + O(1)$	$O(n)$	[C90][AR14]
trees	$\log n + O(1)$	$O(n)$	[ADK17]
$\Delta \leq 2$	$\log n + O(1)$	$2n$	[AAHKS20]



# Results

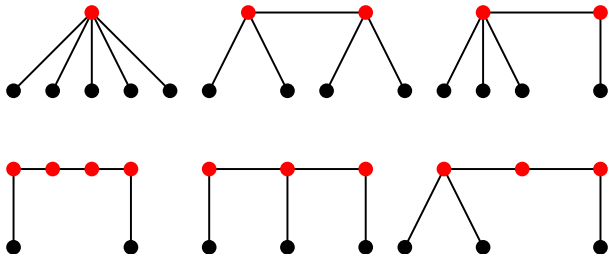
family	label size	U. graph size	refs.
treewidth $k$	$\log n + O(k \lg \lg n)$	$n \cdot (\log n)^{O(k)}$	[GL07]
planar, $H \boxtimes P$	$\log n + \tilde{O}(\sqrt{\lg n})$	$n^{1+o(1)}$	[DEGJx21]
outerplanar $\Delta$	$\log n + O(1)$	$O(n)$	[C90][AR14]
trees	$\log n + O(1)$	$O(n)$	[ADK17]
$\Delta \leq 2$	$\log n + O(1)$	$2n$	[AAHKS20]
caterpillars	$\log n + O(1)$	$256n$	[BGL06]
caterpillars	$\log n + O(1)$	$384n$	[ADK17]

# Results

family	label size	U. graph size	refs.
treewidth $k$	$\log n + O(k \lg \lg n)$	$n \cdot (\log n)^{O(k)}$	[GL07]
planar, $H \boxtimes P$	$\log n + \tilde{O}(\sqrt{\lg n})$	$n^{1+o(1)}$	[DEGJx21]
outerplanar $\Delta$	$\log n + O(1)$	$O(n)$	[C90][AR14]
trees	$\log n + O(1)$	$O(n)$	[ADK17]
$\Delta \leq 2$	$\log n + O(1)$	$2n$	[AAHKS20]
caterpillars	$\log n + O(1)$	$256n$	[BGL06]
caterpillars	$\log n + O(1)$	$384n$	[ADK17]
caterpillars	$\log n + O(1)$	$8n$	new
bounded pw	$\log n + O(1)$	$O(n)$	new

# I. Universal Graph for Caterpillars

A **caterpillar forest** is a forest in which the nodes of degree at least two induce paths.

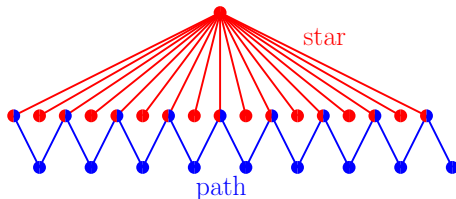


## Theorem 1

Caterpillar forests with  $n$  nodes have a universal graph  $\mathcal{U}_n$  with  $8n$  nodes.

# Lower Bound (Folk)

Every universal graph for the star and the path with  $n$  nodes requires  $\lfloor 3n/2 \rfloor$  nodes.



# Universal Graph $\mathcal{U}_n$ for Caterpillar

Nodes:

- independent set  $S$  of  $6n$  nodes
- layers  $L_i$  of  $n/2^i$  nodes for  $i$  from 0 to  $\lceil \log(n/6) \rceil$

$\Rightarrow 6n + \sum_{i \geq 0} n/2^i < 8n$  nodes in total

Each node  $u$  of layer  $L_i$  is adjacent to:

- an interval  $I(u)$  of  $6 \cdot 2^i$  nodes of  $S$
- its predecessor and successor in  $L_i$  and all nodes vertically below

$S$  .....

# Universal Graph $\mathcal{U}_n$ for Caterpillar

Nodes:

- independent set  $S$  of  $6n$  nodes
- layers  $L_i$  of  $n/2^i$  nodes for  $i$  from 0 to  $\lceil \log(n/6) \rceil$

$\Rightarrow 6n + \sum_{i \geq 0} n/2^i < 8n$  nodes in total

Each node  $u$  of layer  $L_i$  is adjacent to:

- an interval  $I(u)$  of  $6 \cdot 2^i$  nodes of  $S$
- its predecessor and successor in  $L_i$  and all nodes vertically below



# Universal Graph $\mathcal{U}_n$ for Caterpillar

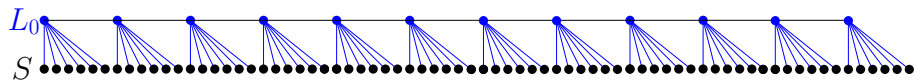
Nodes:

- independent set  $S$  of  $6n$  nodes
- layers  $L_i$  of  $n/2^i$  nodes for  $i$  from 0 to  $\lceil \log(n/6) \rceil$

$\Rightarrow 6n + \sum_{i \geq 0} n/2^i < 8n$  nodes in total

Each node  $u$  of layer  $L_i$  is adjacent to:

