Reasoning about connectivity without paths*

Alberto Casagrande acasagrande@units.it Eugenio Omodeo eomodeo@units.it

Dep. of Mathematics and Geosciences University of Trieste Trieste, Italy

Abstract. In graph theory connectivity is stated, prevailingly, in terms of paths. While exploiting a proof assistant to check formal reasoning about graphs, we chose to work with an alternative characterization of connectivity: for, within the framework of the underlying set theory, it requires virtually no preparatory notions.

We say that a graphs devoid of isolated vertices is *connected* if no subset of its set of edges, other than the empty set and the set of all edges, is vertex disjoint from its complementary set. Before we can work with this notion smoothly, we must prove that every connected graph has a *non-cut* vertex, i.e., a vertex whose removal does not disrupt connectivity.

This paper presents such a proof in accurate formal terms and copes with *hypergraphs* to achieve greater generality.

1 Introduction

Connectivity plays a crucial role not only in graph theory, but also in topology. The number of connected components of a graph is a topological invariant, it corresponds to the multiplicity of the eigenvalue 0 in the Laplacian matrix that represents the graph, and, in the recent years, it has been related to the number of claw-free subgraphs of the graph itself [1]. Because of the ubiquity of this notion, it deserves an autonomous and insightful treatment in a large scale formalization effort, such as the one envisioned in [8] or in [6].

Non-cut vertices are vertices whose removal preserves the graph connectivity. The notion of connectivity is traditionally given in term of *paths*, and in such terms one proves that any graph contains non-cut vertices. These vertices are largely used in inductive proofs over connected graphs: the pattern is to apply the inductive hypothesis to a graph deprived of a non-cut vertex and, then, to prove that the investigated property is retained when the vertex is reinstated.

While formally defining the notion of path is not really a problem, from a foundational point of view, it appears to be an *out-of-focus* effort in this context: it would in fact bring into play notions (e.g., natural numbers) which are barely related to the theme of discourse.

^{*} This work has been partially supported by Istituto Nazionale di Alta Matematica (INdAM).

This paper exploits a path-free notion of connectivity to formally prove that *connected undirected hypergraphs* are always endowed with non-cut vertices.

Our immediate motivation for undertaking this study on connectivity is a formalization task that has been successfully carried out recently with the assistance of the proof checker Referee/Ætnanova [6]: as reported in [5], taking advantage of the Milanič and Tomescu representation theorem for *connected claw-free graphs* [4], we could achieve with relative ease the proof that any such graph owns a near-perfect matching and has a Hamiltonian cycle in its square. We took it for granted that every connected graph has a spanning tree. This left us with a proof obligation, and since the existence of a spanning tree plainly reduces to the proof that every connected graph has a non-cut vertex, we are now beginning to fill the gap, with the formalization task on which we will report below.

This paper is organized as follows: Section 2 introduces the notation and all the needed notions. Section 3 formalizes the result aimed at and splits the proof of it into multiple steps that are detailed in Sections 3.1, 3.2, and 3.3 (some basic properties and technical lemmas are proved in the appendix). Finally, Section 4 draws our conclusions and suggests future work.

2 Preliminaries

We work with in the Zermelo-Fraenkel set theory (ZF) and all the notions treated in this paper are defined through it. Besides the standard Boolean propositional functions (i.e., \land , \neg), the used formal language provides intersection (\cap), union (\cup), and difference (\backslash) over sets as well as both the membership (\in) and inclusion (\supseteq) relations. From a formal point of view, the Boolean functions \lor and \rightarrow and the relations over sets \notin , \supseteq , \subseteq , =, and \neq are shortcuts for non-atomic formulæ whose semantics is the standard one. While the notion of *cardinality of a set* is not formally included in the adopted language, it is worthwhile to introduce the relation $|\cdot|_{\geq n}$ that associates each finite set with the number of elements belonging to it. This relation is not really necessary, but it enables more natural definitions for the subsequent notions. The relation $|\cdot|_{\geq n}$ is defined as:

$$|S|_{\geq n} := \begin{cases} \top & \text{if } n = 0\\ \exists v \in S \ |S \setminus \{v\}|_{\geq n-1} & \text{otherwise} \end{cases}$$

It is easy to see that, for any natural number $n \in \mathbb{N}$ and any set S, $|S|_{\geq n}$ holds if and only if $|S| \geq n$.

We characterize finitude as proposed by Tarski [7]: a set *S* is finite if and only if every not empty class of subsets of *S* contains an element which is minimal with respect to \subseteq . This clue is captured by the following definition

Finite
$$(S) := \forall P \in \wp(\wp(S)) \setminus \{\emptyset\} \exists M \wp(M) \cap P = \{M\}$$
.

Tarski himself proved that if *S* is finite then every not empty class of subsets of *S* contains also a maximal element [7].

The basic notions of (hyper)graph, edge, and node are defined as follows.

Definiton 1 An edge is a finite set endowed with at least two elements. A hypergraph *G* is a finite set of edges, i.e., **Graph** (*G*) :- **Finite** (*G*) $\land \forall e \in G$ ($|e|_{\geq 2} \land$ **Finite** (e)). The elements of the edges of *G* are called nodes, or vertices, of *G*.

By standard terminology, the word graph refers to hypergraphs whose edges have cardinality 2. However, we take the freedom to abbreviate "hypergraph" into "graph" because this work deals exclusively with the more general notion.

In accordance with the literature (e.g., see [2, 3]), our definition does not allow graphs to have self-loops, namely singleton edges, and it explicitly requires that each of the edges contains at least 2 distinct elements.

Any subset of a graph is a graph.

Lemma 1 $P \subseteq G \land Graph(G) \rightarrow Graph(P)$

Proof. The claim follows directly from the definition of **Graph** (\cdot) .

Let *Nodes* (*G*), *Cov* (*G*, *P*), and *Contains* (*G*, *v*) represent the set of nodes of *G*, the set of edges in *G* that share nodes with any edge in *P*, and the set of edges in *G* that contain the node *v*. More formally,

$$Nodes(G) \stackrel{\text{def}}{=} \bigcup_{e \in G} e \qquad Cov(G, P) \stackrel{\text{def}}{=} \{e \in G \mid e \cap Nodes(P) \neq \emptyset\}$$

$$Contains(G, v) = \{e \in G \mid v \in e\}$$

If *Contains* (G, v) is a singleton, then v is said to be a *boundary vertex*.



Fig. 1: In above figures, ovals represent the elements of *G*.

If, for every nonnull set $P \subsetneq G$, P shares some nodes with the graph $G \setminus P$, then G is said to be *connected* (in short, **Conn** (G)).

Conn (*G*) :- **Graph** (*G*)
$$\land \forall P \ (\emptyset \subsetneq P \subsetneq G \rightarrow Nodes (P) \cap Nodes (G \setminus P) \neq \emptyset)$$

If the graph *G* is connected and $G \setminus \{e\}$ is not connected, then *e* is said to be a *cutting edge*.



Fig. 2: Let *G* and *P* be $\{\{a, b\}, \{a, f\}, \{c, d, e\}, \{d, e\}\}$ and $\{\{a, b\}, \{a, f\}\}$, respectively. The graph *G* is not connected as *Nodes* (*P*) \cap *Nodes* (*G* \ *P*) is empty and *P* \subseteq *G*.

Let us notice that it is not always the case that by removing a node v from all edges of a graph G we get a graph. As a matter of fact, some of the edges of G that contain v may have cardinality 2. If we remove v from such edges, we obtain sets whose cardinality is 1 and, by definition of edge, these are not edges. For instance, if G contains an edge $\{a, b\}$ and v is a, then $\{a, b\} \setminus \{a\}$ is $\{b\}$ which is not an edge because has cardinality 1.

Filter (*G*, *v*) is the set obtained by first removing *v* from all edges of *G* and then filtering out all the resulting sets whose cardinality is less than 2. Since all the elements of *Filter* (*G*, *v*) have cardinality 2 at least, *Filter* (*G*, *v*) is a graph by definition of graph.

