Decreasing verification radius in local certification ⋆

Laurent Feuilloley, Jan Janoušek, Jan Matyáš Křišťan, Josef Erik Sedláček

Decreasing verification radius in local certification *

Laurent Feuilloley[2][0000−0002−3994−0898], Jan Janoušek[1][0000−0001−9329−1000], Jan Matyáš Křišťan[1,3][0000−0001−6657−0020], and Josef Erik Sedláček[1,4][0009−0001−7429−2937]

[1] Faculty of Information Technology, CTU in Prague, Czech Republic
[2] CNRS, INSA Lyon, UCBL, LIRIS, UMR5205, F-69622 Villeurbanne, France
[3] kristja6@fit.cvut.cz
[4] sedlajo5@fit.cvut.cz

Abstract. This paper deals with *local certification*, specifically locally checkable proofs: given a *graph property*, the task is to certify whether a graph satisfies the property. The verification of this certification needs to be done *locally* without the knowledge of the whole graph. More precisely, a distributed algorithm, called a *verifier*, is executed on each vertex. The verifier observes the local neighborhood up to a constant distance and either accepts or rejects.
We examine the trade-off between the visibility radius and the size of certificates. We describe a procedure that decreases the radius by encoding the neighbourhood of each vertex into its certificate. We also provide a corresponding lower bound on the required certificate size increase, showing that such an approach is close to optimal.

Keywords: Local certification, locally checkable proofs, proof-labeling schemes, graphs, distributed computing, self-stabilization

1 Introduction

The problem studied in this paper involves certifying a global graph property without having knowledge of the entire graph. In particular, we study the model of locally checkable proofs of Göös and Suomela [14].

In this model, a distributed algorithm called a *verifier* examines the local neighbourhood of each vertex up to some fixed distance, called the *radius*. On each vertex, the verifier either accepts if it cannot deny that the graph has the desired property, or rejects if it is certain that the property is not satisfied. The final decision about the property is then made as follows: If the verifier rejected on at least one vertex, the decision is that the property is not satisfied. If it accepts on all vertices, the decision is that the property holds.

To enhance the decision-making capabilities of the model, the vertices are equipped with unique identifiers and possibly more general labels. Furthermore, each vertex is given a *certificate*.

* The paper is eligible for the best student paper award.

Certificates, are bit-strings that are used to help the verifier in deciding the answer about the property. The verifier reads the certificates in its local view as a part of its input. For each graph that satisfies the property, the verifier must accept for at least one assignment of certificates. If the graph does not satisfy the property, the verifier must reject every assignment of certificates.

The key notion of local certification is that of a *proof labeling scheme*, which is a pair $(f, \mathcal{A})$, where $\mathcal{A}$ is the verifier and f, often called a *prover*, gives each graph with the property an assignment of certificates that is accepted by $\mathcal{A}$.

An intuitive example of a problem requiring certificates is k-colorability, where k is a constant. Clearly, $\mathcal{O}(\log k)$ bits are enough, as the coloring can be provided in the certificates. If the graph is not k-colorable, then at least one vertex will be adjacent to a vertex with the same color (or use a color greater than k) and it would reject.

1.1 Previous work

Similar models, related to the LOCAL model [18], have been studied under different names [16,14]. The name *local certification* is a general term used for the similar models [7].

The concept of local certification is relevant to self-stabilization [6,1], as it is a component of many self-stabilizing algorithms [16].

Since local certification has been introduced, lower bounds and upper bounds for the size of the certificates for many graph properties and problems have been proven [5,2,9,15]. Also, the strength of the model in general and under various restrictions was studied [13,12,11]. For a general survey see [7].

It has been previously shown how, and under which conditions, certificate size can be decreased at the cost of increasing the visibility radius [17,8,10]. We provide a similar result, showing how the visibility radius can be decreased at the cost of increasing the certificate size.

There is a subtle but crucial distinction between these two problems. While the previous results allow increasing the radius while decreasing the size of certificates in the general case, the implied inverse procedure of decreasing the radius and increasing the certificate size works only for the very particular type of proof labeling schemes that result from the original procedure. The novelty of our results lies in allowing the decrease of the radius of *any* proof labeling schemes.

1.2 Our contribution

We show a general procedure for decreasing the radius of a proof labeling scheme at the expense of increasing the size of its certificates. We also provide a corresponding lower bound on the necessary certificate size increase.

Consider the following motivation for this result. Given a network, it may be needed to check that it satisfies a given property, such as absence of a cycle. Communication between nodes of the network may be bounded by distance,

which corresponds to the visibility radius of the verifier. Under these conditions, proof labeling schemes provide a template for verifying properties of this network. Our procedure then allows to decrease the communication distance at the cost of increasing the memory and computational requirements of each node.

In Section 3 we formally describe the procedure. Given a proof labeling scheme $(f, \mathcal{A})$ with radius r certifying some property $\mathcal{P}$, we show how to construct a proof labeling scheme with radius $(r-\delta)$ certifying $\mathcal{P}$ with size bounded by $\mathcal{O}((\Delta - 1)^{\delta}(\Delta \log(n) + s(n) + \ell(n)))$, where Δ is the maximum degree of the certified graph, s is the certificate size of $(f, \mathcal{A})$, and ℓ is the size of labels on the vertices of the input graph. In Section 4, we show that the increase by a factor of $C \cdot \Delta^{\delta-1}$ is necessary for some graph properties, while in Section 5 we prove that in some cases the multiplicative logarithmic terms can be replaced by an additive one.

2 Preliminaries

All the graphs are assumed to be undirected and simple with possible labels. We also assume that all graphs are connected, as two different connected components have no way to interact with each other. Formally $G = (V, E, L)$ where $L \colon V \to \{0,1\}^*$. The vertices are assumed to have assigned integer *identifiers*, formally for a graph on n vertices, we assume that $V = \{1, \ldots, n\}$.

Neighbours of a vertex v are denoted as $N_G(v)$, if G is clear from the context, $N(v)$ is used. Distance between two vertices u, v is denoted as $d_G(u, v)$, and the subscript is omitted if G is clear from the context. We denote the set of vertices within distance r from v as $V[v, r]$, also called the r-local neighbourhood of v.

A *graph property* is formally a set of graphs that is closed under isomorphism, that is, its membership does not depend on the choice of identifiers. Note that it may depend on the labels of the graph. A *certificate assignment* P for G is a function $P \colon V(G) \to \{0,1\}^*$ that associates with each vertex a *certificate*. We say that P has size s if $|P(v)| \leq s(n)$ for every v. A *verifier* is a function that takes as an input a graph G, its certificate assignment P and $v \in V(G)$ and outputs either 0 or 1. In this case, we say that the verifier is invoked on v.

We denote the induced subgraph $G[V[v, r]]$ as $G[v, r]$ and the restriction of P to $V[v, r]$ is denoted as $P[v, r]$, that is $P[v, r] \colon V[v, r] \to \{0,1\}^*$. A verifier $\mathcal{A}$ is *r-local* if $\mathcal{A}(G, P, v) = \mathcal{A}(G[v, r], P[v, r], v)$ for all G, P, and v.

An r-local *proof labeling scheme* certifying a property of labeled graphs $\mathcal{P}$ is a pair $(f, \mathcal{A})$, where $\mathcal{A}$ is an r-local verifier and f, called the *prover*, assigns to each $G \in \mathcal{P}$ a certificate assignment P such that the following properties hold.

- *Completeness*: If $G \in \mathcal{P}$, then $\mathcal{A}(G[v, r], P[v, r], v) = 1$ for all v, where $P = f(G)$.
- *Soundness*: If $G \notin \mathcal{P}$, then for every certificate assignment P', there is v such that $\mathcal{A}(G[v, r], P'[v, r], v) = 0$.