- an interval  $I(u)$  of  $6 \cdot 2^i$  nodes of  $S$
- its predecessor and successor in  $L_i$  and all nodes vertically below



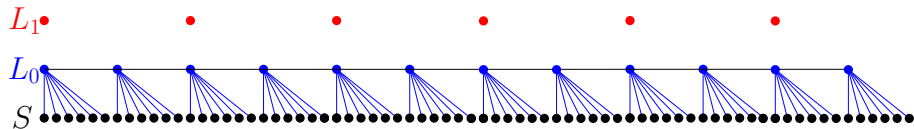
# Universal Graph $\mathcal{U}_n$ for Caterpillar

Nodes:

- independent set  $S$  of  $6n$  nodes
  - layers  $L_i$  of  $n/2^i$  nodes for  $i$  from 0 to  $\lceil \log(n/6) \rceil$
- $\Rightarrow 6n + \sum_{i \geq 0} n/2^i < 8n$  nodes in total

Each node  $u$  of layer  $L_i$  is adjacent to:

- an interval  $I(u)$  of  $6 \cdot 2^i$  nodes of  $S$
- its predecessor and successor in  $L_i$  and all nodes vertically below





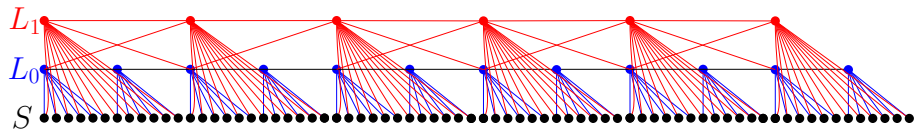
# Universal Graph $\mathcal{U}_n$ for Caterpillar

Nodes:

- independent set  $S$  of  $6n$  nodes
  - layers  $L_i$  of  $n/2^i$  nodes for  $i$  from 0 to  $\lceil \log(n/6) \rceil$
- $\Rightarrow 6n + \sum_{i \geq 0} n/2^i < 8n$  nodes in total

Each node  $u$  of layer  $L_i$  is adjacent to:

- an interval  $I(u)$  of  $6 \cdot 2^i$  nodes of  $S$
- its predecessor and successor in  $L_i$  and all nodes vertically below



# Universal Graph $\mathcal{U}_n$ for Caterpillar

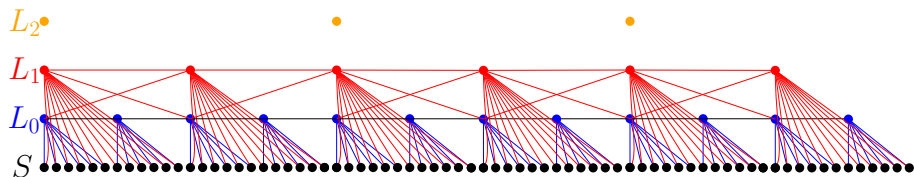
Nodes:

- independent set  $S$  of  $6n$  nodes
- layers  $L_i$  of  $n/2^i$  nodes for  $i$  from 0 to  $\lceil \log(n/6) \rceil$

$\Rightarrow 6n + \sum_{i \geq 0} n/2^i < 8n$  nodes in total

Each node  $u$  of layer  $L_i$  is adjacent to:

- an interval  $I(u)$  of  $6 \cdot 2^i$  nodes of  $S$
- its predecessor and successor in  $L_i$  and all nodes vertically below



# Universal Graph $\mathcal{U}_n$ for Caterpillar

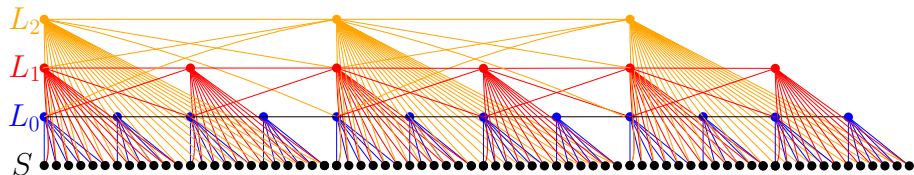
Nodes:

- independent set  $S$  of  $6n$  nodes
- layers  $L_i$  of  $n/2^i$  nodes for  $i$  from 0 to  $\lceil \log(n/6) \rceil$

$\Rightarrow 6n + \sum_{i \geq 0} n/2^i < 8n$  nodes in total

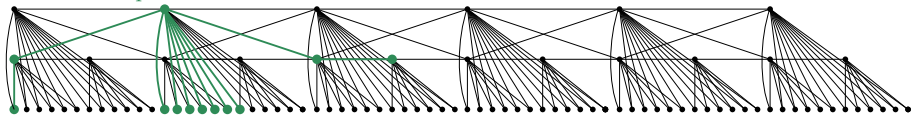
Each node  $u$  of layer  $L_i$  is adjacent to:

- an interval  $I(u)$  of  $6 \cdot 2^i$  nodes of  $S$
- its predecessor and successor in  $L_i$  and all nodes vertically below

