

An Axiomatic Characterization of the Single-Source Shortest Path Problem

By Christine Rizkallah

May 26, 2024

Abstract

This theory is split into two sections. In the first section, we give a formal proof that a well-known axiomatic characterization of the single-source shortest path problem is correct. Namely, we prove that in a directed graph $G = (V, E)$ with a non-negative cost function on the edges the single-source shortest path function $\mu : V \rightarrow \mathbb{R} \cup \{\infty\}$ is the only function that satisfies a set of four axioms. The first axiom states that the distance from the source vertex s to itself should be equal to zero. The second states that the distance from s to a vertex $v \in V$ should be infinity if and only if there is no path from s to v . The third axiom is called triangle inequality and states that if there is a path from s to v , and an edge $(u, v) \in E$, the distance from s to v is less than or equal to the distance from s to u plus the cost of (u, v) . The last axiom is called justification, it states that for every vertex v other than s , if there is a path p from s to v in G , then there is a predecessor edge (u, v) on p such that the distance from s to v is equal to the distance from s to u plus the cost of (u, v) .

In the second section, we give a formal proof of the correctness of an axiomatic characterization of the single-source shortest path problem for directed graphs with general cost functions $c : E \rightarrow \mathbb{R}$. The axioms here are more involved because we have to account for potential negative cycles in the graph. The axioms are summarized in the three isabelle locales.

Contents

1	Shortest Path (with non-negative edge costs)	2
2	Shortest Path (with general edge costs)	4
	<i>theory ShortestPath</i>	
	imports	
	<i>Graph-Theory.Graph-Theory</i>	
	begin	

1 Shortest Path (with non-negative edge costs)

The following theory is used in the verification of a certifying algorithm's checker for shortest path. For more information see [1].

locale *basic-sp* =
fin-digraph +
fixes *dist* :: 'a \Rightarrow *ereal*
fixes *c* :: 'b \Rightarrow *real*
fixes *s* :: 'a
assumes *general-source-val*: $dist\ s \leq 0$
assumes *trian*:
 $\bigwedge e. e \in arcs\ G \implies$
 $dist\ (head\ G\ e) \leq dist\ (tail\ G\ e) + c\ e$

locale *basic-just-sp* =
basic-sp +
fixes *num* :: 'a \Rightarrow *enat*
assumes *just*:
 $\bigwedge v. \llbracket v \in verts\ G; v \neq s; num\ v \neq \infty \rrbracket \implies$
 $\exists e \in arcs\ G. v = head\ G\ e \wedge$
 $dist\ v = dist\ (tail\ G\ e) + c\ e \wedge$
 $num\ v = num\ (tail\ G\ e) + (enat\ 1)$

locale *shortest-path-pos-cost* =
basic-just-sp +
assumes *s-in-G*: $s \in verts\ G$
assumes *tail-val*: $dist\ s = 0$
assumes *no-path*: $\bigwedge v. v \in verts\ G \implies dist\ v = \infty \longleftrightarrow num\ v = \infty$
assumes *pos-cost*: $\bigwedge e. e \in arcs\ G \implies 0 \leq c\ e$

locale *basic-just-sp-pred* =
basic-sp +
fixes *num* :: 'a \Rightarrow *enat*
fixes *pred* :: 'a \Rightarrow 'b *option*
assumes *just*:
 $\bigwedge v. \llbracket v \in verts\ G; v \neq s; num\ v \neq \infty \rrbracket \implies$
 $\exists e \in arcs\ G.$
 $e = the\ (pred\ v) \wedge$
 $v = head\ G\ e \wedge$
 $dist\ v = dist\ (tail\ G\ e) + c\ e \wedge$
 $num\ v = num\ (tail\ G\ e) + (enat\ 1)$

sublocale *basic-just-sp-pred* \subseteq *basic-just-sp*
<proof>

locale *shortest-path-pos-cost-pred* =
basic-just-sp-pred +
assumes *s-in-G*: $s \in verts\ G$
assumes *tail-val*: $dist\ s = 0$

assumes *no-path*: $\bigwedge v. v \in \text{verts } G \implies \text{dist } v = \infty \longleftrightarrow \text{num } v = \infty$
assumes *pos-cost*: $\bigwedge e. e \in \text{arcs } G \implies 0 \leq c e$

sublocale *shortest-path-pos-cost-pred* \subseteq *shortest-path-pos-cost*
 <proof>

lemma *tail-value-helper*:

assumes *hd p = last p*
assumes *distinct p*
assumes $p \neq []$
shows $p = [\text{hd } p]$
 <proof>

lemma (in *basic-sp*) *dist-le-cost*:

fixes $v :: 'a$
fixes $p :: 'b \text{ list}$
assumes *awalk s p v*
shows $\text{dist } v \leq \text{awalk-cost } c p$
 <proof>

lemma (in *fin-digraph*) *witness-path*:

assumes $\mu c s v = \text{ereal } r$
shows $\exists p. \text{apath } s p v \wedge \mu c s v = \text{awalk-cost } c p$
 <proof>

lemma (in *basic-sp*) *dist-le- μ* :

fixes $v :: 'a$
assumes $v \in \text{verts } G$
shows $\text{dist } v \leq \mu c s v$
 <proof>

lemma (in *basic-just-sp*) *dist-ge- μ* :

fixes $v :: 'a$
assumes $v \in \text{verts } G$
assumes $\text{num } v \neq \infty$
assumes $\text{dist } v \neq -\infty$
assumes $\mu c s s = \text{ereal } 0$
assumes $\text{dist } s = 0$
assumes $\bigwedge u. u \in \text{verts } G \implies u \neq s \implies$
 $\text{num } u \neq \infty \implies \text{num } u \neq \text{enat } 0$
shows $\text{dist } v \geq \mu c s v$
 <proof>

lemma (in *shortest-path-pos-cost*) *tail-value-check*:

fixes $u :: 'a$
assumes $s \in \text{verts } G$
shows $\mu c s s = \text{ereal } 0$
 <proof>

lemma (in *shortest-path-pos-cost*) *num-not0*:

fixes $v :: 'a$
assumes $v \in \text{verts } G$
assumes $v \neq s$
assumes $\text{num } v \neq \infty$
shows $\text{num } v \neq \text{enat } 0$

<proof>

lemma (in *shortest-path-pos-cost*) *dist-ne-ninf*:

fixes $v :: 'a$
assumes $v \in \text{verts } G$
shows $\text{dist } v \neq -\infty$

<proof>

theorem (in *shortest-path-pos-cost*) *correct-shortest-path*:

fixes $v :: 'a$
assumes $v \in \text{verts } G$
shows $\text{dist } v = \mu \ c \ s \ v$

<proof>

corollary (in *shortest-path-pos-cost-pred*) *correct-shortest-path-pred*:

fixes $v :: 'a$
assumes $v \in \text{verts } G$
shows $\text{dist } v = \mu \ c \ s \ v$

<proof>

end

theory *ShortestPathNeg*

imports *ShortestPath*

begin

2 Shortest Path (with general edge costs)

locale *shortest-paths-locale-step1* =

fixes $G :: ('a, 'b) \text{ pre-digraph (structure)}$
fixes $s :: 'a$
fixes $c :: 'b \Rightarrow \text{real}$
fixes $\text{num} :: 'a \Rightarrow \text{nat}$
fixes $\text{parent-edge} :: 'a \Rightarrow 'b \text{ option}$
fixes $\text{dist} :: 'a \Rightarrow \text{ereal}$
assumes $\text{graphG}: \text{fin-digraph } G$
assumes $s\text{-assms}$:
 $s \in \text{verts } G$
 $\text{dist } s \neq \infty$
 $\text{parent-edge } s = \text{None}$
 $\text{num } s = 0$

assumes *parent-num-assms*:

$\bigwedge v. \llbracket v \in \text{verts } G; v \neq s; \text{dist } v \neq \infty \rrbracket \implies$
 $(\exists e \in \text{arcs } G. \text{parent-edge } v = \text{Some } e \wedge$
 $\text{head } G e = v \wedge \text{dist } (\text{tail } G e) \neq \infty \wedge$
 $\text{num } v = \text{num } (\text{tail } G e) + 1)$

assumes *noPedge*: $\bigwedge e. e \in \text{arcs } G \implies$

$\text{dist } (\text{tail } G e) \neq \infty \implies \text{dist } (\text{head } G e) \neq \infty$

sublocale *shortest-paths-locale-step1* \subseteq *fin-digraph* *G*

<proof>

definition (**in** *shortest-paths-locale-step1*) *enum* :: 'a \Rightarrow *enat* **where**

enum *v* = (if (dist *v* = $\infty \vee \text{dist } v = -\infty$) then ∞ else *num* *v*)

locale *shortest-paths-locale-step2* =

shortest-paths-locale-step1 +
basic-just-sp *G* *dist* *c* *s* *enum* +

assumes *source-val*: $(\exists v \in \text{verts } G. \text{enum } v \neq \infty) \implies \text{dist } s = 0$

assumes *no-edge-Vm-Vf*:

$\bigwedge e. e \in \text{arcs } G \implies \text{dist } (\text{tail } G e) = -\infty \implies \forall r. \text{dist } (\text{head } G e) \neq \text{ereal } r$

function (**in** *shortest-paths-locale-step1*) *pwalk* :: 'a \Rightarrow 'b *list*

where

pwalk *v* =

(if ($v = s \vee \text{dist } v = \infty \vee v \notin \text{verts } G$)

then \llbracket

else *pwalk* (*tail* *G* (*the* (*parent-edge* *v*))) @ [*the* (*parent-edge* *v*)]

)

