

On non-bijective tree-questionable-width

Laurent Lyaudet*

November 24, 2025

Abstract

In this article, we show that there is an infinity of non-bijective tree-questionable-width types, including 8 main types. The study of these types comes from an example of tree-questionable-decomposition of unbounded degree and depth 2 for all binary structure. We hierarchise these types and give links with bijective tree-questionable-width.

Initial version: 2025/06/12 Current version: 2025/11/24

Keywords: principle of first difference, principle of first differences, first difference principle, binary structures, 2-structures, graphs, tree-questionable-width, hierarchical decompositions

1 Introduction

(Balanced) (bijective or non-bijective) tree-questionable-width was introduced in Lyaudet (2019). We give new non-bijective variants, study the links with the maximum degree of the decomposition tree, and the links with the bijective case. What follows comes from the detailed study of:

Example 1.1. *Let S be a binary structure of cardinality n ; in a non-bijective decomposition, we can repeat each vertex ($n - 1$ times in the finite case), to make “cherries” for each adjacency between two vertices with two leaves, one for each vertex, plus an internal node that links the two leaves/vertices with the correct adjacency type. Then, we link all these “cherries” to a unique root node that collects all the adjacencies.*

This example works or not depending on the meaning we give to principle of first differences on a tree. (In our previous articles, we wrote first difference principle with singular difference, because we memorised the terms “principe de première différence” after reading the article Sierpiński (1932). This error was reinforced by the fact that for the bijective case, there is indeed a unique difference for each pair of vertices, as in the works by Sierpiński. Our works on the non-bijective case give a new meaning to the plural “differences”. Since our previous articles, we also found that principle of first differences was used in English instead of first difference(s) principle.) Our example

*<https://lyaudet.eu/laurent/>, laurent.lyaudet@gmail.com

dates from the end of 2019 or the beginning of 2020, but the clarification and what follows is more recent; because in a first time, we just had the reflex to focus on the bijective case with bounded degree, as we did with the binary trees in Lyaudet (2022), Lyaudet (2025b) and Lyaudet (2025a).

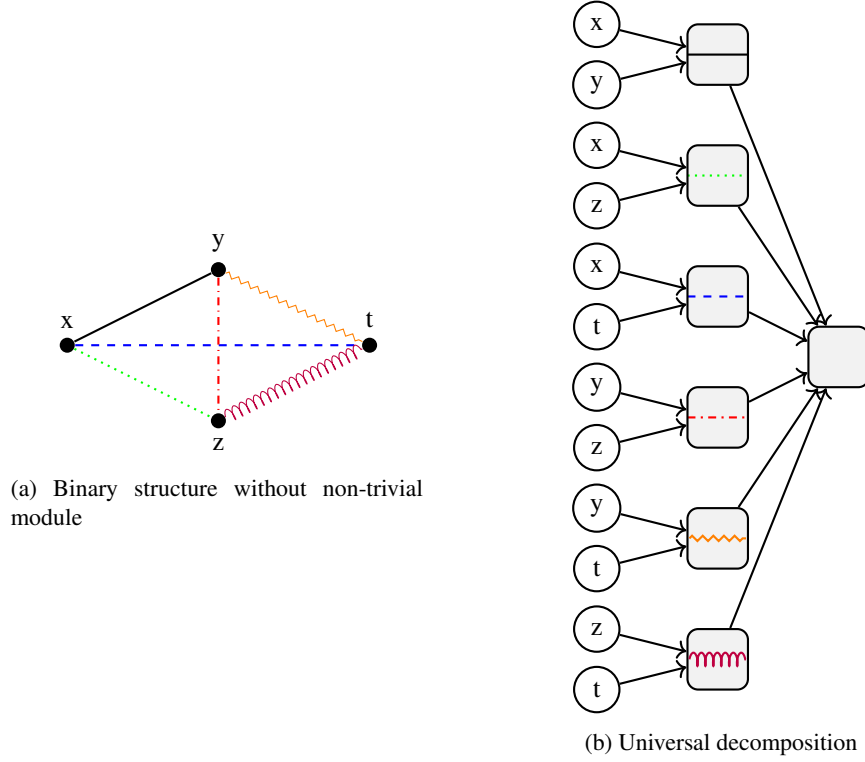


Figure 1: Example 1.1

2 Definitions

In this article, the letters x , y and z will correspond to vertices of a binary structure. Whilst the letter u will correspond to nodes of a tree(-questionable-decomposition) (u is a little like n upside down, but n is already used for the cardinality of the decomposed binary structure).

Definition 2.1 (Mapping-run). *Let V be a set of vertices, a (V) -mapping-run is a sequence of mappings from V to vertices of binary structures (the image binary structure is fixed per mapping). The supremum of cardinals of image binary structures, denoted by κ , is called the width of the mapping-run. The length of the mapping-run is denoted by λ . Then we talk about a (V, κ, λ) -mapping-run.*

We start by technical details on trees intersections.

We say that a leaf of $\text{CT}(T_1, T_2)$ is a tight junction point of $\text{CT}(T_1, T_2)$.
We say that a node of $\text{CT}(T_1, T_2)$ is a short junction point of $\text{CT}(T_1, T_2)$ if this node has a son in T_1 that is not in $\text{CT}(T_1, T_2)$, and a son in T_2 that is not in $\text{CT}(T_1, T_2)$.
We say that a node of $\text{CT}(T_1, T_2)$ is a medium junction point of $\text{CT}(T_1, T_2)$ if this node has a son in T_1 or a son in T_2 that is not in $\text{CT}(T_1, T_2)$.
We say that a node of $\text{CT}(T_1, T_2)$ is a wide junction point of $\text{CT}(T_1, T_2)$ if this node has at least two sons in the tree induced by the union of nodes of T_1 and T_2 .
In a similar way, we define a tight, resp. short, resp. medium, resp. wide, ascending path as a directed path from a tight, resp. short, resp. medium, resp. wide, junction point of $\text{CT}(T_1, T_2)$ to the root of $\text{CT}(T_1, T_2)$ for some pair $\{T_1, T_2\}$. Hence, a tight ascending path is a leaf-root path.

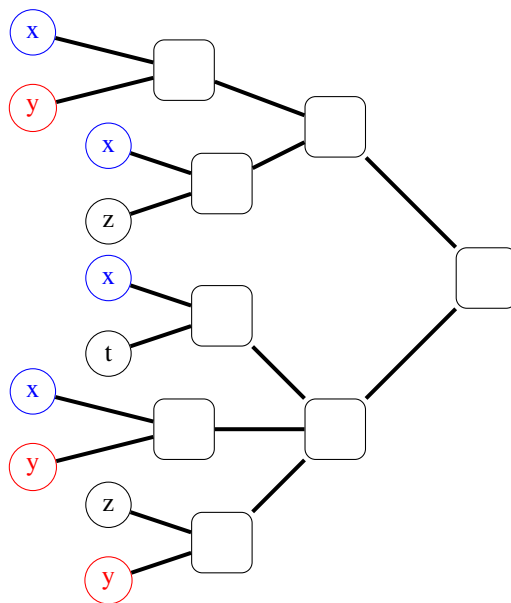


Figure 2: Decomposition tree

Lemma 2.3. *A tight junction point is a short junction point. A short junction point is a medium junction point. A medium junction point is a wide junction point.*

Lemma 2.4. *A tight ascending path is a short ascending path. A short ascending path is a medium ascending path. A medium ascending path is a wide ascending path. Hence, a leaf-root path is tight, short, medium, and wide.*

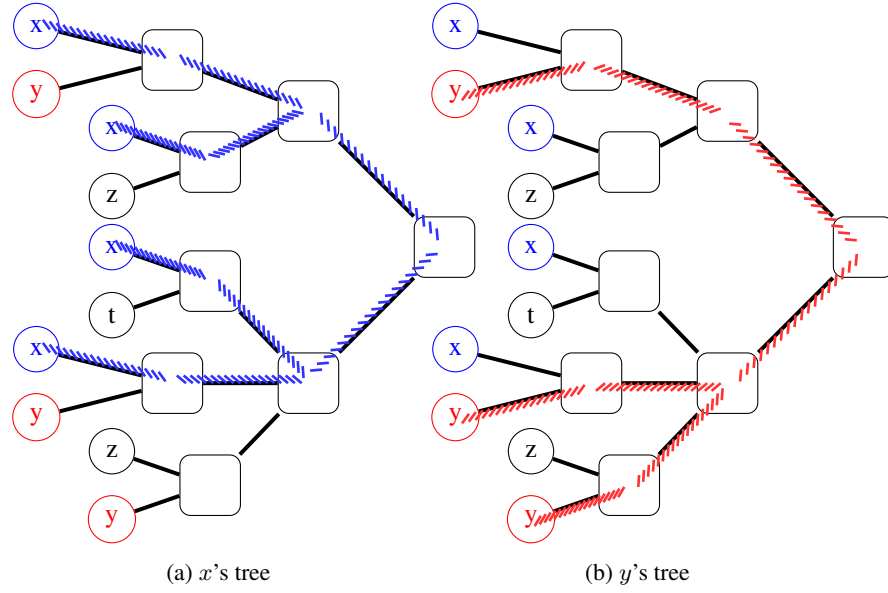


Figure 3: Trees induced by two vertices

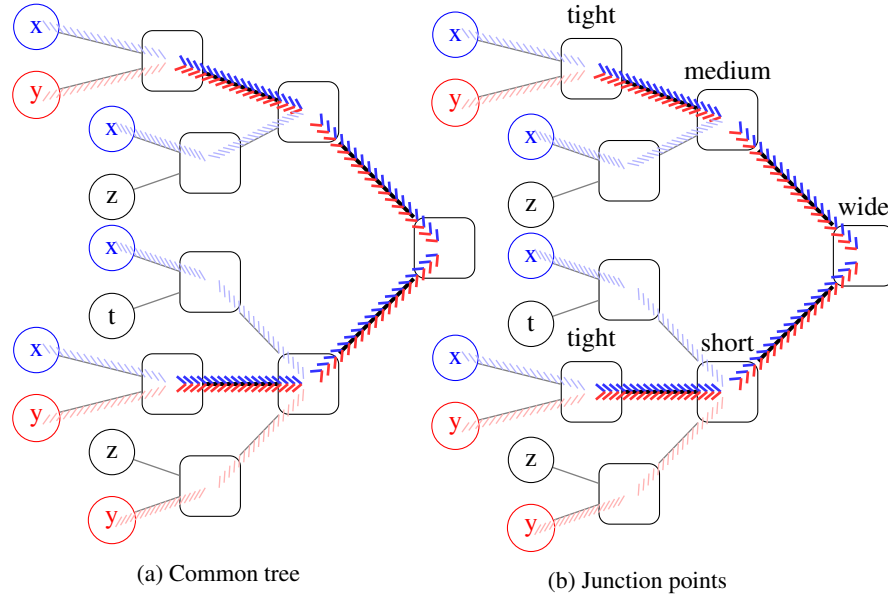


Figure 4: Common tree and junction points

Lemma 2.5. *All (tight, short, medium, or wide) ascending path is included in at least one leaf-root path.*

There are many definitions that generalize rooted trees in the infinite setting, we use this one: We replace the tree by a well-founded order, where the minimal elements/nodes correspond to the leaves; there is a maximum element/node (corresponding to the root); for any two nodes u_1, u_2 , the initial sections generated by them, $IS(u_1), IS(u_2)$, (subsets of elements/nodes less than u_1 , resp. u_2) are either disjoint (disjoint subtrees), or one is contained in the other (indicating that u_1 is an ancestor or a descendant of u_2). As a result, the final section generated by u_1 , $FS(u_1)$, is always a chain/path with a maximum element corresponding to the root. The “inner nodes” are thus the non-minimal elements of this partial order. In the partial suborder induced by $IS(u_1) \setminus \{u_1\}$ (that may be without any maximal element), we look at the set of “complete subtrees” maximal for inclusion, that is to say that these are the maximal subsets of nodes $F \subseteq IS(u_1) \setminus \{u_1\}$ such that $\forall u_3, u_4 \in F$, if u_3 is incomparable with u_4 , then $\exists u_5 \in F$ such that $u_3 < u_5$ and $u_4 < u_5$; such a subtree is an initial section generated by a chain; we say this subtree is a “son” of u_1 ; the cardinality of this set of “sons”/maximal subtrees inclusion-wise defines the degree of u_1 . When we reason on some “sons” in the infinite case, since each son is a set of nodes and not a unique node, we consider that it is the “same” son when the two sets of nodes intersect. The reader can verify that in all the reasonings of this article, no “son” according to a first context intersects two “sons” according to a second context; it validates the junction points definitions and all the rest.

Definition 2.6 (Tree-questionable-decomposition). *Let S be a binary structure. A tree-questionable-decomposition D of S is a triplet (T, ll, nl) (T as tree, ll as leaf labels et nl as node labels) such that:*

- T is a rooted tree;
- the function ll is a surjective mapping (a bijection in the bijective case) from the leaves of T to the vertices of S ;
- hence, each internal node u is associated to the subset of vertices of S union of the values $ll(l)$ for all leaves l under the node u ; it defines $ll(u)$;
- nl is a mapping with domain the internal nodes of T , such that $nl(u)$ is an $(ll(u))$ -mapping-run. (It is possible in tree-questionable-decompositions to have nodes of the decompositions that are mapped to an empty mapping-run/of length 0.);
- as a consequence, each vertex x of S corresponds to a subtree (that is a path in the bijective case) of T , we denote by T_x this subtree, resp. path;
- for every pair of vertices $\{x, y\}$, we will look at the principle of first differences on $CT(T_x, T_y)$, according to the distinct junction points;
- let P be a tight, short, medium or wide ascending path relatively to $\{x, y\}$; we can define the $(\{x, y\})$ -mapping-run obtained by concatenating the $(ll(u))$ -mapping-runs restricted to $\{x, y\}$ when we take the nodes u of P from the junction point to the root; if a first difference between the images of x and of y exists in this mapping-run, it is the question of P ; when the path P has a question, it

must correspond to two vertices of same adjacency type as between x and y to be valid;

- *we say that the tree-questionable-decomposition is tight, resp. short, resp. medium, resp. wide, if all the questions of tight, resp. short, resp. medium, resp. wide, ascending paths are valid;*
- *we say that the tree-questionable-decomposition is 1-tight, resp. 1-short, resp. 1-medium, resp. 1-wide, if it is tight, resp. short, resp. medium, resp. wide, and all the tight, resp. short, resp. medium, resp. wide, ascending paths have a valid question;*
- *in the finite case, if $0 < p \leq 1$, we say that the tree-questionable-decomposition is p -tight, resp. p -short, resp. p -medium, resp. p -wide, if it is tight, resp. short, resp. medium, resp. wide, and for all pair of vertices the proportion of tight, resp. short, resp. medium, resp. wide, ascending paths having a valid question is at least p ;*
- *we say that the tree-questionable-decomposition is ϵ -tight, resp. ϵ -short, resp. ϵ -medium, resp. ϵ -wide, if it is tight, resp. short, resp. medium, resp. wide, and for all pair of vertices at least one of the tight, resp. short, resp. medium, resp. wide, ascending paths has a valid question.*

From D we define the following parameters:

- *K is the supremum of the widths of the mapping-runs of each node of T . K is called the width of the decomposition (it is a cardinal);*
- *Λ is the supremum of the lengths of the mapping-runs of each node of T . Λ is called the mapping length of the decomposition (it is an ordinal). Hence, we have only $(V, \leq K, \leq \Lambda)$ -mapping-runs on the nodes of T ;*
- *α is the depth of the tree T . α is called the structural depth of the decomposition (it is an ordinal);*
- *β is the depth of the extended tree T' obtained by replacing each internal node by a path of nodes, one for each mapping of the mapping-run associated to the original node. If this sequence is empty, we delete the node, which may be a problem if it is the root; hence, we keep the root and the leaves, and if the depth isn't a limit ordinal, we redefine β by the previous ordinal of the depth ("depth - 1"). This depth is the supremum of the lengths of the mapping-runs obtained by concatenating the mapping-runs on a leaf-root path. β is called the mapping depth of the decomposition (it is an ordinal);*
- *γ is the depth of the extended tree T'' obtained by replacing each internal node by a path of nodes, one for each mapping of the mapping-run associated to the original node. If this sequence is empty, we keep at least one node (This is a little like we had an "identity" mapping on each node without mapping. This identity mapping sends each vertex to the unique vertex of a binary structure of*

cardinality 1 and doesn't create a (first) difference.). γ is called the combined depth of the decomposition (it is an ordinal) (We dropped the notion of logical depth, that was corresponding by lack of precision sometimes to mapping depth and sometimes to combined depth, and whose name was not very well found.);

- Δ is the supremum of the degrees of the nodes of T . Δ is called the degree of the decomposition (it is a cardinal).
- σ is the supremum of the cardinals of the sets of leaves of T associated to each vertex by ll^{-1} . σ is called the individual surjectivity of the decomposition (it is a cardinal).
- Σ is the cardinal of the set of leaves of T . Σ is called the global surjectivity of the decomposition (it is a cardinal ;).

