FORMALIZED MATHEMATICS
Volume 31, 2023, Pages 31–52
DOI: 10.2478/forma-2023-0004
e-ISSN: 1898–9934

sciendo.com/journal/forma

# Introduction to Graph Enumerations

Sebastian Koch[1]
Mainz, Germany

**Summary.** In this article sets of certain subgraphs of a graph are formalized in the Mizar system [7], [1], based on the formalization of graphs in [11] briefly sketched in [12]. The main result is the spanning subgraph theorem.

MSC: 05C05 05C30 68V20

Keywords: graph enumeration; spanning tree

MML identifier: `GLENUM00`, version: `8.1.12 5.74.1441`

## Introduction

Subsets of the set of all subgraphs of a graphs are rather rarely addressed directly (cf. [13], [4], [3]), but used as a tool in a wide variety of graph theory topics; e.g. they are needed for graph factorisation, graph reconstruction, random graphs, counting a special type of subgraphs and proving that every connected graph has a spanning subgraph (cf. [2], [14], [5]). The latter is proven in Section 7 of this article, together with the sharper result that we can even guarantee a spanning graph containing an arbitrary edge of the connected graph. As a necessity for that the set of all subtrees of a graph was introduced, as Jessica Enright and Piotr Rudnicki wished for in [6]. This article lays the groundwork for further formalization of any of these topics, in some sense extending and reusing [8] and [10]. It is noteworthy that the attribute `plain` from [9] was utilized here.

[1]mailto: `fly.high.android@gmail.com`

## 1. Subgraph Set and Subgraph Relation

From now on $G$, $G_1$, $G_2$ denote graphs and $H$ denotes a subgraph of $G$.

Let us consider $G$. The functor $G$.allSubgraphs() yielding a graph-membered set is defined by the term

(Def. 1) {the plain subgraph of $G$ induced by $V$ and $E$, where $V$ is a non empty subset of the vertices of $G$, $E$ is a subset of the edges of $G$ : $E \subseteq G$.edgesBetween($V$)}.

We introduce the notation $G$.allSG() as a synonym of $G$.allSubgraphs(). Let $G$ be a finite graph. One can check that $G$.allSG() is finite. Now we state the propositions:

(1) $G_2 \in G_1$.allSG() if and only if $G_2$ is a plain subgraph of $G_1$.

(2) $H\restriction$(the graph selectors) $\in G$.allSG(). The theorem is a consequence of (1).

(3) $G\restriction$(the graph selectors) $\in G$.allSG(). The theorem is a consequence of (2).

Let us consider $G$. Let $V$ be a non empty subset of the vertices of $G$. The functor createGraph($V$) yielding a plain subgraph of $G$ is defined by the term

(Def. 2) createGraph($V, \emptyset$, the function from $\emptyset$ into $V$, the function from $\emptyset$ into $V$).

Let us note that createGraph($V$) is edgeless. Now we state the propositions:

(4) Let us consider a non empty subset $V$ of the vertices of $G$. Then createGraph($V$) $\in G$.allSG().

(5) Let us consider a non empty subset $V$ of the vertices of $G$, and a subgraph $H$ of $G$ induced by $V$ and $\emptyset$. Then $H \approx$ createGraph($V$).

(6) Let us consider a subgraph $H$ of $G$ with edges the edges of $G$ removed. Then $H \approx$ createGraph($\Omega_\alpha$), where $\alpha$ is the vertices of $G$. The theorem is a consequence of (5).

(7) $G$ is edgeless if and only if $G \approx$ createGraph($\Omega_\alpha$), where $\alpha$ is the vertices of $G$. The theorem is a consequence of (6).

(8) Let us consider a non empty subset $V$ of the vertices of $G_1$. Suppose $V \subseteq$ the vertices of $G_2$. Then createGraph($V$) is a subgraph of $G_2$.

(9) $G$ is edgeless if and only if $G$.allSG() $=$ the set of all createGraph($V$) where $V$ is a non empty subset of the vertices of $G$. The theorem is a consequence of (1), (7), (4), and (3).

Let us consider $G$. Let $v$ be a vertex of $G$. The functor createGraph($v$) yielding a plain subgraph of $G$ is defined by the term

(Def. 3) createGraph($\{v\}$).

Let us note that createGraph($v$) is trivial and edgeless. Now we state the propositions:

(10) Let us consider a vertex $v$ of $G$. Then createGraph($v$) $\in G$.allSG().

(11) Let us consider a vertex $v$ of $G$, and a subgraph $H$ of $G$ induced by $\{v\}$ and $\emptyset$. Then $H \approx$ createGraph($v$).

(12) Let us consider a vertex $v$ of $G_1$. Suppose $v \in$ the vertices of $G_2$. Then createGraph($v$) is a subgraph of $G_2$.

Let $G$ be a non edgeless graph and $e$ be an edge of $G$.

The functor createGraph($e$) yielding a plain subgraph of $G$ is defined by

(Def. 4) there exists a non empty subset $V$ of the vertices of $G$ and there exist functions $S$, $T$ from $\{e\}$ into $V$ such that $it$ = createGraph($V, \{e\}, S, T$) and $\{$(the source of $G$)($e$), (the target of $G$)($e$)$\} = V$ and $S = e \dot{\longmapsto}$(the source of $G$)($e$) and $T = e \dot{\longmapsto}$(the target of $G$)($e$).

Let us consider a non edgeless graph $G$ and an edge $e$ of $G$. Now we state the propositions:

(13) (i) the edges of createGraph($e$) $= \{e\}$, and

(ii) the vertices of createGraph($e$) $= \{$(the source of $G$)($e$), (the target of $G$)($e$)$\}$.

(14) $e$ joins (the source of $G$)($e$) to (the target of $G$)($e$) in createGraph($e$). The theorem is a consequence of (13).

Let us consider a non edgeless graph $G$, an edge $e$ of $G$, and objects $e_0$, $v$, $w$. Now we state the propositions:

(15) Suppose $e_0$ joins $v$ to $w$ in createGraph($e$). Then

(i) $e_0 = e$, and

(ii) $v =$ (the source of $G$)($e$), and

(iii) $w =$ (the target of $G$)($e$).

The theorem is a consequence of (13).

(16) If $e_0$ joins $v$ and $w$ in createGraph($e$), then $e_0 = e$. The theorem is a consequence of (15).

Let $G$ be a non edgeless graph and $e$ be an edge of $G$. One can check that createGraph($e$) is non edgeless, non-multi, connected, and finite. Let us consider a non edgeless graph $G$ and an edge $e$ of $G$. Now we state the propositions:

(17) createGraph($e$) is loopless if and only if $e \notin G$.loops(). The theorem is a consequence of (14) and (15).

(18) createGraph($e$) is acyclic if and only if $e \notin G$.loops(). The theorem is a consequence of (17), (13), and (16).

(19) createGraph($e$) $\in G$.allSG().

(20) Let us consider a non edgeless graph $G$, an edge $e$ of $G$, and a subgraph $H$ of $G$ induced by $\{$(the source of $G)(e)$, (the target of $G)(e)\}$ and $\{e\}$. Then $H \approx$ createGraph$(e)$. The theorem is a consequence of (13).

(21) Let us consider a non edgeless graph $G$, an edge $e$ of $G$, and a subset $V$ of the vertices of $G$. Then every supergraph of createGraph$(e)$ extended by the vertices from $V$ is a subgraph of $G$.