<proof>

termination (**in** *shortest-paths-locale-step1*)

<proof>

lemma (**in** *shortest-paths-locale-step1*) *pwalk-simps*:

$v = s \vee \text{dist } v = \infty \vee v \notin \text{verts } G \implies \text{pwalk } v = \llbracket$

$v \neq s \implies \text{dist } v \neq \infty \implies v \in \text{verts } G \implies$

pwalk *v* = *pwalk* (*tail* *G* (*the* (*parent-edge* *v*))) @ [*the* (*parent-edge* *v*)]

<proof>

definition (**in** *shortest-paths-locale-step1*) *pwalk-verts* :: 'a \Rightarrow 'a *set* **where**

pwalk-verts *v* = {*u*. *u* \in *set* (*awalk-verts* *s* (*pwalk* *v*))}

locale *shortest-paths-locale-step3* =

shortest-paths-locale-step2 +

fixes *C* :: ('a \times ('b *awalk*)) *set*

assumes *C-se*:

$C \subseteq \{(u, p). \text{dist } u \neq \infty \wedge \text{awalk } u p u \wedge \text{awalk-cost } c p < 0\}$

assumes *int-neg-cyc*:

$\bigwedge v. v \in \text{verts } G \implies \text{dist } v = -\infty \implies$

$(fst \ 'C) \cap pwalk\text{-}verts \ v \neq \{\}$

locale *shortest-paths-locale-step2-pred* =
shortest-paths-locale-step1 +
fixes *pred* :: 'a \Rightarrow 'b option
assumes *bj*: basic-just-sp-pred *G* *dist* *c* *s* *enum* *pred*
assumes *source-val*: $(\exists v \in verts \ G. \ enum \ v \neq \infty) \implies dist \ s = 0$
assumes *no-edge-Vm-Vf*:
 $\bigwedge e. \ e \in arcs \ G \implies dist \ (tail \ G \ e) = -\infty \implies \forall r. \ dist \ (head \ G \ e) \neq ereal \ r$

lemma (*in shortest-paths-locale-step1*) *num-s-is-min*:

assumes $v \in verts \ G$
assumes $v \neq s$
assumes $dist \ v \neq \infty$
shows $num \ v > 0$
 $\langle proof \rangle$

lemma (*in shortest-paths-locale-step1*) *path-from-root-Vr-ex*:

fixes $v :: 'a$
assumes $v \in verts \ G$
assumes $v \neq s$
assumes $dist \ v \neq \infty$
shows $\exists e. \ s \rightarrow^* tail \ G \ e \wedge$
 $e \in arcs \ G \wedge head \ G \ e = v \wedge dist \ (tail \ G \ e) \neq \infty \wedge$
 $parent\text{-}edge \ v = Some \ e \wedge num \ v = num \ (tail \ G \ e) + 1$
 $\langle proof \rangle$

lemma (*in shortest-paths-locale-step1*) *path-from-root-Vr*:

fixes $v :: 'a$
assumes $v \in verts \ G$
assumes $dist \ v \neq \infty$
shows $s \rightarrow^* v$
 $\langle proof \rangle$

lemma (*in shortest-paths-locale-step1*) *μ -V-less-inf*:

fixes $v :: 'a$
assumes $v \in verts \ G$
assumes $dist \ v \neq \infty$
shows $\mu \ c \ s \ v \neq \infty$
 $\langle proof \rangle$

lemma (*in shortest-paths-locale-step2*) *enum-not0*:

assumes $v \in verts \ G$
assumes $v \neq s$
assumes $enum \ v \neq \infty$
shows $enum \ v \neq enat \ 0$
 $\langle proof \rangle$

lemma (in *shortest-paths-locale-step2*) *dist-Vf- μ* :

fixes $v :: 'a$

assumes $vG: v \in \text{verts } G$

assumes $\exists r. \text{dist } v = \text{ereal } r$

shows $\text{dist } v = \mu \text{ c s } v$

<proof>

lemma (in *shortest-paths-locale-step1*) *pwalk-awalk*:

fixes $v :: 'a$

assumes $v \in \text{verts } G$

assumes $\text{dist } v \neq \infty$

shows $\text{awalk } s (\text{pwalk } v) v$

<proof>

lemma (in *shortest-paths-locale-step3*) *μ -ninf*:

fixes $v :: 'a$

assumes $v \in \text{verts } G$

assumes $\text{dist } v = -\infty$

shows $\mu \text{ c s } v = -\infty$

<proof>

lemma (in *shortest-paths-locale-step3*) *correct-shortest-path*:

fixes $v :: 'a$

assumes $v \in \text{verts } G$

shows $\text{dist } v = \mu \text{ c s } v$

<proof>

end

References

- [1] E. Alkassar, S. Böhme, K. Mehlhorn, and C. Rizkallah. A framework for the verification of certifying computations. *Journal of Automated Reasoning*, 2013. To Appear.