We then say that D is a $(K; \Lambda; \alpha, \beta, \gamma; \Delta; \sigma, \Sigma)$ -tree-questionable-decomposition. Note that we can have $\alpha < \beta$ or $\alpha > \beta$, and that we always have $\gamma \geq \max(\alpha, \beta)$ and $\Lambda \leq \beta$.

The original definition given in Lyaudet (2019) corresponds to 1-tight tree-questionable-decompositions.

The introductory example (Example 1.1) can be reworded:

Lemma 2.7. *Let S be an (infinite) binary structure of cardinality n , it has a $(2; 1; 2, 1, 2; \Delta; \sigma, \Sigma)$ -tree-questionable-decomposition of unbounded degree that is 1-tight, ϵ -tight, ϵ -short, ϵ -medium, and ϵ -wide. $\Delta = \frac{n \times (n-1)}{2}$ in the finite case, and $\Delta = n$ in the infinite case. $\sigma = n - 1$ and $\Sigma = n \times (n - 1)$ in the finite case, and $\sigma = n$ and $\Sigma = n$ in the infinite case.*

We could complexify further the variants of tree-questionable-decompositions to have or not a default adjacency type. Normally, the modular decompositions and the tree-questionable-decompositions do not treat differently the distinct adjacency types, including the non-adjacency in the case of graphs. But we could break this symmetry to say that a pair of vertices for which no ascending path has a question must have the default adjacency type. This additional choice is interesting only if we allow ascending paths without questions. Indeed, for the “1” types, the variants with and without default adjacency type are equivalent; because it is sufficient to add mappings of width 2, or more if available, at the end of the mapping-run of the root of the decomposition to fix all the missing adjacencies with the default adjacency type. This addition of mappings increases the mapping length, the mapping depth, and the combined depth by a logarithm of the number of vertices n in the finite case, or by at most the first ordinal of cardinality n in the infinite case. (If n is at most the cardinal of the powerset of a set of cardinality n' , we can take the first ordinal of cardinality n' .) We can consider that this is negligible and it justifies that the tree-questionable-decompositions with a default adjacency type are only truly useful to “ ϵ ” and “ p ” types. Moreover, it is interesting only for some classes of binary structures where the adjacencies that are not of the default type are sparse; and in this article where we consider arbitrary binary structures most of the time, we can always assume that all the used adjacency types are distinct of the default adjacency type, even distinct between them. We leave it to the reader to reread all the lemmas, corollaries, and theorems of this article to see that the results are still valid for the decompositions with a default adjacency type.

3 Comparisons

We obtain all the comparisons for the 8 main types: 1-tight, 1-short, 1-medium, 1-wide, ϵ -tight, ϵ -short, ϵ -medium, and ϵ -wide. We provide some ideas (lemmas and counter-examples) for other cases.

Lemma 3.1. *If a tree-questionable-decomposition is short, then it is tight. If a tree-questionable-decomposition is medium, then it is short. If a tree-questionable-decomposition is wide, then it is medium.*

Lemma 3.2. *If a tree-questionable-decomposition is 1-tight, resp. 1-short, resp. 1-medium, resp. 1-wide, then it is ϵ -tight, resp. ϵ -short, resp. ϵ -medium, resp. ϵ -wide.*

In the finite case, if $0 < p_1 < p_2 \leq 1$, and if a tree-questionable-decomposition is p_2 -tight, resp. p_2 -short, resp. p_2 -medium, resp. p_2 -wide, then it is ϵ -tight and p_1 -tight, resp. ϵ -short and p_1 -short, resp. ϵ -medium and p_1 -medium, resp. ϵ -wide and p_1 -wide.

Lemma 3.3. *If a tree-questionable-decomposition is 1-short, then it is 1-tight. If a tree-questionable-decomposition is 1-medium, then it is 1-short. If a tree-questionable-decomposition is 1-wide, then it is 1-medium.*

Lemma 3.4. *If a tree-questionable-decomposition is ϵ -short, then it is ϵ -tight. If a tree-questionable-decomposition is ϵ -medium, then it is ϵ -short. If a tree-questionable-decomposition is ϵ -wide, then it is ϵ -medium.*

Proof:

For all pair of vertices, by Lemma 2.5, if there is a short, resp. medium, resp. wide, ascending path with a valid question, then it is included in a leaf-root path. Hence, this leaf-root path has a question, and, by Lemma 3.1, this question is valid. Only the invalid questions separate the ϵ -tight, ϵ -short, ϵ -medium, and ϵ -wide types. Hence, the given inclusions follow from a second application of Lemma 3.1. ■

All the other inclusions for the 8 main types are false; here are counter-examples.

The example decomposition (Example 1.1) that motivated this article is 1-tight, ϵ -tight, ϵ -short, ϵ -medium, and ϵ -wide, but neither 1-short, nor 1-medium, nor 1-wide.

Example 3.5. *The following decomposition is of all main types except 1-wide. Consider a binary structure with 2 vertices x, y , such that x is adjacent to y . We make a cherry with two leaves for x and y ; and we fix the adjacency between x and y by a mapping on the root of the cherry. (In the rest of the article, we will call this construction an (x, y) -adjacent cherry, independantly of the fact that x and y are adjacent or not in the decomposed graph. If the mapping on the root of the cherry fixes a non-adjacency between x and y , we say it is an (x, y) -non-adjacent cherry. If there is no mapping on the root of the cherry, we say it is an (x, y) -free cherry.) We duplicate this cherry. We link the roots of these two cherries to the true root of the decomposition, that has no mapping.*

Example 3.6. *The following decomposition is ϵ -tight, ϵ -short, ϵ -medium, and ϵ -wide, but neither 1-tight, nor 1-short, nor 1-medium, nor 1-wide. Consider a binary structure with two vertices x, y , such that x is adjacent to y . We make an (x, y) -adjacent cherry. We make a second (x, y) -free cherry. We link the roots of the two cherries to the true root of the decomposition that has no mapping.*

Example 3.7. *The following decomposition is 1-tight, and ϵ -tight, but neither 1-short, nor ϵ -short, nor 1-medium, nor ϵ -medium, nor 1-wide, nor ϵ -wide. Consider a binary structure with 3 vertices x, y and z , such that x is adjacent to z ; and otherwise everything is non-adjacent. We make an (x, y) -non-adjacent cherry. We link the root of this cherry to the true root of the decomposition. This true root has also a repeated leaf-son x , a repeated leaf-son y , and a leaf-son z . The mapping-run of the true root fixes x adjacent to y and z (but this is covered by the cherry between x and y for the types 1-tight, and ϵ -tight), then fixes y (and x) non-adjacent to z .*

Example 3.8. *The following decomposition is 1-tight, ϵ -tight, 1-short, ϵ -short, but neither 1-medium, nor ϵ -medium, nor 1-wide, nor ϵ -wide. Consider a binary structure with 3 vertices x, y and z , such that x is adjacent to z ; and otherwise everything is non-adjacent. We make an (x, y) -non-adjacent cherry. We link the root of this cherry to the true root of the decomposition. This true root has also a repeated leaf-son x and a leaf-son z . The mapping-run of the true root fixes x adjacent to y and z (but this is covered by the cherry between x and y for the types 1-tight, ϵ -tight, 1-short, and ϵ -short), then fixes y (and x) non-adjacent to z .*

Example 3.9. *The following decomposition is 1-tight, ϵ -tight, 1-short, ϵ -short, 1-medium, and ϵ -medium, but neither 1-wide, nor ϵ -wide. Consider a binary structure with 3 vertices x, y and z , such that x is adjacent to z ; and otherwise everything is non-adjacent. We make an (x, y) -non-adjacent cherry. We duplicate this cherry and we link the two cherries. We link the union of the two cherries to the true root of the decomposition. This true root has also a leaf-son z . The mapping-run of the true root fixes x adjacent to y and z (but this is covered by the cherries between x and y for the types 1-tight, ϵ -tight, 1-short, ϵ -short, 1-medium, and ϵ -medium), then fixes y (and x) non-adjacent to z .*

We gave 6 counter-examples for the inclusions. If you take the time to draw the Hasse diagram, and to put the separation lines linked to the counter-examples, you will see that the last 4 counter-examples are sufficient.

Here is a complement to see that all the combinations are possible. The types have 2 ladders: one for 1, one for ϵ , with 4 levels (tight, short, medium, wide), one level in ladder 1 implies at least the same level in ladder ϵ . Hence, there is 14 combinations of types according to these ladders:

1. ϵ -tight,
2. ϵ -short,
3. ϵ -medium,
4. ϵ -wide,

5. 1-tight, ϵ -tight,
6. 1-tight, ϵ -short,
7. 1-tight, ϵ -medium,
8. 1-tight, ϵ -wide,
9. 1-short, ϵ -short,
10. 1-short, ϵ -medium,
11. 1-short, ϵ -wide,
12. 1-medium, ϵ -medium,
13. 1-medium, ϵ -wide,
14. 1-wide, ϵ -wide,

It is possible to obtain all these combinations allowed by the lemmas. The combination 14 doesn't need a counter-example, since it doesn't separate any type. The combination 4 corresponds to counter-example 3.6 above. The combination 5 corresponds to counter-example 3.7 above. The combination 8 corresponds to (counter-)example 1.1 above. The combination 9 corresponds to counter-example 3.8 above. The combination 12 corresponds to counter-example 3.9 above. The combination 13 corresponds to counter-example 3.5 above. Here are the counter-examples for the 7 other combinations.

Consider a binary structure with 2 vertices x, y , such that x is adjacent to y . All the decompositions counter-examples are combinations of at most two branches: one " ϵ " branch, and one "1" branch.

We call " ϵ -tight" branch an (x, y) -adjacent cherry, followed by a node that has also two leaf-sons for x and y , and that has a mapping that fixes a non-adjacency between x and y . We call " ϵ -short" branch an (x, y) -adjacent cherry, followed by a node that has also a leaf-son for x , and that has a mapping that fixes a non-adjacency between x and y . We call " ϵ -medium" branch two (x, y) -adjacent cherries, followed by a node that links them, and that has a mapping that fixes a non-adjacency between x and y . We call " ϵ -wide" branch an (x, y) -adjacent cherry.

We call "1-nothing" branch an (x, y) -free cherry. We call "1-tight" branch an (x, y) -adjacent cherry, followed by a node that has also two leaf-sons for x and y , and that has no mapping. We call "1-short" branch an (x, y) -adjacent cherry, followed by a node that has also a leaf-son for x , and that has no mapping. A simple (x, y) -adjacent cherry alone can be used as a "1-medium" branch, and a "1-wide" branch, on top of " ϵ -wide" branch.

Following these two branches, each decomposition has only a root without any mapping.

For the combination 1, one take an " ϵ -tight" branch and a "1-nothing" branch. For the combination 2, one take an " ϵ -short" branch and a "1-nothing" branch. For the combination 3, one take an " ϵ -medium" branch and a "1-nothing" branch. For the

combination 4, one take an “ ϵ -wide” branch and a “1-nothing” branch. For the combination 5, one take an “ ϵ -tight” branch (and a “1-tight” branch). For the combination 6, one take an “ ϵ -short” branch and a “1-tight” branch. For the combination 7, one take an “ ϵ -medium” branch and a “1-tight” branch. For the combination 8, one take (an “ ϵ -wide” branch and) a “1-tight” branch. For the combination 9, one take an “ ϵ -short” branch (and a “1-short” branch). For the combination 10, one take an “ ϵ -medium” branch and a “1-short” branch. For the combination 11, one take (an “ ϵ -wide” branch and) a “1-short” branch. For the combination 12, one take an “ ϵ -medium” branch (and a “1-medium” branch). For the combination 13, one take an “ ϵ -wide” branch and a “1-medium” branch. For the combination 14, one take just an (x, y) -adjacent cherry.

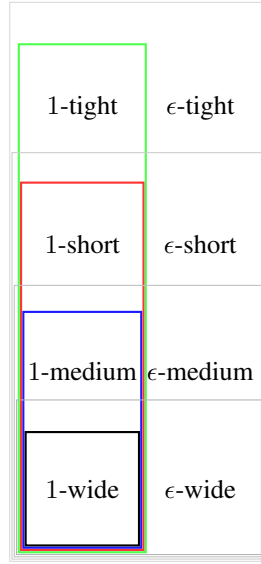


Figure 5: Inclusion of the 8 main types

Lemma 3.3 has no equivalent in the finite case and with the intermediate proportions of valid questions.

In particular, p -wide or p -medium doesn't imply p -short or p -tight. Just consider for example a decomposition D of a binary structure with two vertices x and y ; D has two branches: one deep and one “wide”; on the root of the deep branch there is a valid question. The deep branch is a comb that starts with a cherry for x and y , then adds r leaves with x and adds r internal nodes. The “wide” “branch” contains s cherries for x and y . The last internal node of the deep branch and all the cherries of the “wide” “branch” are linked to the root of D . D is $\frac{1}{s+1}$ -tight, $\frac{1}{s+1}$ -short, $\frac{r+1}{r+s+1}$ -medium, and $\frac{r+1}{r+s+2}$ -wide.

To show that p -short doesn't imply p -tight, it is enough to add r leaves with y under the r internal nodes of the previous example. D becomes $\frac{1}{s+1}$ -tight, $\frac{r+1}{r+s+1}$ -short, $\frac{r+1}{r+s+1}$ -medium, and $\frac{r+1}{r+s+2}$ -wide.

To see that p -wide doesn't imply p -medium, the example is barely more compli-

cated. Just consider for example a decomposition D of a binary structure with two vertices x and y ; D is made of a balanced binary tree of depth l plus a last level made of cherries for x and y ; the root of this binary tree has a valid question; this binary tree is then followed toward the root of D by a comb that adds s leaves with x and adds s internal nodes. The last added internal node is the root of D . D is $\frac{2^l}{2^l+s}$ -medium and $\frac{2^{l+1}-1}{2^{l+1}-1+s}$ -wide. Hence, for $s = 2^{l+1}$, we get about $\frac{1}{3}$ -medium and $\frac{1}{2}$ -wide. And for $s = 2^{l+h}$, we get $\frac{1}{2^{h+1}}$ -medium, and $\frac{1-\frac{1}{2^{l+1}}}{2^{h-1}+1-\frac{1}{2^{l+1}}}$ -wide. Thus the coefficient to go from wide to medium is $\frac{1}{2^{h+1}} \times \frac{2^{h-1}+1-\frac{1}{2^{l+1}}}{1-\frac{1}{2^{l+1}}} = \frac{2^{h-1}+1-\frac{1}{2^{l+1}}}{2^{h+1}} \times \frac{1}{1-\frac{1}{2^{l+1}}}$. For all $l \geq 0$, $\frac{2^{h-1}+1-\frac{1}{2^{l+1}}}{2^{h+1}}$ goes to $\frac{1}{2}$ when we increase h ; $\frac{1}{1-\frac{1}{2^{l+1}}}$ goes to 1 when we increase l ; hence, we can make it go to $\frac{1}{2}$ by superior values.

