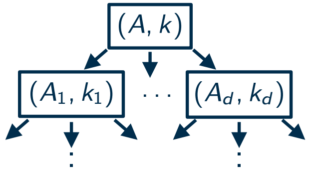
Parameterized Algorithms

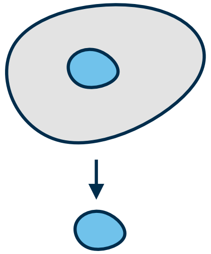
Treewidth & Dynamic Programming

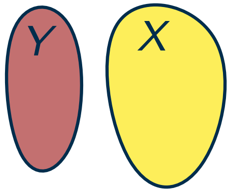
Thomas Bläsius

Content

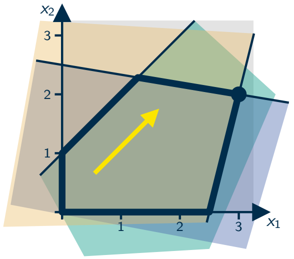
Basic toolbox
- bounded search trees
- kernelization
- iterative compression

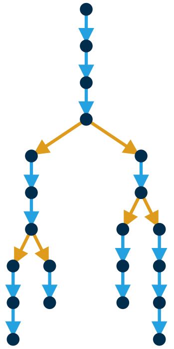


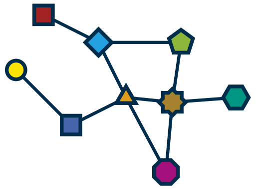




Extended toolbox
- linear programs
- branch-and-reduce
- color coding

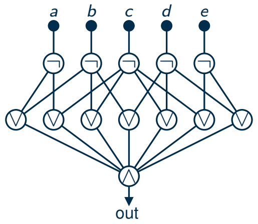


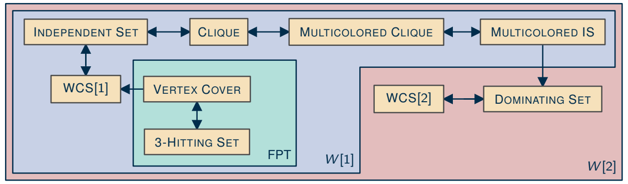




Tree width
- dynamic programming
- chordal and planar graphs
- Courcelle’s theorem

Lower bonds
- kernel lower bounds
- parameterized reductions
- circuits and the W-hierarchy
- ETH and SETH



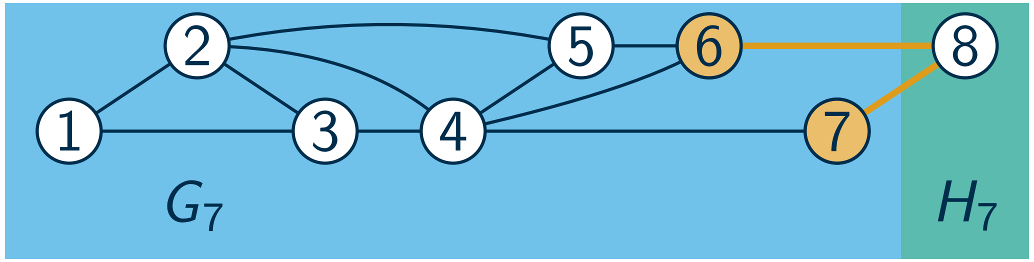


Graphs with order

Graph $G = (V, E)$ with vertex order $V = \{v_1, \ldots, v_n\}$

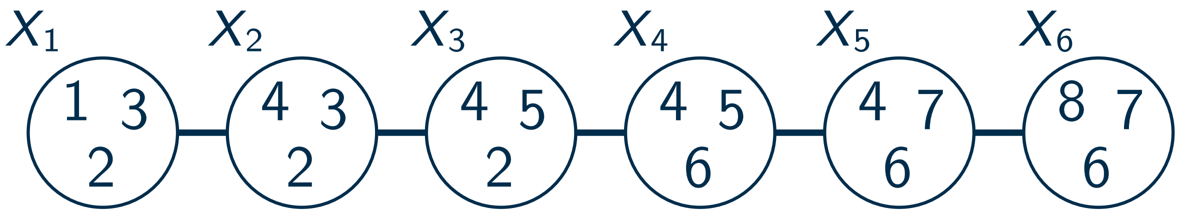
- $V_i = \{v_1, \ldots, v_i\}$
- $G_i = G[V_i]$ and $H_i = G[V \setminus V_i]$
- k_i = #vertices in G_i with edges to H_i
- $k = \max\{k_i\}$

$k_7 = 2$
$\Rightarrow k = 2$



Alternative perspective: Path decomposition

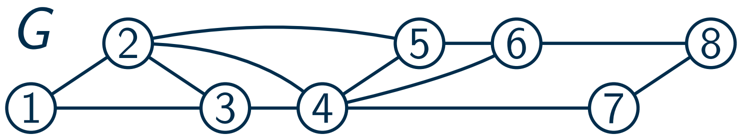
- path $x_1, \ldots, x_r$ with **bags** $X_1, \ldots, X_r$ ($X_i \subseteq V$), such that:
  - $X_1 \cup \cdots \cup X_r = V$
  - $\{u, v\} \in E \Rightarrow u, v$ share a bag
  - the bags of every vertex form a subpath

