A Mechanized Proof of the Max-Flow Min-Cut Theorem for Countable Networks

Andreas Lochbihler
Digital Asset (Switzerland) GmbH, Zurich, Switzerland

Abstract

Aharoni et al. [3] proved the max-flow min-cut theorem for countable networks, namely that in every countable network with finite edge capacities, there exists a flow and a cut such that the flow saturates all outgoing edges of the cut and is zero on all incoming edges. In this paper, we formalize their proof in Isabelle/HOL and thereby identify and fix several problems with their proof. We also provide a simpler proof for networks where the total outgoing capacity of all vertices other than the source is finite. This proof is based on the max-flow min-cut theorem for finite networks.

2012 ACM Subject Classification Mathematics of computing → Network flows; Theory of computation → Higher order logic; Theory of computation → Logic and verification

Keywords and phrases flow network, optimization, infinite graph, Isabelle/HOL

Digital Object Identifier 10.4230/LIPIcs.ITP.2021.1

Related Version A full version with informal proofs and more counterexamples is available [18].

Supplementary Material The formalization is available in the Archive of Formal Proofs [16].

Funding Swiss National Science Foundation grant 153217 "Formalising Computational Soundness for Protocol Implementations". This work was partially done while the author was at ETH Zurich.

Acknowledgements We thank Ron Aharoni and Eli Berger for helping to clarify the weaknesses in the original proofs. S. Reza Sefidgar and the anonymous reviewers helped to improve the presentation.

1 Introduction

The max-flow min-cut (MFMC) theorem for finite networks [10] has wide-spread applications: network analysis, optimization, scheduling, etc. Aharoni et al. [3] have generalized this theorem to countable networks, i.e., graphs with countably many vertices and edges, as follows:

▶ **Theorem 1.** *Let $\Delta = (V, E, s, t, c)$ be a directed graph with countably many edges $E \subseteq V \times V$, vertices s and t and a capacity function $c :: E \to \mathbb{R}_{\geq 0}$. There exists a flow f and an s-t-cut C such that f saturates all outgoing edges e of C, i.e. $f(e) = c(e)$, and is 0 on all incoming edges.*

The countable MFMC theorem is used, e.g., in probability [22] and programming language theory [17], privacy [7], and for random walks [21]. Here, we formalize this theorem in Isabelle.

Traditionally, the max-flow min-cut theorem is stated in terms of equality of values: The value of the maximum flow is equal to the value of the minimum cut. Here, a flow $f :: E \Rightarrow \mathbb{R}_{\geq 0}$ assigns values to the edges of Δ such that the incoming and outgoing amounts in every vertex are the same, except for the source s and the sink t. The value $|f|$ is the amount that leaves the source s, i.e., $|f| = \sum_{x \in \mathrm{OUT}(s)} f(s, x)$ where $\mathrm{OUT}(x) = \{y \mid (x, y) \in E\}$. Dually, an s-t-cut partitions the vertices into two sets $(C, V - C)$ such that C contains the source s but not the sink t. Its value $|C|$ is the total capacity of the edges that leave C: $|C| = \sum_{e \in \mathrm{OUT}(C)} c(e)$ where $\mathrm{OUT}(C) = \{(x, y) \in E \mid x \in C \wedge y \notin C\}$.

For finite networks, the equality-of-values condition $|f| = |C|$ is equivalent to the flow f saturating the cut C. In infinite networks, the saturation condition is preferable. For example, Fig. 1 shows a network with source s and sink t and countably many vertices x_i. The edge

© Andreas Lochbihler;
licensed under Creative Commons License CC-BY 4.0

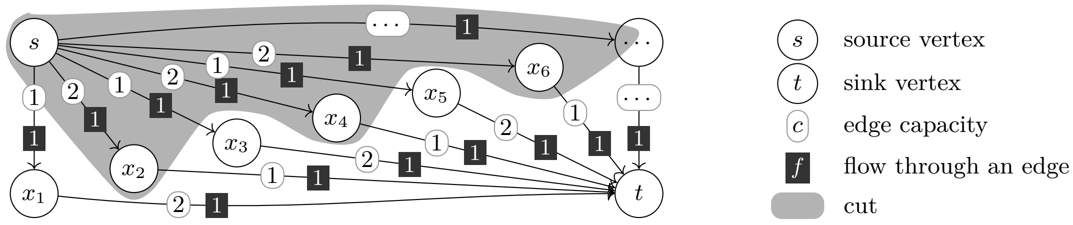

Figure 1 A countable network with a flow and a cut of infinite value.

capacities are given as white rounded rectangles on the edges. The black rectangles denote a flow f and the vertices in the grey area form a cut C. The flow f saturates the outgoing edges of C and we have $|f| = \infty = |C|$. However, there is another flow g given by $g(e) = {}^{1}\!/\!{}_{2} f(e)$ that sends only half the amount of f. Still, $|g| = \infty = |C|$. So the equality-of-values condition does not distinguish between f and g. Yet, we should consider only f a maximum flow, not g, as one can obviously increase g on some edges. The cut-saturation condition achieves this as it compares the finite capacities of individual edges with the flow through them.

This subtlety highlights the main challenge in proving the max-flow min-cut theorem for countable networks: avoiding infinite summations. Aharoni et al.'s proof performs an elaborate dance around this problem, transforming the network several times on the way. Our formalization follows these steps through all the transformations (Sect. 3) until the problem is reduced to finding some sort of matching in an infinite bipartite graph. The original proof then jumps back to arbitrary networks. Our proof forks into two proofs: The first takes a shortcut to a significantly simpler argument based on the max-flow min-cut theorem for finite networks (Sect. 4.1). This shortcut works only for networks where the sum of the capacities of the outgoing edges of any vertex other than the source is finite. This condition is met in some applications [7, 17]. The second proof follows the original (Sect. 4.2).

Our main contributions are as follows:

- We have formalized Aharoni et al.'s strong version of the max-flow min-cut theorem for countable networks in Isabelle/HOL. The resulting formalization is usable in other formalizations; e.g., we have applied it to the problem of proving parametricity of a probabilistic programming language with recursion [17]. The formalization has clarified the definitions and theorems and has revealed several problems in the original proofs (Sect. 6), which we have fixed. In particular, the reduction to bipartite graphs did not work as expected and required more general theorems.
- We give an alternative proof for the case when every inner vertex of a network has only finite total outgoing capacity. This local boundedness assumption allows us to reuse Lammich and Sefidgar's formalization of the max-flow min-cut theorem for finite networks [14] by applying a majorised convergence argument. This proof is considerably simpler and suffices for some use cases in programming languages and privacy [7, 17].

Neither of the two proofs requires a large background theory; basic notions like infinite summations, monotone and majorised convergence, and fixpoints of increasing functions suffice. The formalization therefore does not rely on specific Isabelle/HOL features and could have been done similarly in other systems like HOL4 and Coq.

The formalization started in 2015 and a first version was published in the Archive of Formal Proofs in 2016. This paper describes the cleaned-up version for Isabelle2021 [16], which also includes the simpler proof for the bounded case. This paper first presents the corrected proof using conventional mathematical notation (Sects. 2–4). We discuss the formalization aspects in Sect. 5 and the problems with the original proof in Sect. 6.

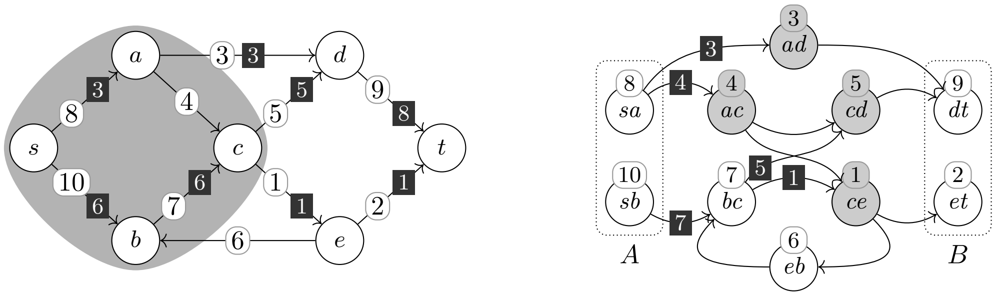

Figure 2 Example of a network (left) and a flow (values of 0 are omitted) with an orthogonal cut, and the corresponding web (right) with a maximal wave (black rectangles) and its set of terminal vertices (grey circles). Capacities and weights are shown as labels in rounded rectangles.

2 Graphs, Networks, and Webs

In this section, we introduce the relevant notions for graphs, networks, and webs. The terminology and notation follows [3] to ease the comparison and make the presentation accessible to mathematicians. Formalization considerations will be discussed in Sect. 5.

