FORMALIZED MATHEMATICS
Volume 13, Number 2, Pages 279–293
University of Białystok, 2005

Weighted and Labeled Graphs[1]

Gilbert Lee[2]
University of Victoria, Victoria, Canada

Summary. In the graph framework of [17] we introduce new selectors: weights for edges and labels for both edges and vertices. We introduce also a number of tools for accessing and modifying these new fields.

MML identifier: `GLIB_003`, version: `7.5.01 4.39.921`

The articles [20], [19], [22], [14], [23], [9], [6], [15], [1], [18], [21], [7], [12], [10], [11], [3], [24], [4], [13], [2], [5], [8], [17], and [16] provide the notation and terminology for this paper.

1. Preliminaries

Let D be a set, let f_1 be a finite sequence of elements of D, and let f_2 be a FinSubsequence of f_1. Then $\mathrm{Seq}\, f_2$ is a finite sequence of elements of D.

Let F be a real-yielding binary relation and let X be a set. One can check that $F \restriction X$ is real-yielding.

Next we state two propositions:

(1) Let $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}$ be sets and p be a finite sequence. Suppose $p = \langle x_1 \rangle ^\frown \langle x_2 \rangle ^\frown \langle x_3 \rangle ^\frown \langle x_4 \rangle ^\frown \langle x_5 \rangle ^\frown \langle x_6 \rangle ^\frown \langle x_7 \rangle ^\frown \langle x_8 \rangle ^\frown \langle x_9 \rangle ^\frown \langle x_{10} \rangle$. Then $\mathrm{len}\, p = 10$ and $p(1) = x_1$ and $p(2) = x_2$ and $p(3) = x_3$ and $p(4) = x_4$ and $p(5) = x_5$ and $p(6) = x_6$ and $p(7) = x_7$ and $p(8) = x_8$ and $p(9) = x_9$ and $p(10) = x_{10}$.

(2) Let f_1 be a finite sequence of elements of $\mathbb{R}$ and f_2 be a FinSubsequence of f_1. If for every natural number i such that $i \in \mathrm{dom}\, f_1$ holds $0 \leq f_1(i)$, then $\sum \mathrm{Seq}\, f_2 \leq \sum f_1$.

[1]This work has been partially supported by NSERC, Alberta Ingenuity Fund and iCORE.
[2]Part of author's MSc work.

ISSN 1426–2630

2. Definitions

The natural number WeightSelector is defined by:

(Def. 1) WeightSelector $= 5$.

The natural number ELabelSelector is defined as follows:

(Def. 2) ELabelSelector $= 6$.

The natural number VLabelSelector is defined as follows:

(Def. 3) VLabelSelector $= 7$.

Let G be a graph structure. We say that G is weighted if and only if:

(Def. 4) WeightSelector $\in \operatorname{dom} G$ and G(WeightSelector) is a many sorted set indexed by the edges of G.

We say that G is elabeled if and only if:

(Def. 5) ELabelSelector $\in \operatorname{dom} G$ and there exists a function f such that $G(\text{ELabelSelector}) = f$ and $\operatorname{dom} f \subseteq$ the edges of G.

We say that G is vlabeled if and only if:

(Def. 6) VLabelSelector $\in \operatorname{dom} G$ and there exists a function f such that $G(\text{VLabelSelector}) = f$ and $\operatorname{dom} f \subseteq$ the vertices of G.

Let us mention that there exists a graph structure which is graph-like, weighted, elabeled, and vlabeled.

A w-graph is a weighted graph. A e-graph is a elabeled graph. A v-graph is a vlabeled graph. A we-graph is a weighted elabeled graph. A wv-graph is a weighted vlabeled graph. A ev-graph is a elabeled vlabeled graph. A wev-graph is a weighted elabeled vlabeled graph.

Let G be a w-graph. The weight of G yielding a many sorted set indexed by the edges of G is defined by:

(Def. 7) The weight of $G = G(\text{WeightSelector})$.

Let G be a e-graph. The elabel of G yields a function and is defined by:

(Def. 8) The elabel of $G = G(\text{ELabelSelector})$.

Let G be a v-graph. The vlabel of G yielding a function is defined by:

(Def. 9) The vlabel of $G = G(\text{VLabelSelector})$.

Let G be a graph and let X be a set. One can check the following observations:

* $G.\text{set}(\text{WeightSelector}, X)$ is graph-like,
* $G.\text{set}(\text{ELabelSelector}, X)$ is graph-like, and
* $G.\text{set}(\text{VLabelSelector}, X)$ is graph-like.

Let G be a finite graph and let X be a set. One can check the following observations:

* $G.\text{set}(\text{WeightSelector}, X)$ is finite,

* G.set(ELabelSelector, X) is finite, and
* G.set(VLabelSelector, X) is finite.