This last counter-example can be analysed to show that p -wide implies $\frac{p}{2}$ -medium.

Lemma 3.10. *In the finite case, if $0 < p \leq 1$ and if a tree-questionable-decomposition is p -wide, then it is $\frac{p}{2}$ -medium, and even $\frac{p}{2-p}$ -medium.*

Proof:

We will show that the counter-example above is optimal. Let us remark first that a counter-example on a binary structure with more than two vertices can be pruned into a counter-example with only two vertices, since the minimum ratio/coefficient must be attained for some pair of vertices. Let us call them x and y . Since it brings nothing to have a mapping that fixes a wrong adjacency type between x and y , we can assume that for all node, we have either no mapping, or we have a unique mapping that fixes the valid adjacency type for all paths coming from junction points below in the tree. Let vmp be the number of medium ascending paths with a valid question, wmp be the number of medium ascending paths without question, vwp be the number of wide ascending paths with a valid question, and wwp be the number of wide ascending paths without question. The counter-example is $\frac{vmp}{vmp+wmp}$ -medium and $\frac{vwp}{vwp+wwp}$ -wide. The goal of the counter-example is to minimise $\frac{vmp \times (vwp+wwp)}{(vmp+wmp) \times vwp}$. Since a medium junction point is also a wide junction point, the medium ascending paths without question cannot be more than the wide ascending paths without question: $wmp \leq wwp$. Hence, the closest wmp gets to wwp , the more the ratio decreases. It shows that the parameter s of our counter-example is optimal, since it meets equality. We can simplify the ratio by $\frac{vmp \times (vwp+wwp)}{(vmp+wmp) \times vwp} = \frac{vmp \times vwp + vmp \times wwp}{vmp \times vwp + wmp \times vwp} = \frac{vmp \times vwp + wmp \times vmp}{vmp \times vwp + wmp \times vwp}$. Likewise, $vmp \leq vwp$, but this time we want to get the biggest possible value of vwp compared to vmp in order to decrease the ratio. Let us look at the structure of the subtrees with a valid question. By minimality of the counter-example, apart from its leaves, all its nodes are in CT. Hence all the internal nodes are wide junction points. And if a node that has only leaves as sons is of degree more than 2, we see immediately that we can delete at least one son. Thus, if the leaves are level 0, the property to be in CT and to have degree two is true at level 1. By induction on the levels, if a node above in the tree has more than two sons, we can separate them to make a cherry whose root is a wide but not medium junction point, and

at most one leaf of the cherry is a medium junction point (if the separated node was), the second leaf being always a wide junction point. Hence, it increases vwp by one, but not vmp . This is also the case, if only one of the two sons is a leaf and that we replace it by an x, y cherry. Thus, the “valid branches” of an optimal counter-example are always binary trees ended by x, y cherries, whose only nodes that are medium junction points are those of level 1. If there is more than one such branch, we can move one to “graft” it on another such branch in place of a cherry; it will remove one medium junction point but it may be compensated by a new one after pruning the original attachment point of the branch. Hence, our counter-example with a valid binary tree followed by an invalid comb is optimal. And it is sufficient to see that we never meet the ratio $\frac{vmp}{vwp} > \frac{1}{2}$, but that we can get as close as we want with balanced binary trees. Here is the proof for the bound $\frac{p}{2-p}$: $p = \frac{vwp}{vwp+wwp} \Leftrightarrow \frac{p}{vwp} = \frac{1}{vwp+wwp} \Leftrightarrow \frac{vwp}{p} = vwp + wwp \Leftrightarrow wwp = \frac{vwp}{p} - vwp = \frac{vwp-p \times vwp}{p} = vwp \times \frac{1-p}{p}$; $\frac{vmp}{vmp+wwp} \geq \frac{vmp}{vmp+wwp} \geq \frac{\frac{vwp}{2}}{\frac{vwp}{2}+wwp} = \frac{vwp}{vwp+2wwp} = \frac{vwp}{vwp+2 \times (vwp \times \frac{1-p}{p})} = \frac{1}{1+2 \times \frac{1-p}{p}} = \frac{p}{p+2 \times (1-p)} = \frac{p}{2-p}$. ■

4 Results on the degree

Lemma 4.1. *If a tree-questionable-decomposition is bijective, it is of all types (the 8 main types, and the other types for the finite case).*

Proof:

$CT(T_x, T_y)$ is a path. There is only one mapping-run, and it has a valid question. ■

Lemma 4.2. *Let S be a finite binary structure. Let D be a 1-short, resp. ϵ -short, resp. 1-medium, resp. ϵ -medium, resp. 1-wide, resp. ϵ -wide, $(K; \Lambda; \alpha, \beta, \gamma; \Delta; \sigma, \Sigma)$ -tree-questionable-decomposition of S with degree $\Delta > 2$. If we replace each node of degree more than 2 by a binary subtree whose root has the mapping-run of the node, we obtain a 1-short, resp. ϵ -short, resp. 1-medium, resp. ϵ -medium, resp. 1-wide, resp. ϵ -wide, $(K; \Lambda; \alpha', \beta, \gamma'; 2; \sigma, \Sigma)$ -tree-questionable-decomposition of S .*

Proof:

We can separate each node of degree more than 2 into a cherry. We put no mapping on the two leaves of the cherry. The questions will thus not be on these leaves.

All the proof hereafter works for all pair of vertices $\{x, y\}$.

We do not create any question, but we create/move junction points toward the leaves only if the separated node was a junction point. Indeed, if a short junction point appears on a leaf of the cherry, then one of its sons is in T_x but not in T_y , and another one is in T_y but not in T_x ; since these sons were sons of the root of the cherry, this root was indeed a short junction point. Similarly, if a medium junction

point appears on a leaf of the cherry, then at least one of its sons is in T_x but not in T_y (or in T_y but not in T_x); since this son was a son of the root of the cherry, this root was indeed a medium junction point. Last, if a wide junction point appears on a leaf of the cherry, then at least two of its sons are in the union of T_x and T_y ; since these sons were sons of the root of the cherry, this root was indeed a wide junction point.

Hence, the new short, resp. medium, resp. wide, ascending paths have a question if and only if there was a question starting from the original short, resp. medium, resp. wide, junction point. Thus, the condition to always have a question on each short, resp. medium, resp. wide, ascending path and that it is valid is fulfilled for 1-short, resp. 1-medium, resp. 1-wide. And the condition to have a question on at least one short, resp. medium, resp. wide, ascending path and that it is valid is fulfilled for ϵ -short, resp. ϵ -medium, resp. ϵ -wide. ■

Theorem 4.3. *Let S be a finite binary structure. A 1-short, resp. ϵ -short, resp. 1-medium, resp. ϵ -medium, resp. 1-wide, resp. ϵ -wide, $(K; \Lambda; \alpha, \beta, \gamma; \Delta; \sigma, \Sigma)$ -tree-questionable-decomposition can be converted into a 1-short, resp. ϵ -short, resp. 1-medium, resp. ϵ -medium, resp. 1-wide, resp. ϵ -wide, $(K; \Lambda; \leq \alpha \times \lceil \lg(\Delta) \rceil, \beta, \leq \gamma \times \lceil \lg(\Delta) \rceil; 2; \sigma, \Sigma)$ -tree-questionable-decomposition.*

Proof:

We just need to apply Lemma 4.2 with balanced binary subtrees (We separate each node of degree more than 2 into a cherry in a balanced way.). ■

If we look at the number of leaves Σ of the decomposition, the construction with balanced binary subtrees is not optimal when the depths of the subtrees corresponding to the sons of a node are very different, in particular if there is only one son of large depth and the other sons are leaves. Because it only gives a multiplicative bound for the depth increase. Here is a reasoning to see it. Each node has been replaced by a balanced binary subtree, and the maximal depth is obtained along one of the leaf-root paths. Let $P = (u_0, u_1, \dots, u_{\alpha^-})$, $\alpha^- \leq \alpha$ be the path before transformation, and $P' = (u_0, u'_1, \dots, u'_{\alpha'})$ be the corresponding leaf-root path, among those of maximum length after transformation (The leaf u_0 hasn't been modified.). The new depth is equal to $\sum_{i=1}^{\alpha^-} \lceil \lg(d(u_i)) \rceil = \sum_{i=1}^{\alpha^-} (1 + \lceil \lg(d(u_i)) \rceil - 1) < \alpha^- + \sum_{i=1}^{\alpha^-} \lg(d(u_i)) \leq \alpha + \lg(\prod_{i=1}^{\alpha^-} d(u_i))$. If we look at the worst case, where the quantity $\prod_{i=1}^{\alpha^-} d(u_i)$ is maximised, we see it happens for a path where each node has roughly the same number of leaf-sons. $\alpha' < \alpha + \lg(\prod_{i=1}^{\alpha^-} \lceil \frac{\Sigma}{\alpha^-} \rceil) \leq \alpha + \lg(\prod_{i=1}^{\alpha} \lceil \frac{\Sigma}{\alpha} \rceil) < \alpha + \lg(\frac{\Sigma + \alpha}{\alpha} \alpha) = \alpha \times (1 + \lg(\frac{\Sigma + \alpha}{\alpha}))$.

However, one can show that an additive bound is possible:

Theorem 4.4. *Let S be a finite binary structure. A 1-short, resp. ϵ -short, resp. 1-medium, resp. ϵ -medium, resp. 1-wide, resp. ϵ -wide, $(K; \Lambda; \alpha, \beta, \gamma; \Delta; \sigma, \Sigma)$ -tree-questionable-decomposition with $\Delta > 2$ can be converted into a 1-short, resp. ϵ -short, resp. 1-medium, resp. ϵ -medium, resp. 1-wide, resp. ϵ -wide, $(K; \Lambda; \leq \alpha + \lceil \lg(\Sigma - \alpha + 1) \rceil - 1, \beta, \leq \gamma + \lceil \lg(\Sigma - \alpha + 1) \rceil - 1; 2; \sigma, \Sigma)$ -tree-questionable-decomposition.*

Proof:

We only need to apply Lemma 4.2 with binary subtrees that minimise the global depth increase. For this, we process the nodes of degree more than 2 by level order: first the nodes whose sons are all leaves, etc. When a node u has more than two sons, we need to start by making a cherry that joins two of its sons whose subtrees depth are minimal. Then we repeat this construction. Indeed, let u_1 be a son of u (whose subtree is) of minimal depth, and let u_2 be a son of u (whose subtree is) of minimal depth if we exclude u_1 . Assume that the construction wasn't respected, and that we joined u_1 with a son (resp. a join of sons) u_3 of depth larger than u_2 , and that we joined u_2 with a son (resp. a join of sons) u_4 of depth larger than u_1 . Let pr be the depth function. We have currently $pr(u_1, u_3) = pr(u_3) + 1$ and $pr(u_2, u_4) = pr(u_4) + 1$. If instead, we join u_1 and u_2 , and u_3 and u_4 , we obtain $pr(u_1, u_2) = pr(u_2) + 1$ and $pr(u_3, u_4) = \max(pr(u_3), pr(u_4)) + 1$; so maybe we just need to switch also the roots of the 2 cherries so that $\max(pr(u_3), pr(u_4)) + 1$ ends on the deepest side between $pr(u_3) + 1$ and $pr(u_4) + 1$. Hence, there is always among the binary subtrees that minimise the global depth, one that is obtained by taking this rule to join first the sons of smallest depths.

In order to find the worst case in terms of structural depth increase, there are only two parameters that must not increase: α , the start structural depth, and Σ , the number of leaves. We will say that a son of a node is an eldest son if its subtree has maximum depth among the sons and its number of leaves in its subtree is maximum among the sons whose subtree has maximum depth. We will search for such a worst case that minimises the sum of the degrees minus 2 of the nodes of level at least 2, and that maximises the number of leaves in the eldest sons under each node of level at least 2. And we will prove by induction on the levels that it is a comb whose nodes are of degree 2, extended by a star/a fan of larger degree. We will call this comb plus a fan, a large fan.

For the nodes of level 0, they are leaves and the large fan is only a single leaf.

For the nodes of level 1, all the sons are leaves, there is nothing to change. And the large fan is only a fan without a comb above.

For the nodes of higher level, by induction hypothesis, we can assume that all the subtrees of their sons are large fans. At least one son is a leaf, because we lose nothing in depth if the first join among its sons is between some son and some leaf (if a son node of minimal depth is not a leaf, we lose nothing by taking all its leaves except one to put them in the fan of some other son). Also, at most one son is a leaf, because if we had two leaf-sons, we would lose nothing by replacing them by a cherry, and it would decrease the sum of the degrees minus 2. If one of the sons is a leaf and another one is a cherry, we can replace them by a node with 3 leaf-sons. Hence, we have always exactly one leaf-son and the other sons have at least 3 leaves each in their subtrees. At level j , we lose also nothing by stretching a large fan of a son by taking leaves in its fan to extend its comb. Hence without loss of generality, either the large fan is of depth $j - 1$, or its fan has only two leaves left. Moreover, except for the eldest son ("THE son" is justified hereinafter by an

arbitrary choice in case of ties that will only reinforce the “preference” for this son in the later transformations), either the large fan is of depth $j - 1$ and its fan has $2^l + 1$ leaves for some l (because at most $2^l - 1$ more leaves in the fan can go without depth increase loss to the fan of a chosen eldest son), or its fan has only two leaves left. Since we have exactly one leaf-son, and if we have also a son of depth less than $j - 1$ with two leaves in its fan, we take such a son with minimal depth, we can join it with the leaf-son to decrease the number of sons, hence the sum of the degrees minus 2. Hence, it stops only when all the sons have depth $j - 1$ except for a leaf-son. Otherwise, the sum of the degrees minus 2 of the nodes of level at least 2 is not minimal or the number of leaves of the eldest son is not maximal. Consider the smallest number of leaves in a fan $2^{l_m} + 1$ besides the leaf-son (Note that $2 = 2^0 + 1$ and that $l_m = 0$ is possible.). If we have three sons with $2^{l_m} + 1$ leaves in their fan, the first join will join one with the leaf-son, and the second join will join the other two. But a merge of the two sons of the second join would have gave one only son with $2^{l_m+1} + 2 + (j - 2)$ leaves in its fan, hence at least one more leaf to give to the eldest son. Thus, we have at most two sons with $2^{l_m} + 1$ leaves in their fan. If there are two such sons, the first two joins join one of these sons with the leaf-son, then join the result of the first join with the second son. We would have obtained the same result in terms of depth increase, if we had merged the two sons to have one son with $2^{l_m+1} + 1$ leaves in the fan (giving the additional leaves to the eldest son), and joining the new son to the leaf-son. Hence, we can assume that there is only one son with $2^{l_m} + 1$ leaves in its fan. But then, the unique join between a son with $2^{l_m} + 1$ leaves and the leaf-son will not bring more depth increase to a son with at least $2^{l_m+1} + 1$ leaves than the leaf-son alone. Hence, by repeating this reasoning, we can put all the leaves in the fan of the eldest son except for the leaf-son.

