Computational complexity of distance edge labeling☆

Dušan Knop[a,b,1], Tomáš Masařík[a,b,*]

[a] *Department of Applied Mathematics of the Faculty of Mathematics and Physics at the Charles University in Prague, Malostranské náměstí 25, Praha 1, 118 00, Czech Republic.*
[b] *Computer Science Institute of Charles University, Faculty of Mathematics and Physics, Charles University in Prague, Malostranské náměstí 25, Praha 1, 118 00, Czech Republic.*

Abstract

The problem of Distance Edge Labeling is a variant of Distance Vertex Labeling (also known as $L_{2,1}$ labeling) that has been studied for more than twenty years and has many applications, such as frequency assignment.

The Distance Edge Labeling problem asks whether the edges of a given graph can be labeled such that the labels of adjacent edges differ by at least two and the labels of edges at distance two differ by at least one. Labels are chosen from the set $\{0, 1, \ldots, \lambda\}$ for λ fixed.

We present a full classification of its computational complexity—a dichotomy between the polynomial-time solvable cases and the remaining cases which are NP-complete. We characterize graphs with $\lambda \leq 4$ which leads to a polynomial-time algorithm recognizing the class and we show NP-completeness for $\lambda \geq 5$ by several reductions from Monotone Not All Equal 3-SAT.

Moreover, there is an absolute constant $c > 0$ such that there is no 2^{cn}-time algorithm deciding the Distance Edge Labeling problem unless the exponential time hypothesis fails.

Keywords: Computational complexity, distance labeling, line-graphs, exponential time hypothesis.

☆The results presented in this paper are an extension of a result presented at the conference IWOCA 2015 [1]. Research was supported by the project SVV-2015-260223 and by the project CE-ITI P202/12/G061 of GAČR.

*Corresponding author

Email addresses: knop@kam.mff.cuni.cz (Dušan Knop), masarik@kam.mff.cuni.cz (Tomáš Masařík)

[1] Author was supported by the project GAUK 1784214.

1. Introduction

We study the computational complexity of the distance edge-labeling problem. This problem belongs to a wider class of problems that generalize the graph coloring problem. The task is to assign a set of colors to each vertex, such that whenever two vertices are adjacent, their colors differ from each other. For a survey about this famous graph problem and related algorithms, see a survey by Formanowicz and Tanaś [2].

We are interested in the so-called *distance labeling*. In this generalization of the former problem the condition enforcing different colors is extended and takes into account also the second neighborhood of a vertex (or an edge). The second neighborhood is the set of vertices (or edges) at distance at most 2. For a survey about distance labelings, we refer to the article by Tiziana Calamoneri [3], as well as her updated online survey [4].

Graph distance labeling has been first studied by Griggs and Yeh [5, 6] in 1992. The problem has many applications, the most important one being frequency assignment [7]. The complexity of $\mathrm{L}_{2,1}$ labeling for a fixed parameter λ has been established by Fiala et al. [8]. They show a dichotomy between polynomial cases for $\lambda \leq 3$ and NP-complete cases for $\lambda \geq 4$.

Moreover, for the usual graph coloring problem there is a theorem of Vizing [9], which states that for the edge-coloring number $\chi'(G)$ it holds that $\Delta \leq \chi' \leq \Delta + 1$, where Δ is the maximum degree of the graph. For $\mathrm{L}_{2,1}$ labeling there is a general bound due to Havet et al. [10], namely $\lambda \leq \Delta^2$, for $\Delta \geq 79$.

Before we proceed to the formal definition of the corresponding decision problem, we give several definitions of a labeling mapping of a graph and of the minimal distance edge-labeling number. Note that the distance edge-labeling is equivalent to the distance vertex-labeling of the associated line-graphs. A *line-graph* $L(G)$ is a graph derived from another graph G such that vertices of $L(G)$ are edges of G and two vertices a, b of $L(G)$ are connected by an edge whenever a, b (as edges of G) are adjacent. We define the *distance* between edges of a graph as their distance in the corresponding line-graph.

Definition 1 (Edge-labeling mapping). Let $G(V, E)$ be a graph. A mapping $f'_{2,1}\colon E \to \mathbb{N}$ is an *edge-labeling*, if it satisfies:

- $|f'_{2,1}(e) - f'_{2,1}(e')| \geq 2$ for neighboring edges (i.e. those in the distance one),
- $|f'_{2,1}(e) - f'_{2,1}(e')| \geq 1$ for edges at distance two.

As usual, we are interested in a labeling that minimizes the number of labels used by a feasible labeling.

Definition 2 (Minimum distance edge-labeling). Let G be a graph and $f'_{2,1}$ an edge-labeling mapping, we define the graph parameter $\lambda'_{2,1}$ as:

$$\lambda'_{2,1}(G) := \min_{f'_{2,1}} \max_{e \in E} f'_{2,1}(e).$$

The size of the range of a (not necessarily optimal) edge-labeling mapping $f'_{2,1}$ is called the *span*.

Definition 3 (DISTANCE EDGE LABELING **problem (also known as** $L'_{2,1}$) **).**

Input:	A graph G.
Parameter:	$\lambda \in \mathbb{N}$.
Question:	Is $\lambda'_{2,1}(G) \leq \lambda$?

1.1. Our results

Our main result is the following theorem about the dichotomy of the DISTANCE EDGE LABELING problem.

Theorem 1 (Dichotomy of distance edge-labeling). *The problem* $L'_{2,1}$ *is polynomial-time solvable if and only if* $\lambda \leq 4$. *Otherwise it is* NP*-complete.*

We derive Theorem 1 as a combination of Theorem 3 that describes all graphs with $\lambda'_{2,1} \leq 4$ and Theorem 8 presenting the NP-completeness result. Note that Theorem 8 also extends to the following inapproximability result:

Corollary 1. *The* DISTANCE EDGE LABELING *problem cannot be approximated within a factor of* $6/5 - \varepsilon$, *unless* P = NP.

Moreover, according to [11], the proof implies that the DISTANCE EDGE LABELING is paraNP-hard while parameterized by its span.

Using the well-established exponential time hypothesis it is possible to prove an exponential lower bound for a fixed span greater than 5.

Corollary 2 (An exponential lower bound for distance-edge labeling). *For every fixed span* $\lambda \geq 5$ *of* DISTANCE EDGE LABELING *problem there is a positive real s such that* DISTANCE EDGE LABELING *parametrized its size n cannot be solved in time* $2^{cn} n^{O(1)}$ *unless ETH fails.*

We prove it in Section 4 of the paper.

1.2. Preliminaries

We state several basic and well-known observations with the connection to Definition 3, as well as some notation used in this paper. For further standard notation in graph theory, we refer to the monograph of Diestel [12]. Without loss of generality we deal with connected simple undirected graphs.

The first observation gives a trivial lower-bound on $\lambda'_{2,1}(G)$.

Observation 1 (Max-degree lower-bound). *Let G be a graph and let* Δ *be its maximum degree. Then* $\lambda'_{2,1}(G) \geq 2(\Delta - 1)$.

Note that this observation gives also an upper bound on the max-degree of a graph G with $\lambda'_{2,1}(G) \leq \lambda$ for a given $\lambda \in \mathbb{N}$.

Observation 2 (The symmetry of distance labeling). *Let G be a graph, a mapping* $f\colon E \to \mathbb{N}$ *be a (not necessarily optimal) labeling with span* λ. *Then also the mapping* $f'(e) := \lambda - f(e)$ *is a valid labeling with the same span.*

We call such a derived labeling of the edges of a graph a λ*-inversion*.

1.2.1. On the Exponential Time Hypothesis

In this subsection we survey the Exponential Time Hypothesis (ETH for short)—a complexity theoretic assumption introduced by Impagliazzo, Paturi and Zane [13] an assumption for proving lower bounds for NP-hard combinatorial problems. We follow a survey on this topic by Lokshtanov et al. [14], which we also recommend as it contains more details on this topic.

A formal definition of a parameterized problem is taken according to Flumm and Grohe [11]:

Definition 4 (Parameterized problem). Let Σ be a finite alphabet. A parametrization of Σ^* (set of all words over the alphabet Σ) is a polynomial-time computable mapping $\kappa : \Sigma^* \to \mathbb{N}$. A parameterized problem is a pair $(L, \kappa) \in \Sigma^* \times \mathbb{N}$, where κ is a parametrization.

The hypothesis states that there is no subexponential time algorithm for a 3-SAT if we measure the time complexity by the number of variables in the input formula—denoted by n.

Exponential Time Hypothesis [13] There is a positive real s such that 3-SAT with parameter n cannot be solved in time $2^{sn}(n+m)^{\mathrm{O}(1)}$.

Due to a famous Sparsification Lemma of Calabro et al. [15] the Exponential Time Hypothesis has an equivalent form—but parameterized by the number of clauses m in a 3-CNF formula (instead of the number of variables).

Exponential Time Hypothesis—clause form There is a positive real s' such that 3-SAT with parameter m cannot be solved in time $2^{s'm}(n+m)^{\mathrm{O}(1)}$.

We will use this version of ETH for our lower bound result.

In a connection to ETH there is also a stronger version of the hypothesis—namely the Strong Exponential Time Hypothesis (SETH for short). The SETH gives a hypothesis on the lower bound on the running time of a more complex algorithm for solving a general SAT (without a bound on the number of variables in a clause).

Assuming ETH holds, it is possible to analyze a sequence of real constants $\{s_k\}$ for every $k \geq 3$, where s_k is defined as

$$s_k = \inf\{\delta\colon \text{there is an } O^*(2^{\delta n}) \text{ algorithm for } k\text{-SAT with } n \text{ variables}\}.$$

We further define the limit of this sequence as s_∞.

Theorem 2 ([16], [13]). *Assuming ETH, the sequence $\{s_k\}_{k\geq 3}$ is increasing infinitely often. Furthermore, $s_k \leq s_\infty(1 - \frac{d}{k})$ for some constant $d > 0$.*

Strong Exponential Time Hypothesis (SETH) $s_\infty = 1$

It is still an important open problem in this area, whether a "clause version" of SETH can be formulated bases on the SETH. But a Strong version of the Sparsification Lemma cannot be proved, as was shown by Santhanam and Srinivasan [17].

It the theory of parameterized algorithms and exact exponential algorithms, both ETH and SETH became a standard tools to derive lower bounds on the running time of (both kinds of) algorithms. Even though there is a specialized type of reduction between problems preserving the lower bound—namely the (Turing) SERF-T reduction—usually it is sufficient when the parameter dependence is linear in the parameterized reduction.

For the sake of completeness, we give a full definition of the SERF-T reduction, which we give in a parameterized setting:

Definition 5 (SERF-T reduction). A SERF-T reduction from parameterized problem (A, κ) to a parameterized problem (B, λ) is a Turing reduction M from A to B that has the following properties:

1. Given $\varepsilon > 0$ and an instance x of the problem A the Turing machine M runs in time $\mathrm{O}(2^{\varepsilon\kappa(x)}) \cdot |x^{\mathrm{O}(1)}|$, where $|x|$ is the bit-size of the input x.
2. For any query M running on the input x makes to the problem B with the input y it holds that $|y| = |x|^{\mathrm{O}(1)}$ and $\lambda(y) = \alpha\kappa(x)$, where the constant α may depend on ε while the constant hidden in the $\mathrm{O}(\cdot)$ notation may not.

2. Polynomial cases

In this section we give a full description of graphs admitting a labeling with a small number of labels, in particular graphs G with $\lambda'_{2,1}(G) \leq 4$. Moreover, these graphs can be recognized in polynomial time. This leads to Theorem 3, which is the main result of this section.

For the ease of presentation we split the proof and statement of the Theorem 3 into several lemmas, each for a particular value of $\lambda'_{2,1}(G)$.

Theorem 3 (Polynomial cases of distance edge-labeling). *For any graph G and for $\lambda = 0, 1, 2, 3, 4$ the* Distance Edge Labeling *problem $\lambda'_{2,1}(G) = \lambda$ (or $\lambda'_{2,1}(G) \leq \lambda$) can be solved in polynomial time. Moreover, it is possible to compute such a labeling in polynomial time.*

The proof is implied by following lemmas.