Let G be a loopless graph and let X be a set. One can check the following observations:

* G.set(WeightSelector, X) is loopless,
* G.set(ELabelSelector, X) is loopless, and
* G.set(VLabelSelector, X) is loopless.

Let G be a trivial graph and let X be a set. One can check the following observations:

* G.set(WeightSelector, X) is trivial,
* G.set(ELabelSelector, X) is trivial, and
* G.set(VLabelSelector, X) is trivial.

Let G be a non trivial graph and let X be a set. One can verify the following observations:

* G.set(WeightSelector, X) is non trivial,
* G.set(ELabelSelector, X) is non trivial, and
* G.set(VLabelSelector, X) is non trivial.

Let G be a non-multi graph and let X be a set. One can check the following observations:

* G.set(WeightSelector, X) is non-multi,
* G.set(ELabelSelector, X) is non-multi, and
* G.set(VLabelSelector, X) is non-multi.

Let G be a non-directed-multi graph and let X be a set. One can verify the following observations:

* G.set(WeightSelector, X) is non-directed-multi,
* G.set(ELabelSelector, X) is non-directed-multi, and
* G.set(VLabelSelector, X) is non-directed-multi.

Let G be a connected graph and let X be a set. One can check the following observations:

* G.set(WeightSelector, X) is connected,
* G.set(ELabelSelector, X) is connected, and
* G.set(VLabelSelector, X) is connected.

Let G be an acyclic graph and let X be a set. One can verify the following observations:

* G.set(WeightSelector, X) is acyclic,
* G.set(ELabelSelector, X) is acyclic, and
* G.set(VLabelSelector, X) is acyclic.

Let G be a w-graph and let X be a set. Observe that G.set(ELabelSelector, X) is weighted and G.set(VLabelSelector, X) is weighted.

Let G be a graph and let X be a many sorted set indexed by the edges of G. Note that G.set(WeightSelector, X) is weighted.

Let G be a graph, let W_1 be a non empty set, and let W be a function from the edges of G into W_1. Note that G.set(WeightSelector, W) is weighted.

Let G be a e-graph and let X be a set. Note that G.set(WeightSelector, X) is elabeled and G.set(VLabelSelector, X) is elabeled.

Let G be a graph, let Y be a set, and let X be a partial function from the edges of G to Y. One can check that G.set(ELabelSelector, X) is elabeled.

Let G be a graph and let X be a many sorted set indexed by the edges of G. One can verify that G.set(ELabelSelector, X) is elabeled.

Let G be a v-graph and let X be a set. Note that G.set(WeightSelector, X) is vlabeled and G.set(ELabelSelector, X) is vlabeled.

Let G be a graph, let Y be a set, and let X be a partial function from the vertices of G to Y. Note that G.set(VLabelSelector, X) is vlabeled.

Let G be a graph and let X be a many sorted set indexed by the vertices of G. One can verify that G.set(VLabelSelector, X) is vlabeled.

Let G be a graph. Note that G.set(ELabelSelector, $\emptyset$) is elabeled and G.set(VLabelSelector, $\emptyset$) is vlabeled.

Let G be a graph. Note that there exists a subgraph of G which is weighted, elabeled, and vlabeled.

Let G be a w-graph and let G_2 be a weighted subgraph of G. We say that G_2 inherits weight if and only if:

(Def. 10) The weight of G_2 = (the weight of G)$\restriction$(the edges of G_2).

Let G be a e-graph and let G_2 be a elabeled subgraph of G. We say that G_2 inherits elabel if and only if:

(Def. 11) The elabel of G_2 = (the elabel of G)$\restriction$(the edges of G_2).

Let G be a v-graph and let G_2 be a vlabeled subgraph of G. We say that G_2 inherits vlabel if and only if:

(Def. 12) The vlabel of G_2 = (the vlabel of G)$\restriction$(the vertices of G_2).

Let G be a w-graph. Observe that there exists a weighted subgraph of G which inherits weight.

Let G be a e-graph. One can check that there exists a elabeled subgraph of G which inherits elabel.

Let G be a v-graph. One can verify that there exists a vlabeled subgraph of G which inherits vlabel.

Let G be a we-graph. Note that there exists a weighted elabeled subgraph of G which inherits weight and elabel.

Let G be a wv-graph. Observe that there exists a weighted vlabeled subgraph of G which inherits weight and vlabel.

Let G be a ev-graph. Observe that there exists a elabeled vlabeled subgraph of G which inherits elabel and vlabel.

Let G be a wev-graph. One can verify that there exists a weighted elabeled vlabeled subgraph of G which inherits weight, elabel, and vlabel.

Let G be a w-graph. A w-subgraph of G is a weighted subgraph of G inheriting weight.

Let G be a e-graph. A e-subgraph of G is a elabeled subgraph of G inheriting elabel.

Let G be a v-graph. A v-subgraph of G is a vlabeled subgraph of G inheriting vlabel.