(22) Let us consider an edgeless graph $G$, a graph union set $S$, and a graph union $G'$ of $S$. Suppose for every vertex $v$ of $G$, there exists an element $H'$ of $S$ such that $v \in$ the vertices of $H'$. Then $G$ is a subgraph of $G'$.

(23) Let us consider a non edgeless graph $G$, a graph union set $S$, and a graph union $G'$ of $S$. Suppose for every vertex $v$ of $G$, there exists an element $H'$ of $S$ such that $v \in$ the vertices of $H'$ and for every edge $e$ of $G$, there exists an element $H'$ of $S$ such that createGraph$(e)$ is a subgraph of $H'$. Then $G$ is a subgraph of $G'$. The theorem is a consequence of (13).

(24) Let us consider an edgeless graph $G$, a graph union set $S$, and a graph union $G'$ of $S$. Suppose for every vertex $v$ of $G$, createGraph$(v) \in S$. Then $G$ is a subgraph of $G'$. The theorem is a consequence of (22).

(25) Let us consider a non edgeless graph $G$, a graph union set $S$, and a graph union $G'$ of $S$. Suppose for every vertex $v$ of $G$, createGraph$(v) \in S$ and for every edge $e$ of $G$, createGraph$(e) \in S$. Then $G$ is a subgraph of $G'$. The theorem is a consequence of (23).

(26) Let us consider a non edgeless graph $G$, a set $E$, an edge $e$ of $G$, and a subgraph $H$ of $G$ with edges $E$ removed. If $e \notin E$, then createGraph$(e)$ is a subgraph of $H$. The theorem is a consequence of (13).

Let us consider a non edgeless graph $G$, a subgraph $H$ of $G$ with loops removed, a graph union set $S$, and a graph union $G'$ of $S$. Now we state the propositions:

(27) Suppose for every vertex $v$ of $G$, there exists an element $H'$ of $S$ such that $v \in$ the vertices of $H'$ and for every edge $e$ of $G$ such that $e \notin G$.loops() there exists an element $H'$ of $S$ such that createGraph$(e)$ is a subgraph of $H'$. Then $H$ is a subgraph of $G'$. The theorem is a consequence of (13) and (26).

(28) Suppose for every vertex $v$ of $G$, createGraph$(v) \in S$ and for every edge $e$ of $G$ such that $e \notin G$.loops() holds createGraph$(e) \in S$. Then $H$ is a subgraph of $G'$. The theorem is a consequence of (27).

Let us consider $G$. Let us observe that $G$.allSG() is non empty, $\cup$-tolerating, and plain. Let $S$ be a non empty subset of $G$.allSG(). Let us observe that an element of $S$ is a subgraph of $G$. Now we state the propositions:

(29) $G_2$.allSG() $\subseteq$ $G_1$.allSG() if and only if $G_2$ is a subgraph of $G_1$. The theorem is a consequence of (3) and (1).

(30) $G_1 \approx G_2$ if and only if $G_1$.allSG() = $G_2$.allSG(). The theorem is a consequence of (29).

Let us consider $G_1$ and $G_2$. Let $F$ be a partial graph mapping from $G_1$ to $G_2$. The functor SG2SGFunc($F$) yielding a function from $G_1$.allSG() into $G_2$.allSG() is defined by

(Def. 5) for every plain subgraph $H$ of $G_1$, $it(H) = \mathrm{rng}(F{\restriction}H)$.

One can verify that SG2SGFunc($F$) is non empty and graph-yielding and dom(SG2SGFunc($F$)) is graph-membered and dom(SG2SGFunc($F$)) is plain.

Now we state the proposition:

(31) Let us consider a partial graph mapping $F$ from $G_1$ to $G_2$. If $F$ is weak subgraph embedding, then SG2SGFunc($F$) is one-to-one. The theorem is a consequence of (1).

Let $G_1$ be a graph, $G_2$ be a $G_1$-isomorphic graph, and $F$ be an isomorphism between $G_1$ and $G_2$. Let us observe that SG2SGFunc($F$) is one-to-one. Now we state the propositions:

(32) Let us consider a partial graph mapping $F$ from $G_1$ to $G_2$. Suppose $F$ is onto. Then rng SG2SGFunc($F$) = $G_2$.allSG(). The theorem is a consequence of (1).

(33) If $G_2$ is $G_1$-directed-isomorphic, then $G_1$.allSG() and $G_2$.allSG() are directed-isomorphic. The theorem is a consequence of (32), (31), and (1).

(34) If $G_2$ is $G_1$-isomorphic, then $G_1$.allSG() and $G_2$.allSG() are isomorphic. The theorem is a consequence of (32), (31), and (1).

(35) $G$ is a graph union of $G$.allSG(). The theorem is a consequence of (3) and (1).

(36) (i) $G$ is loopless iff $G$.allSG() is loopless, and

(ii) $G$ is non-multi iff $G$.allSG() is non-multi, and

(iii) $G$ is non-directed-multi iff $G$.allSG() is non-directed-multi, and

(iv) $G$ is simple iff $G$.allSG() is simple, and

(v) $G$ is directed-simple iff $G$.allSG() is directed-simple, and

(vi) $G$ is acyclic iff $G$.allSG() is acyclic, and

(vii) $G$ is edgeless iff $G$.allSG() is edgeless.

Let $G$ be a loopless graph. Observe that $G$.allSG() is loopless. Let $G$ be a non-multi graph. Let us observe that $G$.allSG() is non-multi. Let $G$ be a non-directed-multi graph. One can verify that $G$.allSG() is non-directed-multi. Let $G$ be a simple graph. One can check that $G$.allSG() is simple.

Let $G$ be a directed-simple graph. Let us note that $G$.allSG() is directed-simple. Let $G$ be an acyclic graph. Let us observe that $G$.allSG() is acyclic. Let $G$ be an edgeless graph. One can verify that $G$.allSG() is edgeless. Now we state the propositions:

(37) The vertices of $G$.allSG() $= 2^{\alpha} \setminus \{\emptyset\}$, where $\alpha$ is the vertices of $G$. The theorem is a consequence of (1).

(38) The edges of $G$.allSG() $= 2^{\alpha}$, where $\alpha$ is the edges of $G$. The theorem is a consequence of (1).

Let us consider $G$. The functor SubgraphRel($G$) yielding a binary relation on $G$.allSG() is defined by

(Def. 6) for every elements $H_1$, $H_2$ of $G$.allSG(), $\langle H_1, H_2 \rangle \in$ *it* iff $H_1$ is a subgraph of $H_2$.

Now we state the propositions:

(39) $\langle H \restriction$(the graph selectors), $G \restriction$(the graph selectors)$\rangle \in$ SubgraphRel($G$). The theorem is a consequence of (2) and (3).

(40) field SubgraphRel($G$) $=$ $G$.allSG().
Proof: $G$.allSG() $\subseteq$ field SubgraphRel($G$). □

(41) SubgraphRel($G$) partially orders $G$.allSG().

Let us consider $G$. One can verify that SubgraphRel($G$) is reflexive, antisymmetric, transitive, and partial-order. Now we state the propositions:

(42) $G \restriction$(the graph selectors) is maximal in SubgraphRel($G$). The theorem is a consequence of (3), (40), (1), and (39).

(43) SubgraphRel($H$) $=$ SubgraphRel($G$) $|^2$ $H$.allSG(). The theorem is a consequence of (29) and (40).