First observe, that for $\lambda < 4$ the graph cannot contain a vertex of degree 3. We use P_i as a symbol for the *path* on i vertices.

Lemma 4 (Graphs with $\boldsymbol{\lambda'_{2,1}(G) \leq 3}$).

- *The only graphs with $\lambda'_{2,1}(G) = 0$ are P_1 or P_2.*
- *There is no graph with $\lambda'_{2,1}(G) = 1$.*
- *The only graph with $\lambda'_{2,1}(G) = 2$ is P_3.*
- *Finally, graphs with $\lambda'_{2,1}(G) = 3$ are P_4 and P_5.*

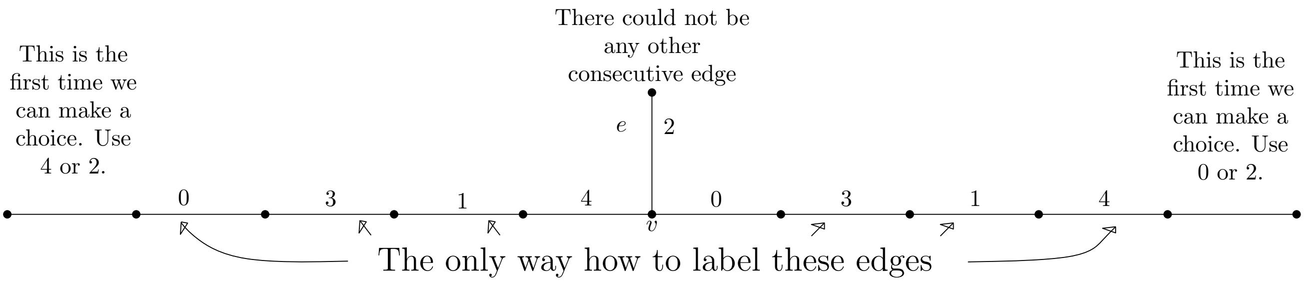
When $\lambda = 4$, the graph may contain vertices of degree 3. We call a vertex *hairy* if it is of degree 3 and at least one of its neighbors is of degree 1. We call this degree one vertex, together with the connecting edge, *pendant*. Note that any vertex of degree 3 in a graph G satisfying $\lambda(G) = 4$ cannot have all its neighbors of degree 2 or greater. It is easy to see that there is no labeling of span 4 of such a graph. We say that two hairy vertices are *consecutive*, if there is no other hairy vertex on a path between them or if there is the only hairy vertex on a cycle. In this particular case the vertex is consecutive to itself.

For the purpose of the following lemmas, we say that a graph is a *generalized cycle* if it is a cycle with several (possibly 0) pendant edges. We say that a graph is a *generalized path* if it is a path with several (possibly 0) pendant edges. All observations made in the last paragraphs imply the following lemma:

Lemma 5. *Let G be a graph satisfying $\lambda'_{2,1}(G) \leq 4$, then G is either a generalized path or a generalized cycle.*

On the contrary not every generalized cycle or path has $\lambda'_{2,1} \leq 4$. The following lemmas state all the conditions for a generalized cycle or path to satisfy $\lambda'_{2,1} \leq 4$.

Notation in the proofs. Both proofs are done by a case analysis. We use sequences of numbers representing labels on edges. Note that it follows from Observation 1 that only numbers $0, 2, 4$ can occur around a hairy vertex and any pendant edge must get label 2. For labelings we use sequences of numbers describing labels of consecutive edges and a symbol "|" for a hairy vertex—so there is a pendant edge on that vertex with label 2. This gives us immediately the following observation.



Observation 3 (The labeling of a hairy vertex and its neighborhood). *The neighborhood of a hairy vertex can be labeled only by a sequence* $0314|0314$ *or its λ-inversion* $4130|4130$.

Lemma 6. *Let $G = (V, E)$ be a generalized path. Let W be the set of all hairy vertices. Then $\lambda'_{2,1}(G) \leq 4$ if and only if for every consecutive pair $u, v \in W$ their distance $d = d_G(u, v)$ is either* 4*, or at least* 8.

Proof. We need to show that each sequence can be correctly labeled or that it is impossible to label it at all.

The easier fact is the existence of correct labelings. The following sequences can be extended by a sequence 0314 at the beginning to get sequences of length at least 8:

- $|0314|(d = 4)$,
- $|031420314|(d = 9)$,
- $|0314204130|(d = 10)$,
- $|03140240314|(d = 11)$.

Now we have to show that there are no valid sequences of length $1, 2, 3, 5, 6, 7$. Observation 3 banns immediately sequences of length $1, 2, 3$. Furthermore, the same observation also implies that there is no chance to overlap two sequences which is necessary to get lengths $5, 6$ or 7. □

Lemma 7. *Let $G = (V, E)$ be a generalized cycle. Let W be the set of all hairy vertices. Then $\lambda'_{2,1}(G) \leq 4$ if and only if for every consecutive pair $u, v \in W$ their distance $d = d_G(u, v)$ fulfills one of the following:*

- *$d = 4, 8, 9$ or $d \geq 11$,*
- *if there exists a consecutive pair with $d = 10$, then there is even number of such consecutive pairs, or there exists a consecutive pair with $d = 13, 14, 16$ or greater.*

Proof. Firstly it is easy to observe that cycles of any length without a hairy vertex can be labeled correctly.

For the proof we use all the facts already proved in the proof of Lemma 6. The difference between a generalized path and a cycle is that the generalized cycle is closed, and so we have to care about used labeling.

For all lengths of sequences presented so far, the sequence starts with the label 0 and ends with the label 4. Recall that the sequence of length 10 was $|0314204130|$. This sequence starts and ends with the same label—and this cause the incorrectness of labeling.

First we present a new sequence $|03140240240314|(d = 14)$—this sequence proves, that the only sequence that has to start and end with the same label is the one with length 10. Now it is clear that if there is even number of pairs with $d = 10$ then the constructed labeling is correct, which finishes the proof of the first part.

For the second part of the lemma, we have to show that for $d = 13, 14$ or $d \geq 16$, there is also a sequence that starts and ends with the label 0 and the impossibility of such a labeling for all other d. As usually, we begin with the desired sequences:

- $|4130240240314|(d = 13)$,
- $|41302403140314|(d = 14)$,
- $|413041302403140314|(d = 18)$.

In all these sequences the subsequence 024 can be repeated arbitrarily.

For the rest we already know, that all the sequences that starts and ends with 0 have to start with the subsequence |0314 and end with the subsequence 4130|. As these subsequences cannot be glued together, we have to glue them through another subsequence, which we call a *connector.* Note that the connector subsequence cannot be of length 1, because the starting and ending subsequences starts and ends with the same label. This already forbids all $d \leq 10$.

The connector can be the sequence 20 or 02. The resulting sequence is the sequence 0314204130, which we are already familiar with. Again it is impossible to prolong the sequence by a subsequence of length one, two or five. It is easy to see that the only possibilities are to put:

- 0314 to the beginning,
- 4130 to the end,
- 420 right after the connector.

This forbids the sequences of length $11, 12, 15$ and finishes the proof. □

3. NP-complete cases

Theorem 8. *The problem* Distance Edge Labeling *is* NP*-complete for every fixed* $\lambda \geq 5$*.*

The proof of the hardness result is done for every $\lambda \geq 5$. However as there is a natural difference between odd and even λ, the proof is divided according to the parity of λ to two basic general cases. The proof of the even (odd) part is contained in Subsection 3.2 and 3.3 respectively.

Furthermore, as the gadgets developed to carry the labeling does not work for small cases, we have to exclude the borderline values $\lambda = 5, 6, 7$ from the general proof. Their correctness is proven in Subsection 3.4.

Our basic reduction tool is the Monotone Not All Equal 3-SAT problem which all cases are reduced from. We say a formula φ is a *3-MCNF (monotone conjunctive normal form)* if it is a conjunction of clauses with exactly 3 logical variables without negations.

Definition 6 (Monotone Not All Equal 3-SAT **problem).**

Input:	A 3-MCNF formula φ.
Question:	Is it possible to find an assignment such that each clause has at least one literal set to true and at least one literal set to false?

The above problem is also known as MNAE-3-SAT. It is a specialized version of NAE-3-SAT, which was shown to be NP-complete by Schaefer [18] by a more general argument about CSP's. We can find MNAE-3-SAT in the list of NP-complete problems in the monograph of Garey and Johnson [19].

The reduction procedure. For a 3-MCNF formula φ and positive integer $\lambda \geq 5$ we show how to build a graph G_φ^λ. We will ensure that $\lambda'_{2,1}(G_\varphi^\lambda) \leq \lambda$ if and only if the answer to the question of MNAE-3-SAT problem is "YES". In our proofs the main focus is to prove the correspondence between a satisfying assignment to the variables of φ and the λ-labeling of the graph G_φ^λ. We call this the *correctness of a gadget.*

Definition 7 (Odd and Even sets). For any $\lambda \in \mathbb{N}$ we define two subsets of the set $\{0, \ldots, \lambda\}$. The *odd subset* $\mathbb{O} = \{l \in \mathbb{N} : l \leq \lambda, l \text{ odd}\}$ and the *even subset* $\mathbb{E} = \{l \in \mathbb{N} : l \leq \lambda, l \text{ even}\}$.

Example 1. *Take $\lambda = 10$ (even). Now according to Observation 1, the maximum possible degree of a vertex in a graph admitting a distance labeling with λ labels is* 6. *Moreover, only labels from the set $\mathbb{E}$ can appear on edges incident with such a vertex.*

3.1. Basic lemmas

We state some auxiliary lemmas that are used in our reductions.

Lemma 9 (Labeling of edges incident to a maximum degree vertex). *Let $\lambda \in \mathbb{N}$, let G be a graph with $\lambda'_{2,1}(G) \leq \lambda$ and its maximum degree vertex v.*

Then:

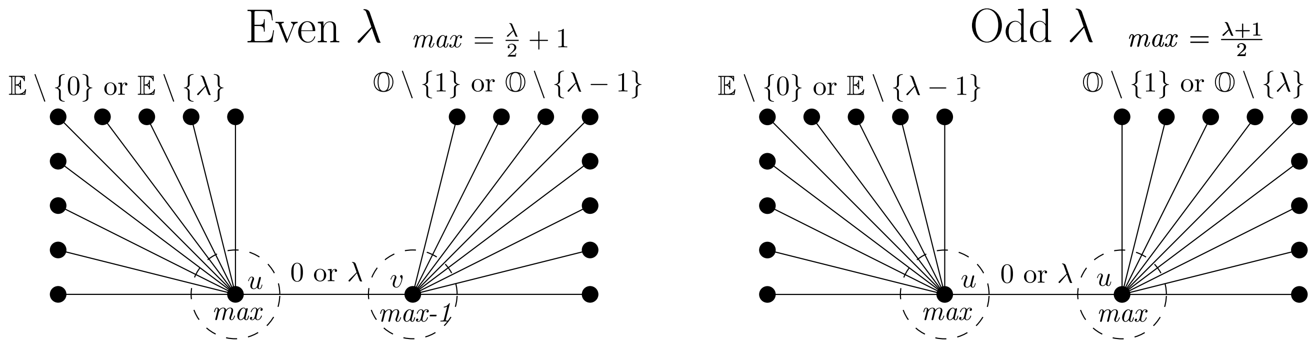
even λ: *If $deg(v) = \frac{\lambda}{2} + 1$ then vertex v has its incident edges labeled by labels from the set $\mathbb{E}$.*

odd λ: *If $deg(v) = \frac{\lambda+1}{2}$ then vertex v has its incident edges labeled by labels from the one of the sets: $\mathbb{O}$, $\mathbb{O} \setminus \{1\} \cup \{0\}$, $\mathbb{E}$ or $\mathbb{E} \setminus \{\lambda - 1\} \cup \{\lambda\}$.*

