FORMALIZED MATHEMATICS
Vol. 29, No. 4, Pages 249–278, 2021
DOI: 10.2478/forma-2021-0023

https://sciendo.com/journal/forma

About Graph Sums

Sebastian Koch
Johannes Gutenberg University
Mainz, Germany[1]

Summary. In this article the sum (or disjoint union) of graphs is formalized in the Mizar system [4], [1], based on the formalization of graphs in [9].

MSC: 05C76 68V20

Keywords: graph union; graph sum

MML identifier: `GLIB_015`, version: `8.1.11 5.68.1412`

0. Introduction

The sum of graphs has already been formalized in Mizar to a certain extent in [7], in the case where the vertices and edges of the graphs are disjoint. This disjoint union matches the definitions often given in the literature (cf. [2], [10], [11], [3]). However, graphs are added together most of the time without much concern about what kind of objects actually constitute the vertices and edges. This article's goal is to formalize that practice. Naturally, in this paper the sum is generalized to families of multidigraphs, i.e. the graphs of [9].

The first section introduces functors to replace the concrete objects behind vertices and edges of a graph with other objects, which will later be used in section 5.

In the second section graph selector variants for `Graph-yielding` functions are described in a similar way as it was done for `Graph-membered` sets in section 1 of [7].

[1]The author is enrolled in the Johannes Gutenberg University in Mayence, Germany, mailto: `skoch02@students.uni-mainz.de`

ISSN 1426–2630(Print), 1898-9934(Online)

Isomorphisms between two `Graph-membered` sets or two `Graph-yielding` functions are formalized in section 3. They are the foundation for isomorphisms between unions (section 4) and sums (section 6) of graphs.

Section 4 introduces attributes `vertex-disjoint` and `edge-disjoint` for sets or functions of graphs. A lot of attention is given to graph unions of vertex-disjoint sets of graphs, since these essentially are the graph sums.

The rest of the article then focuses on graph sums, that are vertex-disjoint unions of the range of a function of graphs, which is isomorphic to a given graph function not necessarily vertex-disjoint, so that in future articles authors do not need to create a vertex-disjoint function themselves. This "canonical" distinction function is formalized in section 5. A second distinction function is provided that leaves exactly one graph of the original graph function as it was. Isomorphism theorems between these two distinction functions and the original functions are provided as well and needed for the sum isomorphisms in the next section.

Section 6 introduces the mode `GraphSum` of a (not necessarily vertex-disjoint) graph function as a graph (directed) isomorphic to the union of the range of the distinction function. The second distinction function is used to provide a graph sum that is a supergraph of a given graph in the graph function.

Finally the last section defines the graph sum of two graph as a supergraph of the first graph using the general definition from section 6.

1. Replacing Vertices and Edges

Let G be a graph, V be a non empty, one-to-one many sorted set indexed by the vertices of G, and E be a one-to-one many sorted set indexed by the edges of G. The functor replaceVerticesEdges(V, E) yielding a plain graph is defined by

(Def. 1) there exist functions S, T from $\operatorname{rng} E$ into $\operatorname{rng} V$ such that $S = V \cdot$ (the source of G) $\cdot (E^{-1})$ and $T = V \cdot$ (the target of G) $\cdot (E^{-1})$ and $it =$ createGraph($\operatorname{rng} V, \operatorname{rng} E, S, T$).

The functor replaceVertices(V) yielding a plain graph is defined by the term

(Def. 2) replaceVerticesEdges(V, id_α), where α is the edges of G.

Let E be a one-to-one many sorted set indexed by the edges of G. The functor replaceEdges(E) yielding a plain graph is defined by the term

(Def. 3) replaceVerticesEdges(id_α, E), where α is the vertices of G.

Now we state the propositions:

(1) Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, and a one-to-one many sorted set E indexed

by the edges of G. Then

(i) the vertices of $\mathrm{replaceVerticesEdges}(V, E) = \mathrm{rng}\, V$, and

(ii) the edges of $\mathrm{replaceVerticesEdges}(V, E) = \mathrm{rng}\, E$, and

(iii) the source of $\mathrm{replaceVerticesEdges}(V, E) = V \cdot (\text{the source of } G) \cdot (E^{-1})$, and

(iv) the target of $\mathrm{replaceVerticesEdges}(V, E) = V \cdot (\text{the target of } G) \cdot (E^{-1})$.

(2) Let us consider a graph G, and a non empty, one-to-one many sorted set V indexed by the vertices of G. Then

(i) the vertices of $\mathrm{replaceVertices}(V) = \mathrm{rng}\, V$, and

(ii) the edges of $\mathrm{replaceVertices}(V) =$ the edges of G, and

(iii) the source of $\mathrm{replaceVertices}(V) = V \cdot (\text{the source of } G)$, and

(iv) the target of $\mathrm{replaceVertices}(V) = V \cdot (\text{the target of } G)$.

The theorem is a consequence of (1).

(3) Let us consider a graph G, and a one-to-one many sorted set E indexed by the edges of G. Then

(i) the vertices of $\mathrm{replaceEdges}(E) =$ the vertices of G, and

(ii) the edges of $\mathrm{replaceEdges}(E) = \mathrm{rng}\, E$, and

(iii) the source of $\mathrm{replaceEdges}(E) = (\text{the source of } G) \cdot (E^{-1})$, and

(iv) the target of $\mathrm{replaceEdges}(E) = (\text{the target of } G) \cdot (E^{-1})$.

The theorem is a consequence of (1).

(4) Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, a one-to-one many sorted set E indexed by the edges of G, and objects e, v, w. Suppose e joins v to w in G. Then $E(e)$ joins $V(v)$ to $V(w)$ in $\mathrm{replaceVerticesEdges}(V, E)$. The theorem is a consequence of (1).

(5) Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, and objects e, v, w. Suppose e joins v to w in G. Then e joins $V(v)$ to $V(w)$ in $\mathrm{replaceVertices}(V)$. The theorem is a consequence of (4).

(6) Let us consider a graph G, a one-to-one many sorted set E indexed by the edges of G, and objects e, v, w. If e joins v to w in G, then $E(e)$ joins v to w in $\mathrm{replaceEdges}(E)$. The theorem is a consequence of (4).

(7) Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, a one-to-one many sorted set E indexed by

the edges of G, and objects e, v, w. Suppose e joins v and w in G. Then $E(e)$ joins $V(v)$ and $V(w)$ in replaceVerticesEdges(V, E). The theorem is a consequence of (4).

(8) Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, and objects e, v, w. Suppose e joins v and w in G. Then e joins $V(v)$ and $V(w)$ in replaceVertices(V). The theorem is a consequence of (5).

(9) Let us consider a graph G, a one-to-one many sorted set E indexed by the edges of G, and objects e, v, w. If e joins v and w in G, then $E(e)$ joins v and w in replaceEdges(E). The theorem is a consequence of (6).

(10) Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, a one-to-one many sorted set E indexed by the edges of G, and objects e, v, w. Suppose $e \in \operatorname{dom} E$ and v, $w \in \operatorname{dom} V$ and $E(e)$ joins $V(v)$ to $V(w)$ in replaceVerticesEdges(V, E). Then e joins v to w in G. The theorem is a consequence of (1).

(11) Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, and objects e, v, w. Suppose v, $w \in \operatorname{dom} V$ and e joins $V(v)$ to $V(w)$ in replaceVertices(V). Then e joins v to w in G. The theorem is a consequence of (2) and (10).

(12) Let us consider a graph G, a one-to-one many sorted set E indexed by the edges of G, and objects e, v, w. Suppose $e \in \operatorname{dom} E$ and $E(e)$ joins v to w in replaceEdges(E). Then e joins v to w in G. The theorem is a consequence of (3) and (10).

(13) Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, a one-to-one many sorted set E indexed by the edges of G, and objects e, v, w. Suppose $e \in \operatorname{dom} E$ and v, $w \in \operatorname{dom} V$ and $E(e)$ joins $V(v)$ and $V(w)$ in replaceVerticesEdges(V, E). Then e joins v and w in G. The theorem is a consequence of (10).

(14) Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, and objects e, v, w. Suppose v, $w \in \operatorname{dom} V$ and e joins $V(v)$ and $V(w)$ in replaceVertices(V). Then e joins v and w in G. The theorem is a consequence of (11).

(15) Let us consider a graph G, a one-to-one many sorted set E indexed by the edges of G, and objects e, v, w. Suppose $e \in \operatorname{dom} E$ and $E(e)$ joins v and w in replaceEdges(E). Then e joins v and w in G. The theorem is a consequence of (12).

Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, and a one-to-one many sorted set E indexed by the edges of G. Now we state the propositions:

(16) There exists a partial graph mapping F from G to replaceVerticesEdges (V, E) such that

(i) $F_{\mathbb{V}} = V$, and

(ii) $F_{\mathbb{E}} = E$, and

(iii) F is directed-isomorphism.