(44) Let us consider a non empty subset $S$ of $G$.allSG(), and a graph union $G'$ of $S$. Suppose SubgraphRel($G$) $|^2$ $S$ is a linear order. Let us consider a walk $W$ of $G'$. Then there exists an element $H$ of $S$ such that $W$ is a walk of $H$.
Proof: Define $\mathcal{P}$[walk of $G'$] $\equiv$ there exists an element $H$ of $S$ such that $\$_1$ is a walk of $H$. For every trivial walk $W$ of $G'$, $\mathcal{P}[W]$. For every walk $W$ of $G'$ and for every object $e$ such that $e \in W$.last().edgesInOut() and $\mathcal{P}[W]$ holds $\mathcal{P}[W$.addEdge$(e)]$. For every walk $W$ of $G'$, $\mathcal{P}[W]$. □

## 2. Induced Subgraph Set

Let us consider $G$. The functor $G$.allInducedSG() yielding a subset of $G$.allSG() is defined by the term

(Def. 7) the set of all the plain subgraph of $G$ induced by $V$ where $V$ is a non empty subset of the vertices of $G$.

Now we state the proposition:

(45) $G_2 \in G_1$.allInducedSG() if and only if there exists a non empty subset $V$ of the vertices of $G_1$ such that $G_2$ is a plain subgraph of $G_1$ induced by $V$.

Let $G$ be a vertex-finite graph. Observe that $G$.allInducedSG() is finite. Now we state the propositions:

(46) Let us consider a non empty subset $V$ of the vertices of $G$, and a subgraph $H$ of $G$ induced by $V$. Then $H\restriction$(the graph selectors) $\in G$.allInducedSG(). The theorem is a consequence of (45).

(47) $G\restriction$(the graph selectors) $\in G$.allInducedSG(). The theorem is a consequence of (46).

Let us consider $G$. Observe that $G$.allInducedSG() is non empty, $\cup$-tolerating, and plain. Now we state the propositions:

(48) $G_2$.allInducedSG() $\subseteq G_1$.allInducedSG() if and only if there exists a non empty subset $V$ of the vertices of $G_1$ such that $G_2$ is a subgraph of $G_1$ induced by $V$. The theorem is a consequence of (47) and (45).

(49) $G_1 \approx G_2$ if and only if $G_1$.allInducedSG() $=$ $G_2$.allInducedSG(). The theorem is a consequence of (48).

Let us consider a partial graph mapping $F$ from $G_1$ to $G_2$. Now we state the propositions:

(50) If $F$ is total and onto, then $G_2$.allInducedSG() $\subseteq$ rng(SG2SGFunc($F$) $\restriction G_1$.allInducedSG()). The theorem is a consequence of (49).

(51) If $F$ is total and continuous, then rng(SG2SGFunc($F$)$\restriction G_1$.allInducedSG()) $\subseteq G_2$.allInducedSG(). The theorem is a consequence of (45).

(52) If $F$ is isomorphism, then rng(SG2SGFunc($F$)$\restriction G_1$.allInducedSG()) $=$ $G_2$.allInducedSG(). The theorem is a consequence of (50) and (51).

(53) If $G_2$ is $G_1$-directed-isomorphic, then $G_1$.allInducedSG() and $G_2$.allInducedSG() are directed-isomorphic. The theorem is a consequence of (52), (31), and (45).

(54) If $G_2$ is $G_1$-isomorphic, then $G_1$.allInducedSG() and $G_2$.allInducedSG() are isomorphic. The theorem is a consequence of (52), (31), and (45).

(55) $G$ is a graph union of $G$.allInducedSG(). The theorem is a consequence of (47).

(56) (i) $G$ is loopless iff $G$.allInducedSG() is loopless, and

(ii) $G$ is non-multi iff $G$.allInducedSG() is non-multi, and

(iii) $G$ is non-directed-multi iff $G$.allInducedSG() is non-directed-multi, and

(iv) $G$ is simple iff $G$.allInducedSG() is simple, and

(v) $G$ is directed-simple iff $G$.allInducedSG() is directed-simple, and

(vi) $G$ is acyclic iff $G$.allInducedSG() is acyclic, and

(vii) $G$ is edgeless iff $G$.allInducedSG() is edgeless, and

(viii) $G$ is chordal iff $G$.allInducedSG() is chordal, and

(ix) $G$ is loopfull iff $G$.allInducedSG() is loopfull.

Let $G$ be a loopless graph. One can verify that $G$.allInducedSG() is loopless. Let $G$ be a non-multi graph. Note that $G$.allInducedSG() is non-multi. Let $G$ be a non-directed-multi graph. Observe that $G$.allInducedSG() is non-directed-multi. Let $G$ be a simple graph. One can verify that $G$.allInducedSG() is simple. Let $G$ be a directed-simple graph. Note that $G$.allInducedSG() is directed-simple. Let $G$ be an acyclic graph. Observe that $G$.allInducedSG() is acyclic. Let $G$ be an edgeless graph. One can verify that $G$.allInducedSG() is edgeless. Let $G$ be a chordal graph. Note that $G$.allInducedSG() is chordal. Let $G$ be a loopfull graph. Let us note that $G$.allInducedSG() is loopfull. Now we state the propositions:

(57) $G$ is edgeless if and only if $G$.allInducedSG() = the set of all createGraph ($V$) where $V$ is a non empty subset of the vertices of $G$. The theorem is a consequence of (9), (45), and (47).

(58) $G$ is edgeless if and only if $G$.allSG() = $G$.allInducedSG(). The theorem is a consequence of (9), (57), and (45).

(59) The vertices of $G$.allInducedSG() $= 2^{\alpha} \setminus \{\emptyset\}$, where $\alpha$ is the vertices of $G$. The theorem is a consequence of (37).

## 3. Spanning Subgraph Set

Let us consider $G$. The functor $G$.allSpanningSG() yielding a subset of $G$.allSG() is defined by the term

(Def. 8) $\{H$, where $H$ is an element of $\Omega_{G.\mathrm{allSG}()} : H$ is spanning$\}$.

We introduce the notation $G$.allFactors() as a synonym of $G$.allSpanningSG().

Now we state the propositions:

(60) $G_2 \in G_1$.allSpanningSG() if and only if $G_2$ is a plain, spanning subgraph of $G_1$. The theorem is a consequence of (1).

(61) Let us consider a spanning subgraph $H$ of $G$. Then $H\restriction$(the graph selectors) $\in G$.allSpanningSG(). The theorem is a consequence of (60).

(62) $G\restriction$(the graph selectors) $\in G$.allSpanningSG(). The theorem is a consequence of (61).

(63) createGraph($\Omega_\alpha$) $\in G$.allSpanningSG(), where $\alpha$ is the vertices of $G$. The theorem is a consequence of (60).

(64) Let us consider a non edgeless graph $G$, an edge $e$ of $G$, and a plain supergraph $H$ of createGraph($e$) extended by the vertices from the vertices of $G$. Then $H \in G$.allSpanningSG(). The theorem is a consequence of (21) and (60).

Let $G$ be a graph. Let us note that $G$.allSpanningSG() is non empty, $\cup$-tolerating, and plain. Now we state the propositions:

(65) $G_2$.allSpanningSG() $\subseteq G_1$.allSpanningSG() if and only if $G_2$ is a spanning subgraph of $G_1$. The theorem is a consequence of (62) and (60).