▶ **Definition 2** (Graph). *A* (directed) graph $G = (V, E)$ *consists of a set of vertices* V *and a set of directed edges* $E \subseteq V \times V$*. A graph is countable iff its set of edges is countable. The neighbours of a vertex* $x \in V$ *are given by* $\mathrm{OUT}_G(x) = \{\, y \mid (x, y) \in E \,\}$ *and* $\mathrm{IN}_G(x) = \{\, y \mid (y, x) \in E \,\}$*. If the graph* G *is obvious from the context, we drop the subscript* G*.*

Given a function $f :: E \to \mathbb{R}_{\geq 0}$*, the* in-degree $d^-_f :: V \to \mathbb{R}^\infty_{\geq 0}$ *of* f *given by* $d^-_f(x) = \sum_{y \in \mathrm{IN}(x)} f(y, x)$ *assigns to each vertex* $x \in V$ *the sum of* f *over all incoming edges to* x*. Analogously,* $d^+_f(x) = \sum_{y \in \mathrm{OUT}(x)} f(x, y)$ *denotes* f*'s* out-degree *of* $x \in V$*. If* $d^+_f(x) = 0$*, then* x *is a* sink *for* f*. The set* $\mathrm{SINK}(f)$ *denotes the set of sinks for* f*.*

▶ **Definition 3** (Network). *A* network $\Delta = (V, E, s, t, c)$ *is a graph* (V, E) *with two dedicated vertices, the source* s *and the sink* t*, and a capacity function* $c :: E \to \mathbb{R}_{\geq 0}$*. A network is countable iff the graph is countable.*

▶ **Definition 4** (Flow). *For a network* $\Delta = (V, E, s, t, c)$*, a* flow $f :: E \to \mathbb{R}_{\geq 0}$ *in* Δ *satisfies*

1. *(Capacity restriction)* $f(x, y) \leq c(x, y)$ *for all* $(x, y) \in E$*, and*
2. *(Kirchhoff's 1*st *law)* $d^-_f(x) = d^+_f(x)$ *for all* $x \in V - \{\, s, t \,\}$*.*

The value $|f|$ *of a flow* f *is* f*'s out-degree of* s*:* $|f| = d^+_f(s)$*.*

▶ **Definition 5** (Orthogonal cut). *In a network* $\Delta = (V, E, s, t, c)$*, a set of vertices* C *is a* cut *iff* $s \in C$ *and* $t \notin C$*. A cut* C *is* orthogonal *to a flow* f *iff* f *saturates all edges going out of* C *(i.e.,* $f(x, y) = c(x, y)$ *for all* $(x, y) \in E$ *with* $x \in C$ *and* $y \notin C$*) and* f *is zero on all edges entering* C *(i.e.,* $f(x, y) = 0$ *for all* $(x, y) \in E$ *with* $x \notin C$ *and* $y \in C$*).*

We have already seen an orthogonal pair of a flow of infinite value and a cut in Fig. 1. Another example of an orthogonal flow-cut pair of value 9 is shown in Fig. 2 on the left.

A network constrains the capacities of the edges in a graph, but the throughput of a vertex is unconstrained. So the sums on the two sides of Kirchhoff's first law may be infinite. To avoid such infinite sums, a web constrains the throughput of a vertex and leaves the edge capacity unconstrained. Section 3.1 explains how to convert between networks and webs.

▶ **Definition 6** (Web). *A* web $\Gamma = (V, E, A, B, w)$ *is a graph* (V, E) *with two sets of vertices* $A, B \subseteq V$ *(the sides* A *and* B*) and a weight function* $w :: V \to \mathbb{R}_{\geq 0}$*. We refer to the components of* Γ *by* V_Γ*,* E_Γ*,* A_Γ*,* B_Γ*, and* w_Γ*.*

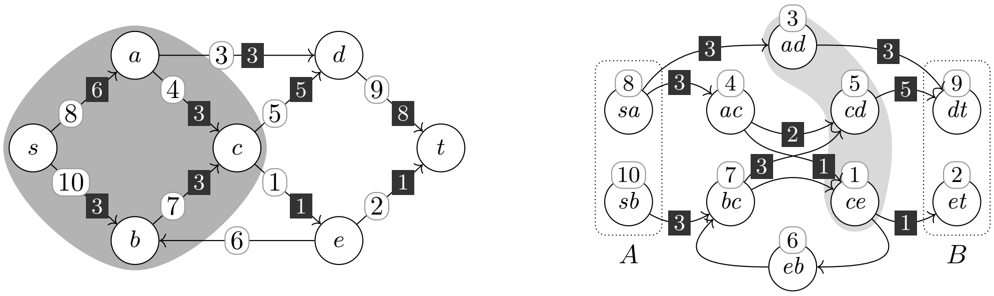

Figure 3 The network and web from Fig. 2 with a different flow (left) and a web-flow (right).

The two vertex sets A and B correspond to the source and sink of a network, respectively. Currents in a web take the role of flows in a network. The difference is that vertices may leak some of the incoming current (condition 2), i.e., they need not preserve the current.

▶ **Definition 7** (Current). *Given a web $\Gamma = (V, E, A, B, w)$, a* current $f :: E \to \mathbb{R}_{\geq 0}$ *satisfies*

1. *(weight restriction) $d^-_f(x) \leq w(x)$ and $d^+_f(x) \leq w(x)$ for all $x \in V$,*
2. *(flow reflection) $d^-_f(x) \geq d^+_f(x)$ for all $x \in V - A$, and*
3. *(side restriction) $d^-_f(x) = 0$ for $x \in A$ and $d^+_f(y) = 0$ for $y \in B$.*

A current f is called a web-flow *if $d^-_f(x) = d^+_f(x)$ for all $x \in V - (A \cup B)$. If $d^+_f(x) \geq w(x)$, then f* exhausts *x. If $x \in A$ or $d^-_f(x) \geq w(x)$, then f* saturates *x. A saturated sink x is called* terminal*. The set of saturated vertices is written as $\mathrm{SAT}(f)$ and the set of terminal vertices as $\mathrm{TER}(f) = \mathrm{SAT}(f) \cap \mathrm{SINK}(f)$.*

Figure 2 shows an example web on the right where the weight of the vertices are shown in rounded rectangles. It is derived from the network on the left as we will see in Sect. 3.1. The black rectangles specify a current f whose terminal vertices $\mathrm{TER}(f)$ are shown in grey. It exhausts none of the vertices. The current f is not a web-flow because some vertices are leaking, e.g., $d^-_f(bc) = 7 > 6 = d^+_f(bc)$.

Figure 3 shows a different flow and current for same network and web, respectively. The flow on the left differs from the one in Fig. 2 only in that three units are routed through (s, a) and (a, c) instead of through (s, b) and (b, c). So the vertex c now mixes the units coming from a with the three units coming from b and outputs five of them to d and one to e. On the right, a web-flow is shown, which refines the flow on the left as will be explained in Sect. 3.1. The light-grey area contains the exhausted vertices, namely ad, cd, and ce. There are no terminal vertices as the three sinks dt, et, and eb are disjoint from the saturated vertices sa, sb, ad, cd, and ce.

▶ **Definition 8** (Essential vertex). *Given sets of vertices S and B in a graph $G = (V, E)$, a vertex $x \in S$ is* essential *in S iff there is a path from x to a vertex in B which does not contain a vertex in $S - \{x\}$. The set of essential vertices of S is written as $\mathcal{E}_{G,B}(S)$.*

▶ **Definition 9** (Separation and roofing). *A set S of vertices in graph G* separates *a vertex x from a set of vertices B iff every path from x to a vertex in B contains a vertex in S. The set S is said to* separate *a set of vertices A from B iff it separates every vertex in A from B.*

The roofing *of S and B (notation $\mathrm{RF}_{G,B}(S)$) consists of all vertices which S separates from B. The* strict roofing *excludes essential vertices: $\mathrm{RF}^\circ_{G,B}(S) = \mathrm{RF}_{G,B}(S) - \mathcal{E}_{G,B}(S)$.*

In a web $\Gamma = (V, E, A, B, w)$, S is A-B-separating *iff it separates A and B. If f is a current in Γ, we abbreviate $\mathcal{E}(f) = \mathcal{E}_{\Gamma,B}(\mathrm{TER}(f))$ and $\mathrm{RF}(f) = \mathrm{RF}_{\Gamma,B}(\mathrm{TER}(f))$ and $\mathrm{RF}^\circ(f) = \mathrm{RF}^\circ_{\Gamma,B}(\mathrm{TER}(f))$.*

In the web in Fig. 2, the grey vertices $\mathrm{TER}(f)$ separate A from B. The vertex ac is not essential in $\mathrm{TER}(f)$ as all paths from ac to B pass either through cd or ce, which are both in $\mathrm{TER}(f)$. The roofing $\mathrm{RF}(f)$ contains all the vertices to the left of ad, cd, and ce, inclusive, i.e., $\mathrm{RF}(f) = \{sa, sb, ac, bc, ad, eb, cd, ce\}$. The strict roofing $\mathrm{RF}^\circ(f)$ excludes the essential vertices ad, eb, and ce. Since ac is not essential in $\mathrm{TER}(f)$, the strict roofing includes ac.