Filter
$$(G, v) \stackrel{\text{def}}{=} \{e \setminus \{v\} \mid e \in G \land |e \setminus \{v\}|_{\geq 2}\}$$

Notice that, if $\{v, w\}$ is the only edge in *G* that contains *w*, then *w* does not belong to *Nodes* (*Filter* (*G*, *v*)). When we write *Nodes* (*G*), *Cov* (*G*, *P*), *Contains* (*G*, *v*), or *Filter* (*G*, *v*), we implicitly assume that both of *G* and *P* are graphs.

Let *G* and *v* be a graph and a node, respectively. We define *Lost* (*G*, *v*) to be the set of nodes of *G* that are not nodes of *Filter* (*G*, *v*) and differ from *v*.

Lost
$$(G, v) \stackrel{\text{def}}{=} Nodes (G) \setminus (Nodes (Filter (G, v)) \cup \{v\})$$

If *Lost* (G, v) is nonnull, then we say that v is a *losing vertex* of G and all of its elements are said to be *lost by* v.

Whenever *G* is connected and either *Filter* (*G*, *v*) is not connected or some of the nodes in *G* other than *v* do not belong to *Filter* (*G*, *v*), *v* is a *cut vertex* of *G* (**Cutting** (*G*, *v*) holds).

Cutting (G, v) :=**Conn** $(G) \land |G|_{\geq 2} \land (\neg$ **Conn** $(Filter (G, v)) \lor Lost (G, v) \neq \emptyset)$

We call *non-cut vertex* any vertex that is not a cut vertex.

3 Hypergraphs have non-cut vertices

Our goal is to provide a proof that every connected hypergraph contains a non-cut vertex. This is encoded by the following corollary.



Fig. 3: The property **Cutting** (*G*, *v*) holds whenever *Filter* (*G*, *v*) is not connected or it contains fewer nodes than *Nodes* (*G*) \ {*v*}. Both **Cutting** (*G*, *a*) and **Cutting** (*G*, *c*) do hold since *Filter* (*G*, *a*) lost the node *f* (see Fig. 3b) and *Filter* (*G*, *c*) is not connected (see Fig. 3c).

Corollary 1 *Conn* (*G*) \rightarrow (*G* = $\emptyset \lor \exists v \in Nodes$ (*G*) \neg *Cutting* (*G*, *v*))

We split the proof of above corollary into the proofs of two statements: (1) if *G* has a losing vertex, then *G* has is a non-cut vertex (see Section 3.1); (2) any connected hypergraph *G* contains a node *v* such that *Filter*(*G*, *v*) is connected (see Section 3.3). By definition of **Cutting**(·), this suffices to yield the claim of Corollary 1.

3.1 Losing vertices yield non-cut vertices

In this section, we prove that, if *G* has a losing vertex, then it has a non-cut vertex too. The following theorem formalizes this statement.

Theorem 1 (*Graph* (*G*) \land *Lost* (*G*, *v*) $\neq \emptyset$) $\rightarrow \exists v' \in Nodes$ (*G*) \neg *Cutting* (*G*, *v'*)

First of all we need to prove that v is a losing vertex if and only if there exists a vertex lost by v such that *Contains* $(G, v') = \{\{v, v'\}\}$. By definition of *Filter* (\cdot) , *Filter* (G, v) contains all the set $e \setminus \{v\}$ such that e is an edge of G and $e \setminus \{v\}$ has at least cardinality 2. Hence, $v' \neq v$ is a node of G and it is not a node of *Filter* (G, v)if and only if there exists an edge $e \in G$ such that $|e \setminus \{v\}| < 2$. Since e has at least cardinality 2 by definition of **Graph** (\cdot) , we can conclude that e equals $\{v, v'\}$.

Lemma 2 Graph (G) \rightarrow ($v' \in Lost(G, v) \leftrightarrow Contains(G, v') = \{\{v, v'\}\}$)

Proof. (→) By definition of *Nodes* (·), if $v' \in Nodes$ (*G*), then there exists $e \in G$ such that $v' \in e$. Analogously, if $v' \notin Nodes$ (*Filter* (*G*, *v*)), then $v' \notin e'$ for all $e' \in Filter$ (*G*, *v*). However, *Filter* (*G*, *v*) = $\{e \setminus \{v\} \mid e \in G \land |e \setminus \{v\}|_{\geq 2}\}$ by definition. Hence, if $v' \notin Nodes$ (*Filter* (*G*, *v*)), either $v' \notin e \setminus \{v\}$ or $|e \setminus \{v\}| < 2$ for all $e \in G$ and, if $v' \notin Nodes$ (*Filter* (*G*, *v*)) and $v' \in Nodes$ (*G*), then either v' = v or $|e \setminus \{v\}| < 2$ for all $e \in G$ such that $v' \in e$. Since $|e|_{\geq 2}$ by definition of **Graph** (·) and $v' \in Nodes$ (*G*) \ (*Nodes* (*Filter* (*G*, *v*)) $\cup \{v\}$) holds by hypothesis, $e = \{v, v'\}$ for all $e \in G$ such that $v' \in e$. Moreover, $\{v, v'\} \in G$ because if $v' \in Nodes$ (*G*) \ (*Nodes* (*Filter* (*G*, *v*)) $\cup \{v\}$), then $v' \in Nodes$ (*G*) and there exists $e \in G$ such that $v' \in e$. By definition of **Contains** (·), it follows that **Contains** (*G*, $v') = \{\{v, v'\}\}$.

(←) Let us assume that *Contains* (*G*, *v'*) = {{*v*, *v'*}. By definition of **Graph** (·), if $e \in G$, then $|e|_{\geq 2}$. By definition of *Contains* (·), *Contains* (*G*, *v'*) ⊆ *G* and, thus, {*v*, *v'*} ∈ *G* and $v \neq v'$. Moreover, $v' \in Nodes$ (*G*) by definition of *Nodes* (·) and $v' \notin e$ for all $e \in G \setminus \{v, v'\}$ by definition of *Contains* (·). By definition of *Filter* (·), *Filter* (*G*, *v*) is the set { $e \setminus \{v\} | e \in G \land |e \setminus \{v\}|_{\geq 2}$ }. Since *Contains* (*G*, *v'*) = {{*v*, *v'*} by hypothesis, if $|e \setminus \{v\}|_{\geq 2}$ holds, then $v' \notin e$. Thus, *v'* belongs neither to *Filter* (*G*, *v*) nor to *Filter* (*G*, *v*) ∪ {*v*} because $v \neq v'$ and $v' \in Nodes$ (*G*) \(*Filter* (*G*, *v*) ∪ {*v*}). \Box

Our next step is to prove that any boundary vertex is non-cut. Since v belongs to a single edge, for every subset P of G, either $v \in Nodes(P)$ and $v \notin Nodes(G \setminus P)$ or $v \notin Nodes(P)$ and $v \in Nodes(G \setminus P)$. In either case, if the set $Nodes(P) \cap Nodes(G \setminus P)$ is nonnull, so is $(Nodes(P) \cap Nodes(G \setminus P)) \setminus \{v\}$. If G is connected, either $G \setminus e$ is empty or v' belongs to another edge of G. In the former case, $G \setminus e$ is connected. In the latter, $Nodes(G) \setminus \{v\} = Nodes(G \setminus e)$ and $G \setminus e$ must be connected. Thus, if G is connected and v is contained exclusively by an edge e whose cardinality is 2, then $G \setminus e$ is still connected.