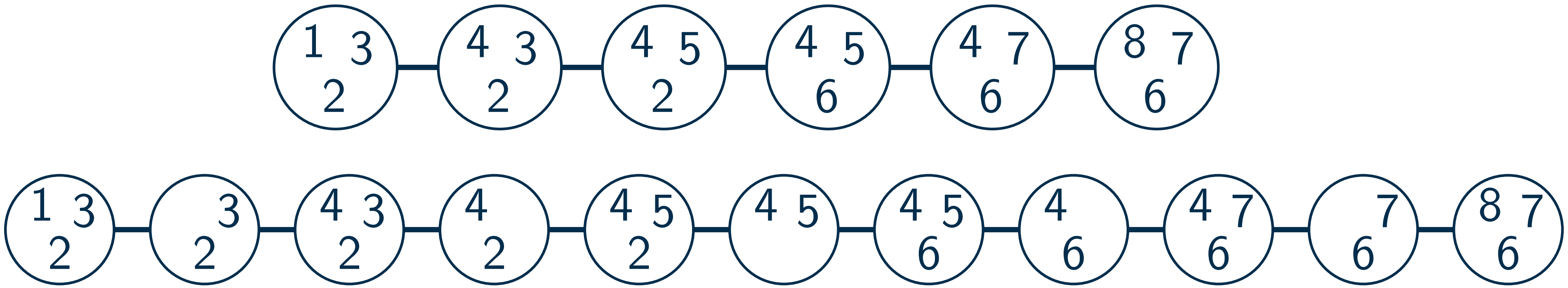


Pathwidth and nice path decompositions

Pathwidth

- width of a decomposition: $\max\{|X_i|\} - 1$
- pathwidth of a graph: minimum width of a decomposition





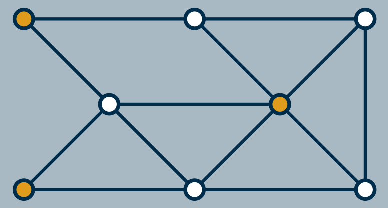
Nice path decompositions

- only two node types: introduce and forget
- x_i is introduce node if $X_i = X_{i-1} \cup \{v\}$
- x_i is forget node if $X_i = X_{i-1} \setminus \{v\}$

Lemma
If G has a path decomposition of width p, then G has a nice path decomposition of width p.

Dynamic programming

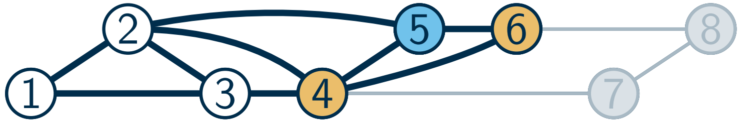
Problem: INDEPENDENT SET
Given a graph, a parameter p, and a path decomposition of width p. Is there an independent set of size k?
(vertices $V' \subseteq V$ with $\{u, v\} \notin E$ for $u, v \in V'$)

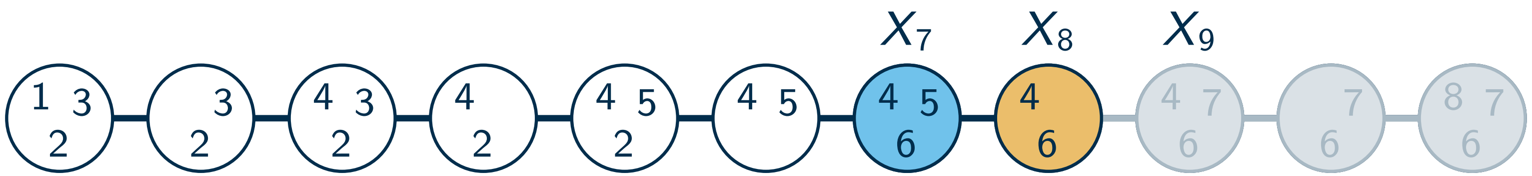


Dynamic programming over nice path decomposition

- as before: $V_i = X_1 \cup \cdots \cup X_i$ and $G_i = G[V_i]$
- step i: $\forall X'_i \subseteq X_i$, compute max IS U_i in G_i with $U_i \cap X_i = X'_i$

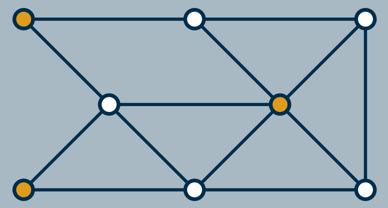
subset $X'_7 \subseteq X_7$	$\emptyset$	$\{4\}$	$\{5\}$	$\{6\}$	$\{4,5\}$	$\{4,6\}$	$\{5,6\}$	$\{4,5,6\}$
largest IS size in G_7	1	2	2	2	$-\infty$	$-\infty$	$-\infty$	$-\infty$
subset $X'_8 \subseteq X_8$	$\emptyset$	$\{4\}$		$\{6\}$		$\{4,6\}$		
largest IS size in G_8	2	2		2		$-\infty$		





Dynamic programming

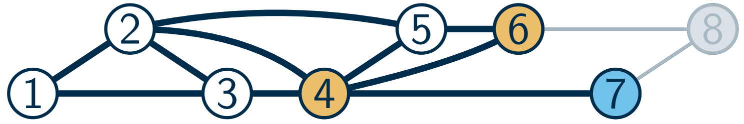
Problem: INDEPENDENT SET
Given a graph, a parameter p, and a path decomposition of width p. Is there an independent set of size k? (vertices $V' \subseteq V$ with $\{u, v\} \notin E$ for $u, v \in V'$)

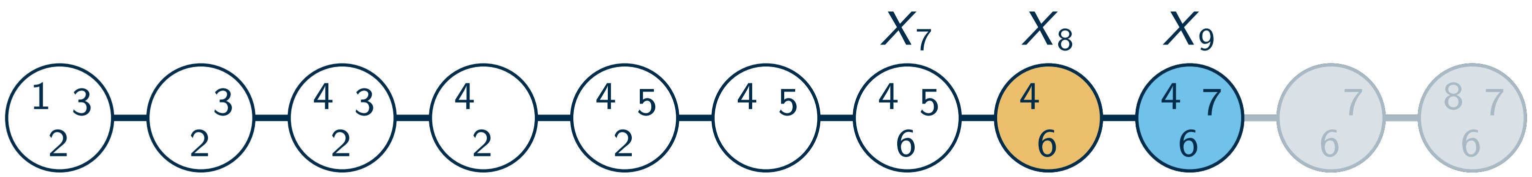


Dynamic programming over nice path decomposition

- as before: $V_i = X_1 \cup \cdots \cup X_i$ and $G_i = G[V_i]$
- step i: $\forall X'_i \subseteq X_i$, compute max IS U_i in G_i with $U_i \cap X_i = X'_i$