The theorem is a consequence of (1) and (4).

(17) replaceVerticesEdges(V, E) is G-directed-isomorphic.

The theorem is a consequence of (16).

Let G be a loopless graph, V be a non empty, one-to-one many sorted set indexed by the vertices of G, and E be a one-to-one many sorted set indexed by the edges of G. One can verify that replaceVerticesEdges(V, E) is loopless and replaceVertices(V) is loopless.

Let E be a one-to-one many sorted set indexed by the edges of G. Let us observe that replaceEdges(E) is loopless.

Let G be a non loopless graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. One can verify that replaceVerticesEdges(V, E) is non loopless and replaceVertices(V) is non loopless.

Let E be a one-to-one many sorted set indexed by the edges of G. Let us note that replaceEdges(E) is non loopless.

Let G be a non-multi graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Observe that replaceVerticesEdges(V, E) is non-multi and replaceVertices(V) is non-multi.

Let E be a one-to-one many sorted set indexed by the edges of G. Let us note that replaceEdges(E) is non-multi.

Let G be a non non-multi graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Observe that replaceVerticesEdges(V, E) is non non-multi and replaceVertices(V) is non non-multi.

Let E be a one-to-one many sorted set indexed by the edges of G. One can check that replaceEdges(E) is non non-multi.

Let G be a non-directed-multi graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Note that replaceVerticesEdges(V, E) is non-directed-multi and replaceVertices(V) is non-directed-multi.

Let E be a one-to-one many sorted set indexed by the edges of G. One can check that replaceEdges(E) is non-directed-multi.

Let G be a non non-directed-multi graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Note that replaceVerticesEdges$(V,$

$E)$ is non non-directed-multi and replaceVertices(V) is non non-directed-multi.

Let E be a one-to-one many sorted set indexed by the edges of G. Let us observe that replaceEdges(E) is non non-directed-multi.

Let G be a simple graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. One can verify that replaceVerticesEdges(V, E) is simple and replaceVertices(V) is simple.

Let E be a one-to-one many sorted set indexed by the edges of G. Let us observe that replaceEdges(E) is simple.

Let G be a directed-simple graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. One can check that replaceVerticesEdges (V, E) is directed-simple and replaceVertices(V) is directed-simple.

Let E be a one-to-one many sorted set indexed by the edges of G. One can verify that replaceEdges(E) is directed-simple.

Let G be a trivial graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Note that replaceVerticesEdges(V, E) is trivial and replaceVertices(V) is trivial.

Let E be a one-to-one many sorted set indexed by the edges of G. One can check that replaceEdges(E) is trivial.

Let G be a non trivial graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Note that replaceVerticesEdges(V, E) is non trivial and replaceVertices(V) is non trivial.

Let E be a one-to-one many sorted set indexed by the edges of G. Let us observe that replaceEdges(E) is non trivial.

Let G be a vertex-finite graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. One can verify that replaceVerticesEdges(V, E) is vertex-finite and replaceVertices(V) is vertex-finite.

Let E be a one-to-one many sorted set indexed by the edges of G. Let us observe that replaceEdges(E) is vertex-finite.

Let G be a non vertex-finite graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. One can verify that replaceVerticesEdges (V, E) is non vertex-finite and replaceVertices(V) is non vertex-finite.

Let E be a one-to-one many sorted set indexed by the edges of G. Let us note that replaceEdges(E) is non vertex-finite.

Let G be an edge-finite graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Observe that replaceVerticesEdges(V, E) is edge-finite and replaceVertices(V) is edge-finite.

Let E be a one-to-one many sorted set indexed by the edges of G. Let us note that replaceEdges(E) is edge-finite.

Let G be a non edge-finite graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Observe that replaceVerticesEdges(V, E) is non edge-finite and replaceVertices(V) is non edge-finite.

Let E be a one-to-one many sorted set indexed by the edges of G. One can check that replaceEdges(E) is non edge-finite.

Let G be a finite graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Note that replaceVerticesEdges(V, E) is finite and replaceVertices(V) is finite.

Let E be a one-to-one many sorted set indexed by the edges of G. One can check that replaceEdges(E) is finite.

Let G be an acyclic graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Let us note that replaceVerticesEdges(V, E) is acyclic and replaceVertices(V) is acyclic.

Let E be a one-to-one many sorted set indexed by the edges of G. Note that replaceEdges(E) is acyclic.

Let G be a non acyclic graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Let us note that replaceVerticesEdges(V, E) is non acyclic and replaceVertices(V) is non acyclic.

Let E be a one-to-one many sorted set indexed by the edges of G. One can verify that replaceEdges(E) is non acyclic.

Let G be a connected graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. One can check that replaceVerticesEdges(V, E) is connected and replaceVertices(V) is connected.

Let E be a one-to-one many sorted set indexed by the edges of G. One can verify that replaceEdges(E) is connected.

Let G be a non connected graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. One can check that replaceVerticesEdges(V, E) is non connected and replaceVertices(V) is non connected.

Let E be a one-to-one many sorted set indexed by the edges of G. Observe that replaceEdges(E) is non connected.

Let G be a tree-like graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Let us observe that replaceVerticesEdges(V, E) is tree-like and replaceVertices(V) is tree-like.

Let E be a one-to-one many sorted set indexed by the edges of G. Observe that replaceEdges(E) is tree-like.

Let G be a chordal graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. One can verify that replaceVerticesEdges(V, E) is chordal and replaceVertices(V) is chordal.

Let E be a one-to-one many sorted set indexed by the edges of G. Let us observe that replaceEdges(E) is chordal.

Let G be an edgeless graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. One can check that replaceVerticesEdges(V, E) is edgeless and replaceVertices(V) is edgeless.

Let E be a one-to-one many sorted set indexed by the edges of G. One can verify that replaceEdges(E) is edgeless.

Let G be a non edgeless graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. One can check that replaceVerticesEdges(V, E) is non edgeless and replaceVertices(V) is non edgeless.

Let E be a one-to-one many sorted set indexed by the edges of G. Observe that replaceEdges(E) is non edgeless.

Let G be a loopfull graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Let us observe that replaceVerticesEdges(V, E) is loopfull and replaceVertices(V) is loopfull. Let E be a one-to-one many sorted set indexed by the edges of G. Observe that replaceEdges(E) is loopfull.

Let G be a non loopfull graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Let us observe that replaceVerticesEdges(V, E) is non loopfull and replaceVertices(V) is non loopfull.

Let E be a one-to-one many sorted set indexed by the edges of G. Note that replaceEdges(E) is non loopfull.

Let G be a locally-finite graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Let us note that replaceVerticesEdges(V, E) is locally-finite and replaceVertices(V) is locally-finite.

Let E be a one-to-one many sorted set indexed by the edges of G. Note that replaceEdges(E) is locally-finite.

Let G be a non locally-finite graph and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Let us note that replaceVerticesEdges(V, E) is non locally-finite and replaceVertices(V) is non locally-finite. Let E be a one-to-one many sorted set indexed by the edges of G. One can verify that replaceEdges(E) is non locally-finite.

Let c be a non zero cardinal number, G be a c-vertex graph, and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Let us observe that replaceVerticesEdges(V, E) is c-vertex and replaceVertices(V) is c-vertex.

Let E be a one-to-one many sorted set indexed by the edges of G. One can verify that replaceEdges(E) is c-vertex.

Let c be a cardinal number, G be a c-edge graph, and V be a non empty, one-to-one many sorted set indexed by the vertices of G. Let us observe that replaceVerticesEdges(V, E) is c-edge and replaceVertices(V) is c-edge.

Let E be a one-to-one many sorted set indexed by the edges of G. One can verify that replaceEdges(E) is c-edge. Now we state the propositions:

(18) Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, a one-to-one many sorted set E indexed by the edges of G, and a walk W_1 of G. Then there exists a walk W_2 of replaceVerticesEdges(V, E) such that

(i) $V \cdot W_1$.vertexSeq() $= W_2$.vertexSeq(), and

(ii) $E \cdot W_1$.edgeSeq() $= W_2$.edgeSeq().

The theorem is a consequence of (16).

(19) Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, and a walk W_1 of G. Then there exists a walk W_2 of replaceVertices(V) such that

(i) $V \cdot W_1$.vertexSeq() $= W_2$.vertexSeq(), and

(ii) W_1.edgeSeq() $= W_2$.edgeSeq().

The theorem is a consequence of (18).

(20) Let us consider a graph G, a one-to-one many sorted set E indexed by the edges of G, and a walk W_1 of G. Then there exists a walk W_2 of replaceEdges(E) such that

(i) W_1.vertexSeq() $= W_2$.vertexSeq(), and

(ii) $E \cdot W_1$.edgeSeq() $= W_2$.edgeSeq().

The theorem is a consequence of (18).