(66) $G_1 \approx G_2$ if and only if $G_1$.allSpanningSG() $= G_2$.allSpanningSG(). The theorem is a consequence of (65).

Let us consider a partial graph mapping $F$ from $G_1$ to $G_2$. Now we state the propositions:

(67) Suppose $\operatorname{rng} F_{\mathbb{V}} =$ the vertices of $G_2$.
Then rng(SG2SGFunc($F$)$\restriction G_1$.allSpanningSG()) $\subseteq G_2$.allSpanningSG().

(68) Suppose $F$ is onto and $F_{\mathbb{V}}$ is one-to-one and total.
Then rng(SG2SGFunc($F$)$\restriction G_1$.allSpanningSG()) $= G_2$.allSpanningSG(). The theorem is a consequence of (67), (32), (1), and (60).

(69) If $F$ is isomorphism, then rng(SG2SGFunc($F$)$\restriction G_1$.allSpanningSG()) $=$ $G_2$.allSpanningSG(). The theorem is a consequence of (68).

(70) If $G_2$ is $G_1$-directed-isomorphic, then $G_1$.allSpanningSG() and $G_2$.allSpanningSG() are directed-isomorphic. The theorem is a consequence of (69), (31), and (60).

(71) If $G_2$ is $G_1$-isomorphic, then $G_1$.allSpanningSG() and $G_2$.allSpanningSG() are isomorphic. The theorem is a consequence of (69), (31), and (60).

(72) $G$ is a graph union of $G$.allSpanningSG(). The theorem is a consequence of (62).

(73) (i) $G$ is loopless iff $G$.allSpanningSG() is loopless, and

(ii) $G$ is non-multi iff $G$.allSpanningSG() is non-multi, and

(iii) $G$ is non-directed-multi iff $G$.allSpanningSG() is non-directed-multi, and

(iv) $G$ is simple iff $G$.allSpanningSG() is simple, and

(v) $G$ is directed-simple iff $G$.allSpanningSG() is directed-simple, and

(vi) $G$ is acyclic iff $G$.allSpanningSG() is acyclic, and

(vii) $G$ is edgeless iff $G$.allSpanningSG() is edgeless.

Let $G$ be a loopless graph. Note that $G$.allSpanningSG() is loopless. Let $G$ be a non-multi graph. Observe that $G$.allSpanningSG() is non-multi. Let $G$ be a non-directed-multi graph. One can verify that $G$.allSpanningSG() is non-directed-multi. Let $G$ be a simple graph. Note that $G$.allSpanningSG() is simple.

Let $G$ be a directed-simple graph. Observe that $G$.allSpanningSG() is directed-simple. Let $G$ be an acyclic graph. One can verify that $G$.allSpanningSG() is acyclic. Let $G$ be an edgeless graph. Note that $G$.allSpanningSG() is edgeless. Now we state the propositions:

(74) $G$ is edgeless if and only if $G$.allSpanningSG() $= \{G\restriction$(the graph selectors)$\}$. The theorem is a consequence of (60) and (62).

(75) The vertices of $G$.allSpanningSG() $=$ {the vertices of $G$}. The theorem is a consequence of (60).

(76) The edges of $G$.allSpanningSG() $= 2^{\alpha}$, where $\alpha$ is the edges of $G$. The theorem is a consequence of (38) and (60).

(77) $G$.allInducedSG() $\cap G$.allSpanningSG() $= \{G\restriction$(the graph selectors)$\}$. The theorem is a consequence of (45), (60), (47), and (62).

## 4. Forest Subgraph Set

Let us consider $G$. The functor $G$.allForests() yielding a subset of $G$.allSG() is defined by the term

(Def. 9) $\{H$, where $H$ is an element of $\Omega_{G.\mathrm{allSG}()}$ : $H$ is acyclic$\}$.

Now we state the propositions:

(78) $G_2 \in G_1$.allForests() if and only if $G_2$ is a plain, acyclic subgraph of $G_1$. The theorem is a consequence of (1).

(79) Let us consider an acyclic subgraph $H$ of $G$. Then $H\restriction$(the graph selectors) $\in G$.allForests(). The theorem is a consequence of (78).

(80) $G$ is acyclic if and only if $G\restriction$(the graph selectors) $\in G$.allForests(). The theorem is a consequence of (79) and (78).

(81) Let us consider a non empty subset $V$ of the vertices of $G$. Then createGraph($V$) $\in G$.allForests().

(82) Let us consider a vertex $v$ of $G$. Then createGraph($v$) $\in G$.allForests().

(83) Let us consider a non edgeless graph $G$, and an edge $e$ of $G$. Suppose $e \notin G$.loops(). Then createGraph($e$) $\in G$.allForests(). The theorem is a consequence of (18) and (78).

(84) Let us consider a non edgeless graph $G$, an edge $e$ of $G$, a subset $V$ of the vertices of $G$, and a plain supergraph $H$ of createGraph($e$) extended by the vertices from $V$. If $e \notin G$.loops(), then $H \in G$.allForests(). The theorem is a consequence of (18), (21), and (78).

Let us consider $G$. Let us note that $G$.allForests() is non empty, $\cup$-tolerating, plain, acyclic, and simple. Now we state the propositions:

(85) $H$.allForests() $\subseteq G$.allForests(). The theorem is a consequence of (78).

(86) Let us consider a loopless graph $G_2$.
Suppose $G_2$.allForests() $\subseteq G_1$.allForests(). Then $G_2$ is a subgraph of $G_1$.
Proof: The edges of $G_2 \subseteq$ the edges of $G_1$. $\square$

(87) Let us consider a subgraph $H$ of $G$ with loops removed.
Then $G$.allForests() $=$ $H$.allForests(). The theorem is a consequence of (85) and (78).

(88) Let us consider loopless graphs $G_1$, $G_2$. Then $G_1 \approx G_2$ if and only if $G_1$.allForests() $=$ $G_2$.allForests(). The theorem is a consequence of (87) and (86).

(89) Let us consider a subgraph $G_3$ of $G_1$ with loops removed, and a subgraph $G_4$ of $G_2$ with loops removed. Then $G_3 \approx G_4$ if and only if $G_1$.allForests() $=$ $G_2$.allForests(). The theorem is a consequence of (87) and (88).

Let us consider a partial graph mapping $F$ from $G_1$ to $G_2$. Now we state the propositions:

(90) If $F$ is weak subgraph embedding,
then rng(SG2SGFunc($F$)$\restriction G_1$.allForests()) $\subseteq G_2$.allForests(). The theorem is a consequence of (78) and (1).

(91) If $F$ is one-to-one and onto, then $G_2$.allForests() $\subseteq$ rng(SG2SGFunc($F$) $\restriction G_1$.allForests()). The theorem is a consequence of (78).

(92) If $F$ is isomorphism, then $G_2$.allForests() $=$ rng(SG2SGFunc($F$) $\restriction G_1$.allForests()). The theorem is a consequence of (90) and (91).

(93) If $G_2$ is $G_1$-directed-isomorphic, then $G_1$.allForests() and $G_2$.allForests() are directed-isomorphic. The theorem is a consequence of (92), (31), and (78).

(94) If $G_2$ is $G_1$-isomorphic, then $G_1$.allForests() and $G_2$.allForests() are isomorphic. The theorem is a consequence of (92), (31), and (78).