Lemma 10 (Adjacent vertices with maximum degree, even span version). *Let $\lambda \in \mathbb{N}$, λ even and let $G = (V, E)$ be a graph with $\lambda'_{2,1}(G) \leq \lambda$. Take two neighboring vertices $u, v \in V$ such that $deg(u) = \frac{\lambda}{2} + 1$, $deg(v) = \frac{\lambda}{2}$ and $\{u, v\} \in E$.*

Then there are only two possibilities:

- *The edge $\{u, v\}$ is labeled by 0, all the edges incident to u are labeled by the elements from the set $\mathbb{E} \setminus \{0\}$ and finally all the edges incident to v are labeled by the elements from the set $\mathbb{O} \setminus \{1\}$.*
- *The edge $\{u, v\}$ is labeled by λ, all the edges incident to u are labeled by the elements from the set $\mathbb{E} \setminus \{\lambda\}$ and finally all the edges incident to v are labeled by the elements from the set $\mathbb{O} \setminus \{\lambda - 1\}$.*



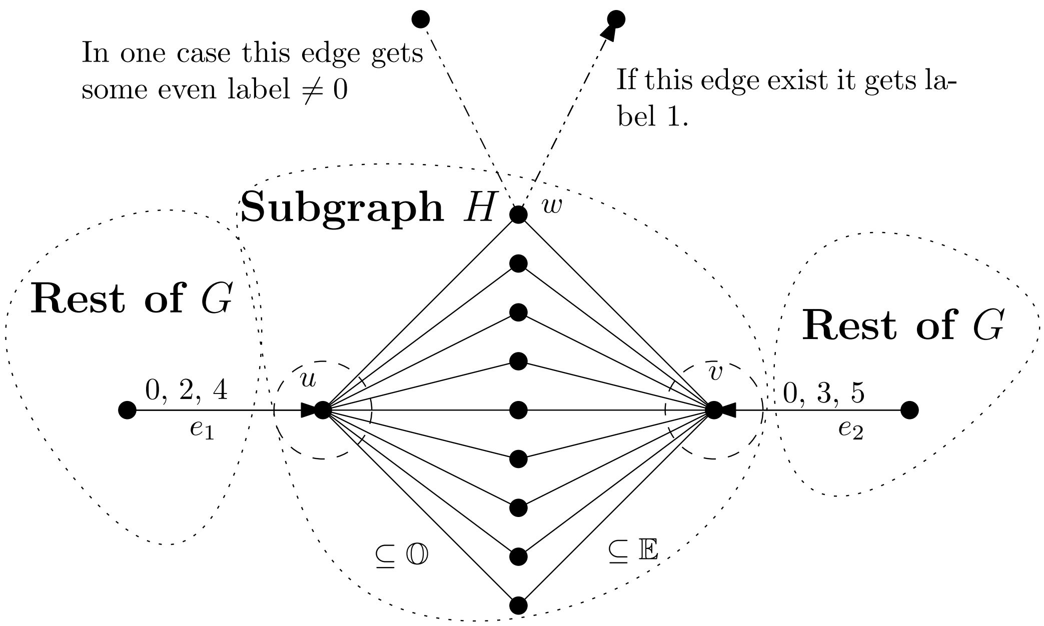
Lemma 11 (Adjacent vertices with maximum degree, odd span version). *Let $\lambda \in \mathbb{N}$, λ odd and let $G = (V, E)$ be a graph with $\lambda'_{2,1}(G) \leq \lambda$. Take two neighboring vertices $u, v \in V$ such that $deg(u) = deg(v) = \frac{\lambda+1}{2}$.*

Then there are only two possibilities:

- *The edge $\{u, v\}$ is labeled by 0, all the edges incident to u are labeled by the elements from the set $\mathbb{E} \setminus \{0\}$ and finally all the edges incident to v are labeled by the elements from the set $\mathbb{O} \setminus \{1\}$.*
- *The edge $\{u, v\}$ is labeled by λ, all the edges incident to u are labeled by the elements from the set $\mathbb{E} \setminus \{\lambda - 1\}$ and finally all the edges incident to v are labeled by the elements from the set $\mathbb{O} \setminus \{\lambda\}$.*

A proof of both lemmas above is an easy application of Lemma 9.

Notation in gadgets. We further use *max* as the number for the maximum degree in graph G with $\lambda'_{2,1}(G) \leq \lambda$. We also use directed edges in gadget graphs. An outgoing edge represents an *output*, while an ingoing edge represents an *input* to the gadget. We build all the gadgets so that the labels on output edges can take only several values.



Lemma 12 (A correct labeling of joint even and odd part). *Let $\lambda \in \mathbb{N}$, G be a graph with $\lambda'_{2,1}(G) \leq \lambda$ and H be its subgraph represented by the complete bipartite graph $K_{2,\max-1}$ such that:*

- *The only two edges connecting $G \setminus H$ to H are e_1 and e_2, where $u \in e_1$ and $v \in e_2$.*
- *The graph H contains vertices $u \neq v$, $deg_G(u) = deg_G(v) \geq 4$ and their common neighbors, call them N. Vertices from N are not adjacent, but exactly one of them w may have zero, one or two other neighbors outside H.*

- *Moreover, each edge $\{u, z\}, z \in N$ can be labeled only by odd labels ($\mathbb{O}$) and each edge $\{v, z\}, z \in N$ can be labeled only by even labels ($\mathbb{E}$) and has no other condition on them from the rest of G. (It's essential that they can be labeled by arbitrary label of appropriate set except the labels of edges e_1 and e_2.)*

There are four cases depending on labels of e_1 and e_2, on the degree of u and v and on the number of neighbors of w. If one of the following cases happen:

I. *Both e_1, e_2 have label 0, $deg_G(u) = deg_G(v) = \max$ and the vertex w has one output edge. (for λ odd)*

II. *Both e_1, e_2 have label 0, $deg_G(u) = deg_G(v) = \max -1$ and vertex w has two output edges. (for λ even)*

III. *The edge e_1 has label 2 and edge e_2 has label 3 and $deg_G(v) = deg_G(u) = \max -1$. (for λ odd)*

IV. *The edge e_1 has label 4 and edge e_2 has label 5 and $deg_G(v) = deg_G(u) = \max -1$. (for λ odd)*

Then *all edges incident to vertices of N can be labeled correctly.*

I. *The output edge incident to w has to have a label 1.*

II. *The edges incident to w has to be labeled by 1 and some $s \neq 0$ even.*

The idea of the proof is to construct an auxiliary bipartite graph. Each edge of H is labeled by some label from the correct set and it is represented by a vertex. Two vertices are connected whenever they be incident in graph H without breaking condition of a correct labeling. It can be shown that such graph is almost r-regular for some r. Moreover we can delete some edges from that graph and then it becomes r-regular. Then we can found perfect matching using Hall marriage theorem [20].

Then it is easy to show that the labeling of the output edge have to use label 1 because it is the only unused label and it cannot be placed anywhere else. The other edge incident to the vertex w has an arbitrary nonzero even label and we have exactly one left for this purpose.

Proof. We start by a construction of an auxiliary 2-regular bipartite graph H_A. We use the graph H_A to represent the incompatibility relation between the set $\mathbb{E}$ and the set $\mathbb{O}$. Set $k := \deg_G(u) = \deg_G(v)$. The *left partite* represents $k-1$ odd labels, by which we can label the H-neighborhood of the vertex u. While the *right partite* represents $k-1$ even labels, by which we can label the H-neighborhood of the vertex v. Of course by this we do not use the labels of edges e_1 and e_2.

Vertices are connected by an edge, whenever corresponding edges in graph H cannot be incident.

Note that every vertex in graph H_A has degree at most 2, as we would like H_A to be 2-regular, we have to add several edges to H_A, which we do as follows (see Figure 1):

I. In this case the left partite represents labels in the set $\mathbb{O} \setminus \{1\}$, while the right partite represents labels in the set $\mathbb{E} \setminus \{0\}$. The only vertices with degree one are: λ and 2. It is possible to add edge $\{2, \lambda\}$.

II. The left partite represents labels in the set $\mathbb{O} \setminus \{1\}$, while the right partite represents labels in the set $\mathbb{E} \setminus \{0, \lambda\}$. The only vertices with degree one are: $\lambda - 1$ and 2. Then we can add edge $\{2, \lambda - 1\}$.

III. The left partite represents labels in the set $\mathbb{O} \setminus \{1, 3\}$, while the right partite represents labels in the set $\mathbb{E} \setminus \{2, 4\}$. Vertices with degree less than two represents the following labels: 0 (degree zero), 5 and λ (both degree one). Then we can add two edges: $\{0, 5\}$ and $\{0, \lambda\}$.

IV. The left partite represents labels in the set $\mathbb{O} \setminus \{3, 5\}$, while the right partite represents labels in the set $\mathbb{E} \setminus \{4, 6\}$. Vertices with degree one represents the following labels: 0, 2, 7 and λ. Then we can add two edges: $\{2, 7\}$ and $\{0, \lambda\}$.

Now we create the graph complement $\overline{H_A}$ of the auxiliary graph H_A. We can see $\overline{H_A}$ is $(k-3)$-regular bipartite graph and then it has perfect matching by Hall's marriage theorem and so this perfect matching describes a correct labeling of the graph H.

It remains to show that in cases I. and II. it is possible to extend the labeling to the output edges incident to the vertex w. By *inner edges incident to w* we mean the edges $\{u, w\}, \{v, w\}$.

I. In this case the only label left for the output edge is label 1. That label is incompatible only with label 2 but there are at least two edges in the matching not containing label 2. So it is possible to set labels of the inner edges incident to the vertex w correctly.

II. In this case we have to label two edges incident to the vertex w. Note that we can use labels 1 and λ because all other labels are used in the close neighborhood. Similarly to the previous case we have to exclude those labelings that associate label $\lambda - 1$ or 2 with an inner edge incident to w. This is possible as there are at least 3 edges in the perfect matching. □

The main reductions proof idea. We would like to give a reader the general idea used in proofs of all cases. We will develop some gadgets to model the two parts of the input of MNAE-3-SAT. Namely the logical variables and the formula itself, which we model clause by clause. Moreover, in general-case reductions we need some middle-pieces to glue them together.

To prove that the gadget for a variable works correctly we need to check that there is no any other labeling of output edges in the *variable gadget* than the one described in the image, or its λ-inversion. Note that the only possible labels on an output edge are 0 (or 1) and λ (or $\lambda - 1$)—these will represent the logical value of the variable. For now on, we mostly omit the λ-inversion case in the proof. Every