▶ **Lemma 10** ([2, Lemma 2.14]). *If S separates A from B in G, so does $\mathcal{E}_{G,B}(S)$.*

The key tool for the proof is the concept of a wave. Waves are currents whose terminal vertices separate A from B and which are zero outside of the roofing of the terminal vertices. Intuitively, a wave's essential terminal vertices identify a bottleneck in the web: since the wave saturates them, all other separating sets between the A side and the terminal vertices must allow at least the same current.

▶ **Definition 11** (Wave). *A current f in Γ is a* wave *iff $\mathrm{TER}(f)$ is A-B-separating and $d^+_f(x) = 0$ for $x \notin \mathrm{RF}(f)$.*

In Fig. 2, the current f is 0 outside of $\mathrm{RF}(f)$, i.e., on the edges entering B. So f is a wave. Conversely, the web-flow g in Fig. 3 is not a wave as $\mathrm{TER}(g) = \{\}$ does not separate A from B.

3 From Networks to Bipartite Webs and Back

Aharoni et al.'s proof proceeds in four steps [3]:

1. Transform the network into a web.
2. Find a maximal wave in the web. Its roofing determines the cut.
3. Trim the wave, i.e., reduce the wave such that strictly roofed vertices preserve the current.
4. Extend the wave to a web-flow. This uses a reduction to bipartite webs in which every current is a web-flow by definition.

In this section, we cover these steps up to the reduction to bipartite webs. The next section takes care of actually finding a suitable current in the bipartite web.

3.1 From Networks to Webs

The first step reduces a network Δ to a web, which we denote by $\mathrm{web}(\Delta)$. Every edge e becomes a vertex of $\mathrm{web}(\Delta)$ with weight $c(e)$. Every two incident edges (x, y) and (y, z) in the network induce an edge between the vertices (x, y) and (y, z) in $\mathrm{web}(\Delta)$. The side A consists of the edges leaving s and B of the edges entering t. Formally:

$$V_{\mathrm{web}(\Delta)} = E_\Delta \qquad w_{\mathrm{web}(\Delta)}(e) = c(e) \qquad A_{\mathrm{web}(\Delta)} = \{(s, y) \mid (s, y) \in E_\Delta\}$$
$$E_{\mathrm{web}(\Delta)} = \{((x, y), (y, z)) \mid (x, y) \in E_\Delta \wedge (y, z) \in E_\Delta\} \qquad B_{\mathrm{web}(\Delta)} = \{(x, t) \mid (x, t) \in E_\Delta\}$$

For example, Figs. 2 and 3 show the same network Δ on the left and the corresponding web $\mathrm{web}(\Delta)$ on the right. Webs have the advantage over networks that the current makes explicit how the incoming flow is split up into the outgoing edges of a vertex. In Fig. 3, e.g., the web-flow on the right specifies that the three units flowing from sa to ac split up into two units going to cd and one unit going to ce. The flow in the network on the left cannot express this detail: the vertex c mixes the two incoming flows of 3 units each and distributes somehow into five and one outgoing units.

Webs therefore allow us to capture flow preservation more precisely than networks. For if a flow f through a network vertex x is infinite, then flow preservation at x merely states that both sums are infinite: $d^-_f(x) = d^+_f(x) = \infty$. This creates problems if we want to

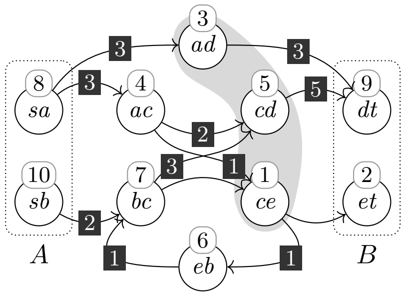

Figure 4 A separating set (grey area) that is not orthogonal to the shown web-flow.

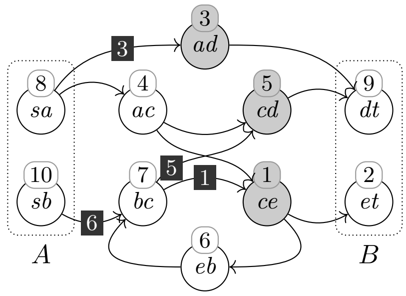

Figure 5 A trimming of the wave from Fig. 2.

subtract two infinite flows f and g from one another because $d^-_f(x) - d^-_g(x) = \infty - \infty$ is not meaningful. So even if both f and g satisfy Kirchhoff's first law at a vertex, it is not clear that their difference $f - g$ satisfies it. In the corresponding web, in contrast, a web-flow g specifies precisely the finite amount each incoming edge contributes to each outgoing edge. So for a web-flow or current g, the sums $d^-_g(x)$ and $d^+_g(x)$ are finite because they are bounded by the finite vertex weights, i.e., the edge capacities in the network. Accordingly, subtraction of flows has nice algebraic properties such as $d^-_f(x) - d^-_g(x) = d^-_{f-g}(x)$ if $f \geq g$.

We next transfer the orthogonality notion from networks to webs. We show that an A-B-separating set S and an orthogonal web-flow f in $\mathrm{web}(\Delta)$ induce a cut $\hat{S}$ and an orthogonal flow $\hat{f}$ in the original network Δ. Figure 3 illustrates the reduction: The flow $\hat{f}$ in the network Δ on the left corresponds to the web-flow f in $\mathrm{web}(\Delta)$ on the right. The set $\mathcal{E}(\mathrm{SAT}(f))$ in grey on the right is orthogonal to the web-flow f and yields the cut $\hat{S}$ on the left.

▶ **Definition 12** (Orthogonal current). *Let $\Gamma = (V, E, A, B, w)$ be a web. A set of vertices S is* orthogonal *to a current f iff*

(i) $d^-_f(x) = w(x)$ *for* $x \in S - A$,
(ii) $d^+_f(x) = w(x)$ *for* $x \in (S \cap A) - B$, *and*
(iii) $f(x, y) = 0$ *for* $x \in V - \mathrm{RF}^\circ(S)$ *and* $y \in \mathrm{RF}(S)$.

Intuitively, an orthogonal current exhausts the vertices in S unless the vertex belongs to both sides. Condition (iii) ensures that nothing flows back into the roofed vertices. For example, the web-flow in Fig. 4 is not orthogonal to the vertices in the grey area, because one unit flows from the essential vertex *ce* back to the roofed vertex *eb*.

▶ **Lemma 13** (Reduction from networks to webs). *Let $\Delta = (V, E, s, t, c)$ be a network with $s \neq t$ and no outgoing edge from t and no direct edge from s to t. Suppose that all edges have positive capacity, i.e., $c(e) > 0$ for $e \in E$.*

(a) *Let f be a web-flow in $\mathrm{web}(\Delta)$. Define $\hat{f}$ by $\hat{f}(e) = \max(d^+_f(e), d^-_f(e))$ for $e \in E$. Then, $\hat{f}$ is a flow in Δ.*
(b) *Let S be an A-B-separating set in $\mathrm{web}(\Delta)$. Define $\hat{S} = \mathrm{RF}_{\Delta,\{t\}}(\{x \mid \exists y.\, (x, y) \in \mathcal{E}(S)\})$. Then $\hat{S}$ is a cut in Δ.*
(c) *Let an A-B-separating set S be orthogonal to a web-flow f. Then $\hat{S}$ is orthogonal to $\hat{f}$.*

By this lemma, to find a cut and an orthogonal flow in a network Δ, it suffices to find a separating set of vertices in $\mathrm{web}(\Delta)$ and an orthogonal web-flow f. In the next section, we focus on finding a suitable separating set, namely the terminal vertices of a maximal wave.

3.2 Maximal Waves and Trimmings

Waves and currents can be ordered pointwise: if f and g are waves or currents in $\Gamma = (V, E, A, B, w)$, then $f \leq g$ iff $f(e) \leq g(e)$ for all $e \in E$. The waves in a countable web form a chain-complete partial order (ccpo), and so do the currents. Therefore, every countable web contains a maximal wave [3, Cor. 4.4] by Zorn's lemma.

Recall that a wave's terminal vertices describe a bottleneck in the web. Intuitively, the maximal wave identifies a narrowest bottleneck in the web: Roughly speaking, the roofed part cannot contain a tighter bottleneck because if so, the current could not saturate the terminal vertices due to the flow reflection condition. Conversely, if a separating set beyond the terminal vertices formed a tighter bottleneck, then we could extend the wave and saturate that smaller bottleneck, which contradicts maximality. Here, it is crucial that a wave may partially leak the incoming current of some vertices, i.e., they need not preserve the current.

A trimming of a wave reduces the current such that the incoming current is preserved on the strict roofing. For example, the wave in Fig. 2 on the right is maximal. Its trimming is shown in Fig. 5. The current is reduced on the edge from sb to bc from 7 to 6 and on the edge from sa to ac from 4 to 0.

