

A Computationally Informed Hierarchical Theory of Network Coding Rate Regions

John MacLaren Walsh

Department of Electrical and Computer Engineering

Drexel University

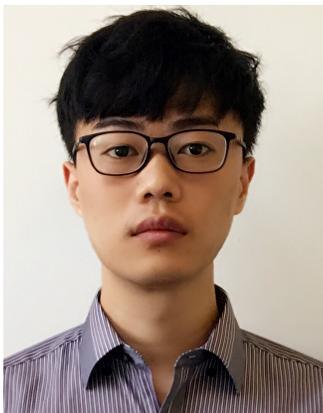
Philadelphia, PA

jwalsh@ece.drexel.edu



Thanks to NSF CCF-1421828 & NSF CCF-1053702.

Collaborators & Co-Authors



Yirui Liu

Ph.D. Candidate,
ASPITRG
Drexel University

*Common information
based network
combination operators*



Alejandro Erick Trofimoff

Ph.D. Candidate,
ASPITRG
Drexel University

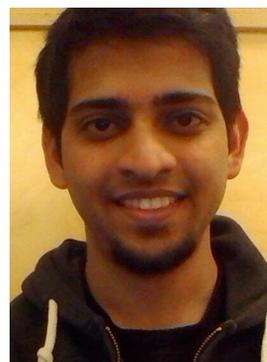
*ordering probabilistic
supports for mapping
nonlinear EVs*



Congduan Li, Ph.D.

Postdoctoral Fellow,
City Univ. of Hong Kong

*rate region database
network operators
forbidden net. minors*



Jayant Apte, Ph.D.

Data Scientist,
HVH Precision Analytics

*network symmetry
software: ITCP & ITAP*



Steven Weber, Ph.D.

Professor, Dept. of ECE
Drexel University

*co-advisor for
Congduan Li*



Yunshu Liu, Ph.D.

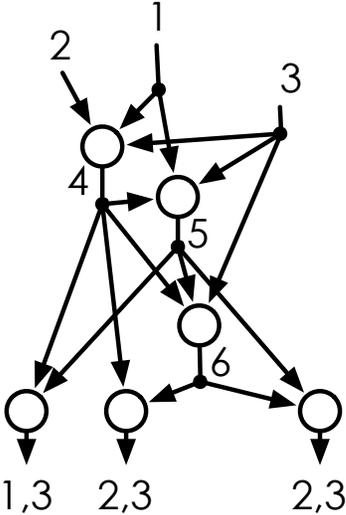
Senior Applied Researcher
eBay

*nonlinear entropic
vectors & codes*

What is a network coding problem?

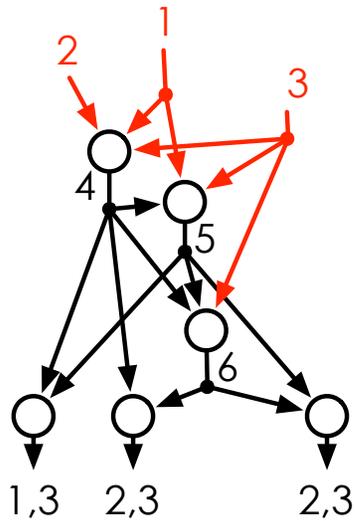
What is a Network Coding Problem?

network coding
problem



What is a network coding problem?

**network coding
problem**



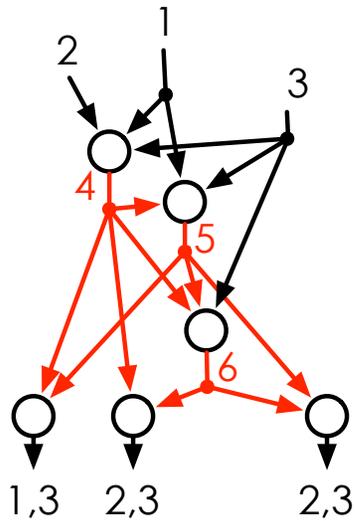
What is a Network Coding Problem?

A labelled directed acyclic hypergraph, including:

1. independent sources

What is a network coding problem?

**network coding
problem**



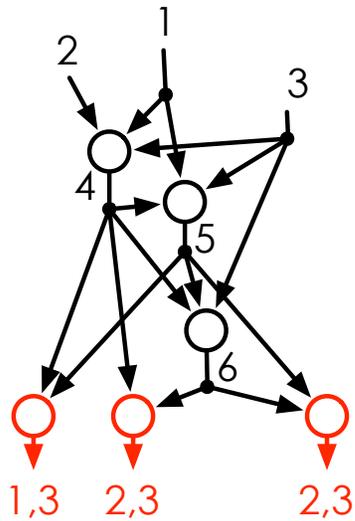
What is a Network Coding Problem?

A labelled directed acyclic hypergraph, including:

1. independent sources
2. messages: outgoing encoded from incoming

What is a network coding problem?