(21) Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, a one-to-one many sorted set E indexed by the edges of G, and a walk W_2 of replaceVerticesEdges(V, E). Then there exists a walk W_1 of G such that

(i) $V \cdot W_1$.vertexSeq() $= W_2$.vertexSeq(), and

(ii) $E \cdot W_1$.edgeSeq() $= W_2$.edgeSeq().

The theorem is a consequence of (16).

(22) Let us consider a graph G, a non empty, one-to-one many sorted set V indexed by the vertices of G, and a walk W_2 of replaceVertices(V). Then there exists a walk W_1 of G such that

(i) $V \cdot W_1$.vertexSeq() $= W_2$.vertexSeq(), and

(ii) W_1.edgeSeq() $= W_2$.edgeSeq().

The theorem is a consequence of (21).

(23) Let us consider a graph G, a one-to-one many sorted set E indexed by the edges of G, and a walk W_2 of replaceEdges(E). Then there exists a walk W_1 of G such that

(i) W_1.vertexSeq() $= W_2$.vertexSeq(), and

(ii) $E \cdot W_1$.edgeSeq() $= W_2$.edgeSeq().

The theorem is a consequence of (21).

2. Graph Selectors of Graph-yielding Functions

Let F be a graph-yielding function. The functors: the vertices of F, the edges of F, the source of F, and the target of F yielding functions are defined by conditions

(Def. 4) dom the vertices of F = dom F and for every object x such that $x \in$ dom F there exists a graph G such that $G = F(x)$ and (the vertices of $F)(x)$ = the vertices of G,

(Def. 5) dom the edges of F = dom F and for every object x such that $x \in$ dom F there exists a graph G such that $G = F(x)$ and (the edges of $F)(x)$ = the edges of G,

(Def. 6) dom the source of F = dom F and for every object x such that $x \in$ dom F there exists a graph G such that $G = F(x)$ and (the source of $F)(x)$ = the source of G,

(Def. 7) dom the target of F = dom F and for every object x such that $x \in$ dom F there exists a graph G such that $G = F(x)$ and (the target of $F)(x)$ = the target of G,

respectively. Let us observe that the source of F is function yielding and the target of F is function yielding.

Let F be an empty, graph-yielding function. One can verify that the vertices of F is empty and the edges of F is empty and the source of F is empty and the target of F is empty.

Let F be a non empty, graph-yielding function. One can verify that the vertices of F is non empty and the edges of F is non empty and the source of F is non empty and the target of F is non empty.

Let F be a graph-yielding function. One can check that the vertices of F is non-empty.

Let F be a non empty, graph-yielding function. The functors: the vertices of F, the edges of F, the source of F, and the target of F are defined by conditions

(Def. 8) dom the vertices of F = dom F and for every element x of dom F, (the vertices of $F)(x)$ = the vertices of $F(x)$,

(Def. 9) dom the edges of F = dom F and for every element x of dom F, (the edges of $F)(x)$ = the edges of $F(x)$,

(Def. 10) dom the source of F = dom F and for every element x of dom F, (the source of $F)(x)$ = the source of $F(x)$,

(Def. 11) dom the target of F = dom F and for every element x of dom F, (the target of $F)(x)$ = the target of $F(x)$,

respectively.

Let us consider a graph-yielding function F. Now we state the propositions:

(24) The vertices of $\operatorname{rng} F = \operatorname{rng}$(the vertices of F).

(25) The edges of $\operatorname{rng} F = \operatorname{rng}$(the edges of F).

(26) The source of $\operatorname{rng} F = \operatorname{rng}$(the source of F).

(27) The target of $\operatorname{rng} F = \operatorname{rng}$(the target of F).

3. Isomorphisms between Graph-membered Sets or Graph-yielding Functions

Let S_1, S_2 be graph-membered sets. We say that S_1 and S_2 are directed-isomorphic if and only if

(Def. 12) there exists a one-to-one function f such that $\operatorname{dom} f = S_1$ and $\operatorname{rng} f = S_2$ and for every graph G such that $G \in S_1$ holds $f(G)$ is a G-directed-isomorphic graph.

One can check that the predicate is reflexive and symmetric. We say that S_1 and S_2 are isomorphic if and only if

(Def. 13) there exists a one-to-one function f such that $\operatorname{dom} f = S_1$ and $\operatorname{rng} f = S_2$ and for every graph G such that $G \in S_1$ holds $f(G)$ is a G-isomorphic graph.

Let us note that the predicate is reflexive and symmetric.

Let us consider graph-membered sets S_1, S_2, S_3. Now we state the propositions:

(28) If S_1 and S_2 are directed-isomorphic and S_2 and S_3 are directed-isomorphic, then S_1 and S_3 are directed-isomorphic.

(29) If S_1 and S_2 are isomorphic and S_2 and S_3 are isomorphic, then S_1 and S_3 are isomorphic.

Let us consider graph-membered sets S_1, S_2. Now we state the propositions:

(30) If S_1 and S_2 are directed-isomorphic, then S_1 and S_2 are isomorphic.

(31) If S_1 and S_2 are directed-isomorphic, then $\overline{\overline{S_1}} = \overline{\overline{S_2}}$.

(32) If S_1 and S_2 are isomorphic, then $\overline{\overline{S_1}} = \overline{\overline{S_2}}$.

(33) Let us consider empty, graph-membered sets S_1, S_2. Then S_1 and S_2 are directed-isomorphic.

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

(34) $\{G_1\}$ and $\{G_2\}$ are directed-isomorphic if and only if G_2 is G_1-directed-isomorphic.

(35) $\{G_1\}$ and $\{G_2\}$ are isomorphic if and only if G_2 is G_1-isomorphic.

Let us consider graph-membered sets S_1, S_2. Now we state the propositions:

(36) Suppose S_1 and S_2 are isomorphic. Then

(i) if S_1 is empty, then S_2 is empty, and

(ii) if S_1 is loopless, then S_2 is loopless, and

(iii) if S_1 is non-multi, then S_2 is non-multi, and

(iv) if S_1 is simple, then S_2 is simple, and

(v) if S_1 is acyclic, then S_2 is acyclic, and

(vi) if S_1 is connected, then S_2 is connected, and

(vii) if S_1 is tree-like, then S_2 is tree-like, and

(viii) if S_1 is chordal, then S_2 is chordal, and

(ix) if S_1 is edgeless, then S_2 is edgeless, and

(x) if S_1 is loopfull, then S_2 is loopfull.

(37) Suppose S_1 and S_2 are directed-isomorphic. Then

(i) if S_1 is non-directed-multi, then S_2 is non-directed-multi, and

(ii) if S_1 is directed-simple, then S_2 is directed-simple.

Let F_1, F_2 be graph-yielding functions. We say that F_1 and F_2 are directed-isomorphic if and only if

(Def. 14) there exists a one-to-one function p such that $\operatorname{dom} p = \operatorname{dom} F_1$ and $\operatorname{rng} p = \operatorname{dom} F_2$ and for every object x such that $x \in \operatorname{dom} F_1$ there exist graphs G_1, G_2 such that $G_1 = F_1(x)$ and $G_2 = F_2(p(x))$ and G_2 is G_1-directed-isomorphic.

Let us observe that the predicate is reflexive and symmetric. We say that F_1 and F_2 are isomorphic if and only if

(Def. 15) there exists a one-to-one function p such that $\operatorname{dom} p = \operatorname{dom} F_1$ and $\operatorname{rng} p = \operatorname{dom} F_2$ and for every object x such that $x \in \operatorname{dom} F_1$ there exist graphs G_1, G_2 such that $G_1 = F_1(x)$ and $G_2 = F_2(p(x))$ and G_2 is G_1-isomorphic.

Observe that the predicate is reflexive and symmetric.

Let us consider non empty, graph-yielding functions F_1, F_2. Now we state the propositions:

(38) Suppose $\operatorname{dom} F_1 = \operatorname{dom} F_2$ and for every element x_1 of $\operatorname{dom} F_1$ and for every element x_2 of $\operatorname{dom} F_2$ such that $x_1 = x_2$ holds $F_2(x_2)$ is $F_1(x_1)$-directed-isomorphic. Then F_1 and F_2 are directed-isomorphic.

(39) Suppose $\operatorname{dom} F_1 = \operatorname{dom} F_2$ and for every element x_1 of $\operatorname{dom} F_1$ and for every element x_2 of $\operatorname{dom} F_2$ such that $x_1 = x_2$ holds $F_2(x_2)$ is $F_1(x_1)$-isomorphic. Then F_1 and F_2 are isomorphic.

Let us consider graph-yielding functions F_1, F_2, F_3. Now we state the propositions:

(40) If F_1 and F_2 are directed-isomorphic and F_2 and F_3 are directed-isomorphic, then F_1 and F_3 are directed-isomorphic.

(41) If F_1 and F_2 are isomorphic and F_2 and F_3 are isomorphic, then F_1 and F_3 are isomorphic.