We say that $(f, \mathcal{A})$ has size $s : \mathbb{N} \to \mathbb{N}$ if $|f(G)(v)| \leq s(|V(G)|)$ for all $G \in \mathcal{P}$ and all $v \in V(G)$.

3 Decreasing the radius of a proof labeling scheme

The goal of this section is to show that given an r-local proof labeling scheme $(f_r, \mathcal{A}_r)$ certifying property $\mathcal{P}$, we can construct an $(r-\delta)$-local proof labeling scheme $(f, \mathcal{A})$ certifying $\mathcal{P}$ for any $\delta < r$ at the cost of increasing the certificate size by a certain amount. The increase of the certificate size can be expressed as a function of the size of the input graph and its maximum degree. The result is precisely formulated as follows.

Theorem 1. *Given an r-local proof labeling scheme $(f_r, \mathcal{A}_r)$ of size s certifying a graph property $\mathcal{P}$, for every $\delta < r$, we can construct an $(r-\delta)$-local proof labeling scheme certifying $\mathcal{P}$ with certificates of size*

$$\mathcal{O}((\Delta - 1)^{\delta}(\Delta \log(n) + s(n) + \ell(n)))$$

where $\ell(n)$ is the maximum size of a label and $\Delta \geq 3$ is the maximum degree of the input graph.

Note that in the case of $\Delta = 2$, the maximum size of a δ-neighborhood of a vertex grows only linearly with δ and we may obtain the bound on certificate size of $\mathcal{O}(\delta(\Delta \log(n) + s(n) + \ell(n)))$.

3.1 Overview of the proof technique

When the verifier $\mathcal{A}_r$ is invoked on v, it is given $G[v, r]$ and $P[v, r]$ on its input. If we want to reduce that information to $G[v, r-\delta], P[v, r-\delta]$, a first step can be to *move* the now missing information into the certificates. The first obstacle comes from the fact that information in the certificates may not be true (as opposed to $G[v, r]$ provided on the input) and must be verified.

3.2 Encoding neighborhood into certificates

The essential idea is simple, we have each vertex hold its distance δ neighborhood in its certificate. This allows other vertices within distance $r - \delta$ to gain information about the entire distance r neighborhood and feed this information to the original r-local verifier.

Instead of having each vertex explicitly encode its δ neighborbood, we define the notion of *packets*, which are then *broadcast* into the extended neighborhood. The effect is the same as encoding the δ neighborhood in the certificate, and the notion allows us to break down the verification into simple checks and the proof of correctness into simple observations.

Definition 1. *We say that (D, L, C, ω, d) is a* packet*, if*

- *D is a set of vertices,*
- *$L, C \in \{0,1\}^*$,*
- *ω is the identifier of so-called* origin vertex*,*

– *and* $d \in \{0, \ldots, \delta\}$.

Note that a packet can be easily encoded and decoded into a binary string. Furthermore, it is easy to locally check that the individual elements of the packet correspond to the definition and have the correct types.

Intuitively, D, L, and C are used to carry the local information around ω. In particular, D will be the identifiers of $N(v)$, L will be the label on v, and C is a certificate on v eventually passed on to $\mathcal{A}_r$. We use d to keep the distance of the packet from ω to ensure the correct distribution of the packet among other vertices.

The origin of a packet p is denoted as $\omega(p)$, a similar notation is used with the other components of the packet. Given a certificate assignment $P\colon V(G) \to \{0,1\}^*$ and $v \in V$, $\texttt{pts}(P, v)$ denotes the set of packets encoded in $P(v)$ (if $P(v)$ is not a valid encoding of packets then $\texttt{pts}(P, v) = \emptyset$). We define $\texttt{has-pt}(P, v, x)$ as true if and only if there is $p \in \texttt{pts}(P, v)$ such that $\omega(p) = x$ and if true, we denote such packet p as $\texttt{pt}(P, v, x)$.

Definition 2. *We say that packet* $p = (D, L, C, \omega, d)$ *is* well-formed *if and only if* $D = N_G(\omega)$ *and* L *is the label on* ω.

Now, we show how well-formed packets can be used to reconstruct the neighborhood. Let $\mathcal{B}$ be a set of packets and $\omega(\mathcal{B}) \coloneqq \{\omega(p) \mid p \in \mathcal{B}\}$. Let $G(\mathcal{B}) \coloneqq (\omega(\mathcal{B}), E(\mathcal{B}), L(\mathcal{B}))$, where $E(\mathcal{B}) \coloneqq \{\{x, y\} \mid x, y \in \omega(\mathcal{B})$ and $\exists p \in \mathcal{B}$ such that $\omega(p) = x$ and $y \in D(p)\}$ and $L(\mathcal{B})(v) = L(p)$ such that $\omega(p) = v$.

Observation 1. Let $\mathcal{B}$ be a set of well-formed packets such that $\omega(\mathcal{B}) = V[v, r]$. Then $G(\mathcal{B}) = G[v, r]$.

Proof. For each $\{x, y\} \in E(G[v, r])$, there is $p \in \mathcal{B}$ with $\omega(p) = x$ and $y \in D(p) = N(x)$ as p is well-formed. On the other hand, for each p with $\omega(p) = x$ and $y \in D(p)$, there is $\{x, y\} \in E(G[v, r])$ if also $y \in V[v, r]$. □

Similarly, we can reconstruct the encoded certificate assignment, we define $\mathcal{C}(\mathcal{B}) : \omega(\mathcal{B}) \to \{0,1\}^*$ so that $C(\mathcal{B})(v) = C(p)$ where $p \in \mathcal{B}$ such that $\omega(p) = v$.

3.3 Constructing the $(r - \delta)$-local verifier

Recall that we are given an r-local proof labeling scheme $(f_r, \mathcal{A}_r)$ certifying the property of labeled graphs $\mathcal{P}$. We now define the $(r - \delta)$-local verifier. The main task of the verifier is to ensure that all packets are distributed as needed and all are well-formed, that is they can be used to reconstruct wider neighbourhoods.

Let $\mathcal{B}[P, u, r - \delta] := \bigcup_{v \in V[u, r-\delta]} \texttt{pts}(P, v)$, that is $\mathcal{B}[P, u, r - \delta]$ is the set of all packets that are visible from u to distance $r - \delta$ and are encoded in P. We define $\mathcal{A}$ so that $\mathcal{A}(G[v, r - \delta], P[v, r - \delta], v) = 1$ if and only if the following conditions are all satisfied:

Condition B1: For each x, there is at most one $p \in \texttt{pts}(P, v)$ such that $\omega(p) = x$,

Condition B2: $(N(v), L(v), C', v, 0) \in \texttt{pts}(P, v)$ for some $C' \in \{0,1\}^*$,
Condition B3: if $\texttt{has-pt}(P, v, u)$ and $v \neq u$ then $1 \leq d(\texttt{pt}(P, v, u)) = 1 + \min\{d(\texttt{pt}(P, x, u)) \mid \texttt{has-pt}(P, x, u) \text{ and } x \in N(v)\}$,
Condition B4: $\texttt{has-pt}(P, v, u)$ for $v \neq u$ if and only if there is $x \in N(v)$ and $p' = \texttt{pt}(P, x, u)$ such that $d(p') < \delta$,
Condition B5: for every $p = \texttt{pt}(P, v, u) \in \texttt{pts}(P, v)$ and every existing $p' = \texttt{pt}(P, x, u)$ such that $x \in N(v)$, it holds $D(p) = D(p'), L(p) = L(p'), C(p) = C(p')$,
Condition B6: let $G' = G(\mathcal{B}[P, v, r - \delta])$ and $P' = C(\mathcal{B}[P, v, r - \delta])$, then $\mathcal{A}_r(G'[v, r], P'[v, r], v) = 1$.