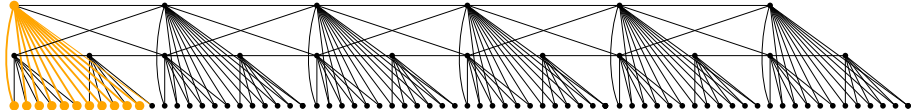


# Embedding Caterpillars Forests in $\mathcal{U}_{12}$

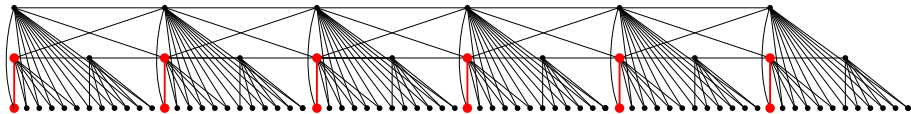
Some caterpillar of 12 nodes



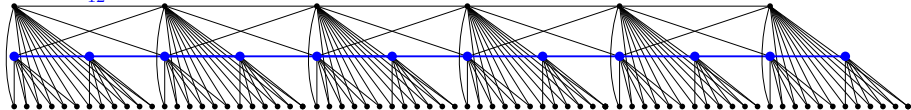
Star of 12 nodes



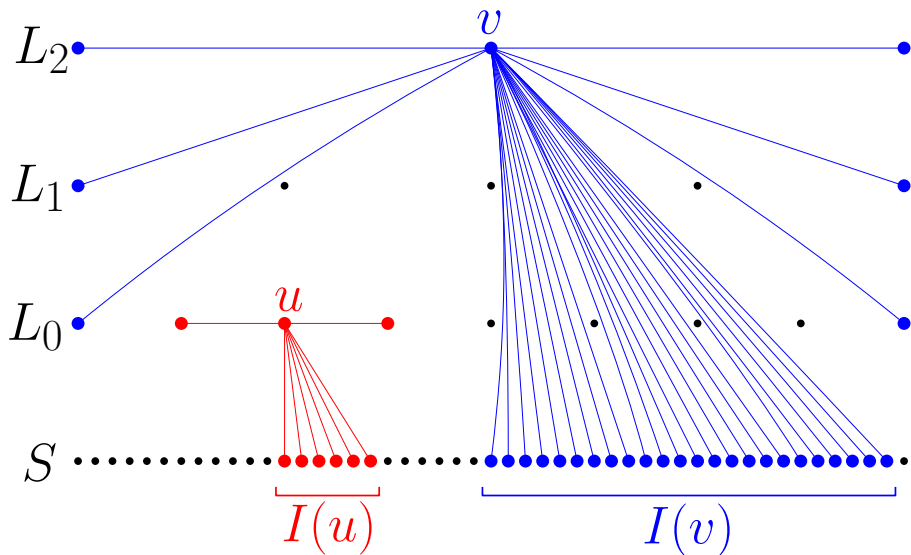
Matching of 6 edges



Path  $P_{12}$  of 12 nodes



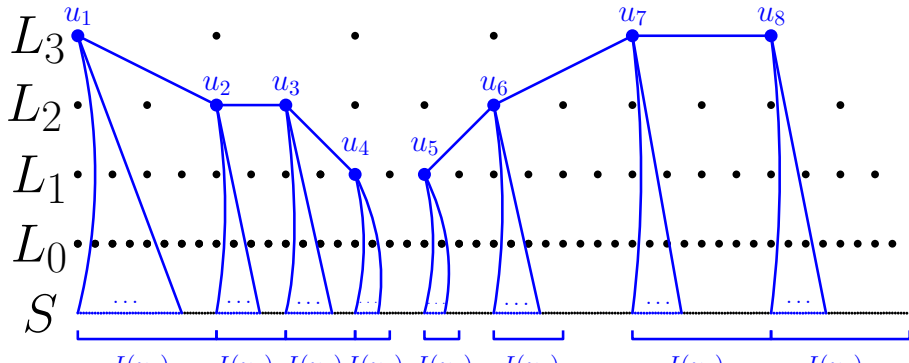
# Adjacency in $\mathcal{U}_n$



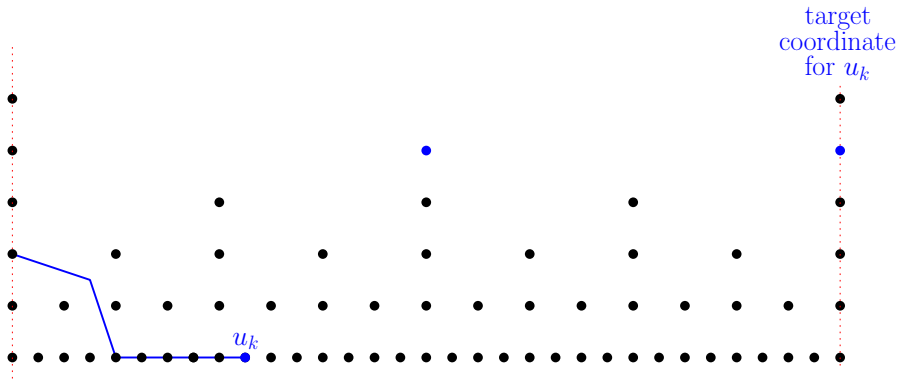