(42) Let us consider graph-yielding functions F_1, F_2. If F_1 and F_2 are directed-isomorphic, then F_1 and F_2 are isomorphic.

(43) Let us consider empty, graph-yielding functions F_1, F_2. Then

(i) F_1 and F_2 are directed-isomorphic, and

(ii) F_1 and F_2 are isomorphic.

Let us consider graph-yielding functions F_1, F_2. Now we state the propositions:

(44) If F_1 and F_2 are directed-isomorphic, then $\overline{\overline{F_1}} = \overline{\overline{F_2}}$.

(45) If F_1 and F_2 are isomorphic, then $\overline{\overline{F_1}} = \overline{\overline{F_2}}$.

Let us consider graphs G_1, G_2 and objects x, y. Now we state the propositions:

(46) $x \dot{\longmapsto} G_1$ and $y \dot{\longmapsto} G_2$ are directed-isomorphic if and only if G_2 is G_1-directed-isomorphic.

(47) $x \dot{\longmapsto} G_1$ and $y \dot{\longmapsto} G_2$ are isomorphic if and only if G_2 is G_1-isomorphic.

Let us consider graph-yielding functions F_1, F_2. Now we state the propositions:

(48) Suppose F_1 and F_2 are isomorphic. Then

(i) if F_1 is empty, then F_2 is empty, and

(ii) if F_1 is loopless, then F_2 is loopless, and

(iii) if F_1 is non-multi, then F_2 is non-multi, and

(iv) if F_1 is simple, then F_2 is simple, and

(v) if F_1 is acyclic, then F_2 is acyclic, and

(vi) if F_1 is connected, then F_2 is connected, and

(vii) if F_1 is tree-like, then F_2 is tree-like, and

(viii) if F_1 is chordal, then F_2 is chordal, and

(ix) if F_1 is edgeless, then F_2 is edgeless, and

(x) if F_1 is loopfull, then F_2 is loopfull.

(49) Suppose F_1 and F_2 are directed-isomorphic. Then

(i) if F_1 is non-directed-multi, then F_2 is non-directed-multi, and

(ii) if F_1 is directed-simple, then F_2 is directed-simple.

Let I be a set and F_1, F_2 be graph-yielding many sorted sets indexed by I. Note that F_1 and F_2 are directed-isomorphic if and only if the condition (Def. 16) is satisfied.

(Def. 16) there exists a permutation p of I such that for every object x such that $x \in I$ there exist graphs G_1, G_2 such that $G_1 = F_1(x)$ and $G_2 = F_2(p(x))$ and G_2 is G_1-directed-isomorphic.

One can check that the predicate is reflexive and symmetric. Let us note that F_1 and F_2 are isomorphic if and only if the condition (Def. 17) is satisfied.

(Def. 17) there exists a permutation p of I such that for every object x such that $x \in I$ there exist graphs G_1, G_2 such that $G_1 = F_1(x)$ and $G_2 = F_2(p(x))$ and G_2 is G_1-isomorphic.

Note that the predicate is reflexive and symmetric.

4. Distinguishing the Vertex and Edge Sets of Several Graphs from Each Other

Let S be a graph-membered set. We say that S is vertex-disjoint if and only if

(Def. 18) for every graphs G_1, G_2 such that $G_1, G_2 \in S$ and $G_1 \neq G_2$ holds the vertices of G_1 misses the vertices of G_2.

We say that S is edge-disjoint if and only if

(Def. 19) for every graphs G_1, G_2 such that $G_1, G_2 \in S$ and $G_1 \neq G_2$ holds the edges of G_1 misses the edges of G_2.

Now we state the proposition:

(50) Let us consider a graph-membered set S. Then S is vertex-disjoint and edge-disjoint if and only if for every graphs G_1, G_2 such that $G_1, G_2 \in S$ and $G_1 \neq G_2$ holds the vertices of G_1 misses the vertices of G_2 and the edges of G_1 misses the edges of G_2.

Let us note that every graph-membered set which is trivial is also vertex-disjoint and edge-disjoint and every graph-membered set which is edgeless is also edge-disjoint and every graph-membered set which is edge-disjoint is also $\cup$-tolerating and every graph-membered set which is vertex-disjoint and $\cup$-tolerating is also edge-disjoint.

Let G be a graph. One can check that $\{G\}$ is vertex-disjoint and edge-disjoint.

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

(51) $\{G_1, G_2\}$ is vertex-disjoint if and only if $G_1 = G_2$ or the vertices of G_1 misses the vertices of G_2.

(52) $\{G_1, G_2\}$ is edge-disjoint if and only if $G_1 = G_2$ or the edges of G_1 misses the edges of G_2.

One can verify that there exists a graph-membered set which is non empty, $\cup$-tolerating, vertex-disjoint, edge-disjoint, acyclic, simple, directed-simple, loopless, non-multi, and non-directed-multi.

Let S be a vertex-disjoint, graph-membered set. Note that the vertices of S is mutually-disjoint.

Let S be an edge-disjoint, graph-membered set. One can verify that the edges of S is mutually-disjoint.

Let S be a vertex-disjoint, graph-membered set. Observe that every subset of S is vertex-disjoint.

Let S_1 be a vertex-disjoint, graph-membered set and S_2 be a set. Let us note that $S_1 \cap S_2$ is vertex-disjoint and $S_1 \setminus S_2$ is vertex-disjoint.

Let S be an edge-disjoint, graph-membered set. One can verify that every subset of S is edge-disjoint.

Let S_1 be an edge-disjoint, graph-membered set and S_2 be a set. Let us observe that $S_1 \cap S_2$ is edge-disjoint and $S_1 \setminus S_2$ is edge-disjoint.

Let us consider graph-membered sets S_1, S_2. Now we state the propositions:

(53) If $S_1 \cup S_2$ is vertex-disjoint, then S_1 is vertex-disjoint and S_2 is vertex-disjoint.

(54) If $S_1 \cup S_2$ is edge-disjoint, then S_1 is edge-disjoint and S_2 is edge-disjoint.

Let us consider vertex-disjoint graph union sets S_1, S_2, a graph union G_1 of S_1, and a graph union G_2 of S_2. Now we state the propositions:

(55) If S_1 and S_2 are directed-isomorphic, then G_2 is G_1-directed-isomorphic.
Proof: Consider h being a one-to-one function such that $\operatorname{dom} h = S_1$ and $\operatorname{rng} h = S_2$ and for every graph G such that $G \in S_1$ holds $h(G)$ is a G-directed-isomorphic graph. Define $\mathcal{Q}[\text{object}, \text{object}] \equiv$ there exists an element G of S_1 and there exists a partial graph mapping F from G to $h(G)$ such that $\$_1 = G$ and $\$_2 = F$ and F is directed-isomorphism. For every element G of S_1, there exists an object F such that $\mathcal{Q}[G, F]$.

Consider H being a many sorted set indexed by S_1 such that for every element G of S_1, $\mathcal{Q}[G, H(G)]$. For every element G of S_1, there exists a partial graph mapping F from G to $h(G)$ such that $H(G) = F$ and F is directed-isomorphism. Set $V = \operatorname{rng} \operatorname{pr1}(H)$. Set $E = \operatorname{rng} \operatorname{pr2}(H)$. For every object y such that $y \in V$ holds y is a function. For every functions f_1, f_2 such that f_1, $f_2 \in V$ holds f_1 tolerates f_2. For every object y such that $y \in E$ holds y is a function. For every functions g_1, g_2 such that g_1, $g_2 \in E$ holds g_1 tolerates g_2. $\square$

(56) Suppose S_1 and S_2 are isomorphic. Then there exists a vertex-disjoint

graph union set S_3 and there exists a subset E of the edges of G_2 and there exists a graph union G_3 of S_3 such that S_1 and S_3 are directed-isomorphic and G_3 is a graph given by reversing directions of the edges E of G_2.

PROOF: Consider h being a one-to-one function such that $\operatorname{dom} h = S_1$ and $\operatorname{rng} h = S_2$ and for every graph G such that $G \in S_1$ holds $h(G)$ is a G-isomorphic graph. Define $\mathcal{Q}[\text{object}, \text{object}] \equiv$ there exists an element G of S_1 and there exists a partial graph mapping F from G to $h(G)$ such that $\$_1 = G$ and $\$_2 = F$ and F is isomorphism. For every element G of S_1, there exists an object F such that $\mathcal{Q}[G, F]$. Consider H being a many sorted set indexed by S_1 such that for every element G of S_1, $\mathcal{Q}[G, H(G)]$. For every element G of S_1, there exists a partial graph mapping F from G to $h(G)$ such that $H(G) = F$ and F is isomorphism. Define $\mathcal{R}[\text{object}, \text{object}] \equiv$ there exists an element G of S_1 and there exists a subset E of the edges of $h(G)$ such that $\$_1 = G$ and $\$_2 = E$ and for every graph G' given by reversing directions of the edges E of $h(G)$, there exists a partial graph mapping F from G to G' such that $F = H(G)$ and F is directed-isomorphism.