Note that each condition requires information only about neighbours within distance at most $r - \delta$ and hence can be locally verified. Condition B1 makes the reasoning easier, as we can assume that at most one packet from an originating vertex is present in the certificate. Condition B2 ensures that every vertex has a well-formed packet originating from the vertex itself. Condition B3 allows us to inductively prove that each packet correctly holds its distance from its originating vertex. Condition B4 ensures that a packet originating from a vertex is distributed to vertices within a distance δ from it. Condition B5 ensures that the packet is correctly *copied* from one vertex to another. Condition B6 ensures that the original verifier accepts the described graph and certificate assignment.

Now, we can establish properties of the encoded packets, given that the verifier accepts them. To make the notation shorter, we denote $\mathcal{A}(G[v, r-\delta], P[v, r-\delta], v)$ as $\mathcal{A}(v)$. First, we establish that the encoded distances are equal to the actual distances in the graph.

The following lemmas follow from the definitions of the six Conditions. The reasoning is straightforward, so we have decided to move the proofs into an appendix. They are provided in Appendix A.

Lemma 1. *If $\mathcal{A}(v) = 1$ for all v, then $d(\texttt{pt}(P, u, v)) = d_G(u, v)$ for all $u, v \in V$ such that $\texttt{has-pt}(P, u, v)$.*

Next, we establish that the local information of each vertex is indeed distributed into its δ neighborhood, given that the verifier accepts the certificates.

Lemma 2. *If $\mathcal{A}(v) = 1$ for all v, then for every $u, x \in V$ it holds $\texttt{has-pt}(P, u, x)$ if and only if $u \in V[x, \delta]$.*

We now show that all packets with the same origin must hold the same local information.

Lemma 3. *If $\mathcal{A}(v) = 1$ for all v, then $C(\texttt{pt}(P, v, x)) = C(\texttt{pt}(P, u, x))$ for all v, u, x such that $\texttt{has-pt}(P, v, x)$ and $\texttt{has-pt}(P, u, x)$.*

Next, we show that the information carried by the packets must be well-formed.

Lemma 4. *If $\mathcal{A}(v) = 1$ for all v, then $\texttt{pt}(P, v, x)$ is well-formed for every v, x such that $\texttt{has-pt}(P, v, x)$.*

It remains to establish that each vertex has access to enough packets to compute its distance r neighborhood.

Lemma 5. *If $\mathcal{A}(v) = 1$ for all v, then for every $u \in V$ it holds $V[u, r] \subseteq \omega(\mathcal{B}[P, u, r - \delta])$.*

Next, we establish that if the original r-local verifier $\mathcal{A}_r$ accepts based on both the local information and the information in packets, the graph indeed satisfies the property $\mathcal{P}$.

Lemma 6. *If $\mathcal{A}(v) = 1$ for all v, then $G \in \mathcal{P}$.*

Finally, we are ready to prove the main theorem.

Proof (Theorem 1). It follows from Lemma 6 that $\mathcal{A}$ accepts G only if $G \in \mathcal{P}$. It remains to show that any $G \in \mathcal{P}$ has a certificate assignment accepted by $\mathcal{A}$, with each certificate of size at most $\mathcal{O}((\Delta - 1)^\delta(\Delta \log(n) + s(n) + \ell(n)))$.

Given $G \in \mathcal{P}$, we construct P so that for each u, $P(u)$ is an encoding of packets with one packet for each vertex $x \in V[u, \delta]$. A packet $p_x = (D, L, C, \omega, d)$ for vertex u is constructed by setting $D \coloneqq N_G(x)$, $L \coloneqq L_G(x)$, $C \coloneqq f_r(G)(x)$, $\omega \coloneqq x$, and $d \coloneqq d_G(x, u)$. It remains to check that the verifier returns 1 on each vertex.

Note that Conditions B1, B2, B3, B4, and B5 hold from the construction. Finally, note that by Observation 1, we have

$$\mathcal{A}_r(G(\mathcal{B}[P, v, r - \delta])[v, r], C(\mathcal{B}[P, v, r - \delta])[v, r], v) = \mathcal{A}_r(G[v, r], f_r(G)[v, r]) = 1$$

for each v and hence Condition B6 holds. Thus, the certificate assignment P is accepted by $\mathcal{A}$.

Now, we proceed to bound the maximum size of a certificate assigned by P. Assuming we encode a set of vertices as a set of their identifiers, each of size $\mathcal{O}(\log(n))$, the size of encoding of (D, L, C, ω, d) is bounded by $\mathcal{O}(\Delta(n) \log(n) + \ell(n) + s(n))$. We have $|\texttt{pts}(P, v)| = |V[v, \delta]| \leq (\Delta(\Delta - 1)^\delta - 2)/(\Delta - 2)$ for $\Delta \geq 3$ and for all v, thus the size of encoding of $P(v)$ is bounded by $\mathcal{O}((\Delta - 1)^\delta(\Delta \log(n) + s(n) + \ell(n)))$.

In the case of $\Delta = 2$, we have $|V[v, \delta]| = 2\delta + 1$ and then the size of encoding of $P(v)$ if bounded by $\mathcal{O}(\delta(\Delta \log(n) + s(n) + \ell(n)))$. □

4 Lower bound on the increase of certificate size

This section aims to show that there are proof labeling schemes for which the radius can be decreased by δ only if we also increase the certificate size by $C(\Delta - 1)^{\delta - 1}$, where C is a fixed constant. We present a property of labeled graphs, for which we also provide a proof labeling scheme and both an upper

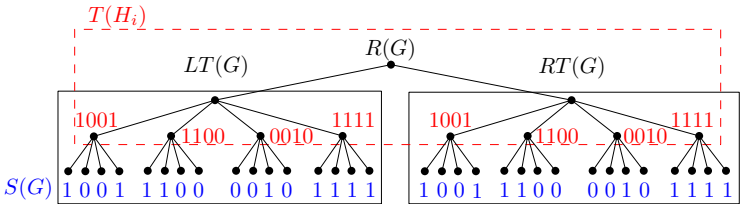

Fig. 1. An example of a graph with property $\mathcal{P}_\Delta$ with $\Delta = 5$. Here, $R(G)$ is the root, $LT(G)$ and $RT(G)$ are the left and the right subtrees, $S(G)$ is the binary sequence in the leaves, and the red strings are certificates. The subgraph $T(H_i)$ is used in the proof of Lemma 8 and corresponds to $r = 2$.

and a lower bound on its size. Later, we describe a property of graphs without labels utilizing very similar idea.

Let $\Delta \geq 3$, then we define $\mathcal{P}_\Delta$ so that a labeled $G \in \mathcal{P}_\Delta$ if and only if it satisfies all of the following three properties. For an example of a graph with the property, see Figure 1.

Property 1 (Tree structure). G has a single vertex of degree 2, denoted as $R(G)$ (or just R, when a concrete G is irrelevant or clear from the context), which is adjacent to two complete $(\Delta - 1)$-nary trees of the same size.

Property 2 (Label structure). For every vertex v except for the root $R(G)$, the label $L(v)$ encodes an integer $a \in \{1, 2, \ldots, \Delta - 1\}$ that uniquely defines its order among its siblings. Additionally, if $deg(v) = 1$, then $L(v)$ also encodes one bit $b \in \{0, 1\}$. Therefore, on leaves $L(v)$ encodes a pair (a, b). For $L(R(G))$, the label is empty.