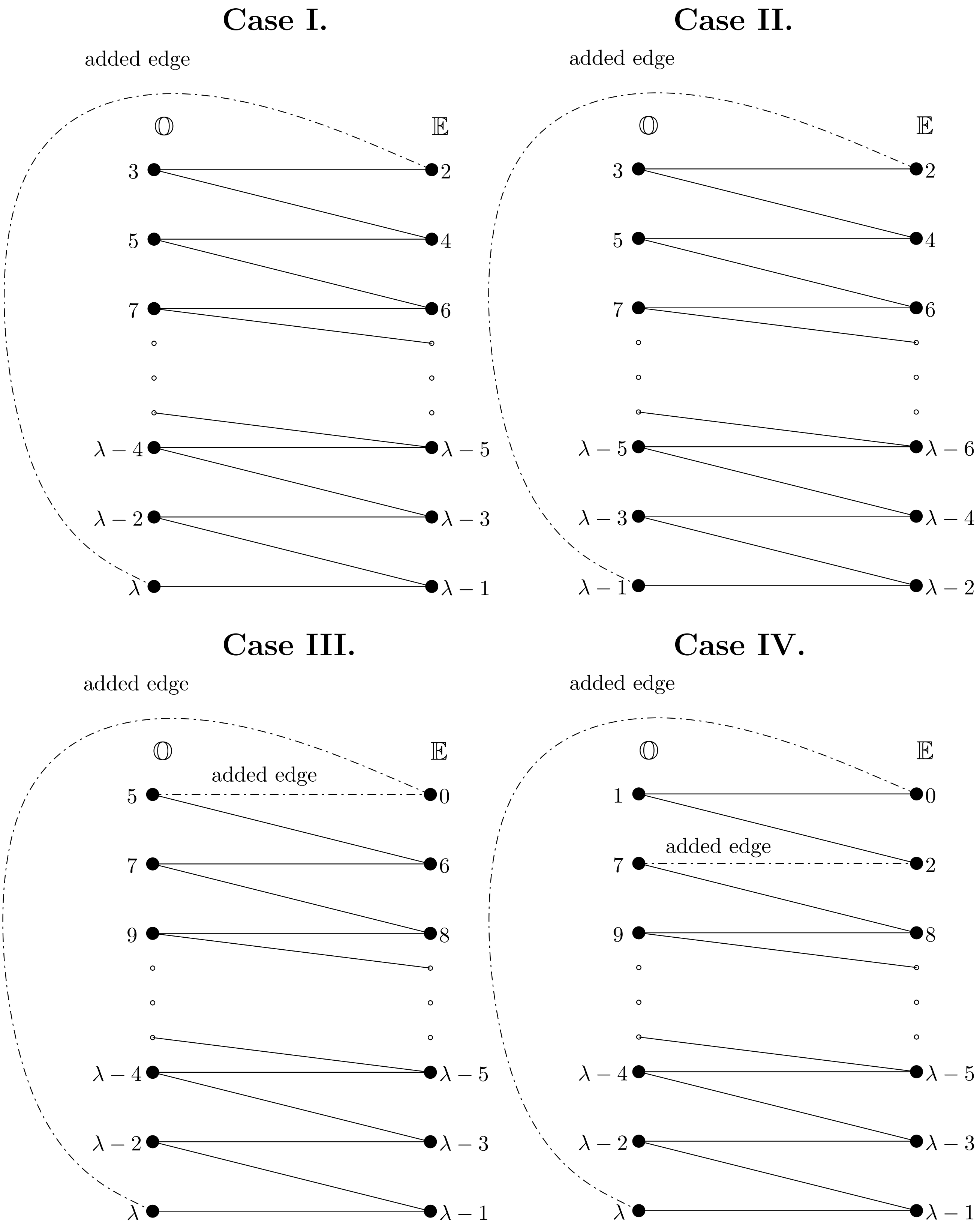


Figure 1: Four cases for bipartite graph perfect matching.

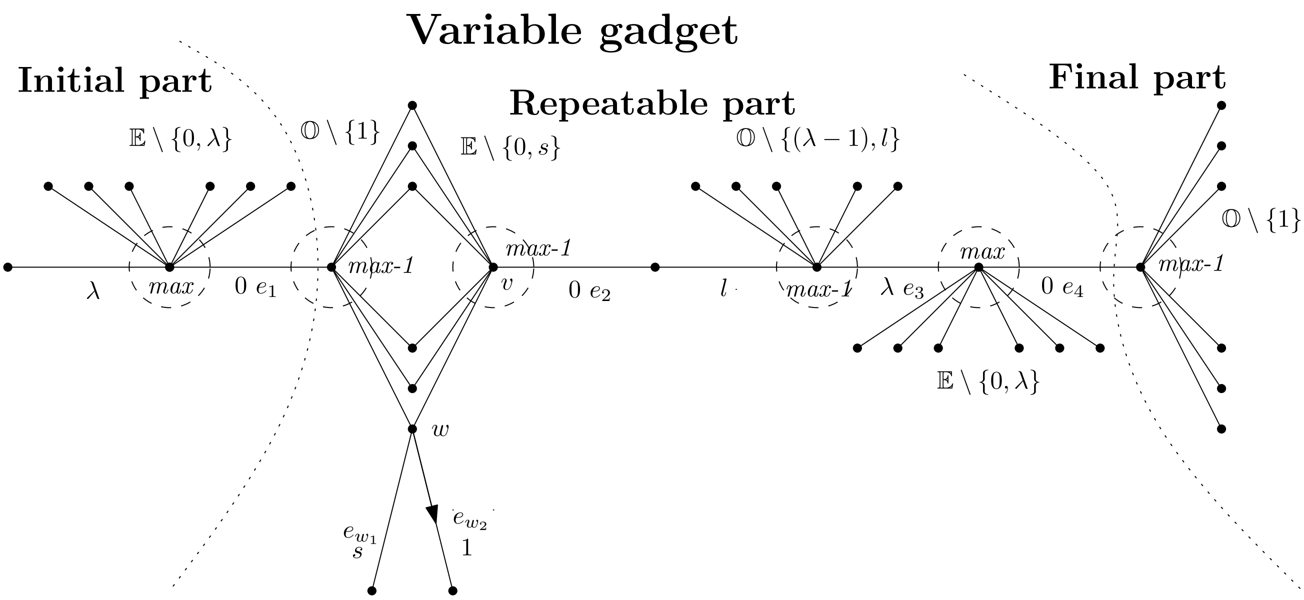
variable gadget contains a part with an output edge such that it is possible to repeat it arbitrarily—we call this part *repeatable*.

For a clause, we use a gadget for a given span with exactly 3 input edges. This *clause gadget* has to admit a labeling whenever at most two input edges represents the same logical value. On the other hand it does not admit a labeling when all input edges represents the same logical value.

3.2. Even $\lambda \geq 8$

Lemma 13. *The* Distance Edge Labeling *problem is* NP*-hard for every even* $\lambda \geq 8$.

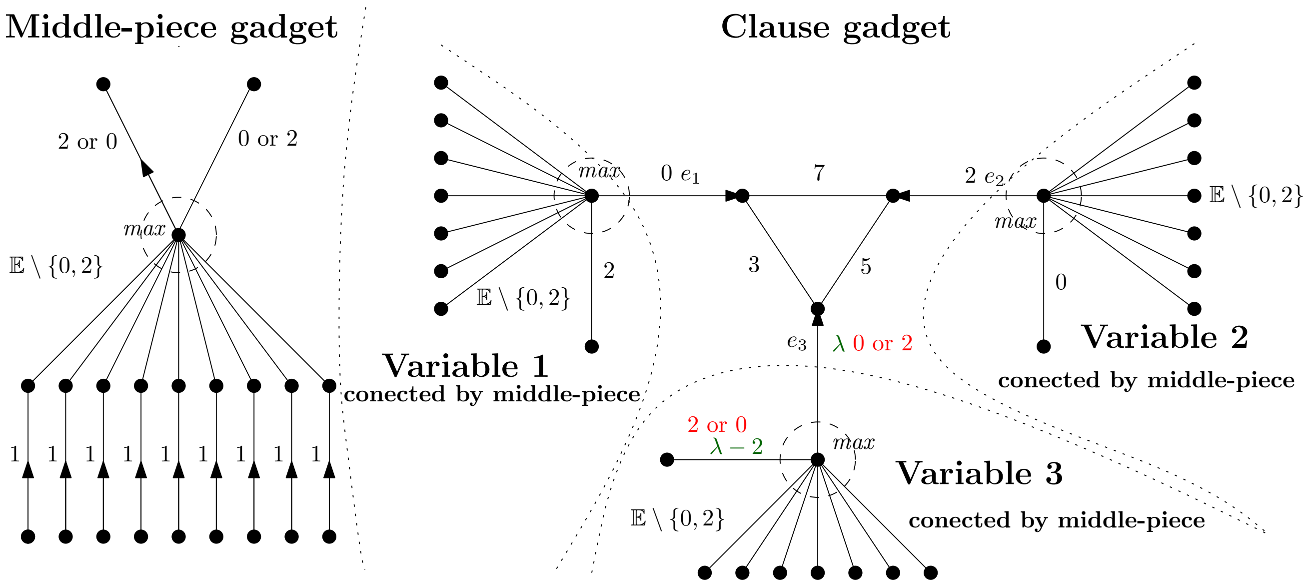
Proof. We divide the *variable gadget* into three logical parts. The *initial part* and the *final part* are only technical support for achieving the unique correct labeling. The main case analysis is done in the *repeatable part*.



By Lemma 10 we set the label of e_1 to 0. Now we have only two possible sets how to label all the edges incident to the vertex v depending on whether label 1 is used or not: $\mathbb{E}\cup\{1\}\setminus\{0,2\}$ and $\mathbb{E}\setminus\{s\in\mathbb{E}\}$. If we label edges incident to v from the set $\mathbb{E}\cup\{1\}\setminus\{0,2\}$ it is impossible to label both edges e_{w_1}, e_{w_2} incident to the vertex w, because we need to use both $0,2$ labels on them. But the label 0 is already used for the edge e_1 which is at distance two. This implies that edge e_2 must be labeled by 0. Then we can prove that presented labeling is correct by Lemma 12 part II. So, if we label these edges from the set $\mathbb{E}\setminus\{s\in\mathbb{E}\}$ then it is possible to label the output edge by s or by 1. Later the *middle-piece gadget* further restricts the output, so that the only possible label for the output edge is 1 because any $s\in\mathbb{E}$ is forbidden as an input of the *middle-piece gadget*. Edges e_3 and e_4 need to have labels 0 or λ by Lemma 10. As e_3 is at distance two to edge e_2 labeled by 0, it cannot have label 0. This allows us to repeat Repeatable part with the first edge labeled by 0.

The *middle-piece gadget* gives us only two possible outputs: 2 or 0. This is because Lemma 9. Moreover, this lemma implies that the only possible label of input edges is 1.

The output of the middle-piece gadget is plugged into the *clause gadget*. Its correctness is straightforward and it is shown in the following picture.



□

3.3. Odd $\lambda \geq 9$

Lemma 14. *The* Distance Edge Labeling *problem is* NP*-hard for every odd* $\lambda \geq 9$.

This case is more complicated than the previous one. A reason for this is in the difference between Lemma 11 and Lemma 10. In either case there are only two possible labelings but in Lemma 11 the degree of the vertex u equals to the degree of the vertex v, while this is not true in Lemma 10. So, we were able to distinguish them easily in the even case shown before.

Proof. We start with correctness of the *variable gadget*; see Figure 2.

We prove that neighboring edges of vertex v are labeled by labels from $\mathbb{O} \setminus \{1\} \cup \{0\}$. We proceed by contradiction. Suppose that these edges are labeled by $\mathbb{E}$ (according to Lemma 9 this is the only other option) then edges incident to the vertex u has labels from $\mathbb{O} \setminus \{1\} \cup \{0\}$. Then exists the edge $e = \{v, z\}$ that is labeled by some odd $l \neq \lambda$. So the neighborhood of the vertex z can be labeled either by a set $\mathbb{E} \setminus \{0, 2, l-1, l+1\} \cup \{1\}$ or by a set $\mathbb{E} \setminus \{0, l-1, l+1\}$. Neither of them is sufficiently large to label all the edges. The correctness of the other labeling is shown in the image.

Lemma 12 parts III. and IV. ensures that it is possible to repeat the repeatable part of the gadget. Note that the repeatable part consists of two identical parts, but it is possible to use only one of them as an output, because these parts are labeled λ-symmetrically.