# Embedding in $\mathcal{U}_n$

$\delta(u) :=$  number of leaves adjacent to internal node  $u$

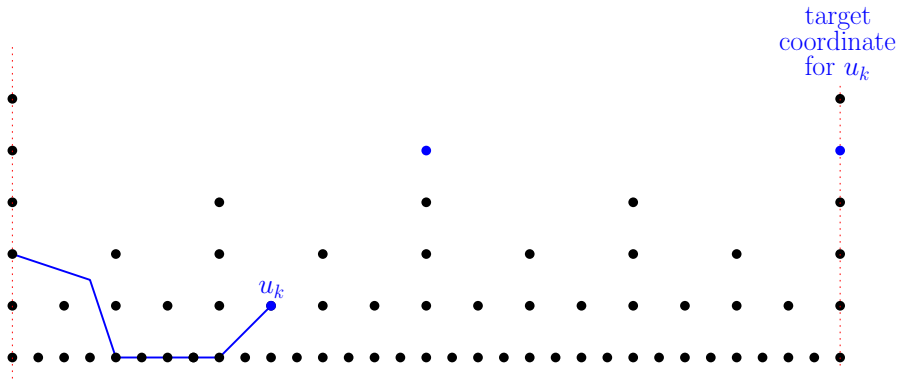
**Basic idea:** Embed each internal node  $u_k$  in  $u \in V(\mathcal{U}_n)$  such that  $I(u)$  contains at least  $\delta(u)$  nodes.



# How to embed $u_k$ ?

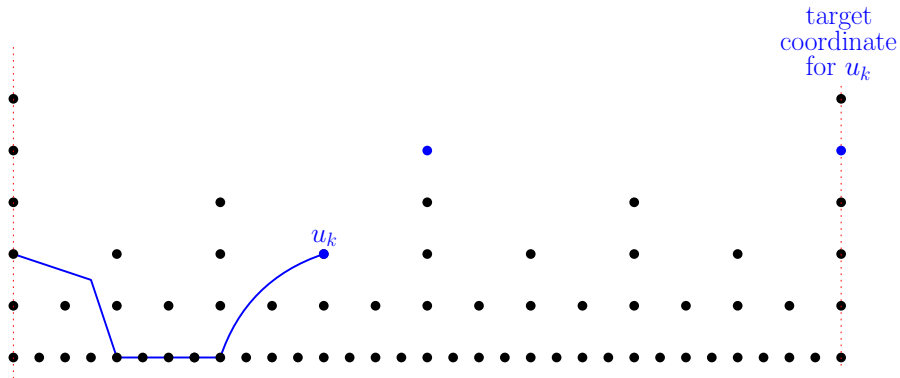


# How to embed $u_k$ ?

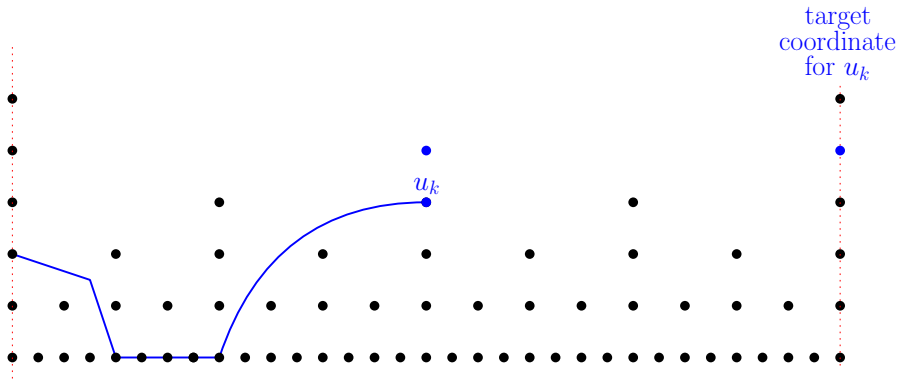




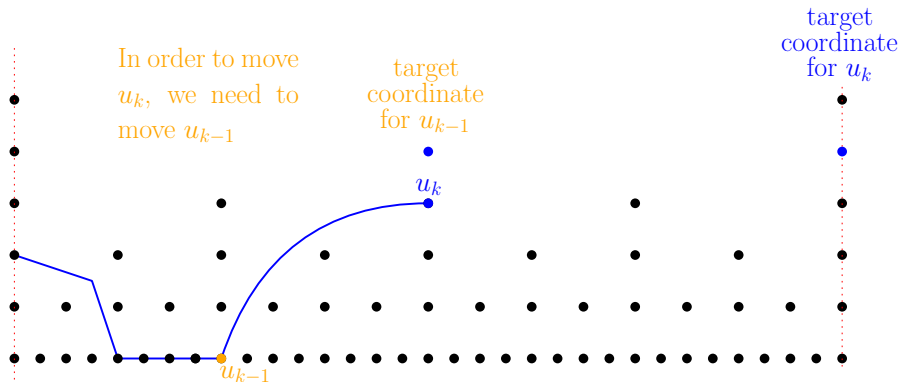
# How to embed $u_k$ ?



