JOURNAL OF FORMALIZED MATHEMATICS
Volume 15, Released 2003, Published 2003
Inst. of Computer Science, Univ. of Białystok

Dijkstra's Shortest Path Algorithm

Jing-Chao Chen
Donghua University
Shanghai

Summary. The article formalizes Dijkstra's shortest path algorithm [11]. A path from a source vertex v to a target vertex u is said to be the shortest path if its total cost is minimum among all v-to-u paths. Dijkstra's algorithm is based on the following assumptions:

- All edge costs are non-negative.
- The number of vertices is finite.
- The source is a single vertex, but the target may be all other vertices.

The underlying principle of the algorithm may be described as follows: the algorithm starts with the source; it visits the vertices in order of increasing cost, and maintains a set V of visited vertices (denoted by UsedVx in the article) whose cost from the source has been computed, and a tentative cost $D(u)$ to each unvisited vertex u. In the article, the set of all unvisited vertices is denoted by UnusedVx. $D(u)$ is the cost of the shortest path from the source to u in the subgraph induced by $V \cup \{u\}$. We denote the set of all unvisited vertices whose D-values are not infinite (i.e. in the subgraph each of which has a path from the source to itself) by OuterVx. Dijkstra's algorithm repeatedly searches OuterVx for the vertex with minimum tentative cost (this procedure is called findmin in the article), adds it to the set V and modifies D-values by a procedure, called Relax. Suppose the unvisited vertex with minimum tentative cost is x, the procedure Relax replaces $D(u)$ with $\min\{D(u), D(u) + cost(x,u)\}$ where u is a vertex in UnusedVx, and cost(x,u) is the cost of edge (x,u). In the Mizar library, there are several computer models, e.g. SCMFSA and SCMPDS etc. However, it is extremely difficult to use these models to formalize the algorithm. Instead, we adopt functions in the Mizar library, which seem to be pseudo-codes, and are similar to those in the functional programming language, e.g. Lisp. To date, there is no rigorous justification with respect to the correctness of Dijkstra's algorithm. The article presents first the rigorous justification.

MML Identifier: `GRAPHSP`.

WWW: `http://mizar.org/JFM/Vol15/graphsp.html`

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

1. PRELIMINARIES

For simplicity, we follow the rules: X denotes a set, i, j, k, m, n denote natural numbers, p denotes a finite sequence of elements of X, and i_1 denotes an integer.

Next we state three propositions:

(1) For every finite sequence p and for every set x holds $x \notin \mathrm{rng}\, p$ and p is one-to-one iff $p \frown \langle x \rangle$ is one-to-one.

(2) If $1 \leq i_1$ and $i_1 \leq \mathrm{len}\, p$, then $p(i_1) \in X$.

(3) If $1 \leq i_1$ and $i_1 \leq \operatorname{len} p$, then $p_{i_1} = p(i_1)$.

For simplicity, we follow the rules: G denotes a graph, p_1, q_1 denote finite sequences of elements of the edges of G, p, q denote oriented chains of G, W denotes a function, U, V, e, e_1 denote sets, and v_1, v_2, v_3, v_4 denote vertices of G.

The following three propositions are true:

(4) If W is weight of G and $\operatorname{len} p_1 = 1$, then $\operatorname{cost}(p_1, W) = W(p_1(1))$.

(5) If $e \in$ the edges of G, then $\langle e \rangle$ is a Simple oriented chain of G.

(6) Let p be a Simple oriented chain of G. Suppose $p = p_1 \frown q_1$ and $\operatorname{len} p_1 \geq 1$ and $\operatorname{len} q_1 \geq 1$. Then (the target of $G)(p(\operatorname{len} p)) \neq$ (the target of $G)(p_1(\operatorname{len} p_1))$ and (the source of $G)(p(1)) \neq$ (the source of $G)(q_1(1))$.

2. The Fundamental Properties of Directed Paths and Shortest Paths

We now state several propositions:

(7) p is oriented path from v_1 to v_2 in V iff p is oriented path from v_1 to v_2 in $V \cup \{v_2\}$.