▶ **Definition 14** (Trimming). *Let f be a wave in $\Gamma = (V, E, A, B, w)$. A wave g is called a* trimming *of f iff*
 (i) $g \leq f$,
 (ii) $d^+_g(x) = d^-_g(x)$ *for all* $x \in \mathrm{RF}^\circ(f) - A$, *and*
 (iii) $\mathcal{E}(\mathrm{TER}(g)) - A = \mathcal{E}(\mathrm{TER}(f)) - A$.

▶ **Lemma 15** ([3, Lemma 4.8]). *Every wave in a countable web has a trimming.*

Proof. The trimming for a wave f is constructed as the transfinite fixpoint iteration of the one-step trimming function $trim_1$ starting at f. For a wave g, $trim_1(g)$ picks some strictly roofed vertex z where Kirchhoff's first law does not hold, i.e., $z \in \mathrm{RF}^\circ(g) - A \wedge d^+_g(z) \neq d^-_g(z)$. Then, $trim_1$ reduces the current on z's incoming edges by the factor $\frac{d^+_g(z)}{d^-_g(z)}$ so that Kirchhoff's first law holds at z afterwards.

$$trim_1(g)(y, x) = \begin{cases} g(y, x) & \text{if } g \text{ is a trimming} \\ \mathsf{if}\ x = z\ \mathsf{then}\ \frac{d^+_g(z)}{d^-_g(z)} * g(y, x)\ \mathsf{else}\ g(y, x) & \text{if such a } z \text{ exists} \end{cases}$$

The fixpoint exists by Bourbaki-Witt's fixpoint theorem [8] as $trim_1$ is decreasing, i.e., $trim_1(g) \leq g$, and the set of waves g with $g \leq f$ is a chain-complete partial order w.r.t. $\geq$. The proof that the fixpoint satisfies the trimming conditions relies on d^+ and d^- being pointwise order-continuous, which holds by monotone convergence as the web is countable. ◀

3.3 A Linkage in the Quotient of a Web

The trimming of a maximal wave f describes the first half of the web-flow we are looking for (Fig. 5). For the second half, we consider the residual web beyond f's terminal vertices, which is called the quotient Γ/f. Figure 6 shows the quotient for the web and wave f from Fig. 2. The essential terminal vertices of the wave become the side A. The quotient does not include the roofed vertex eb even though it is reachable from $\mathcal{E}(\mathrm{TER}(f))$ as we want to construct an orthogonal current and nothing may flow back into roofed vertices. The formal definition is a bit complicated so that it also works when there are edges between vertices in $\mathcal{E}(\mathrm{TER}(f))$ or when $\mathcal{E}(\mathrm{TER}(f))$ contains vertices from B. The details are discussed in Sect. 6.

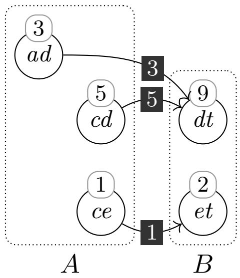

Figure 6 The quotient of the web and wave of Fig. 2 with a linkage.

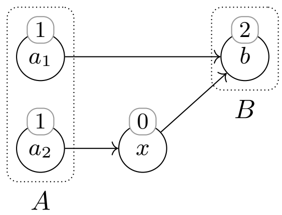

Figure 7 A web that contains no non-zero wave, but the zero wave is a hindrance.

▶ **Definition 16** (Quotient). *Let $\Gamma = (V, E, A, B, w)$ and f be a wave in Γ. The quotient Γ/f is the following web:*

- $E_{\Gamma/f} = \{(x, y) \in E \mid x \notin \mathrm{RF}^{\circ}_{\Gamma}(f) \wedge y \notin \mathrm{RF}_{\Gamma}(f)\}$
- $A_{\Gamma/f} = \mathcal{E}_{\Gamma}(\mathrm{TER}_{\Gamma}(f)) - (B - A)$ *and* $B_{\Gamma/f} = B$
- $w_{\Gamma/f}(x) = w(x)$ *for* $x \in V - (\mathrm{RF}^{\circ}_{\Gamma}(f) \cup (\mathrm{TER}_{\Gamma}(f) \cap B))$ *and* $w_{\Gamma/f}(x) = 0$ *for* $x \in \mathrm{TER}_{\Gamma}(f) \cap B$.

In the quotient Γ/f, we now look for a web-flow g that saturates all vertices in A, i.e., $\mathrm{TER}(f)$. Such a web-flow is called a linkage. Then, the web-flow in Γ is given by the trimming of f plus g. Figure 6 shows such a linkage; together with the trimmed wave from Fig. 5, they form the orthogonal web-flow whose reduction (Lemma 13) yields the network flow shown in Fig. 2.

▶ **Definition 17** (Linkage [3, Def. 4.1]). *A web-flow f in a web $\Gamma = (V, E, A, B, w)$ is called a* linkage *iff f exhausts all vertices in A, i.e., $d^{+}_{f}(a) = w(a)$ for all $a \in A$.*

Under what conditions does a web Γ contain a linkage? Certainly, there must not be a bottleneck beyond the A side. Waves describe such bottlenecks. So if the zero wave is the only wave in Γ, then the A side is the only bottleneck. Moreover, we need that all vertices in A are essential for separation unless their weight is 0. For example, the web in Fig. 7 contains only the zero wave, but not a linkage. The problem is that the vertex a_2 with weight 1 is bottlenecked by the zero-weight vertex $x \in \mathcal{E}(\mathrm{TER}(\mathbf{0}))$. Such a situation is called a hindrance.

▶ **Definition 18** (Hindrance, looseness, [3, Def. 4.5]). *A wave f in a web $\Gamma = (V, E, A, B, w)$ is a* $>\varepsilon$-hindrance *iff there is a vertex $a \in A - \mathcal{E}(\mathrm{TER}(f))$ such that $\varepsilon < w(a) - d^{+}_{f}(a)$. Also, f is a* hindrance *iff there exists a $\varepsilon > 0$ such that f is a $>\varepsilon$-hindrance. A web is called* hindered *(respectively* $>\varepsilon$-hindered*) iff it contains a hindrance (respectively a $>\varepsilon$-hindrance). A web is called* loose *iff it contains no non-zero wave and the zero wave is not a hindrance.*

▶ **Lemma 19** ([3]). *If f is a maximal wave in the web $\Gamma = (V, E, A, B, w)$, then Γ/f is loose.*

3.4 Reduction to Bipartite Webs

To find linkages in countable loose webs, Aharoni et al. [3] transform webs into bipartite webs. A web $\Omega = (V, E, A, B, w)$ is *bipartite* iff there are only edges from nodes in A to nodes in B, i.e., iff $V = A \cup B$ and $A \cap B = \emptyset$ and $E \subseteq A \times B$.

We briefly review the transformation described in [1]; Fig. 8 shows an example. In this section, we always assume that the web $\Gamma = (V, E, A, B, w)$ has no incoming edges to vertices in A, no outgoing edges from vertices in B, no loops, and that A and B are

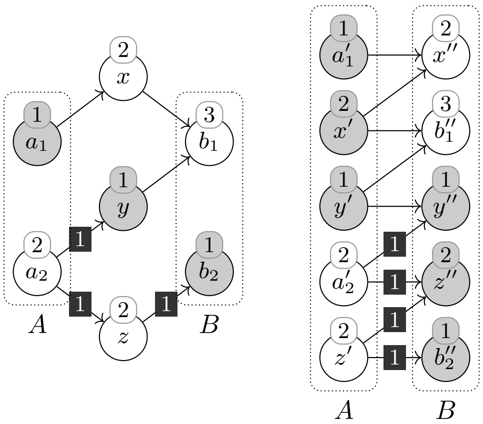

Figure 8 An unhindered web Γ (left) and its bipartite reduction $\mathrm{bp}(\Gamma)$ (right). The wave f in $\mathrm{bp}(\Gamma)$ induces the wave $\tilde{f}$ in Γ.

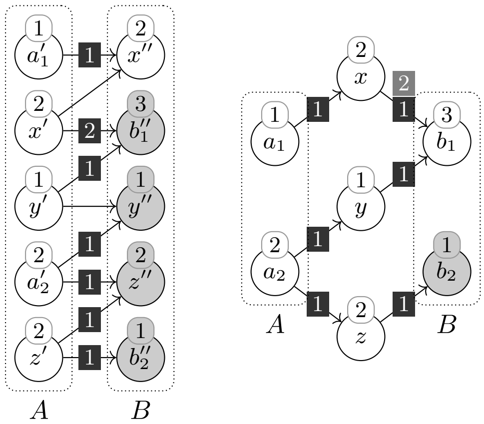

Figure 9 A linkage g in $\mathrm{bp}(\Gamma)$ (left) that yields a linkage (right) in the web Γ from Fig. 8 by trimming $\tilde{g}$ at vertex x.

disjoint. In the bipartite web $\mathrm{bp}(\Gamma)$, there are two copies x' and x'' for every vertex $x \in V - (A \cup B)$. Vertices $x \in A$ and $y \in B$ only have one copy x' and y'', respectively. The edges are $E_{\mathrm{bp}(\Gamma)} = \{(x', y'') \mid (x, y) \in E\} \cup \{(x', x'') \mid x \in V - (A \cup B)\}$ and the sides $A_{\mathrm{bp}(\Gamma)} = \{x' \mid x \in V - B\}$ and $B_{\mathrm{bp}(\Gamma)} = \{x'' \mid x \in V - A\}$ and the weight function $w(x') = w(x)$ for $x \in V - B$ and $w(x'') = w(x)$ for $x \in V - A$.