Let us consider a subgraph $G_3$ of $G_1$ with loops removed and a subgraph $G_4$ of $G_2$ with loops removed. Now we state the propositions:

(95) If $G_4$ is $G_3$-directed-isomorphic, then $G_1$.allForests() and $G_2$.allForests() are directed-isomorphic. The theorem is a consequence of (87) and (93).

(96) If $G_4$ is $G_3$-isomorphic, then $G_1$.allForests() and $G_2$.allForests() are isomorphic. The theorem is a consequence of (87) and (94).

(97) Every subgraph of $G$ with loops removed is a graph union of $G$.allForests(). The theorem is a consequence of (35), (82), (83), (13), (87), and (78).

(98) $G$ is loopless if and only if $G$ is a graph union of $G$.allForests(). The theorem is a consequence of (97).

(99) The edges of $G = G$.loops() if and only if $G$.allForests() is edgeless. The theorem is a consequence of (78) and (83).

(100) The edges of $G = G$.loops() if and only if $G$.allForests() = the set of all createGraph($V$) where $V$ is a non empty subset of the vertices of $G$. The theorem is a consequence of (99), (78), and (81).

(101) The vertices of $G$.allForests() $= 2^{\alpha} \setminus \{\emptyset\}$, where $\alpha$ is the vertices of $G$. The theorem is a consequence of (37) and (81).

## 5. Spanning Forest Subgraph Set

Let us consider $G$. The functor $G$.allSpanningForests() yielding a subset of $G$.allSG() is defined by the term

(Def. 10) $\{H$, where $H$ is an element of $\Omega_{G.\mathrm{allSG}()} : H$ is spanning and acyclic$\}$.

Now we state the propositions:

(102) $G_2 \in G_1$.allSpanningForests() if and only if $G_2$ is a plain, spanning, acyclic subgraph of $G_1$. The theorem is a consequence of (1).

(103) $G$.allSpanningForests() $=$ $G$.allSpanningSG() $\cap G$.allForests(). The theorem is a consequence of (102), (60), and (78).

(104) Let us consider a spanning, acyclic subgraph $H$ of $G$. Then $H\restriction$(the graph selectors) $\in$ $G$.allSpanningForests(). The theorem is a consequence of (102).

(105) $G$ is acyclic if and only if $G\restriction$(the graph selectors) $\in G$.allSpanningForests(). The theorem is a consequence of (103), (80), and (62).

(106) createGraph($\Omega_{\alpha}$) $\in G$.allSpanningForests(), where $\alpha$ is the vertices of $G$. The theorem is a consequence of (81), (63), and (103).

(107) Let us consider a non edgeless graph $G$, an edge $e$ of $G$, and a plain supergraph $H$ of createGraph($e$) extended by the vertices from the vertices of $G$. If $e \notin G$.loops(), then $H \in G$.allSpanningForests(). The theorem is a consequence of (64), (84), and (103).

Let us consider $G$. One can check that $G$.allSpanningForests() is non empty, $\cup$-tolerating, plain, acyclic, and simple. Now we state the propositions:

(108) Let us consider a spanning subgraph $H$ of $G$. Then $H$.allSpanningForests() $\subseteq$ $G$.allSpanningForests(). The theorem is a consequence of (102).

(109) Let us consider a loopless graph $G_2$. Suppose $G_2$.allSpanningForests() $\subseteq$ $G_1$.allSpanningForests(). Then $G_2$ is a spanning subgraph of $G_1$. The theorem is a consequence of (102), (107), and (13).

(110) Let us consider a subgraph $H$ of $G$ with loops removed. Then $G$.allSpanningForests() $=$ $H$.allSpanningForests(). The theorem is a consequence of (108) and (102).

(111) Let us consider loopless graphs $G_1$, $G_2$. Then $G_1 \approx G_2$ if and only if $G_1$.allSpanningForests() $=$ $G_2$.allSpanningForests(). The theorem is a consequence of (110) and (109).

(112) Let us consider a subgraph $G_3$ of $G_1$ with loops removed, and a subgraph $G_4$ of $G_2$ with loops removed. Then $G_3 \approx G_4$ if and only if $G_1$.allSpanningForests() $=$ $G_2$.allSpanningForests(). The theorem is a consequence of (110) and (111).

Let us consider a partial graph mapping $F$ from $G_1$ to $G_2$. Now we state the propositions:

(113) Suppose $F$ is weak subgraph embedding and rng $F_{\mathbb{V}}$ = the vertices of $G_2$. Then rng(SG2SGFunc($F$)$\restriction G_1$.allSpanningForests()) $\subseteq$ $G_2$.allSpanning Forests(). The theorem is a consequence of (67), (90), and (103).

(114) Suppose $F$ is weak subgraph embedding and onto. Then $G_2$.allSpanningForests() = rng(SG2SGFunc($F$)$\restriction G_1$.allSpanning Forests()). The theorem is a consequence of (113), (68), (91), (103), and (31).

Let us consider graphs $G_1$, $G_2$. Now we state the propositions:

(115) If $G_2$ is $G_1$-directed-isomorphic, then $G_1$.allSpanningForests() and $G_2$.allSpanningForests() are directed-isomorphic. The theorem is a consequence of (114), (31), and (102).

(116) If $G_2$ is $G_1$-isomorphic, then $G_1$.allSpanningForests() and $G_2$.allSpanningForests() are isomorphic. The theorem is a consequence of (114), (31), and (102).

Let us consider a subgraph $G_3$ of $G_1$ with loops removed and a subgraph $G_4$ of $G_2$ with loops removed. Now we state the propositions:

(117) If $G_4$ is $G_3$-directed-isomorphic, then $G_1$.allSpanningForests() and $G_2$.allSpanningForests() are directed-isomorphic. The theorem is a consequence of (110) and (115).

(118) If $G_4$ is $G_3$-isomorphic, then $G_1$.allSpanningForests() and $G_2$.allSpanningForests() are isomorphic. The theorem is a consequence of (110) and (116).

(119) Every subgraph of $G$ with loops removed is a graph union of $G$.allSpanningForests(). The theorem is a consequence of (35), (106), (107), (13), (110), and (102).

(120) $G$ is loopless if and only if $G$ is a graph union of $G$.allSpanningForests(). The theorem is a consequence of (119).

(121) The edges of $G = G$.loops() if and only if $G$.allSpanningForests() is edgeless. The theorem is a consequence of (99), (103), and (107).

(122) The edges of $G = G$.loops() if and only if for every subgraph $H$ of $G$ with loops removed, $G$.allSpanningForests() $= \{H \restriction$(the graph selectors)$\}$. The theorem is a consequence of (102) and (104).

(123) The vertices of $G$.allSpanningForests() $= \{$the vertices of $G\}$. The theorem is a consequence of (103) and (75).

## 6. Connected Subgraph Set

Let us consider $G$. The functor $G$.allConnectedSG() yielding a subset of $G$.allSG() is defined by the term

(Def. 11) $\{H$, where $H$ is an element of $\Omega_{G.\mathrm{allSG}()}$ : $H$ is connected$\}$.

Now we state the propositions:

(124) $G_2 \in G_1$.allConnectedSG() if and only if $G_2$ is a plain, connected subgraph of $G_1$. The theorem is a consequence of (1).

(125) Let us consider a connected subgraph $H$ of $G$. Then $H \restriction$(the graph selectors) $\in G$.allConnectedSG(). The theorem is a consequence of (124).

(126) $G$ is connected if and only if $G \restriction$(the graph selectors) $\in$ $G$.allConnectedSG(). The theorem is a consequence of (125) and (124).