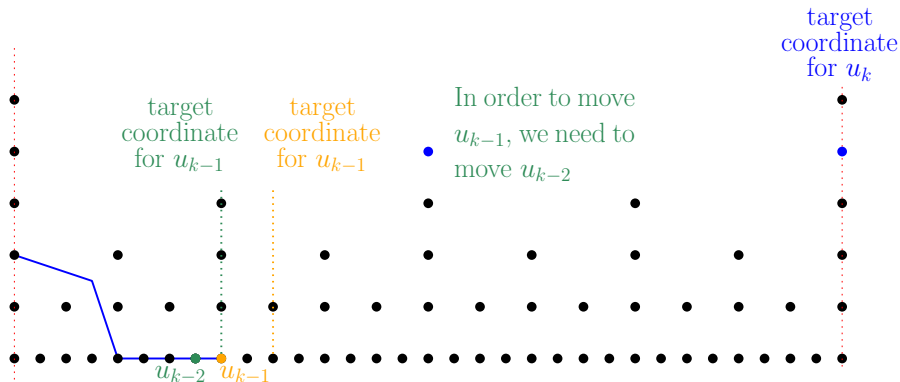
# How to embed $u_k$ ?



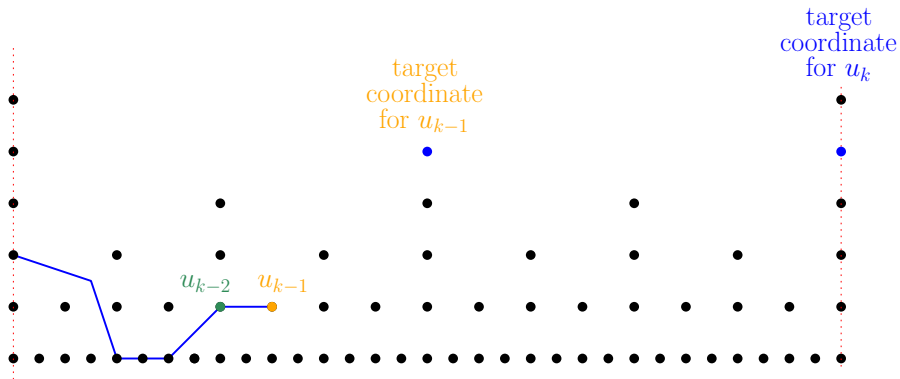
# How to embed $u_k$ ?



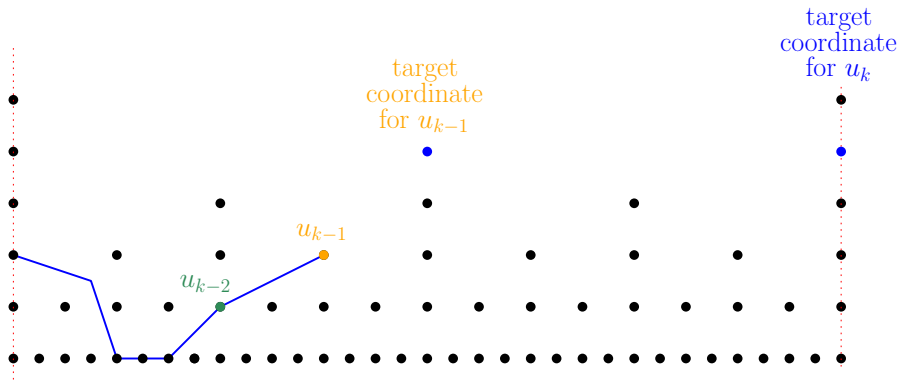
# How to embed $u_k$ ?



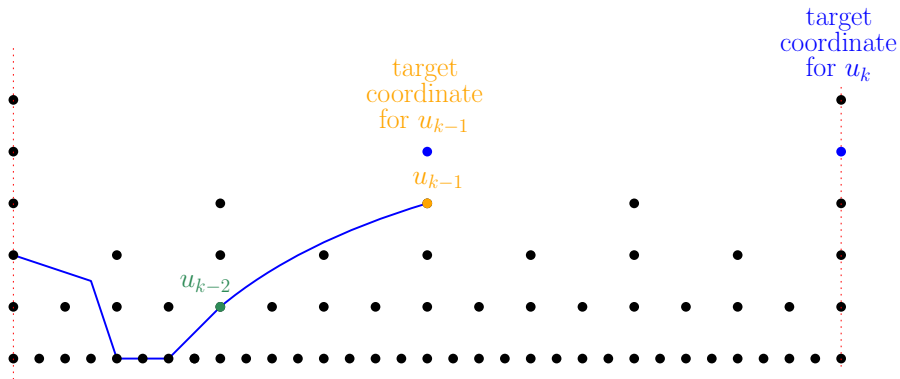
# How to embed $u_k$ ?