This allows us to naturally define the *left* and *right* subtrees of G, i.e. the subtrees rooted on the first and second child of $R(G)$ respectively. We denote those as $LT(G)$ and $RT(G)$. Furthermore, it allows us to order the leaves of G. We denote as $S(v)$ the binary string created by taking the values of b on all leaves in their natural order in the subtree rooted at v. We define $S(G) = S(R(G))$.

Property 3 (String structure). $S(G) = XX$ for some binary string X, i.e. $S(G)$ is a result of concatenating a string X with itself once.

Now we describe a proof labeling scheme of $\mathcal{P}_\Delta$. The following lemma provides an upper bound on the optimal certificate size for a given radius. This is then used together with a corresponding lower bound, to show a lower bound on the necessary increase of the certificate size of $\mathcal{P}_\Delta$ when decreasing the radius.

Lemma 7. *Graph property $\mathcal{P}_\Delta$ has an r-local proof labeling scheme of size $C \cdot n/(\Delta - 1)^{r-1}$ for every $r \geq 1$ and a fixed C.*

Proof. We will show how to certify the Properties 1, 2, and 3. To certify Property 1, each vertex can be given in its certificates the identifier of the root (the vertex

of degree 2), its distance from the root, and the depth of the tree. It is well known that distances are enough to certify the acyclicity of a graph and described for instance in [7]. To verify the depth of the whole tree, it suffices to ensure that each certificate has the same value of this depth and on leaves, this depth is verified to be equal to the distance from the root.

To certify Property 2, it suffices to check the structure of each label. Checking the uniqueness of a among siblings can be verified by each parent.

To certify Property 3, each vertex v holds $S(v)$ in its certificate. This is again trivial to verify on a leaf and for any other v, assuming that all children c_i hold correct $S(c_i)$, it suffices to check that $S(v)$ is their concatenation.

Now the key idea of reducing the size of certificates is to encode the value of $S(v)$ only in the vertices with $d(v, R) \geq r$. Certificates of the vertices with $d(v, R) < r$ are left empty. Note that the number of leaves of the subtree of v with $d(v, R) = k$ is at most $n/(2(\Delta - 1)^{k-1})$ as there are $(2(\Delta - 1)^{k-1})$ vertices with distance k from R. Hence, we can encode $S(v)$ using at most $\mathcal{O}(n)/(\Delta - 1)^{k-1}$ bits. Now, R may compute $S(G)$ as the concatenation of $S(v)$ for all v with $d(v, R) = r$ and check that it satisfies Property 3.

The rest of the properties require only $\mathcal{O}(\log(n))$ bits in certificates, thus there is an r-local proof labeling scheme that certifies $\mathcal{P}_\Delta$ with certificates of size at most $C \cdot n/(\Delta - 1)^{r-1}$ for some fixed C and large enough n. □

Now, we show a lower bound on the required certificate size to locally certify $\mathcal{P}_\Delta$.

Lemma 8. *For all r-local proof labeling schemes certifying $\mathcal{P}_\Delta$ of size s, it holds*

$$s(n) \geq \frac{n \cdot \varepsilon}{12(\Delta - 1)^r}$$

for a large enough n and all $\varepsilon < 1$.

Proof. The idea of the proof is inspired by the proof of Theorem 6.1 of Göös and Suomela [14]. Following their approach, we will show that for every supposed proof labeling scheme of size less than $(n \cdot \varepsilon)/(12(\Delta - 1)^r)$, we can construct an instance not in $\mathcal{P}_\Delta$ which the verifier would necessarily accept.

Suppose there exists an r-local proof labeling scheme $(\mathcal{A}, f)$ certifying $\mathcal{P}_\Delta$ such that for every n' there exists $n \geq n'$ such that $s(n) < (n \cdot \varepsilon)/(12(\Delta - 1)^r)$. For an instance $H_i \in \mathcal{P}_\Delta$, let $T(H_i)$ denote $V[R(H_i), r]$. Let $H_j \in \mathcal{P}_\Delta$ and let $\sim$ be a binary relation on $\mathcal{P}_\Delta$ defined so that $H_i \sim H_j$ if and only if $f(H_i)[T(H_i)] = f(H_j)[T(H_j)]$ and $H_i[T(H_i)] = H_j[T(H_j)]$, that is both the subgraphs on $T(H_i)$, $T(H_j)$, and their certificates as assigned by f are the same. The equality of induced subgraphs here means the equality of the identifiers, the labels, and the edges. Note that $\sim$ is an equivalence. See again Figure 1 for an illustration.

Let $\mathcal{P}_\Delta[n]$ be the set of instances in $\mathcal{P}_\Delta$ on n vertices and with a fixed identifier assignment, meaning the identifier of a vertex with a given position in the tree is the same in all the instances.

Claim. For all n', there exists $n \geq n'$ and $H_1, H_2 \in \mathcal{P}_\Delta[n]$ such that $H_1 \sim H_2$ and $S(H_1) \neq S(H_2)$.

Proof. We will show that for large enough n, the number of possible binary sequences in the leaves of instances in $\mathcal{P}_\Delta[n]$ is greater than the number of equivalence classes of $\sim$ when restricted to $\mathcal{P}_\Delta[n]$.

By the assumption, each vertex has less than $(n \cdot \varepsilon)/(12(\Delta-1)^r)$ certificate bits, thus for an instance $H_i \in \mathcal{P}_\Delta[n]$, there are at most $2^{(n\cdot\varepsilon)/(12(\Delta-1)^r)\cdot|T(H_i)|}$ different certificate assignments on $T(H_i)$, and at most $(\Delta-1)^{|T(H_i)|}$ different assignments of labels on $T(H_i)$. The rest of the structure on $T(H_i)$, including the identifiers is fixed by the fact that $H_i \in \mathcal{P}_\Delta[n]$.

Furthermore, observe that $|T(H_i)| = 1 + 2\sum_{i=0}^{r-1}(\Delta-1)^i \leq 3(\Delta-1)^r$ as $\Delta \geq 3$. In total, we have that $\sim$ has on $\mathcal{P}_\Delta[n]$ at most $2^{(n\cdot\varepsilon)/4} \cdot (\Delta-1)^{3(\Delta-1)^r}$ different equivalence classes.

On the other hand, each instance has at least $n/4$ leaves in the left subtree and thus there are at least $2^{n/4}$ different possible binary strings in the left subtree. It remains to observe that

$$2^{(n\cdot\varepsilon)/4} \cdot (\Delta-1)^{3(\Delta-1)^r} < 2^{n/4}$$

for large enough n. Therefore by the pigeonhole principle, there are $H_1, H_2 \in \mathcal{P}_\Delta[n]$ such that $S(H_1) \neq S(H_2)$ and $H_1 \sim H_2$. □

Now, we take $H_1, H_2 \in \mathcal{P}_\Delta[n]$ such that $H_1 \sim H_2$ and $S(H_1) \neq S(H_2)$ and construct $H' = (V', E', L')$ by starting with $H_1[T(H_1)] = H_2[T(H_2)]$ and completing the left subtree by $LT(H_1)$ and the right subtree by $RT(H_2)$. Formally, let $L_S(G)$ be the neighbour of $R(G)$ in $LT(G)$ and $R_S(G)$ the neighbour in $RT(G)$. Then

$$\begin{aligned} V' = {} & V(LT(H_1)) \cup V(RT(H_2)) \cup \{R(H_1)\} \\ E' = {} & E(LT(H_1)) \cup E(RT(H_2)) \cup \{R(H'), L_S(H_1)\} \cup \{R(H'), R_S(H_2)\}). \end{aligned}$$

Observe that the identifier assignment of H' is the same as those of H_1 and H_2, hence by construction, we have that H' satisfies Properties 1 and 2 and the verifier can not reject H' on their basis. Furthermore, observe that $H' \notin \mathcal{P}_\Delta$ as the string in the leaves does not satisfy Property 3.