(127) Let us consider a vertex $v$ of $G$. Then createGraph($v$) $\in G$.allConnectedSG().

(128) Let us consider a non edgeless graph $G$, and an edge $e$ of $G$. Then createGraph($e$) $\in G$.allConnectedSG().

Let us consider $G$. One can check that $G$.allConnectedSG() is non empty, $\cup$-tolerating, plain, and connected. Now we state the propositions:

(129) $H$.allConnectedSG() $\subseteq G$.allConnectedSG(). The theorem is a consequence of (124).

(130) If $G_2$.allConnectedSG() $\subseteq G_1$.allConnectedSG(), then $G_2$ is a subgraph of $G_1$.

Proof: The edges of $G_2 \subseteq$ the edges of $G_1$. $\square$

(131) $G_1 \approx G_2$ if and only if $G_1$.allConnectedSG() $=$ $G_2$.allConnectedSG(). The theorem is a consequence of (129) and (130).

Let us consider a partial graph mapping $F$ from $G_1$ to $G_2$. Now we state the propositions:

(132) If $F$ is total, then rng(SG2SGFunc($F$)$\restriction G_1$.allConnectedSG()) $\subseteq$ $G_2$.allConnectedSG(). The theorem is a consequence of (124) and (1).

(133) If $F$ is one-to-one and onto, then $G_2$.allConnectedSG() $\subseteq$ rng(SG2SGFunc($F$)$\restriction G_1$.allConnectedSG()). The theorem is a consequence of (124).

(134) If $F$ is isomorphism, then $G_2$.allConnectedSG() $=$ rng(SG2SGFunc($F$) $\restriction G_1$.allConnectedSG()). The theorem is a consequence of (132) and (133).

(135) If $G_2$ is $G_1$-directed-isomorphic, then $G_1$.allConnectedSG() and $G_2$.allConnectedSG() are directed-isomorphic. The theorem is a consequence of (134), (31), and (124).

(136) If $G_2$ is $G_1$-isomorphic, then $G_1$.allConnectedSG() and $G_2$.allConnectedSG() are isomorphic. The theorem is a consequence of (134), (31), and (124).

(137) $G$ is a graph union of $G$.allConnectedSG(). The theorem is a consequence of (35), (127), (24), (128), and (25).

## 7. Tree Subgraph Set and Subtree Relation

Let us consider $G$. The functor $G$.allTrees() yielding a subset of $G$.allSG() is defined by the term

(Def. 12) $\{H$, where $H$ is an element of $\Omega_{G.\text{allSG}()}$ : $H$ is tree-like$\}$.

Now we state the propositions:

(138) $G_2 \in G_1$.allTrees() if and only if $G_2$ is a plain, tree-like subgraph of $G_1$. The theorem is a consequence of (1).

(139) $G$.allTrees() $=$ $G$.allForests() $\cap G$.allConnectedSG(). The theorem is a consequence of (138), (78), and (124).

(140) Let us consider a tree-like subgraph $H$ of $G$. Then $H\restriction$(the graph selectors) $\in G$.allTrees(). The theorem is a consequence of (138).

(141) $G$ is tree-like if and only if $G\restriction$(the graph selectors) $\in G$.allTrees(). The theorem is a consequence of (140) and (138).

(142) Let us consider a vertex $v$ of $G$. Then createGraph($v$) $\in G$.allTrees().

(143) Let us consider a non edgeless graph $G$, and an edge $e$ of $G$. Suppose $e \notin G$.loops(). Then createGraph($e$) $\in G$.allTrees(). The theorem is a consequence of (18) and (138).

Let us consider $G$. Observe that $G$.allTrees() is non empty, $\cup$-tolerating, plain, tree-like, and simple. Now we state the propositions:

(144) $H$.allTrees() $\subseteq G$.allTrees(). The theorem is a consequence of (138).

(145) Let us consider a loopless graph $G_2$.
Suppose $G_2$.allTrees() $\subseteq G_1$.allTrees(). Then $G_2$ is a subgraph of $G_1$. The theorem is a consequence of (142), (138), (143), and (13).

(146) Let us consider a subgraph $H$ of $G$ with loops removed. Then $G$.allTrees() $= H$.allTrees(). The theorem is a consequence of (144) and (138).

(147) Let us consider loopless graphs $G_1$, $G_2$. Then $G_1 \approx G_2$ if and only if $G_1$.allTrees() $= G_2$.allTrees(). The theorem is a consequence of (146) and (145).

(148) Let us consider a subgraph $G_3$ of $G_1$ with loops removed, and a subgraph $G_4$ of $G_2$ with loops removed. Then $G_3 \approx G_4$ if and only if $G_1$.allTrees() $=$ $G_2$.allTrees(). The theorem is a consequence of (146) and (147).

Let us consider a partial graph mapping $F$ from $G_1$ to $G_2$. Now we state the propositions:

(149) If $F$ is weak subgraph embedding,
then rng(SG2SGFunc($F$)$\restriction G_1$.allTrees()) $\subseteq G_2$.allTrees(). The theorem is a consequence of (139), (90), and (132).

(150) If $F$ is weak subgraph embedding and onto, then $G_2$.allTrees() $=$ rng(SG2SGFunc($F$)$\restriction G_1$.allTrees()). The theorem is a consequence of (91), (133), (139), (149), and (31).

Let us consider graphs $G_1$, $G_2$. Now we state the propositions:

(151) If $G_2$ is $G_1$-directed-isomorphic, then $G_1$.allTrees() and $G_2$.allTrees() are directed-isomorphic. The theorem is a consequence of (150), (31), and (138).

(152) If $G_2$ is $G_1$-isomorphic, then $G_1$.allTrees() and $G_2$.allTrees() are isomorphic. The theorem is a consequence of (150), (31), and (138).

Let us consider a subgraph $G_3$ of $G_1$ with loops removed and a subgraph $G_4$ of $G_2$ with loops removed. Now we state the propositions:

(153) If $G_4$ is $G_3$-directed-isomorphic, then $G_1$.allTrees() and $G_2$.allTrees() are directed-isomorphic. The theorem is a consequence of (146) and (151).

(154) If $G_4$ is $G_3$-isomorphic, then $G_1$.allTrees() and $G_2$.allTrees() are isomorphic. The theorem is a consequence of (146) and (152).

(155) Every subgraph of $G$ with loops removed is a graph union of $G$.allTrees(). The theorem is a consequence of (35), (142), (143), (13), (146), and (138).

(156) $G$ is loopless if and only if $G$ is a graph union of $G$.allTrees(). The theorem is a consequence of (155).

(157) The edges of $G = G$.loops() if and only if $G$.allTrees() is edgeless. The theorem is a consequence of (138) and (143).

(158) The edges of $G = G$.loops() if and only if $G$.allTrees() = the set of all createGraph($v$) where $v$ is a vertex of $G$. The theorem is a consequence of (157), (138), and (142).

Let us consider $G$. The functor SubtreeRel($G$) yielding a binary relation on $G$.allTrees() is defined by the term

(Def. 13) SubgraphRel($G$) $|^2$ $G$.allTrees().

Now we state the propositions:

(159) Let us consider plain, tree-like subgraphs $H_1$, $H_2$ of $G$. Then $\langle H_1, H_2\rangle \in$ SubtreeRel($G$) if and only if $H_1$ is a subgraph of $H_2$. The theorem is a consequence of (1) and (138).