**network coding
problem**



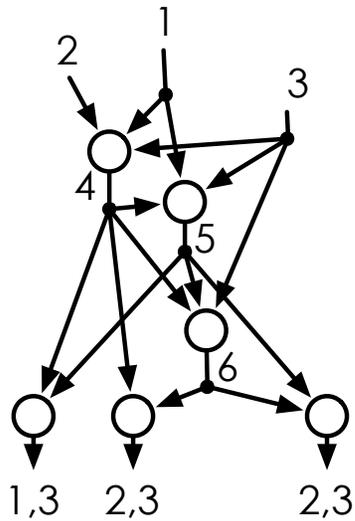
What is a Network Coding Problem?

A labelled directed acyclic hypergraph, including:

1. independent sources
2. messages: outgoing encoded from incoming
3. sinks: subsets of sources decoded from messages

What is a network coding problem?

**network coding
problem**



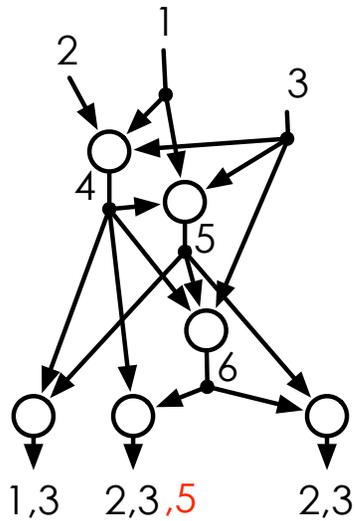
What is a Network Coding Problem?

A labelled directed acyclic hypergraph, including:

1. independent sources
2. messages: outgoing encoded from incoming
3. sinks: subsets of sources decoded from messages

What is a network coding problem?

network coding
problem



What is a Network Coding Problem?

A labelled directed acyclic hypergraph, including:

1. independent sources

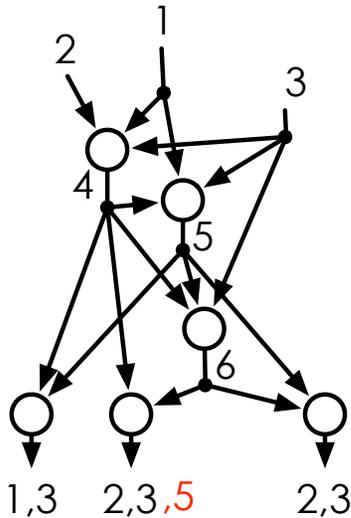
2. messages: outgoing encoded from incoming

and edges

3. sinks: subsets of sources  decoded from messages

What is a network coding problem?

**network coding
problem**



What is a Network Coding Problem?

A labelled directed acyclic hypergraph, including:

1. independent sources

2. messages: outgoing encoded from incoming

and edges

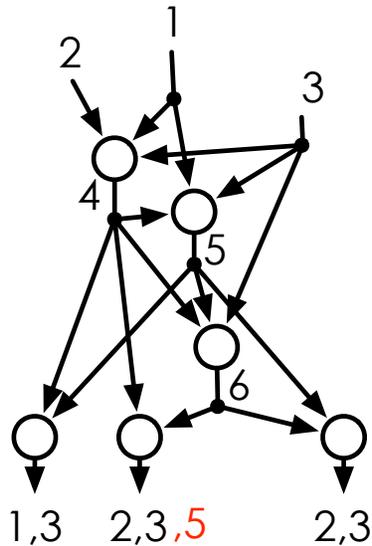
3. sinks: subsets of sources  decoded from messages

By drawing the right graph, this includes:

1. index coding
2. Distributed storage (exact & functional repair)
3. Coded Caching

What is a network coding problem?

**network coding
problem**



What is a Network Coding Problem?

A labelled directed acyclic hypergraph, including:

1. independent sources

2. messages: outgoing encoded from incoming

and edges

3. sinks: subsets of sources ▲ decoded from messages

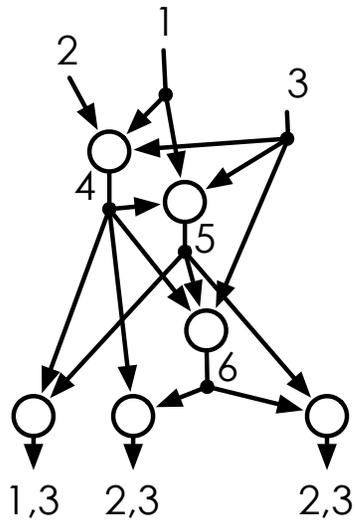
Also, core class of multiterminal information theory problems: embedded special cases

1. no noise. messages overheard perfectly
2. sources independent
3. sources reproduced perfectly

What is a network coding capacity region?

