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

Andreas Lochbihler

Digital Asset
Motivation

CryptHOL

Relational logic for discrete probability distributions
Motivation

CryptHOL

CryptHOL: Game-based Proofs in Higher-order Logic

David A. Basin, Andreas Lochbihler, and S. Reza Seifgar
Institute of Information Security, Department of Computer Science, ETH Zurich, Zurich, Switzerland

Abstract. Game-based proofs are a well-established paradigm for structuring security arguments and simplifying their understanding. We present a novel framework, CryptHOL, for rigorous game-based proofs that is

Relational logic for discrete probability distributions

A general framework for probabilistic characterizing formulae

Joshua Sack1 and Lijun Zhang2
1 Department of Mathematics and Statistics, California State University Long Beach
2 DTU Informatics, Technical University of Denmark

Abstract. Recently, a general framework on characteristic formulae was proposed by Axen et al. It offers a simple theory that allows one to easily obtain characteristic formulae of many non-probabilistic behavioral relations. Our paper studies their techniques in a probabilistic setting. We provide a general method
Motivation

CryptHOL

CryptHOL: Game-based Proofs in Higher-order Logic
David A. Basin, Andreas Lochbihler, and S. Reza Sefidgar
Institute of Information Security, Department of Computer Science, ETH Zürich, Zürich, Switzerland

Abstract. Game-based proofs are a well-established paradigm for structuring security arguments and simplifying their understanding. We present a novel framework, CryptHOL, for rigorous game-based proofs that is

Lifting operator

Relational logic for discrete probability distributions

The Max-Flow Min-Cut theorem for countable networks
Ron Aharoni\textsuperscript{a,1,3}, Eli Berger\textsuperscript{b,2}, Agelos Georgakopoulos\textsuperscript{c,3}, Amitai Perlstein\textsuperscript{a}, Philipp Sprüssel\textsuperscript{c,3}

\textsuperscript{a} Department of Mathematics, Technion, Haifa, Israel 32000
\textsuperscript{b} Department of Mathematics, Faculty of Science and Science Education, Haifa University, Israel 32000
\textsuperscript{c} Universität Hamburg, Germany

\textbf{A R T I C L E I N F O}

Article history: Received 4 December 2007

Keywords: 

\textbf{A B S T R A C T}

We prove a strong version of the Max-Flow Min-Cut theorem for countable networks, namely that in every such network there exist a flow and a cut that are “orthogonal” to each other, in the sense that the flow saturates the cut and is zero on the reverse cut. If
Motivation

CryptHOL

A recent mathematical theorem
✓ formalized in Isabelle/HOL
✓ found and fixed many mistakes and glitches
✓ simpler variant with additional assumptions

Relational logic for discrete probability distributions

The Max-Flow Min-Cut theorem for countable networks

Ron Aharoni$^{a,1,3}$, Eli Berger$^{b,2}$, Agelos Georgakopoulos$^{c,3}$, Amitai Perlstein$^a$, Philipp Sprüßel$^{c,3}$

$^a$ Department of Mathematics, Technion, Haifa, Israel 32000
$^b$ Department of Mathematics, Faculty of Science and Science Education, Haifa University, Israel 32000
$^c$ Universität Hamburg, Germany

ARTICLE INFO

Article history:
Received 4 December 2007

Keywords:

ABSTRACT

We prove a strong version of the Max-Flow Min-Cut theorem for countable networks, namely that in every such network there exist a flow and a cut that are “orthogonal” to each other, in the sense that the flow saturates the cut and is zero on the reverse cut. If
The Max-Flow-Min-Cut Theorem for Finite Networks

Network
• graph $G = (V, E)$
The Max-Flow-Min-Cut Theorem for Finite Networks

Network
- graph $G = (V, E)$
- capacity $c : E \rightarrow \mathbb{R}_{\geq 0}$
The Max-Flow-Min-Cut Theorem for Finite Networks

Network
- graph $G = (V, E)$
- capacity $c : E \rightarrow \mathbb{R}_{\geq 0}$
- source $s$, sink $t$
The Max-Flow-Min-Cut Theorem for Finite Networks

Network
- graph $G = (V, E)$
- capacity $c : E \rightarrow \mathbb{R}_{\geq 0}$
- source $s$, sink $t$

Flow $f : E \rightarrow \mathbb{R}_{\geq 0}$
- Capacity $f(e) \leq c(e)$
- Preservation $\sum_{e \in \text{in}(x)} f(e) = \sum_{e \in \text{out}(x)} f(e)$ for $x \in V \setminus \{s, t\}$
The Max-Flow-Min-Cut Theorem for Finite Networks

Network
- graph $G = (V, E)$
- capacity $c : E \to \mathbb{R}_{\geq 0}$
- source $s$, sink $t$