For every element G of S_1, there exists an object E such that $\mathcal{R}[G, E]$ by [5, (89)]. Consider A being a many sorted set indexed by S_1 such that for every element G of S_1, $\mathcal{R}[G, A(G)]$. For every element G of S_1, $A(G)$ is a subset of the edges of $h(G)$. For every element G of S_1 and for every graph G' given by reversing directions of the edges $A(G)$ of $h(G)$, there exists a partial graph mapping F from G to G' such that $F = H(G)$ and F is directed-isomorphism. Define $\mathcal{U}(\text{element of } S_1) =$ the graph given by reversing directions of the edges $A(\$_1)$ of $h(\$_1)$. Consider B being a many sorted set indexed by S_1 such that for every element G of S_1, $B(G) = \mathcal{U}(G)$. For every object y such that $y \in \bigcup \operatorname{rng} A$ holds $y \in$ the edges of G_2. □

(57) If S_1 and S_2 are isomorphic, then G_2 is G_1-isomorphic. The theorem is a consequence of (56) and (55).

(58) Let us consider a vertex-disjoint graph union set S, a graph union G of S, and a walk W of G. Then there exists an element H of S such that W is a walk of H.

PROOF: Define $\mathcal{P}[\text{walk of } G] \equiv$ there exists an element H of S such that $\$_1$ is a walk of H. For every trivial walk W of G, $\mathcal{P}[W]$ by [8, (128)]. For every walk W of G and for every object e such that $e \in W.\text{last}().\text{edgesInOut}()$ and $\mathcal{P}[W]$ holds $\mathcal{P}[W.\text{addEdge}(e)]$ by [7, (21)], [8, (16)], [9, (67)], [6, (117)]. For every walk W of G, $\mathcal{P}[W]$ by [8, Sch.1]. □

Let us consider a vertex-disjoint graph union set S and a graph union G of S. Now we state the propositions:

(59) If G is connected, then there exists a graph H such that $S = \{H\}$. The theorem is a consequence of (58).

(60) (i) S is non-multi iff G is non-multi, and

(ii) S is non-directed-multi iff G is non-directed-multi, and

(iii) S is acyclic iff G is acyclic.

The theorem is a consequence of (58).

(61) (i) S is simple iff G is simple, and

(ii) S is directed-simple iff G is directed-simple.

The theorem is a consequence of (60).

Let S be a vertex-disjoint, non-multi graph union set. Let us note that every graph union of S is non-multi.

Let S be a vertex-disjoint, non-directed-multi graph union set. One can check that every graph union of S is non-directed-multi.

Let S be a vertex-disjoint, simple graph union set. Let us observe that every graph union of S is simple.

Let S be a vertex-disjoint, directed-simple graph union set. Observe that every graph union of S is directed-simple.

Let S be a vertex-disjoint, acyclic graph union set. Let us note that every graph union of S is acyclic.

Now we state the propositions:

(62) Let us consider a vertex-disjoint graph union set S, an element H of S, and a graph union G of S. Then H is a subgraph of G induced by the vertices of H.

(63) Let us consider a vertex-disjoint graph union set S, and a graph union G of S. Then

(i) S is chordal iff G is chordal, and

(ii) S is loopfull iff G is loopfull.

The theorem is a consequence of (58) and (62).

(64) Let us consider a vertex-disjoint graph union set S, a graph union G of S, an element H of S, a vertex v of G, and a vertex w of H. If $v = w$, then G.reachableFrom(v) $=$ H.reachableFrom(w). The theorem is a consequence of (58).

(65) Let us consider a vertex-disjoint graph union set S, and a graph union G of S. Then G.componentSet() $= \bigcup$ the set of all H.componentSet() where H is an element of S. The theorem is a consequence of (64).

(66) Let us consider a vertex-disjoint, non empty, graph-membered set S. Then the set of all H.componentSet() where H is an element of S is mutually-disjoint.

(67) Let us consider a non empty, connected, graph-membered set S. Then the set of all H.componentSet() where H is an element of $S =$ SmallestPartition(the vertices of S).

Let us consider a vertex-disjoint graph union set S and a graph union G of S. Now we state the propositions:

(68) $\overline{\overline{S}} \subseteq G$.numComponents(). The theorem is a consequence of (66) and (65).

(69) If S is connected, then $\overline{\overline{S}} = G$.numComponents(). The theorem is a consequence of (67) and (65).

Let F be a graph-yielding function. We say that F is vertex-disjoint if and only if

(Def. 20) for every objects x_1, x_2 such that $x_1, x_2 \in \operatorname{dom} F$ and $x_1 \neq x_2$ there exist graphs G_1, G_2 such that $G_1 = F(x_1)$ and $G_2 = F(x_2)$ and the vertices of G_1 misses the vertices of G_2.

We say that F is edge-disjoint if and only if

(Def. 21) for every objects x_1, x_2 such that $x_1, x_2 \in \operatorname{dom} F$ and $x_1 \neq x_2$ there exist graphs G_1, G_2 such that $G_1 = F(x_1)$ and $G_2 = F(x_2)$ and the edges of G_1 misses the edges of G_2.

Observe that every graph-yielding function which is trivial is also vertex-disjoint and edge-disjoint and every graph-yielding function which is vertex-disjoint is also one-to-one.

Let F be a non empty, graph-yielding function. Let us observe that F is vertex-disjoint if and only if the condition (Def. 22) is satisfied.

(Def. 22) for every elements x_1, x_2 of $\operatorname{dom} F$ such that $x_1 \neq x_2$ holds the vertices of $F(x_1)$ misses the vertices of $F(x_2)$.

Observe that F is edge-disjoint if and only if the condition (Def. 23) is satisfied.

(Def. 23) for every elements x_1, x_2 of $\operatorname{dom} F$ such that $x_1 \neq x_2$ holds the edges of $F(x_1)$ misses the edges of $F(x_2)$.

Let us consider a non empty, graph-yielding function F. Now we state the propositions:

(70) F is vertex-disjoint if and only if for every elements x_1, x_2 of $\operatorname{dom} F$ such that $x_1 \neq x_2$ holds (the vertices of F)(x_1) misses (the vertices of F)(x_2).

(71) F is edge-disjoint if and only if for every elements x_1, x_2 of $\operatorname{dom} F$ such that $x_1 \neq x_2$ holds (the edges of F)(x_1) misses (the edges of F)(x_2).

(72) F is vertex-disjoint and edge-disjoint if and only if for every elements x_1, x_2 of $\operatorname{dom} F$ such that $x_1 \neq x_2$ holds the vertices of $F(x_1)$ misses the vertices of $F(x_2)$ and the edges of $F(x_1)$ misses the edges of $F(x_2)$.

(73) F is vertex-disjoint and edge-disjoint if and only if for every elements x_1, x_2 of $\operatorname{dom} F$ such that $x_1 \neq x_2$ holds (the vertices of $F)(x_1)$ misses (the vertices of $F)(x_2)$ and (the edges of $F)(x_1)$ misses (the edges of $F)(x_2)$. The theorem is a consequence of (70) and (71).

Let x be an object and G be a graph. One can check that $x \dot{\longmapsto} G$ is vertex-disjoint and edge-disjoint and $\langle G \rangle$ is vertex-disjoint and edge-disjoint and there exists a graph-yielding function which is non empty, vertex-disjoint, and edge-disjoint.

Let F be a vertex-disjoint, graph-yielding function. Observe that $\operatorname{rng} F$ is vertex-disjoint.

Let F be an edge-disjoint, graph-yielding function. Let us note that $\operatorname{rng} F$ is edge-disjoint.

Let us consider non empty, one-to-one, graph-yielding functions F_1, F_2. Now we state the propositions:

(74) If F_1 and F_2 are directed-isomorphic, then $\operatorname{rng} F_1$ and $\operatorname{rng} F_2$ are directed-isomorphic.

(75) If F_1 and F_2 are isomorphic, then $\operatorname{rng} F_1$ and $\operatorname{rng} F_2$ are isomorphic.

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

(76) $\langle G_1, G_2 \rangle$ is vertex-disjoint if and only if the vertices of G_1 misses the vertices of G_2.

(77) $\langle G_1, G_2 \rangle$ is edge-disjoint if and only if the edges of G_1 misses the edges of G_2.

5. Distinguishing the Range of a Graph-yielding Function

Let f be a function and x be an object. The functor $\coprod(f, x)$ yielding a many sorted set indexed by $f(x)$ is defined by the term

(Def. 24) $\langle f(x) \longmapsto \langle f, x \rangle, \mathrm{id}_{f(x)} \rangle$.

Now we state the propositions:

(78) Let us consider a function f, and objects x, y. Suppose $x \in \operatorname{dom} f$ and $y \in f(x)$. Then $\coprod(f, x)(y) = \langle f, x, y \rangle$.