Finally, it is sufficient to see that if the totality of a worse case is a large fan of depth α , then the maximum depth increase with this algorithm is at most $\lceil \lg(\Sigma - \alpha + 1) \rceil - 1$. ■

Corollary 4.5. *Let S be a finite binary structure of cardinality n , it has an ϵ -tight, ϵ -short, ϵ -medium, and ϵ -wide $(2; 1; \lceil \lg(n) + \lg(n - 1) \rceil, 1, \lceil \lg(n) + \lg(n - 1) \rceil; 2; n - 1, n \times (n - 1))$ -tree-questionable-decomposition.*

Proof:

$$\lceil \lg(\frac{n \times (n-1)}{2}) \rceil + 1 = \lceil \lg(n) + \lg(n - 1) - \lg(2) \rceil + 1 = \lceil \lg(n) + \lg(n - 1) \rceil. \quad \blacksquare$$

Corollary 4.6. *Modulo a logarithmic increase on the depth, a bijective tree-questionable-decomposition of a finite binary structure can be converted into a bijective tree-questionable-decomposition, (binary/)/with degree 2.*

Thus, the degree doesn't matter much when we search for bijective tree-questionable-decompositions that are balanced in a logarithmic way.

Lemma 4.7. *In a binary tree, a short junction point is a leaf.*

Corollary 4.8. *If a tree-questionable-decomposition of degree 2 is tight, resp. 1-tight, resp. ϵ -tight, resp. p -tight, then it is short, resp. 1-short, resp. ϵ -short, resp. p -short. If a tree-questionable-decomposition of degree 2 is p -short, then it is p -tight.*

5 Pruning

5.1 Goal and definitions

In this section, we want to delete some leaves and nodes to remove useless redundancies from a tree-questionable-decomposition.

Definition 5.1 ((Light/Heavy) Pruning). *A pruning of a tree-questionable-decomposition is done by deleting some leaves; then by induction on the levels*

- *we delete every internal node that has no more son left,*
- *and, in case of heavy pruning, if possible, we merge every son that has only one son left with its son, by concatenating their mapping-runs, which is not always possible in the infinite case because a node can have a son without maximal node;*

finally, we limit the mappings to the new leaf labels of the internal nodes; and if it creates mappings that have only one image vertex, we delete these mappings. If some nodes of degree 1 ends up with an empty mapping-run, we also delete them, except if it is the root.

Clearly, none of the 8 parameters $(K; \Lambda; \alpha, \beta, \gamma; \Delta; \sigma, \Sigma)$ increases with a light pruning. And only the mapping length Λ can increase with a heavy pruning. Normally, we should only consider heavy prunings and normalised decompositions, where there is no superfluous degree 1 node. Because with unnecessary degree 1 nodes, we can always decrease Λ to 1, and make equal the structural depth and the combined depth, which is slightly stupid. But, at the moment, we miss a result showing that the mapping depth increases only slightly if all the adjacencies are fixed as soon as the vertices are here in the subtree. (If that was the case, we could delete a node of degree 1 that is not on a limit ordinal, and delete also all its mapping-run instead of concatenating it with the son node.) In fact, two things counter this result:

- for the decompositions that are not of type “1”, it is legitimate to exploit their power to not fix all the time all the adjacencies;
- otherwise, there is also the example of a comb without adjacency/without mapping, followed by a mapping-run of logarithmic length on the root to fix all the adjacencies, for a clique for example; we see well that in this case, it is possible to rebalance the tree before fixing the adjacencies to correct this problem of exponential increase of mapping depth, but generalising this rebalancing to obtain a true correct bound doesn’t seem to be simple.

The function ll was defined for all nodes of a tree-questionable-decomposition. We can also extend it to subtrees of the decomposition by taking the union of its values on the nodes of the subtree. It is then easy to see that the value of ll for the subtree of a son (valid definition for the finite and infinite cases) is equal to the value of ll on a son-node (that will always exist in the finite case). When the nodes or sons (subtrees) are called u and u_i , we denote $F = ll(u)$ and $F_i = ll(u_i)$. We will denote also:

- $F^2 = \{\{x, y\}; x, y \in F\}$,
- $F_i^2 = \{\{x, y\}; x, y \in F_i\}$,
- $F^2SV = \{\{x, y\}; x, y \in F \text{ and either } x = y, \text{ or they have at least one valid question on their leaf-root paths starting from the subtree of } u \}$,
- $F_i^2SV = \{\{x, y\}; x, y \in F_i \text{ and either } x = y, \text{ or they have at least one valid question on their leaf-root paths starting from the subtree of } u_i \}$,
- $F^2V = \{\{x, y\}; x, y \in F, x \neq y \text{ and they have at least one valid question on their leaf-root paths starting from the subtree of } u \}$,
- $F_i^2V = \{\{x, y\}; x, y \in F_i, x \neq y \text{ and they have at least one valid question on their leaf-root paths starting from the subtree of } u_i \}$.

Definition 5.2 (Root of the pruning). *The unique node of the following lemma is called the root of the pruning.*

Lemma 5.3. *For every pruning, there is a node u such that the pruning only occurs under u , and such that F is not modified. If we take such a node that is minimal in the tree/in the partial order, it is unique.*

Proof:

When we prune while keeping at least one leaf per vertex, if u is the root of the decomposition, then F is not modified. If the pruning occurs in many sons of the root, the root is clearly the root of the pruning. Otherwise, there is a unique son u_1 that contains the pruning, we can thus descend in this son, if F_1 is not modified. Since we consider partial orders that are well-founded, we always end on a unique root of the pruning. ■

5.2 Pruning and junction points

Lemma 5.4. *No pruning creates wide junction points.*

Proof:

The union tree that wraps the common tree is pruned; hence, there can be two branches in the union tree after pruning only if they were already there before pruning. ■

Lemma 5.5. *The only junction points that can be modified by a pruning are under the root of the pruning, this root being included. This root can become a tight junction point, but it cannot become a short, medium, or wide junction point, if it wasn't before.*

Lemma 5.6. *No pruning that deletes totally a son of the root of the pruning creates a short or medium(, or wide) junction point.*

5.3 Pruning when a son covers another son

Lemma 5.7. *If a node u of a tree-questionable-decomposition of any type has 2 sons u_1 and u_2 , such that $F_1 = F_2 = \{x\}$, then we can delete the leaf-son u_1 . (If someone especially vicious gave you the decomposition, it may be that u_1 is a subtree that is not reduced to a leaf-son, but you still have the right to delete it ;).)*

Proof:

No junction point is modified, even in the tight case (cf. Lemma 5.5 for the other cases); we have the same ascending paths with the same valid questions. ■

Lemma 5.8. *If a node u of a 1-tight, 1-short, 1-medium, or 1-wide tree-questionable-decomposition has 2 sons u_1 and u_2 , such that $F_1 \subseteq F_2$, then we can delete all the subtree rooted at u_1 .*

Proof:

Indeed, we cannot create a junction point that would create an ascending path without valid question or with an invalid question, and we keep at least one ascending path with valid question. ■

Similarly:

Lemma 5.9. *If a node u of a 1-tight, 1-short, 1-medium, 1-wide, ϵ -tight, ϵ -short, ϵ -medium, ϵ -wide, or p -tight tree-questionable-decomposition has 2 sons u_1 and u_2 , such that $F_1 = \{x\} \subseteq F_2$, then we can delete the leaf-son/all the subtree rooted at u_1 .*

Proof:

Indeed, for any pair, no junction point apart from u is modified. For the tight types of decompositions, u is always junction point or not for the same pairs.

For the other types, it may be that u is no more a junction point, whilst it was one before. If there was no valid question on the ascending path starting from u , it makes no difference. But whatever, for every pair that loses u as a junction point, there is at least another junction point under another son of u ; hence, this junction point has necessarily a valid question on his ascending path, either at the beginning, or in the common part with the ascending path starting from u . ■

Lemma 5.10. *If a node u of an ϵ -tight, ϵ -short, ϵ -medium, or ϵ -wide tree-questionable-decomposition has 2 sons u_1 and u_2 , such that $F_1^2 SV \subseteq F_2^2 SV$, then we can delete all the subtree rooted at u_1 .*

Because again, we cannot create a junction point that would create an ascending path with an invalid question, and we keep at least one ascending path with valid question.

From Lemma 5.8, we obtain also:

Lemma 5.11. *If a node u of a 1-tight, 1-short, 1-medium, or 1-wide tree-questionable-decomposition has the sons $(u_i)_{i \in \delta}$ such that $\exists j, F_j \supseteq \bigcup_{i \in \delta, i \neq j} F_i$, then we can delete all the subtrees of $u_i, i \in \delta, i \neq j$.*

From Lemma 5.10, we obtain also:

Lemma 5.12. *If a node u of an ϵ -tight, ϵ -short, ϵ -medium, or ϵ -wide, tree-questionable-decomposition has the sons $(u_i)_{i \in \delta}$ such that $\exists j, F_j^2 SV \supseteq \bigcup_{i \in \delta, i \neq j} F_i^2 SV$, then we can delete all the subtrees of $u_i, i \in \delta, i \neq j$.*

These two last lemmas will be useful to bound the depth.

5.4 Partial pruning below a son

Lemma 5.13. *If a 1-wide tree-questionable-decomposition has two leaves mapped to the same vertex, we can delete/prune one of these leaves.*

Proof:

A leaf pruning doesn't create any wide junction point (but a wide junction point can become a medium junction point; it excludes this result for 1-medium; and a medium junction point can become a short junction point; it excludes this result for 1-short). And since the leaf is redundant, there is still at least one wide junction point for each pair of vertices. Since with 1-wide type, every wide junction point generates a valid question, the property to still have a valid question is fulfilled. ■

Corollary 5.14. *The 1-wide non-bijective tree-questionable-width is equal to the bijective tree-questionable-width.*

Lemma 5.15. *Let D be a 1-tight, resp. 1-short, resp. 1-medium, resp. 1-wide, tree-questionable-decomposition. Let u be a node of D with sons $(u_i)_{i \in \delta}$. Let u_j be one of its sons. Let X be such that $X \subseteq \bigcup_{i \in \delta, i \neq j} F_i$. When we prune all the leaves corresponding to the set of vertices X in the subtree of the son u_j , we obtain a 1-tight, resp. 1-short, resp. 1-medium, resp. 1-wide, tree-questionable-decomposition, if for every pair of vertices $\{x, y\}$ such that $\{x, y\} \cap X \neq \emptyset$, if the node u is a tight, resp. short, resp. medium, resp. wide, junction point after pruning, then it was a tight, resp. short, resp. medium, resp. wide, junction point before pruning.*

Proof:

Let $\{x, y\}$ be a pair of vertices. Their junction points in the sons other than u_1 of u have not been modified, neither their questions that are thus valid. Similarly for the junction points outside of $\text{IS}(u)$ that are not modified just as their questions, since $X \subseteq \bigcup_{i \in \delta, i \neq j} F_i$. The only junction points that are maybe modified are those in the subtree of u_1 and u itself. Hence, we can assume that $\{x, y\} \subseteq F$.

If $\{x, y\} \cap X = \emptyset$, then no junction point between them was created, and no junction point disappeared if we consider that the nodes merging in the pruning definition yield morally the “same” node.

Otherwise, all the junction points between $\{x, y\}$ in the subtree of u_1 were removed. There is no junction point creation, except if u becomes one, which may be a problem if it has no valid question in the “1” case, or if it has an invalid question.

Finally, since $X \subseteq \bigcup_{i \in \delta, i \neq j} F_i$, for every pair of vertices $\{x, y\}$ such that $\{x, y\} \cap X \neq \emptyset$: If $\exists k \in \delta \setminus \{j\}, \{x, y\} \subseteq F_k$, we have a valid junction point between them in F_k ; hence, we just need to avoid that u wasn't a junction point before, and becomes one after. If $\nexists k \in \delta \setminus \{j\}, \{x, y\} \subseteq F_k$, then, since $\{x, y\} \subseteq F$, u is a junction point (tight, hence short, etc.) after pruning; hence here again, we just need to avoid that u wasn't a junction point before, and becomes one after, *i.e.* the case where it wasn't a junction point. ■

Theorem 5.16. *The 1-medium non-bijective tree-questionable-width is equal to the bijective tree-questionable-width.*

Proof:

If a 1-medium tree-questionable-decomposition has two leaves corresponding to the same vertex, we have a node u with 2 sons u_1, u_2 that have intersecting leaf labels $F_1 \cap F_2 \neq \emptyset$. If we apply Lemma 5.15, with $X = F_1 \cap F_2$ and $u_j = u_2$, it just remains to show that for all pair of vertices $\{x, y\}$ such that $\{x, y\} \cap X \neq \emptyset$ if the node u is a junction point after, then it was a junction point before. (The wide case is trivial here, because it is obvious that if we have at least 2 sons in CT after pruning, then, *a fortiori*, we have also at least 2 sons in CT before pruning.) If u is a medium junction point after, then, without loss of generality, one of its sons has a leaf for x but not for y after, hence also before for x . If this son had leaves for x and for y before, then it means that we are in the case where $x \notin X, y \in X$ and the corresponding son is $u_j = u_2$, hence $\{x, y\} \subseteq F_2$. But then $x \notin F_1, y \in X \subseteq F_1$ hence u was a medium junction point before. ■

From Lemma 5.15, we also deduce:

Lemma 5.17. *Let D be an ϵ -tight, resp. ϵ -short, resp. ϵ -medium, resp. ϵ -wide, tree-questionable-decomposition. Let u be a node of D with sons $(u_i)_{i \in \delta}$. Let u_j be one of its sons. Let X be such that $X \subseteq \bigcup_{i \in \delta, i \neq j} F_i$. When we prune all the leaves corresponding to the set of vertices X in the subtree of the son u_j , we obtain an ϵ -tight, resp. ϵ -short, resp. ϵ -medium, resp. ϵ -wide, tree-questionable-decomposition, if*

- for every pair of vertices $\{x, y\}$ such that $\{x, y\} \cap X \neq \emptyset$, if the node u is a tight, resp. short, resp. medium, resp. wide, junction point after pruning, then it was a tight, resp. short, resp. medium, resp. wide, junction point before pruning,
- and $F_j^2 V \cap \{\{x, y\}; x \in X, y \in F\} \subseteq \bigcup_{i \in \delta, i \neq j} F_i^2 V$.