Flow $f : E \to \mathbb{R}_{\geq 0}$
- Capacity $f(e) \leq c(e)$
- Preservation $\sum_{e \in \text{in}(x)} f(e) = \sum_{e \in \text{out}(x)} f(e)$ for $x \in V - \{s, t\}$

Lammich and Sefidgar [ITP 2016]
The Max-Flow-Min-Cut Theorem for Finite Networks

Network
- graph $G = (V, E)$
- capacity $c : E \rightarrow \mathbb{R}_{\geq 0}$
- source $s$, sink $t$

Flow $f : E \rightarrow \mathbb{R}_{\geq 0}$
- Capacity $f(e) \leq c(e)$
- Preservation $\sum_{e \in \text{in}(x)} f(e) = \sum_{e \in \text{out}(x)} f(e)$
  for $x \in V - \{s, t\}$
- Value $|f| = \sum_{e \in \text{out}(s)} f(e) = \sum_{e \in \text{in}(t)} f(e)$
  $|f| = 6 + 3 = 8 + 1$
The Max-Flow-Min-Cut Theorem for Finite Networks

Cut $C \subseteq V$
- $s \in C$, $t \notin C$
- Value $|C| = \sum_{(x,y) \in E, x \in C, y \notin C} c(x,y)$

Network
- graph $G = (V, E)$
- capacity $c : E \to \mathbb{R}_{\geq 0}$
- source $s$, sink $t$

Flow $f : E \to \mathbb{R}_{\geq 0}$
- Capacity $f(e) \leq c(e)$
- Preservation $\sum_{e \in \text{in}(x)} f(e) = \sum_{e \in \text{out}(x)} f(e)$ for $x \in V - \{s, t\}$
- Value $|f| = \sum_{e \in \text{out}(s)} f(e) = \sum_{e \in \text{in}(t)} f(e)$

$|f| = 6 + 3 = 8 + 1$
The Max-Flow-Min-Cut Theorem for Finite Networks

**Flow** \( f : E \rightarrow \mathbb{R}_{\geq 0} \)
- Capacity \( f(e) \leq c(e) \)
- Preservation \( \sum_{e \in \text{in}(x)} f(e) = \sum_{e \in \text{out}(x)} f(e) \) for \( x \in V - \{s, t\} \)

**Value** \( |f| = \sum_{e \in \text{out}(s)} f(e) = \sum_{e \in \text{in}(t)} f(e) \)
- \( |f| = 6 + 3 = 8 + 1 \)

**Network**
- graph \( G = (V, E) \)
- capacity \( c : E \rightarrow \mathbb{R}_{\geq 0} \)
- source \( s \), sink \( t \)

**Cut** \( C \subseteq V \)
- \( s \in C, t \notin C \)
- Value \( |C| = \sum_{(x,y) \in E, x \in C, y \notin C} c(x,y) \)
- \( |C| = 3 + 5 + 1 \)
The Max-Flow-Min-Cut Theorem for Finite Networks

Max-Flow Min-Cut Theorem
In every finite network, there exist a cut \( C \) and a flow \( f \) s.t. \( |C| = |f| \).

Lammich and Sefidgar [ITP 2016]

Network
- graph \( G = (V, E) \)
- capacity \( c : E \rightarrow \mathbb{R}_{\geq 0} \)
- source \( s \), sink \( t \)

Cut \( C \subseteq V \)
- \( s \in C \), \( t \notin C \)
- Value \( |C| = \sum_{(x,y) \in E, x \in C, y \notin C} c(x,y) \)  
  \( |C| = 3 + 5 + 1 \)

Flow \( f : E \rightarrow \mathbb{R}_{\geq 0} \)
- Capacity \( f(e) \leq c(e) \)
- Preservation \( \sum_{e \in \text{in}(x)} f(e) = \sum_{e \in \text{out}(x)} f(e) \)  
  for \( x \in V - \{s, t\} \)
- Value \( |f| = \sum_{e \in \text{out}(s)} f(e) = \sum_{e \in \text{in}(t)} f(e) \)  
  \( |f| = 6 + 3 = 8 + 1 \)
Challenges with Countable Networks

\[ C = \{ s, x_1, x_2, \ldots \} \quad |C| = \infty \]

\[ f(e) = 1 \quad |f| = \infty \]
Challenges with Countable Networks

\[ \begin{align*} C &= \{ s, x_1, x_2, \ldots \} \quad |C| = \infty \\ f(e) &= 1 \quad |f| = \infty \\ g(e) &= \frac{1}{2} \quad |g| = \infty \end{align*} \]
Challenges with Countable Networks

\[
C = \{s, x_1, x_2, \ldots\} \quad |C| = \infty
\]

\[
f(e) = 1 \quad |f| = \infty
\]

\[
g(e) = \frac{1}{2} \quad |g| = \infty
\]

Avoid infinite sums!
Challenges with Countable Networks

\[ C = \{ s, x_1, x_2, \ldots \} \quad |C| = \infty \]

\[ f(e) = 1 \quad |f| = \infty \]

\[ g(e) = \frac{1}{2} \quad |g| \neq \infty \]

Avoid infinite sums!