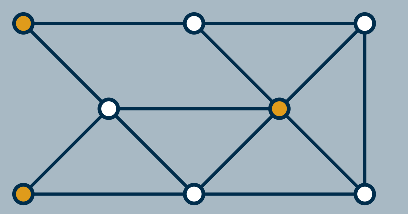
subset $X'_8 \subseteq X_8$	$\emptyset$		$\{4\}$		$\{6\}$		$\{4, 6\}$	
largest IS size in G_8	2		2		2		$-\infty$	
subset $X'_9 \subseteq X_9$	$\emptyset$	$\{7\}$	$\{4\}$	$\{4, 7\}$	$\{6\}$	$\{6, 7\}$	$\{4, 6\}$	$\{4, 6, 7\}$
largest IS size in G_9	2	3	2	$-\infty$	2	3	$-\infty$	$-\infty$





Dynamic programming

Problem: INDEPENDENT SET
Given a graph, a parameter p, and a path decomposition of width p. Is there an independent set of size k? (vertices $V' \subseteq V$ with $\{u, v\} \notin E$ for $u, v \in V'$)

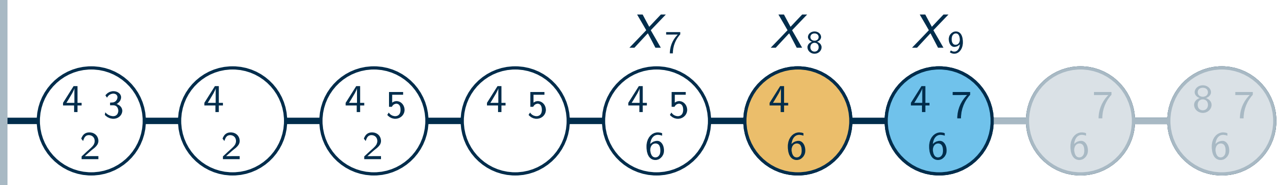


Dynamic programming over nice path decomposition

- as before: $V_i = X_1 \cup \dots \cup X_i$ and $G_i = G[V_i]$
- step i: $\forall X'_i \subseteq X_i$, compute max IS U_i in G_i with $U_i \cap X_i = X'_i$

subset $X'_8 \subseteq X_8$	$\emptyset$		$\{4\}$		$\{6\}$		$\{4, 6\}$	
largest IS size in G_8	2		2		2		$-\infty$	
subset $X'_9 \subseteq X_9$	$\emptyset$	$\{7\}$	$\{4\}$	$\{4, 7\}$	$\{6\}$	$\{6, 7\}$	$\{4, 6\}$	$\{4, 6, 7\}$
largest IS size in G_9	2	3	2	$-\infty$	2	3	$-\infty$	$-\infty$

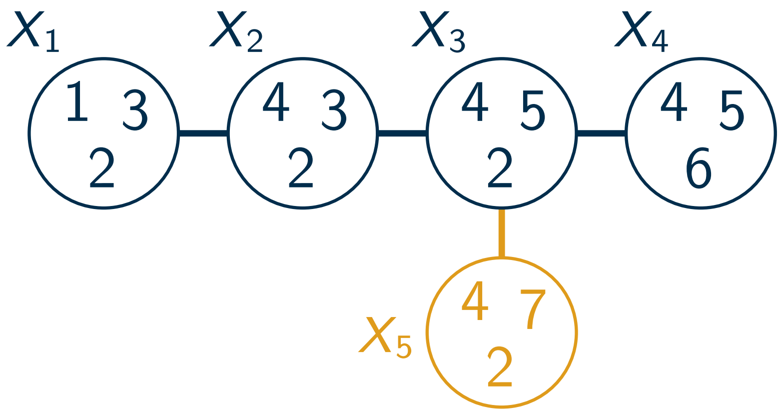
Theorem
INDEPENDENT SET is FPT with respect to the pathwidth p (the DP runs in $O(2^p n^c)$).



Treewidth

Tree
~~Path~~ decomposition

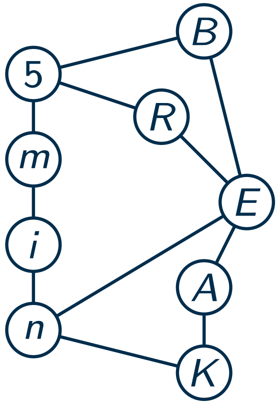
- ~~path~~ tree on nodes $x_1, \dots, x_r$ with bags $X_1, \dots, X_r$, such that:
  - $X_1 \cup \dots \cup X_r = V$
  - $\{u, v\} \in E \Rightarrow u, v$ share a bag
  - the bags of every vertex form a sub~~path~~tree



Treewidth

- width of a decomposition:
  $\max\{|X_i|\} - 1$
- treewidth of a graph:
  minimum width of a decomposition

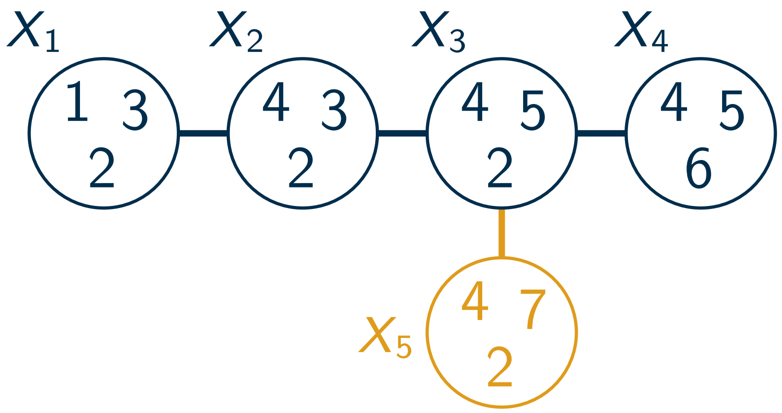
What is the treewidth of this graph?