(8) p is shortest path from v_1 to v_2 in V w.r.t. W iff p is shortest path from v_1 to v_2 in $V \cup \{v_2\}$ w.r.t. W.

(9) Suppose p is shortest path from v_1 to v_2 in V w.r.t. W and q is shortest path from v_1 to v_2 in V w.r.t. W. Then $\operatorname{cost}(p, W) = \operatorname{cost}(q, W)$.

(10) Let G be an oriented graph, v_1, v_2 be vertices of G, and e_2, e_3 be sets. Suppose $e_2 \in$ the edges of G and $e_3 \in$ the edges of G and e_2 orientedly joins v_1, v_2 and e_3 orientedly joins v_1, v_2. Then $e_2 = e_3$.

(11) Suppose that

(i) the vertices of $G = U \cup V$,

(ii) $v_1 \in U$,

(iii) $v_2 \in V$, and

(iv) for all v_3, v_4 such that $v_3 \in U$ and $v_4 \in V$ it is not true that there exists e such that $e \in$ the edges of G and e orientedly joins v_3, v_4.

Then there exists no p which is oriented path from v_1 to v_2.

(12) Suppose that

(i) the vertices of $G = U \cup V$,

(ii) $v_1 \in U$,

(iii) for all v_3, v_4 such that $v_3 \in U$ and $v_4 \in V$ it is not true that there exists e such that $e \in$ the edges of G and e orientedly joins v_3, v_4, and

(iv) p is oriented path from v_1 to v_2.

Then p is oriented path from v_1 to v_2 in U.

3. The Basic Theorems for Dijkstra's Shortest Path Algorithm (continue)

We adopt the following convention: G is a finite graph, P, Q are oriented chains of G, and v_1, v_2, v_3 are vertices of G.

One can prove the following proposition

(13) Suppose that W is nonnegative weight of G and P is shortest path from v_1 to v_2 in V w.r.t. W and $v_1 \neq v_2$ and $v_1 \neq v_3$ and Q is shortest path from v_1 to v_3 in V w.r.t. W and it is not true that there exists e such that $e \in$ the edges of G and e orientedly joins v_2, v_3 and P is longest in shortest path from v_1 in V w.r.t. W. Then Q is shortest path from v_1 to v_3 in $V \cup \{v_2\}$ w.r.t. W.

For simplicity, we adopt the following rules: G is a finite oriented graph, P, Q are oriented chains of G, W is a function from the edges of G into $\mathbb{R}_{\geq 0}$, and v_1, v_2, v_3, v_4 are vertices of G.

One can prove the following propositions:

(14) Suppose $e \in$ the edges of G and $v_1 \neq v_2$ and $P = \langle e \rangle$ and e orientedly joins v_1, v_2. Then P is shortest path from v_1 to v_2 in $\{v_1\}$ w.r.t. W.

(15) Suppose that $e \in$ the edges of G and P is shortest path from v_1 to v_2 in V w.r.t. W and $v_1 \neq v_3$ and $Q = P \frown \langle e \rangle$ and e orientedly joins v_2, v_3 and $v_1 \in V$ and for every v_4 such that $v_4 \in V$ it is not true that there exists e_1 such that $e_1 \in$ the edges of G and e_1 orientedly joins v_4, v_3. Then Q is shortest path from v_1 to v_3 in $V \cup \{v_2\}$ w.r.t. W.

(16) Suppose that

(i) the vertices of $G = U \cup V$,

(ii) $v_1 \in U$, and

(iii) for all v_3, v_4 such that $v_3 \in U$ and $v_4 \in V$ it is not true that there exists e such that $e \in$ the edges of G and e orientedly joins v_3, v_4.

Then P is shortest path from v_1 to v_2 in U w.r.t. W if and only if P is shortest path from v_1 to v_2 in W.