**Max-Flow Min-Cut Theorem** [Aharoni et al.]

There exist a cut \( C \) and a flow \( f \) s.t.

- \( f(x, y) = c(x, y) \) for \((x, y) \in E, x \in C, y \notin C\)
- \( f(x, y) = 0 \) for \((x, y) \in E, x \notin C, y \in C\)
More Infinite Sums
More Infinite Sums

Flow preservation
More Infinite Sums

Flow preservation

\[ \sum\left(\frac{1}{n^2}\right) = \frac{\pi^2}{6} \]
More Infinite Sums

Flow preservation

Web: Bound vertex throughput

dualize
More Infinite Sums

Flow preservation

Web: Bound vertex throughput

dualize

3 + 3
5 + 1
3 + 3
5 + 1
More Infinite Sums

Flow preservation

Web: Bound vertex throughput

dualize

3 + 3

3 + 3

5 + 1

5 + 1

3

5

4

5

4

5

3 + 2

3 + 2 ≤ 5

0

2

1

1

1

7

1

7

1

1
Transformations

1. Adapt proof to weakened induction invariant
2. New proof using finite MFMC theorem if total neighbour weight is finite
Transformations

1. Adapt proof to weakened induction invariant
2. New proof using finite MFMC theorem if total neighbour weight is finite

Dualize
Transformations

1. adapt proof to weakened induction invariant
2. new proof using finite MFMC theorem if total neighbour weight is finite
Transformations

1. Adapt proof to weakened induction invariant
2. New proof using finite MFMC theorem if total neighbour weight is finite

find max. wave

dualize
Transformations

1. adapt proof to weakened induction invariant
2. new proof using finite MFMC theorem if total neighbour weight is finite

find max. wave backpressure

dualize
Transformations

1. Dualize
2. Find max. wave
3. Backpressure

A

B

dualize

focus

backpressure

not preserved

make

bipartite

induction invariant

weakened

adapt proof to

find max.

wave

make

bipartite
Transformations

1. Adapt proof to weakened induction invariant
2. New proof using finite MFMC theorem if total neighbour weight is finite

Dualize

Find max. wave

Focus

Backpressure
Transformations

1. Dualize
2. Make bipartite

- Find max. wave
- Focus
- Backpressure

- Adapt proof to weakened induction invariant
- New proof using finite MFMC theorem if total neighbour weight is finite

- Transformations
- Dualize
- Find max. wave
- Focus
- Backpressure
Transformations

1. Dualize

2. Find max. wave

3. Focus

4. Backpressure
Transformations

1. Dualize
2. Find max. wave
3. Backpressure

Duality invariant

Find max. wave

Backpressure

Dualize

Find max. wave

Backpressure

Dualize

Find max. wave

Backpressure

Dualize
Transformations

1. Adapt proof to weakened induction invariant
2. New proof using finite MFMC theorem if total neighbour weight is finite

- Dualize
- Find max. wave
- Backpressure
- Focus
- Make bipartite
Transformations

1. adapt proof to weakened induction invariant
2. new proof using finite MFMC theorem if total neighbour weight is finite

looseness not preserved

find max. wave

backpressure
Transformations

1. adapt proof to weakened induction invariant
2. new proof using finite MFMC theorem if total neighbour weight is finite

looseness not preserved
Backpressure fixpoint

\[ \text{Pick a leaking vertex and reduce incoming flow.} \]

\[ f = \text{fix}(bp_G) \]

\[ \text{Flow} \geq 0, \geq 0 \]

\( \text{Knaster-Tarski?} \)

\( \text{Bourbaki-Witt!} \)

\( \lambda \to \forall = \text{Isabelle} \beta \alpha \text{HOL} \)
Backpressure fixpoint

Backpressure $bp_G$: Flow $\Rightarrow$ Flow

Pick a leaking vertex and reduce incoming flow.

If $f = \text{fix}(bp_G)$, Flow = ($E \Rightarrow R \geq 0$, $\geq 0$) is a ccpo

Knaster-Tarski? $bp_G$ is not monotone
Bourbaki-Witt! $bp_G$ is decreasing!
Backpressure fixpoint

Pick a leaking vertex and reduce incoming flow.

If any $f = \text{fix}(bp_G)$, then $\text{Flow} = (E \Rightarrow R \geq 0, \geq 0)$ is a ccpo.

Knaster-Tarski? $bp_G$ is not monotone!