Lemma 3 (*Conn*(*G*) \land *Contains*(*G*, *v*) = {{*v*, *v'*}}) \rightarrow *Conn*(*G* \setminus *Contains*(*G*, *v*))

Proof. Let us assume that there exist G_0 , v_0 , and v'_0 such that both the formulæ **Conn** (G_0) and **Contains** (G_0, v_0) = {{ v_0, v'_0 }} hold, while the formula **Conn** ($G_0 \setminus Contains(G_0, v_0)$) does not. By definition of **Conn** (·), there exists a nonnull set P_0 such that P_0 is a proper subset of $G_0 \setminus Contains(G_0, v_0)$ and *Nodes* $(P_0) \cap Nodes ((G_0 \setminus Contains (G_0, v_0)) \setminus P_0) = \emptyset$. Thus, the set *Nodes* $(P_0) \cap$ *Nodes* (($G_0 \setminus P_0$) \ *Contains* (G_0, v_0)) is empty. However, the formula **Conn** (G_0) holds and, since P_0 is also a subset of G_0 , *Nodes* $(P_0) \cap Nodes$ $(G_0 \setminus P_0)$ must be not empty by definition of **Conn** (·). By writing $G_0 \setminus P_0$ as the union of $(G_0 \setminus P_0)$ *Contains* (G_0, v_0) and ($G_0 \setminus P_0$) \cap *Contains* (G_0, v_0), we infer that *Nodes* (P_0) \cap (*Nodes* ($(G_0 \setminus P_0) \setminus Contains(G_0, v_0)$)) \cup *Nodes* ($(G_0 \setminus P_0) \cap Contains(G_0, v_0)$)) is nonnull by Lemma 10. Hence, the set *Nodes* $((G_0 \setminus P_0) \cap Contains(G_0, v_0)) \cap$ *Nodes* $(P_0) \neq \emptyset$, *Nodes* $((G_0 \setminus P_0) \cap Contains (G_0, v_0)) \neq \emptyset$, and, by the definition of Nodes (\cdot) , $(G_0 \setminus P_0) \cap Contains(G_0, v_0) \neq \emptyset$. Since Contains $(G_0, v_0) = \{\{v_0, v'_0\}\},\$ $\{v_0, v'_0\} \in G_0 \setminus P_0 \text{ and } \{v_0, v'_0\} \notin P_0.$ However, $P_0 \subseteq G_0$ and *Contains* $(G_0, v_0) = \{e \in V_0, v'_0\}$ $G_0 | v_0 \in e \} = \{\{v_0, v'_0\}\}$; therefore *Contains* $(P_0, v_0) = \emptyset$, and $v_0 \notin Nodes$ (P_0) . Moreover, $(G_0 \setminus P_0) \cap Contains(G_0, v_0)$ must be subset of $\{\{v_0, v'_0\}\}$ and, by Lemma 9, *Nodes* $((G_0 \setminus P_0) \cap Contains(G_0, v_0))$ is subset of *Nodes* $(\{\{v_0, v'_0\}\})$ which is $\{v_0, v'_0\}$ by the definition of *Nodes* (·). The set *Nodes* $((G_0 \setminus P_0) \cap Contains(G_0, v_0)) \cap$ *Nodes* (P_0) is nonnull; therefore it must equal { v'_0 }, and v'_0 belongs to *Nodes* (P_0). It follows that:

Nodes $(P_0 \cup Contains(G_0, v_0)) =$

$=$ Nodes $(P_0) \cup$ Nodes $(Contains (G_0, v_0))$	By Lemma 10
$= Nodes(P_0) \cup Nodes(\{\{v_0, v'_0\}\})$	By hypothesis
$= Nodes(P_0) \cup \{v_0, v'_0\}$	By definition of <i>Nodes</i> (\cdot)
$= Nodes(P_0) \cup \{v_0\}$	Because $v_0 \notin Nodes(P_0)$

and, thus, *Nodes* $(G_0 \setminus (P_0 \cup Contains (G_0, v_0))) \cap Nodes (P_0 \cup Contains (G_0, v_0))$ is equal to both the sets *Nodes* $((G_0 \setminus Contains (G_0, v_0)) \setminus P_0)) \cap (Nodes (P_0) \cup \{v_0\})$ and *Nodes* $((G_0 \setminus Contains (G_0, v_0)) \setminus P_0)) \cap \{v_0\}$. However, $G_0 \setminus Contains (G_0, v_0)$ is equal to the set $\{e \in G_0 \mid v_0 \notin e\}$ by definition of *Contains* (·). Hence, v_0 does not belong to *Nodes* $(G_0 \setminus Contains (G_0, v_0))$ by definition of *Nodes* (·) and it does not also belong to *Nodes* $(G_0 \setminus Contains (G_0, v_0) \setminus P_0)$ by Lemma 9. Thus, the set *Nodes* $(G_0 \setminus Contains (G_0, v_0) \setminus P_0)$ by Lemma 9. Thus, the set *Nodes* $(G_0, v_0) \cup P_0) \cap \{v_0\}$ is empty, so is the set *Nodes* $(G_0 \setminus (Contains (G_0, v_0) \cup P_0)) \cap Nodes (P_0 \cup Contains (G_0, v_0))$, and the formula **Conn** (G_0) does not hold by definition of **Conn** (·). Since this last statement contradicts our assumptions, the claim must hold. \Box

Lemma 3 allows us to prove that whenever a vertex v belongs to a single edge e of G, G filtered w.r.t. v is still connected. As a matter of fact, *Filter* (G, v) contains all the set $e' \setminus \{v\}$ that have at least cardinality 2. Since v belongs only to e, if $e \setminus \{v\}$ is still an edge, i.e., it contains at least two nodes, the proof is straightforward because *Nodes* ($G \setminus \{v\} = Nodes$ (*Filter* (G, v)) and, thus, because of the above remarks, *Filter* (G, v) is connected. If, instead, $e \setminus \{v\}$ is not an edge, then there exists a node v' of G such that $e = \{v, v'\}$ and *Contains* (G, v) = $\{\{v, v'\}\}$. As a matter of fact, since G is connected either $G \setminus e$ is empty or v' belongs to another edge of G. In the former case, $G \setminus e$ is connected. In the latter, *Nodes* ($G \setminus \{v\} = Nodes$ ($G \setminus e$) and, because of the above remarks, $G \setminus e = Filter$ (G, v) must be connected.



Fig. 4: If the set *Contains* (G, v) is a singleton and the hypergraph *G* is connected, the hypergraph *Filter* (G, v) is connected and *v* is a non-cut vertex.

Lemma 4 (*Conn* (*G*) \land *Contains* (*G*, *v*) = {*e*}) \rightarrow *Conn* (*Filter* (*G*, *v*))

Proof. By definition, the set *Filter* (*G*, *v*) is equal to $\{e' \setminus \{v\} | e' \in G \land |e' \setminus \{v\}|_{\geq 2}\}$. Since *Contains* (*G*, *v*) = $\{e\}$, *Filter* (*G*, *v*) = (*G* \ $\{e\}$) $\cup \{e \setminus \{v\} | |e \setminus \{v\}|_{\geq 2}\}$. There are two cases: $|e \setminus \{v\}|_{\geq 2}$ does not hold or it does. In the former case, since $|e|_{\geq 2}$ by definition of **Graph** (·), $v \in e$, there exists a v' such that $e = \{v, v'\}$, and *Filter* (*G*, *v*) = *G* \ $\{e\}$. The claim follows from Lemma 3. If, otherwise, $|e \setminus \{v\}|_{\geq 2}$ holds, then *Filter* (*G*, *v*) = (*G* \ $\{e\}$) $\cup \{e \setminus \{v\}\}$. Let us assume that there exist G_0 , v_0 , and e_0 such that both the formulæ **Conn** (G_0) \land *Contains* (G_0 , v_0) = $\{e_0\}$ and \neg **Conn** (*Filter* (G_0 , v_0)) hold. By definition of **Conn** (·), *Nodes* ($G_0 \setminus P$) \cap *Nodes* (*P*) $\neq \emptyset$ for all *P* such that $\emptyset \subseteq P \subseteq G_0$. Since \neg **Conn** (*Filter* (G_0 , v_0)) holds, there exists a P_0 such that $\emptyset \subsetneq P_0 \subsetneq (G_0 \setminus \{e_0\}) \cup \{e_0 \setminus \{v_0\}\} = Filter (G_0, v_0)$ such that *Nodes* $(P_0) \cap Nodes$ (*Filter* $(G_0, v_0) \setminus P_0$) is empty. Let be P_1 be one of P_0 and *Filter* $(G_0, v_0) \setminus P_0$ such that $e_0 \setminus \{v_0\} \notin P_1$. Of course, *Nodes* $(P_1) \cap$ *Nodes* (*Filter* $(G_0, v_0) \setminus P_1$) = \emptyset and $P_1 \subseteq Filter (G_0, v_0) \setminus \{e_0 \setminus \{v_0\}\} \subseteq G_0$. The set *Filter* $(G_0, v_0) \setminus P_1$ is equal to $((G_0 \setminus \{e_0\}) \setminus P_1) \cup \{e_0 \setminus \{v_0\}\}$ and to $((G_0 \setminus P_1) \setminus \{e_0\}) \cup \{e_0 \setminus \{v_0\}\}$. Thus, *Nodes* (*Filter* $(G_0, v_0) \setminus P_1$) and *Nodes* $(G_0 \setminus P_1) \setminus \{v_0\}$ are the same set by the definition of *Nodes* (·) and *Nodes* $(G_0 \setminus P_1) \cap Nodes$ $(P_1) \subseteq \{v_0\}$. However, v_0 does not belong to *Nodes* (*Filter* (G_0, v_0)), by definition of both *Nodes* (·) and *Filter* (G_0, v_0) , and $P_1 \subsetneq Filter$ (G_0, v_0) . Hence, by Lemma 9, $v_0 \notin Nodes$ (P_1) and *Nodes* $(G_0 \setminus P_1) \cap Nodes$ $(P_1) = \emptyset$. Thus, **Conn** (G_0) does not hold by definition of **Conn** (·). This contradicts our assumptions and proves the claim. \Box

