Brief Announcement: Decreasing Verification Radius in Local Certification

Jan Matyáš Křišťan
Faculty of Information Technology, Czech Technical University in Prague, Czech Republic

Josef Erik Sedláček
Faculty of Information Technology, Czech Technical University in Prague, Czech Republic

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.

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.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Theory of computation → Graph algorithms analysis

Keywords and phrases Local certification, locally checkable proofs, proof-labeling schemes, graphs, distributed computing

Digital Object Identifier 10.4230/LIPIcs.DISC.2024.49

Related Version *Full Version*: `https://arxiv.org/abs/2408.10757`

Funding 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.

Acknowledgements We would like to thank Laurent Feuilloley for his helpful discussions and suggestions.

1 Introduction

The problem studied in this paper involves certifying a global graph property without having complete knowledge of the entire graph. In particular, we study the model of locally checkable proofs of Göös and Suomela [4].

In this model, an 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 does not hold. 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 does not hold. 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*. 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.

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 gives each graph with the property a certificate assignment that is accepted by $\mathcal{A}$. An intuitive example is k-colorability. If k is a constant the coloring can be provided in the certificates.

Previous work and our contribution

Similar models have been studied under different names [5, 6]. The name *local certification* is a general term used for the similar models [1].

It has been previously shown how, and under which conditions, certificate size can be decreased at the cost of increasing the visibility radius [2, 3]. We provide a similar result, showing how the visibility radius can be decreased at the cost of increasing the certificate size. We also provide a corresponding lower bound on the necessary certificate size increase.

There is a crucial distinction between these two problems. While the mentioned results allow increasing the radius while decreasing the size of certificates in the general case, the implied inverse procedure of decreasing the radius 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.

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 assigned integer *identifiers*, and we assume that $V = \{1, \ldots, n\}$. The set of neighbours of a vertex v is denoted as $N_G(v)$, distance between u, v as $d_G(u, v)$, and the set of vertices within distance r from v as $V[v, r]$, also called the r-local neighborhood of v.

A *graph property* is a set of graphs that is closed under isomorphism, that is, its membership does not depend on the choice of identifiers. A *certificate assignment* P for G is a function $P\colon V(G) \to \{0,1\}^*$ that associates a *certificate* with each vertex. 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.

We denote the induced subgraph $G[V[v, r]]$ as $G[v, r]$, and the restriction of P to $V[v, r]$ 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 assigns to each $G \in \mathcal{P}$ a certificate assignment 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 a size $s : \mathbb{N} \to \mathbb{N}$ if $|f(G)(v)| \leq s(|V(G)|)$ for all $G \in \mathcal{P}$ and $v \in V(G)$.

3 Decreasing the radius of a proof labeling scheme

In this section, we state that given an r-local $(f_r, \mathcal{A}_r)$ certifying a property $\mathcal{P}$, we can construct an $(r - \delta)$-local $(f, \mathcal{A})$ certifying $\mathcal{P}$ for any $\delta < r$ at the cost of increasing the certificate size. 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 the 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 $\mathcal{O}(\delta(\Delta \log(n) + s(n) + \ell(n)))$.

While the idea is simple, the proof is technical; therefore, due to space constraints, we decided to omit the proof from this brief announcement. The complete proof is available in the full version of the paper. Here we provide only the following 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.

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

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 and a lower bound on its size.

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), 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). The label $L(R(G))$ is empty.

This allows us to define $LT(G)$ and $RT(G)$ as the the subtrees rooted at the first and second child of $R(G)$ respectively. Furthermore, it allows us to naturally 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 $\mathcal{P}_\Delta$.

▶ **Lemma 2.** *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.*

The 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.

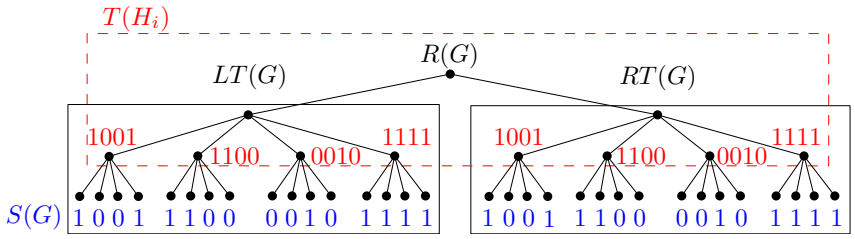

Figure 1 An example of a graph with property $\mathcal{P}_\Delta$ with $\Delta = 5$. Here, $R(G)$ is the root, $LT(G)$ and $RT(G)$ are the left and the right subtrees, $S(G)$ is the sequence in the leaves, and the red strings are certificates. The subgraph $T(H_i)$ is used in the proof of Lemma 3 and corresponds to $r = 2$.

The proof is straightforward but lengthy. The main idea is to encode for each vertex v in its certificate the string $S(v)$ of the whole subtree rooted in v. Since the verifier can see up to distance r, it is not necessary to encode the string in the vertices for which $d(v, R) < r$. See Figure 1 for an example and the full paper for the whole proof.

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

▶ **Lemma 3.** *For all r-local proof labeling schemes certifying $\mathcal{P}_\Delta$ of size s, it holds that $s(n) \geq (n \cdot \varepsilon)/(12(\Delta-1)^r)$ for a large enough n and all $\varepsilon < 1$.*

Proof. The idea is inspired by the proof of Theorem 6.1 of Göös and Suomela [4]. 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 $\sim$ 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 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 4. 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 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

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

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

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

▷ Claim 5. For all $v \in V(H')$ it holds $\mathcal{A}[H'[v, r], P[v, r]] = 1$.

It follows from the construction that the local neighbourhood of any v with is exactly the same as in the original graph. For the complete proof see the full version of the paper.

We have demonstrated that 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. ◀

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 6.** *There is an r-local proof labeling scheme of size s_r such that after decreasing its radius by δ, for any possible resulting $r - \delta$-local proof labeling scheme of size $s'_{r-\delta}$ and every large enough n, it holds that $s'_{r-\delta}(n) \geq s_r(n) \cdot C(\Delta - 1)^{\delta-1}$ 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 2, it can be certified by a proof labeling scheme of size s_r with $s_r(n) \leq C' \cdot n/(\Delta - 1)^{r-1}$ for every large enough n. By Lemma 3, for every large enough n and a fixed C, we have:

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

5 Conclusion

A 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.

In Section 4, the presented results require that we allow labels on the vertices of the input graph. We believe that the same results can be achieved for graphs without labels, by substituting the labels with an appropriate construction.

References

1 Laurent Feuilloley. Introduction to local certification. *Discrete Mathematics & Theoretical Computer Science*, 23(Distributed Computing and Networking), 2021. doi:10.46298/dmtcs.6280.

2 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.

3 Orr Fischer, Rotem Oshman, and Dana Shamir. Explicit space-time tradeoffs for proof labeling schemes in graphs with small separators. In Quentin Bramas, Vincent Gramoli, and Alessia Milani, editors, *25th International Conference on Principles of Distributed Systems (OPODIS 2021)*, volume 217 of *LIPIcs*, pages 21:1–21:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.OPODIS.2021.21.

4 Mika Göös and Jukka Suomela. Locally checkable proofs in distributed computing. *Theory of Computing*, 12(1):1–33, 2016. doi:10.4086/toc.2016.v012a019.

5 Amos Korman, Shay Kutten, and David Peleg. Proof labeling schemes. *Distributed Comput.*, 22(4):215–233, 2010. doi:10.1007/S00446-010-0095-3.

6 David Peleg. *Distributed computing: a locality-sensitive approach*. SIAM, 2000. doi:10.1137/1.9780898719772.