(79) Let us consider a function f, and objects x, z. Suppose $x \in \operatorname{dom} f$ and $z \in \operatorname{rng} \coprod(f, x)$. Then there exists an object y such that

(i) $y \in f(x)$, and

(ii) $z = \langle f, x, y \rangle$.

The theorem is a consequence of (78).

(80) Let us consider a function f, and an object x. Then $\operatorname{rng} \coprod(f, x) = \{\langle f, x\rangle\} \times f(x)$. The theorem is a consequence of (79) and (78).

Let us consider a function f and objects x_1, x_2. Now we state the propositions:

(81) $\operatorname{rng} \coprod(f, x_1)$ misses $f(x_2)$. The theorem is a consequence of (79).

(82) If $x_1 \neq x_2$, then $\operatorname{rng} \coprod(f, x_1)$ misses $\operatorname{rng} \coprod(f, x_2)$. The theorem is a consequence of (79).

Let f be a function and x be an object. One can verify that $\coprod(f, x)$ is one-to-one.

Let f be an empty function. One can verify that $\coprod(f, x)$ is empty.

Let f be a non empty, non-empty function and x be an element of $\operatorname{dom} f$. One can verify that $\coprod(f, x)$ is non empty.

Let F be a non empty, graph-yielding function and x be an element of $\operatorname{dom} F$. One can check that $\coprod(\text{the vertices of } F, x)$ is non empty and (the vertices of $F(x)$)-defined and $\coprod(\text{the edges of } F, x)$ is (the edges of $F(x)$)-defined and $\coprod(\text{the vertices of } F, x)$ is total as a (the vertices of $F(x)$)-defined function and $\coprod(\text{the edges of } F, x)$ is total as a (the edges of $F(x)$)-defined function.

The functor $\coprod F$ yielding a graph-yielding function is defined by

(Def. 25) $\operatorname{dom} it = \operatorname{dom} F$ and for every element x of $\operatorname{dom} F$, $it(x) =$ replaceVerticesEdges$(\coprod(\text{the vertices of } F, x), \coprod(\text{the edges of } F, x))$.

Note that $\coprod F$ is non empty and $\coprod F$ is plain.

Let us consider a non empty, graph-yielding function F and an element x of $\operatorname{dom} F$. Now we state the propositions:

(83) (The vertices of $\coprod F)(x) = \{\langle\text{the vertices of } F, x\rangle\} \times$ (the vertices of $F)(x)$. The theorem is a consequence of (1) and (80).

(84) (The edges of $\coprod F)(x) = \{\langle\text{the edges of } F, x\rangle\} \times$ (the edges of $F)(x)$. The theorem is a consequence of (1) and (80).

Let F be a non empty, graph-yielding function. Note that $\coprod F$ is vertex-disjoint and edge-disjoint.

Let us consider a non empty, graph-yielding function F, an element x of $\operatorname{dom} F$, and an element x' of $\operatorname{dom}(\coprod F)$. Now we state the propositions:

(85) Suppose $x = x'$. Then there exists a partial graph mapping G from $F(x)$ to $(\coprod F)(x')$ such that

(i) $G_{\mathbb{V}} = \coprod(\text{the vertices of } F, x)$, and

(ii) $G_{\mathbb{E}} = \coprod(\text{the edges of } F, x)$, and

(iii) G is directed-isomorphism.

The theorem is a consequence of (16).

(86) If $x = x'$, then $(\coprod F)(x')$ is $F(x)$-directed-isomorphic. The theorem is a consequence of (85).

(87) Let us consider a non empty, graph-yielding function F. Then F and $\coprod F$ are directed-isomorphic. The theorem is a consequence of (86) and (38).

Let us consider non empty, graph-yielding functions F_1, F_2. Now we state the propositions:

(88) If F_1 and F_2 are directed-isomorphic, then $\coprod F_1$ and $\coprod F_2$ are directed-isomorphic. The theorem is a consequence of (87) and (40).

(89) If F_1 and F_2 are isomorphic, then $\coprod F_1$ and $\coprod F_2$ are isomorphic. The theorem is a consequence of (42), (87), and (41).

Let us consider a non empty, graph-yielding function F, an element x of $\operatorname{dom} F$, an element x' of $\operatorname{dom}(\coprod F)$, and objects v, e, w. Now we state the propositions:

(90) Suppose $x = x'$. Then suppose e joins v to w in $F(x)$. Then $\langle$the edges of F, x, $e\rangle$ joins $\langle$the vertices of F, x, $v\rangle$ to $\langle$the vertices of F, x, $w\rangle$ in $(\coprod F)(x')$. The theorem is a consequence of (85) and (78).

(91) Suppose $x = x'$. Then suppose e joins v and w in $F(x)$. Then $\langle$the edges of F, x, $e\rangle$ joins $\langle$the vertices of F, x, $v\rangle$ and $\langle$the vertices of F, x, $w\rangle$ in $(\coprod F)(x')$. The theorem is a consequence of (90).

Let us consider a non empty, graph-yielding function F, an element x of $\operatorname{dom} F$, an element x' of $\operatorname{dom}(\coprod F)$, and objects v', e', w'. Now we state the propositions:

(92) Suppose $x = x'$ and e' joins v' to w' in $(\coprod F)(x')$. Then there exist objects v, e, w such that

(i) e joins v to w in $F(x)$, and

(ii) $e' = \langle$the edges of F, x, $e\rangle$, and

(iii) $v' = \langle$the vertices of F, x, $v\rangle$, and

(iv) $w' = \langle$the vertices of F, x, $w\rangle$.

The theorem is a consequence of (85), (83), (80), (79), (84), and (78).

(93) Suppose $x = x'$ and e' joins v' and w' in $(\coprod F)(x')$. Then there exist objects v, e, w such that

(i) e joins v and w in $F(x)$, and

(ii) $e' = \langle$the edges of F, x, $e\rangle$, and

(iii) $v' = \langle$the vertices of F, x, $v\rangle$, and

(iv) $w' = \langle$the vertices of F, x, $w\rangle$.

The theorem is a consequence of (92).

Let F be a non empty, loopless, graph-yielding function. One can verify that $\coprod F$ is loopless.

Let F be a non empty, non loopless, graph-yielding function. Note that $\coprod F$ is non loopless.

Let F be a non empty, non-multi, graph-yielding function. Observe that $\coprod F$ is non-multi.

Let F be a non empty, non non-multi, graph-yielding function. One can verify that $\coprod F$ is non non-multi.

Let F be a non empty, non-directed-multi, graph-yielding function. Note that $\coprod F$ is non-directed-multi.

Let F be a non empty, non non-directed-multi, graph-yielding function. One can verify that $\coprod F$ is non non-directed-multi.

Let F be a non empty, simple, graph-yielding function. Observe that $\coprod F$ is simple.

Let F be a non empty, directed-simple, graph-yielding function. One can check that $\coprod F$ is directed-simple.

Let F be a non empty, acyclic, graph-yielding function. Let us observe that $\coprod F$ is acyclic.

Let F be a non empty, non acyclic, graph-yielding function. One can check that $\coprod F$ is non acyclic.

Let F be a non empty, connected, graph-yielding function. Let us note that $\coprod F$ is connected.

Let F be a non empty, non connected, graph-yielding function. Let us observe that $\coprod F$ is non connected.

Let F be a non empty, tree-like, graph-yielding function. One can check that $\coprod F$ is tree-like.

Let F be a non empty, edgeless, graph-yielding function. Observe that $\coprod F$ is edgeless.

Let F be a non empty, non edgeless, graph-yielding function. One can verify that $\coprod F$ is non edgeless.

Let F be a non empty, graph-yielding function and z be an element of $\operatorname{dom} F$. The functor $\coprod(F, z)$ yielding a graph-yielding function is defined by the term

(Def. 26) $\coprod F +\cdot (z, F(z){\restriction}(\text{the graph selectors}))$.

Let us note that $\coprod(F, z)$ is non empty. Now we state the propositions:

(94) Let us consider a non empty, graph-yielding function F, and an element z of $\operatorname{dom} F$. Then $\operatorname{dom} F = \operatorname{dom}(\coprod(F, z))$.

(95) Let us consider a non empty, graph-yielding function F, an element z of $\operatorname{dom} F$, and a graph-yielding function G. Then $G = \coprod(F, z)$ if and only

if $\operatorname{dom} G = \operatorname{dom} F$ and $G(z) = F(z)\restriction$(the graph selectors) and for every element x of $\operatorname{dom} F$ such that $x \neq z$ holds $G(x) =$ replaceVerticesEdges($\coprod$(the vertices of F, x), $\coprod$(the edges of F, x)).

(96) Let us consider a non empty, graph-yielding function F, and an element z of $\operatorname{dom} F$. Then $\coprod(F, z)(z) = F(z)\restriction$(the graph selectors).