4. The Definition of Assignment Statement

Let f be a function and let i, x be sets. We introduce $f_i{:=}x$ as a synonym of $f + \cdot (i, x)$.

We now state the proposition

(17) For all sets x, y and for every function f holds $\operatorname{rng}(f_x{:=}y) \subseteq \operatorname{rng} f \cup \{y\}$.

Let f be a finite sequence of elements of $\mathbb{R}$, let x be a set, and let r be a real number. Then $f_x{:=}r$ is a finite sequence of elements of $\mathbb{R}$.

Let i, k be natural numbers, let f be a finite sequence of elements of $\mathbb{R}$, and let r be a real number. The functor $(f, i) := (k, r)$ yielding a finite sequence of elements of $\mathbb{R}$ is defined as follows:

(Def. 1) $(f, i) := (k, r) = f_i{:=}k_k{:=}r$.

In the sequel f, g, h denote elements of $\mathbb{R}^*$ and r denotes a real number.

One can prove the following propositions:

(18) If $i \neq k$ and $i \in \operatorname{dom} f$, then $((f, i) := (k, r))(i) = k$.

(19) If $m \neq i$ and $m \neq k$ and $m \in \operatorname{dom} f$, then $((f, i) := (k, r))(m) = f(m)$.

(20) If $k \in \operatorname{dom} f$, then $((f, i) := (k, r))(k) = r$.

(21) $\operatorname{dom}((f, i) := (k, r)) = \operatorname{dom} f$.

5. The Definition of Pascal–Like "while" - "do" Statement

Let X be a set. Then id_X is an element of X^X.

Let X be a set and let f, g be functions from X into X. Then $g \cdot f$ is a function from X into X.

Let X be a set and let f, g be elements of X^X. Then $g \cdot f$ is an element of X^X.

Let X be a set, let f be an element of X^X, and let g be an element of X. Then $f(g)$ is an element of X.

Let X be a set and let f be an element of X^X. The functor repeat f yields a function from $\mathbb{N}$ into X^X and is defined as follows:

(Def. 2) $(\mathrm{repeat} f)(0) = \mathrm{id}_X$ and for every natural number i holds $(\mathrm{repeat} f)(i+1) = f \cdot (\mathrm{repeat} f)(i)$.

We now state two propositions:

(22) For every element F of $(\mathbb{R}^*)^{\mathbb{R}^*}$ and for every element f of $\mathbb{R}^*$ and for all natural numbers n, i holds $(\mathrm{repeat} F)(0)(f) = f$.

(23) Let F, G be elements of $(\mathbb{R}^*)^{\mathbb{R}^*}$, f be an element of $\mathbb{R}^*$, and i be a natural number. Then $(\mathrm{repeat}(F \cdot G))(i+1)(f) = F(G((\mathrm{repeat}(F \cdot G))(i)(f)))$.

Let g be an element of $(\mathbb{R}^*)^{\mathbb{R}^*}$ and let f be an element of $\mathbb{R}^*$. Then $g(f)$ is an element of $\mathbb{R}^*$.

Let f be an element of $\mathbb{R}^*$ and let n be a natural number. The functor $\mathrm{OuterVx}(f,n)$ yielding a subset of $\mathbb{N}$ is defined as follows:

(Def. 3) $\mathrm{OuterVx}(f,n) = \{i : i \in \mathrm{dom} f \wedge 1 \leq i \wedge i \leq n \wedge f(i) \neq -1 \wedge f(n+i) \neq -1\}$.

Let f be an element of $(\mathbb{R}^*)^{\mathbb{R}^*}$, let g be an element of $\mathbb{R}^*$, and let n be a natural number. Let us assume that there exists i such that $\mathrm{OuterVx}((\mathrm{repeat} f)(i)(g),n) = \emptyset$. The functor $\mathrm{LifeSpan}(f,g,n)$ yields a natural number and is defined by:

(Def. 4) $\mathrm{OuterVx}((\mathrm{repeat} f)(\mathrm{LifeSpan}(f,g,n))(g),n) = \emptyset$ and for every natural number k such that $\mathrm{OuterVx}((\mathrm{repeat} f)(k)(g),n) = \emptyset$ holds $\mathrm{LifeSpan}(f,g,n) \leq k$.