According to the definition of **Cutting**(·), **Cutting**(*G*, *v*) holds if and only if *G* is connected, *G* contains at least two edges, and either there exists a node $v' \neq v$ of *G* that is not a node of *Filter*(*G*, *v*) or *Filter*(*G*, *v*) is not connected. If *v* is a boundary vertex and *G* is connected, *Filter*(*G*, *v*) must be connected too by Lemma 4. It follows that, under the above conditions, **Cutting**(*G*, *v*) holds if and only if *G* is connected, *G* contains at least two edges, and there exists a node $v' \neq v$ of *G* that is not a node of *Filter*(*G*, *v*). However, the latter condition holds if and only if *Contains*(*G*, *v'*) = {{*v*, *v'*}} by Lemma 2. Thus, **Conn**(*G* \ *Contains*(*G*, *v*)) holds by Lemma 3 and **Conn**(*Filter*(*G*, *v*)). It follows that if *v* is a boundary vertex of *G*, then *v* is a non-cut edge.

Lemma 5 *Contains* $(G, v) = \{e\} \rightarrow \neg Cutting(G, v)$

Proof. Let us assume that there exist G_0 , v_0 , and e_0 such that *Contains* (G_0 , v_0) is $\{e\}$ and the formula **Cutting** (G_0 , v_0) holds. By definition of **Cutting** (·), both the formulæ ¬**Conn** (*Filter* (G_0 , v_0)) ∨ *Nodes* (*Filter* (G_0 , v_0)) ∪ $\{v_0\} \subseteq Nodes$ (G_0) and **Conn** (G_0) ∧ $|G_0|_{\geq 2}$ hold. However, **Conn** (*Filter* (G_0 , v_0)) must hold by Lemma 4. It follows that *Nodes* (*Filter* (G_0 , v_0)) ∪ $\{v_0\} \subseteq Nodes$ (G_0), there exists a $v_1 \in Nodes$ (G_0) \ *Nodes* (*Filter* (G_0 , v_0)) ∪ $\{v_0\}$, and, by Lemma 2, *Contains* (G_0 , v_1) = $\{\{v_0, v_1\}\}$. Since *Contains* (G_0 , v_0) = $\{e_0\}$, from definition of *Contains* (·) it follows that e_0 and $\{v_0, v_1\}$ are the same edge and, by Lemma 12, the set *Cov* (G_0 , $\{e_0\}$) is equal to the set ($\bigcup_{v \in Nodes}(e_0)$ *Contains* (G_0 , v_0)), which is $\{e_0\}$ by definition of *Nodes* (·). By Lemma 13, *Nodes* ($\{e_0\}$) ∩ *Nodes* ($G_0 \setminus \{e_0\}$) is empty and **Conn** (G_0) does not hold by definition of **Conn** (·). Since this contradicts our assumptions, the claim must hold.

As a direct consequence of Lemma 2 and Lemma 5, if *G* has a vertex $v' \neq v$ which is not a vertex of *Filter* (*G*, *v*), *G* has a non-cut vertex too.

Proof of Theorem 1. If *Nodes* (*Filter* (G, v)) \cup {v} is a proper subset of *Nodes* (G), then the set *Lost* (G, v) is nonnull and it contains a vertex a v'. By Lemma 2, the set *Contains* (G, v') is equal to {{v, v'}} and, by Lemma 5, \neg **Cutting** (G, v'). \Box

3.2 Not all graphs *Filter* (*G*, *v*) are disconnected

This section proves that if *G* is connected, then it contains a node *v* such that *Filter* (*G*, *v*) is connected too.

Theorem 2 *Conn* (*G*) \rightarrow (*G* = $\emptyset \lor \exists v \in Nodes$ (*G*) *Conn* (*Filter* (*G*, *v*)))

First of all, let us prove that, for any nonnull set *P* which is a proper subset of *G* and such that none of the nodes of *P* is a node of *Filter* (*G*, *v*) \ *P* and for any edge $e \in G$, either *e* does not contain any node of *P* or *e* does not contain any node of *Filter* (*G*, *v*) \ *P*. Namely, if *Filter* (*G*, *v*) is partitioned into two disconnected subgraphs (i.e., *P* and *Filter* (*G*, *v*) \ *P*), none of the edges of *G* "touches" both of these graphs. This is due to the fact that, besides vertices lost by *v*, which are boundary vertices by Lemma 2, only *v* belongs to *Nodes* (*G*) and not to *Nodes* (*Filter* (*G*, *v*) \ *P*) are subsets of *Lost* (*G*, *v*) \cup {*v*} for each $e \in G$. Thus, either $e \cap Nodes$ (*P*) or $e \cap Nodes$ (*Filter* (*G*, *v*) \ *P*) must be empty.

Lemma 6 (*Graph* (*G*) \land *P* \subseteq *Filter* (*G*, *v*) \land *Nodes* (*P*) \cap *Nodes* (*Filter* (*G*, *v*) \setminus *P*) = \emptyset) $\rightarrow \forall e \in G(e \cap Nodes(P) = \emptyset \lor e \cap Nodes(Filter(G, v) \setminus P) = \emptyset)$

Proof. Let us assume that there exist G_0 , P_0 , v_0 , and e_0 such that both the formulæ **Graph** $(G_0) \land P_0 \subseteq$ *Filter* $(G_0, v_0) \land Nodes$ $(P_0) \cap Nodes$ (*Filter* $(G_0, v_0) \land P_0$) = Ø and $e_0 \cap Nodes$ (*Filter* $(G_0, v_0) \land P_0$) ≠ Ø $\land e_0 \cap Nodes$ (P_0) ≠ Ø $\land e_0 \in G_0$ hold. Either (a) $v_0 \notin e_0$ or (b) $v_0 \in e_0$ holds. In the former case $e_0 \in$ *Filter* (G_0, v_0) , by definition of *Filter* (·) and **Graph** (·), and we get a contradiction by Lemma 11. Hence, case (b) must hold. By our assumptions, there exist $v_1 \in e_0 \cap Nodes$ (*Filter* $(G_0, v_0) \land P_0$) and $v_2 \in e_0 \cap Nodes$ (P_0). Since *Nodes* (*Filter* $(G_0, v_0) \land P_0$) \cap *Nodes* (P_0) is empty, $v_1 \neq v_2$ and $\{v_0, v_1, v_2\} \subseteq e_0$. Moreover, $v_0 \notin Nodes$ (*Filter* (G_0, v_0)) by definition of *Filter* (·) and, since $P_0 \subseteq Filter$ (G_0, v_0), $v_0 \notin Nodes$ (P_0) and $v_0 \notin Nodes$ (*Filter* $(G_0, v_0) \land P_0$) by Lemma 9. It follows that $v_0 \neq v_1$ and $v_0 \neq v_2$, $e_1 = e_0 \land \{v_0\}$ belongs to *Filter* (G_0, v_0) by definition of *Filter* (·), and $|e_1|_{\geq 2}$. Since both $e_1 \cap Nodes$ (P_0) and $e_1 \cap Nodes$ (*Filter* ($G_0, v_0 \land V_0$) $\land P_0$) is nonnull by Lemma 11. This contradict our assumptions, hence the claim must hold. \Box

In the light of the above considerations, it is easy to see also that, if *G* is connected and *Lost* (*G*, *v*) is empty, then *v* is the only vertex in both of the graphs *Cov* (*G*, *P*) and *Cov* (*G*, *Filter* (*G*, *v*) \setminus *P*).