Treewidth

~~Path~~ Tree decomposition

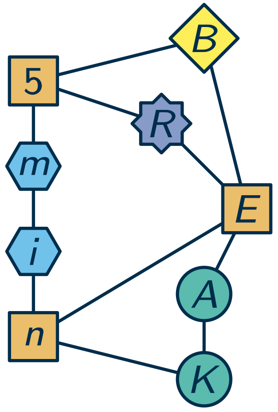
- ~~path~~ tree on nodes $x_1, \dots, x_r$ with bags $X_1, \dots, X_r$, such that:
  - $X_1 \cup \dots \cup X_r = V$
  - $\{u, v\} \in E \Rightarrow u, v$ share a bag
  - the bags of every vertex form a sub~~path~~tree

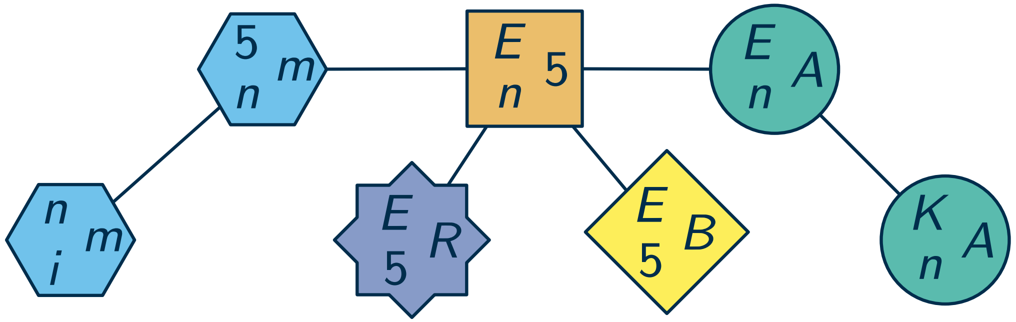


Treewidth

- width of a decomposition:
  $\max\{|X_i|\} - 1$
- treewidth of a graph:
  minimum width of a decomposition

What is the treewidth of this graph?





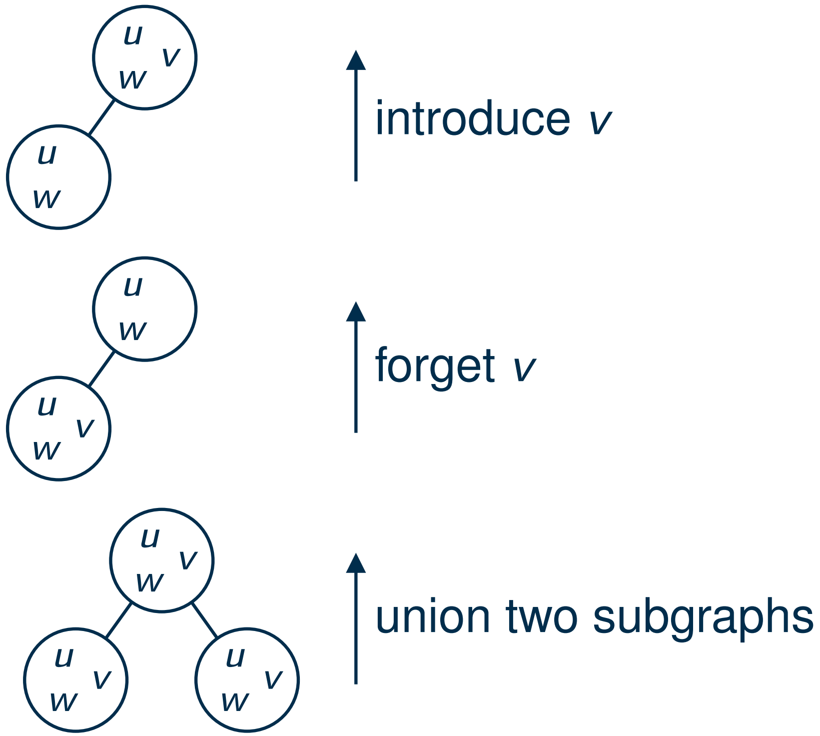
$\Rightarrow$ treewidth 2

Nice tree decompositions

Three types of nodes

- x_i is **introduce node** if
  - x_i has exactly one child x_j and
  - $X_i = X_j \cup \{v\}$ for one vertex $v \in V$
- x_i is **forget node** if
  - x_i has exactly one child x_j and
  - $X_i = X_j \setminus \{v\}$ for a vertex $v \in V$
- x_i is **join node** if
  - x_i has exactly two children x_j, x_k
  - and $X_i = X_j = X_k$

intuition: DP runs bottom-up

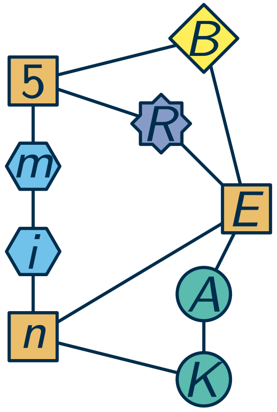


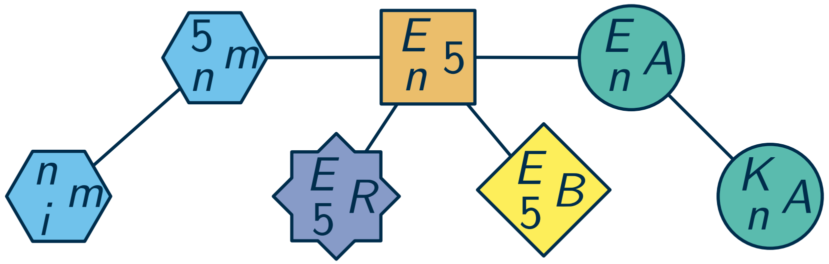
Theorem
If G has treewidth t, then it has a nice tree decomposition of width t.