5.5 Pruning when some sons cover another son

Lemma 5.18. *If a node u of a 1-short, 1-medium, or 1-wide tree-questionable-decomposition has the sons $(u_i)_{i \in \delta}$ such that $\exists j, F_j \subseteq \bigcup_{i \in \delta, i \neq j} F_i$, then we can delete all the subtree of u_j .*

Proof:

Since $F_j \subseteq \bigcup_{i \in \delta, i \neq j} F_i$, the deletion of the subtree of u_j doesn't create junction points above in the tree. And u isn't a short, medium or wide junction point after pruning, if it wasn't already before. Hence, we can conclude by Lemma 5.15. ■

Corollary 5.19. *In a 1-short tree-questionable-decomposition without possible pruning, for each son, there is a vertex that only this son has in the leaves of its subtree.*

Lemma 5.20. *If a node u of a 1-tight tree-questionable-decomposition has the sons $(u_i)_{i \in \delta}$ such that $\exists j, F_j^2 \subseteq \bigcup_{i \in \delta, i \neq j} F_i^2$, then we can delete all the subtree of u_j .*

Proof:

$F_j^2 \subseteq \bigcup_{i \in \delta, i \neq j} F_i^2$ implies in particular that $F_j \subseteq \bigcup_{i \in \delta, i \neq j} F_i$. Since $F_j \subseteq \bigcup_{i \in \delta, i \neq j} F_i$, the deletion of the subtree of u_j doesn't create junction points above in the tree. Since $\{\{x, y\}; x, y \in F_j, x \neq y\} \subseteq \bigcup_{i \in \delta, i \neq j} \{\{x, y\}; x, y \in F_i, x \neq y\}$, for each pair $\{x, y\} \subseteq F_j^2$, $\exists i \in \delta, x, y \in F_i \Leftrightarrow \exists i \in (\delta \setminus \{j\}), x, y \in F_i$. Hence, u is a tight junction point after pruning if and only if it is one before. Thus, we can conclude by Lemma 5.15. ■

Corollary 5.21. *In a 1-tight tree-questionable-decomposition without possible pruning, for each son, there is a vertex or a pair of vertices that only this son has in the leaves of its subtree.*

From Lemma 5.4, we have:

Corollary 5.22. *In an ϵ -wide tree-questionable-decomposition, we can select at most $\frac{n \times (n-1)}{2}$ internal nodes, each of these internal nodes containing at least one question that fixes the adjacency between two vertices, then select at most $n \times (n-1)$ leaves whose vertices use the selected questions, and prune all the other leaves.*

Lemma 5.23. *If a node u of an ϵ -short, ϵ -medium, or ϵ -wide tree-questionable-decomposition has the sons $(u_i)_{i \in \delta}$ such that $\exists j, F_j^2 SV \subseteq \bigcup_{i \in \delta, i \neq j} F_i^2 SV$, then we can delete all the subtree of u_j .*

Corollary 5.24. *In an ϵ -short tree-questionable-decomposition without possible pruning, for each son, there is either a vertex that only this son has in the leaves of its subtree, or there is a pair of vertices that only this son has in the leaves of its subtree with a valid question.*

Lemma 5.25. *If a node u of an ϵ -tight tree-questionable-decomposition has the sons $(u_i)_{i \in \delta}$ such that $\exists j, F_j^2 \subseteq \bigcup_{i \in \delta, i \neq j} F_i^2 \wedge F_j^2 V \subseteq \bigcup_{i \in \delta, i \neq j} F_i^2 V$, then we can delete all the subtree of u_j .*

Corollary 5.26. *In an ϵ -tight tree-questionable-decomposition without possible pruning, for each son, there is either a vertex that only this son has in the leaves of its subtree, or there is a pair of vertices that only this son has in the leaves of its subtree, or there is a pair of vertices that only this son has in the leaves of its subtree with a valid question.*

6 Bounds on the parameters

6.1 Width

The width K of the tree-questionable-decompositions is easy to bound. For all binary structure with at least two elements, the width must be at least 2. And it's useless to have a width above n , since a single mapping to a binary structure of cardinality n is enough to decompose everything.

Lemma 6.1. *In a tree-questionable-decomposition without useless redundancy, $2 \leq K \leq n$.*

6.2 Mapping length

The mapping length Λ of the tree-questionable-decompositions is harder to bound. For all binary structure with at least two elements, the mapping length must be at least one. It is also the case for bigger binary structures, like independent sets, cliques, or cographs on a binary decomposition tree. If the width is K and $n \leq K$, a mapping length of 1 is enough for all binary structure, hence *a fortiori* if some of its adjacencies are alf (see Lyaudet (2020)). Each mapping of width K cuts properly the binary structure into at least K parts separated by alf-adjacencies.

For $K = 2$, we see immediately that each mapping increases by at least one the number of parts separated by alf-adjacencies. Hence $n - 1$ mappings are enough to end up with singletons: $\Lambda_{\max}(n, 2) \leq n - 1$. And of course, $\Lambda_{\max}(n) \leq \Lambda_{\max}(n, 2)$.

As an example of construction that gives $\Lambda_{\max}(n, 2)$, we can take a binary structure where a vertex has the same adjacency type with all the other vertices, but no edge non-incident to this vertex uses this type of adjacency; then, among the other vertices, one vertex has the same adjacency type with all the other remaining vertices, but no edge non-incident to this vertex uses this type of adjacency, etc.

For $K = 3$, we can nevertheless have to use a mapping of width 2 from time to time. Since the width 3 is enough, either we obtain singletons after each cut by mapping, or

we can cut recursively by a mapping of width at most 3. Assume that some part can be cut into 2, and that one of its subparts can also be cut into 2, then we can directly cut it into 3 subparts with a single mapping of width 3; we gain 2 parts in one step. Assume that 2 or 3 parts can be cut by a mapping of width 2 at a given step, then we can directly do this cut in 4 or 6 parts with a single mapping of width 3; we gain 2 or 4 parts in one step; in particular, it is true for parts of size 2, hence cuttable into 2 singletons. Thus, the worse case is to have a unique part of size 2, for a gain of 1 part in one step. But this case can happen only once, if we wait for the end to cut all parts of size 2. And the worst case, if we ignore this part of size 2, is to have to cut into 2 then a subpart into 3, whilst the other subpart is a singleton, since we gain 3 parts in two steps: $\frac{3}{2}$ gained in one step. The case where we must cut into 2, then the two subparts into 3 has a gain of 5 parts in 3 steps: $\frac{5}{3}(> \frac{3}{2})$ gained in one step. Since in total we must gain $n - 1$ parts, we have: $\Lambda_{\max}(n, 3) \leq 1 + \lceil \frac{2}{3} \times (n - 2) \rceil$. We can see that this bound is optimal if we take the following construction: a first cut into 2 of which a singleton followed by a second cut into 3 of which 2 singletons, we repeat these two levels l times, then we end by a cut into 2. It gives $n = 3 \times l + 2$ cut into singletons with $2 \times l + 1$ steps, and $1 + \lceil \frac{2}{3} \times ((3 \times l + 2) - 2) \rceil = 1 + \lceil \frac{2}{3} \times (3 \times l) \rceil = 1 + 2 \times l$.

For $K = 4$, the previous reasoning on the cuts of parts of size 2 applies, including for grouping with cuts into 3 into a single cut into 4. Cutting into 2 then into 4 yields a gain of 4 parts in 2 steps, which is equivalent to the gain of 2 parts in 1 step when we cut into 3, the difference being only on a last cut into 2 after a cut into 2 then into 4. Hence, we obtain $\Lambda_{\max}(n, 4) \leq 1 + \lceil \frac{1}{2} \times (n - 2) \rceil$.

For $K = 5$, we can now group two cuts into 3 (parallel or consecutive) into a single cut into 5. Cutting into 2 then into 5 yields a gain of 5 parts in 2 steps, but allows a last cut into 2 after a cut into 2 then into 5. And it is well a worse case than to alternate cuts into 3 then into 4 which yields also a gain of 5 parts in 2 steps, and allows a last cut into 3 after a cut into 3 then into 4. Hence, we obtain $\Lambda_{\max}(n, 5) \leq 1 + \lceil \frac{2}{5} \times (n - 2) \rceil$.

For $K = 2 \times k, k \geq 3, k \in \mathbb{N}$, we can group two cuts into at most k (parallel or consecutive) into a single cut into K , similarly for all couple of cuts into at most l and $2 \times k + 1 - l$ parts. Cutting into 2 then into K yields a gain of K parts in 2 steps, but allows a last cut into 2 after a cut into 2 then into K . And it is well a worse case than to have cuts into $k + 1$ or an alternation less wide than 2 and K because of the final gain. Hence, we obtain $\Lambda_{\max}(n, K) \leq 1 + \lceil \frac{2}{K} \times (n - 2) \rceil$.

For $K = 2 \times k + 1, k \geq 3, k \in \mathbb{N}$, we can group two cuts into at most $k + 1$ (parallel or consecutive) into a single cut into K , similarly for all couple of cuts into at most l and $2 \times k + 2 - l$ parts. Cutting into 2 then into K yields a gain of K parts in 2 steps, but allows a last cut into 2 after a cut into 2 then into K . And it is well a worse case than to have cuts with alternation less wide than 2 and K because of the final gain. Hence, we obtain $\Lambda_{\max}(n, K) \leq 1 + \lceil \frac{2}{K} \times (n - 2) \rceil$.

Thus, we obtain:

Theorem 6.2. *In a tree-questionable-decomposition without useless redundancy, $1 \leq \Lambda \leq n - 1$, $1 \leq \Lambda \leq 1 + \lceil \frac{2}{K} \times (n - 2) \rceil$ if $K \geq 2$.*

We see that: $K \times \Lambda \leq 2 \times K + 2 \times (n - 2) = 2 \times (K + n - 2)$.

6.3 Depths

The three depths are linked together, because a bound on the structural depth multiplied by the mapping length is our sole general mean to bound the other depths at the moment. And this bound is attained for all the binary structures that generalise the cographs, either by decomposing with a binary tree or with a unique internal node adjacent to all the leaves. Maybe a better bound exists when the structural depth, the width, and the mapping length are larger than their minimum simultaneously.

In this subsection, we assume that we did heavy pruning to avoid to keep uselessly nodes of degree 1.

Thanks to Lemma 5.11, we see immediately that for the types 1-tight, 1-short, 1-medium and 1-wide, it is useless to have a structural depth above $n - 1$.

Lemma 6.3. *In a 1-tight, 1-short, 1-medium or 1-wide tree-questionable-decomposition without useless redundancy, $1 \leq \alpha \leq n - 1$, $\Lambda \leq \beta \leq \gamma \leq \Lambda \times \alpha \leq \Lambda \times n - 1$. And even $\beta \leq \gamma \leq \frac{n \times (n-1)}{2}$, because the number of vertices increases by at least one at each level, and $\Lambda \leq n - 1$.*

The last bound $\beta \leq \gamma \leq \frac{n \times (n-1)}{2}$ is attained, in particular for binary structures where all the adjacencies are distinct and the decomposition of Lemma 7.6 in Lyaudet (2019).

Thanks to Lemma 5.12, we see immediately that for the types ϵ -tight, ϵ -short, ϵ -medium and ϵ -wide, it is useless to have a structural depth above $\frac{n^2}{2} - 1$.

Lemma 6.4. *In an ϵ -tight, ϵ -short, ϵ -medium or ϵ -wide tree-questionable-decomposition without useless redundancy, $1 \leq \alpha \leq \frac{n^2}{2} - 1$, $\Lambda \leq \beta \leq \gamma \leq \Lambda \times \alpha \leq \Lambda \times (\frac{n^2}{2} - 1)$.*

It is possible that a better bound exists for the “ ϵ ” types. On the other hand, we still have no bound for the “ p ” types, which is embarrassing.

6.4 Degree

In order to simplify, we consider in this subsection that all the binary structures have at least 3 vertices.

Thanks to Corollary 5.19, we have:

Lemma 6.5. *In a 1-short, 1-medium or 1-wide tree-questionable-decomposition without useless redundancy, $2 \leq \Delta \leq n$.*

Thanks to Corollary 5.21, and the fact that an exclusive singleton is disjunct from the pairs, we have:

Lemma 6.6. *In a 1-tight tree-questionable-decomposition without useless redundancy, $2 \leq \Delta \leq \frac{n \times (n-1)}{2}$.*

Thanks to Corollary 5.24, and the fact that an exclusive singleton is disjunct from the pairs with valid questions, we have:

Lemma 6.7. *In an ϵ -short, ϵ -medium or ϵ -wide tree-questionable-decomposition without useless redundancy, $2 \leq \Delta \leq \frac{n \times (n-1)}{2}$.*

Thanks to Corollary 5.26, and the fact that an exclusive singleton is disjunct from the pairs with or without valid questions, and that a pair with valid question excludes the same pair with or without valid questions, we have:

Lemma 6.8. *In an ϵ -tight tree-questionable-decomposition without useless redundancy, $2 \leq \Delta \leq \frac{n \times (n-1)}{2}$.*

6.5 Surjectivity

In this subsection, we give a few results on the maximal number of leaves Σ that one can obtain in a tree-questionable-decomposition after pruning. By the results of the previous section, this maximum (and minimum) number of leaves is equal to the number of vertices for the 1-medium and 1-wide widths.

Here is an optimal example that is valid for the 1-short tree-questionable-decompositions.

Example 6.9. *At the root, we have the n vertices to distribute between two sons. We associate each son to a distinct subset of $n - 1$ vertices. Let x, y be the excluded vertices. Hence, the root is a short or tight junction point only for $\{x, y\}$, which we solve by a mapping of width 2. We repeat this at the level below with 4 subsets with $n - 2$ vertices, etc. Thus, we obtain a tree of depth $n - 1$, all internal nodes have degree 2. We obtain $\Sigma = 2^{n-1}$ as a possible value for a 1-short tree-questionable-decomposition of width 2 (or more). We have also $\sigma = 2^{n-2}$ if we keep a same common vertex in all the final pairs.*

This example is optimal for the 1-short tree-questionable-decompositions, because:

- if we have only two sons below each node, clearly, we should give them the maximum possible number of common vertices, and we cannot do more than this example, by Lemma 5.8,
- if we have more than two sons below each node, by Lemma 5.18, we cannot do better than “ $\delta \times \text{possible_leaves}(n - \delta + 1)$ ”; and since “ $\text{possible_leaves}(i + 1) \geq 2 \times \text{possible_leaves}(i)$ ”, increasing the degree yields an exponential loss.

Theorem 6.10. *In a 1-short tree-questionable-decomposition without useless redundancy, $1 \leq \sigma \leq 2^{n-2}$, $n \leq \Sigma \leq 2^{n-1}$.*

For the 1-tight width, here is an example:

Example 6.11. *At the root, we have all the n vertices to distribute in the sons. If the degree isn't bounded, we can associate 3 sons to a distinct subset of $n - 1$ vertices. We repeat this at the level below with 9 subsets with $n - 2$ vertices, etc. Thus, we obtain a tree of depth $n - 1$, the nodes at level 1 have degree 2, those at level 2 and above have degree 3. Only the nodes at level 1 have a mapping to fix an adjacency. We obtain $\Sigma = 2 \times 3^{n-2}$ as possible value for a 1-tight tree-questionable-decomposition of width 2 (or more), since only the nodes at level 1 are tight junction points. We have also $\sigma = 3^{n-2}$ if we keep a same common vertex in all the final pairs.*

Thanks to Lemma 5.20, we avoid the same construction as Example 6.11 with $n - 1$ sons, then $n - 2$ sons, etc. that would have yield $n!$ leaves. And thanks to Lemma 5.15, we can even go further, because these three sons at each level contain many redundancies.

For the 1-tight width, here is an optimal example:

Example 6.12. *At the root, we have all the n vertices to distribute in the sons. If the degree isn't bounded, we can associate 3 sons of which 2 to a distinct subset of $n - 1$ vertices, and the third is the root of a cherry to fix the adjacency between the two vertices excluded from the two main sons. We repeat this at the level below with 4 subsets with $n - 2$ vertices and 2 cherries, etc. Thus, we obtain a tree of depth $n - 1$, the nodes at level 1 have degree 2, those at level 2 and above have degree 3. Only the nodes at level 1 have a mapping to fix an adjacency. There is exactly $2^{n-2} - 1$ nodes of degree 3. And in a regular tree of degree 3 with k internal nodes, the number of leaves is $3 + 2 \times (k - 1)$. Thus, we obtain $\Sigma = 2 \times (3 + 2 \times (2^{n-2} - 1 - 1)) = 2 \times (3 + 2^{n-1} - 4) = 2^n - 2$ as possible value for a 1-tight tree-questionable-decomposition of width 2 (or more), since only the nodes at level 1 are tight junction points. We have also $\sigma = 2^{n-2}$ if we keep a same common vertex in all the final pairs of the two main sons.*

This example shows that, for the 1-tight tree-questionable-decompositions, the maximal global surjectivity more than doubles for each additional vertex beyond 2, and the maximal individual surjectivity doubles or maybe more than doubles for each additional vertex beyond 2.