Let f be an element of $(\mathbb{R}^*)^{\mathbb{R}^*}$ and let n be a natural number. The functor $\mathrm{WhileDo}(f,n)$ yielding an element of $(\mathbb{R}^*)^{\mathbb{R}^*}$ is defined by:

(Def. 5) $\mathrm{dom}\,\mathrm{WhileDo}(f,n) = \mathbb{R}^*$ and for every element h of $\mathbb{R}^*$ holds $(\mathrm{WhileDo}(f,n))(h) = (\mathrm{repeat} f)(\mathrm{LifeSpan}(f,h,n))(h)$.

6. Defining a Weight Function for an Oriented Graph

Let G be an oriented graph and let v_1, v_2 be vertices of G. Let us assume that there exists a set e such that $e \in$ the edges of G and e orientedly joins v_1, v_2. The functor $\mathrm{Edge}(v_1,v_2)$ is defined as follows:

(Def. 6) There exists a set e such that $\mathrm{Edge}(v_1,v_2) = e$ and $e \in$ the edges of G and e orientedly joins v_1, v_2.

Let G be an oriented graph, let v_1, v_2 be vertices of G, and let W be a function. The functor $\mathrm{Weight}(v_1,v_2,W)$ is defined as follows:

(Def. 7) $\mathrm{Weight}(v_1,v_2,W) = \begin{cases} W(\mathrm{Edge}(v_1,v_2)), & \text{if thereexists a set } e \text{ such that } e \in \text{ the edges of } G \text{ and } e \text{ orientedly joins } v_1, v_2, \\ -1, & \text{otherwise.} \end{cases}$

Let G be an oriented graph, let v_1, v_2 be vertices of G, and let W be a function from the edges of G into $\mathbb{R}_{\geq 0}$. Then $\mathrm{Weight}(v_1,v_2,W)$ is a real number.

In the sequel G denotes an oriented graph, v_1, v_2 denote vertices of G, and W denotes a function from the edges of G into $\mathbb{R}_{\geq 0}$.

We now state three propositions:

(24) $\mathrm{Weight}(v_1,v_2,W) \geq 0$ iff there exists a set e such that $e \in$ the edges of G and e orientedly joins v_1, v_2.

(25) $\mathrm{Weight}(v_1,v_2,W) = -1$ iff it is not true that there exists a set e such that $e \in$ the edges of G and e orientedly joins v_1, v_2.

(26) If $e \in$ the edges of G and e orientedly joins v_1, v_2, then $\mathrm{Weight}(v_1,v_2,W) = W(e)$.

7. Basic Operations for Dijkstra's Shortest Path Algorithm

Let f be an element of $\mathbb{R}^*$ and let n be a natural number. The functor $\mathrm{UnusedVx}(f,n)$ yields a subset of $\mathbb{N}$ and is defined by:

(Def. 8) $\mathrm{UnusedVx}(f,n) = \{i : i \in \mathrm{dom}\, f \wedge 1 \leq i \wedge i \leq n \wedge f(i) \neq -1\}$.

Let f be an element of $\mathbb{R}^*$ and let n be a natural number. The functor $\mathrm{UsedVx}(f,n)$ yielding a subset of $\mathbb{N}$ is defined by:

(Def. 9) $\mathrm{UsedVx}(f,n) = \{i : i \in \mathrm{dom}\, f \wedge 1 \leq i \wedge i \leq n \wedge f(i) = -1\}$.

Next we state the proposition

(27) $\mathrm{UnusedVx}(f,n) \subseteq \mathrm{Seg}\, n$.

Let f be an element of $\mathbb{R}^*$ and let n be a natural number. Observe that $\mathrm{UnusedVx}(f,n)$ is finite. The following propositions are true:

(28) $\mathrm{OuterVx}(f,n) \subseteq \mathrm{UnusedVx}(f,n)$.