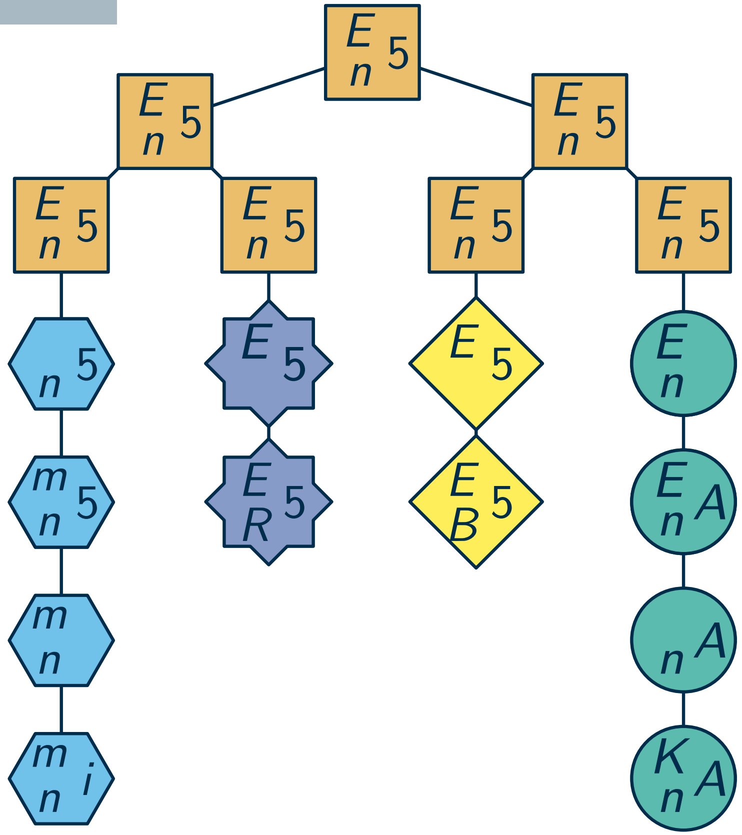
Making things nice

Theorem
If G has treewidth t, then it has a nice tree decomposition of width t.

Example







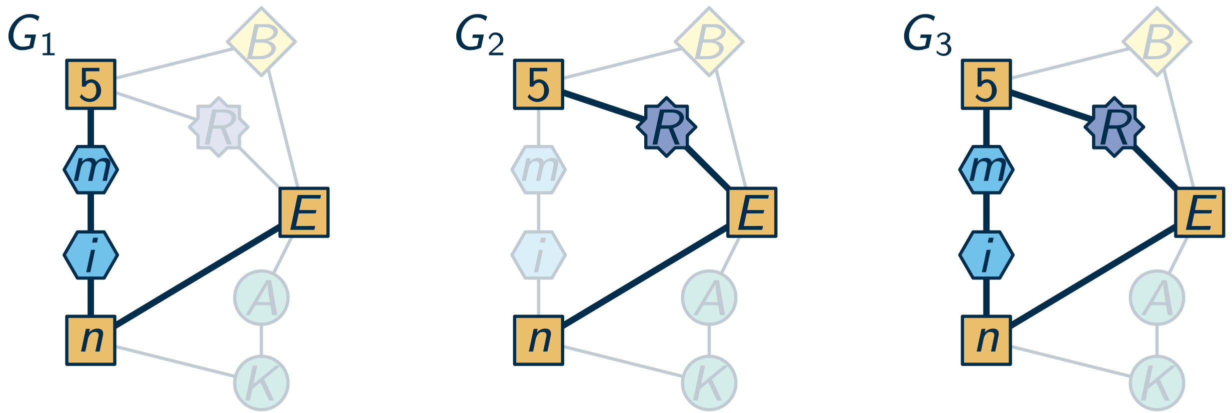
DP over nice tree decompositions

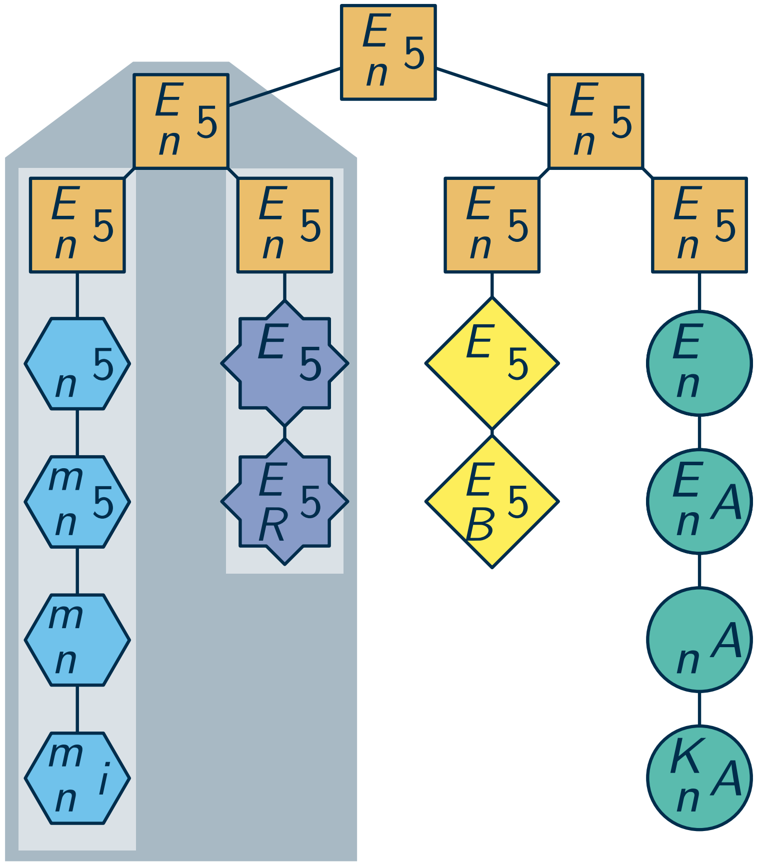
Reminder: INDEPENDENT SET
- vertex set with no edges
- maximize size
- parameter: treewdith t

Dynamic programming over a nice tree decomposition

- introduce/forget node: as before for the path decomposition
- join node:

	$\emptyset$	$\{E\}$	$\{5\}$	$\{n\}$	$\{E, 5\}$	$\{E, n\}$	$\{n, 5\}$	$\{E, n, 5\}$
G_1	1	2	2	2	3	$-\infty$	2	$-\infty$
G_2	1	1	1	2	2	$-\infty$	2	$-\infty$
G_3	2	2	2	3	3	$-\infty$	2	$-\infty$



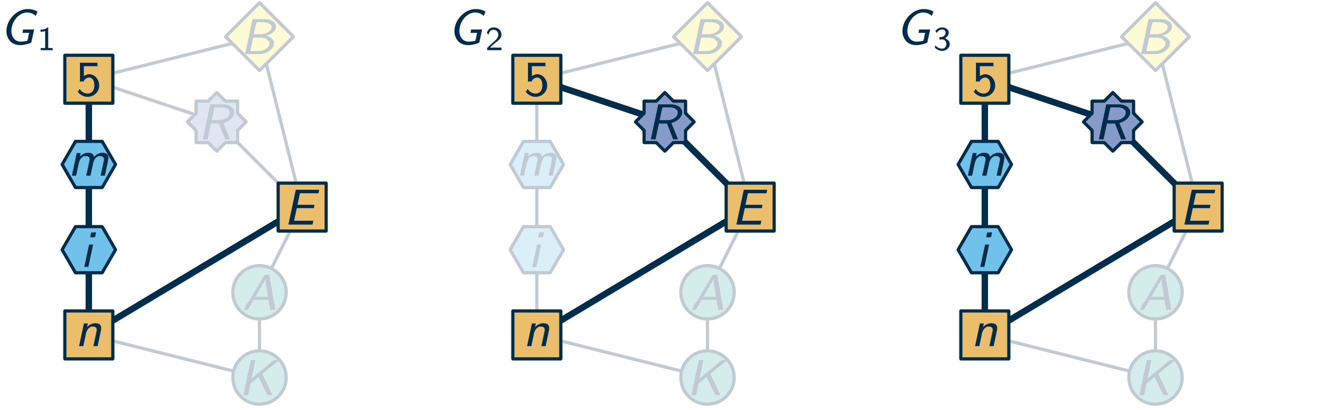


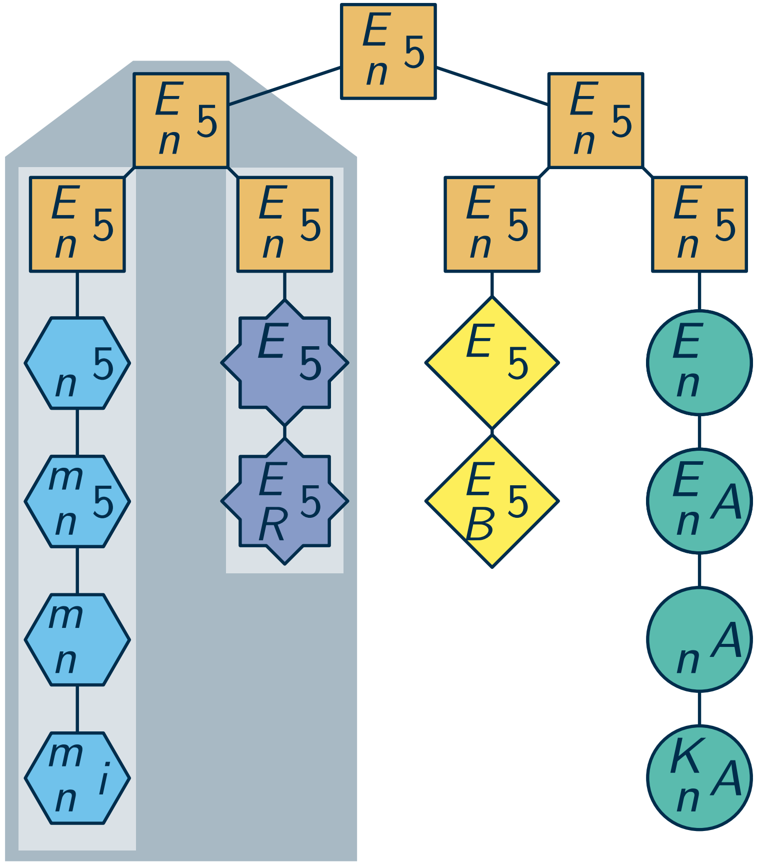
DP over nice tree decompositions

Reminder: INDEPENDENT SET
- vertex set with no edges
- maximize size
- parameter: treewdith t

Dynamic programming over a nice tree decomposition
- introduce/forget node: as before for the path decomposition
- join node:

	$\emptyset$	$\{E\}$	$\{5\}$	$\{n\}$	$\{E, 5\}$	$\{E, n\}$	$\{n, 5\}$	$\{E, n, 5\}$
G_1	1	2	2	2	3	$-\infty$	2	$-\infty$
G_2	1	1	1	2	2	$-\infty$	2	$-\infty$
G_3	2	2	2	3	3	$-\infty$	2	$-\infty$



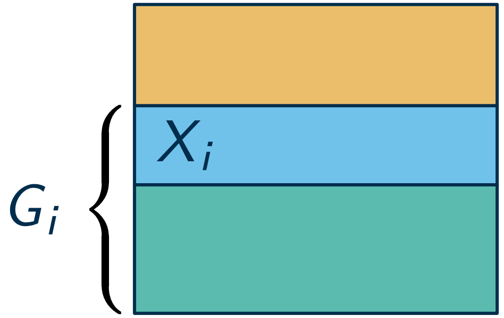


Theorem: INDEPENDENT SET is FPT with respect to treewidth.