An A-B-separating set S in $\mathrm{bp}(\Gamma)$ induces an A-B-separating set $\widetilde{S}$ in Γ given by $\widetilde{S} = (A_S \cap B_S) \cup (A \cap A_S) \cup (B \cap B_S)$ where $A_S = \{v \mid v' \in S\}$ and $B_S = \{v \mid v'' \in S\}$ [1]. Moreover, a wave f in $\mathrm{bp}(\Gamma)$ induces a wave $\tilde{f}$ in Γ given by $\tilde{f}(x, y) = f(x', y'')$ for $(x, y) \in E$ with $\mathrm{TER}_\Gamma(\tilde{f}) = \mathrm{TER}_{\mathrm{bp}(\Gamma)}(f)$ [3, Lemma 6.3].

▶ **Lemma 20.** *If* Γ *is loose, then* $\mathrm{bp}(\Gamma)$ *is unhindered.*

Aharoni et al. wrongly claimed the stronger statement that if Γ is loose then $\mathrm{bp}(\Gamma)$ is loose [3, below Thm. 6.5]. We provide a counterexample in Sect. 6. Note that the reduction bp does not preserve unhinderedness either.

Conversely, a linkage g in $\mathrm{bp}(\Gamma)$ yields a linkage in Γ as illustrated in Fig. 9: For $\tilde{g}$ as defined above, we have $d^+_{\tilde{g}}(a) = d^+_g(a') = w(a)$ for $a \in A_\Gamma$ and $d^+_{\tilde{g}}(x) \geq d^-_{\tilde{g}}(x)$ for all $x \notin B$. So the out-flow of some vertices may surpass the in-flow, e.g., x in Fig. 9. Analogously to the trimming of waves, we can trim $\tilde{g}$ using a fixpoint iteration to obtain the linkage in Γ.

▶ **Lemma 21** ([3]). *If* $\mathrm{bp}(\Gamma)$ *contains a linkage and* Γ *is countable, then* Γ *contains a linkage.*

4 Linkability in unhindered bipartite webs

By the results in Sect. 3, the max-flow min-cut theorem for the countable case (Thm. 1) follows from the following theorem, which we prove in this section.

▶ **Theorem 22** (Bipartite linkability)**.** *A countable unhindered bipartite web contains a linkage.*

In fact, we present two ways how to construct such a linkage in an unhindered bipartite web. Both ways enumerate the vertices in $A = \{a_1, a_2, a_3, \ldots\}$ and construct a sequence of web-flows f_i that exhaust $\{a_1, \ldots, a_i\}$ so that the limit f exhausts all of A. The difference is in how the f_i are constructed and in the limit argument. In Sect. 4.1, each f_i is constructed independently as the limit of maximum flows in a finite network; the existence and the linkage property of the limit for these f_i themselves is shown using diagonalization and majorised convergence. Unfortunately, this construction only works if the neighbours of any a_i vertex have finite total weight.

In contrast, f_{i+1} in Sect. 4.2 saturates a_{i+1} by extending the previous web-flow f_i with a sequence of augmenting flows in the so-called residual network, similar to how classic max-flow algorithms for finite networks work [9]. This construction avoids taking infinite summations and thus yields a proof of Thm. 22 without additional assumptions. However, the proof is more involved than in the bounded case.

4.1 The Bounded Case

We first prove Thm. 22 for the case where the neighbours of each vertex in A have only bounded total weight, i.e., $\sum_{y \in \mathrm{OUT}(x)} w(y) < \infty$ for all $x \in A$. The general case is shown in the next section.

The next lemma states the crucial property of unhindered bipartite webs, namely that the total weight of any finite set of A vertices is at most the total weight of their neighbours in B.

▶ **Lemma 23.** *Let $\Omega = (V, E, A, B, w)$ be a countable unhindered bipartite web and $X \subseteq A$ be finite. Then, $\sum_{x \in X} w(x) \le \sum_{y \in E[X]} w(y)$ where $E[X] = \{y \mid \exists x \in X.\ (x, y) \in E\}$ denotes the neighbours of X.*

This lemma allows us to understand a linkage in an unhindered bipartite web as an $A \times B$ matrix over the reals where the weights on A are the row sums of the countable matrix and the edges describe the matrix elements that may be non-zero. In the proof below, we will use the following result about the existence of a countable matrix with given marginals. It is a corollary of a theorem by Kellerer [12, Satz 4.1]. In the formalization, we have proved the corollary directly by adapting Kellerer's proof to this special case. This proof uses the max-flow min-cut theorem for *finite* networks.

▶ **Proposition 24** (Matrix with given marginals)**.** *Let $f : A \to \mathbb{R}_{\ge 0}$ and $g : B \to \mathbb{R}_{\ge 0}$ for countable sets A, B such that $\sum_{i \in A} f(i) = \sum_{j \in B} g(j) < \infty$, and let $R \subseteq A \times B$. Assume that $\sum_{i \in X} f(x) \le \sum_{j \in R[X]} g(j)$ for all $X \subseteq A$. Then, there exists a function $h : A \times B \to \mathbb{R}_{\ge 0}$ such that for all $i \in A$ and $j \in B$:*

- *$h(i, j) = 0$ if $(i, j) \notin R$,*
- *$f(i) = \sum_{j \in \mathbb{N}} h(i, j)$, and*
- *$g(j) = \sum_{i \in \mathbb{N}} h(i, j)$.*

We can now prove bipartite linkability in the bounded case. The proof starts with a sequence of increasing finite subsets A_n of A that converge to A, and suitable, possibly infinite subsets B_n of their neighbours in B. For these subsets, we obtain a $A_n \times B_n$ matrix h_n with the right marginals. This sequence h_n converges and its limit yields the desired linkage, using a majorised convergence argument with the bound on the neighbours.

▶ **Theorem 25** (Bounded bipartite linkability)**.** *A countable unhindered bipartite web $\Omega = (V, E, A, B, w)$ contains a linkage if $\sum_{y \in \mathrm{OUT}(x)} w(y) < \infty$ for all $x \in A$.*

Together with the reduction from Sect. 3, this yields a proof for Thm. 1 when only the source s in the network $\Delta = (V, E, s, t, c)$ may have outgoing edges whose total capacity is infinite, i.e., $d_c^+(x) < \infty$ for $x \in V - \{s\}$. The MFMC use cases in probability theory [22] and privacy [7] satisfy this condition.

4.2 The Unbounded Case

We now show that Thm. 22 holds even when the neighbours of a vertex have infinite total weight. Our proof generalizes Aharoni et al.'s from loose to unhindered bipartite webs. For the remainder of this section, we always assume that $\Omega = (V, E, A, B, w)$ is a countable bipartite web. We write $\Omega \ominus f$ for the bipartite web Ω where the weight of the vertices has been reduced by the current f that flows through them.

▶ **Definition 26** (Residual web). *If $\Omega = (V, E, A, B, w)$ is a bipartite web and f a current in Ω, we write $\Omega \ominus f$ for the web (V, E, A, B, w') where the new weight function w' is given by $w'(x) = w(x) - d^+_f(x)$ for $x \in A$ and $w'(x) = w(x) - d^-_f(x)$ for $x \in B$.*

The proof rests on the following step: If Ω is unhindered, then we can find a current f that saturates some vertex $a \in A$ such that the residual web $\Omega \ominus f$ is unhindered again.

▶ **Lemma 27** (Vertex saturation in unhindered bipartite webs). *If Ω is unhindered and $a \in A$, then there exists a current f in Ω such that $d^+_f(a) = w(a)$ and $\Omega \ominus f$ is unhindered.*

With this lemma, we can now prove that countable unhindered bipartite webs are linkable (Thm. 22). The proof is analogous to [3, Thm. 6.5], but uses our Lemma 27 instead.

Proof of Thm. 22. Enumerate the vertices in A as $a_1, a_2, \ldots$. Recursively define a family f_n of currents in Ω as follows:

(i) f_0 is the zero current.

(ii) For $n > 0$, pick a current g_n in $\Omega \ominus f_{n-1}$ such that $d^+_g(a_n) = w_{\Omega \ominus f_{n-1}}(a_n)$ and $\Omega \ominus f_{n-1} \ominus g$ is unhindered. Set $f_n = f_{n-1} + g$.