(29) $\mathrm{OuterVx}(f,n) \subseteq \mathrm{Seg}\, n$.

Let f be an element of $\mathbb{R}^*$ and let n be a natural number. Note that $\mathrm{OuterVx}(f,n)$ is finite.

Let X be a finite subset of $\mathbb{N}$, let f be an element of $\mathbb{R}^*$, and let us consider n. The functor $\mathrm{Argmin}(X,f,n)$ yields a natural number and is defined by the conditions (Def. 10).

(Def. 10)(i) If $X \neq \emptyset$, then there exists i such that $i = \mathrm{Argmin}(X,f,n)$ and $i \in X$ and for every k such that $k \in X$ holds $f_{2 \cdot n+i} \leq f_{2 \cdot n+k}$ and for every k such that $k \in X$ and $f_{2 \cdot n+i} = f_{2 \cdot n+k}$ holds $i \leq k$, and

(ii) if $X = \emptyset$, then $\mathrm{Argmin}(X,f,n) = 0$.

The following propositions are true:

(30) If $\mathrm{OuterVx}(f,n) \neq \emptyset$ and $j = \mathrm{Argmin}(\mathrm{OuterVx}(f,n),f,n)$, then $j \in \mathrm{dom}\, f$ and $1 \leq j$ and $j \leq n$ and $f(j) \neq -1$ and $f(n+j) \neq -1$.

(31) $\mathrm{Argmin}(\mathrm{OuterVx}(f,n),f,n) \leq n$.

Let n be a natural number. The functor $\mathrm{findmin}\, n$ yielding an element of $(\mathbb{R}^*)^{\mathbb{R}^*}$ is defined by:

(Def. 11) $\mathrm{dom}\, \mathrm{findmin}\, n = \mathbb{R}^*$ and for every element f of $\mathbb{R}^*$ holds $(\mathrm{findmin}\, n)(f) = (f, n \cdot n + 3 \cdot n + 1) := (\mathrm{Argmin}(\mathrm{OuterVx}(f,n),f,n), -1)$.

Next we state four propositions:

(32) If $i \in \mathrm{dom}\, f$ and $i > n$ and $i \neq n \cdot n + 3 \cdot n + 1$, then $(\mathrm{findmin}\, n)(f)(i) = f(i)$.

(33) If $i \in \mathrm{dom}\, f$ and $f(i) = -1$ and $i \neq n \cdot n + 3 \cdot n + 1$, then $(\mathrm{findmin}\, n)(f)(i) = -1$.

(34) $\mathrm{dom}(\mathrm{findmin}\, n)(f) = \mathrm{dom}\, f$.

(35) If $\mathrm{OuterVx}(f,n) \neq \emptyset$, then there exists j such that $j \in \mathrm{OuterVx}(f,n)$ and $1 \leq j$ and $j \leq n$ and $(\mathrm{findmin}\, n)(f)(j) = -1$.

Let f be an element of $\mathbb{R}^*$ and let n, k be natural numbers. The functor newpathcost(f,n,k) yields a real number and is defined by:

(Def. 12) newpathcost$(f,n,k) = f_{2\cdot n+f_{n\cdot n+3\cdot n+1}} + f_{2\cdot n+n\cdot f_{n\cdot n+3\cdot n+1}+k}$.

Let n, k be natural numbers and let f be an element of $\mathbb{R}^*$. We say that f has better path at n, k if and only if:

(Def. 13) $f(n+k) = -1$ or $f_{2\cdot n+k} >$ newpathcost(f,n,k) but $f_{2\cdot n+n\cdot f_{n\cdot n+3\cdot n+1}+k} \geq 0$ but $f(k) \neq -1$.

Let f be an element of $\mathbb{R}^*$ and let n be a natural number. The functor Relax(f,n) yields an element of $\mathbb{R}^*$ and is defined by the conditions (Def. 14).