Let F be a non empty, graph-yielding function and z be an element of $\operatorname{dom} F$. Observe that $\coprod(F, z)$ is plain. Now we state the propositions:

(97) Let us consider a non empty, graph-yielding function F, and an element z of $\operatorname{dom} F$. Then (the vertices of $\coprod(F, z))(z) =$ (the vertices of $F)(z)$. The theorem is a consequence of (94) and (96).

(98) Let us consider a non empty, graph-yielding function F, and elements x, z of $\operatorname{dom} F$. Suppose $x \neq z$. Then (the vertices of $\coprod(F, z))(x) =$ (the vertices of $\coprod F)(x)$. The theorem is a consequence of (95).

Let us consider a non empty, graph-yielding function F and an element z of $\operatorname{dom} F$. Now we state the propositions:

(99) The vertices of $\coprod(F, z) =$ (the vertices of $\coprod F) +\cdot (z$, the vertices of $F(z))$. The theorem is a consequence of (97) and (98).

(100) (The edges of $\coprod(F, z))(z) =$ (the edges of $F)(z)$. The theorem is a consequence of (94) and (96).

(101) Let us consider a non empty, graph-yielding function F, and elements x, z of $\operatorname{dom} F$. Suppose $x \neq z$. Then (the edges of $\coprod(F, z))(x) =$ (the edges of $\coprod F)(x)$. The theorem is a consequence of (95).

(102) Let us consider a non empty, graph-yielding function F, and an element z of $\operatorname{dom} F$. Then the edges of $\coprod(F, z) =$ (the edges of $\coprod F) +\cdot (z$, the edges of $F(z))$. The theorem is a consequence of (100) and (101).

Let F be a non empty, graph-yielding function and z be an element of $\operatorname{dom} F$. Let us note that $\coprod(F, z)$ is vertex-disjoint and edge-disjoint.

Let us consider a non empty, graph-yielding function F, elements x, z of $\operatorname{dom} F$, and an element x' of $\operatorname{dom}(\coprod(F, z))$. Now we state the propositions:

(103) Suppose $x \neq z$ and $x = x'$. Then there exists a partial graph mapping G from $F(x)$ to $\coprod(F, z)(x')$ such that

- (i) $G_{\mathbb{V}} = \coprod$(the vertices of F, x), and
- (ii) $G_{\mathbb{E}} = \coprod$(the edges of F, x), and
- (iii) G is directed-isomorphism.

The theorem is a consequence of (85).

(104) If $x = x'$, then $\coprod(F, z)(x')$ is $(F(x))$-directed-isomorphic. The theorem is a consequence of (96) and (103).

Let us consider a non empty, graph-yielding function F and an element z of $\operatorname{dom} F$. Now we state the propositions:

(105) F and $\coprod(F, z)$ are directed-isomorphic. The theorem is a consequence of (104) and (38).

(106) $\coprod F$ and $\coprod(F, z)$ are directed-isomorphic. The theorem is a consequence of (87), (105), and (40).

(107) Let us consider non empty, graph-yielding functions F_1, F_2, an element z_1 of $\operatorname{dom} F_1$, and an element z_2 of $\operatorname{dom} F_2$. Suppose F_1 and F_2 are directed-isomorphic. Then $\coprod(F_1, z_1)$ and $\coprod(F_2, z_2)$ are directed-isomorphic. The theorem is a consequence of (105) and (40).

Let us consider a non empty, graph-yielding function F, an element z of $\operatorname{dom} F$, an element z' of $\operatorname{dom}(\coprod(F, z))$, and objects v, e, w. Now we state the propositions:

(108) If $z = z'$, then e joins v to w in $F(z)$ iff e joins v to w in $\coprod(F, z)(z')$. The theorem is a consequence of (96).

(109) If $z = z'$, then e joins v and w in $F(z)$ iff e joins v and w in $\coprod(F, z)(z')$. The theorem is a consequence of (96).

Let us consider a non empty, graph-yielding function F, elements x, z of $\operatorname{dom} F$, an element x' of $\operatorname{dom}(\coprod(F, z))$, and objects v, e, w. Now we state the propositions:

(110) Suppose $x \neq z$ and $x = x'$. Then suppose e joins v to w in $F(x)$. Then $\langle$the edges of F, x, $e\rangle$ joins $\langle$the vertices of F, x, $v\rangle$ to $\langle$the vertices of F, x, $w\rangle$ in $\coprod(F, z)(x')$. The theorem is a consequence of (90).

(111) Suppose $x \neq z$ and $x = x'$. Then suppose e joins v and w in $F(x)$. Then $\langle$the edges of F, x, $e\rangle$ joins $\langle$the vertices of F, x, $v\rangle$ and $\langle$the vertices of F, x, $w\rangle$ in $\coprod(F, z)(x')$. The theorem is a consequence of (91).

Let us consider a non empty, graph-yielding function F, elements x, z of $\operatorname{dom} F$, an element x' of $\operatorname{dom}(\coprod(F, z))$, and objects v', e', w'. Now we state the propositions:

(112) Suppose $x \neq z$ and $x = x'$ and e' joins v' to w' in $\coprod(F, z)(x')$. Then there exist objects v, e, w such that

(i) e joins v to w in $F(x)$, and

(ii) $e' = \langle$the edges of F, x, $e\rangle$, and

(iii) $v' = \langle$the vertices of F, x, $v\rangle$, and

(iv) $w' = \langle$the vertices of F, x, $w\rangle$.

The theorem is a consequence of (92).

(113) Suppose $x \neq z$ and $x = x'$ and e' joins v' and w' in $\coprod(F, z)(x')$. Then there exist objects v, e, w such that

(i) e joins v and w in $F(x)$, and

(ii) $e' = \langle$the edges of F, x, $e\rangle$, and

(iii) $v' = \langle$the vertices of F, x, $v\rangle$, and

(iv) $w' = \langle$the vertices of F, x, $w\rangle$.

The theorem is a consequence of (93).

Let F be a non empty, loopless, graph-yielding function and z be an element of $\operatorname{dom} F$. One can check that $\coprod(F, z)$ is loopless.

Let F be a non empty, non loopless, graph-yielding function. Let us observe that $\coprod(F, z)$ is non loopless.

Let F be a non empty, non-multi, graph-yielding function. Let us note that $\coprod(F, z)$ is non-multi.

Let F be a non empty, non non-multi, graph-yielding function. One can check that $\coprod(F, z)$ is non non-multi.

Let F be a non empty, non-directed-multi, graph-yielding function. Let us observe that $\coprod(F, z)$ is non-directed-multi.

Let F be a non empty, non non-directed-multi, graph-yielding function. Let us observe that $\coprod(F, z)$ is non non-directed-multi.

Let F be a non empty, simple, graph-yielding function. Let us observe that $\coprod(F, z)$ is simple.

Let F be a non empty, directed-simple, graph-yielding function. Note that $\coprod(F, z)$ is directed-simple.

Let F be a non empty, acyclic, graph-yielding function. Let us observe that $\coprod(F, z)$ is acyclic.

Let F be a non empty, non acyclic, graph-yielding function. Let us note that $\coprod(F, z)$ is non acyclic.

Let F be a non empty, connected, graph-yielding function. One can check that $\coprod(F, z)$ is connected.

Let F be a non empty, non connected, graph-yielding function. Let us observe that $\coprod(F, z)$ is non connected.

Let F be a non empty, tree-like, graph-yielding function. Let us note that $\coprod(F, z)$ is tree-like.

Let F be a non empty, edgeless, graph-yielding function. One can verify that $\coprod(F, z)$ is edgeless.

Let F be a non empty, non edgeless, graph-yielding function. Observe that $\coprod(F, z)$ is non edgeless.

Let us consider graphs G_2, H and a partial graph mapping F from G_2 to H. Now we state the propositions:

(114) If F is directed and weak subgraph embedding, then there exists a supergraph G_1 of G_2 such that G_1 is H-directed-isomorphic.
PROOF: Set $c =$ (the vertices of H) $\longmapsto$ (the vertices of G_2). $\mathrm{rng}\langle c, \mathrm{id}_\alpha \rangle \cap \mathrm{rng}(F_{\mathbb{V}})^{-1} = \emptyset$, where α is the vertices of H. Set $d =$ (the edges of H) $\longmapsto$ (the edges of G_2). $\mathrm{rng}\langle d, \mathrm{id}_\alpha \rangle \cap \mathrm{rng}(F_{\mathbb{E}})^{-1} = \emptyset$, where α is the edges of H. □

(115) If F is weak subgraph embedding, then there exists a supergraph G_1 of G_2 such that G_1 is H-isomorphic. The theorem is a consequence of (114).

6. The Sum of Graphs

Let F be a non empty, graph-yielding function.

A graph sum of F is a graph defined by

(Def. 27) there exists a graph union G' of $\mathrm{rng} \coprod F$ such that *it* is G'-directed-isomorphic.