 $\begin{array}{l} \textbf{Lemma 7} \quad (Lost (G, v) = \emptyset \land Conn (G) \land v \in Nodes (G) \land \\ \emptyset \subsetneq P \subsetneq Filter (G, v) \land Nodes (P) \cap Nodes (Filter (G, v) \setminus P) = \emptyset) \rightarrow \\ Nodes (Cov (G, P)) \cap Nodes (Cov (G, Filter (G, v) \setminus P)) = \{v\} \end{array}$

Proof. By the definition of **Conn** (·), **Graph** (*G*) holds. By the definition of *Lost* (·), if *Lost* (*G*, *v*) = \emptyset , then *Nodes* (*Filter* (*G*, *v*)) \supseteq *Nodes* (*G*) \ {*v*}. By Lemma 15, *Nodes* (*Filter* (*G*, *v*)) \subseteq *Nodes* (*G*)\{*v*}, hence *Nodes* (*G*)\{*v*} = *Nodes* (*Filter* (*G*, *v*)) and, by Lemma 16, *Cov* (*G*, *P*) \cup *Cov* (*G*, *Filter* (*G*, *v*) \ *P*) = *G*. Since *Nodes* (*P*) \cap

Nodes (*Filter* (*G*, *v*) \ *P*) = \emptyset for all nonnull *P* that is also a proper subset of *Filter* (*G*, *v*), *Cov* (*G*, *P*) \cap *Cov* (*G*, *Filter* (*G*, *v*) \ *P*) = \emptyset by Lemma 17. As a consequence, *Cov* (*G*, *Filter* (*G*, *v*) \ *P*) is equal to *G**Cov* (*G*, *P*) and *Nodes* (*Cov* (*G*, *P*)) \cap *Nodes* (*Cov* (*G*, *Filter* (*G*, *v*) \ *P*)) $\neq \emptyset$. By the definitions of *Nodes* (·) and *Cov* (·), *v'* belongs to *Nodes* (*Cov* (*G*, *P*)) \cap *Nodes* (*Cov* (*G*, *Filter* (*G*, *v*) \ *P*)) if and only if there exist *e*, *e'* \in *G* such that *e* \cap *Nodes* (*P*) and *e'* \cap *Nodes* (*Filter* (*G*, *v*) \ *P*) are nonnull and *v'* \in *e* \cap *e'*. Since *e*, *e'* \in *G*, *e* \cap *e'* is equal to (*e* \cap *e'*) \cap *Nodes* (*G*) by the definition of *Nodes* (·). Moreover *v* \in *Nodes* (*G*) by hypothesis, hence, *Nodes* (*G*) is equal to *Nodes* (*Filter* (*G*, *v*)) \cup {*v*} and to *Nodes* (*Filter* (*G*, *v*) \ *P*) \cup *Nodes* (*P*) \cup {*v*} by Lemma 10. Thus, *v'* \in *e* \cap *e'* if and only if either (a) *v'* \in *e* \cap (*e'* \cap *Nodes* (*P*)), (b) *v'* \in *e'* \cap (*e* \cap *Nodes* (*Filter* (*G*, *v*) \ *P*)), or (c) *v'* \in (*e* \cap *e'*) \cap {*v*}. However, due to Lemma 6, *e'* \cap *Nodes* (*P*) and *e* \cap *Nodes* (*Filter* (*G*, *v*) \ *P*) must be empty and both cases (a) and (b) are not possible. It follows that the claim holds.

By Lemma 2 and Lemma 4, if *G* is connected and *Lost* (*G*, *v*) is nonnull, then there exists a v' such that *Filter* (*G*, v') is connected. In order to prove Theorem 2 we are left to prove that, whenever *G* is connected and *Lost* (*G*, *v*) is empty, there exists a v' such that *Filter* (*G*, v') is connected.



Fig. 5: A graphical sketch of the proof of Theorem 2.

Let us assume by contradiction that the set *Filter* (*G*, *v*) is not connected for all $v \in Nodes$ (*G*). From the finiteness of *G*, we can deduce that, for every vertex *v* of *G*, there exists a subset of *Filter* (*G*, *v*) \cap *G* that is maximal and connected. Let us call it C_v and let *C* be the set of all these C_v 's. Since *G* is finite, so is *C* and, then, there exists a graph C_* in *C* that is maximal. Let v^* be such that C_* is subset of *Filter* (*G*, v^*) \cap *G*. Since *Filter* (*G*, v^*) is disconnected, there exists a nonnull Q_* that is a proper subset of *Filter* (*G*, v^*) such that *Nodes* (Q_*) \cap *Nodes* (*Filter* (*G*, v^*) $\setminus Q_*$) is empty, $C_* \subseteq Q_*$, and Q_* is minimal. However, none of the edges of *G* goes from Q_* to *Filter* (*G*, v^*) $\setminus Q_*$. Hence, for any node v' of *Filter* $(G, v^*) \setminus Q_*$ and for any edge *e* of *Contains* (G, v'), *e* does not share any nodes with Q_* . Hence, *Cov* (G_0, C_*) is connected. Moreover, since *Filter* (G, v') is not connected either and *Contains* (G, v^*) must share some nodes with Q_* , there exists a nonnull Q' that is a proper subset of *Filter* (G, v') such that *Nodes* $(Q') \cap$ *Nodes* (*Filter* $(G, v') \setminus Q'$) is empty and $Q_* \subseteq Q'$. Since *Contains* (G, v^*) must share some nodes with Q_* and Q_* is the minimal set that contains C_* , C_* must be a proper subset of *Cov* (G_0, C_*) . This contradicts our assumptions and, thus, there must exist a $v \in$ *Nodes* (G) such that *Filter* (G, v) is connected.