(Def. 14)(i) dom Relax$(f,n) =$ dom f, and

(ii) for every natural number k such that $k \in$ dom f holds if $n < k$ and $k \leq 2\cdot n$, then if f has better path at n, $k -' n$, then $(\mathrm{Relax}(f,n))(k) = f_{n\cdot n+3\cdot n+1}$ and if f does not have better path at n, $k -' n$, then $(\mathrm{Relax}(f,n))(k) = f(k)$ and if $2\cdot n < k$ and $k \leq 3\cdot n$, then if f has better path at n, $k -' 2\cdot n$, then $(\mathrm{Relax}(f,n))(k) = \mathrm{newpathcost}(f,n,k -' 2\cdot n)$ and if f does not have better path at n, $k -' 2\cdot n$, then $(\mathrm{Relax}(f,n))(k) = f(k)$ and if $k \leq n$ or $k > 3\cdot n$, then $(\mathrm{Relax}(f,n))(k) = f(k)$.

Let n be a natural number. The functor Relax n yielding an element of $(\mathbb{R}^*)^{\mathbb{R}^*}$ is defined by:

(Def. 15) dom Relax $n = \mathbb{R}^*$ and for every element f of $\mathbb{R}^*$ holds $(\mathrm{Relax}\,n)(f) = \mathrm{Relax}(f,n)$.

One can prove the following propositions:

(36) $\mathrm{dom}(\mathrm{Relax}\,n)(f) = \mathrm{dom}\,f$.

(37) If $i \leq n$ or $i > 3\cdot n$ and if $i \in \mathrm{dom}\,f$, then $(\mathrm{Relax}\,n)(f)(i) = f(i)$.

(38) $\mathrm{dom}(\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(i)(f) = \mathrm{dom}(\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(i+1)(f)$.

(39) If $\mathrm{OuterVx}((\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(i)(f),n) \neq \emptyset$, then $\mathrm{UnusedVx}((\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(i+1)(f),n) \subset \mathrm{UnusedVx}((\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(i)(f),n)$.

(40) If $g = (\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(i)(f)$ and $h = (\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(i+1)(f)$ and $k = \mathrm{Argmin}(\mathrm{OuterVx}(g,n),g,n)$ and $\mathrm{OuterVx}(g,n) \neq \emptyset$, then $\mathrm{UsedVx}(h,n) = \mathrm{UsedVx}(g,n) \cup \{k\}$ and $k \notin \mathrm{UsedVx}(g,n)$.

(41) There exists i such that $i \leq n$ and $\mathrm{OuterVx}((\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(i)(f),n) = \emptyset$.

(42) $\mathrm{dom}\,f = \mathrm{dom}(\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(i)(f)$.

Let f, g be elements of $\mathbb{R}^*$ and let us consider m, n. We say that f, g are equal at m, n if and only if:

(Def. 16) $\mathrm{dom}\,f = \mathrm{dom}\,g$ and for every k such that $k \in \mathrm{dom}\,f$ and $m \leq k$ and $k \leq n$ holds $f(k) = g(k)$.

The following propositions are true:

(43) f, f are equal at m, n.

(44) If f, g are equal at m, n and g, h are equal at m, n, then f, h are equal at m, n.

(45) $(\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(i)(f)$, $(\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(i+1)(f)$ are equal at $3\cdot n+1$, $n\cdot n+3\cdot n$.

(46) Let F be an element of $(\mathbb{R}^*)^{\mathbb{R}^*}$, f be an element of $\mathbb{R}^*$, and n, i be natural numbers. If $i < \mathrm{LifeSpan}(F,f,n)$, then $\mathrm{OuterVx}((\mathrm{repeat}\,F)(i)(f),n) \neq \emptyset$.

(47) f, $(\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(i)(f)$ are equal at $3\cdot n+1$, $n\cdot n+3\cdot n$.