Lemma 6.13. *Among the 1-tight tree-questionable-decompositions that maximise the global surjectivity and the individual surjectivity, there is always one such that:*

- *for every pair of vertices under a node that is not the root of a cherry, there is a son that contains this pair of vertices,*
- *no node that is not the root of a cherry has a leaf-son,*
- *the width K has value 2, the mapping length Λ and the mapping depth β have value 1, and only the roots of the cherries fix adjacencies; in particular, every adjacency is fixed as soon as possible, and not in nodes above,*
- *a node has degree 2 if and only if it is the root of a cherry,*
- *every vertex under a son belongs to at least one pair exclusive to this son,*
- $\Delta \leq 3$.

Proof:

Let u be a node with the vertices F in the sons below him. Clearly, even if it implies to increase the degree, for every pair in F^2 , there is a son of u that contains that pair; because except if u is already itself the root of a cherry, we can add a cherry-son that fixes the adjacency for this pair. (We need to fix the adjacency in order to have a cherry that is not prunable, because otherwise, it brings nothing. And outside of case “1”, it is not without consequences, because it could allow other pruning elsewhere

in the tree, *cf.* Lemma 5.10 in particular.) If the cherry-son has two vertices under him that were in leaf-sons of u , the surjectivities are not modified; if the cherry-son has a vertex under him that was in a leaf-son of u , the global surjectivity increases by 1, and the individual surjectivity is not modified or increases by 1; if the cherry-son has no vertex under him that was in a leaf-son of u , the global surjectivity increases by 2, and the individual surjectivity is not modified or increases by 1. It implies that no node such that $|F| \geq 3$ has a leaf-son; because either this leaf-son can be pruned by Lemma 5.9, or it adds pairs to F^2 that can allow the addition of another cherry-son; in particular, it applies to nodes of degree 2 that have only one leaf-son and to nodes of larger degree. Hence, no node that is not the root of a cherry has a leaf-son.

Thus, only the roots of the cherries are tight junction points and need to have a mapping that fixes an adjacency, and the width has value 2, the mapping length and the mapping depth have value 1. By doing this, we have deleted all the mappings on nodes that are not the roots of cherries, and we have an “A.S.A.P.” (As Soon As Possible) tree-questionable-decomposition. This step of cleaning of internal mappings is crucial, because it forbids to go back with a partial or total pruning of one of the added cherries, without creating a tight junction point on some internal node and thus a tight ascending path without valid question; which is incompatible with 1-tight type. Similarly, since we consider the 1-tight type and that we haven’t modified F for any internal node, if another type of pruning is possible after, it was possible before.

If a node of degree 2 has no leaf-son and none of its two sons can prune the other, they both have an exclusive vertex, and hence a pair of vertices that can have a cherry addition as before. Hence, a node has degree 2 if and only if it is the root of a cherry.

Let us apply now Lemma 5.15. We see that the nodes of degree more than 2 are not tight junction points. The lemma applies unless the pruning makes that some pair of vertices is no more in the leaves of at least one son. Hence, any vertex below a son belongs to at least one pair that is exclusive to this son.

Let us assume that we have a better example with nodes of degree more than 3. Let u be such a node, none of its sons has a set of vertices included in the set of vertices of another son by Lemma 5.8. If we merge two sons, we obtain a son with at least one vertex more, which is strictly better for global surjectivity and at least as good for individual surjectivity. This merging of two sons can allow prunings, that is if the new son contains all the pairs of vertices that were exclusive to another son before the merging (another son distinct of the merged sons), *cf.* Corollary 5.21. Assume for a contradiction that we have at least 4 sons such that the merging of any pair of sons, say u_1 et u_2 , contains all the pairs of vertices that were exclusive to another son, say u_3 , before the merging. Since any vertex below a son belongs to at least one pair that is exclusive to this son, $F_3 \subseteq (F_1 \cup F_2)$. Similarly, if another node u_i is in the same case as u_3 , we have $F_i \subseteq (F_1 \cup F_2)$. Hence, except if $(F_1 \cup F_2) = F$, we can take all these (at least 3) sons and put them in a sub-branch that becomes a new son, and decrease the degree of u . If we cannot, $(F_1 \cup F_2) = F$; hence if we cannot for every pair of sons u_i and u_j , then

$(F_i \cup F_j) = F, \forall i, j \in \delta, i \neq j$. Thus, if a vertex of F is not under a son, it is under all the other sons, and at most one son doesn't have a given vertex of F . Hence, all the exclusive pairs, say $\{x, y\}$ that are neither in a first son, nor a second son must be in a third son. Thus, we can prune all the other sons beyond 3. ■

It only remains to show that the cut into 3 proposed in Example 6.12 maximises the surjectivities among the cuts into 3.

For the individual surjectivity, our construction keeps a vertex in the two largest branches among the three at each level. If we try to put it in the 3 branches, each branch must have an exclusive vertex to disallow any pruning. Thus, we obtain at most $\sigma_{max}(n-2) \times 3$ instead of $\sigma_{max}(n-1) \times 2$; since $\sigma_{max}(n) \geq 2 \times \sigma_{max}(n-1)$, we have $\sigma_{max}(n-2) \times 2 \leq \sigma_{max}(n-1) \Leftrightarrow \sigma_{max}(n-2) \times 3 \leq \sigma_{max}(n-1) \times 1, 5 < \sigma_{max}(n-1) \times 2$; we cannot do better than the proposed example.

For the global surjectivity, assume that the smallest branch contains at least 3 vertices, each one belongs to a pair exclusive to this branch. But then, there is at least two exclusive pairs that cannot share more than one common vertex. If they don't have a common vertex, the two largest branches must avoid at least two vertices each; and since we obtain at most $\Sigma_{max}(n-2) \times 3$ instead of $\Sigma_{max}(n-1) \times 2 + 2$, and that $\Sigma_{max}(n) > 2 \times \Sigma_{max}(n-1) > 1,5 \times \Sigma_{max}(n-1)$, the global surjectivity decreases. If they have a common vertex x , the largest branch can avoid only x , but the second largest branch must avoid at least two vertices (if it avoids x , it must avoid another one in order to not contain the largest branch; and otherwise it must avoid the other two vertices of the pairs exclusive to the third branch). And since we obtain at most $\Sigma_{max}(n-1) + \Sigma_{max}(n-2) \times 2$ instead of $\Sigma_{max}(n-1) \times 2 + 2$, and that $\Sigma_{max}(n) > 2 \times \Sigma_{max}(n-1)$, we have $\Sigma_{max}(n-2) \times 2 < \Sigma_{max}(n-1) \Rightarrow \Sigma_{max}(n-2) \times 2 < \Sigma_{max}(n-1) + 2 \Leftrightarrow \Sigma_{max}(n-1) + \Sigma_{max}(n-2) \times 2 < \Sigma_{max}(n-1) \times 2 + 2$, and the global surjectivity decreases.

Theorem 6.14. *In a 1-tight tree-questionable-decomposition without useless redundancy, $1 \leq \sigma \leq 2^{n-2}$, $n \leq \Sigma \leq 2^n - 2$.*

Let us recall that in Lyaudet (2019), we showed that one can decompose all binary structure with a comb of depth $n - 1$ and degree 2 that is a bijective tree-questionable-decomposition. The interest of these examples is mostly to demonstrate the limits of our current pruning techniques, and of our results to decrease the degree, since here no parameter yields nice results.

Example 6.12 doesn't work for the 1-short tree-questionable-decompositions, because we must have a mapping on each internal node and by Lemma 5.18 we can prune the non-main son.

Finding the maximum number of leaves in the decompositions of “ ϵ ” type seems to be an interesting problem of extremal combinatorics, because we think that Δ^α yields a really too large bound compared to the real worst case. Although the ϵ -wide case is already resolved by Corollary 5.22. Another unresolved problem is to minimise only σ in the examples 6.9 and 6.12, because we easily see that for a little larger n , we can have $\sigma \leq 2^{n-3}$.

6.6 Assessment of the current bounds

To sum up, here are the minimum values for the parameters:

Parameters	Types
$(2; 1; 1, 1, 1; 2; 1, n)$	All

And here are our current knowledge on the maximum values of the parameters, relatively to our pruning results:

Parameters	Types
$(n; n-1; n-1, \frac{n \times (n-1)}{2}, \frac{n \times (n-1)}{2}; n; 1, n)$	Bijjective
$(n; n-1; n-1, \frac{n \times (n-1)}{2}, \frac{n \times (n-1)}{2}; n; 2^{n-2}, 2^{n-1})$	1-short
$(n; n-1; n-1, \frac{n \times (n-1)}{2}, \frac{n \times (n-1)}{2}; \frac{n \times (n-1)}{2}; 2^{n-2}, 2^n - 2)$	1-tight
$(n; n-1; \frac{n^2}{2} - 1?, (n-1) \times (\frac{n^2}{2} - 1)?, (n-1) \times (\frac{n^2}{2} - 1)?; \frac{n \times (n-1)}{2}; n-1, n \times (n-1))$	ϵ -wide
$(n; n-1; \frac{n^2}{2} - 1?, (n-1) \times (\frac{n^2}{2} - 1)?, (n-1) \times (\frac{n^2}{2} - 1)?; \frac{n \times (n-1)}{2}; 2^{n-2}?, 2^n - 2?)$	ϵ -tight, short or medium
$(n; n-1; ?, ?, ?, ?, ?)$	"p"

Here above, we omitted the bounds $\Lambda \leq 1 + \lceil \frac{2}{K} \times (n-2) \rceil$.

7 Universal decompositions comparisons

The examples of Subsection 6.5 work for all binary structure, and give us points of reference for trade-off between the distinct parameters of tree-questionable-decompositions. Here is a comparison of 8 examples that decomposes all binary structure. Each of these examples is better on at least one parameter, except for the examples 6.11, 6.12 that can be greatly improved:

Example 7.1. *At the root, we have all the n vertices to distribute in the sons. We partition the set of vertices into 3 parts A, B, C of size at most $\lceil \frac{n}{3} \rceil$. We can associate 3 sons to a distinct subset of at most $\lceil \frac{2 \times n}{3} \rceil$ vertices. We repeat this at the level below with 9 subsets with $\lceil \frac{4 \times n}{9} \rceil$ vertices, etc. The nodes at level 1 have degree 2, those at level 2 and above have degree 3. Only the nodes at level 1 have a mapping to fix an adjacency.*

To calculate the depth of this example of universal decomposition, we see that $r_0 = n, r_{i+1} = r_i \times \frac{2}{3}$ implies $r_i = n \times \frac{2^i}{3}$, and hence $r_i \leq 1 \Leftrightarrow i \geq \log_{\frac{3}{2}}(n)$. If we set $n_0 = n, n_{i+1} = \lceil n_i \times \frac{2}{3} \rceil$, since $\lceil \lceil r \rceil \times \frac{2}{3} \rceil \leq \lceil r \times \frac{2}{3} + \frac{2}{3} \rceil$, we have $n_i \leq n \times \frac{2^i}{3} + \sum_{j=0}^{i-1} \frac{2^j}{3} \leq n \times \frac{2^i}{3} + \frac{1-2^i}{1-\frac{2}{3}} < n \times \frac{2^i}{3} + \frac{1}{\frac{1}{3}} = n \times \frac{2^i}{3} + 3$, i.e. $n_i < r_i + 3$. Thus $n_i < 4 \Leftrightarrow n \times \frac{2^i}{3} \leq 1 \Leftrightarrow i \geq \log_{\frac{3}{2}}(n)$. But $n_i < 4 \Leftrightarrow n_i \leq 3$, since n_i is an integer. And since 2 steps of cutting are needed for a set of 3 vertices, the total depth is at most $2 + \lceil \log_{\frac{3}{2}}(n) \rceil$, of which one level of degree 2 and not 3. The total number of leaves, the global surjectivity, is at most $2 \times 3^{1 + \lceil \log_{\frac{3}{2}}(n) \rceil} < 2 \times 3^{2 + \log_{\frac{3}{2}}(n)} \leq 2 \times 9 \times 3^{\log_{\frac{3}{2}}(n)} \leq 18 \times 3^{\log_{\frac{3}{2}}(n)} \leq 18 \times e^{\ln(3) \times (\frac{\ln(n)}{\ln(\frac{3}{2})})} \leq 18 \times n^{\frac{\ln(3)}{\ln(\frac{3}{2})}} \leq 18 \times n^{2.709511291}$. For the individual surjectivity, it already gives at most $2^{1 + \lceil \log_{\frac{3}{2}}(n) \rceil} < 2^{2 + \log_{\frac{3}{2}}(n)} \leq 4 \times 2^{\log_{\frac{3}{2}}(n)} \leq 4 \times e^{\ln(2) \times (\frac{\ln(n)}{\ln(\frac{3}{2})})} \leq 4 \times n^{\frac{\ln(2)}{\ln(\frac{3}{2})}} \leq 4 \times n^{1.709511291}$. And it also shows that the global surjectivity can be improved to $n \times 4 \times n^{1.709511291} \leq 4 \times n^{2.709511291}$.

Thus, for this example, the computation of the global surjectivity by the mean of the computation of the individual surjectivity is more precise.

(Note that $\frac{\ln(2)}{\ln(\frac{3}{2})} + 1 = \frac{\ln(2)}{\ln(3)-\ln(2)} + 1 = \frac{\ln(2)+\ln(3)-\ln(2)}{\ln(3)-\ln(2)} = \frac{\ln(3)}{\ln(3)-\ln(2)} = \frac{\ln(3)}{\ln(\frac{3}{2})}$.)

One can do even better. We will see it in two steps. The first step is a slight variant, when the second step needs more involved computations. To compute the depth, we see that $r_i \leq 2 \Leftrightarrow n \times \frac{2^i}{3} \leq 2 \Leftrightarrow \frac{n}{2} \times \frac{2^i}{3} \leq 1 \Leftrightarrow i \geq \log_{\frac{3}{2}}(\frac{n}{2})$. Since $n_i < r_i + 3$, $n_i < 5 \Leftrightarrow n \times \frac{2^i}{3} \leq 2 \Leftrightarrow i \geq \log_{\frac{3}{2}}(\frac{n}{2})$. But $n_i < 5 \Leftrightarrow n_i \leq 4$, since n_i is an integer. And since 3 steps of cutting are needed for a set of 4 vertices, the total depth is at most $3 + \lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil$, of which one level of degree 2 and not 3. It

yields for individual surjectivity: $\sigma \leq 2^{2+\lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil} < 2^{3+\log_{\frac{3}{2}}(\frac{n}{2})} \leq 8 \times 2^{\log_{\frac{3}{2}}(\frac{n}{2})} \leq 8 \times e^{\frac{\ln(2) \times (\frac{\ln(n)}{\ln(\frac{3}{2})})}{\ln(\frac{3}{2})}} \leq 8 \times e^{\frac{\ln(2) \times (\frac{\ln(n)-\ln(2)}{\ln(\frac{3}{2})})}{\ln(\frac{3}{2})}} \leq 8 \times e^{\frac{\ln(2) \times (\frac{\ln(n)}{\ln(\frac{3}{2})})}{\ln(\frac{3}{2})}} \div e^{\frac{\ln(2) \times (\frac{\ln(2)}{\ln(\frac{3}{2})})}{\ln(\frac{3}{2})}} \leq 8 \times n^{\frac{\ln(2)}{\ln(\frac{3}{2})}} \div e^{\frac{\ln(2) \times (\frac{\ln(2)}{\ln(\frac{3}{2})})}{\ln(\frac{3}{2})}} \leq 8 \times n^{1.709511291} \div e^{\ln(2) \times 1.709511291} \leq 8 \times n^{1.709511291} \div e^{1.184942932} \leq 8 \times n^{1.709511291} \div 3.22100884 \leq 2.483693898 \times n^{1.709511291}$. Hence $\Sigma \leq 2.483693898 \times n^{2.709511291}$.