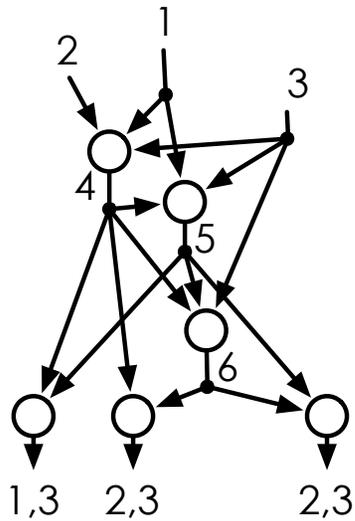
What is a Network Coding Capacity Region?

network coding
problem



What is a network coding capacity region?

network coding
problem

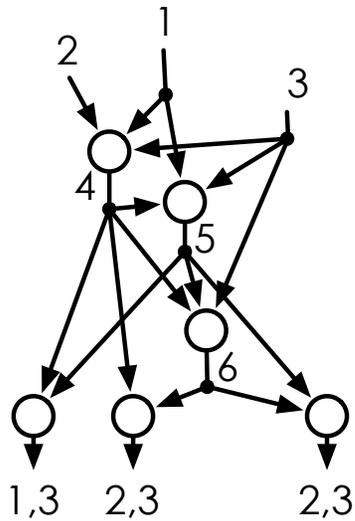


What is a Network Coding Capacity Region?

Source rates $H(Y_k)$, $k \in \{1, \dots, K\}$ and edge rates R_e , $e \in \{K + 1, \dots, N\}$ are *achievable* if:

What is a network coding capacity region?

**network coding
problem**



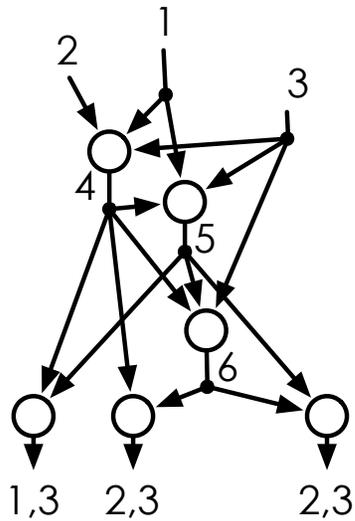
What is a Network Coding Capacity Region?

Source rates $H(Y_k)$, $k \in \{1, \dots, K\}$ and edge rates R_e , $e \in \{K + 1, \dots, N\}$ are *achievable* if:

\exists sequence of codes: edge encoders & sink decoders s.t.

What is a network coding capacity region?

**network coding
problem**



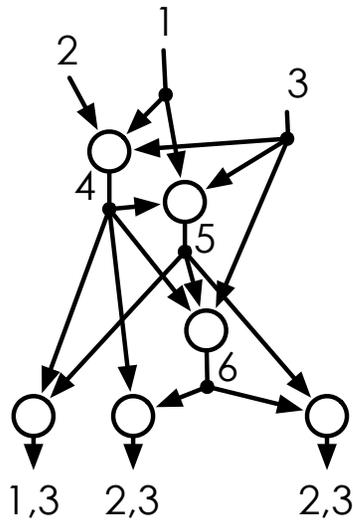
What is a Network Coding Capacity Region?

Source rates $H(Y_k)$, $k \in \{1, \dots, K\}$ and edge rates R_e , $e \in \{K + 1, \dots, N\}$ are *achievable* if:

\exists sequence of codes: edge encoders & sink decoders s.t.
source k has $nH(Y_k)$ bits,

What is a network coding capacity region?

network coding
problem



What is a Network Coding Capacity Region?

Source rates $H(Y_k)$, $k \in \{1, \dots, K\}$ and edge rates R_e , $e \in \{K + 1, \dots, N\}$ are *achievable* if:

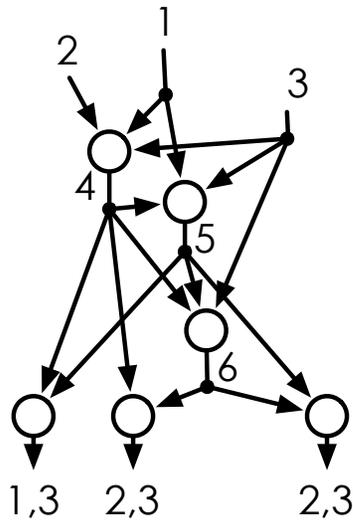
\exists sequence of codes: edge encoders & sink decoders s.t.

source k has $nH(Y_k)$ bits,

message on edge $e \in \{k + 1, \dots, N\}$, $U_e = f_e(\mathbf{X}_{\text{In}(e)})$,
 nR_e bits encoding incoming messages & source bits,

What is a network coding capacity region?

network coding
problem



What is a Network Coding Capacity Region?

Source rates $H(Y_k)$, $k \in \{1, \dots, K\}$ and edge rates R_e , $e \in \{K + 1, \dots, N\}$ are *achievable* if:

\exists sequence of codes: edge encoders & sink decoders s.t.

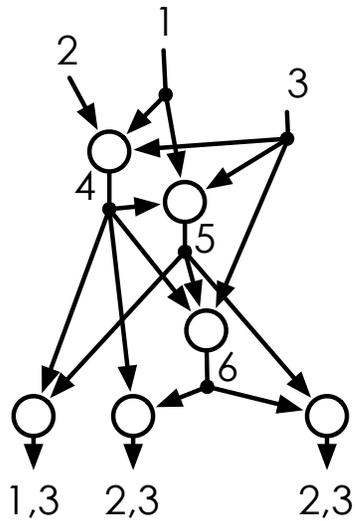
source k has $nH(Y_k)$ bits,

message on edge $e \in \{k + 1, \dots, N\}$, $U_e = f_e(\mathbf{X}_{\text{In}(e)})$,
 nR_e bits encoding incoming messages & source bits,

$\lim_{n \rightarrow \infty} \mathbb{P}[\text{decoding error}] = 0$.