A simple induction on n shows that f_n is a well-defined current in Ω and $\Omega \ominus f_n$ is unhindered for all n; here, Lemma 27 applied to $\Omega \ominus f_{n-1}$ ensures that g_n exists. Set $g(e) = \sup\{f_n(e) \mid n \in \mathbb{N}\}$ for $e \in E$. Then, g is a current in Ω with $d^+_g(x) = w(x)$ for all $x \in A$. As every current in a bipartite web is a web-flow, g is the linkage we are looking for. ◀

The proof of the saturation lemma 27 uses the following theorems and lemmas, which have already been proven by Aharoni et al. [3]. We have formalized all of them and fixed the glitches in the original statements and proofs.

▶ **Theorem 28** (Flow attainability [3, Thm. 5.1]). *Let $\Delta = (V, E, s, t, c)$ be a countable network with $s \neq t$, no loops and no incoming edges to s, and such that for all $x \in V - \{t\}$, the sum of capacities of the incoming edges to x or the sum of capacities of the outgoing edges from x is finite, i.e., $d^-_c(x) < \infty$ or $d^+_c(x) < \infty$. Then there exists a flow f in Δ such that $d^+_f(s) = \sup\{|g| \mid g \text{ is a flow in } \Delta\}$ and $d^-_f(x) \leq |f|$ for all $x \in V$.*

▶ **Lemma 29** ([3, Lemma 6.7]). *Let $\Omega = (V, E, A, B, w)$ be a countable bipartite web and let $u :: V \to \mathbb{R}_{\geq 0}$ such that $u(x) = 0$ for $x \in A$, $u(y) \leq w(y)$ for $y \in B$, and $\varepsilon = \sum_{x \in B} u(x) < \infty$. Let $\Omega' = (V, E, A, B, w - u)$ be the web Ω with w reduced by u. If Ω' is $>\varepsilon$-hindered, then Ω is hindered.*

▶ **Lemma 30** ([3, Cor. 6.8]). *Let g be a current in Ω with $\varepsilon := \sum_{b \in B} d^-_g(b) < \infty$. If $\Omega \ominus g$ is $>\varepsilon$-hindered, then Ω is hindered.*

▶ **Lemma 31** ([3, Lem 6.9]). *Let Ω be loose and $b \in B$ with $w(b) > 0$. For every $\delta > 0$, there exists an $\varepsilon > 0$ such that $\varepsilon < \delta$ and Ω with the weight of b reduced by ε is unhindered.*

5 Discussion of the Formalization

We have formalized all definitions, theorems, and proofs mentioned in this paper in Isabelle/HOL. This includes all the lemmas and underlying theory. In this section, we discuss the challenges we faced and the design decisions we made. The issues with the original definitions, theorems, and proofs and their corrections are discussed in the next section.

Graphs are formalized using Isabelle's record package [20] as an extensible record with one field for the edge relation, given as a binary predicate over the vertices of type α. This yields the projection function $\mathsf{edge} :: \alpha\ \mathsf{graph} \Rightarrow \alpha \Rightarrow \alpha \Rightarrow \mathsf{bool}$ for the edge field.[1] From this, we derive the set E of edges as an abbreviation.

record α $\mathsf{graph} = \mathsf{edge} :: \alpha \Rightarrow \alpha \Rightarrow \mathsf{bool}$
definition $\mathsf{vertex} :: \alpha\ \mathsf{graph} \Rightarrow \alpha \Rightarrow \mathsf{bool}$ where $\mathsf{vertex}\ G\ x = (\exists y.\ \mathsf{edge}\ G\ x\ y \lor \mathsf{edge}\ G\ y\ x)$
type-synonym $\alpha\ \mathsf{edge} = \alpha \times \alpha$
abbreviation $\mathsf{E} :: \alpha\ \mathsf{graph} \Rightarrow \alpha\ \mathsf{edge\ set}$ where $\mathsf{E}_G = \{(x, y).\ \mathsf{edge}\ G\ x\ y\}$

We derive the set of vertices from edges of the graph rather than modelling them separately. This has the advantage that we encode the condition $E \subseteq V \times V$ in the construction and do not have to carry around this well-formedness condition in our formalization. Conversely, graphs in this model cannot have isolated vertices. This is without loss of generality as isolated vertices cannot contribute to any flow or cut.

Networks are formalized as an extension of the record graph. So all operations on graphs also work for networks. The same applies to webs.

record α $\mathsf{network} = \alpha\ \mathsf{graph}\ +$
  $\mathsf{capacity} :: \alpha \Rightarrow \mathsf{ennreal}$
  $\mathsf{source} :: \alpha$
  $\mathsf{sink} :: \alpha$

record α $\mathsf{web} = \alpha\ \mathsf{graph}\ +$
  $\mathsf{weight} :: \alpha \Rightarrow \mathsf{ennreal}$
  $\mathsf{A} :: \alpha\ \mathsf{set}$
  $\mathsf{B} :: \alpha\ \mathsf{set}$

Records provide a simple and lightweight means for grouping the components of a network or web. Particular properties such as countability, finite capacity and weights, and disjoint sides A and B, are formalized as locales [5]. For example, the locale countable-network below enforces that there are only countably many edges, the source is not the sink, and the capacities are finite and 0 outside of the edges. Using the **(structure)** annotation on a record variable like Δ [4], we can omit the network (or web) as subscripts, e.g., in the assumption countable E; Isabelle automatically fills in the corresponding parameter. We use this notational convenience mainly for definitions that need custom syntax anyway, e.g., $\mathcal{E}$, RF, and RF$^\circ$. For plain HOL functions without special syntax like capacity and source, it is usually faster to type the record parameter than to enter special syntax.

locale countable-network = **fixes** $\Delta :: \alpha\ \mathsf{network}$ **(structure)**
  assumes countable E **and** $\mathsf{source}\ \Delta \neq \mathsf{sink}\ \Delta$
    and $e \notin \mathsf{E} \Longrightarrow \mathsf{capacity}\ \Delta\ e = 0$ **and** $\mathsf{capacity}\ \Delta\ e < \infty$

Since flows, cuts, and capacities are always non-negative, we use the extended non-negative reals ennreal from Isabelle/HOL's library everywhere. Summations like the in-degree d^- are

[1] The record package achieves extensibility with structural subtyping by internally generalizing α graph to (α, β) graph-scheme, where β is the extension slot for further fields. For example, β is instantiated with the singleton type unit for graph. All operations on graph are actually defined on graph-scheme so that they also work for all record extensions. We omit this technicality from the presentation.

expressed using the Lebesgue integral nn-integral over the counting measure count-space A on the set A. So every subset of A is measurable and all points have equal weight. Moreover, every function is integrable and we need not discharge neither integrability nor summability conditions in the proofs. Just the finiteness conditions of the form $\sum_{x \in A} < \infty$ are ubiquitous.

We also formalize capacities and weights as ennreal and explicitly require them being finite in the locales. This avoids coercions from the real numbers real into ennreal, which would complicate the proof formalization. For example, the in-degree $d^-_f(f)$ of y is defined as follows where $\sum_{x \in A} g$ desugars to nn-integral (count-space A) $(\lambda x.\ g)$. We let the summation range over UNIV, the set of all values of α, not only the neighbours of y. Instead, we enforce that f is 0 outside of E, e.g., via the capacity assumption in countable-network. This way, d-IN depends only on f and not on the graph. This simplifies the formalization because when we consider f in the context of different graphs, d-IN f is trivially the same for all of them.

definition d-IN :: (α edge $\Rightarrow$ ennreal) $\Rightarrow \alpha \Rightarrow$ ennreal where d-IN $f\ y = \sum_{x \in \mathsf{UNIV}} f\ (x, y)$

Regarding the mathematical background theory, we found that most relevant theorems were readily available in the Isabelle/HOL library: limits, infinite summations via the Lebesgue integral, monotone and majorised convergence, lim sup and lim inf. There is even a generic formalization of Cantor's diagonalization argument by Immler [11]. The Bourbaki-Witt fixpoint theorem [8], however, was missing. We therefore ported the Coq formalization by Smolka et al. [23] to Isabelle/HOL. It is now part of Isabelle/HOL's library. We have also contributed many lemmas about ennreal and nn-integral to the library.

Apart from identifying and fixing glitches and mistakes in definitions and proofs (Sect. 6), we faced three main challenges during the formalization. First, the definition and proof principles in the paper are often not suitable for direct formalization. For example, the original proofs construct trimmings, linkages and saturating flows using transfinite iteration and transfinite induction with ordinals. We have replaced them with fixpoints of increasing or decreasing functions in a chain-complete partial order, using Bourbaki-Witt's fixpoint theorem (Lemmas 15, 21, and 27). This way, we did not need to formalize ordinals and their theory.

Second, applying the theorems from the Isabelle library often needs a small twist. The proof for the existence of a maximal wave in Sect. 3.2 demonstrates this. The proof that the least upper bound $\bigsqcup_{i \in I} f_i$ for a chain f_i of currents in a web Γ is a current relies on Beppo Levi's monotone convergence theorem. The challenge here was that the monotone convergence theorem applies only to countable increasing sequences, whereas Isabelle's formalizaton of chain-complete partial orders demands the existence of least upper bounds for arbitrary (uncountable) chains. We bridge the gap by finding a countable subsequence of any such chain, which relies on the currents being non-zero only on the countably many edges.