(48) Suppose that

(i) $1 \leq n$,

(ii) $1 \in \operatorname{dom} f$,

(iii) $f(n+1) \neq -1$,

(iv) for every i such that $1 \leq i$ and $i \leq n$ holds $f(i) = 1$, and

(v) for every i such that $2 \leq i$ and $i \leq n$ holds $f(n+i) = -1$.

Then $1 = \mathrm{Argmin}(\mathrm{OuterVx}(f,n), f, n)$ and $\mathrm{UsedVx}(f,n) = \emptyset$ and $\{1\} = \mathrm{UsedVx}((\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(1)(f), n)$.

(49) If $g = (\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(1)(f)$ and $h = (\mathrm{repeat}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n))(i)(f)$ and $1 \leq i$ and $i \leq \mathrm{LifeSpan}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n, f, n)$ and $m \in \mathrm{UsedVx}(g,n)$, then $m \in \mathrm{UsedVx}(h,n)$.

Let p be a finite sequence of elements of $\mathbb{N}$, let f be an element of $\mathbb{R}^*$, and let i, n be natural numbers. We say that p is vertex sequence at f, i, n if and only if:

(Def. 17) $p(\operatorname{len} p) = i$ and for every k such that $1 \leq k$ and $k < \operatorname{len} p$ holds $p(\operatorname{len} p - k) = f(n + p_{(\operatorname{len} p - k)+1})$.

Let p be a finite sequence of elements of $\mathbb{N}$, let f be an element of $\mathbb{R}^*$, and let i, n be natural numbers. We say that p is simple vertex sequence at f, i, n if and only if:

(Def. 18) $p(1) = 1$ and $\operatorname{len} p > 1$ and p is vertex sequence at f, i, n and one-to-one.

One can prove the following proposition

(50) Let p, q be finite sequences of elements of $\mathbb{N}$, f be an element of $\mathbb{R}^*$, and i, n be natural numbers. Suppose p is simple vertex sequence at f, i, n and q is simple vertex sequence at f, i, n. Then $p = q$.

Let G be a graph, let p be a finite sequence of elements of the edges of G, and let v_5 be a finite sequence. We say that p is oriented edge sequence at v_5 if and only if:

(Def. 19) $\operatorname{len} v_5 = \operatorname{len} p + 1$ and for every n such that $1 \leq n$ and $n \leq \operatorname{len} p$ holds (the source of $G)(p(n)) = v_5(n)$ and (the target of $G)(p(n)) = v_5(n+1)$.

Next we state two propositions:

(51) Let G be an oriented graph, v_5 be a finite sequence, and p, q be oriented chains of G. Suppose p is oriented edge sequence at v_5 and q is oriented edge sequence at v_5. Then $p = q$.

(52) Let G be a graph, v_6, v_7 be finite sequences, and p be an oriented chain of G. Suppose p is oriented edge sequence at v_6 and oriented edge sequence at v_7 and $\operatorname{len} p \geq 1$. Then $v_6 = v_7$.

8. Data Structure for Dijkstra's Shortest Path Algorithm

Let f be an element of $\mathbb{R}^*$, let G be an oriented graph, let n be a natural number, and let W be a function from the edges of G into $\mathbb{R}_{\geq 0}$. We say that f is input of Dijkstra algorithm G to n in W if and only if the conditions (Def. 20) are satisfied.

(Def. 20)(i) $\operatorname{len} f = n \cdot n + 3 \cdot n + 1$,

(ii) $\mathrm{Seg}\, n =$ the vertices of G,

(iii) for every i such that $1 \leq i$ and $i \leq n$ holds $f(i) = 1$ and $f(2 \cdot n + i) = 0$,

(iv) $f(n+1) = 0$,

(v) for every i such that $2 \leq i$ and $i \leq n$ holds $f(n+i) = -1$, and

(vi) for all vertices i, j of G and for all k, m such that $k = i$ and $m = j$ holds $f(2 \cdot n + n \cdot k + m) = \mathrm{Weight}(i, j, W)$.