What is a network coding capacity region?

network coding
problem



What is a Network Coding Capacity Region?

Source rates $H(Y_k)$, $k \in \{1, \dots, K\}$ and edge rates R_e , $e \in \{K + 1, \dots, N\}$ are *achievable* if:

\exists sequence of codes: edge encoders & sink decoders s.t.

source k has $nH(Y_k)$ bits,

message on edge $e \in \{k + 1, \dots, N\}$, $U_e = f_e(\mathbf{X}_{\text{In}(e)})$,
 nR_e bits encoding incoming messages & source bits,

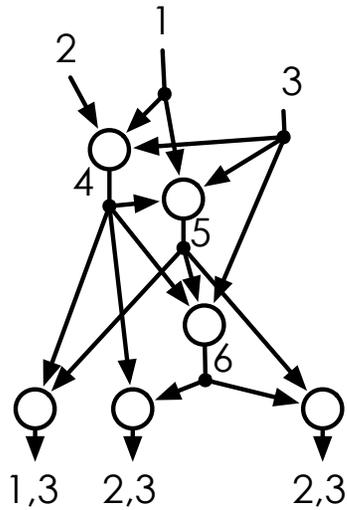
$\lim_{n \rightarrow \infty} \mathbb{P}[\text{decoding error}] = 0$.

Closure of set of all such achievable vectors

$\mathbf{r} = [H(Y_k), R_e | k \in \{1, \dots, K\}, e \in \{K + 1, \dots, N\}]$
 is *capacity region*, \mathcal{R}^* , a convex cone.

What is a network coding capacity region?

**network coding
problem**



inequality description

$$R_4 \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$

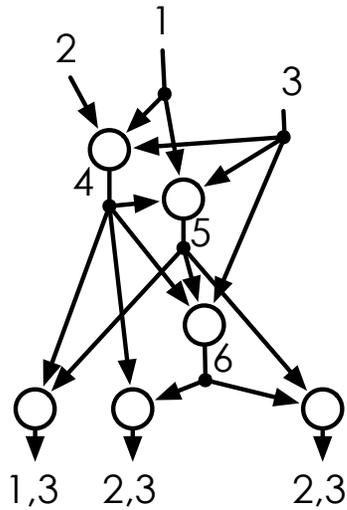
$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$

$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

rate region

What is a network coding capacity region?

**network coding
problem**



inequality description

$$R_4 \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$

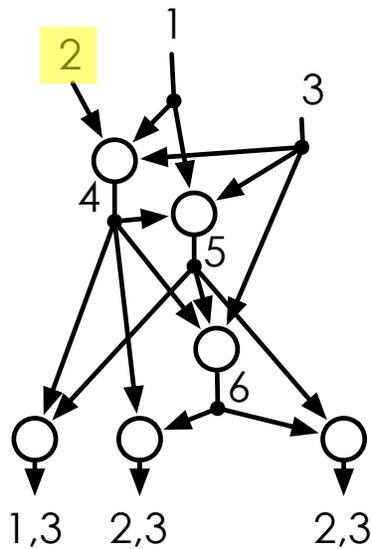
$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$

$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

rate region

What is a network coding capacity region?

network coding
problem



inequality description

$$R_4 \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$

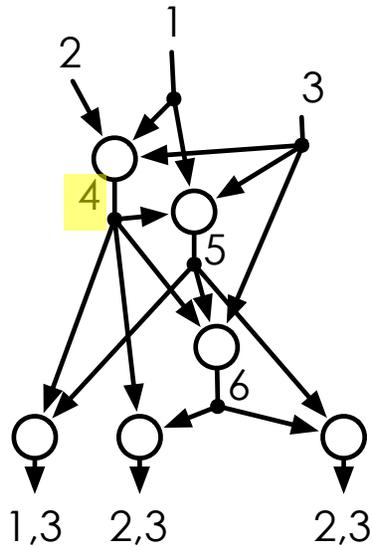
$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$

$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

rate region

What is a network coding capacity region?