(160) field SubtreeRel($G$) $=$ $G$.allTrees(). The theorem is a consequence of (40).

(161) SubtreeRel($G$) partially orders $G$.allTrees(). The theorem is a consequence of (41) and (160).

Let us consider $G$. Let us observe that SubtreeRel($G$) is reflexive, antisymmetric, transitive, and partial-order. Now we state the propositions:

(162) SubtreeRel($H$) = SubtreeRel($G$) $|^2$ $H$.allTrees(). The theorem is a consequence of (43) and (144).

(163) Let us consider a loopless graph $G$. Then $G$ is edgeless if and only if SubtreeRel($G$) $=$ $\mathrm{id}_{G.\mathrm{allTrees}()}$. The theorem is a consequence of (160), (138), (159), (143), and (13).

(164) Let us consider a subgraph $H$ of $G$ with loops removed. Then SubtreeRel($G$) = SubtreeRel($H$). The theorem is a consequence of (146) and (162).

(165) The edges of $G = G$.loops() if and only if SubtreeRel($G$) $= \mathrm{id}_{G.\mathrm{allTrees}()}$. The theorem is a consequence of (164), (163), and (146).

(166) $G$.allTrees() has the upper Zorn property w.r.t. SubtreeRel($G$). The theorem is a consequence of (160), (159), (44), (35), and (138).

Let $G$ be a connected graph.

EVERY CONNECTED GRAPH HAS A SPANNING TREE: there exists a subgraph of $G$ which is plain, spanning, and tree-like.

Now we state the proposition:

(167) Let us consider a connected graph $G$, and an object $e$. Suppose $e \in$ (the edges of $G$) $\setminus$ ($G$.loops()). Then there exists a plain, spanning, tree-like subgraph $T$ of $G$ such that $e \in$ the edges of $T$.

## 8. Spanning Tree Subgraph Set

Let us consider $G$. The functor $G$.allSpanningTrees() yielding a subset of $G$.allSG() is defined by the term

(Def. 14) $\{H$, where $H$ is an element of $\Omega_{G.\mathrm{allSG}()}$ : $H$ is spanning and tree-like$\}$.

Now we state the propositions:

(168) $G_2 \in G_1$.allSpanningTrees() if and only if $G_2$ is plain, spanning, acyclic subgraph of $G_1$ and connected. The theorem is a consequence of (1).

(169) $G$.allSpanningTrees() $=$ $G$.allSpanningSG() $\cap G$.allTrees(). The theorem is a consequence of (168), (60), and (138).

(170) $G$.allSpanningTrees() $=$ $G$.allConnectedSG() $\cap G$.allSpanningForests(). The theorem is a consequence of (168), (102), and (124).

(171) Let us consider a spanning, acyclic subgraph $H$ of $G$. Suppose $H$ is connected. Then $H \restriction$(the graph selectors) $\in G$.allSpanningTrees(). The theorem is a consequence of (168).

(172) $G$ is tree-like if and only if $G \restriction$(the graph selectors) $\in G$.allSpanningTrees(). The theorem is a consequence of (169), (141), and (62).

(173) $G$ is connected if and only if $G$.allSpanningTrees() $\neq \emptyset$. The theorem is a consequence of (168).

Let $G$ be a non connected graph. Let us note that $G$.allSpanningTrees() is empty. Let $G$ be a connected graph. Observe that $G$.allSpanningTrees() is non empty, tree-like, and simple. Now we state the propositions:

(174) Let us consider a connected graph $G$, and a connected, spanning subgraph $H$ of $G$. Then $H$.allSpanningTrees() $\subseteq G$.allSpanningTrees(). The theorem is a consequence of (168).

(175) Let us consider a loopless, connected graph $G_2$. Suppose $G_2$.allSpanning Trees() $\subseteq G_1$.allSpanningTrees(). Then $G_2$ is a spanning subgraph of $G_1$. The theorem is a consequence of (168) and (167).

(176) Let us consider a subgraph $H$ of $G$ with loops removed. Then $G$.allSpanningTrees() $=$ $H$.allSpanningTrees(). The theorem is a consequence of (174) and (168).

(177) Let us consider loopless, connected graphs $G_1$, $G_2$. Then $G_1 \approx G_2$ if and only if $G_1$.allSpanningTrees() $=$ $G_2$.allSpanningTrees(). The theorem is a consequence of (176) and (175).

(178) Let us consider connected graphs $G_1$, $G_2$, a subgraph $G_3$ of $G_1$ with loops removed, and a subgraph $G_4$ of $G_2$ with loops removed. Then $G_3 \approx G_4$ if and only if $G_1$.allSpanningTrees() = $G_2$.allSpanningTrees(). The theorem is a consequence of (176) and (177).

Let us consider a partial graph mapping $F$ from $G_1$ to $G_2$. Now we state the propositions:

(179) Suppose $F$ is weak subgraph embedding and rng $F_{\mathbb{V}}$ = the vertices of $G_2$. Then rng(SG2SGFunc($F$)$\restriction G_1$.allSpanningTrees()) $\subseteq G_2$.allSpanning Trees(). The theorem is a consequence of (132), (113), and (170).

(180) Suppose $F$ is weak subgraph embedding and onto. Then $G_2$.allSpanning Trees() = rng(SG2SGFunc($F$)$\restriction G_1$.allSpanningTrees()). The theorem is a consequence of (179), (133), (114), (170), and (31).

(181) If $G_2$ is $G_1$-directed-isomorphic, then $G_1$.allSpanningTrees() and $G_2$.allSpanningTrees() are directed-isomorphic. The theorem is a consequence of (180), (31), and (168).

(182) If $G_2$ is $G_1$-isomorphic, then $G_1$.allSpanningTrees() and $G_2$.allSpanningTrees() are isomorphic. The theorem is a consequence of (180), (31), and (168).

Let us consider a subgraph $G_3$ of $G_1$ with loops removed and a subgraph $G_4$ of $G_2$ with loops removed. Now we state the propositions:

(183) If $G_4$ is $G_3$-directed-isomorphic, then $G_1$.allSpanningTrees() and $G_2$.allSpanningTrees() are directed-isomorphic. The theorem is a consequence of (176) and (181).

(184) If $G_4$ is $G_3$-isomorphic, then $G_1$.allSpanningTrees() and $G_2$.allSpanning Trees() are isomorphic. The theorem is a consequence of (176) and (182).

(185) Let us consider a connected graph $G$. Then every subgraph of $G$ with loops removed is a graph union of $G$.allSpanningTrees(). The theorem is a consequence of (35), (168), (167), and (176).

(186) Every loopless, connected graph is a graph union of $G$.allSpanningTrees(). The theorem is a consequence of (185).

(187) $G$ is tree-like if and only if $G$.allSpanningTrees() = $\{G\restriction$(the graph selectors)$\}$. The theorem is a consequence of (168) and (172).

(188) $G$ is connected if and only if the vertices of $G$.allSpanningTrees() = $\{$the vertices of $G\}$. The theorem is a consequence of (123) and (170).

## 9. Component Subgraph Set

Let us consider $G$. The functor $G$.allComponents() yielding a subset of $G$.allSG() is defined by the term

(Def. 15) $\{H$, where $H$ is an element of $\Omega_{G.\mathrm{allSG}()}$ : $H$ is component-like$\}$.

Now we state the propositions:

(189) $G_2 \in G_1$.allComponents() if and only if $G_2$ is a plain component of $G_1$. The theorem is a consequence of (1).