Now, we choose the certificate assignment on H' as

$$P(v) = \begin{cases} f(H_1)(v) & \text{if } v \in LT(H') \cup \{R(H')\} \\ f(H_2)(v) & \text{otherwise} \end{cases}$$

Recall that f is the prover of the proof labeling scheme of $\mathcal{P}_\Delta$.

Claim. For all $v \in V(H')$ it holds $\mathcal{A}[H'[v,r], P[v,r]] = 1$.

Proof. First, recall that $T(H') = T(H_1) = T(H_2)$, and by construction of P, we also have $P[T(H')] = f(H_1)[T(H_1)] = f(H_2)[T(H_2)]$. Observe that if $v \in LT(H') \cup \{R(H')\}$, we have $H'[v,r] = H_1[v,r]$ and $P[v,r] = f(H_1)[v,r]$ and thus $\mathcal{A}[H'[v,r], P[v,r]] = 1$. Similarly, if $v \in RT(H')$, we have $H'[v,r] = H_2[v,r]$ and $P[v,r] = f(H_2)[v,r]$ and thus $\mathcal{A}[H'[v,r], P[v,r]] = 1$. □

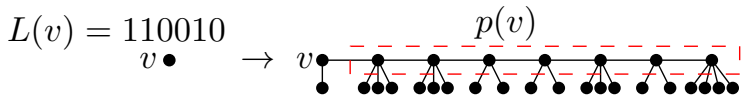

Fig. 2. An example of a vertex v in a graph $G \in \mathcal{P}^l$ and the same vertex in $g(G)$

Hence, there is an instance $H' \notin \mathcal{P}_\Delta$ which is accepted by $\mathcal{A}$, contradicting the assumption that $(f, \mathcal{A})$ certifies $\mathcal{P}_\Delta$. This finishes the proof. □

Similar approach can be used in a graph property without labels. The idea behind it is to encode the labels using some subgraphs.

We denote the optimal certificate size of an r-local proof labeling scheme for a property $\mathcal{P}$ as $s^*(\mathcal{P}, r, n)$. An r-local proof labeling scheme has optimal certificate size s if there is no other r-local proof labeling scheme of size s' such that there is n with $s'(n) < s(n)$.

Lemma 9. *For every property $\mathcal{P}^l$ with r-local proof labeling scheme of at most polynomial and at least logarithmic size, there exists a property $\mathcal{P}^u$ with no labels, except for unique identifiers, such that*

$$s^*(\mathcal{P}^l, r, n) \leq c_1 s^*(\mathcal{P}^u, r, n) \leq c_2 s^*(\mathcal{P}^l, r, n),$$

for every r and every large enough n, and where c_1, c_2 are fixed positive constants.

The result is not surprising and the idea of the proof is simple but technical, therefore we decided to move the proof to Appendix B. An example of the encoding of a label into a subgraph can be seen in the Figure 2. A single leaf vertex and a path $p(v)$ is attached to each vertex v. The i-th vertex of $p(v)$ has 2 or 3 leaf neighbours depending on the i-th bit of $L(v)$. The last vertex represents the end of the string. We then assume that we are provided with a proof labeling scheme for the unlabeled property and make the decision based on this scheme.

Now, we are ready to prove there are proof labeling schemes, such that the increase of certificate size by $C(\Delta-1)^{\delta-1}$ is necessary when decreasing the radius by δ.

Theorem 2. *There is an r-local proof labeling scheme, certifying a property without labels, of size s_r such that after decreasing its radius by δ, it holds for any possible resulting $r - \delta$-local proof labeling schema of size $s_{r-\delta}$ and every large enough n*

$$s_{r-\delta}(n) \geq s_r(n) \cdot C(\Delta - 1)^{\delta-1}$$

where Δ is the maximum degree of the input graph and C is a fixed constant.

Proof. Consider the property $\mathcal{P}_\Delta$. By Lemma 7, it can be certified by an r-local proof labeling schema of size s_r with $s_r(n) \leq C' \cdot n/(\Delta-1)^{r-1}$ for every large enough n. By Lemma 8, we have

$$s_{r-\delta}(n) \geq (n \cdot \varepsilon)/(12(\Delta-1)^{r-\delta}) \geq \frac{\varepsilon}{12C'} \cdot s_r(n) \cdot (\Delta-1)^{r-1-(r-\delta)} =$$
$$= s_r(n) \cdot C(\Delta-1)^{\delta-1}$$

for every large enough n and a fixed C.

By Lemma 9, for any r and $\mathcal{P}_\Delta$, there exists a property $\mathcal{P}^u$ with optimal certificate size s^u_r of an r-local proof labeling scheme such that: $C_1 s^u_r(n) \geq s_r(n) \geq C_2 s^u_r(n)$. Applied on both s_r and $s_{r-\delta}$, we obtain

$$C_1 s^u_{r-\delta}(n) \geq s_{r-\delta}(n) \geq s_r(n) C(\Delta-1)^{\delta-1} \geq s^u_r(n) C_2 \cdot C(\Delta-1)^{\delta-1}.$$

□

5 Saving on the log factors on paths

In Section 4, we proved that in general when reducing the verification radius, a blow-up of the certificate size is necessary. Roughly, the size of the certificates for radius 1 needs to be multiplied by the size of the ball of radius d, compared with certificates for radius d. In the expression of Theorem 1, there are two additional terms: one related to the input labeling (which seems difficult to remove), and one related to identifiers, of the form "size of the ball" multiplied by $O(\log n)$. In this section, we show that when the graph is a path, this last term can be replaced by a simple additive $O(\log n)$. In other words, it is not always necessary to encode locally all the identifiers of the ball at distance d.

Theorem 3. *Consider a property $\mathcal{P}$ on paths. If there exists a proof-labeling scheme at distance d for $\mathcal{P}$ using certificates of size $s(n)$, then there exists a proof-labeling scheme at distance 1, with certificates of size $O((2d+1)s(n) + \log n)$.*

Note that without additional work, Theorem 1 gives size $O((2d+1)s(n) + (2d+1)\log n)$ (forgetting about input labels).

The full proof is deferred to Appendix C, we only give the intuition here. In the general approach to reduce the verification radius, we need to write the identifiers of all the nodes of the ball at distance d in the new certificates (in addition to the old certificates and the inputs). This is because otherwise the new verifier would not be able to safely simulate the old verifier. Now, if we imagine that the identifier assignment is fixed, and not adversarial, we could go without that: a node would only check its own identifier and deduce safely the identifiers of its neighbors (forgetting about symmetry issues). What we prove is that if the graph is a path, then the prover can give and certify a new identifier assignment, 1, 2, 3, ..., and give certificates related to this virtual assignment. And then the verifier can check that it would accept in the virtual identifier assignment, which is enough to prove the correctness of the input graph. This requires the prover to give only one identifier in the new certificates (the one of the node at hand) and not all the identifiers of the ball, which makes the multiplicative $O(\log n)$ become an additive $O(\log n)$. We refer the reader to the appendix for discussion of this "virtual identifier" technique, which, as far as we know, has never been used.

6 Conclusion

There are several remaining interesting open questions regarding the role of radius in local certification. An open question is the price of reducing radius considering some other family of graphs than bounded degree graphs. An example of such a family may be planar graphs (degeneracy).