Let G be a we-graph. A we-subgraph of G is a weighted elabeled subgraph of G inheriting weight and elabel.

Let G be a wv-graph. A wv-subgraph of G is a weighted vlabeled subgraph of G inheriting weight and vlabel.

Let G be a ev-graph. A ev-subgraph of G is a elabeled vlabeled subgraph of G inheriting elabel and vlabel.

Let G be a wev-graph. A wev-subgraph of G is a weighted elabeled vlabeled subgraph of G inheriting weight, elabel, and vlabel.

Let G be a graph and let V, E be sets. One can verify that there exists a subgraph of G induced by V and E which is weighted, elabeled, and vlabeled.

Let G be a w-graph and let V, E be sets. One can verify that there exists a weighted subgraph of G induced by V and E which inherits weight.

Let G be a e-graph and let V, E be sets. One can verify that there exists a elabeled subgraph of G induced by V and E which inherits elabel.

Let G be a v-graph and let V, E be sets. One can verify that there exists a vlabeled subgraph of G induced by V and E which inherits vlabel.

Let G be a we-graph and let V, E be sets. Note that there exists a weighted elabeled subgraph of G induced by V and E which inherits weight and elabel.

Let G be a wv-graph and let V, E be sets. Observe that there exists a weighted vlabeled subgraph of G induced by V and E which inherits weight and vlabel.

Let G be a ev-graph and let V, E be sets. Note that there exists a elabeled vlabeled subgraph of G induced by V and E which inherits elabel and vlabel.

Let G be a wev-graph and let V, E be sets. Observe that there exists a weighted elabeled vlabeled subgraph of G induced by V and E which inherits weight, elabel, and vlabel.

Let G be a w-graph and let V, E be sets. A induced w-subgraph of G, V, E is a weighted subgraph of G induced by V and E inheriting weight.

Let G be a e-graph and let V, E be sets. A induced e-subgraph of G, V, E is a elabeled subgraph of G induced by V and E inheriting elabel.

Let G be a v-graph and let V, E be sets. A induced v-subgraph of G, V, E is a vlabeled subgraph of G induced by V and E inheriting vlabel.

Let G be a we-graph and let V, E be sets. A induced we-subgraph of G, V, E is a weighted elabeled subgraph of G induced by V and E inheriting weight and elabel.

Let G be a wv-graph and let V, E be sets. A induced wv-subgraph of G, V, E is a weighted vlabeled subgraph of G induced by V and E inheriting weight and vlabel.

Let G be a ev-graph and let V, E be sets. A induced ev-subgraph of G, V, E is a elabeled vlabeled subgraph of G induced by V and E inheriting elabel and vlabel.

Let G be a wev-graph and let V, E be sets. A induced wev-subgraph of G, V, E is a weighted elabeled vlabeled subgraph of G induced by V and E inheriting weight, elabel, and vlabel.

Let G be a w-graph and let V be a set. A induced w-subgraph of G, V is a induced w-subgraph of G, V, G.edgesBetween(V).

Let G be a e-graph and let V be a set. A induced e-subgraph of G, V is a induced e-subgraph of G, V, G.edgesBetween(V).

Let G be a v-graph and let V be a set. A induced v-subgraph of G, V is a induced v-subgraph of G, V, G.edgesBetween(V).

Let G be a we-graph and let V be a set. A induced we-subgraph of G, V is a induced we-subgraph of G, V, G.edgesBetween(V).

Let G be a wv-graph and let V be a set. A induced wv-subgraph of G, V is a induced wv-subgraph of G, V, G.edgesBetween(V).

Let G be a ev-graph and let V be a set. A induced ev-subgraph of G, V is a induced ev-subgraph of G, V, G.edgesBetween(V).

Let G be a wev-graph and let V be a set. A induced wev-subgraph of G, V is a induced wev-subgraph of G, V, G.edgesBetween(V).

Let G be a w-graph. We say that G is real-weighted if and only if:

(Def. 13) The weight of G is real-yielding.

Let G be a w-graph. We say that G is nonnegative-weighted if and only if:

(Def. 14) $\operatorname{rng}(\text{the weight of } G) \subseteq \mathbb{R}_{\geq 0}$.

Let us note that every w-graph which is nonnegative-weighted is also real-weighted.

Let G be a e-graph. We say that G is real-elabeled if and only if:

(Def. 15) The elabel of G is real-yielding.

Let G be a v-graph. We say that G is real-vlabeled if and only if:

(Def. 16) The vlabel of G is real-yielding.

Let G be a wev-graph. We say that G is real-wev if and only if:

(Def. 17) G is real-weighted, real-elabeled, and real-vlabeled.

Let us note that every wev-graph which is real-wev is also real-weighted, real-elabeled, and real-vlabeled and every wev-graph which is real-weighted, real-elabeled, and real-vlabeled is also real-wev.