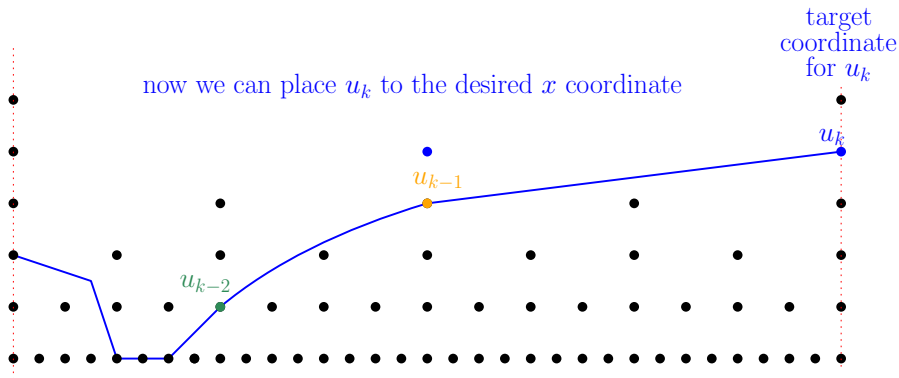
# How to embed $u_k$ ?



# How to embed $u_k$ ?



# How to embed $u_k$ ?

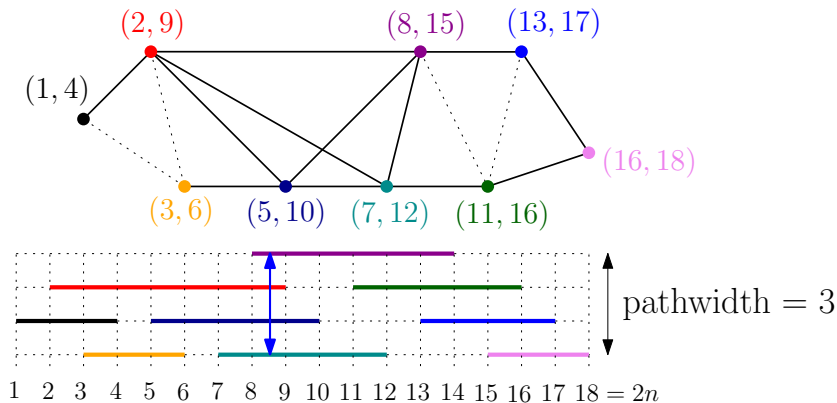




## II. Graphs of Pathwidth $p$

The **pathwidth** of  $G$  is the maximum clique size minus one of an interval graph containing  $G$  as subgraph.

( $p = 1 \Leftrightarrow G$  is a caterpillar forest)



# Adjacency Labeling Scheme

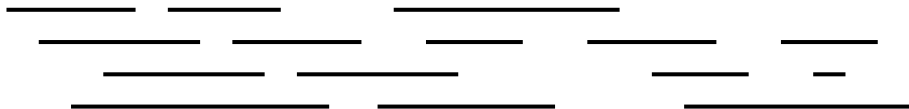
## Theorem 2

There is an adjacency labeling scheme for graphs of  $n$  nodes and pathwidth  $p$  with  $\log n + O(p)$ -bit labels.  
 $\Rightarrow$  universal graph of  $n \cdot 2^{O(p)}$  nodes.

- previous upper bound:  $\log n + O(p \log \log n)$  bits
- matching lower bound:  $\log n + \Omega(p)$  bits  
[  $\geq \frac{1}{n} \log(\mathcal{L}_{n,p})$  bits where  $\mathcal{L}_{n,p} = n! \cdot 2^{\Theta(np)}$  is the number of pathwidth- $p$  labeled graphs with  $n$  nodes ]

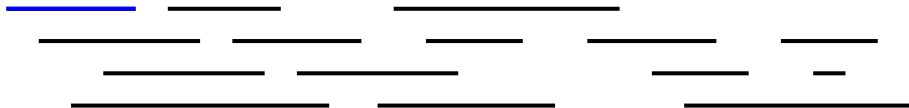
# Dominating Path Decomposition

- start with the interval with the leftmost left end
- at each step, take as the next interval the neighbor with the rightmost right end



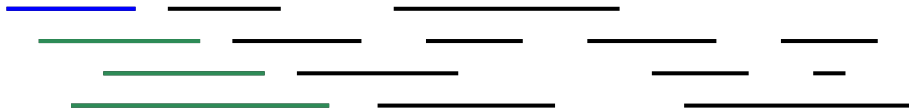
# Dominating Path Decomposition

- start with the interval with the leftmost left end
- at each step, take as the next interval the neighbor with the rightmost right end



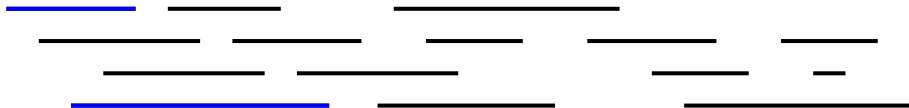
# Dominating Path Decomposition

- start with the interval with the leftmost left end
- at each step, take as the next interval the neighbor with the rightmost right end



# Dominating Path Decomposition

- start with the interval with the leftmost left end
- at each step, take as the next interval the neighbor with the rightmost right end