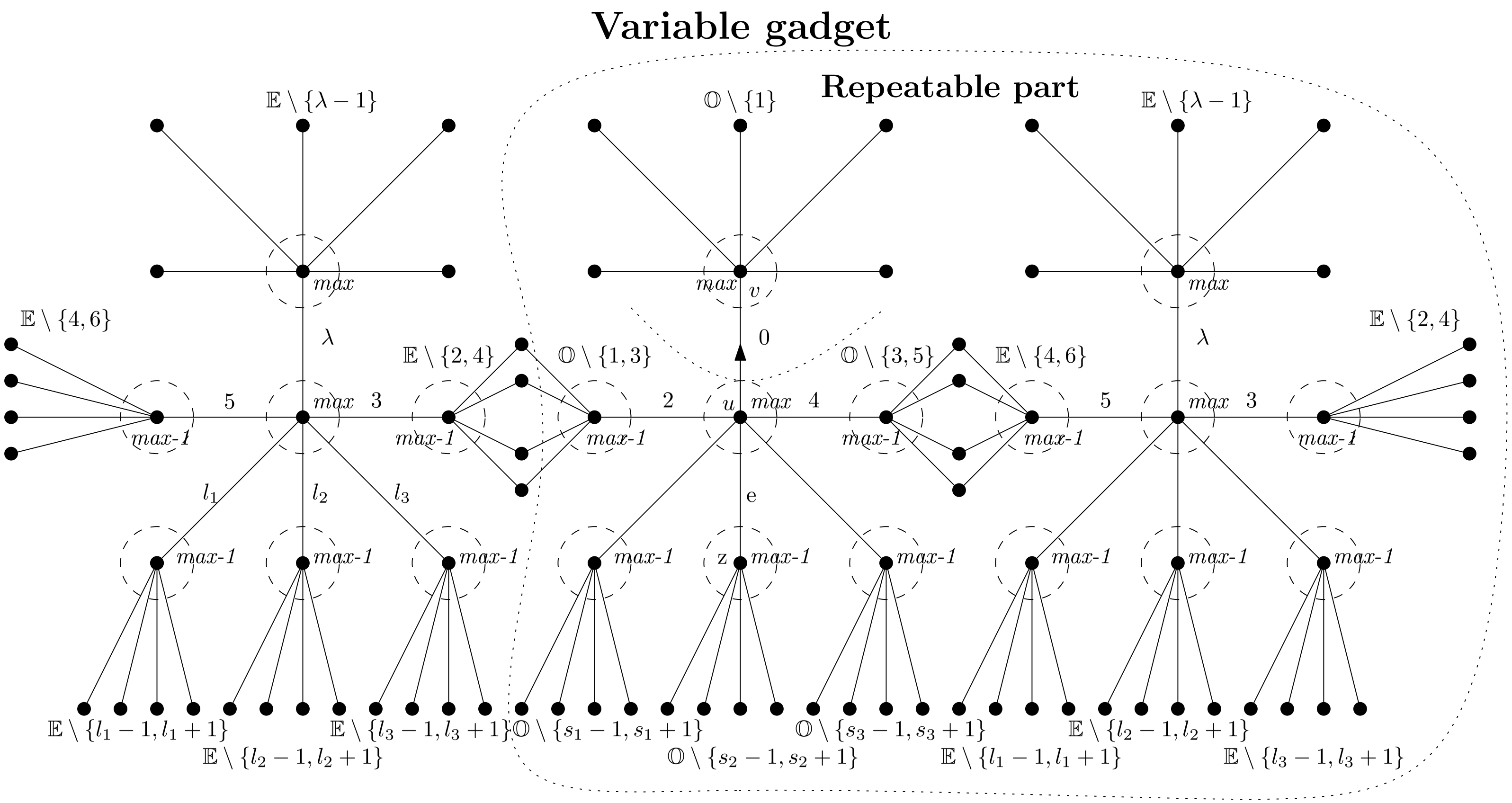
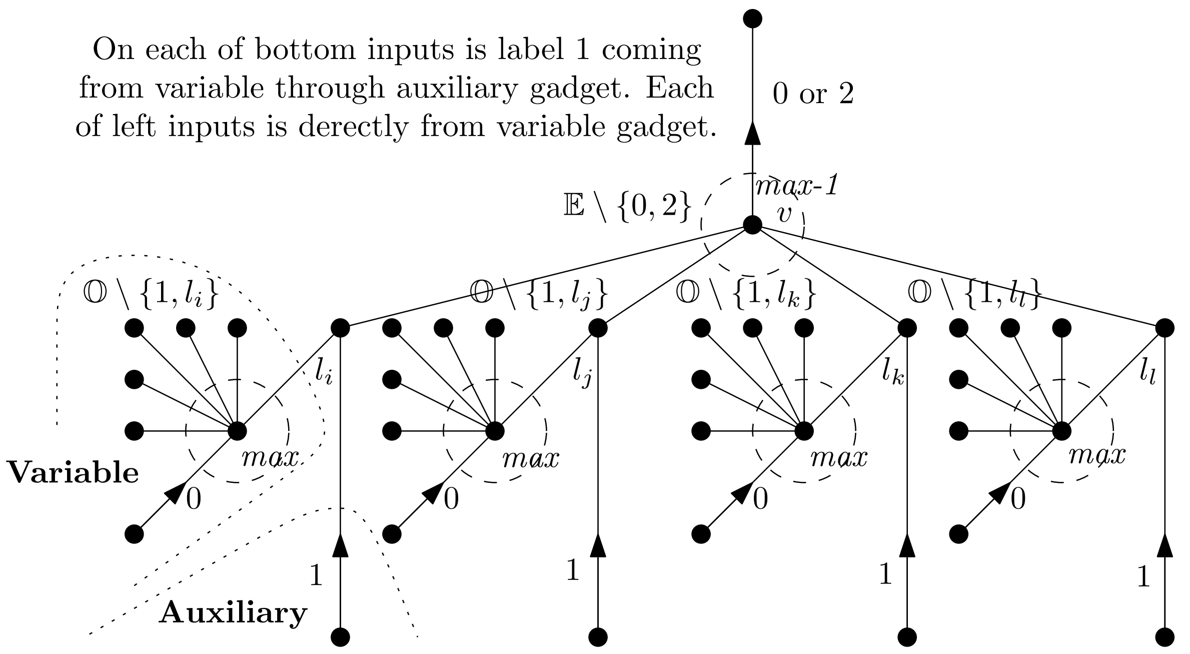
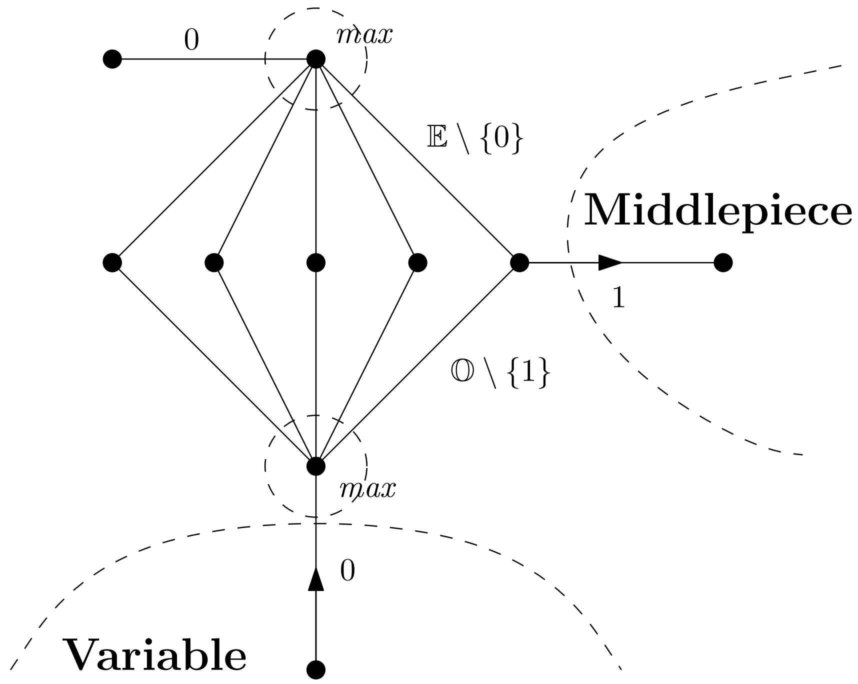


Figure 2: The variable gadget for odd general λ.



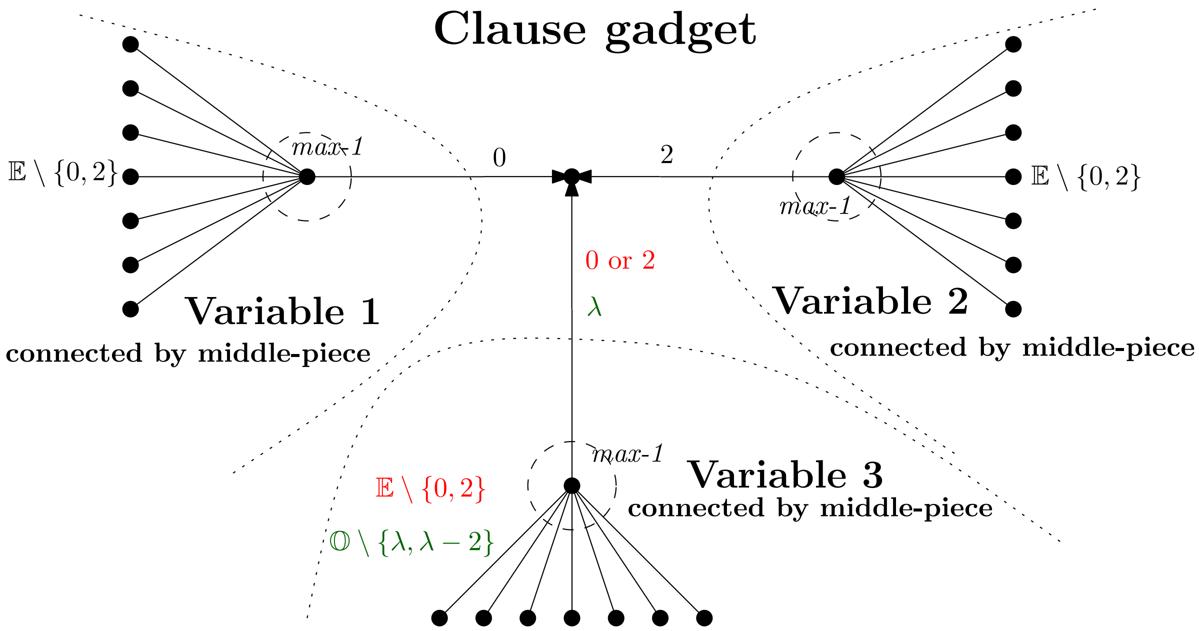


The correctness of the *auxiliary gadget* is described in Lemma 12 part I. The purpose of this gadget is to create an edge with label 1.

The *middle-piece gadget* has two kinds of inputs. Both kinds of inputs correspond to the *variable gadget*, but one of them is connected to the middle-piece through the *auxiliary gadget*.

The edges incident to the vertex v can by labeled only by labels from the set $\mathbb{E}$. This is ensured by the variable inputs, because they contains each label from the set $\mathbb{O} \setminus \{1\}$ and also by auxiliary inputs containing label 1. Note, that we can create as many such inputs as it is needed. Moreover, the label 1 forbids labels 0 and 2 anywhere besides the output edge.

Each output from the *middle-piece gadget* is plugged into the *clause gadget* in the following way. Its correctness is again straightforward and it is shown in the image which completes the proof.