**network coding
problem**



inequality description

$$R_4 \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

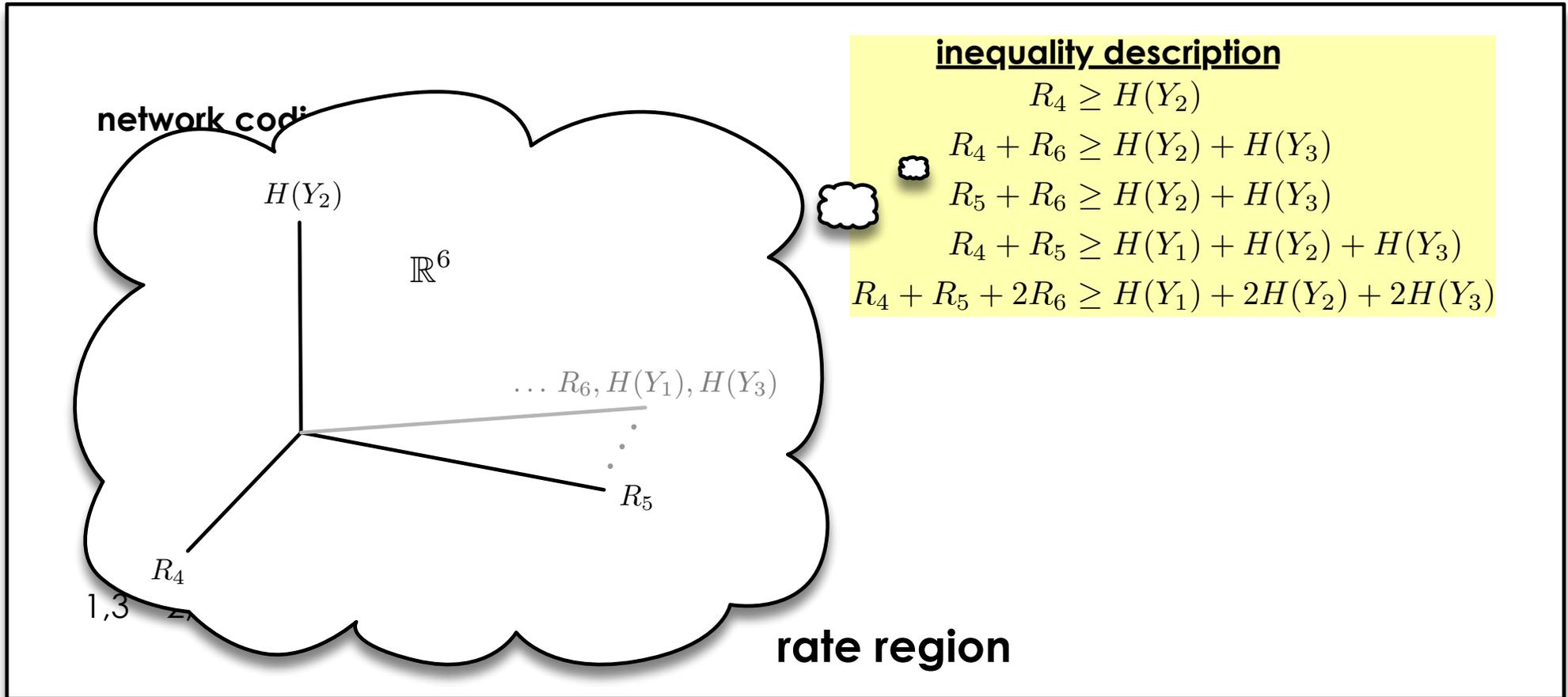
$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$

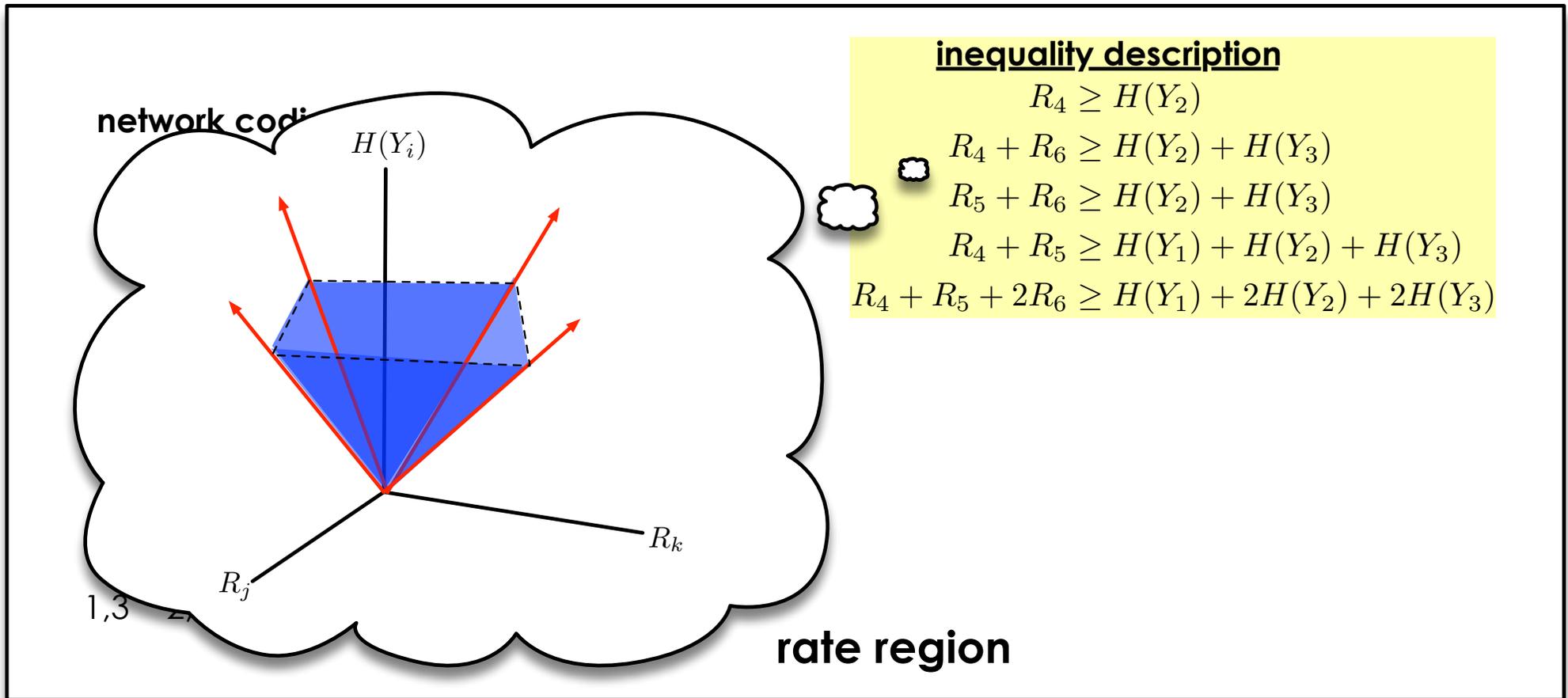
$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