If we try to find the optimal bound for the depth of the previous example by computing the first values, we see that it is at best one less than the bound given above, *i.e.* $2 + \lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil$. We will show that the bound $2 + \lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil$ is exact in this third step. A simple script shows that we alternate between ranges of increasing length between this bound and a smaller value. The first values to attain the bound are: 10, 14, 15, 20 à 22, 29 à 34, 43 à 51, 64 à 76, 95 à 115, 142 à 172. This sequence of integers doesn't exist yet in OEIS. If we take the numbers that start a range where this bound is attained, we seem to obtain a subsequence of <https://oeis.org/A003312>, $a(1) = 3$; $a(n+1) = a(n) + \lfloor \frac{a(n)-1}{2} \rfloor$. If we take the numbers that end a range where this bound is attained, we seem to obtain a subsequence of <https://oeis.org/A147788>, $a(n) = \lfloor 2 \times (\frac{3}{2})^n \rfloor$. We easily verify that for $n = 10$ et $n = 14$, for example, we obtain $2 + \lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil$. ($\lceil \frac{2}{3} \times 14 \rceil = 10$, $\lceil \frac{2}{3} \times 10 \rceil = 7$, $\lceil \frac{2}{3} \times 7 \rceil = 5$, $\lceil \frac{2}{3} \times 5 \rceil = 4$, $\lceil \frac{2}{3} \times 4 \rceil = 3$, $\lceil \frac{2}{3} \times 3 \rceil = 2$).

The recurrence formula of range start is $n_{i+1} = n_i + \lfloor \frac{n_i-1}{2} \rfloor$. But, we see immediately that:

- if n_i is even, $\lceil \frac{2}{3} \times (n_i + \lfloor \frac{n_i-1}{2} \rfloor) \rceil = \lceil \frac{2}{3} \times (n_i + \frac{n_i}{2} - 1) \rceil = \lceil n_i - \frac{2}{3} \rceil = n_i$;
- if n_i is odd, $\lceil \frac{2}{3} \times (n_i + \lfloor \frac{n_i-1}{2} \rfloor) \rceil = \lceil \frac{2}{3} \times (n_i + \frac{n_i}{2} - \frac{1}{2}) \rceil = \lceil n_i - \frac{1}{3} \rceil = n_i$.

And moreover, we see that if we take $n_{i+1} = n_i + \lfloor \frac{n_i-1}{2} \rfloor - 1$, we have

- if n_i is even, $\lceil \frac{2}{3} \times (n_i + \lfloor \frac{n_i-1}{2} \rfloor - 1) \rceil = \lceil \frac{2}{3} \times (n_i + \frac{n_i}{2} - 2) \rceil = \lceil n_i - \frac{4}{3} \rceil = n_i - 1$;
- if n_i is odd, $\lceil \frac{2}{3} \times (n_i + \lfloor \frac{n_i-1}{2} \rfloor - 1) \rceil = \lceil \frac{2}{3} \times (n_i + \frac{n_i}{2} - \frac{3}{2}) \rceil = \lceil n_i - 1 \rceil = n_i - 1$.

Hence the recurrence of range start, independently of its initial condition, gives us the smallest next integer such that a balanced cut into two thirds by superior integer value yields the previous number. This is this number that minimises the logarithm with three halves base, hence that maximises the additive constant.

Another recurrence formula is $n_{i+1} = n_i + \lceil \frac{n_i-1}{2} \rceil = n_i + \lfloor \frac{n_i}{2} \rfloor$. But, we see immediately that:

- if n_i is even, $\lceil \frac{2}{3} \times (n_i + \lceil \frac{n_i-1}{2} \rceil) \rceil = \lceil \frac{2}{3} \times (n_i + \frac{n_i}{2}) \rceil = \lceil n_i \rceil = n_i$;
- if n_i is odd, $\lceil \frac{2}{3} \times (n_i + \lceil \frac{n_i-1}{2} \rceil) \rceil = \lceil \frac{2}{3} \times (n_i + \frac{n_i}{2} - \frac{1}{2}) \rceil = \lceil n_i - \frac{1}{3} \rceil = n_i$.

And moreover, we see that if we take $n_{i+1} = n_i + \lceil \frac{n_i-1}{2} \rceil + 1$, we have

- if n_i is even, $\lceil \frac{2}{3} \times (n_i + \lceil \frac{n_i-1}{2} \rceil + 1) \rceil = \lceil \frac{2}{3} \times (n_i + \frac{n_i}{2} + 1) \rceil = \lceil n_i + \frac{2}{3} \rceil = n_i + 1$;
- if n_i is odd, $\lceil \frac{2}{3} \times (n_i + \lceil \frac{n_i-1}{2} \rceil + 1) \rceil = \lceil \frac{2}{3} \times (n_i + \frac{n_i}{2} - \frac{1}{2} + 1) \rceil = \lceil n_i + \frac{1}{3} \rceil = n_i + 1$.

Hence this other recurrence, independently of its initial condition, gives us the largest next integer such that a balanced cut into two thirds by superior integer value yields the previous number. But this value is less than the range ends values that we found.

To prove the exact bound, we need to show that the value of the logarithm with three halves base stays always important enough for the constant part to be able to stay at 2 instead of needing to increase at 3, which is our additive constant proven at the second step. Thus, we need a lower bound on the increase of the logarithm. Moreover, clearly this increase is minimal when we take an integer that is the smallest possible whilst increasing the depth by 1, it corresponds to the recurrence of range start. Let us prove a lower bound on $\lceil \log_{\frac{3}{2}} (\frac{n_i + \lfloor \frac{n_i-1}{2} \rfloor}{2}) \rceil$. We will need a lower bound on terms like $\ln(1-x)$. We use the infinite series expansion when $0 < x < 1$.

$$\begin{aligned}
\ln(1-x) &= - \sum_{k=1}^{\infty} \frac{x^k}{k} \\
&= -x - \frac{x^2}{2} - \sum_{k=3}^{\infty} \frac{x^k}{k} \\
&> -x - \frac{x^2}{2} - \frac{x^3}{3} \times \sum_{k=0}^{\infty} x^k \\
&= -x - \frac{x^2}{2} - \frac{x^3}{3 \times (1-x)}
\end{aligned} \tag{1}$$

We see that for $0 < x \leq a < 1$, we have on top $\frac{x^3}{3 \times (1-x)} < \frac{x^2 \times a}{3 \times (1-a)}$ and hence $\ln(1-x) > -x - \frac{x^2}{2} - \frac{x^2 \times a}{3 \times (1-a)} = -x - x^2 \times (\frac{1}{2} + \frac{a}{3 \times (1-a)})$. Let us keep in memory that for $0 < x \leq a < 1$, $\ln(1-x) > -x - x^2 \times (\frac{1}{2} + \frac{a}{3 \times (1-a)})$.

We deal now with the main sequence of inequalities:

$$\begin{aligned}
\left\lceil \log_{\frac{3}{2}} \left(\frac{n_{i+1}}{2} \right) \right\rceil &= \left\lceil \log_{\frac{3}{2}} \left(\frac{n_i + \lfloor \frac{n_i-1}{2} \rfloor}{2} \right) \right\rceil \\
&\geq \left\lceil \log_{\frac{3}{2}} \left(\frac{n_i + \frac{n_i-1}{2} - 1}{2} \right) \right\rceil \\
&= \left\lceil \log_{\frac{3}{2}} \left(\frac{n_i + \frac{n_i}{2} - \frac{3}{2}}{2} \right) \right\rceil \\
&= \left\lceil \log_{\frac{3}{2}} \left(\frac{3}{2} \times \frac{n_i}{2} - \frac{3}{4} \right) \right\rceil \\
&= \left\lceil \log_{\frac{3}{2}} \left(\frac{3}{2} \times \frac{n_i}{2} \times \left(1 - \frac{\frac{3}{4}}{\frac{3}{2} \times \frac{n_i}{2}} \right) \right) \right\rceil \\
&= \left\lceil 1 + \log_{\frac{3}{2}} \left(\frac{n_i}{2} \right) + \log_{\frac{3}{2}} \left(1 - \frac{1}{n_i} \right) \right\rceil \\
&\geq i + \left\lceil \log_{\frac{3}{2}} \left(\frac{n_1}{2} \right) + \sum_{k=1}^i \log_{\frac{3}{2}} \left(1 - \frac{1}{n_k} \right) \right\rceil \\
&\geq i + \left\lceil \log_{\frac{3}{2}} \left(\frac{n_1}{2} \right) + \sum_{k=1}^{\infty} \log_{\frac{3}{2}} \left(1 - \frac{1}{n_k} \right) \right\rceil \\
&= i + \left\lceil \log_{\frac{3}{2}} \left(\frac{n_1}{2} \right) + \frac{1}{\ln(\frac{3}{2})} \times \sum_{k=1}^{\infty} \ln \left(1 - \frac{1}{n_k} \right) \right\rceil \\
&\geq i + \left\lceil \log_{\frac{3}{2}} \left(\frac{n_1}{2} \right) + \frac{1}{\ln(\frac{3}{2})} \times \left(\sum_{k=1}^{\infty} -\frac{1}{n_k} - \frac{1}{n_k^2} \times \left(\frac{1}{2} + \frac{\frac{1}{n_1}}{3 \times \left(1 - \frac{1}{n_1} \right)} \right) \right) \right\rceil \\
&\geq i + \left\lceil \log_{\frac{3}{2}} \left(\frac{n_1}{2} \right) + \frac{1}{\ln(\frac{3}{2})} \times \left(\sum_{k=1}^{\infty} -\frac{1}{n_k} - \frac{1}{n_k^2} \times \left(\frac{1}{2} + \frac{1}{3 \times (n_1 - 1)} \right) \right) \right\rceil
\end{aligned} \tag{2}$$

Let us note that $n_{i+1} = n_i + \lfloor \frac{n_i-1}{2} \rfloor$ implies:

$$\begin{aligned}
n_{i+1} &> n_i + \frac{n_i - 1}{2} - 1 \\
&= n_i + \frac{n_i}{2} - \frac{3}{2} \\
&= \frac{3}{2} \times (n_i - 1),
\end{aligned} \tag{3}$$

which implies:

$$\begin{aligned}
\frac{n_{i+1}}{n_i} &> \frac{3}{2} \times \frac{n_i - 1}{n_i} \\
&= \frac{3 \times n_1 - 3}{2 \times n_1}.
\end{aligned} \tag{4}$$

Hence

$$\begin{aligned}
\sum_{k=1}^{\infty} \frac{1}{n_k} &< \frac{1}{n_1} \times \sum_{k=0}^{\infty} \left(\frac{1}{\frac{3 \times n_1 - 3}{2 \times n_1}} \right)^k \\
&= \frac{1}{n_1} \times \sum_{k=0}^{\infty} \left(\frac{2 \times n_1}{3 \times n_1 - 3} \right)^k \\
&= \frac{1}{n_1} \times \frac{1}{1 - \frac{2 \times n_1}{3 \times n_1 - 3}} \\
&= \frac{1}{n_1} \times \frac{1}{\frac{3 \times n_1 - 3 - 2 \times n_1}{3 \times n_1 - 3}} \\
&= \frac{1}{n_1} \times \frac{3 \times n_1 - 3}{n_1 - 3} \\
&= \frac{3 \times n_1 - 3}{n_1 \times (n_1 - 3)}.
\end{aligned} \tag{5}$$

Similarly,

$$\begin{aligned}
\sum_{k=1}^{\infty} \frac{1}{n_k^2} &< \frac{1}{n_1^2} \times \sum_{k=0}^{\infty} \left(\frac{1}{\frac{3 \times n_1 - 3}{2 \times n_1}} \right)^{2k} \\
&= \frac{1}{n_1^2} \times \sum_{k=0}^{\infty} \left(\frac{2 \times n_1}{3 \times n_1 - 3} \right)^{2k} \\
&= \frac{1}{n_1^2} \times \frac{1}{1 - \left(\frac{2 \times n_1}{3 \times n_1 - 3} \right)^2} \\
&= \frac{1}{n_1^2} \times \frac{1}{\frac{(3 \times n_1 - 3)^2 - (2 \times n_1)^2}{(3 \times n_1 - 3)^2}} \\
&= \frac{1}{n_1^2} \times \frac{(3 \times n_1 - 3)^2}{9 \times n_1^2 - 18 \times n_1 + 9 - 4 \times n_1^2} \\
&= \frac{(3 \times n_1 - 3)^2}{n_1^2 \times (5 \times n_1^2 - 18 \times n_1 + 9)}.
\end{aligned} \tag{6}$$

Hence

$$\begin{aligned}
\left\lceil \log_{\frac{3}{2}} \left(\frac{n_{i+1}}{2} \right) \right\rceil &\geq i + \left\lceil \log_{\frac{3}{2}} \left(\frac{n_1}{2} \right) + \frac{1}{\ln \left(\frac{3}{2} \right)} \times \left(\sum_{k=1}^{\infty} \frac{1}{n_k} - \frac{1}{n_1^2} \times \left(\frac{1}{2} + \frac{1}{3 \times (n_1 - 1)} \right) \right) \right\rceil \\
&= i + \left\lceil \log_{\frac{3}{2}} \left(\frac{n_1}{2} \right) + \frac{1}{\ln \left(\frac{3}{2} \right)} \times \left(- \sum_{k=1}^{\infty} \frac{1}{n_k} - \sum_{k=1}^{\infty} \frac{1}{n_k^2} \times \left(\frac{1}{2} + \frac{1}{3 \times (n_1 - 1)} \right) \right) \right\rceil \\
&\geq i + \left\lceil \log_{\frac{3}{2}} \left(\frac{n_1}{2} \right) + \frac{1}{\ln \left(\frac{3}{2} \right)} \right. \\
&\quad \times \left(- \frac{3 \times n_1 - 3}{n_1 \times (n_1 - 3)} - \frac{(3 \times n_1 - 3)^2}{n_1^2 \times (5 \times n_1^2 - 18 \times n_1 + 9)} \times \left(\frac{1}{2} + \frac{1}{3 \times (n_1 - 1)} \right) \right) \left. \right\rceil.
\end{aligned} \tag{7}$$

It yields for $n_1 = 10$,

$$\begin{aligned}
\left\lceil \log_{\frac{3}{2}} \left(\frac{n_{i+1}}{2} \right) \right\rceil &\geq i + \left\lceil \log_{\frac{3}{2}}(5) + \frac{1}{\ln\left(\frac{3}{2}\right)} \right. \\
&\quad \times \left(-\frac{3 \times 10 - 3}{10 \times (10 - 3)} - \frac{(3 \times 10 - 3)^2}{10^2 \times (5 \times 10^2 - 18 \times 10 + 9)} \times \left(\frac{1}{2} + \frac{1}{3 \times (10 - 1)} \right) \right) \left. \right\rceil \\
&\geq i + \left\lceil 3,969 + 2,467 \times \left(-\frac{27}{70} - \frac{27^2}{100 \times (500 - 180 + 9)} \times \left(\frac{1}{2} + \frac{1}{27} \right) \right) \right\rceil \\
&= i + \left\lceil 3,969 + 2,467 \times \left(-\frac{27}{70} - \frac{27^2}{100 \times 329} \times \frac{29}{54} \right) \right\rceil \\
&= i + \left\lceil 3,969 + 2,467 \times \left(-\frac{27}{70} - \frac{27}{32900} \times \frac{29}{2} \right) \right\rceil \\
&\geq i + \lceil 3,969 + 2,467 \times (-0,386 - 0,012) \rceil \\
&= i + \lceil 3,969 - 2,467 \times 0,398 \rceil \\
&\geq i + \lceil 2,987 \rceil, \text{ missed.}
\end{aligned} \tag{8}$$