Let G be a graph and let X be a function from the edges of G into $\mathbb{R}$. Note that G.set(WeightSelector, X) is real-weighted.

Let G be a graph and let X be a partial function from the edges of G to $\mathbb{R}$. One can verify that G.set(ELabelSelector, X) is real-elabeled.

Let G be a graph and let X be a real-yielding many sorted set indexed by the edges of G. One can verify that G.set(ELabelSelector, X) is real-elabeled.

Let G be a graph and let X be a partial function from the vertices of G to $\mathbb{R}$. Observe that G.set(VLabelSelector, X) is real-vlabeled.

Let G be a graph and let X be a real-yielding many sorted set indexed by the vertices of G. One can verify that G.set(VLabelSelector, X) is real-vlabeled.

Let G be a graph. Observe that G.set(ELabelSelector, $\emptyset$) is real-elabeled and G.set(VLabelSelector, $\emptyset$) is real-vlabeled.

Let G be a graph, let v be a vertex of G, and let v_1 be a real number. Note that G.set(VLabelSelector, $v \dot{\longmapsto} v_1$) is vlabeled.

Let G be a graph, let v be a vertex of G, and let v_1 be a real number. One can verify that G.set(VLabelSelector, $v \dot{\longmapsto} v_1$) is real-vlabeled.

One can check that there exists a wev-graph which is finite, trivial, tree-like, nonnegative-weighted, and real-wev and there exists a wev-graph which is finite, non trivial, tree-like, nonnegative-weighted, and real-wev.

Let G be a finite w-graph. Note that the weight of G is finite.

Let G be a finite e-graph. Note that the elabel of G is finite.

Let G be a finite v-graph. Note that the vlabel of G is finite.

Let G be a real-weighted w-graph. Observe that the weight of G is real-yielding.

Let G be a real-elabeled e-graph. One can verify that the elabel of G is real-yielding.

Let G be a real-vlabeled v-graph. Observe that the vlabel of G is real-yielding.

Let G be a real-weighted w-graph and let X be a set. Observe that G.set(ELabelSelector, X) is real-weighted and G.set(VLabelSelector, X) is real-weighted.

Let G be a nonnegative-weighted w-graph and let X be a set. One can check that G.set(ELabelSelector, X) is nonnegative-weighted and G.set(VLabelSelector, X) is nonnegative-weighted.

Let G be a real-elabeled e-graph and let X be a set. One can verify that G.set(WeightSelector, X) is real-elabeled and G.set(VLabelSelector, X) is real-elabeled.

Let G be a real-vlabeled v-graph and let X be a set. Observe that G.set(WeightSelector, X) is real-vlabeled and G.set(ELabelSelector, X) is real-vlabeled.

Let G be a w-graph and let W be a walk of G. The functor W.weightSeq() yielding a finite sequence is defined as follows:

(Def. 18) len(W.weightSeq()) = len(W.edgeSeq()) and for every natural number n such that $1 \leq n$ and $n \leq$ len(W.weightSeq()) holds W.weightSeq()(n) = (the weight of G)(W.edgeSeq()(n)).

Let G be a real-weighted w-graph and let W be a walk of G. Then W.weightSeq() is a finite sequence of elements of $\mathbb{R}$.

Let G be a real-weighted w-graph and let W be a walk of G. The functor W.cost() yielding a real number is defined as follows:

(Def. 19) W.cost() = $\sum$(W.weightSeq()).

Let G be a e-graph. The functor G.labeledE() yields a subset of the edges of G and is defined as follows:

(Def. 20) G.labeledE() = dom (the elabel of G).

Let G be a e-graph and let e, x be sets. The functor G.labelEdge(e, x) yielding a e-graph is defined as follows:

(Def. 21) $G.\mathrm{labelEdge}(e,x) = \begin{cases} G.\mathrm{set}(\mathrm{ELabelSelector}, (\text{the elabel of } G) {+}\!\cdot (e \,\dot{\longmapsto}\, x)), \\ \quad \text{if } e \in \text{the edges of } G, \\ G, \text{ otherwise.} \end{cases}$

Let G be a finite e-graph and let e, x be sets. Note that G.labelEdge(e, x) is finite.

Let G be a loopless e-graph and let e, x be sets. Observe that G.labelEdge(e, x) is loopless.

Let G be a trivial e-graph and let e, x be sets. One can check that G.labelEdge(e, x) is trivial.

Let G be a non trivial e-graph and let e, x be sets. One can verify that G.labelEdge(e, x) is non trivial.

Let G be a non-multi e-graph and let e, x be sets. Observe that G.labelEdge(e, x) is non-multi.

Let G be a non-directed-multi e-graph and let e, x be sets. One can check that G.labelEdge(e, x) is non-directed-multi.

Let G be a connected e-graph and let e, x be sets. Observe that G.labelEdge(e, x) is connected.