□

3.4. Exceptional cases $\lambda \in \{5, 6, 7\}$

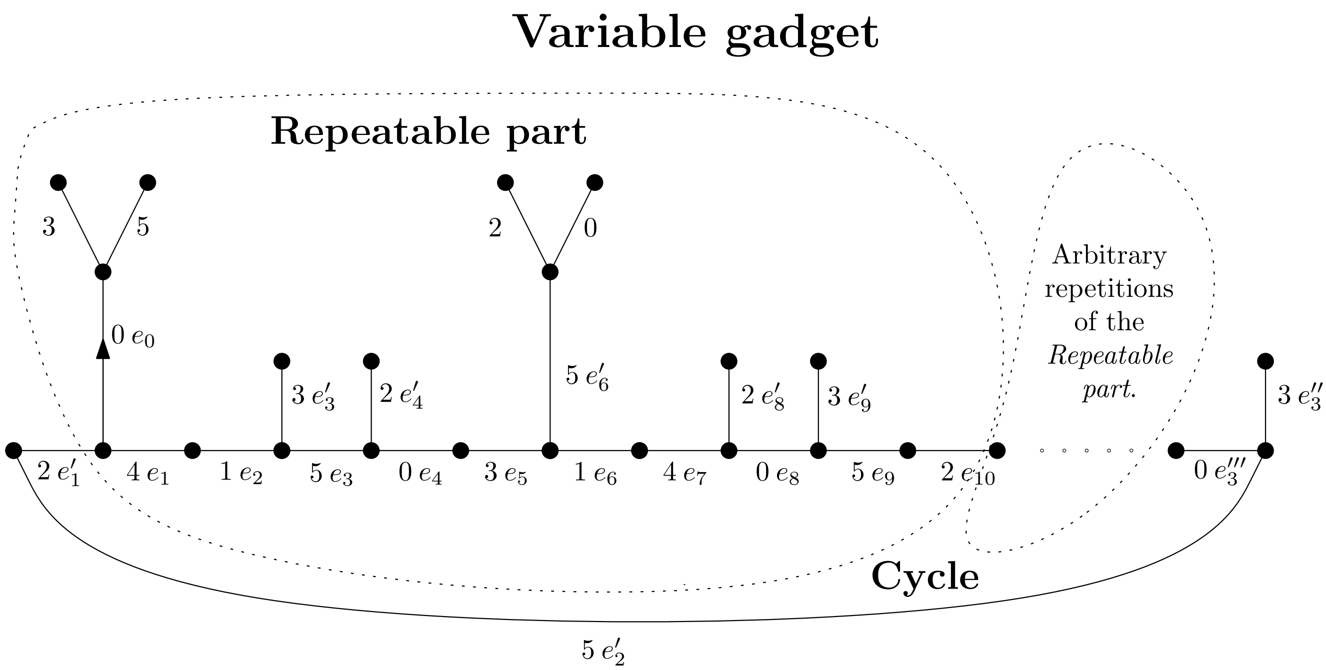
We prove exceptional cases separately for each λ. We were unable to find or change the general reductions to fit these small cases.

Case analysis are mostly in tables for now on. There is shown every possible labeling of the edges highlighted in gadget starting by an edge e_0.

NP-*hardness for* $\lambda = 5$

Lemma 15. *The* Distance Edge Labeling *problem is* NP-*hard for* $\lambda = 5$.

Proof. A variable is represented by the following *variable gadget*:



The correctness is implied by the following table. There is shown every possible labeling that can occur. Lemma 11 is heavily used to reduce the number of admissible labels on edges e_0, e'_5, e_3 and e_8.

The analysis starts assuming e_0 as an output edge get is labeled by 0. Then the labels of other edges are force according to Table 1. That gives us the first value of the variable. The λ-inversion gives us the other value representing the negation.

By repeating the repeatable part while keeping the value on the output edges we can accomplish an arbitrary number of outputs of the variable.

The case analysis shows two possibilities of a correct labeling. The first starts by placing labels $4, 2$ on edges e'_1, e_1. Then the only correct way how to repeat *the repeatable part* is to use labels from Case IV. While for closing *the cycle* we need to label the last repeatable part according to Case V.

The second place those labels on edges e'_1, e_1 in the reverse order. Again, the only correct way how to repeat again the *repeatable part* is labeling as in Case VIII. While for closing *the cycle* we can use either Case VIII. or case IX. to label the last repeatable part.

The outputs of the *variable gadgets* are plugged into the following *clause gadget*; see Figure 3 and Table 2.

As the only input to the clause gadget can be either from a set $\{3, 5\}$ or from a set $\{0, 2\}$, which represent the truth assignment of the appropriate variable. From the labels in the gadget, we can see (up to λ-symmetry) that it is impossible to label the clause if there are three inputs from the set $\{3, 5\}$ and it is possible to label the clause if there is at least one input from the other set as it is shown in the image above. □

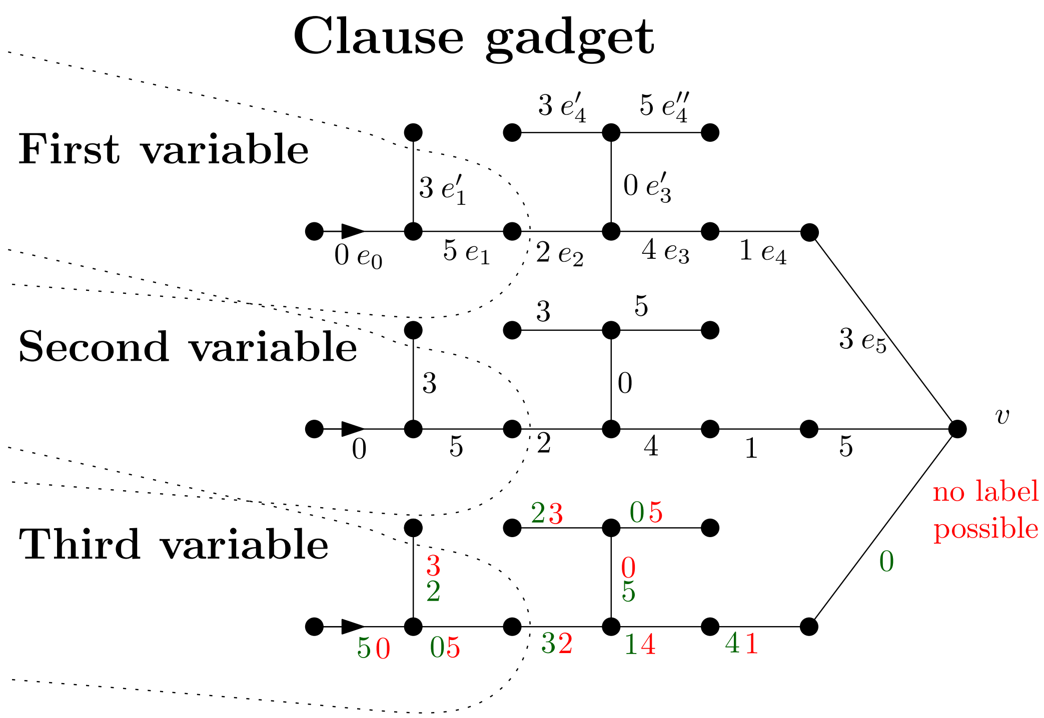


Figure 3: The clause gadget for $\lambda = 5$.

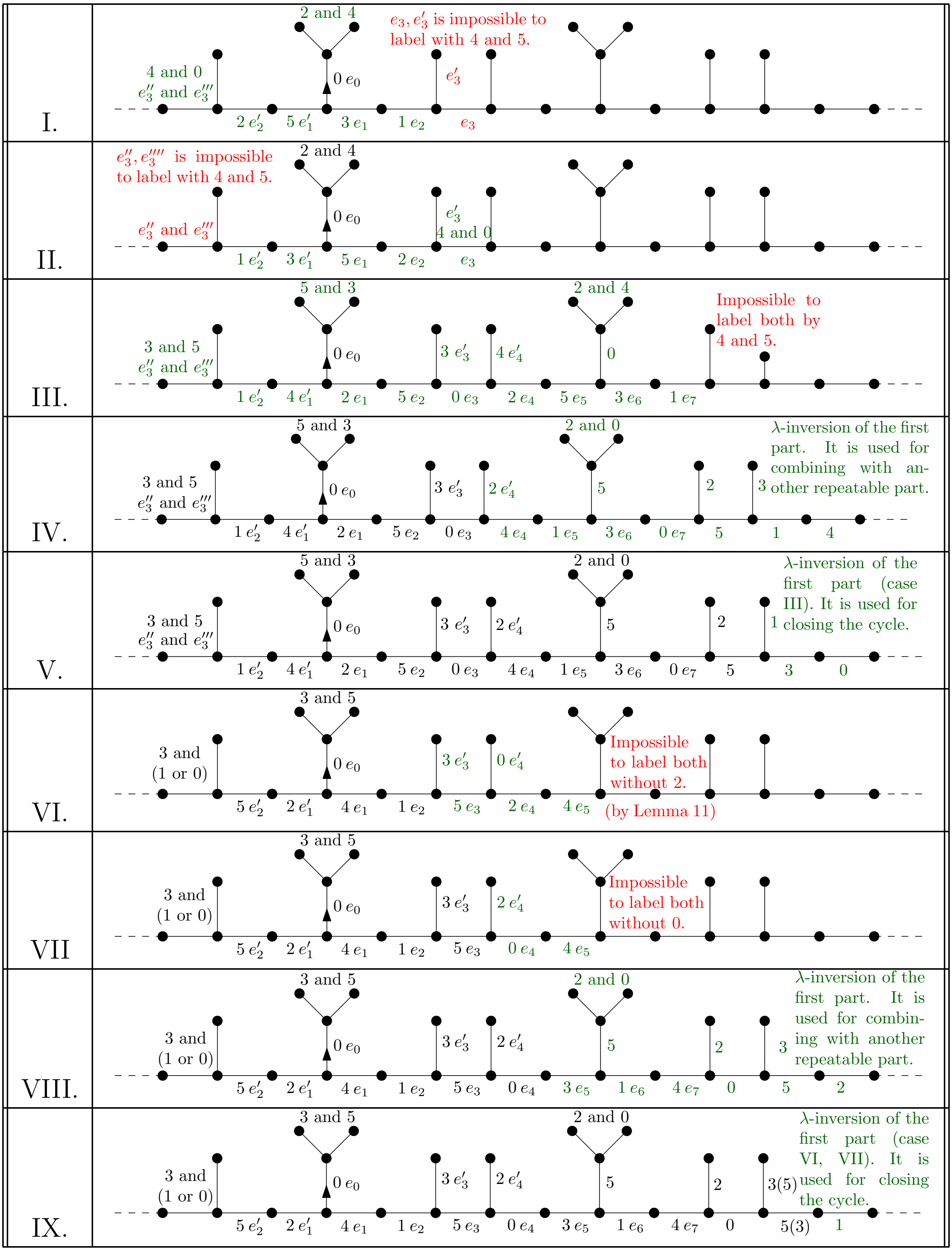


Table 1: Case analysis of for the variable gadget for $\lambda = 5$.

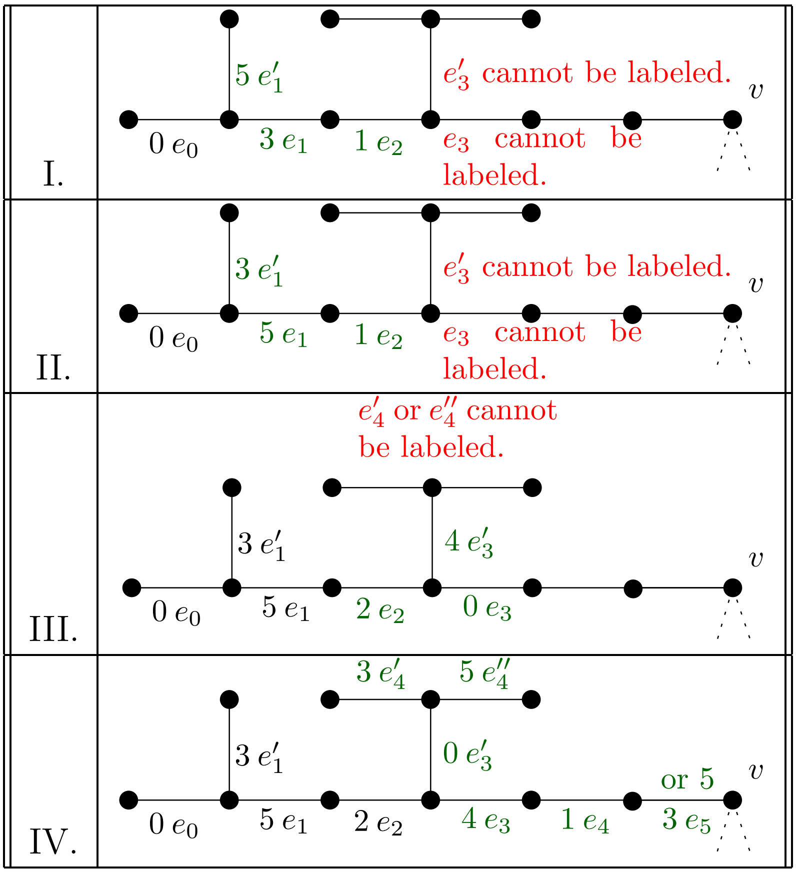
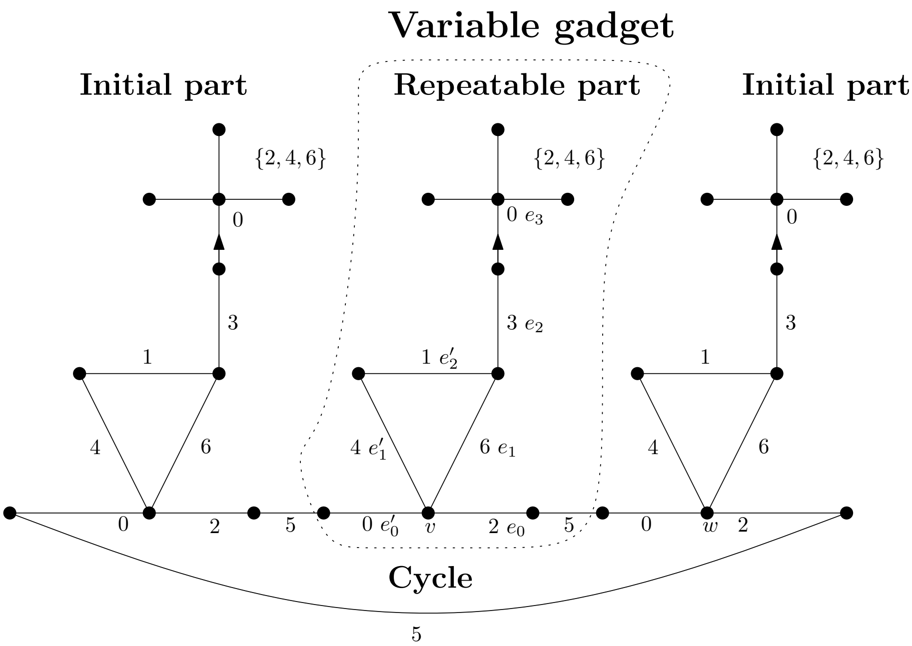