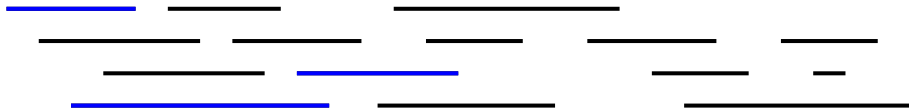
# Dominating Path Decomposition

- start with the interval with the leftmost left end
- at each step, take as the next interval the neighbor with the rightmost right end



# Dominating Path Decomposition

- start with the interval with the leftmost left end
- at each step, take as the next interval the neighbor with the rightmost right end





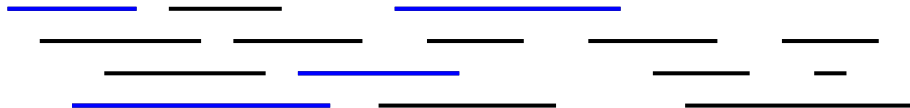
# Dominating Path Decomposition

- start with the interval with the leftmost left end
- at each step, take as the next interval the neighbor with the rightmost right end



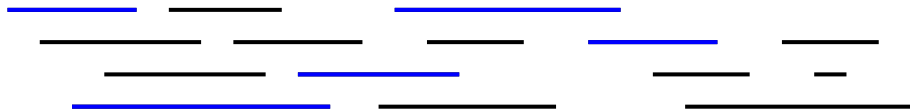
# Dominating Path Decomposition

- start with the interval with the leftmost left end
- at each step, take as the next interval the neighbor with the rightmost right end



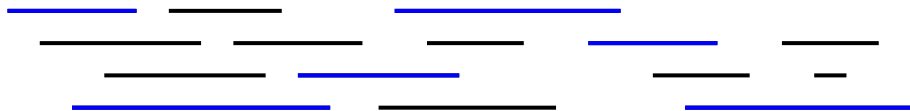
# Dominating Path Decomposition

- start with the interval with the leftmost left end
- at each step, take as the next interval the neighbor with the rightmost right end



# Dominating Path Decomposition

- start with the interval with the leftmost left end
- at each step, take as the next interval the neighbor with the rightmost right end

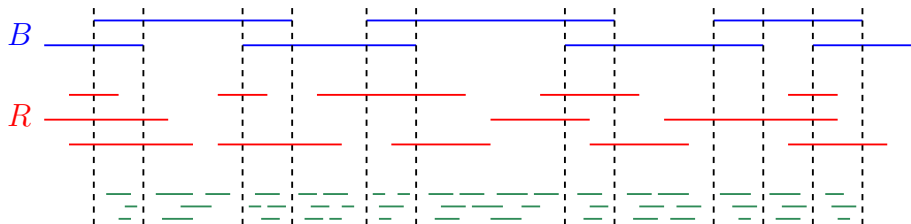


⇒ every interval is contained in the union of two consecutive intervals of the blue path

# Dominating Path Decomposition

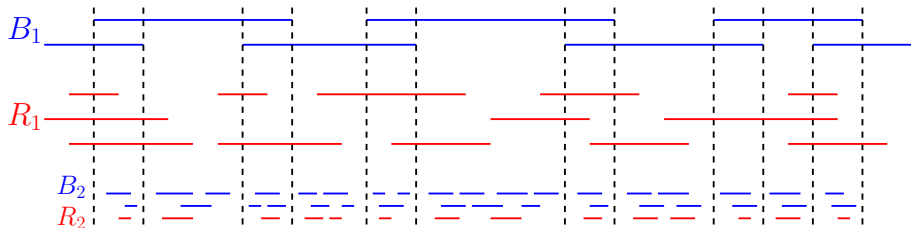
Three kinds of intervals:

- the **blue** intervals  $B$  of the dominating path
- the **red** intervals  $R$  that contain at least one end of a blue interval
- the remaining **green** intervals



# Dominating Path Decomposition

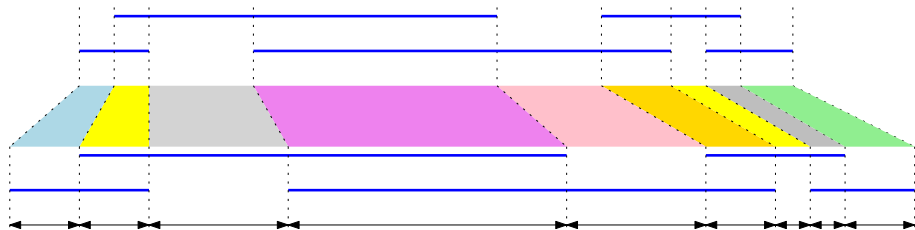
- green intervals have thickness strictly less than the original
- repeat this process to obtain a decomposition into  $(B_1, R_1), (B_2, R_2), \dots, (B_{p+1}, R_{p+1})$



# Blue Node Encoding

Enlarge the spaces between two consecutive ends of the path such that:

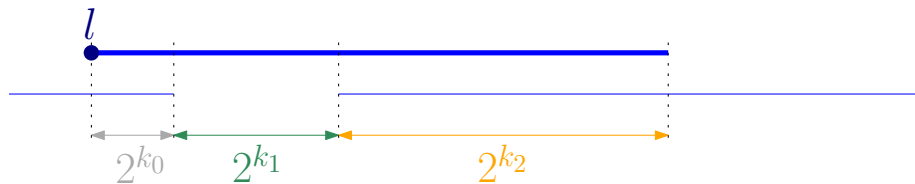
- the spaces between ends are power of two
- two consecutive spaces differ only by a multiplicative factor at most 2



# Blue Node Encoding

In the label of each blue interval, we encode:

- the new left end  $l$
- the integers  $k_0, k_1$  and  $k_2$
- the layer index (= index  $i$  of  $B_i$  containing it)
- some more information (more details later)

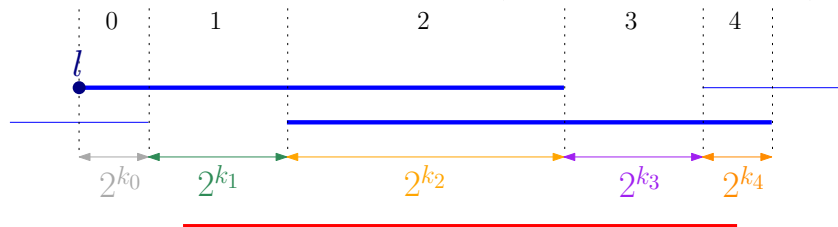




# Red Node Encoding

In the label of each **red** interval, we encode information relative to the two blue intervals containing it:

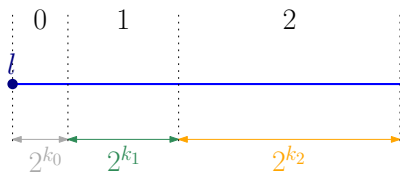
- the left end  $l$  of the first blue interval
- the integers  $k_0, k_1, k_2, k_3$  and  $k_4$
- the range  $\subseteq [0, 4]$  where the interval falls into
- the layer index
- some more information (more details later)



# Adjacency: Easy Case

The node with the smaller layer index is **blue**

⇒ adjacency is easy since one can compute exactly the ends of the blue interval



subcase 1: red interval of the same layer

---

subcase 2: interval (blue or red) with greater layer

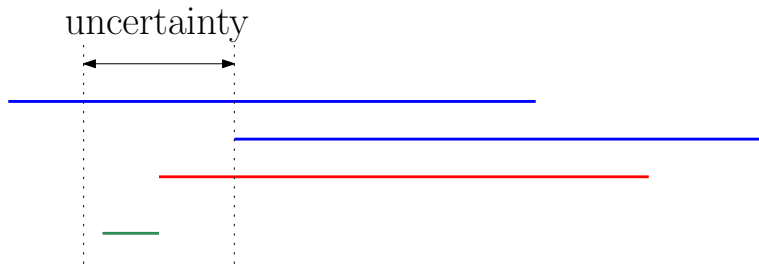
---

---

# Adjacency: Hard Case

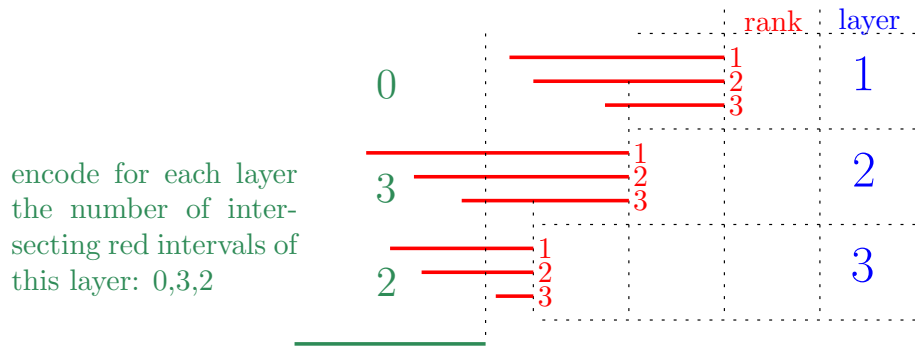
If the node with the smaller layer index is **red**

⇒ adjacency is harder to compute since one can only compute some range containing the ends of the interval



# Adjacency: Hard Case

Each **red** interval stores the rank for each of its ends (order by length)



⇒ the node (blue or red) can check if the **red** interval corresponds to an interval intersecting it

# Conclusion and Open Problems

## Summary

- Universal graph  $\mathcal{U}_n$  for caterpillar forests  
 $3n/2 \leq |V(\mathcal{U}_n)| < 8n$  (pathwidth  $p = 1$ )
- Universal graph for graphs of pathwidth  $p$  with  $n \cdot 2^{\Theta(p)}$  nodes ( $\log n + \Theta(p)$ -bit labels)

## Open Problems

- Improve bounds for universal graph for caterpillars
- $\log n + \Theta(\log p)$ -bit labels for interval graphs of maximum clique size  $p$
- $\log n + \Theta(t)$ -bit labels for graphs of treewidth  $t$