Another question to consider is the price of decreasing radius depending on the properties being certified. While our approach works in general, there may be more efficient certification methods for specific properties.

Acknowledgment. This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS23/205/OHK3/3T/18 and by the Czech Science Foundation Grant no. 24-12046S.

References

1. Karine Altisen, Stéphane Devismes, Swan Dubois, and Franck Petit. *Introduction to Distributed Self-Stabilizing Algorithms*. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2019. doi:10.2200/S00908ED1V01Y201903DCT015.
2. Virgina Ardévol Martínez, Marco Caoduro, Laurent Feuilloley, Jonathan Narboni, Pegah Pournajafi, and Jean-Florent Raymond. Lower bound for constant-size local certification. In *International Symposium on Stabilizing, Safety, and Security of Distributed Systems*, pages 239–253. Springer, 2022.
3. Alkida Balliu, Sebastian Brandt, Fabian Kuhn, Krzysztof Nowicki, Dennis Olivetti, Eva Rotenberg, and Jukka Suomela. Local advice and local decompression. *arXiv preprint arXiv:2405.04519*, 2024.
4. Nicolas Bousquet, Laurent Feuilloley, and Sébastien Zeitoun. Local certification of local properties: Tight bounds, trade-offs and new parameters. In *41st International Symposium on Theoretical Aspects of Computer Science, STACS 2024*, volume 289, pages 21:1–21:18, 2024. doi:10.4230/LIPICS.STACS.2024.21.
5. Keren Censor-Hillel, Ami Paz, and Mor Perry. Approximate proof-labeling schemes. *Theoretical Computer Science*, 811:112–124, 2020.
6. Shlomi Dolev. *Self-Stabilization*. MIT Press, 2000. URL: http://www.cs.bgu.ac.il/%7Edolev/book/book.html.
7. Laurent Feuilloley. Introduction to local certification. *Discrete Mathematics & Theoretical Computer Science*, 23(Distributed Computing and Networking), 2021.
8. Laurent Feuilloley, Pierre Fraigniaud, Juho Hirvonen, Ami Paz, and Mor Perry. Redundancy in distributed proofs. *Distributed Comput.*, 34(2):113–132, 2021. doi:10.1007/S00446-020-00386-Z.
9. Laurent Feuilloley, Pierre Fraigniaud, Pedro Montealegre, Ivan Rapaport, Éric Rémila, and Ioan Todinca. Local certification of graphs with bounded genus. *Discrete Applied Mathematics*, 325:9–36, 2023. URL: https://www.sciencedirect.com/science/article/pii/S0166218X22003833, doi:https://doi.org/10.1016/j.dam.2022.10.004.
10. Orr Fischer, Rotem Oshman, and Dana Shamir. Explicit space-time tradeoffs for proof labeling schemes in graphs with small separators. In *25th International Conference on Principles of Distributed Systems (OPODIS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

11. Pierre Fraigniaud, Mika Göös, Amos Korman, and Jukka Suomela. What can be decided locally without identifiers? In *Proceedings of the 2013 ACM symposium on Principles of distributed computing*, pages 157–165, 2013.
12. Pierre Fraigniaud, Juho Hirvonen, and Jukka Suomela. Node labels in local decision. *Theoretical Computer Science*, 751:61–73, 2018.
13. Pierre Fraigniaud, Amos Korman, and David Peleg. Towards a complexity theory for local distributed computing. *Journal of the ACM (JACM)*, 60(5):1–26, 2013.
14. Mika Göös and Jukka Suomela. Locally checkable proofs in distributed computing. *Theory of Computing*, 12(1):1–33, 2016.
15. Amos Korman and Shay Kutten. Distributed verification of minimum spanning trees. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, PODC '06, New York, NY, USA, 2006. Association for Computing Machinery. doi:10.1145/1146381.1146389.
16. Amos Korman, Shay Kutten, and David Peleg. Proof labeling schemes. *Distributed Comput.*, 22(4):215–233, 2010. doi:10.1007/S00446-010-0095-3.
17. Rafail Ostrovsky, Mor Perry, and Will Rosenbaum. Space-time tradeoffs for distributed verification. In Shantanu Das and Sebastien Tixeuil, editors, *Structural Information and Communication Complexity*, pages 53–70, Cham, 2017. Springer International Publishing.
18. David Peleg. *Distributed computing: a locality-sensitive approach*. SIAM, 2000.

A Omitted proofs from Section 3

In this appendix, proofs of several lemmas from Section 3 are provided.

Proof (Lemma 1). Assume the lemma hypothesis holds. Observe that by Condition B1 and Condition B2 we have that $d(\mathtt{pt}(P,u,v)) = 0$ if and only if $u = v$ and $d_G(u,v) = 0$. From now on, let us use the notation $\mathtt{pt}(u) = \mathtt{pt}(P,u,v)$.

Suppose there is $u \in V$ such that $d(\mathtt{pt}(u)) < d_G(v,u)$, let u be such a vertex with minimum $d(\mathtt{pt}(u))$. By Condition B3 there is $x \in N(u)$ such that $d(\mathtt{pt}(x)) = d(\mathtt{pt}(u)) - 1$. Note that $u \neq v$. It follows from the choice of u that $d(\mathtt{pt}(x)) \geq d_G(x,v)$ and thus $d(\mathtt{pt}(u)) < d_G(v,u) \leq d_G(x,v)+1 \leq d(\mathtt{pt}(x))+1 = d(\mathtt{pt}(u))$, a contradiction.

Now suppose there is $u \in V$ such that $d(\mathtt{pt}(u)) > d_G(v,u)$, let u be such a vertex with minimum $d_G(v,u)$. As $u \neq v$, there is y such that $d_G(u,v) = d_G(y,v)+1$. It follows from the choice of u that $d(\mathtt{pt}(y)) \leq d_G(y,v)$ and by the previous argument $d(\mathtt{pt}(y)) = d_G(y,v)$. Hence, $d_G(v,y) = d(\mathtt{pt}(y)) = d_G(v,u) - 1 < d(\mathtt{pt}(u)) - 1$ and thus $d(\mathtt{pt}(y)) + 1 < d(\mathtt{pt}(u))$, which contradicts $\mathcal{A}(u) = 1$ due to Condition B3. □

Proof (Lemma 2). Let there be $x \in V$ and $u \in V[x,\delta]$ such that $\neg\mathtt{has\text{-}pt}(P,u,x)$, assume that u is such vertex with minimum $d_G(u,x)$. Note that $u \neq x$, otherwise $\mathcal{A}(u) = 0$ due to Condition B2.

Therefore, there is $y \in N(u)$ such that $d_G(u,x) = d_G(y,x) + 1$ and thus $\mathtt{has\text{-}pt}(P,y,x)$ by the choice of u, and by Lemma 1 $\mathtt{pt}(P,y,x)$ has encoded the correct distance. Hence, we have $y \in N(u)$ such that $\mathtt{has\text{-}pt}(P,y,x)$, $d_G(y,x) = d(\mathtt{pt}(P,y,x)) < \delta$, and $\neg\mathtt{has\text{-}pt}(P,u,x)$, thus $\mathcal{A}(u) = 0$ due to Condition B4, a contradiction.