Let G be an acyclic e-graph and let e, x be sets. Observe that G.labelEdge(e, x) is acyclic.

Let G be a we-graph and let e, x be sets. Observe that G.labelEdge(e, x) is weighted.

Let G be a ev-graph and let e, x be sets. Note that G.labelEdge(e, x) is vlabeled.

Let G be a real-weighted we-graph and let e, x be sets. Observe that G.labelEdge(e, x) is real-weighted.

Let G be a nonnegative-weighted we-graph and let e, x be sets. Observe that G.labelEdge(e, x) is nonnegative-weighted.

Let G be a real-elabeled e-graph, let e be a set, and let x be a real number. Observe that G.labelEdge(e, x) is real-elabeled.

Let G be a real-vlabeled ev-graph and let e, x be sets. Note that G.labelEdge(e, x) is real-vlabeled.

Let G be a v-graph and let v, x be sets. The functor G.labelVertex(v, x) yielding a v-graph is defined as follows:

(Def. 22) $$G.\mathrm{labelVertex}(v, x) = \begin{cases} G.\mathrm{set}(\mathrm{VLabelSelector}, \\ \quad (\text{the vlabel of } G) +\cdot (v \dot{\longmapsto} x)), \\ \quad \text{if } v \in \text{the vertices of } G, \\ G, \text{ otherwise.} \end{cases}$$

Let G be a v-graph. The functor G.labeledV() yielding a subset of the vertices of G is defined as follows:

(Def. 23) G.labeledV() $=$ dom (the vlabel of G).

Let G be a finite v-graph and let v, x be sets. One can check that G.labelVertex(v, x) is finite.

Let G be a loopless v-graph and let v, x be sets. One can check that G.labelVertex(v, x) is loopless.

Let G be a trivial v-graph and let v, x be sets. One can check that G.labelVertex(v, x) is trivial.

Let G be a non trivial v-graph and let v, x be sets. Observe that G.labelVertex(v, x) is non trivial.

Let G be a non-multi v-graph and let v, x be sets. Note that G.labelVertex(v, x) is non-multi.

Let G be a non-directed-multi v-graph and let v, x be sets. One can verify that G.labelVertex(v, x) is non-directed-multi.

Let G be a connected v-graph and let v, x be sets. Observe that G.labelVertex(v, x) is connected.

Let G be an acyclic v-graph and let v, x be sets. Note that G.labelVertex(v, x) is acyclic.

Let G be a wv-graph and let v, x be sets. One can check that G.labelVertex(v, x) is weighted.

Let G be a ev-graph and let v, x be sets. Observe that G.labelVertex(v, x) is elabeled.

Let G be a real-weighted wv-graph and let v, x be sets. Observe that G.labelVertex(v, x) is real-weighted.

Let G be a nonnegative-weighted wv-graph and let v, x be sets. Note that G.labelVertex(v, x) is nonnegative-weighted.

Let G be a real-elabeled ev-graph and let v, x be sets. Observe that G.labelVertex(v, x) is real-elabeled.

Let G be a real-vlabeled v-graph, let v be a set, and let x be a real number. Note that G.labelVertex(v, x) is real-vlabeled.

Let G be a real-weighted w-graph. Observe that every w-subgraph of G is real-weighted.

Let G be a nonnegative-weighted w-graph. Observe that every w-subgraph of G is nonnegative-weighted.

Let G be a real-elabeled e-graph. Observe that every e-subgraph of G is real-elabeled.

Let G be a real-vlabeled v-graph. Observe that every v-subgraph of G is real-vlabeled.

Let G_1 be a graph sequence. We say that G_1 is weighted if and only if:

(Def. 24) For every natural number x holds $G_1 \rightarrow x$ is weighted.

We say that G_1 is elabeled if and only if:

(Def. 25) For every natural number x holds $G_1 \rightarrow x$ is elabeled.

We say that G_1 is vlabeled if and only if:

(Def. 26) For every natural number x holds $G_1 \rightarrow x$ is vlabeled.

Let us mention that there exists a graph sequence which is weighted, elabeled, and vlabeled.

A w-graph sequence is a weighted graph sequence. A e-graph sequence is a elabeled graph sequence. A v-graph sequence is a vlabeled graph sequence. A we-graph sequence is a weighted elabeled graph sequence. A wv-graph sequence is a weighted vlabeled graph sequence. A ev-graph sequence is a elabeled vlabeled graph sequence. A wev-graph sequence is a weighted elabeled vlabeled graph sequence.

Let G_1 be a w-graph sequence and let x be a natural number. One can check that $G_1 \rightarrow x$ is weighted.

Let G_1 be a e-graph sequence and let x be a natural number. One can check that $G_1 \rightarrow x$ is elabeled.

Let G_1 be a v-graph sequence and let x be a natural number. Observe that $G_1 \rightarrow x$ is vlabeled.