rate region

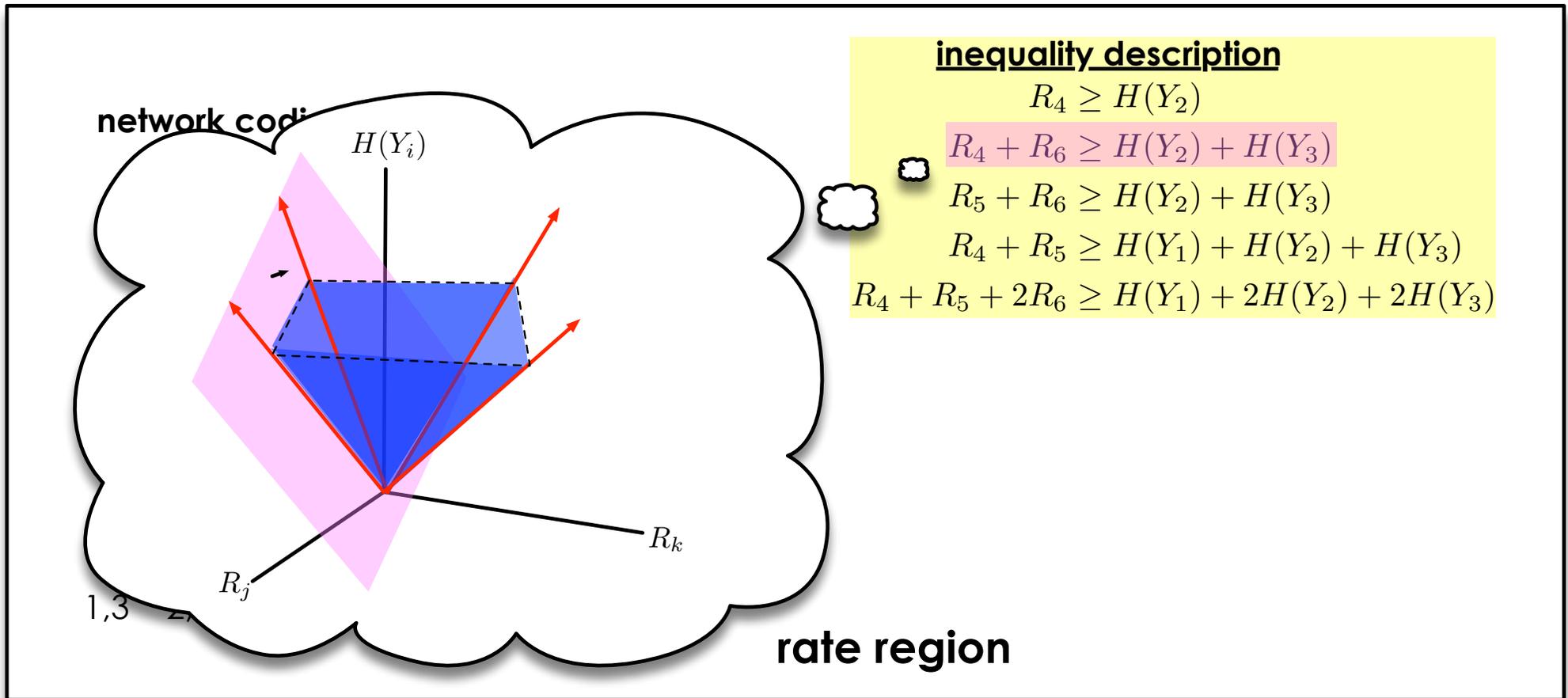
What is a network coding capacity region?