9. The Definition of Dijkstra's Shortest Path Algorithm

Let n be a natural number. The functor DijkstraAlgorithm$n$ yielding an element of $(\mathbb{R}^*)^{\mathbb{R}^*}$ is defined by:

(Def. 21) $\mathrm{DijkstraAlgorithm}\,n = \mathrm{WhileDo}(\mathrm{Relax}\,n \cdot \mathrm{findmin}\,n, n)$.

10. Justifying the Correctness of Dijkstra's Shortest Path Algorithm

For simplicity, we adopt the following rules: p is a finite sequence of elements of $\mathbb{N}$, G is a finite oriented graph, P, Q are oriented chains of G, W is a function from the edges of G into $\mathbb{R}_{\geq 0}$, and v_1, v_2 are vertices of G.

Next we state the proposition

(53) Suppose f is input of Dijkstra algorithm G to n in W and $v_1 = 1$ and $1 \neq v_2$ and $v_2 = i$ and $n \geq 1$ and $g = (\mathrm{DijkstraAlgorithm}\,n)(f)$. Then

(i) the vertices of $G = \mathrm{UsedVx}(g,n) \cup \mathrm{UnusedVx}(g,n)$,

(ii) if $v_2 \in \mathrm{UsedVx}(g,n)$, then there exist p, P such that p is simple vertex sequence at g, i, n and P is oriented edge sequence at p and shortest path from v_1 to v_2 in W and $\mathrm{cost}(P,W) = g(2 \cdot n + i)$, and

(iii) if $v_2 \in \mathrm{UnusedVx}(g,n)$, then there exists no Q which is oriented path from v_1 to v_2.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/card_1.html`.

[2] Grzegorz Bancerek. The fundamental properties of natural numbers. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/nat_1.html`.

[3] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/finseq_1.html`.

[4] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Journal of Formalized Mathematics*, 8, 1996. `http://mizar.org/JFM/Vol8/funct_7.html`.

[5] Czesław Byliński. Functions and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/funct_1.html`.

[6] Czesław Byliński. Functions from a set to a set. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/funct_2.html`.

[7] Czesław Byliński. Partial functions. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/partfun1.html`.

[8] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/finseq_2.html`.

[9] Czesław Byliński. The sum and product of finite sequences of real numbers. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/rvsum_1.html`.

[10] Agata Darmochwał. Finite sets. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/finset_1.html`.

[11] E. W. Dijkstra. A note on two problems in connection with graphs. *Numer. Math.*, 1:269–271, 1959.

[12] Krzysztof Hryniewiecki. Basic properties of real numbers. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/real_1.html`.

[13] Krzysztof Hryniewiecki. Graphs. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/graph_1.html`.

[14] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-Board — part I. *Journal of Formalized Mathematics*, 4, 1992. `http://mizar.org/JFM/Vol4/goboard1.html`.

[15] Yatsuka Nakamura and Piotr Rudnicki. Oriented chains. *Journal of Formalized Mathematics*, 10, 1998. `http://mizar.org/JFM/Vol10/graph_4.html`.

[16] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Journal of Formalized Mathematics*, 5, 1993. `http://mizar.org/JFM/Vol5/binarith.html`.

[17] Andrzej Trybulec. Domains and their Cartesian products. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/domain_1.html`.

[18] Andrzej Trybulec. Tarski Grothendieck set theory. *Journal of Formalized Mathematics*, Axiomatics, 1989. `http://mizar.org/JFM/Axiomatics/tarski.html`.

[19] Andrzej Trybulec. Subsets of real numbers. *Journal of Formalized Mathematics*, Addenda, 2003. `http://mizar.org/JFM/Addenda/numbers.html`.

[20] Michał J. Trybulec. Integers. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/int_1.html`.

[21] Wojciech A. Trybulec. Pigeon hole principle. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/finseq_4.html`.

[22] Zinaida Trybulec. Properties of subsets. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/subset_1.html`.

[23] Edmund Woronowicz. Relations and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/relat_1.html`.

Received March 17, 2003

Published January 2, 2004