Third, we often faced the problem that a statement had some precondition that was not met when we wanted to apply it. In an informal proof, these preconditions would be assumed "without loss of generality" or ignored altogether. We deal with them in two ways: either introduce a reduction that ensures the precondition or generalize the definitions and proofs so that they are not needed. Reductions are in general preferable as generalizations often complicate the definitions and proofs. Additional reductions can be seen, e.g., in Lemma 13. It assumes that there is no direct edge from s to t and all edges have positive capacity. The final theorem 1 does not make these assumptions. We therefore introduce another reduction that splits a potential s-t edge by introducing a new vertex and removes all edges with no capacity. Similarly, the reduction to bipartite webs in Sect. 3.4 assumes that the web does not contain loops. These loops would originate from loops in the original network; so we have another reduction that eliminates loops in networks. Reductions are not always feasible though. The example of the quotient web (Def. 16) is discussed in the next section.

Table 1 Line counts for different parts of the formalization, not counting empty lines

	Shared		Bounded	Unbounded
preliminaries	200	matrix for marginals (Prop. 24)	845	
networks & webs	2214	flow attainability (Thm. 28)		1954
reductions	1248	bipartite linkability (Thms. 25 / 22)	589	3158
total	3662		1434	5112

On the positive side, reasoning about paths in networks and webs was much less of a pain than we had expected. We formalized a finite path as a list of vertices, which allows us to reuse Isabelle's library for lists to manipulate and reason about paths. For example, the predicate distinct expresses that a path does not contain cycles, and π @ $[x]$ @ π' denotes the concatenation of the two paths π @ $[x]$ and $[x]$ @ π'. Moreover, we found that $\mathcal{E}$, RF, and RF° are powerful concepts that allow us to avoid explicitly dealing with paths in the main lemmas about flows—once we had proven enough properties about them.

Table 1 shows line counts of the Isabelle theories for different parts of the formalization, as a proxy for the formalization effort. These counts exclude empty lines. The left part lists the material that is used by both linkability proofs for bipartite webs. This covers the concepts of networks, flows, webs, currents, (maximal) waves, and trimmings, as well as the reductions from networks to webs and from webs to bipartite webs. On the right, the line counts are shown for linkability of bounded (Sect. 4.1) and unbounded (Sect. 4.2) countable bipartite webs, together with the line counts for the helper statements 24 and 28. The unbounded case requires about 3.6 times as much space as the bounded case if we include the formalization of the helper statements. If we exclude the helper statements, the ratio is about 5.4. This highlights how much more complicated the general case is.

We have also generated a PDF from the Isabelle theories using Isabelle's document preparation system. The material corresponding to shared and unbounded fill 236 pages. Aharoni et al. need a bit more than 10 pages in [3]. This gives an expansion factor of about 23. This is much higher than for text book mathematics, where the factor is typically well below 10 [6, 24]. We take this as an indication that the original paper is very dense.

6 Problems in the Original Proof

We now discuss the problems we have identified in the original paper during the formalization. We focus on three representative examples here: the reduction to bipartite webs, the definition of quotient webs, and the notion of trimmings. Further problems are given in the report [18].

Reduction to bipartite webs This is the main problem we have found. Aharoni et al. [3] claim that the reduction to bipartite webs from Sect. 3.4 preserves looseness, but this is not the case. In Fig. 10, the web Γ on the left is loose, its bipartite transformation $\text{bp}(\Gamma)$ on the right is not loose, because it contains the non-zero wave shown. The problem is that there is no path from the (infinitely many) vertices y_i (where $i \in \mathbb{N}$) to b. In a finite web, we could remove all vertices that cannot reach a vertex in B, because they cannot contribute to a web-flow. In the infinite case, however, we cannot do so easily because such infinite paths do occur in infinite networks and absorb parts of the (maximal) flow; an example is given in the conclusion. So their key theorem [3, Thm. 6.5], namely that every countable loose bipartite web contains a linkage, cannot be used to prove the general case.

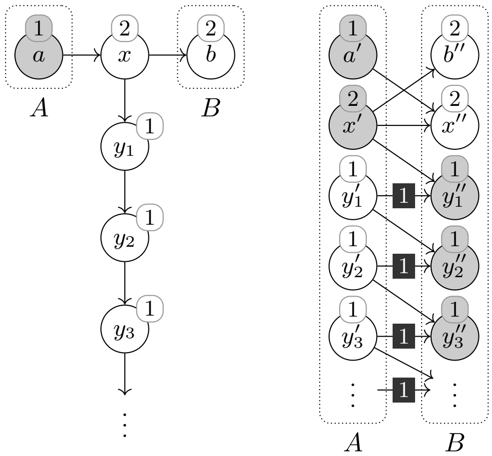

Figure 10 A loose web (left) whose bipartite reduction (right) is not loose as witnessed by the non-zero wave shown.

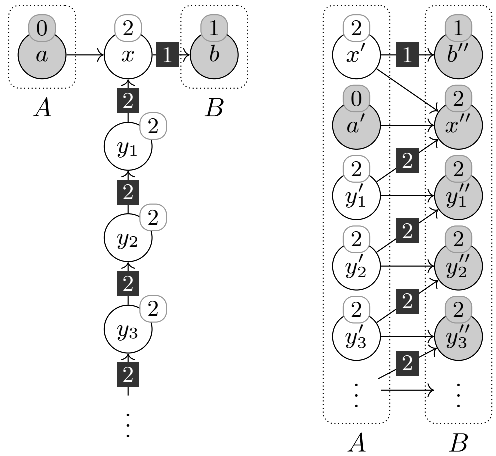

Figure 11 An unhindered web (left) whose bipartite reduction (right) contains a hindrance as witnessed at x'.

Instead, we strengthen the theorem to countable *unhindered* bipartite webs (Thm. 22). The induction invariant now is $\Omega \ominus f_n$ being unhindered rather than being loose, and the induction step (Lemma 27) must also be generalized. Fortunately, the original high-level ideas carry over; our proof composes the lemmas 29, 30 and 31 in a different order. We regain looseness from unhinderedness by first finding a maximal wave and reducing the weights, similar to what is happening in Lemma 19. Note that the reduction bp does not preserve unhinderedness either, as the example in Fig. 11 shows. The web on the left is not loose as it contains the shown wave.

Quotient webs Quotient webs (Def. 16) are an example where the definition had to be changed. This change propagates to the proofs of the basic properties of quotient webs. In detail, the original definition sets the edges as $E_{\Gamma/f} = \{(x,y) \in E \mid x \notin \mathrm{RF}^{\circ}_{\Gamma}(f) \wedge y \notin \mathrm{RF}^{\circ}_{\Gamma}(f)\}$, i.e., an edge may point to one of f's essential terminal vertices. Our Definition 16 excludes these edges. The difference is illustrated in Fig. 12. The quotient Γ/f on the right of the web Γ and the wave f on the left contains the edge (z,x) only with the original definition. This edge invalidates a number of statements, e.g., that $f + g \restriction (\Gamma/f)$ is a current or a wave if g is a current or a wave in Γ, where $g \restriction (\Gamma/f)$ restricts g to the vertices of Γ/f. Take, e.g., $g(a,z) = 2$, $g(z,x) = g(z,y) = 1$, and $g(e) = 0$ otherwise.

Our definition therefore excludes this edge. And while we were at it, we also changed the definition of $A_{\Gamma/f}$ and the weights so that the two sides of the quotient are always disjoint and vertices without edges have weight 0. These changes ensure that the quotient web meets the assumptions of the reduction to bipartite webs (Sect. 3.4). Accordingly, we had to adapt the existing proofs about the quotient web's properties or find new ones.

Trimmings The definition of trimmings (Def. 14) is an example of a small glitch that affects proofs only minimally. For trimmings, Aharoni et al. [3] require the stronger condition $\mathrm{TER}(g) - A = \mathcal{E}(\mathrm{TER}(f)) - A$ instead of $\mathcal{E}(\mathrm{TER}(g)) - A = \mathcal{E}(\mathrm{TER}(f)) - A$. The two are equivalent only if there are no vertices with weight 0, but webs may contain such vertices. So Lemma 15 need not hold for such webs. For example, Fig. 13 shows a wave f that does not have a trimming according to Aharoni et al.'s definition [3, Def. 4.7]. Every wave g has $x \in \mathrm{TER}(g)$ because x has weight 0, but $x \notin \mathcal{E}(\mathrm{TER}(f)) - A = \{y\}$.

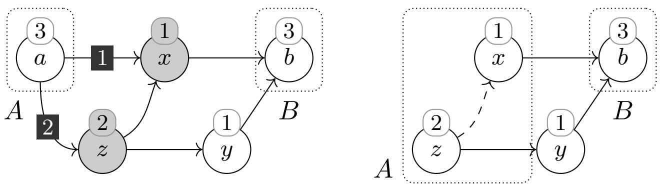

Figure 12 A wave f in a web Γ (left) and the quotient web Γ/f (right). The quotient contains the edge (z, x) only in [3].