It yields for $n_1 = 14$,

$$\begin{aligned}
\left\lceil \log_{\frac{3}{2}} \left(\frac{n_{i+1}}{2} \right) \right\rceil &\geq i + \left\lceil \log_{\frac{3}{2}}(7) + \frac{1}{\ln\left(\frac{3}{2}\right)} \right. \\
&\quad \times \left(-\frac{3 \times 14 - 3}{14 \times (14 - 3)} - \frac{(3 \times 14 - 3)^2}{14^2 \times (5 \times 14^2 - 18 \times 14 + 9)} \times \left(\frac{1}{2} + \frac{1}{3 \times (14 - 1)} \right) \right) \left. \right\rceil \\
&\geq i + \left\lceil 4,799 + 2,467 \times \left(-\frac{39}{154} - \frac{39^2}{196 \times (5 \times 196 - 252 + 9)} \times \left(\frac{1}{2} + \frac{1}{39} \right) \right) \right\rceil \\
&= i + \left\lceil 4,799 + 2,467 \times \left(-\frac{39}{154} - \frac{39^2}{196 \times 737} \times \frac{41}{78} \right) \right\rceil \\
&= i + \left\lceil 4,799 + 2,467 \times \left(-\frac{39}{154} - \frac{39}{144452} \times \frac{41}{2} \right) \right\rceil \\
&\geq i + \lceil 4,799 + 2,467 \times (-0,254 - 0,006) \rceil \\
&= i + \lceil 4,799 - 2,467 \times 0,260 \rceil \\
&\geq i + \lceil 4,157 \rceil \\
&\geq i + \left\lceil \log_{\frac{3}{2}} \left(\frac{n_1}{2} \right) \right\rceil, \text{ won.}
\end{aligned} \tag{9}$$

Since clearly

$$\left\lceil \log_{\frac{3}{2}} \left(\frac{n_{i+1}}{2} \right) \right\rceil \leq i + \left\lceil \log_{\frac{3}{2}} \left(\frac{n_1}{2} \right) \right\rceil, \tag{10}$$

we have

$$\left\lceil \log_{\frac{3}{2}} \left(\frac{n_{i+1}}{2} \right) \right\rceil = i + \left\lceil \log_{\frac{3}{2}} \left(\frac{n_1}{2} \right) \right\rceil, \tag{11}$$

thus, it is won also to show that the optimal bound is attained an infinite number of times.

Theorem 7.2. *For Example 7.1 of universal decomposition, we have $\alpha \leq 2 + \lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil$, $\sigma \leq 1,241846949 \times n^{1,709511291}$, $\Sigma \leq 1,241846949 \times n^{2,709511291}$. (We have also $\alpha \geq 1 + \lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil$, etc.)*

We found no previous work for the optimal bound. Since this cut into two thirds is very frequent in combinatorics, we would be happy if a reader knows an older publication that gives this result, and takes the time to send us an email to let us know. (Note that it is possible to make a mistake when reading articles where the cut stops once the parts have size 2, because then the depth is between $\lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil$ and $1 + \lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil$, and if the article or the book shows the bound $2 + \lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil$, it corresponds in real to our second bound $3 + \lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil$ and not to the optimal bound.)

Let a be the number of distinct adjacency types in a binary structure ($a = 2$ for the simple undirected graphs). We denote $b = \min(n - 1, a)$ and $c = \min(\frac{n \times (n-1)}{2}, a \times (n - 1)) = \min(\frac{n}{2}, a) \times (n - 1)$.

Parameters	Types	Origin
$(n; 1; 1, 1, 1; n; 1, n)$	All	Trivial
$(n; 1; \lceil \lg(n) \rceil, 1, \lceil \lg(n) \rceil; 2; 1, n)$	All	Trivial
$(2; \leq b; n - 1, \leq c, \leq c; 2; 1, n)$	All	Lemma 7.6 Lyaudet (2019)
$(2; 1; 2, 1, 2; \frac{n \times (n-1)}{2}; n - 1, n \times (n - 1))$	1-tight and ϵ	Example 1.1
$(2; 1; \lceil \lg(n) + \lg(n - 1) \rceil, 1, \lceil \lg(n) + \lg(n - 1) \rceil; 2; n - 1, n \times (n - 1))$	ϵ	Corollary 4.5
$(2; 1; n - 1, n - 1, n - 1; 2; 2^{n-3}, 2^{n-1})$	1-short and ϵ	Example 6.9
$(2; 1; n - 1, 1, n - 1; 3; 2^{n-3}, 2^n - 2)$	1-tight and ϵ	Example 6.12
$(2; 1; 3 + \lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil, 1, 3 + \lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil; 3; 1.25 \times n^{1.71}, 1.25 \times n^{2.71})$	1-tight et ϵ	Example 7.1

Here is a zoom on the parameters:

K	Λ	α	β	γ	Δ	σ	Σ
n	1	1	1	1	n	1	n
n	1	$\lceil \lg(n) \rceil$	1	$\lceil \lg(n) \rceil$	2	1	n
2	$\leq \min(n - 1, a)$	$n - 1$	$\leq \min(\frac{n}{2}, a) \times (n - 1)$	$\leq \min(\frac{n}{2}, a) \times (n - 1)$	2	1	n
2	1	2	1	2	$\frac{n \times (n-1)}{2}$	$n - 1$	$n \times (n - 1)$
2	1	$\lceil \lg(n) + \lg(n - 1) \rceil$	1	$\lceil \lg(n) + \lg(n - 1) \rceil$	2	$n - 1$	$n \times (n - 1)$
2	1	$n - 1$	$n - 1$	$n - 1$	2	2^{n-3}	2^{n-1}
2	1	$n - 1$	1	$n - 1$	3	2^{n-3}	$2^n - 2$
2	1	$3 + \lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil$	1	$3 + \lceil \log_{\frac{3}{2}}(\frac{n}{2}) \rceil$	3	$1.25 \times n^{1.71}$	$1.25 \times n^{2.71}$

8 Conclusion

After this study, only the following tree-questionable-width is still truly open in the non-bijective case: the 1-tight balanced tree-questionable-width with degree 2, which is equivalent to the 1-short balanced tree-questionable-width with degree 2. The “ ϵ ” types are too powerful and decomposes everything, no matter the additional constraints. There also remains the more classical open problem of the bijective tree-questionable-width. Our counter-examples contain a lot of redundant and slightly stupid structures; it is possible that canonisation processes of the decompositions yield equality of certain types after canonisation. The pruning results are a first step in this direction, but

it will probably need more complex transformation results. A lot of things are configurable with the tree-questionable-width, particularly in the non-bijective case. We could complexify the problem further: by looking at various bounds on the degree (logarithmic bounds for example); or by looking at bounds on the “surjectivity”, like asking that a vertex of a finite binary structure is the image of at most a logarithmic number of leaves (that σ is logarithmic). Incidentally, among the 8 parameters of tree-questionable-decompositions that we explicitated, only σ and Σ are not bounded above by a polynomial in n ; finding a better bound on these parameters thanks to transformations that only increase slightly the other parameters is one of our priorities. We could even allow invalid questions, and only ask for a majority of valid questions (going from 8 parameters to 20 parameters: 3 parameters for the proportion of tight ascending paths with valid question, the proportion of tight ascending paths without question, the proportion of tight ascending paths with invalid question repeat for short, medium, and wide); in that case or for the variant types with proportion, we could study probabilistic sampling, etc.

Another idea that we probably had between 2019 and 2022 is to apply the principle of first differences no more on paths, no more on trees, but on well-founded partial orders. Indeed, all the ingredients are here with the notions of junction points and ascending paths; we would have a new possibility to have many ascending paths for the same junction point. It would yield por-questionable-decomposition (por for partial order).

We could also extend the “2-ary” principle of first differences to a principle of first differences of larger arity, as we asked in Lyaudet (2020). For example, for 3 vertices, it suffices to consider that the “adjacency”/the hyperedge between these 3 vertices is fixed by the first mapping that maps the 3 vertices to 3 distinct vertices. We can do similarly for some adjacency types/some hyperedges of larger arity. If we take the notation of 2-structures in the book Ehrenfeucht et al. (1999), we can generalise into (a) -structures and $(\leq a)$ -structures. It would be interesting to know if the parameters of the questionable-decompositions of the $(\leq a)$ -structures can be bounded by a function of the parameters of the questionable-decompositions of the (b) -structures obtained by projection on the relations of arity b for $b \leq a$. Already with the ternary relations we can modelise Peano’s arithmetic of strictly positive integers; we would have the 6 following adjacency types: no hyperedge, an hyperedge with 3 slots $s_1 < s_2 < s_3$ and $s_1 + s_2 = s_3$, an hyperedge with 3 slots $s_1 = s_2 < s_3$ and $s_1 + s_2 = s_3$, an hyperedge with 3 slots $s_1 < s_2 < s_3$ and $s_1 \times s_2 = s_3$, an hyperedge with 3 slots $s_1 = s_2 < s_3$ and $s_1 \times s_2 = s_3$, the combination of two hyperedges with 3 slots $s_1 = s_2 < s_3$ and $s_1 + s_2 = s_3$ with $s_1 = s_2 < s_3$ and $s_1 \times s_2 = s_3$, since $2 + 2 = 2 \times 2 = 4$;). For the principle of first differences on trees, hence the tree-questionable-decompositions, with larger arities, we would have to rework the study of junction points, because only the a -ary tight junction points have an obvious definition. The tight junction points are the leaves of the common tree of the a vertices. The short junction points are the nodes of the common tree of the a vertices that for each of the a vertices have at least one son that contains only this vertex among the a vertices. The medium junction points are the nodes of the common tree of the a vertices that for at least one of the a vertices have at least one son that contains only this vertex among the a vertices. The large junction points are the nodes of the common tree of the a vertices that have at least two sons in

the union tree of the trees of the a vertices. But unlike the 2-ary case, where the only variants left of junction points had criteria of numbers of sons/of branches that seemed artificial and not very relevant, the a -ary case could yield other types of junction points based on a majority of the a vertices or other criteria that may have an interest.

Finally, to find the motivation and energy to continue to work on these subjects, we do not drink coffee, but we build castles in the sky. Here is an example of such a castle. Nowadays, the consensus is that there is probably no complete problem for the class $\text{NP} \cap \text{co-NP}$. But there may be a way to “cheat”, for example by taking a PSPACE-complete problem and keeping only the instances that have a positive certificate and a negative certificate. Maybe such a problem exists on 2-structures, or with a larger arity. And since there is an exponential number of universal decompositions such as Example 7.1, maybe also that some of them can be used as a positive certificate and others can be used as a negative certificate (We always have such decompositions, but only some give polynomial algorithms.). It would need two reductions: one that preserves the positive certificates this way, and the second that preserves the negative certificates also this way. The difficulty would be to have a synchronisation of the two reductions so that they both give the same instance of the PSPACE-complete problem on 2-structures. And we could go closer to a result of convergence of the certificates. Indeed, if $P = (\text{NP} \cap \text{co-NP})$, the input can be used at the same time as a positive and negative certificate. We could cheat by saying this is always true for a problem in $\text{NP} \cap \text{co-NP}$, by merging the two certificates so that the bits of even rank code the positive certificate and the bits of odd rank code the negative certificate; the Turing machine that proves uses only the bits of even rank of the mixed certificate; the Turing machine that disproves uses only the bits of odd rank of the mixed certificate; the size of the certificate remains polynomial. But doing this with the parity of the rank of the bits teaches us nothing, whilst a synchronisation of the universal decompositions to show that a single universal decomposition can be used at the same time as a positive and negative certificate would be probably more interesting. We could imagine a kind of exchange lemma between the universal decompositions to “synchronise them”/bring them closer.

Thanks God! Thanks Father! Thanks Jesus! Thanks Holy-Spirit! I thank also Copilot that helped me not lose too much time reinventing the wheel on some computations :).

References

- A. Ehrenfeucht, T. Harju, and G. Rozenberg. *The Theory of 2-Structures - A Framework for Decomposition and Transformation of Graphs*. World Scientific Publishing Co. Pte. Ltd., 1999. ISBN 981-02-4042-2.
- L. Lyaudet. On finite width questionable representations of orders. *CoRR*, abs/1903.02028, 2019. URL <https://arxiv.org/abs/1903.02028>.
- L. Lyaudet. First difference principle applied to modular/questionable-width, clique-width, and rank-width of binary structures. *preprint*, abs/202009, 2020. URL <https://lyaudet.eu/laurent/>

Publi/Journaux/LL2020LargeursStructuresBinaires/
LL2020LargeursStructuresBinaires_v5.pdf.

L. Lyaudet. Diviser n'est pas régner ? *preprint*, abs/202205, 2022.
URL [https://lyaudet.eu/laurent/Publi/Journaux/
LL2022DiviserNestPasRegner/LL2022DiviserNestPasRegner_
v6.pdf](https://lyaudet.eu/laurent/Publi/Journaux/LL2022DiviserNestPasRegner/LL2022DiviserNestPasRegner_v6.pdf).

L. Lyaudet. On tree-width and tree-questionable-width. *preprint*,
abs/202506-2, 2025a. URL [https://lyaudet.eu/laurent/
Publi/Journaux/LL2025TreewidthTreequestionablewidth/
LL2025TreewidthTreequestionablewidth_en_v2.pdf](https://lyaudet.eu/laurent/Publi/Journaux/LL2025TreewidthTreequestionablewidth/LL2025TreewidthTreequestionablewidth_en_v2.pdf).

L. Lyaudet. Sur la largeur arborescente et la largeur arborescente questionnable.
preprint, abs/202506-1, 2025b. URL [https://lyaudet.eu/laurent/
Publi/Journaux/LL2025TreewidthTreequestionablewidth/
LL2025TreewidthTreequestionablewidth_fr_v2.pdf](https://lyaudet.eu/laurent/Publi/Journaux/LL2025TreewidthTreequestionablewidth/LL2025TreewidthTreequestionablewidth_fr_v2.pdf).

W. Sierpiński. Généralisation d'un théorème de Cantor concernant les ensembles or-
donnés dénombrables. *Fundamenta Mathematicae*, 18:280–284, 1932.

Archive

FR V1 2025/06/12:

[https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/
LL2025LAQNonBijective_fr_v1.pdf](https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/LL2025LAQNonBijective_fr_v1.pdf)

EN V1 2025/06/12:

[https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/
LL2025LAQNonBijective_en_v1.pdf](https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/LL2025LAQNonBijective_en_v1.pdf)

FR V2 2025/06/15:

[https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/
LL2025LAQNonBijective_fr_v2.pdf](https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/LL2025LAQNonBijective_fr_v2.pdf)

EN V2 2025/06/15:

[https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/
LL2025LAQNonBijective_en_v2.pdf](https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/LL2025LAQNonBijective_en_v2.pdf)

FR V3 2025/06/30:

[https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/
LL2025LAQNonBijective_fr_v3.pdf](https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/LL2025LAQNonBijective_fr_v3.pdf)

EN V3 2025/06/30:

[https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/
LL2025LAQNonBijective_en_v3.pdf](https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/LL2025LAQNonBijective_en_v3.pdf)

FR V4 2025/11/24:

[https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/
LL2025LAQNonBijective_fr_v4.pdf](https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/LL2025LAQNonBijective_fr_v4.pdf)

EN V4 2025/11/24 this version:

[https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/
LL2025LAQNonBijective_en_v4.pdf](https://lyaudet.eu/laurent/Publi/Journaux/LL2025LAQNonBijective/LL2025LAQNonBijective_en_v4.pdf)