Now we state the proposition:

(116) Let us consider a non empty, graph-yielding function F, a graph sum S of F, and a graph union G' of $\mathrm{rng} \coprod F$. Then S is G'-directed-isomorphic.

Let us consider non empty, graph-yielding functions F_1, F_2, a graph sum S_1 of F_1, and a graph sum S_2 of F_2. Now we state the propositions:

(117) If F_1 and F_2 are directed-isomorphic, then S_2 is S_1-directed-isomorphic. The theorem is a consequence of (74), (88), (55), and (116).

(118) If F_1 and F_2 are isomorphic, then S_2 is S_1-isomorphic. The theorem is a consequence of (89), (57), (75), and (116).

Now we state the propositions:

(119) Let us consider a non empty, graph-yielding function F, and graph sums S_1, S_2 of F. Then S_2 is S_1-directed-isomorphic.

(120) Let us consider an object x, and a graph G. Then every graph sum of $x \dot{\longmapsto} G$ is G-directed-isomorphic. The theorem is a consequence of (17).

(121) Let us consider a non empty, graph-yielding function F, and a graph sum S of F. Suppose S is connected. Then there exists an object x and there exists a connected graph G such that $F = x \dot{\longmapsto} G$. The theorem is a consequence of (59) and (120).

Let X be a non empty set. Observe that there exists a graph-yielding many sorted set indexed by X which is non empty, vertex-disjoint, and edge-disjoint.

Now we state the propositions:

(122) Let us consider a non empty, graph-yielding function F, an element x of $\mathrm{dom}\, F$, and a graph sum S of F. Then there exists a partial graph

mapping M from $F(x)$ to S such that M is strong subgraph embedding. The theorem is a consequence of (62) and (17).

(123) Let us consider a non empty, graph-yielding function F, and an element z of dom F. Then there exists a graph sum S of F such that S is supergraph of $F(z)$ and graph union of rng $\coprod(F, z)$. The theorem is a consequence of (106), (55), (74), (94), and (95).

(124) Let us consider a non empty, graph-yielding function F, and a graph sum S of F. Then

(i) F is loopless iff S is loopless, and

(ii) F is non-multi iff S is non-multi, and

(iii) F is non-directed-multi iff S is non-directed-multi, and

(iv) F is simple iff S is simple, and

(v) F is directed-simple iff S is directed-simple, and

(vi) F is chordal iff S is chordal, and

(vii) F is edgeless iff S is edgeless, and

(viii) F is loopfull iff S is loopfull.

Let F be a non empty, loopless, graph-yielding function. Observe that every graph sum of F is loopless.

Let F be a non empty, non loopless, graph-yielding function. Note that every graph sum of F is non loopless.

Let F be a non empty, non-directed-multi, graph-yielding function. One can verify that every graph sum of F is non-directed-multi.

Let F be a non empty, non non-directed-multi, graph-yielding function. Observe that every graph sum of F is non non-directed-multi.

Let F be a non empty, non-multi, graph-yielding function. Note that every graph sum of F is non-multi.

Let F be a non empty, non non-multi, graph-yielding function. One can verify that every graph sum of F is non non-multi.

Let F be a non empty, simple, graph-yielding function. Observe that every graph sum of F is simple.

Let F be a non empty, directed-simple, graph-yielding function. Observe that every graph sum of F is directed-simple.

Let F be a non empty, edgeless, graph-yielding function. Observe that every graph sum of F is edgeless.

Let F be a non empty, non edgeless, graph-yielding function. Note that every graph sum of F is non edgeless.

Let F be a non empty, loopfull, graph-yielding function. One can verify that every graph sum of F is loopfull.

Let F be a non empty, non loopfull, graph-yielding function. Observe that every graph sum of F is non loopfull. Now we state the proposition:

(125) Let us consider a non empty, graph-yielding function F, and a graph sum S of F. Then

(i) F is acyclic iff S is acyclic, and

(ii) F is chordal iff S is chordal.

The theorem is a consequence of (87), (42), (60), (48), and (63).

Let F be a non empty, acyclic, graph-yielding function. Let us note that every graph sum of F is acyclic.

Let F be a non empty, non acyclic, graph-yielding function. One can check that every graph sum of F is non acyclic.

Now we state the propositions:

(126) Let us consider a non empty, graph-yielding function F, and a graph sum S of F. Then $\overline{\overline{F}} \subseteq S$.numComponents(). The theorem is a consequence of (68).

(127) Let us consider a non empty, connected, graph-yielding function F, and a graph sum S of F. Then $\overline{\overline{F}} = S$.numComponents(). The theorem is a consequence of (69).

7. The Sum of two Graphs

Let G_1, G_2 be graphs.

A graph sum of G_1 and G_2 is a supergraph of G_1 defined by

(Def. 28) *it* is a graph sum of $\langle G_1, G_2 \rangle$.

Now we state the proposition:

(128) Let us consider graphs G_1, G_2, and a graph sum S of G_1 and G_2. Then

(i) G_1 is loopless and G_2 is loopless iff S is loopless, and

(ii) G_1 is non-multi and G_2 is non-multi iff S is non-multi, and

(iii) G_1 is non-directed-multi and G_2 is non-directed-multi iff S is non-directed-multi, and

(iv) G_1 is simple and G_2 is simple iff S is simple, and

(v) G_1 is directed-simple and G_2 is directed-simple iff S is directed-simple, and

(vi) G_1 is acyclic and G_2 is acyclic iff S is acyclic, and

(vii) G_1 is chordal and G_2 is chordal iff S is chordal, and

(viii) G_1 is edgeless and G_2 is edgeless iff S is edgeless, and

(ix) G_1 is loopfull and G_2 is loopfull iff S is loopfull.

The theorem is a consequence of (124).

Let G_1, G_2 be loopless graphs. Note that every graph sum of G_1 and G_2 is loopless.

Let G_1, G_2 be non loopless graphs. Let us observe that every graph sum of G_1 and G_2 is non loopless.

Let G_1, G_2 be non-directed-multi graphs. Let us note that every graph sum of G_1 and G_2 is non-directed-multi.

Let G_1, G_2 be non non-directed-multi graphs. One can verify that every graph sum of G_1 and G_2 is non non-directed-multi.

Let G_1, G_2 be non-multi graphs. Observe that every graph sum of G_1 and G_2 is non-multi.

Let G_1, G_2 be non non-multi graphs. One can check that every graph sum of G_1 and G_2 is non non-multi.

Let G_1, G_2 be simple graphs. Let us observe that every graph sum of G_1 and G_2 is simple.

Let G_1, G_2 be directed-simple graphs. Observe that every graph sum of G_1 and G_2 is directed-simple.

Let G_1, G_2 be acyclic graphs. Let us note that every graph sum of G_1 and G_2 is acyclic.

Let G_1, G_2 be non acyclic graphs. One can verify that every graph sum of G_1 and G_2 is non acyclic.

Let G_1, G_2 be edgeless graphs. Observe that every graph sum of G_1 and G_2 is edgeless.

Let G_1, G_2 be non edgeless graphs. One can check that every graph sum of G_1 and G_2 is non edgeless.

Let G_1, G_2 be loopfull graphs. Let us observe that every graph sum of G_1 and G_2 is loopfull.

Let G_1, G_2 be non loopfull graphs. Note that every graph sum of G_1 and G_2 is non loopfull.

Let us consider graphs G_1, G_2 and a graph sum S of G_1 and G_2. Now we state the propositions:

(129) $S.\mathrm{order}() = G_1.\mathrm{order}() + G_2.\mathrm{order}()$.

(130) $S.\mathrm{size}() = G_1.\mathrm{size}() + G_2.\mathrm{size}()$.

(131) $S.\mathrm{numComponents}() = G_1.\mathrm{numComponents}() + G_2.\mathrm{numComponents}()$.

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] Reinhard Diestel. *Graph theory*. Graduate Texts in Mathematics; 173. Springer, New York, 2nd edition, 2000. ISBN 0-387-98976-5; 0-387-98976-5.

[4] 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.

[5] Sebastian Koch. Miscellaneous graph preliminaries. *Formalized Mathematics*, 28(**1**):23–39, 2020. doi:10.2478/forma-2020-0003.

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

[7] Sebastian Koch. About graph unions and intersections. *Formalized Mathematics*, 28(**2**):155–171, 2020. doi:10.2478/forma-2020-0014.

[8] Gilbert Lee. Walks in graphs. *Formalized Mathematics*, 13(**2**):253–269, 2005.

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

[10] Kenneth H. Rosen, editor. *Handbook of discrete and combinatorial mathematics*. Discrete mathematics and its applications. CRC Press, Boca Raton, second edition, 2018. ISBN 978-1-58488-780-5.

[11] Klaus Wagner. *Graphentheorie*. B.I-Hochschultaschenbücher; 248. Bibliograph. Inst., Mannheim, 1970. ISBN 3-411-00248-4.

Accepted November 30, 2021