Table 2: Four cases for the clause gadget for $\lambda = 5$.

NP-*hardness for* $\lambda = 6$

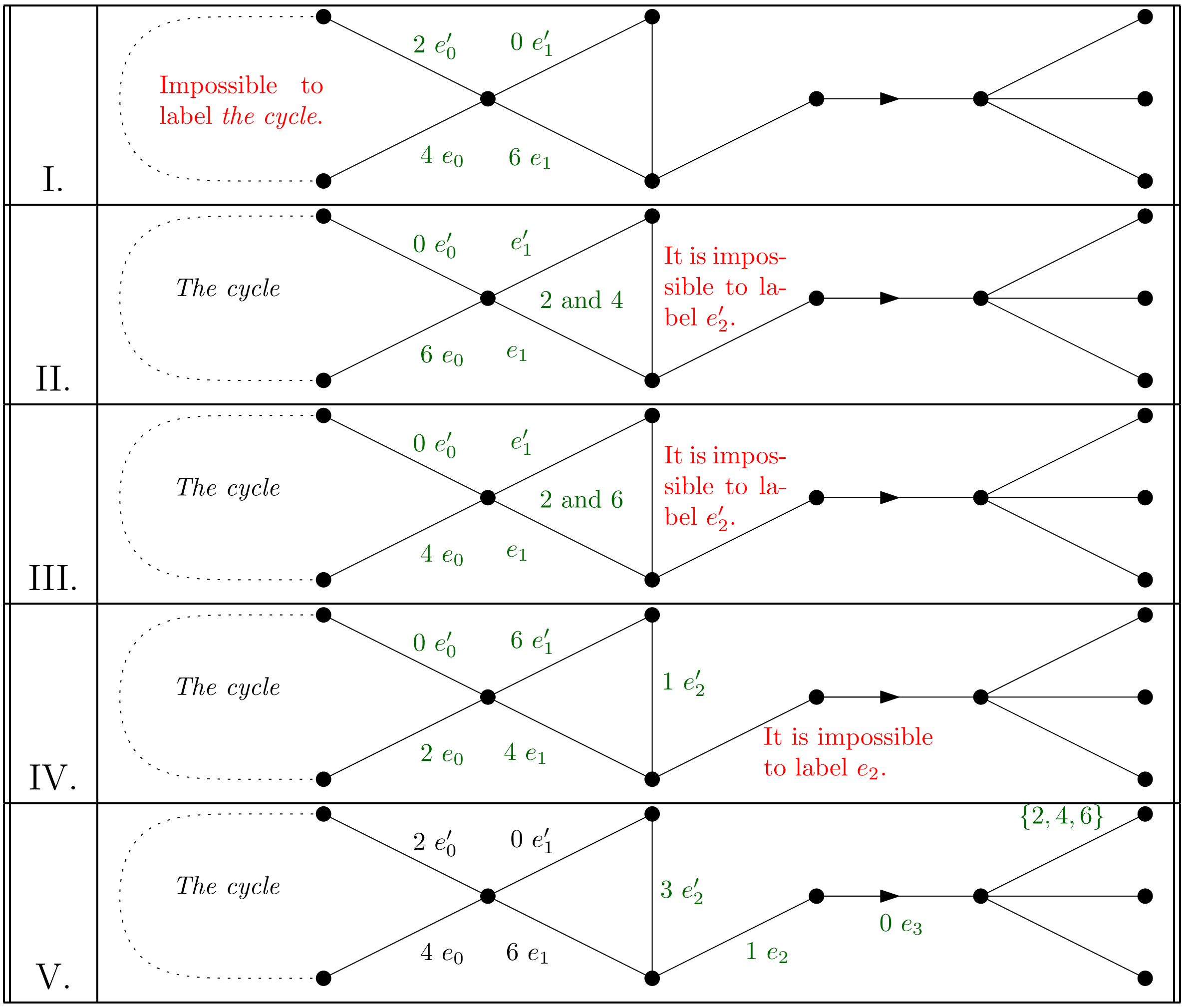
Lemma 16. *The* Distance Edge Labeling *problem is* NP-*hard for* $\lambda = 6$.

Proof. A variable is represented by the following *variable gadget*:



And the case analysis is in the following table.

It starts using Lemma 9 on the vertex v. We do not need to analyse cases that are symmetric along *the cycle*. So, without loss of generality e_0' is labeled by a smaller label than e_0 and as usually we omit cases that are λ-inversions.



We should notice how the only possible labeling on *the cycle* is forced. According to the following table the only admissible labelings around main vertices (e.g. v, w) are consecutive pairs of labels $(0,2), (2,0), (2,4), (4,2)$ and λ-inversions $(6,4), (4,6)$. Notice that any edge connecting main vertices can be labeled only by an odd label $(1,3,5)$. Observe that there cannot be close edges of two consecutive main vertices labeled by the same label. Then we do a small case analyse according to the label used on a connecting edge:

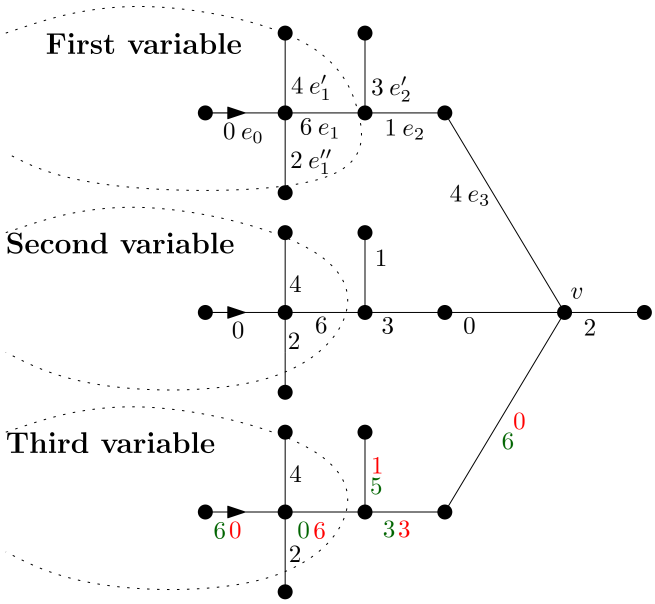
1 used There has to be $(6,4)$ (respectively $(4,6)$) the same on every main vertex from both side and that forms the correct λ-inverse labeling. If not then $(2,4)$ was used somewhere. But after that there is the only choice $(6,4)$ and then we cannot use anything else and so we cannot reach $(2,4)$ again.

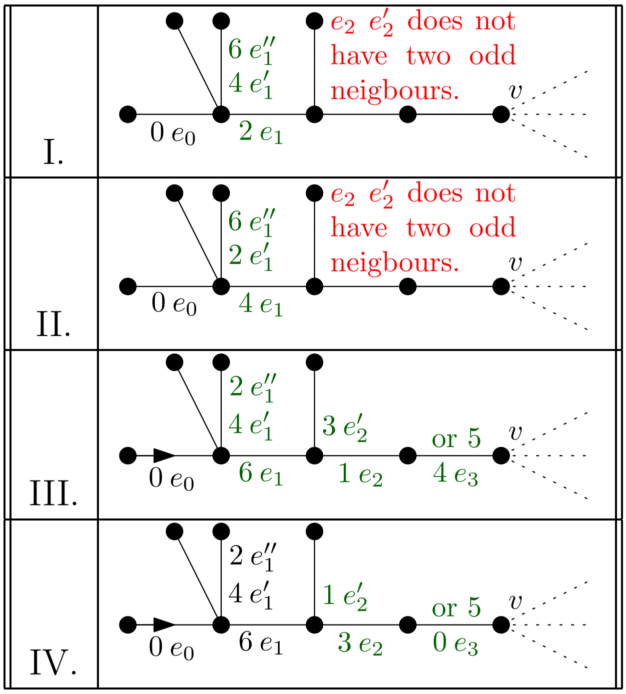
3 used There has to be $(4,6)$ from one side and $(0,2)$ from the other. But next to 2 there can only be 5 and then there has to be again $(0,2)$ so we can never ever reach $(4,6)$.

5 used There has to be $(0,2)$ (respectively $(2,0)$) the same from both side and that forms the correct labeling. If not then $(4,2)$ was used somewhere. But after that there is the only choice $(6,4)$ and then we cannot use anything else and so we cannot reach $(4,2)$ again.