Proof of Theorem 2. As a preamble, notice that if **Conn** (*G*) and **Lost** (*G*, *v*) $\neq \emptyset$ hold together, they imply, respectively, that G is a graph and that there is a $v' \in Lost(G, v)$; hence Contains $(G, v') = \{\{v, v'\}\}$ holds by Lemma 2, and **Conn** (*Filter* (G, v')) holds by Lemma 4. Arguing by contradiction, suppose then that a G_0 exists satisfying **Conn** (G_0), $\forall v \in Nodes (G_0) \neg Conn (Filter (G_0, v))$, $\forall v \in Nodes(G_0) Lost(G_0, v) = \emptyset$, and $G_0 \neq \emptyset$. Since Nodes(Filter(G_0, v_0)) \subseteq **Nodes** $(G_0) \setminus \{v\}$ holds for all v, by Lemma 15, in view of the definition of Lost (·) we get Nodes (Filter (G_0, v)) = Nodes $(G_0) \setminus \{v\}$ for all v. Let C be the set { $P \subseteq G_0 \mid \exists v \in Nodes(G_0) \mid P \subseteq Filter(G_0, v) \land Conn(P)$ }. From the finiteness of G_0 , we get the finiteness of C and hence the existence of an inclusion-maximal $C_* \in C$; thus, for no $v \in Nodes(G_0)$ there exists any $P \subseteq Filter(G_0, v) \cap G_0$ such that **Conn** (*P*) and $C_* \subseteq P$. The set C_* is empty if and only if *Filter* (G_0, v) $\cap G_0$ is empty for all $v \in Nodes(G_0)$, because Conn($\{e\}$) holds for all $e \in G_0$; accordingly, by Lemma 21, if $C_* = \emptyset$ then **Conn** (*Filter* (G_0, v)) holds for all $v \in$ *Nodes* (G_0). However, this would contradict our assumptions; hence $C_* \neq \emptyset$ necessarily holds. It follows from $C_* \in C$ that a $v^* \in Nodes(G_0)$ such that $C_* \subseteq$ *Filter* (G_0, v^*) must exist. Moreover, since **Conn** (*Filter* (G_0, v)) does not hold for any $v \in Nodes(G_0)$, a Q^* must exist such that $\emptyset \subsetneq Q^* \subsetneq Filter(G_0, v^*)$ and *Nodes* $(Q^*) \cap Nodes$ (*Filter* $(G_0, v^*) \setminus Q^*$) = \emptyset by the definition of **Conn** (·). However, C_* is a subset of *Filter* (G_0 , v^*) by construction; thus, by Lemma 18, either $C_* \subseteq Q^*$ or $C_* \subseteq Filter(G_0, v^*) \setminus Q^*$ holds. Since $Filter(G_0, v^*) \setminus (Filter(G_0, v^*) \setminus Q^*)$ Q^*) equals Q^* and *Nodes* (*Filter* $(G_0, v^*) \setminus Q^*$) \cap *Nodes* $(Q^*) = \emptyset$ if and only if the set Nodes (Filter $(G_0, v^*) \setminus (Filter (G_0, v^*) \setminus Q^*)) \cap Nodes (Filter (G_0, v^*) \setminus Q^*)$ is empty, we can assume without loss of generality that $C_* \subseteq Q^*$. By the definition of *Filter* (·), it holds that $v^* \notin Nodes$ (*Filter* (G_0, v^*)) and, therefore, $v^* \notin Nodes(Q^*)$. By Lemma 16, $G_0 = Cov(G_0, Q^*) \cup Cov(G_0, Filter(G_0, v^*) \setminus Q^*)$, while *Cov* (G_0 , *Filter* (G_0 , v^*) \ Q^*) \cap *Cov* (G_0 , Q^*) = \emptyset by Lemma 17. It follows that $Cov(G_0, Filter(G_0, v^*) \setminus Q^*)$ equals $G_0 \setminus Cov(G_0, Q^*)$ and $Nodes(Cov(G_0, Q^*)) \cap$ *Nodes* (*Cov* (G_0 , *Filter* (G_0 , v^*) \ Q^*)) = { v^* } by Lemma 7. Therefore we have $v^* \in Nodes(Cov(G_0, Q^*))$, whence Conn(Cov(G_0, Q^*)), by Lemma 19. Since $\emptyset \subsetneq C_* \subseteq Q^* \subsetneq Filter(G_0, v^*), C_* \subsetneq Cov(G_0, Q^*)$ by Lemma 9, *Filter*(G_0, v^*) \ Q^* is nonnull, and so is *Cov* (G_0 , *Filter* (G_0 , v^*) \ Q^*) = $G_0 \setminus Cov$ (G_0 , Q^*) by Lemma 14. From **Graph** (G_0) it follows that there are no singletons in G_0 . Hence, there exists a $v_1 \in Nodes(G_0 \setminus Cov(G_0, Q^*)) \setminus \{v^*\}$ by the definition of *Nodes*(·). As a consequence of Lemma 20, we get *Cov* (G_0, Q^*) \subseteq *Filter* (G_0, v_1), which contradicts the assumed maximality of C_{v^*} in C. This contradiction proves our claim.

3.3 Every graph has a non-cut vertex

Corollary 1 is a a direct consequence of Theorem 1 and Theorem 2.

Proof of Corollary 1. Suppose that there exists a graph G_0 such that **Conn** (G_0) , $G_0 \neq \emptyset$, and $\forall v \in Nodes(G_0)$ **Cutting** (G_0, v) hold. By definition of **Cutting** (\cdot) , the formula **Conn** $(G_0) \land |G_0|_{\geq 2}$ holds and, for all $v \in Nodes(G_0)$, either the set *Lost* (G_0, v) is nonnull or **Conn** (*Filter* (G_0, v)) does not hold. If *Lost* (G_0, v) is nonnull, then there exists a $v_0 \in Lost(G_0, v)$ such that \neg **Cutting** (G_0, v_0) holds, by Theorem 1, and we get a contradiction. Hence, it must be the case that the formula **Conn** (G_0) holds, *Lost* (G_0, v) is empty, while **Conn** (*Filter* (G_0, v)) does not hold for any $v \in Nodes(G_0)$. However, by Theorem 2, there must exists a $v' \in Nodes(G_0)$ such that **Conn** (*Filter* (G_0, v')) holds. This is the sought contradiction which proves our claim.

4 Conclusions

This paper tackles the proof of a property of connected hypergraphs: every hypergraph has non-cut vertices. While this result is not new at all, the novelty of our work lies in the absence of the notion of path. As a matter of fact, we define as connected only those graphs whose edges cannot be parted into two nonnull subgraphs that do not share nodes. This approach has some relevance from a foundational point of view as it proves that paths are not necessary to identify non-cut vertices.

Since our proof is based on the ZF theory and it is complete down to the details, we plan to both verify its correctness by means of the proof checker Referee/Ætnanova [6] and to include these results in its scenarios on graphs (see, e.g., [5]). Documentation about these new experiments will be made available at http://aetnanova.units.it/scenarios/NonCutVertices/.

References

- G.-B. Chae, E. M. Palmer, and R. W. Robinson. Counting labeled general cubic graphs. Discrete Mathematics, 307(23):2979–2992, 2007.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, et al. *Introduction to algorithms*, volume 2. MIT press Cambridge, 2001.
- 3. M. C. Golumbic. Algorithmic graph theory and perfect graphs, volume 57. Elsevier, 2004.
- 4. M. Milanič and A. I. Tomescu. Set Graphs. I. Hereditarily Finite Sets and Extensional Acyclic Orientations. *Discrete Applied Mathematics*, 161(4-5):677–690, 2013.
- E. G. Omodeo and A. I. Tomescu. Set Graphs. III. Proof Pearl: Claw-Free Graphs Mirrored into Transitive Hereditarily Finite Sets. *Journal of Automated Reasoning*, 52(1):1–29, 2014.
- 6. J. T. Schwartz, D. Cantone, and E. G. Omodeo. *Computational Logic and Set Theory – Applying formalized Logic to Analysis*. Springer, 2011. Foreword by Martin Davis.
- A. Tarski. Sur les ensembles finis. Fundamenta Mathematicae, 6(1):45–95, 1924.
- F. Wiedijk. The QED Manifesto revisited. Studies in Logic, Grammar and Rhetoric, 10(23):121–133, 2007.

A Basic Properties

Lemma 8 $\exists e \ (e \in G \land v \in e) \leftrightarrow v \in Nodes (G)$

Proof. (\leftarrow) By the definition of *Nodes* (\cdot), *Nodes* (G) = $\bigcup_{e \in G} e$. Hence, if $v \in Nodes$ (G), there should exists a $e_0 \in G$ such that $v \in e_0$.

(→) By the definition of *Nodes* (·), *Nodes* (*G*) = $\bigcup_{e \in G} e$. Hence, if there exists a e_0 such that $e_0 \in G \land v \in e_0$, then $v \in Nodes$ (*G*).

Lemma 9 $P \subseteq G \rightarrow Nodes(P) \subseteq Nodes(G)$

Proof. By Lemma 8, if $v \in Nodes(P)$, then there exists $e \in P$ such that $v \in e$. Since $P \subseteq G$ by hypothesis, $e \in G$ and, by Lemma 8, $v \in Nodes(G)$.

Lemma 10 Nodes $(P \cup Q) = Nodes (P) \cup Nodes (Q)$

Proof. (⊇) Since $P \subseteq P \cup Q$, by Lemma 9, *Nodes* (P) \subseteq *Nodes* ($P \cup Q$). Analogously, we get that *Nodes* (Q) \subseteq *Nodes* ($P \cup Q$). Hence, *Nodes* (Q) \cup *Nodes* (P) \subseteq *Nodes* ($P \cup Q$) \cup *Nodes* ($P \cup Q$) = *Nodes* ($P \cup Q$).