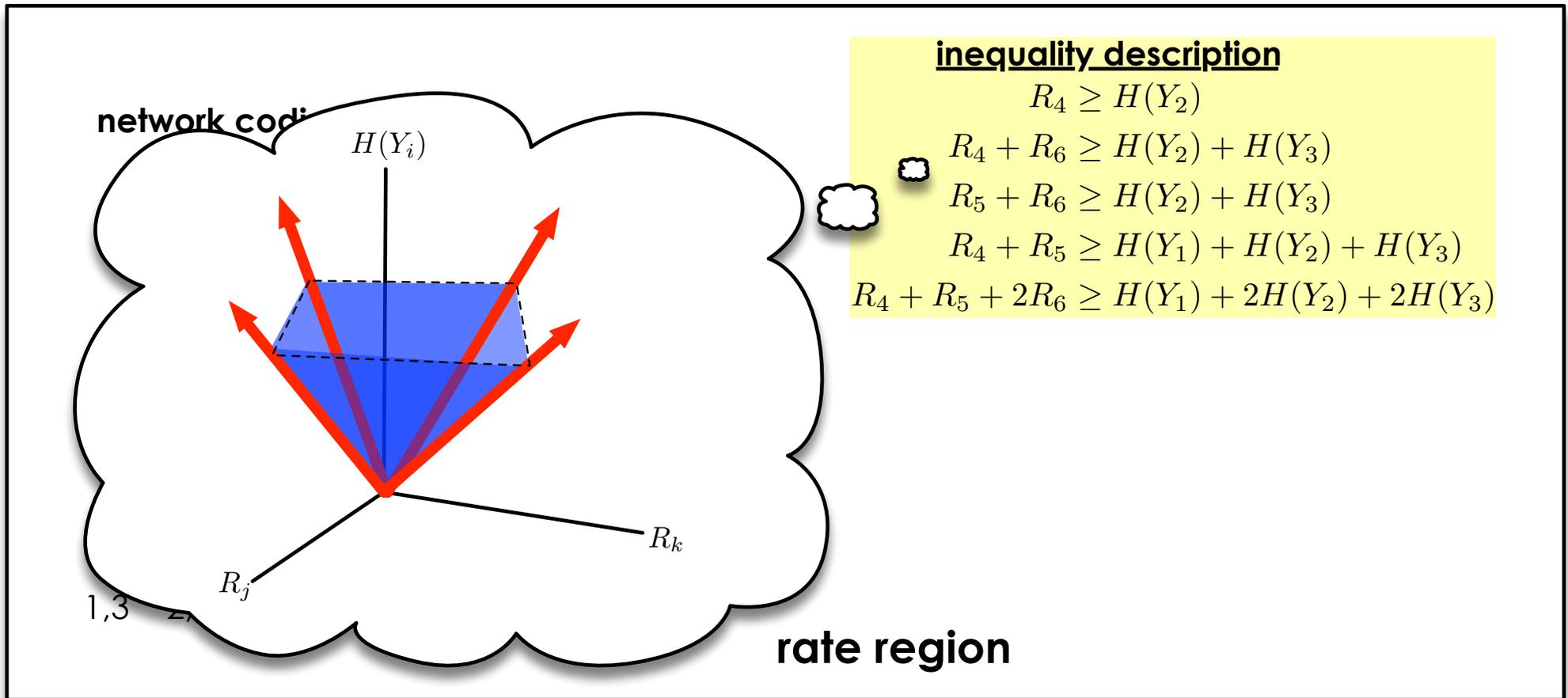
What is a network coding capacity region?



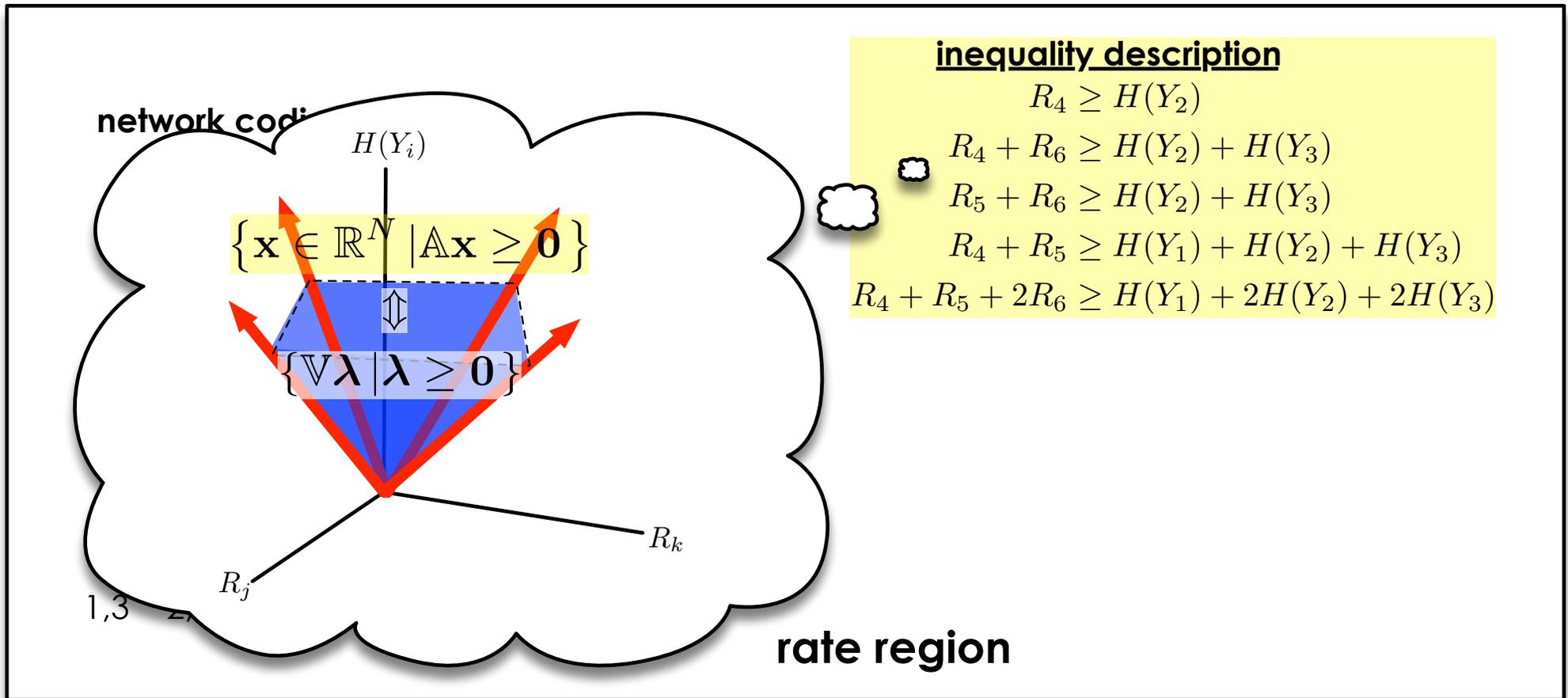
What is a network coding capacity region?