And the clause is represented by the *clause gadget*:

Clause gadget





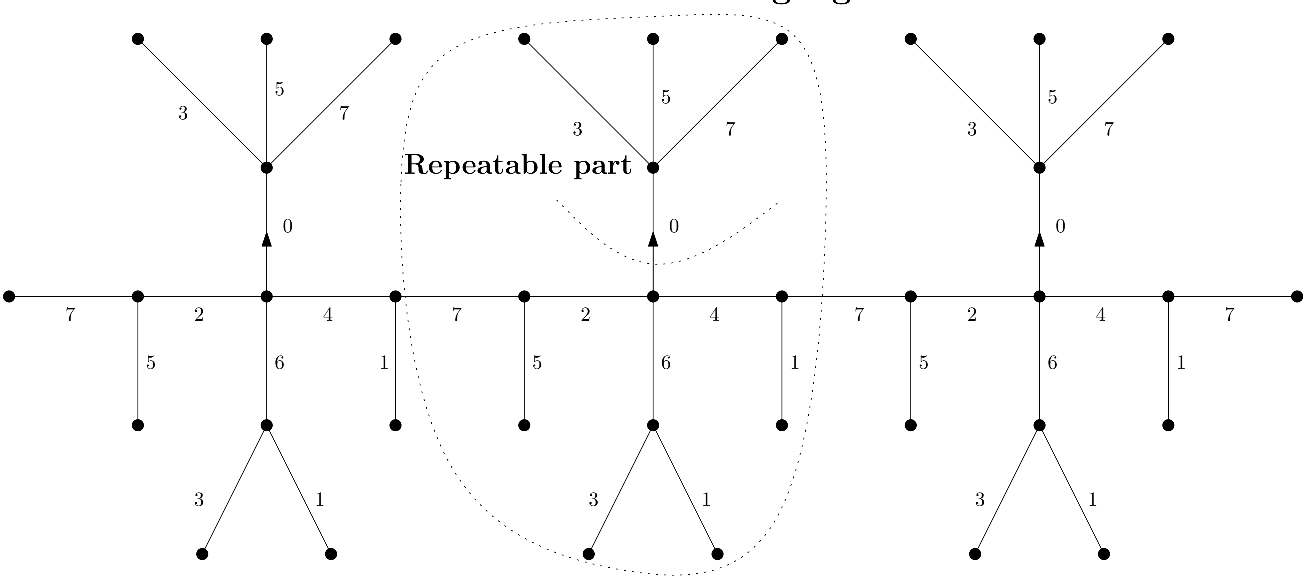
The correctness is implied by a similar case analysis as before given in the table. ∎

NP-*hardness variable gadget for* $\lambda = 7$

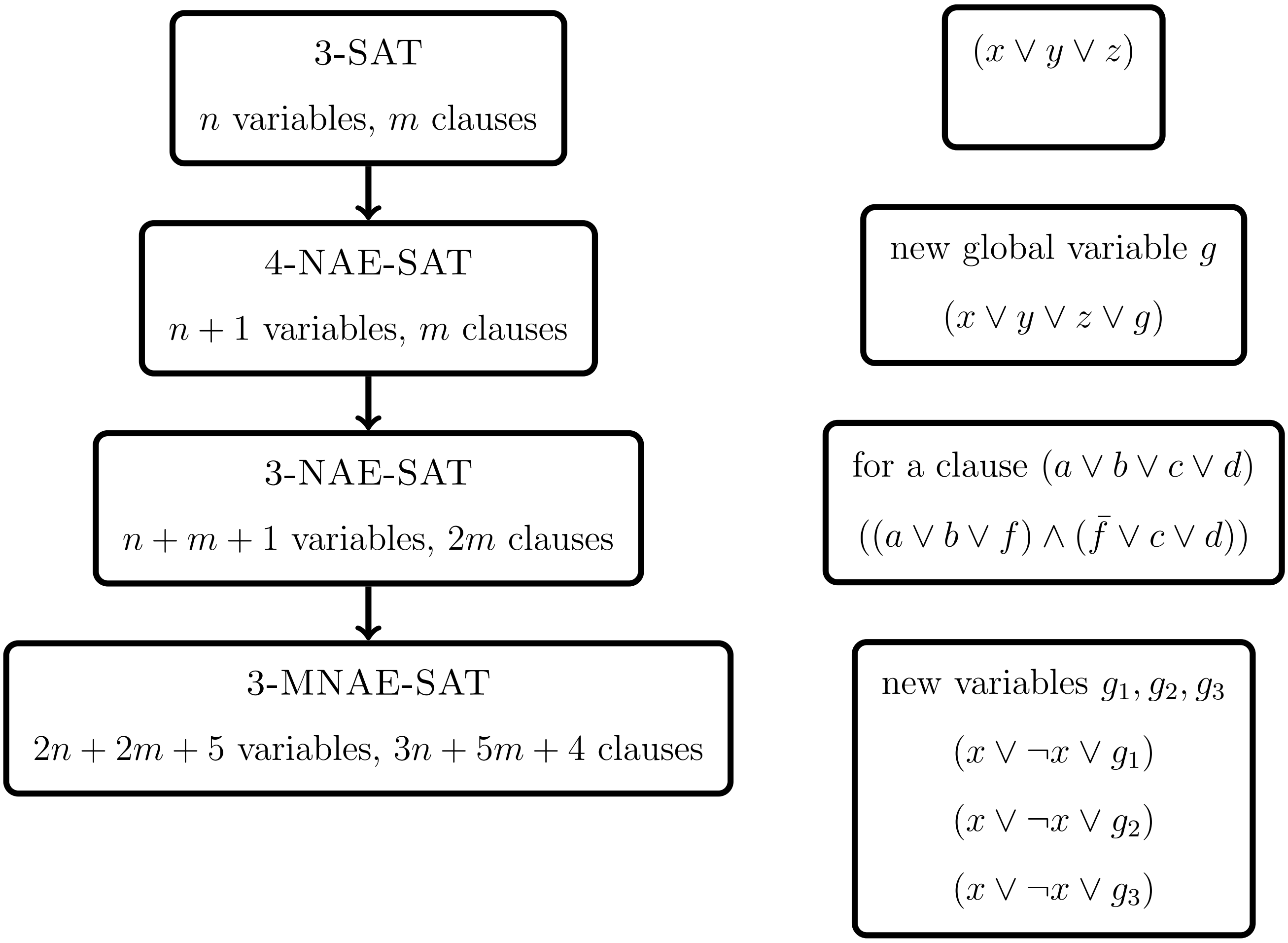
Lemma 17. *The* Distance Edge Labeling *problem is* NP-*hard for* $\lambda = 7$.

Proof. For the case $\lambda = 7$, we show only the variable gadget because in this case it is possible to reuse all the other gadgets from the general case where $\lambda \geq 9$.

Variable gadget



The correctness of the repeatable part is done by the same argument as it is done in the proof of the general case for $\lambda \geq 9$. Then it is straightforward to show that the only possible labeling of connection of repeatable parts is the one shown in the image above. ∎



4. Algorithmic hardness based on ETH

In this section we prove Corollary 2.

The proof is divided into two relatively independent parts. In the first part we show a version of the ETH for the MNAE-3-SAT problem using well-known reduction techniques. We present this part just for an analysis of the ETH-based lower bounds. The second part is a conclusion of the NP reductions we have shown in the previous section.

4.1. An ETH for MNAE-3-SAT

For ease of the presentation follow the overview schema of the size preserving reduction.

Proof. We begin with a classical clause version for 3-SAT as it was mentioned in Section 1.2.1. The input instance contains n variables and m clauses. We add a new global variable g and we add it into every clause. So we have an instance of 4-NAE-SAT. If it is satisfiable then we can set g to false and thus we satisfied 3-SAT version. If it is not then there is a clause that cannot be not-all-equal-satisfied. Thanks to the symmetry of not-all-equalness we can set g to false and thus we know that the clause is not satisfiable because all the other literals must be false as well. We extend the number of variables by 1 and preserve the number of clauses.

From 4-NAE-SAT we proceed to 3-NAE-SAT just by adding an auxiliary variable f for each clause which we evenly split into two. One of them containing f and the other the negation $\bar{f}$. Again, if the modification is satisfied then the original form is easily satisfied as well because either f or $\bar{f}$ cannot be true. The other implication is of the same principle. We thus doubled the number of clauses and extended the number of variables by m.

From 3-NAE-SAT to 3-MNAE-SAT we add new variables representing negations of the former variables. Finally, we add clauses containing the positive and the negative form of the same variable and one of new global variables g_1, g_2, g_3. We finish the reduction by adding a clause $(g_1 \vee g_2 \vee g_3)$. This reduction again equivalently preserves satisfiability since every former variable have to get opposite value than its negation.

In total we extended the number of clauses to $3n + 5m + 4$ and the number of variables to $2n + 2m + 5$. This is still sufficient because the reduction is linear in the parameter and that satisfies the definition of the SERF-T reduction 5. The previous holds because the number of variables is smaller than the number of clauses and so it depends only on the number of clauses and that is the reason why we have begun with the clause 3-SAT variant of the ETH. □

4.2. An ETH for distance edge-labeling

PROOF. It is straightforward to check that for a fixed $\lambda \geq 5$ the size of gadgets is linear in the number of clauses and that according to the definition of the SERF-T reduction 5 is sufficient together with the NP-hardness reduction we have shown in Section 3. This finishes the proof of Corollary 2 □

5. Conclusions

It would be interesting to know whether it is possible to find simpler reduction as well as to find the characterization for generalised version of DISTANCE EDGE LABELING problem, where the labels of neighbouring edges must differ by at least p and the labels of edges in distance two must differ by at least q.

6. Acknowledgements

We would like to thank Juho Lauri for an idea to use an ETH as well as our supervisor Jiří Fiala for fruitful discussions, proofreading and helpful comments.

7. References

References

[1] D. Knop, T. Masařík, Computational complexity of distance edge labeling, in: Combinatorial Algorithms - 26th International Workshop, IWOCA 2015, Verona, Italy, October 5–7, 2015, Revised Selected Papers.

[2] P. Formanowicz, K. Tanaś, A survey of graph coloring - its types, methods and applications, Foundations of Computing and Decision Sciences 37 (3) (2012) 223 – 238. doi:10.2478/v10209-011-0012-y.

[3] T. Calamoneri, The L(h, k)–labelling problem: An updated survey and annotated bibliography, The Computer Journal 54(8) (2011) 1344–1371.

[4] T. Calamoneri, The L(h,k)–labelling problem (online updated survey) (2013).
URL http://wwwusers.di.uniroma1.it/~calamo/survey.html

[5] J. R. Griggs, R. K. Yeh, Labelling graphs with a condition at distance 2, SIAM J. Discrete Math. 5 (4) (1992) 586–595.

[6] R. K. Yeh, Labeling graphs with a condition of distance two, Ph.D. thesis, University of South Carolina (1990).

[7] W. K. Hale, Frequency assignment: theory and applications 68 (1980) 1497–1514.

[8] J. Fiala, T. Kloks, J. Kratochvíl, Fixed-parameter complexity of λ-labelings, Discrete Applied Mathematics 113 (1) (2001) 59 – 72, selected Papers: 12th Workshop on Graph-Theoretic Concepts in Computer Science. doi:http://dx.doi.org/10.1016/S0166-218X(00)00387-5.
URL http://www.sciencedirect.com/science/article/pii/S0166218X00003875

[9] V. G. Vizing, On an estimate of the chromatic class of a p-graph, Diskret. Analiz No. 3 (1964) 25–30.

[10] F. Havet, B. Reed, J.-S. Sereni, Griggs and yeh's conjecture and $l(p,1)$-labellings, SIAM Journal on Discrete Mathematics 26 (1) (2012) 145–168.
URL http://dx.doi.org/10.1137/090763998

[11] J. Flum, M. Grohe, Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[12] R. Diestel, Graph Theory, Electronic library of mathematics, Springer, 2006.
URL http://books.google.cz/books?id=aR2TMYQr2CMC

[13] R. Impagliazzo, R. Paturi, F. Zane, Which problems have strongly exponential complexity?, Journal Comput. Syst. Sci. 63 (4) (2001) 512–530.

[14] D. Lokshtanov, D. Marx, S. Saurabh, Lower bounds based on the exponential time hypothesis, Bulletin of the EATCS 105 (2011) 41–72.

[15] C. Calabro, R. Impagliazzo, R. Paturi, A duality between clause width and clause density for SAT, in: 21st Annual IEEE Conference on Computational Complexity (CCC) 2006, 16-20 July 2006, Prague, Czech Republic, 2006, pp. 252–260. doi:10.1109/CCC.2006.6.

[16] C. Calabro, R. Impagliazzo, R. Paturi, The complexity of satisfiability of small depth circuits, in: Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers, 2009, pp. 75–85. doi:10.1007/978-3-642-11269-0_6.
URL http://dx.doi.org/10.1007/978-3-642-11269-0_6

[17] R. Santhanam, S. Srinivasan, On the limits of sparsification, in: Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I, 2012, pp. 774–785.

[18] T. J. Schaefer, The complexity of satisfiability problems, in: Proceedings of the Tenth Annual ACM Symposium on Theory of Computing, STOC '78, ACM, New York, NY, USA, 1978, pp. 216–226. doi:10.1145/800133.804350.
URL http://doi.acm.org/10.1145/800133.804350

[19] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., New York, NY, USA, 1979.

[20] P. Hall, On representatives of subsets, Journal of the London Mathematical Society s1-10 (1) (1935) 26–30. doi:10.1112/jlms/s1-10.37.26.