Let G_1 be a w-graph sequence. We say that G_1 is real-weighted if and only if:

(Def. 27) For every natural number x holds $G_1 \rightarrow x$ is real-weighted.

We say that G_1 is nonnegative-weighted if and only if:

(Def. 28) For every natural number x holds $G_1 \rightarrow x$ is nonnegative-weighted.

Let G_1 be a e-graph sequence. We say that G_1 is real-elabeled if and only if:

(Def. 29) For every natural number x holds $G_1 \rightarrow x$ is real-elabeled.

Let G_1 be a v-graph sequence. We say that G_1 is real-vlabeled if and only if:

(Def. 30) For every natural number x holds $G_1 \to x$ is real-vlabeled.

Let G_1 be a wev-graph sequence. We say that G_1 is real-wev if and only if:

(Def. 31) For every natural number x holds $G_1 \to x$ is real-wev.

Let us note that every wev-graph sequence which is real-wev is also real-weighted, real-elabeled, and real-vlabeled and every wev-graph sequence which is real-weighted, real-elabeled, and real-vlabeled is also real-wev.

Let us observe that there exists a wev-graph sequence which is halting, finite, loopless, trivial, non-multi, simple, real-wev, nonnegative-weighted, and tree-like.

Let G_1 be a real-weighted w-graph sequence and let x be a natural number. One can check that $G_1 \to x$ is real-weighted.

Let G_1 be a nonnegative-weighted w-graph sequence and let x be a natural number. Observe that $G_1 \to x$ is nonnegative-weighted.

Let G_1 be a real-elabeled e-graph sequence and let x be a natural number. Note that $G_1 \to x$ is real-elabeled.

Let G_1 be a real-vlabeled v-graph sequence and let x be a natural number. One can verify that $G_1 \to x$ is real-vlabeled.

3. Theorems

The following propositions are true:

(3) WeightSelector $= 5$ and ELabelSelector $= 6$ and VLabelSelector $= 7$.

(4)(i) For every w-graph G holds the weight of $G = G$(WeightSelector),

(ii) for every e-graph G holds the elabel of $G = G$(ELabelSelector), and

(iii) for every v-graph G holds the vlabel of $G = G$(VLabelSelector).

(6)[3] For every e-graph G holds dom (the elabel of G) $\subseteq$ the edges of G.

(7) For every v-graph G holds dom (the vlabel of G) $\subseteq$ the vertices of G.

(8) For every graph G and for every set X holds $G =_G G$.set(WeightSelector, X) and $G =_G G$.set(ELabelSelector, X) and $G =_G G$.set(VLabelSelector, X).

(9) For every e-graph G and for every set X holds the elabel of $G =$ the elabel of G.set(WeightSelector, X).

(10) For every v-graph G and for every set X holds the vlabel of $G =$ the vlabel of G.set(WeightSelector, X).

(11) For every w-graph G and for every set X holds the weight of $G =$ the weight of G.set(ELabelSelector, X).

(12) For every v-graph G and for every set X holds the vlabel of $G =$ the vlabel of G.set(ELabelSelector, X).

[3]The proposition (5) has been removed.

(13) For every w-graph G and for every set X holds the weight of G = the weight of G.set(VLabelSelector, X).

(14) For every e-graph G and for every set X holds the elabel of G = the elabel of G.set(VLabelSelector, X).

(15) Let G_3, G_2 be w-graphs and G_4 be a w-graph. Suppose $G_3 =_G G_2$ and the weight of G_3 = the weight of G_2 and G_3 is a w-subgraph of G_4. Then G_2 is a w-subgraph of G_4.

(16) For every w-graph G_3 and for every w-subgraph G_2 of G_3 holds every w-subgraph of G_2 is a w-subgraph of G_3.

(17) Let G_3, G_2 be w-graphs and G_4 be a w-subgraph of G_3. Suppose $G_3 =_G G_2$ and the weight of G_3 = the weight of G_2. Then G_4 is a w-subgraph of G_2.

(18) Let G_3 be a w-graph, G_2 be a w-subgraph of G_3, and x be a set. If $x \in$ the edges of G_2, then (the weight of G_2)(x) = (the weight of G_3)(x).

(19) For every w-graph G and for every walk W of G such that W is trivial holds W.weightSeq() = $\emptyset$.

(20) For every w-graph G and for every walk W of G holds len(W.weightSeq()) = W.length().

(21) For every w-graph G and for all sets x, y, e such that e joins x and y in G holds (G.walkOf(x, e, y)).weightSeq() = $\langle$(the weight of G)(e)$\rangle$.

(22) For every w-graph G and for every walk W of G holds W.reverse().weightSeq() = Rev(W.weightSeq()).

(23) For every w-graph G and for all walks W_2, W_3 of G such that W_2.last() = W_3.first() holds (W_2.append(W_3)).weightSeq() = W_2.weightSeq() $\frown$ W_3.weightSeq().

