An Axiomatic Characterization of the Single-Source Shortest Path Problem

By Christine Rizkallah

March 19, 2025

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 $\mu : V \to \mathbb{R} \cup \{\infty\}$ 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 $c : E \to \mathbb{R}$. 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

theory *ShortestPath*
imports
  Graph-Theory.Graph-Theory
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*:
    $\bigwedge e.\ e \in arcs\ G \Longrightarrow$
      $dist\ (head\ G\ e) \leq dist\ (tail\ G\ e) + c\ e$

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

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*: $\bigwedge v.\ v \in verts\ G \Longrightarrow dist\ v = \infty \longleftrightarrow num\ v = \infty$
  assumes *pos-cost*: $\bigwedge e.\ e \in arcs\ G \Longrightarrow 0 \leq c\ e$

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

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*: $\bigwedge v.\ v \in \mathit{verts}\ G \Longrightarrow \mathit{dist}\ v = \infty \longleftrightarrow \mathit{num}\ v = \infty$
assumes *pos-cost*: $\bigwedge e.\ e \in \mathit{arcs}\ G \Longrightarrow 0 \leq c\ e$

sublocale *shortest-path-pos-cost-pred* $\subseteq$ *shortest-path-pos-cost*
⟨*proof*⟩

lemma *tail-value-helper*:
assumes $\mathit{hd}\ p = \mathit{last}\ p$
assumes $\mathit{distinct}\ p$
assumes $p \neq []$
shows $p = [\mathit{hd}\ p]$
⟨*proof*⟩

lemma (**in** *basic-sp*) *dist-le-cost*:
fixes $v :: 'a$
fixes $p :: 'b\ \mathit{list}$
assumes $\mathit{awalk}\ s\ p\ v$
shows $\mathit{dist}\ v \leq \mathit{awalk\text{-}cost}\ c\ p$
⟨*proof*⟩

lemma (**in** *fin-digraph*) *witness-path*:
assumes $\mu\ c\ s\ v = \mathit{ereal}\ r$
shows $\exists\ p.\ \mathit{apath}\ s\ p\ v \wedge \mu\ c\ s\ v = \mathit{awalk\text{-}cost}\ c\ p$
⟨*proof*⟩

lemma (**in** *basic-sp*) *dist-le-μ*:
fixes $v :: 'a$
assumes $v \in \mathit{verts}\ G$
shows $\mathit{dist}\ v \leq \mu\ c\ s\ v$
⟨*proof*⟩

lemma (**in** *basic-just-sp*) *dist-ge-μ*:
fixes $v :: 'a$
assumes $v \in \mathit{verts}\ G$
assumes $\mathit{num}\ v \neq \infty$
assumes $\mathit{dist}\ v \neq -\infty$
assumes $\mu\ c\ s\ s = \mathit{ereal}\ 0$
assumes $\mathit{dist}\ s = 0$
assumes $\bigwedge u.\ u{\in}\mathit{verts}\ G \Longrightarrow u{\neq}s \Longrightarrow$
$\mathit{num}\ u \neq \infty \Longrightarrow \mathit{num}\ u \neq \mathit{enat}\ 0$
shows $\mathit{dist}\ v \geq \mu\ c\ s\ v$
⟨*proof*⟩

lemma (**in** *shortest-path-pos-cost*) *tail-value-check*:
fixes $u :: 'a$
assumes $s \in \mathit{verts}\ G$
shows $\mu\ c\ s\ s = \mathit{ereal}\ 0$
⟨*proof*⟩

lemma (**in** *shortest-path-pos-cost*) *num-not0*:
  fixes $v :: 'a$
  assumes $v \in verts\ G$
  assumes $v \neq s$
  assumes $num\ v \neq \infty$
  shows $num\ v \neq enat\ 0$
⟨*proof*⟩

lemma (**in** *shortest-path-pos-cost*) *dist-ne-ninf*:
  fixes $v :: 'a$
  assumes $v \in verts\ G$
  shows $dist\ v \neq -\infty$
⟨*proof*⟩

theorem (**in** *shortest-path-pos-cost*) *correct-shortest-path*:
  fixes $v :: 'a$
  assumes $v \in verts\ G$
  shows $dist\ v = \mu\ c\ s\ v$
  ⟨*proof*⟩

corollary (**in** *shortest-path-pos-cost-pred*) *correct-shortest-path-pred*:
  fixes $v :: 'a$
  assumes $v \in verts\ G$
  shows $dist\ v = \mu\ c\ s\ v$
  ⟨*proof*⟩

end
theory *ShortestPathNeg*

imports *ShortestPath*

begin

2 Shortest Path (with general edge costs)

locale *shortest-paths-locale-step1* =
  fixes $G :: ('a, 'b)$ *pre-digraph* (**structure**)
  fixes $s :: 'a$
  fixes $c :: 'b \Rightarrow real$
  fixes $num :: 'a \Rightarrow nat$
  fixes *parent-edge* $:: 'a \Rightarrow 'b\ option$
  fixes $dist :: 'a \Rightarrow ereal$
  assumes *graphG*: *fin-digraph* G
  assumes *s-assms*:
    $s \in verts\ G$
    $dist\ s \neq \infty$
    parent-edge $s = None$
    $num\ s = 0$