(⊆) By Lemma 8, if $v \in Nodes(P \cup Q)$ then there exists a $e \in P \cup Q$ such that $v \in e$. Hence, *e* belongs to either *P* or *Q*. If $e \in P$, then $v \in Nodes(P)$ by Lemma 8. Symmetrically, if $e \in Q$, then $v \in Nodes(Q)$ by Lemma 8. Thus, if $v \in Nodes(P \cup Q)$, then $v \in Nodes(P) \cup Nodes(Q)$. □

Lemma 11 $P \subseteq G \rightarrow (Nodes(P) \cap Nodes(G \setminus P) = \emptyset \leftrightarrow \forall e \in G(e \cap Nodes(P) = \emptyset \lor e \cap Nodes(G \setminus P) = \emptyset))$

Proof. By the definition of *Nodes* (·), *Nodes* (*G*) = $\bigcup_{e \in G} e$. Thus, *Nodes* (*P*) \cap *Nodes* (*G* \ *P*) = \emptyset if and only if both $e \cap Nodes$ (*P*) and $e' \cap Nodes$ (*G* \ *P*) are empty for all $e \in G \setminus P$ and for all $e' \in P$. However, if $P \subseteq G$, then $(G \setminus P) \cup P$ is equal to *G*. Thus, if $P \subseteq G$, then *Nodes* (*P*) $\cap Nodes$ (*G* \ *P*) = \emptyset if and only if $\forall e \in G$ ($e \cap Nodes$ (*P*) = $\emptyset \lor e \cap Nodes$ (*G* \ *P*) = \emptyset).

Lemma 12 $Cov(G, P) = \bigcup_{v \in Nodes(P)} Contains(G, v)$

Proof. We prove that $e \in Cov(G, P)$ if and only if $e \in \bigcup_{v \in Nodes(P)} Contains(G, v)$. By definition of $Cov(\cdot)$, $e \in Cov(G, P)$ if and only if and only if $e \in G$ and $e \cap Nodes(P) \neq \emptyset$. Hence, $e \in Cov(G, P)$ if and only if there exists a $v \in Nodes(P)$ and a $e \in G$ such that $v \in e$. By definition of *Contains*(\cdot), $e \in Contains(G, v)$ if and only if $v \in e$ and $e \in G$. Thus, $e \in Cov(G, P)$ if and only if there exists a $v \in Nodes(P)$ and only if $v \in e$ and $e \in G$. Thus, $e \in Cov(G, P)$ if and only if there exists a $v \in Nodes(P)$ such that $e \in Contains(G, v)$. The thesis follows directly. \Box

Lemma 13 *Cov* (*G*, *P*) \subseteq *P* \leftrightarrow *Nodes* (*G* \ *P*) \cap *Nodes* (*P*) = \emptyset

Proof. By the definition of *Cov* (·), *Cov* (*G*, *P*) is equal to $\{e \in G | e \cap Nodes (P) \neq \emptyset\}$. Hence, *Cov* (*G*, *P*) \subseteq *P* if and only if $e \cap Nodes (P) = \emptyset$ for all $e \in G \setminus P$ and if and only if $(\bigcup_{e \in G \setminus P} e) \cap Nodes (P) = \emptyset$. By definition of *Nodes* (·), this is equivalent to *Nodes* (*G* \ *P*) $\cap Nodes (P) = \emptyset$ and, thus, the thesis holds. **Lemma 14** *Nodes* (*P*) \subseteq *Nodes* (*G*) \land *Nodes* (*P*) $\neq \emptyset \rightarrow Cov(G, P) \neq \emptyset$

Proof. Let us assume that there exist G_0 and P_0 such that both *Nodes* $(P_0) \subseteq$ *Nodes* $(G_0) \land Nodes$ $(P_0) \neq \emptyset$ and *Cov* $(G_0, P_0) = \emptyset$ hold. Since *Cov* (G_0, P_0) is equal to $\{e \in G_0 \mid e \cap Nodes$ $(P_0) \neq \emptyset\}$ by the definition of *Cov* (\cdot) , *Cov* $(G_0, P_0) = \emptyset$ if and only if *Nodes* $(G_0) \cap Nodes$ (P_0) is empty. However, from *Nodes* $(P_0) \subseteq Nodes$ (G_0) , we deduce that *Nodes* $(G_0) \cap Nodes$ (P_0) and *Nodes* (P_0) are the same set and, from *Nodes* $(P_0) \neq \emptyset$, we get that *Nodes* $(G_0) \cap Nodes$ (P_0) is not empty. This leads to a contradiction and proves our goal. □

Lemma 15 *Nodes* (*Filter* (G, v)) \subseteq *Nodes* (G) \ {v}

Proof. By the definition of *Nodes* (·), if *Nodes* (*Filter* (*G*, *v*)) = $\bigcup_{e' \in Filter(G,v)} e'$. However, by the definition of *Filter* (·), $e' \in Filter(G, v)$ if any only if there exists a $e \in G$ such that $e' = e \setminus \{v\}$ and $|e \setminus \{v\}|_{\geq 2}$. Thus, *Nodes* (*Filter* (*G*, *v*)) = $\bigcup_{e \in G} ||e \setminus \{v\}|_{\geq 2} ||e \mid 2} ||e \mid ||e \mid 2 ||e$

B Auxiliary Lemmas

Lemma 16 $(Graph(G) \land Nodes(Filter(G, v)) = Nodes(G) \setminus \{v\} \land$ $P \subseteq Filter(G, v)) \rightarrow Cov(G, P) \cup Cov(G, Filter(G, v) \setminus P) = G$

Proof. Let us assume that there exist G_0 , P_0 , and v_0 such that both the formulæ **Graph** $(G_0) \land Nodes$ (*Filter* (G_0, v_0)) = *Nodes* $(G_0) \setminus \{v_0\} \land P_0 \subseteq Filter$ (G_0, v_0) *Cov* $(G_0, P_0) \cup Cov$ $(G_0, Filter$ $(G_0, v_0) \setminus P_0$) $\neq G_0$ hold. By the definition of *Cov* (\cdot) , *Cov* (G_0, P) is the set $\{e \in G_0 \mid e \cap Nodes$ $(P) \neq \emptyset\}$. Thus, *Cov* $(G_0, P) \subseteq G_0$ and *Cov* $(G_0, P_0) \cup Cov$ $(G_0, Filter$ $(G_0, v_0) \setminus P_0) \subseteq G_0$. However, *Cov* $(G_0, P_0) \cup$ *Cov* $(G_0, Filter <math>(G_0, v_0) \setminus P_0) \neq G_0$ by assumption, hence, there exists a $e_0 \in G_0$ that belongs to neither *Cov* (G_0, P_0) nor *Cov* $(G_0, Filter$ $(G_0, v_0) \setminus P_0)$. By the definition of *Cov* $(\cdot), e_0 \cap Nodes$ $(P_0) = \emptyset$ and $e_0 \cap Nodes$ (*Filter* $(G_0, v_0) \setminus P_0$) = \emptyset . Thus $e_0 \cap (Nodes$ $(P_0) \cup Nodes$ (*Filter* $(G_0, v_0) \setminus P_0$)) = \emptyset and $e_0 \cap Nodes$ (*Filter* (G_0, v_0)) is empty by Lemma 10. Since *Nodes* (*Filter* (G_0, v_0)) = *Nodes* $(G_0) \setminus \{v_0\}$ by assumption, $e_0 \cap (Nodes$ $(G_0) \setminus \{v_0\}$) is empty and $e_0 \cap Nodes$ $(G_0) \subseteq \{v_0\}$. By definition of *Nodes* $(\cdot), e_0 \subseteq Nodes$ (G_0) . Thus, $e_0 \cap Nodes$ $(G_0) = e_0$ and $e_0 \subseteq \{v_0\}$. However, this contradicts $|e_0|_{\geq 2}$ which must hold because of the definition of **Graph** (\cdot) . The claim of this lemma follows readily.

Lemma 17 $(Graph(G) \land Nodes(Filter(G, v) \setminus P) \cap Nodes(P) = \emptyset \land P \subseteq Filter(G, v)) \rightarrow Cov(G, P) \cap Cov(G, Filter(G, v) \setminus P) = \emptyset$

Proof. By Lemma 6, either $e \cap Nodes(P)$ or $e \cap Nodes(Filter(G, v) \setminus P)$ are empty for all $e \in G$. By the definition of *Cov*(·), it follows that either $e \notin Cov(G, P)$ or $e \notin Cov(G, Filter(G, v) \setminus P)$ for all $e \in G$. Thus, $Cov(G, P) \cap Cov(G, Filter(G, v) \setminus P)$ is empty and the claim holds.