(24) Let G be a w-graph, W be a walk of G, and e be a set. If $e \in W$.last().edgesInOut(), then (W.addEdge(e)).weightSeq() = W.weightSeq() $\frown$ $\langle$(the weight of G)(e)$\rangle$.

(25) Let G be a real-weighted w-graph, W_2 be a walk of G, and W_3 be a subwalk of W_2. Then there exists a FinSubsequence w_1 of W_2.weightSeq() such that W_3.weightSeq() = Seq w_1.

(26) Let G_3, G_2 be w-graphs, W_2 be a walk of G_3, and W_3 be a walk of G_2. If $W_2 = W_3$ and the weight of G_3 = the weight of G_2, then W_2.weightSeq() = W_3.weightSeq().

(27) Let G_3 be a w-graph, G_2 be a w-subgraph of G_3, W_2 be a walk of G_3, and W_3 be a walk of G_2. If $W_2 = W_3$, then W_2.weightSeq() = W_3.weightSeq().

(28) For every real-weighted w-graph G and for every walk W of G such that W is trivial holds W.cost() = 0.

(29) Let G be a real-weighted w-graph, v_2, v_3 be vertices of G, and e be a set.

If e joins v_2 and v_3 in G, then $(G.\mathrm{walkOf}(v_2, e, v_3)).\mathrm{cost}() =$ (the weight of $G)(e)$.

(30) For every real-weighted w-graph G and for every walk W of G holds $W.\mathrm{cost}() = W.\mathrm{reverse}().\mathrm{cost}()$.

(31) For every real-weighted w-graph G and for all walks W_2, W_3 of G such that $W_2.\mathrm{last}() = W_3.\mathrm{first}()$ holds $(W_2.\mathrm{append}(W_3)).\mathrm{cost}() = W_2.\mathrm{cost}() + W_3.\mathrm{cost}()$.

(32) Let G be a real-weighted w-graph, W be a walk of G, and e be a set. If $e \in W.\mathrm{last}().\mathrm{edgesInOut}()$, then $(W.\mathrm{addEdge}(e)).\mathrm{cost}() = W.\mathrm{cost}() +$ (the weight of $G)(e)$.

(33) Let G_3, G_2 be real-weighted w-graphs, W_2 be a walk of G_3, and W_3 be a walk of G_2. If $W_2 = W_3$ and the weight of $G_3 =$ the weight of G_2, then $W_2.\mathrm{cost}() = W_3.\mathrm{cost}()$.

(34) Let G_3 be a real-weighted w-graph, G_2 be a w-subgraph of G_3, W_2 be a walk of G_3, and W_3 be a walk of G_2. If $W_2 = W_3$, then $W_2.\mathrm{cost}() = W_3.\mathrm{cost}()$.

(35) Let G be a nonnegative-weighted w-graph, W be a walk of G, and n be a natural number. If $n \in \mathrm{dom}(W.\mathrm{weightSeq}())$, then $0 \leq W.\mathrm{weightSeq}()(n)$.

(36) For every nonnegative-weighted w-graph G and for every walk W of G holds $0 \leq W.\mathrm{cost}()$.

(37) For every nonnegative-weighted w-graph G and for every walk W_2 of G and for every subwalk W_3 of W_2 holds $W_3.\mathrm{cost}() \leq W_2.\mathrm{cost}()$.

(38) Let G be a nonnegative-weighted w-graph and e be a set. If $e \in$ the edges of G, then $0 \leq$ (the weight of $G)(e)$.

(39) Let G be a e-graph and e, x be sets. Suppose $e \in$ the edges of G. Then the elabel of $G.\mathrm{labelEdge}(e, x) =$ (the elabel of $G) {+\cdot} (e \mapsto x)$.

(40) For every e-graph G and for all sets e, x such that $e \in$ the edges of G holds (the elabel of $G.\mathrm{labelEdge}(e, x))(e) = x$.

(41) For every e-graph G and for all sets e, x holds $G =_G G.\mathrm{labelEdge}(e, x)$.

(42) For every we-graph G and for all sets e, x holds the weight of $G =$ the weight of $G.\mathrm{labelEdge}(e, x)$.

(43) For every ev-graph G and for all sets e, x holds the vlabel of $G =$ the vlabel of $G.\mathrm{labelEdge}(e, x)$.

(44) For every e-graph G and for all sets e_1, e_2, x such that $e_1 \neq e_2$ holds (the elabel of $G.\mathrm{labelEdge}(e_1, x))(e_2) =$ (the elabel of $G)(e_2)$.

(45) Let G be a v-graph and v, x be sets. Suppose $v \in$ the vertices of G. Then the vlabel of $G.\mathrm{labelVertex}(v, x) =$ (the vlabel of $G) {+\cdot} (v \mapsto x)$.