Now suppose there is $x \in V, u \notin V[x,\delta]$ such that $\texttt{has-pt}(P,u,x)$, let u be such vertex with minimum $d(u,x)$. As $u \neq x$, it follows from Condition B3 that there is $y \in N(u)$ with $d(\texttt{pt}(P,y,x)) = d(\texttt{pt}(P,u,x)) - 1$. By the choice of u, we have $y \in V[x,\delta]$ and by Lemma 1 $\texttt{pt}(P,y,x)$ has encoded the correct distance. We have $\delta < d_G(u,x) \leq d_G(y,x)+1 \leq \delta+1$, and hence $d_G(u,x) = d_G(y,x)+1$. Therefore $d(\texttt{pt}(P,y,x)) = \delta$ and $\mathcal{A}(u) = 0$ due to Condition B4, a contradiction. □

Proof (Lemma 3).

Suppose there are $v,u,x \in V$ such that $C(\texttt{pt}(P,v,x)) \neq C(\texttt{pt}(P,u,x))$. By Lemma 2, there is $\texttt{pt}(P,y,x)$ if and only if $y \in V[x,\delta]$, and therefore there are two incident vertices $v',u' \in V[x,\delta]$ such that $C(\texttt{pt}(P,v',x)) \neq C(\texttt{pt}(P,u',x))$. Then $\mathcal{A}(v') = 0$ due to Condition B5, a contradiction. □

Proof (Lemma 4). Suppose there is $p = \texttt{pt}(P,v,x)$ that is not well-formed, that is $D(p) \neq N(x)$ or $L(p) \neq L(G)(x)$. Note that due to Condition B2, there is $\texttt{pt}(P,x,x)$ which is well-formed. Due to Lemma 2 and the connectedness of $G[x,\delta]$, there are two incident $u',v' \in V[x,\delta]$ such that $\texttt{pt}(P,u',x)$ is well-formed and $\texttt{pt}(P,v',x)$ is not well-formed. It follows that $\mathcal{A}(v') = 0$ due to Condition B5, a contradiction. □

Proof (Lemma 5). It follows from $\mathcal{A}(v) = 1$ and Condition B2 that every vertex has a packet originating from itself, thus $V[u,r-\delta] \subseteq \omega(\mathcal{B}[P,u,r-\delta])$ for every u. For every $x \in V$ with $r-\delta < d(u,x) \leq r$, note that on a shortest path between u and x, there is y such that $d(u,y)+d(y,x) = d(u,x)$ and $d(u,y) = r-\delta$. It follows that $d(y,x) = d(u,x) - d(u,y) \leq \delta$. Hence by Lemma 2, it holds $\texttt{pt}(P,y,x) \in \mathcal{B}[P,u,r-\delta]$ and thus $x \in \omega(\mathcal{B}[P,u,r-\delta])$. □

Proof (Lemma 6). Suppose that P is the certificate assignment so that $\mathcal{A}(v) = 1$ for all v, then we claim that there is P' such that $\mathcal{A}_r(G[v,r],P'[v,r],v) = 1$ for all v and thus $G \in \mathcal{P}$. Let $P'(u) = C(\texttt{pt}(P,u,u))$ and note that by Lemma 3, we have that $C(\texttt{pt}(P,v,u)) = P'(u)$ for all v with $\texttt{has-pt}(P,v,u)$. Furthermore, by Condition B2 it holds $\texttt{has-pt}(P,u,u)$ for all u, hence P' is well-defined.

Now let $v \in V$, then by Lemma 5 v has access to a packet originating from each vertex of $V[v,r]$. Each such packet is well-formed by Lemma 4, therefore by Observation 1 we have $G(\texttt{pts}(P,v)) = G[v,r]$. Due to Condition B6, we have $\mathcal{A}_r(G(\texttt{pts}(P,v)), C(\texttt{pts}(P,v)), v) = \mathcal{A}_r(G[v,r],P'[v,r]) = 1$ and hence $G \in \mathcal{P}$. □

B Proof of Lemma 9

We will define a graph property of unlabeled graphs and show mappings between the properties. Throughout the following proof, we always consider a fixed radius. When $\mathcal{P}$ and r are clear from the context, we use simply $s^*(n)$ instead of $s^*(\mathcal{P},r,n)$.

Proof. The goal to show that for any P^l, we can construct P^u such that the optimal r-local proof labeling schemes of these two properties have certificate sizes that differ by a constant factor.

This is achieved in two steps: first, we construct a proof labeling scheme of P^u which essentially simulates the optimal proof labeling scheme of P^l with a small overhead in the size of certificates. Subsequently, we also construct a proof labeling scheme of P^l which simulates the optimal proof schema of P^u, again with only a small overhead.

We now define P^u. For that end, we show how to transform each $G \in P^l$ and encode its labels into the graphs structure, the resulting graph will be denoted as $g(G)$.

We define $g(G)$ the graph constructed from G. For an example see Figure 2 First, to each vertex $v \in V$ a leaf vertex is attached. Then, for each $v \in V$ a path $p(v)$ is created of the length of $|L(v)| + 1$ and attached to v. Let $p(v)_i$ be the i-th vertex of the path $p(v)$ and $L(v)_i$ the i-th bit of the label $L(v)$. A certain number of leaf vertices is attached to each vertex of $p(v)$ in the following way:

- if $L(v)_i = 0$ then 2 leaves are attached to $p(v)_i$,
- if $L(v)_i = 1$ then 3 leaves are attached to $p(v)_i$,
- if $i = |L(v)| + 1$ then 4 leaves are attached to $p(v)_i$.

Note that as the property P^l is closed under permutation of identifiers, the resulting graph may also have any choice of identifiers. For convenience, we assume that the identifiers of the *original* vertices of G are preserved in $g(G)$. The set of graphs created by this procedure from P^l is denoted as P^u (with all possible identifier permutations). By the $L(G)$ we denote the largest label on graph G.

Claim. It holds that $|V(g(G))| \leq 5|V(G)| \cdot |L(G) + 1|$.

Proof. For each vertex, one leaf is added and a path of the size of the labels with at most 4 additional leaves for each vertex of the path. Together that is $1 + 4|L(G) + 1|$ vertices per vertex of the original graph. Together that is $|V(G)| \cdot (1 + 4|L(G) + 1|)$. For convenience we simplify the second factor to $5|L(G) + 1|$. □

This increase in the number of vertices is the reason for the at most polynomial certificate size. If we considered certificates with larger size, then the unlabeled graph would be allowed to have asymptotically larger certificates. As any unlabeled property can be certified with certificates of size $O(n^2)$, this restriction is justified.

Now, we are ready to define an r-local proof labeling scheme (f^u, A^u) of P^u based on the optimal proof labeling scheme of P^l, say (f^{l*}, A^{l*}). We shorten $s^*(\mathcal{P}^l, r, n)$ to $s^*_l(n)$, similarly with s^*_u. Note that following arguments do not change with r and thus hold for every r.

Claim. $s^*_u(n) \leq c_2 s^*_l(n)$ for a fixed c_2.

Proof. Let $H \in \mathcal{P}^u$ and $G \in \mathcal{P}^l$ be graphs such that $g(G) = H$. We define $g'(H) = G$.

The certificates assigned by f^u are tuples (b_l, b_p, s, o).

- b_l is one bit denoting whether a vertex v is a leaf or not. If v is a leaf, the rest of the certificate is empty.
- b_p is one bit denoting whether a vertex v is a vertex from $p(w)$ for some vertex w.
- s is a bit string from which the labels are constructed.
- If v has exactly one leaf neighbour, $o = f^{l*}(g'(H))(v)$.