assumes *parent-num-assms*:
$\bigwedge v.\ [\![v \in verts\ G;\ v \neq s;\ dist\ v \neq \infty]\!] \Longrightarrow$
$(\exists\, e \in arcs\ G.\ parent\text{-}edge\ v = Some\ e\ \wedge$
$head\ G\ e = v \wedge dist\ (tail\ G\ e) \neq \infty\ \wedge$
$num\ v\ = num\ (tail\ G\ e) + 1)$
assumes *noPedge*: $\bigwedge e.\ e{\in}arcs\ G \Longrightarrow$
$dist\ (tail\ G\ e) \neq \infty \Longrightarrow dist\ (head\ G\ e) \neq \infty$

sublocale *shortest-paths-locale-step1* $\subseteq$ *fin-digraph G*
⟨*proof*⟩

definition (**in** *shortest-paths-locale-step1*) *enum* :: $'a \Rightarrow enat$ **where**
$enum\ v = (if\ (dist\ v = \infty \vee dist\ v = -\ \infty)\ then\ \infty\ else\ num\ v)$

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

function (**in** *shortest-paths-locale-step1*) *pwalk* :: $'a \Rightarrow 'b\ list$
where
$pwalk\ v =$
$(if\ (v = s \vee dist\ v = \infty \vee v \notin verts\ G)$
$then\ []$
$else\ pwalk\ (tail\ G\ (the\ (parent\text{-}edge\ v)))\ @\ [the\ (parent\text{-}edge\ v)]$
$)$
⟨*proof*⟩
termination (**in** *shortest-paths-locale-step1*)
⟨*proof*⟩

lemma (**in** *shortest-paths-locale-step1*) *pwalk-simps*:
$v = s \vee dist\ v = \infty \vee v \notin verts\ G \Longrightarrow pwalk\ v = []$
$v \neq s \Longrightarrow dist\ v \neq \infty \Longrightarrow v \in verts\ G \Longrightarrow$
$pwalk\ v = pwalk\ (tail\ G\ (the\ (parent\text{-}edge\ v)))\ @\ [the\ (parent\text{-}edge\ v)]$
⟨*proof*⟩

definition (**in** *shortest-paths-locale-step1*) *pwalk-verts* :: $'a\ \Rightarrow\ 'a\ set$ **where**
$pwalk\text{-}verts\ v = \{u.\ u \in set\ (awalk\text{-}verts\ s\ (pwalk\ v))\}$

locale *shortest-paths-locale-step3* =
shortest-paths-locale-step2 +
fixes C :: $('a \times ('b\ awalk))\ set$
assumes *C-se*:
$C \subseteq \{(u, p).\ dist\ u \neq \infty \wedge awalk\ u\ p\ u \wedge awalk\text{-}cost\ c\ p < 0\}$
assumes *int-neg-cyc*:
$\bigwedge v.\ v \in verts\ G \Longrightarrow dist\ v = -\infty \Longrightarrow$

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

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

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 $num\ v > 0$
⟨*proof*⟩

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 $\exists\, e.\ s \to^* tail\ G\ e \wedge$
$e \in arcs\ G \wedge head\ G\ e = v \wedge dist\ (tail\ G\ e) \neq \infty \wedge$
$parent\text{-}edge\ v = Some\ e \wedge num\ v = num\ (tail\ G\ e) + 1$
⟨*proof*⟩

lemma (**in** *shortest-paths-locale-step1*) *path-from-root-Vr*:
fixes $v :: 'a$
assumes $v \in verts\ G$
assumes $dist\ v \neq \infty$
shows $s \to^* v$
⟨*proof*⟩

lemma (**in** *shortest-paths-locale-step1*) *μ-V-less-inf*:
fixes $v :: 'a$
assumes $v \in verts\ G$
assumes $dist\ v \neq \infty$
shows $\mu\ c\ s\ v \neq \infty$
⟨*proof*⟩

lemma (**in** *shortest-paths-locale-step2*) *enum-not0*:
assumes $v \in verts\ G$
assumes $v \neq s$
assumes $enum\ v \neq \infty$
shows $enum\ v \neq enat\ 0$
⟨*proof*⟩

lemma (**in** *shortest-paths-locale-step2*) *dist-Vf-*μ:
  fixes v :: $'a$
  assumes *vG*: $v \in$ *verts* G
  assumes $\exists\, r.$ *dist* $v =$ *ereal* r
  shows *dist* $v = \mu\ c\ s\ v$
⟨*proof*⟩

lemma (**in** *shortest-paths-locale-step1*) *pwalk-awalk*:
  fixes v :: $'a$
  assumes $v \in$ *verts* G
  assumes *dist* $v \neq \infty$
  shows *awalk* s (*pwalk* v) v
⟨*proof*⟩

lemma (**in** *shortest-paths-locale-step3*) μ*-ninf*:
  fixes v :: $'a$
  assumes $v \in$ *verts* G
  assumes *dist* $v = -\infty$
  shows $\mu\ c\ s\ v = -\infty$
⟨*proof*⟩

lemma (**in** *shortest-paths-locale-step3*) *correct-shortest-path*:
  fixes v :: $'a$
  assumes $v \in$ *verts* G
  shows *dist* $v = \mu\ c\ s\ v$
⟨*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.