Bourbaki-Witt! $bp_G$ is decreasing!
Backpressure fixpoint

Pick a leaking vertex and reduce incoming flow.

if any

\[ f = \text{fix}(\text{bp}_G) \]

Flow is a ccpo

Knaster-Tarski?

bp$_G$ is not monotone

Bourbaki-Witt!

bp$_G$ is decreasing!

\[ \lambda \rightarrow \forall \]

Isabelle

\[ \beta \]

\[ \alpha \]

HOL

translate
Backpressure fixpoint

A

B

$\text{Backpressure fixpoint}$

$\text{bp}_G: \text{Flow} \Rightarrow \text{Flow}$

Pick a leaking vertex and reduce incoming flow.

if any $f = \text{fix}(\text{bp}_G)$

$\text{Flow} = (E \Rightarrow_R \geq 0, \geq)$ is a ccpo

Knaster-Tarski? $\text{bp}_G$ is not monotone

Bourbaki-Witt! $\text{bp}_G$ is decreasing!

$\lambda \rightarrow \forall = \text{Isabelle} \beta \alpha \text{HOL}$

translate
Backpressure fixpoint

\[ \lambda \rightarrow \forall \beta = \text{Isabelle} \]

\[ \alpha \]

\[ \text{HOL} \]
Backpressure fixpoint

Backpressure $bp_G : Flow \Rightarrow Flow$

Pick a leaking vertex if any and reduce incoming flow.

$f = \text{fix}(bp_G)$
Backpressure fixpoint

**Backpressure** $bp_G : Flow \Rightarrow Flow$

Pick a leaking vertex $\star$ if any and reduce incoming flow.

$$f = \text{fix}(bp_G)$$

$Flow = (E \Rightarrow \mathbb{R}_{\geq 0}, \geq) \text{ is a ccpo}$

Knaster-Tarski?
Backpressure fixpoint

**Backpressure** $bp_G : Flow \Rightarrow Flow$

Pick a leaking vertex $\dagger$ if any and reduce incoming flow.

$$f = \text{fix}(bp_G)$$

$Flow = (E \Rightarrow \mathbb{R}_{\geq 0}, \geq)$ is a ccpo

*Knaster-Tarski?* $bp_G$ is not monotone
Backpressure fixpoint

Backpressure \(bp_G : Flow \Rightarrow Flow\)

Pick a leaking vertex if any and reduce incoming flow.

\[ f = \text{fix}(bp_G) \]

\( Flow = (E \Rightarrow \mathbb{R}_{\geq 0}, \geq) \) is a ccpo

Knaster-Tarski? \(bp_G\) is not monotone

Bourbaki-Witt! \(bp_G\) is decreasing!

---

Transfinite Constructions in Classical Type Theory

Gert Smolka\textsuperscript{(o)}, Steven Schiöfer, and Christian Doczkal
Saarland University, Saarbrücken, Germany
\{smolka,sciofer,doczkal\}@ps.uni-saarland.de

Abstract. We study a transfinite construction we call tower construction in classical type theory. The construction is inductive and applies to partially ordered types. It yields the set of all points reachable from
Backpressure fixpoint

Backpressure $bp_G : Flow \Rightarrow Flow$

Pick a leaking vertex $\downarrow$ if any and reduce incoming flow.

$f = \text{fix}(bp_G)$

$Flow = (E \Rightarrow \mathbb{R}_{\geq 0}, \geq)$ is a ccpo

Knaster-Tarski? $bp_G$ is not monotone

Bourbaki-Witt! $bp_G$ is decreasing!

Transfinite Constructions in Classical Type Theory

Gert Smolka(30), Steven Schäfer, and Christian Doczkal
Saarland University, Saarbrücken, Germany
{smolka,chaftier,doczkal}@cs.uni-saarland.de

Abstract. We study a transfinite construction we call tower con-struction in classical type theory. The construction is inductive and applies to partially ordered types. It yields the set of all points reachable from
Summary: Avoid infinite sums!
Summary: Avoid infinite sums!

Available in the AFP
isa-afp.org/entries/MFMC_Countable.html
Summary: Avoid infinite sums!

Available in the AFP
isa-afp.org/entries/MFMC_Countable.html

<table>
<thead>
<tr>
<th>Line counts</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Preliminaries</td>
<td>200</td>
</tr>
<tr>
<td>Networks &amp; webs</td>
<td>2214</td>
</tr>
<tr>
<td>Transformations</td>
<td>1248</td>
</tr>
<tr>
<td>Bounded linkability</td>
<td>1434</td>
</tr>
<tr>
<td>Unbounded linkability</td>
<td>5112</td>
</tr>
<tr>
<td>Total</td>
<td>10208</td>
</tr>
</tbody>
</table>
Summary: Avoid infinite sums!

Available in the AFP
isa-afp.org/entries/MFMC_Countable.html

Line counts
Preliminaries 200
Networks & webs 2214
Transformations 1248
Bounded linkability 1434
Unbounded linkability 5112
Total 10208