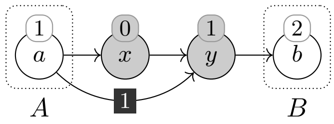

Figure 13 Wave f in a web none of whose trimmings g satisfies Aharoni et al.'s condition $\mathrm{TER}(g) - A = \mathcal{E}(\mathrm{TER}(f)) - A$.

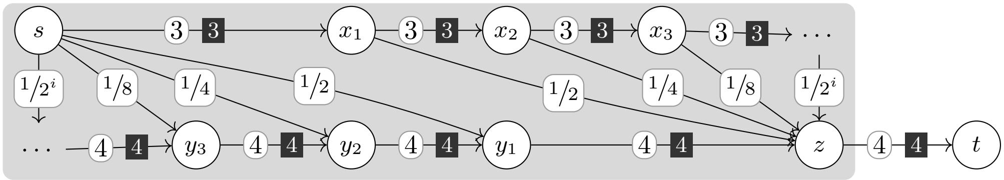

Figure 14 An infinite network with an orthogonal pair of a cut and a flow.

7 Related work

Lee [15] and Lammich and Sefidgar [13, 14] have formalized the MFMC theorem for *finite* networks in Mizar and Isabelle/HOL, respectively. Lammich and Sefidgar additionally formalize and verify several max-flow algorithms. We reused Lammich and Sefidgar's formalization in our proof of Prop. 24. We make no algorithmic considerations, as countable networks are infinite objects that lie beyond the reach of traditional notions of algorithms.

Lyons and Peres [19, Thm. 3.1] consider countable locally finite networks, where every vertex has only finitely many neighbours, and without a sink. They show that the maximum flow's value equals the value of a minimum cut, where a cut here contains an edge of every infinite simple path that starts at the source. Like our proof for the bounded case, their proof extends the MFMC theorem for finite networks using majorised convergence. Since their graphs are locally finite, all summations of interest are finite by construction.

8 Conclusion

In this paper, we have formalized a strong max-flow min-cut theorem for countable networks in Isabelle/HOL. To rule out anomalities due to the network being infinite, the theorem statement avoids imprecise infinite sums and instead compares the saturation edge by edge. During the formalization, we have discovered and fixed a number of problems in the original proof [3].

Arguably, this statement still does not capture the intuition fully. For example, the infinite network in Fig. 14 has a cut of value 4 with an orthogonal flow. This is the cut that the proof of Thm. 1 constructs. Yet, this cut is not minimal: The cut that separates the upper nodes from the lower nodes would be saturated by a flow of 2 units (not shown). This illustrates the intricacies of infinite networks: The out-flow from the source s of value 3 drains away in the infinite ray $s \to x_1 \to x_2 \to x_3 \to \dots$. Conversely, the in-flow to the sink t of value 4 is pulled in via the infinite path $\dots \to y_3 \to y_2 \to y_1 \to z \to t$. So this network shows that the outflow from the source may exceed the capacity of a cut and yet not saturate it.

Aharoni et al. [3, Sects. 7–8] study two restrictions on networks that avoid such anomalies: networks without infinite edge-disjoint paths and locally-finite networks. We have not yet formalized these results. Neither result applies to the network in Fig. 14. So finding a more intuitive statement of the max-flow min-cut theorem for countable networks is still an open problem.

References

1. Ron Aharoni. Menger's theorem for graphs containing no infinite paths. *European Journal of Combinatorics*, 4:201–204, 1983. doi:10.1016/S0195-6698(83)80012-2.
2. Ron Aharoni and Eli Berger. Menger's theorem for infinite graphs. *Inventiones mathematicae*, 176(1):1–62, 2009. doi:10.1007/s00222-008-0157-3.
3. Ron Aharoni, Eli Berger, Agelos Georgakopoulos, Amitai Perlstein, and Philipp Sprüssel. The max-flow min-cut theorem for countable networks. *Journal of Combinatorial Theory, Series B*, 101:1–17, 2010. doi:10.1016/j.jctb.2010.08.002.
4. Clemens Ballarin. Locales and locale expressions in Isabelle/Isar. In Stefano Berardi, Mario Coppo, and Ferruccio Damiani, editors, *Types for Proofs and Programs (TYPES 2003)*, volume 3085 of *LNCS*, pages 34–50. Springer Berlin Heidelberg, 2004. doi:10.1007/978-3-540-24849-1_3.
5. Clemens Ballarin. Locales: A module system for mathematical theories. *Journal of Automated Reasoning*, 52:123–153, 2014. doi:10.1007/s10817-013-9284-7.
6. Clemens Ballarin. Exploring the structure of an algebra text with locales. *Journal of Automated Reasoning*, 64:1093–1121, 2020. doi:10.1007/s10817-019-09537-9.
7. Gilles Barthe, Thomas Espitau, Justin Hsu, Tetsuya Sato, and Pierre-Yves Strub. *-liftings for differential privacy. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 102:1–102:12, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ICALP.2017.102.
8. N. Bourbaki. Sur le théorème de Zorn. *Archiv der Mathematik*, 2(6):434–437, 1949.
9. Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248–264, 1972. doi:10.1145/321694.321699.
10. L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956. doi:10.4153/CJM-1956-045-5.
11. Fabian Immler. Generic construction of probability spaces for paths of stochastic processes in Isabelle/HOL. Master's thesis, Fakultät für Informatik, Technische Universität München, 2012.
12. Hans G. Kellerer. Funktionen auf Produkträumen mit vorgegebenen Marginal-Funktionen. *Mathematische Annalen*, 144:323–344, 1961. doi:10.1007/BF01470505.
13. Peter Lammich and S. Reza Sefidgar. Formalizing the Edmonds-Karp algorithm. In Jasmin Christian Blanchette and Stephan Merz, editors, *Interactive Theorem Proving (ITP 2016)*, volume 9807 of *LNCS*, pages 219–234. Springer, 2016. doi:10.1007/978-3-319-43144-4_14.
14. Peter Lammich and S. Reza Sefidgar. Formalizing network flow algorithms: A refinement approach in Isabelle/HOL. *Journal of Automated Reasoning*, 62:261–280, 2019. doi:10.1007/s10817-017-9442-4.
15. Gilbert Lee. Correctnesss of Ford-Fulkerson's maximum flow algorithm. *Formalized Mathematics*, 13(2):305–314, 2005. URL: https://fm.mizar.org/2005-13/pdf13-2/glib_005.pdf.
16. Andreas Lochbihler. A formal proof of the max-flow min-cut theorem for countable networks. *Archive of Formal Proofs*, 2016. http://www.isa-afp.org/entries/MFMC_Countable.shtml, Formal proof development.
17. Andreas Lochbihler. Probabilistic functions and cryptographic oracles in higher-order logic. In Peter Thiemann, editor, *Programming Languages and Systems (ESOP 2016)*, volume 9632 of *LNCS*, pages 503–531. Springer, 2016. doi:10.1007/978-3-662-49498-1_20.
18. Andreas Lochbihler. A mechanized proof of the max-flow min-cut theorem for countable networks. http://www.andreas-lochbihler.de/pub/lochbihler2021itpl.pdf, 2021.
19. Russell Lyons and Yuval Peres. *Probability on Trees and Networks*. Cambridge University Press, New York, 2017. doi:10.1017/9781316672815.

20 Wolfgang Naraschewski and Markus Wenzel. Object-oriented verification based on record subtyping in higher-order logic. In Jim Grundy and Malcolm Newey, editors, *Theorem Proving in Higher Order Logics (TPHOLs 1998)*, volume 1479 of *LNCS*, pages 349–366. Springer, 1998. doi:10.1007/BFb0055146.

21 Christophe Sabot and Laurent Tournier. Random walks in Dirichlet environment: an overview. *Annales de la Faculté des sciences de Toulouse: Mathématiques*, Ser. 6, 26(2):463–509, 2017. doi:10.5802/afst.1542.

22 Joshua Sack and Lijun Zhang. A general framework for probabilistic characterizing formulae. In Viktor Kuncak and Andrey Rybalchenko, editors, *Verification, Model Checking, and Abstract Interpretation (VMCAI 2012)*, volume 7148 of *LNCS*, pages 396–411. Springer, 2012. doi:10.1007/978-3-642-27940-9_26.

23 Gert Smolka, Steven Schäfer, and Christian Doczkal. Transfinite constructions in classical type theory. In Christian Urban and Xingyuan Zhang, editors, *Interactive Theorem Proving (ITP 2015)*, volume 9236 of *LNCS*, pages 391–404. Springer, 2015. doi:10.1007/978-3-319-22102-1_26.

24 Freek Wiedijk. The de Bruijn factor. https://www.cs.ru.nl/~freek/factor/factor.pdf, 2000.