(190) $G$.allComponents() $\subseteq G$.allInducedSG() $\cap G$.allConnectedSG(). The theorem is a consequence of (189) and (124).

(191) Let us consider a component $H$ of $G$. Then $H{\restriction}$(the graph selectors) $\in G$.allComponents(). The theorem is a consequence of (189).

(192) $G$ is connected if and only if $G{\restriction}$(the graph selectors) $\in G$.allComponents(). The theorem is a consequence of (191) and (189).

Let us consider $G$. Let us observe that $G$.allComponents() is non empty, vertex-disjoint, edge-disjoint, $\cup$-tolerating, plain, and connected. Now we state the propositions:

(193) If $G_2$.allComponents() $\subseteq G_1$.allComponents(), then $G_2$ is a subgraph of $G_1$. The theorem is a consequence of (189).

(194) $G_1 \approx G_2$ if and only if $G_1$.allComponents() $= G_2$.allComponents(). The theorem is a consequence of (189) and (193).

(195) Let us consider a non empty, one-to-one partial graph mapping $F$ from $G_1$ to $G_2$. Suppose $F$ is isomorphism. Then $G_2$.allComponents() $=$ rng(SG2SGFunc($F$)${\restriction}G_1$.allComponents()). The theorem is a consequence of (189).

(196) If $G_2$ is $G_1$-directed-isomorphic, then $G_1$.allComponents() and $G_2$.allComponents() are directed-isomorphic. The theorem is a consequence of (195), (31), and (189).

(197) If $G_2$ is $G_1$-isomorphic, then $G_1$.allComponents() and $G_2$.allComponents() are isomorphic. The theorem is a consequence of (195), (31), and (189).

(198) $G$ is a graph union of $G$.allComponents(). The theorem is a consequence of (35), (189), (22), (14), (13), and (23).

(199) (i) $G$ is loopless iff $G$.allComponents() is loopless, and

(ii) $G$ is non-multi iff $G$.allComponents() is non-multi, and

(iii) $G$ is non-directed-multi iff $G$.allComponents() is non-directed-multi, and

(iv) $G$ is simple iff $G$.allComponents() is simple, and

(v) $G$ is directed-simple iff $G$.allComponents() is directed-simple, and

(vi) $G$ is acyclic iff $G$.allComponents() is acyclic, and

(vii) $G$ is edgeless iff $G$.allComponents() is edgeless, and

(viii) $G$ is chordal iff $G$.allComponents() is chordal, and

(ix) $G$ is loopfull iff $G$.allComponents() is loopfull.

The theorem is a consequence of (198).

Let $G$ be a loopless graph. Observe that $G$.allComponents() is loopless. Let $G$ be a non-multi graph. One can verify that $G$.allComponents() is non-multi. Let $G$ be a non-directed-multi graph. Note that $G$.allComponents() is non-directed-multi. Let $G$ be a simple graph. Observe that $G$.allComponents() is simple. Let $G$ be a directed-simple graph. One can verify that $G$.allComponents() is directed-simple.

Let $G$ be an acyclic graph. Note that $G$.allComponents() is acyclic. Let $G$ be an edgeless graph. Observe that $G$.allComponents() is edgeless. Let $G$ be a chordal graph. One can verify that $G$.allComponents() is chordal. Let $G$ be a loopfull graph. One can check that $G$.allComponents() is loopfull. Now we state the propositions:

(200) $G$ is connected if and only if $G$.allComponents() $= \{G{\restriction}$(the graph selectors)$\}$. The theorem is a consequence of (192) and (189).

(201) The vertices of $G$.allComponents() $=$ $G$.componentSet().

(202) $G$.numComponents() $= \overline{\overline{G.\mathrm{allComponents}()}}$.

PROOF: Define $\mathcal{P}[\mathrm{object}, \mathrm{object}] \equiv$ there exists a plain component $H$ of $G$ such that $\$_1 = H$ and $\$_2 =$ the vertices of $H$. For every object $x$ such that $x \in G$.allComponents() there exists an object $y$ such that $\mathcal{P}[x, y]$. Consider $f$ being a function such that $\mathrm{dom}\, f = G$.allComponents() and for every object $x$ such that $x \in G$.allComponents() holds $\mathcal{P}[x, f(x)]$. □

## REFERENCES


[1] Grzegorz Bancerek, Czesław Byliński, Adam Grabowski, Artur Korniłowicz, Roman Matuszewski, Adam Naumowicz, and Karol Pąk. The role of the Mizar Mathematical Library for interactive proof development in Mizar. *Journal of Automated Reasoning*, 61(1):9–32, 2018. doi:10.1007/s10817-017-9440-6.

[2] John Adrian Bondy and U. S. R. Murty. *Graph Theory*. Graduate Texts in Mathematics, 244. Springer, New York, 2008. ISBN 978-1-84628-969-9.

[3] Ricky W. Butler and Jon A. Sjogren. A PVS graph theory library. Technical report, NASA Langley, 1998.

[4] Ching-Tsun Chou. A formal theory of undirected graphs in higher-order logic. In Thomas F. Melham and Juanito Camilleri, editors, *Higher Order Logic Theorem Proving and Its Applications, 7th International Workshop, Valletta, Malta, September 19–22, 1994, Proceedings*, volume 859 of *Lecture Notes in Computer Science*, pages 144–157. Springer, 1994. doi:10.1007/3-540-58450-1_40.

[5] Reinhard Diestel. *Graph Theory*, volume Graduate Texts in Mathematics; 173. Springer, Berlin, fifth edition, 2017. ISBN 978-3-662-53621-6.

[6] Jessica Enright and Piotr Rudnicki. Helly property for subtrees. *Formalized Mathematics*, 16(**2**):91–96, 2008. doi:10.2478/v10037-008-0013-3.

[7] Adam Grabowski, Artur Korniłowicz, and Adam Naumowicz. Four decades of Mizar. *Journal of Automated Reasoning*, 55(3):191–198, 2015. doi:10.1007/s10817-015-9345-1.

[8] Sebastian Koch. Miscellaneous graph preliminaries. Part I. *Formalized Mathematics*, 29 (**1**):21–38, 2021. doi:10.2478/forma-2021-0003.

[9] Sebastian Koch. Underlying simple graphs. *Formalized Mathematics*, 27(**3**):237–259, 2019. doi:10.2478/forma-2019-0023.

[10] Sebastian Koch. About graph sums. *Formalized Mathematics*, 29(**4**):249–278, 2021. doi:10.2478/forma-2021-0023.

[11] Gilbert Lee and Piotr Rudnicki. Alternative graph structures. *Formalized Mathematics*, 13(**2**):235–252, 2005.

[12] Gilbert Lee and Piotr Rudnicki. Alternative aggregates in Mizar. In Manuel Kauers, Manfred Kerber, Robert Miner, and Wolfgang Windsteiger, editors, *Towards Mechanized Mathematical Assistants*, pages 327–341, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-73086-6. doi:10.1007/978-3-540-73086-6_26.

[13] Lars Noschinski. A graph library for Isabelle. *Mathematics in Computer Science*, 9(1): 23–39, 2015. doi:10.1007/s11786-014-0183-z.

[14] Robin James Wilson. *Introduction to Graph Theory*. Oliver & Boyd, Edinburgh, 1972. ISBN 0-05-002534-1.


*Accepted March 31, 2023*