The verifier $\mathcal{A}^u$:

- On a leaf vertex it accepts if and only if $b_l = 1$.
- On a vertex with 4 leaf neighbours it accepts if and only if $b_p = 1$ and s is empty.
- On a vertex with 3 leaf neighbours accepts if and only if:
  - it has two non-leaf neighbours, one, denoted as w, with the element s one bit shorter, the other either with $b_p = 0$ and the same s or with $b_p = 1$ and s one bit longer.
  - the element s is a concatenation of a 1 and the element s of w.
- On a vertex with 2 leaf neighbours accepts if and only if:
  - it has two non-leaf neighbours, one, denoted as w, with the element s one bit shorter, the other either with $b_p = 0$ and the same s or with $b_p = 1$ and s one bit longer.
  - the element s is a concatenation of a 0 and the element s of w.
- On a vertex with 1 leaf neighbour accepts if and only if:
  - $b_p = 0$,
  - it has exactly one neighbour with $b_p = 1$ and the element s is the same as the element s of that neighbour,
  - $\mathcal{A}^l$ executed for each v with labels corresponding to s and certificates to o accepts on the local neighbourhood $g'(H)[v, r]$.

Now we show the size of $(f^u, \mathcal{A}^u)$. The elements b_l and b_p consist of one bit and o is of the same size as the proof labeling scheme $(f^l, \mathcal{A}^l)$. Recall that in $\mathcal{P}^u$ the structure of $p(v)$ for every v is of a constant size relative to the size of the whole graph. The element s of a vertex v consists of as many bits as there are vertices in $p(v)$. Hence $s^*_u(n) \leq c_2 s^*_l(n)$ for a fixed c_2.

□

Claim. $s^*_l(n) \leq c s^*_u(n)$ for a fixed c.

Proof. We now define a proof labeling scheme (f^l, A^l) based on the optimal proof labeling scheme of P^u, say (f^{u*}, A^{u*}). We want to encode the neighborhood of each $v \in g(G)$, including its $p(v)$, leaves and certificates of all of these into the certificate of $v \in G$.

For each graph $G \in \mathcal{P}$ the certificate assignment $f^l(G)$ constructs the corresponding $g(G)$ and gives it as an input to f^u. For each $v \in V$ the certificate

$f^l(G)(v)$ consists of $f^u(g(G))(v)$ and an encoding of $p(v)$ including certificates of any vertex $w \in p(v)$, as well as the certificate of the one leaf neighbour of v.

On each v, the verifier A^l takes the certificate of v and constructs from it the local subgraph $g(G)[v, r] \cup p(v)$ with the certificates and verifies that the path $p(v)$ and the leaves attached to it were constructed correctly from the label $L(v)$. It then executes A^u on each vertex of the reconstructed graph. If $\mathcal{A}^u$ would reject on any vertex, the verifier $\mathcal{A}^l$ rejects well.

As mentioned above, the subgraph $p(v)$ and the leaves adjacent to any vertex in $p(v)$, along with the certificates and identifiers of those vertices, are encoded in the certificate of any vertex v. This gives us that the size of $f^l(G)(v)$ is at most $5|L(v) + 1| \cdot (|f^u(g(G))| + \log(n))$. The $\log(n)$ accounts for the identifiers of the encoded vertices.

Thus we have proven that $\mathcal{A}^l$ accepts if and only if $\mathcal{A}^u$ accepts and by the first claim it holds $c_1 s_l^*(n) \leq s_u^*(n)$ for some $c_1 > 0$ as $c_1 s_l(n) \leq s_u^*(n)$, where $s(n)$ is the size of f^l.

□

And hence the lemma is proven.

□

C Proof of Theorem 3

Let us first remind the statement.

Theorem. Consider a property $\mathcal{P}$ on paths. If there exists a proof-labeling scheme at distance d for $\mathcal{P}$ using certificates of size $s(n)$, then there exists a proof-labeling scheme at distance 1, with certificates of size $O((2d + 1)s(n) + \log n)$.

The core of the proof of this statement is not specifically about reducing the checkability radius, but about working on a worst case identifier assignment versus working on a more structured identifier assignment. We first state and prove a lemma capturing this intuition, and then show how this adapts to our context.

Throughout this section, suppose the graph is a path. The two *canonical identifier assignments* of a path, whose endpoints are denoted a and b, are the ones where the identifier of every node u is its distance to a (resp. b). (In other words the identifiers are 1,2,3,... in the order of the path.) Let a *weak local certification* be the same as a local certification, except that the identifier assignment is promised to be canonical.

Lemma 10. *Consider a property $\mathcal{P}$ on paths. Any weak local certification with certificates of size $s(n)$ and radius r can be turned into a (standard) local certification with certificates of size $O(s(n) + \log n)$ and radius r. In addition, the new certification is independent of the identifier assignment.*

Proof (Proof of Lemma 10). Let us first describe the behavior of the new prover f' on correct instances, based on the old prover f. The prover f' first chooses

one of the canonical identifier assignments, call it I^*. Then it assigns to every node v a first label containing $I^*(v)$. Finally, the new prover f' simulates the old prover on (G, I^*) (which is possible since I^* is canonical), that is, it assigns a second label to every node v which is $f(G, I^*, v)$.[5] The certificate of a vertex is the concatenation of the two labels.

We now describe the verifier. The new verifier $\mathcal{A}'$ first checks that the certificates it sees are composed of two parts: one encoding a new identifier, and another label (if it is not the case, it rejects). For a vertex v, let us call these two parts $J(v)$ and $L(v)$. The verifier $\mathcal{A}'$ then checks that J corresponds to a canonical ordering in its view (this is straightforward). Finally, it simulates the old verifier taking J as the identifier assignment, and L as certificates. More formally, if it has not rejected yet, then it outputs $\mathcal{A}(G, L, J, v)$.

Let us prove that this scheme is correct. Suppose that $\mathcal{A}'$ accepts on all the vertices of a given path with a given certificate assignment. Then this certificate assignment can be decomposed into a proper canonical ID assignment, and a another labeling which makes $\mathcal{A}$ accept. Since $\mathcal{A}$ is a correct proof-labeling scheme when the ID assignment is canonical, this path must be correct. Conversely, for a negative instance, either the certificates are not properly encoded, or the encoded ID assignment is not canonical, or $\mathcal{A}$ rejects, which leads to a reject in at least one vertex.

The new proof labeling scheme is independent of the identifier assignment, since these are never used in the scheme. □

We now adapt the scheme to prove our theorem.

Proof (Proof of Theorem 3). The proof of the theorem is a mix of the proof of the previous lemma and of the general intuition for reducing the radius. We start with the scheme $(f, \mathcal{A})$ at distance d. First, we build the scheme $(f', \mathcal{A}')$ at distance d given by the proof of Lemma 10. Second, we use the general technique described in the previous sections to reduce the verification radius: a new prover f'' gives to every vertex all the certificates that f' would give to the vertices in the ball at distance d, along with the structure of the graph and the identifiers. Finally, to save on the certificate size, we can remove all identifiers from the certificates, except for the one of the vertex itself. In other words, a vertex receives in its certificate: its new identifier, the certificates that f' would assign to its neighbours at distance at most d, and the structure of this neighbourhood. The key point is that since the identifier assignment used in $(f', \mathcal{A}')$ is canonical, the vertices can easily check the consistency with their neighbours. Then the verifier $\mathcal{A}''$ will simply use the canonical ordering and proceed similarly to the verifier of the proof of Theorem 1. □

We expect that similar techniques can be used for other very structured graphs, such as grids. To which extent it is possible to generalize such a theorem is a very intriguing question. It has a similar flavor as the trade-off conjecture in

[5] Since we are modifying the identifier assignment, we make it appear explicitly in the notation for this proof.

the case of colorings [4], in the sense that in both cases the question boils down to how well we can recover a node labeling in a graph if we are given only some of these labels (in the current proof the labels are the identifiers, while in [4] these are the colors). See also [3] for similar questions.