Lemma 18 (*Nodes* (Q)
$$\cap$$
 Nodes (G \ Q) = $\emptyset \land P \subseteq G \land Conn(P)$) \rightarrow
(P $\subseteq G \setminus Q \lor P \subseteq Q$)

Proof. Let us assume that there exist G_0 , P_0 , and Q_0 such that both the formulæ *Nodes* $(Q_0) \cap Nodes$ $(G_0 \setminus Q_0) = \emptyset \land P_0 \subseteq G_0 \land Conn$ (P_0) and $P_0 \notin G_0 \setminus Q_0 \land P_0 \notin Q_0$ hold. Hence, there exist an $e_0 \in P_0$ that does not belong to $G_0 \setminus Q_0$ and an $e_1 \in P_0$ that does not belong to Q_0 . It follows that both the sets $P_0 \setminus Q_0$ and $P_0 \setminus (G_0 \setminus Q_0)$ are nonnull. Since $P_0 \subseteq G_0$, e_0 belongs to Q_0 . Thus, $P_0 \setminus Q_0$ is a nonnull set that is also proper subset of P_0 . By definition of **Conn** (\cdot) , **Conn** (P_0) holds if and only if the set *Nodes* $(P_0 \setminus S) \cap Nodes$ (S) is nonnull for all not empty S that are also proper subsets of P_0 . In particular, the set *Nodes* $(P_0 \setminus Q_0)) \cap Nodes$ $(P_0 \setminus Q_0)$ is nonnull. Thus,

$\emptyset \subsetneq Nodes(P_0 \setminus (P_0 \setminus Q_0)) \cap Nodes(P_0 \setminus Q_0)$	
=Nodes $(P_0 \cap Q_0) \cap Nodes (P_0 \setminus Q_0)$	
\subseteq <i>Nodes</i> (Q_0) \cap <i>Nodes</i> ($P_0 \setminus Q_0$)	Since $P_0 \cap Q_0 \subseteq P_0$, by Lemma 9
\subseteq <i>Nodes</i> (Q_0) \cap <i>Nodes</i> ($G_0 \setminus Q_0$)	Since $P_0 \subseteq G_0$, by Lemma 9

This contradicts our assumptions and proves the claim.

Lemma 19 (*Conn* $(P \cup Q) \land Nodes (P) \cap Nodes (Q) = \{v\}) \rightarrow Conn (P)$

Proof. Let us assume that there exist P_0 , Q_0 and v_0 such that both the formulæ **Conn** $(P_0 \cup Q_0) \land Nodes(P_0) \cap Nodes(Q_0) = \{v_0\}$ and $\neg Conn(P_0)$ hold. By definition of **Conn** (·), there exists a P_1 such that $\emptyset \subseteq P_1 \subseteq P_0$ and $Nodes(P_1) \cap Nodes(P_0 \setminus P_1)$ is empty. Furthermore,

$\{v_0\} = Nodes(P_0) \cap Nodes(Q_0)$	By assumption
=Nodes $((P_0 \setminus P_1) \cup P_1) \cap Nodes (Q_0)$	Since $P_1 \subseteq P_0$
$= (Nodes (P_0 \setminus P_1) \cup Nodes (P_1)) \cap Nodes (Q_0)$	By Lemma 10
$= (Nodes (P_0 \setminus P_1) \cap Nodes (Q_0)) \cup$	
$(Nodes(P_1) \cap Nodes(Q_0))$	

Thus, either $v_0 \in Nodes(P_0 \setminus P_1) \cap Nodes(Q_0)$ or $v_0 \in Nodes(P_1) \cap Nodes(Q_0)$. Since $Nodes(P_0 \setminus P_1) \cap Nodes(P_1) = \emptyset$, either $v_0 \notin Nodes(P_0 \setminus P_1)$ or $v_0 \notin Nodes(P_1)$. Moreover, $(Nodes(P_0 \setminus P_1) \cap Nodes(Q_0)) \cup (Nodes(P_1) \cap Nodes(Q_0))$ is equal to $\{v_0\}$, hence, one of the two sets $Nodes(P_0 \setminus P_1) \cap Nodes(Q_0)$ and $Nodes(P_1) \cap Nodes(Q_0)$ is empty. Let be P_2 be such that $P_2 \in \{P_1, P_0 \setminus P_1\}$ and $Nodes(P_2) \cap Nodes(Q_0)$ is empty. Since $\emptyset \subsetneq P_1 \subsetneq P_0$, P_2 is nonnull, it is a proper subset of P_0 and $(P_0 \setminus P_2) \cup P_2 = P_0$. Moreover, $P_1 = P_0 \setminus (P_0 \setminus P_1)$ and, hence, $Nodes(P_2) \cap Nodes(P_0 \setminus P_2) = Nodes(P_1) \cap Nodes(P_0 \setminus P_1) = \emptyset$ is empty. By the definition of $Nodes(\cdot)$, $Nodes(P_0) \cap Nodes(Q_0) = \{v_0\}$ if and only if $e \cap e' \subseteq \{v_0\}$ for all $e \in P_0$ and for all $e' \in Q_0$. Hence, $|e \cap e'|_{\geq 2}$ does not hold and, by the definition of **Graph**(·), this means that $P_0 \cap Q_0 = \emptyset$. Since $P_2 \subseteq P_0, P_2 \cap Q_0 \subseteq P_0 \cap Q_0$ and $P_2 \cap Q_0$ is empty. It follows that

 $\emptyset = (Nodes (P_2) \cap Nodes (P_0 \setminus P_2)) \cup \\ (Nodes (P_2) \cap Nodes (Q_0)) \\ = Nodes (P_2) \cap (Nodes (P_0 \setminus P_2) \cup Nodes (Q_0)) \\ = Nodes (P_2) \cap Nodes ((P_0 \setminus P_2) \cup Q_0) \\ = Nodes (P_2) \cap Nodes ((P_0 \cup Q_0) \setminus P_2) \\ \end{bmatrix}$ By Lemma 10 = Nodes (P_2) \cap Nodes ((P_0 \cup Q_0) \setminus P_2) \\ By Lemma 10

By the definition of **Conn** (·), **Conn** ($P_0 \cup Q_0$) does not hold. This contradicts our assumptions and prove the claim.

Lemma 20 $(Graph(P \cup Q) \land Nodes(P) \cap Nodes(Q) = \{v\}) \rightarrow \forall v' \in Nodes(Q) \setminus \{v\} (P \subseteq Filter(P \cup Q, v'))$

Proof. By definition of *Filter* (\cdot) ,

$$\begin{aligned} \textit{Filter} \left(P \cup Q, v' \right) = & \{ e \setminus \{v'\} \mid e \in (P \cup Q) \land |e \setminus \{v'\}|_{\geq 2} \\ = & \{ e \setminus \{v'\} \mid e \in P \land |e \setminus \{v'\}|_{\geq 2} \} \cup \{ e \setminus \{v'\} \mid e \in Q \land |e \setminus \{v'\}|_{\geq 2} \\ \end{aligned}$$

Since $v' \in Nodes(Q) \setminus \{v\}$ and $Nodes(P) \cap Nodes(Q) = \{v\}$, then $v' \notin Nodes(P)$. By definition of $Nodes(\cdot)$, it follows that $v' \notin e$ and $e = e \setminus \{v'\}$ for any $e \in P$. Thus,

$$Filter (P \cup Q, v') = \{e \mid e \in P \land |e|_{\geq 2}\} \cup \{e \setminus \{v'\} \mid e \in Q \land |e \setminus \{v'\}|_{\geq 2}\}$$
$$= P \cup \{e \setminus \{v'\} \mid e \in Q \land |e \setminus \{v'\}|_{\geq 2}\}$$

The claim readily follows from the last equation.

Lemma 21 (*Graph*(*G*) $\land \forall v \in Nodes$ (*G*) *Filter*(*G*, *v*) $\cap G = \emptyset$) $\rightarrow \forall v \in Nodes$ (*G*) *Conn*(*Filter*(*G*, *v*))

Proof. The graph *Filter* (G, v) \cap G is empty for all $v \in Nodes$ (G) if and only if *Contains* (G, v) = G for all $v \in Nodes$ (G), i.e., every edge of G contains all nodes of G. It follows that G is a singleton and that each *Filter* (G, v) is either empty or a singleton; hence, by the definition of **Conn** (\cdot), **Conn** (*Filter* (G, v)) holds for all $v \in Nodes$ (G). The claim follows readily.