DPs on tree decompositions

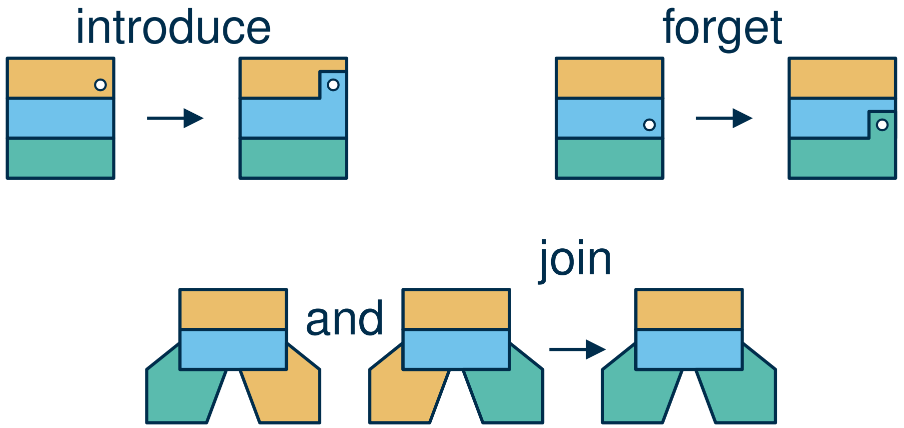
Notation and basic properties

- for node x_i: $V_i = X_i \cup \{v \in V \mid v \in X_j \text{ for successor } x_j \text{ of } x_i\}$
- $G_i = G[V_i]$ and X_i is the interface of G_i
- X_i separates G_i from the rest



Node types from the perspective of G_i

- **introduce:** add vertex to G_i and to the interface
- **forget:** remove a vertex from the interface of G_i
- **join:** union two subgraphs with the same interface

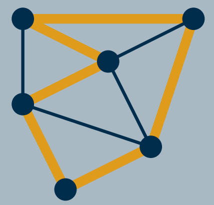


Desired properties for partial solutions on (G_i, X_i)

- for $G_i = G$, the actual solution is included
- in every step: all new partial solutions can be computed based on old ones

DP for Hamiltonian cycle – partial solutions

Problem: HAMILTONIAN CYCLE
Given a graph G. Does G have a cycle that visits all vertices?

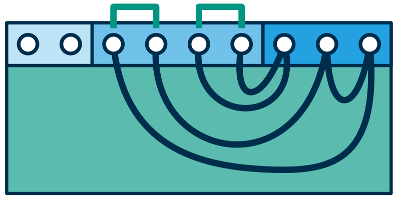


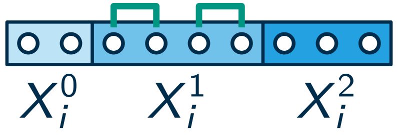
Goal

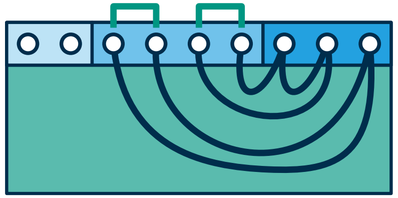
- FPT with respect to treewidth
- given: corresponding tree decomposition

What is a useful definition for partial solutions?

- observation: Hamiltonian cycle in G yields a set of paths in G_i
- define partial solution (first try): set of paths in G_i between interface vertices
- problem: too many sets of paths
- observation: different path sets are indistinguishable from the outside, if
  - same vertices of the interface incident to 0, 1, and 2 selected edges
  - the same pairs of 1-vertices are matched by paths
- define partial solution: partition of X_i into $X_i = X_i^0 \cup X_i^1 \cup X_i^2$ and matching on X_i^1

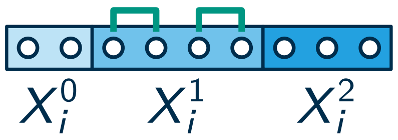






DP for Hamiltonian cycle – forget node

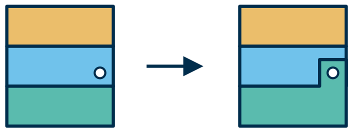
Partial solution:



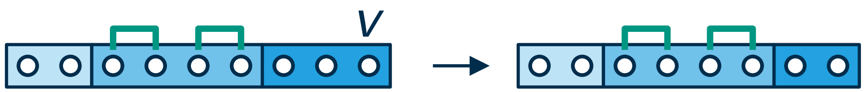
What does the DP do in general?

- process nodes bottom up
- when node is processed: assume all children have been processed
- invariant after processing x_i: we know all partial solutions that
  - have corresponding paths in G_i
  - could be extended to a Hamiltonian cycle (→ the paths must visit all vertices of G_i that are not in the interface X_i)

Computing new partial solutions: forget node



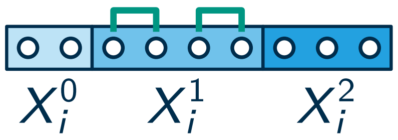
- let v be the vertex that is forgotten
- if $v \in X_i^2$:

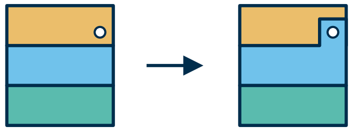


- if $v \notin X_i^2$:
  - v has no chance to be part of the Hamiltonian cycle
  - dead end → ignore this partial solution of the child

DP for Hamiltonian cycle – introduce node

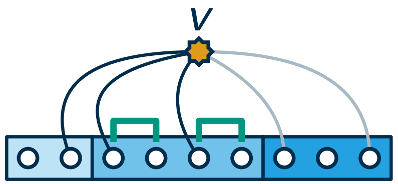
Partial solution:

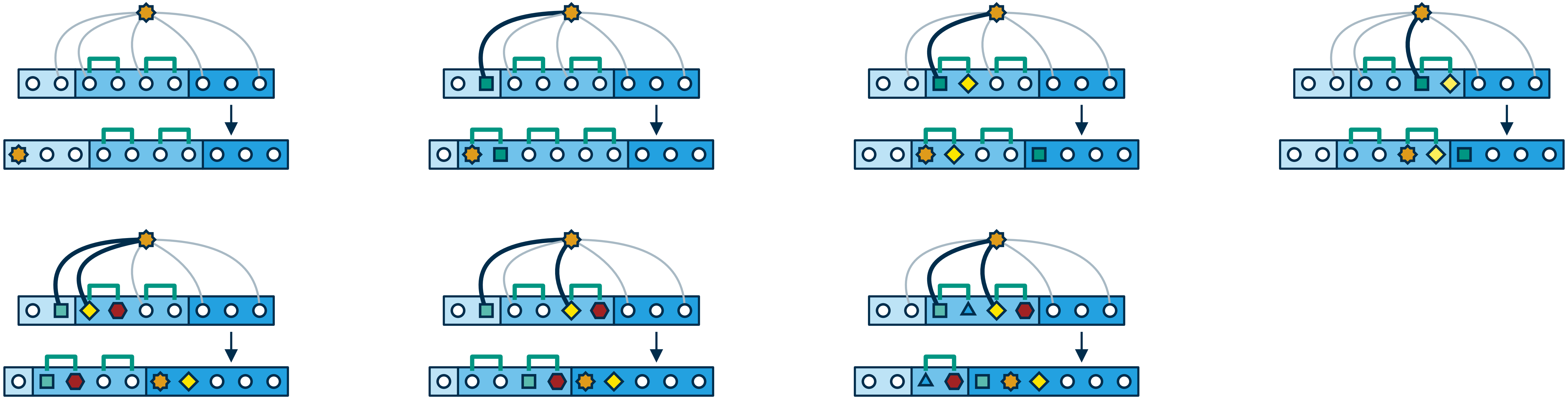




Computing new partial solutions: introduce node

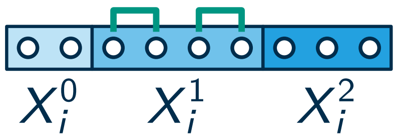
- let v be the introduced vertex and consider its edges to the interface X_i
- up to two edges incident to v can be part of a Hamiltonian cycle
  - edges to X_i^2 cannot be selected
  - not both edges to the same matched pair in X_i^1 can be selected



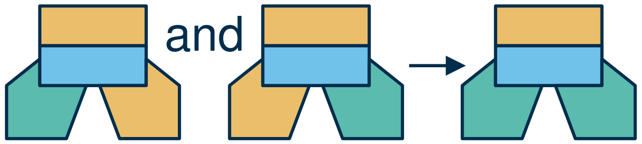


DP for Hamiltonian cycle – join node

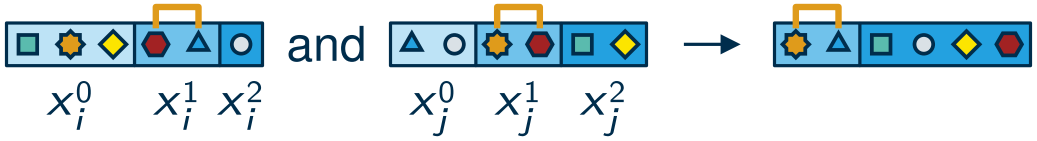
Partial solution:



Computing new partial solutions: join node

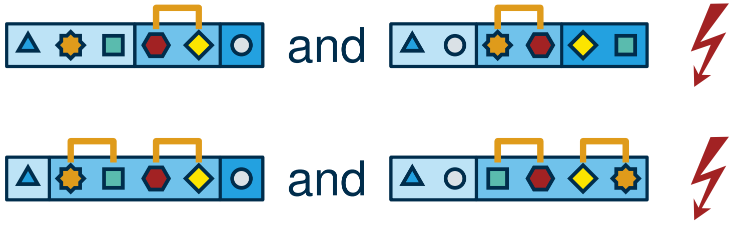


- combining two partial solutions $\rightarrow$ union of selected edges



- union not possible if:
  - for a vertex, the sum of degrees in G_i and G_j is > 2
  - a cycle is closed

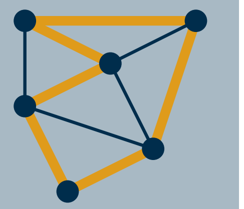
(special case: in the root x_r, we want to create a single cycle such that all vertices are in X_r^2)



- otherwise:
  - sum the vertex degrees to get the partition
  - compute which pairs of 1-vertices belong to the same path

DP for Hamiltonian cycle

Problem: HAMILTONIAN CYCLE
Given a graph G. Does G have a cycle that visits all vertices?



Goal

- FPT with respect to treewidth
- given: corresponding tree decomposition

Is the DP's running time FPT?

- for each tree-node: number of partial solutions only depends on bag (aka interface) size
- computing new partial solutions: also only depends on bag size

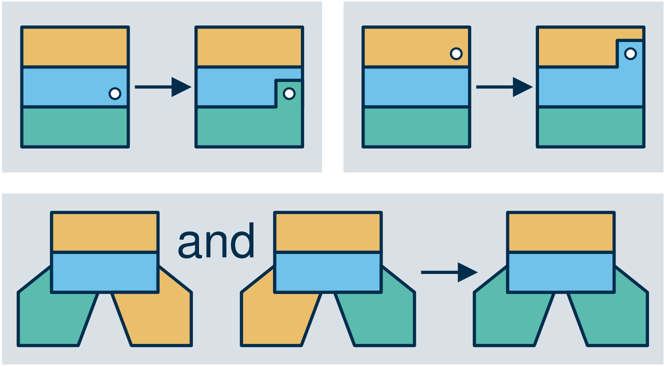
Theorem
HAMILTONIAN PATH is FPT with respect to the treewidth t. (assuming we know a corresponding composition)

Wrap-UP

Treewidth

- structural graph parameter
- measures similarity to trees
  (in regards to separators)

DP on a tree decomposition



- nice tree decompositions are nice
- difficulty: good notion of "partial solution"
  - number of partial solutions only depends on interface size
  - each node type: all partial solutions of parent computable from all partial solutions of children

What is the treewidth of real-world graphs?

- a heuristic on two "randomly" chosen graphs yields:
  - links between political blogs: $n = 642$, $m = 2280$, $t \leq 42$
  - co-author network: $n = 226\,413$, $m = 716\,460$, $t \leq 11\,775$