(46) For every v-graph G and for all sets v, x such that $v \in$ the vertices of G holds (the vlabel of $G.\mathrm{labelVertex}(v, x))(v) = x$.

(47) For every v-graph G and for all sets v, x holds $G =_G G$.labelVertex(v, x).

(48) For every wv-graph G and for all sets v, x holds the weight of $G =$ the weight of G.labelVertex(v, x).

(49) For every ev-graph G and for all sets v, x holds the elabel of $G =$ the elabel of G.labelVertex(v, x).

(50) For every v-graph G and for all sets v_2, v_3, x such that $v_2 \neq v_3$ holds (the vlabel of G.labelVertex$(v_2, x))(v_3) =$ (the vlabel of $G)(v_3)$.

(51) For all e-graphs G_3, G_2 such that the elabel of $G_3 =$ the elabel of G_2 holds G_3.labeledE() $= G_2$.labeledE().

(52) For every e-graph G and for all sets e, x such that $e \in$ the edges of G holds (G.labelEdge(e, x)).labeledE() $= G$.labeledE() $\cup \{e\}$.

(53) For every e-graph G and for all sets e, x such that $e \in$ the edges of G holds G.labeledE() $\subseteq$ (G.labelEdge(e, x)).labeledE().

(54) For every finite e-graph G and for all sets e, x such that $e \in$ the edges of G and $e \notin G$.labeledE() holds card((G.labelEdge(e, x)).labeledE()) $=$ card(G.labeledE()) $+ 1$.

(55) For every e-graph G and for all sets e_1, e_2, x such that $e_2 \notin G$.labeledE() and $e_2 \in$ (G.labelEdge(e_1, x)).labeledE() holds $e_1 = e_2$ and $e_1 \in$ the edges of G.

(56) For every ev-graph G and for all sets v, x holds G.labeledE() $=$ (G.labelVertex(v, x)).labeledE().

(57) For every e-graph G and for all sets e, x such that $e \in$ the edges of G holds $e \in$ (G.labelEdge(e, x)).labeledE().

(58) For all v-graphs G_3, G_2 such that the vlabel of $G_3 =$ the vlabel of G_2 holds G_3.labeledV() $= G_2$.labeledV().

(59) For every v-graph G and for all sets v, x such that $v \in$ the vertices of G holds (G.labelVertex(v, x)).labeledV() $= G$.labeledV() $\cup \{v\}$.

(60) For every v-graph G and for all sets v, x such that $v \in$ the vertices of G holds G.labeledV() $\subseteq$ (G.labelVertex(v, x)).labeledV().

(61) For every finite v-graph G and for all sets v, x such that $v \in$ the vertices of G and $v \notin G$.labeledV() holds card((G.labelVertex(v, x)).labeledV()) $=$ card(G.labeledV()) $+ 1$.

(62) For every v-graph G and for all sets v_2, v_3, x such that $v_3 \notin G$.labeledV() and $v_3 \in$ (G.labelVertex(v_2, x)).labeledV() holds $v_2 = v_3$ and $v_2 \in$ the vertices of G.

(63) For every ev-graph G and for all sets e, x holds G.labeledV() $=$ (G.labelEdge(e, x)).labeledV().

(64) For every v-graph G and for every vertex v of G and for every set x holds $v \in$ (G.labelVertex(v, x)).labeledV().

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.
[2] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[3] Grzegorz Bancerek. The ordinal numbers. *Formalized Mathematics*, 1(**1**):91–96, 1990.
[4] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[5] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.
[6] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[7] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[8] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[9] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.
[10] Czesław Byliński. The sum and product of finite sequences of real numbers. *Formalized Mathematics*, 1(**4**):661–668, 1990.
[11] Czesław Byliński. Some properties of restrictions of finite sequences. *Formalized Mathematics*, 5(**2**):241–245, 1996.
[12] Jing-Chao Chen and Yatsuka Nakamura. The underlying principle of Dijkstra's shortest path algorithm. *Formalized Mathematics*, 11(**2**):143–152, 2003.
[13] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[14] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[15] Jarosław Kotowicz. Real sequences and basic operations on them. *Formalized Mathematics*, 1(**2**):269–272, 1990.
[16] Gilbert Lee. Walks in Graphs. *Formalized Mathematics*, 13(**2**):253–269, 2005.
[17] Gilbert Lee and Piotr Rudnicki. Alternative graph structures. *Formalized Mathematics*, 13(**2**):235–252, 2005.
[18] Yatsuka Nakamura and Piotr Rudnicki. Vertex sequences induced by chains. *Formalized Mathematics*, 5(**3**):297–304, 1996.
[19] Andrzej Trybulec. Subsets of complex numbers. *To appear in Formalized Mathematics.*
[20] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[21] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[22] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[23] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[24] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received February 8, 2005