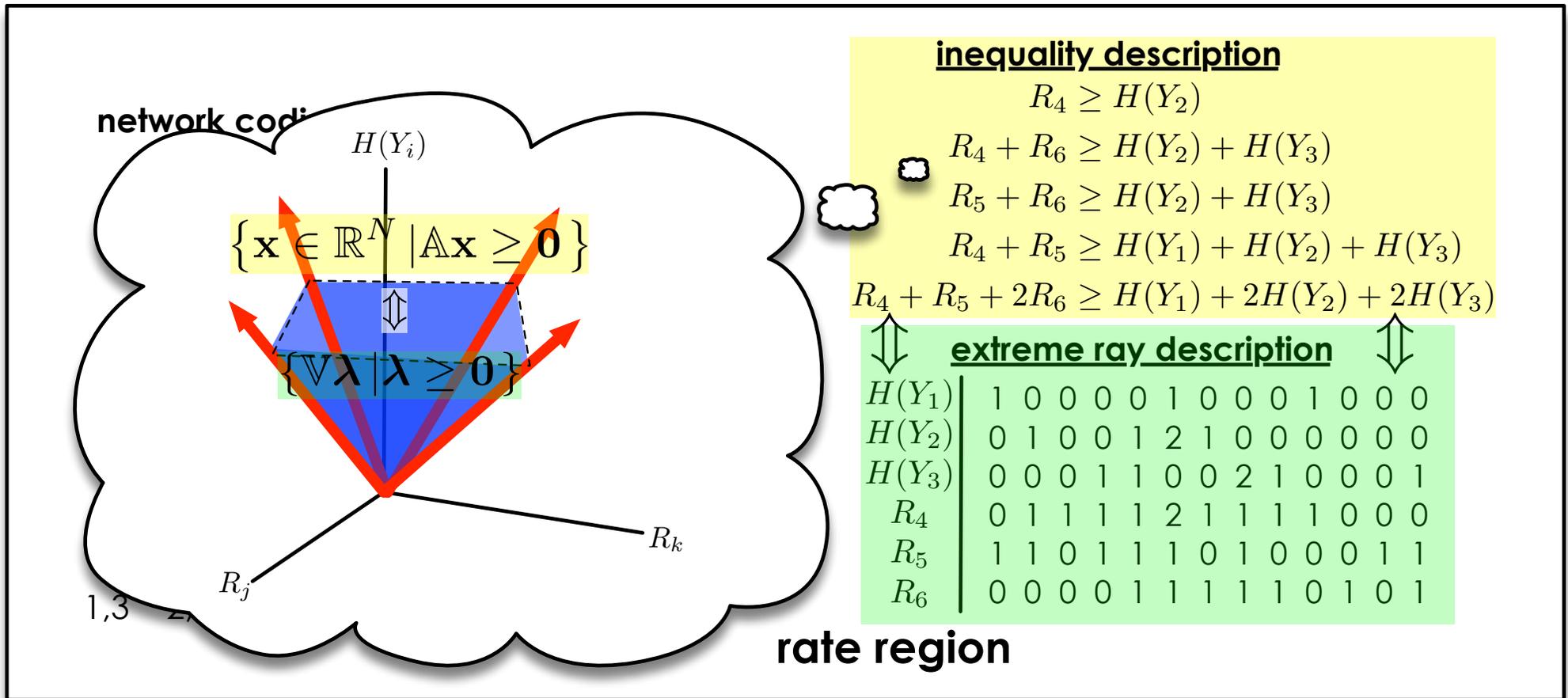
What is a network coding capacity region?



What is a network coding capacity region?

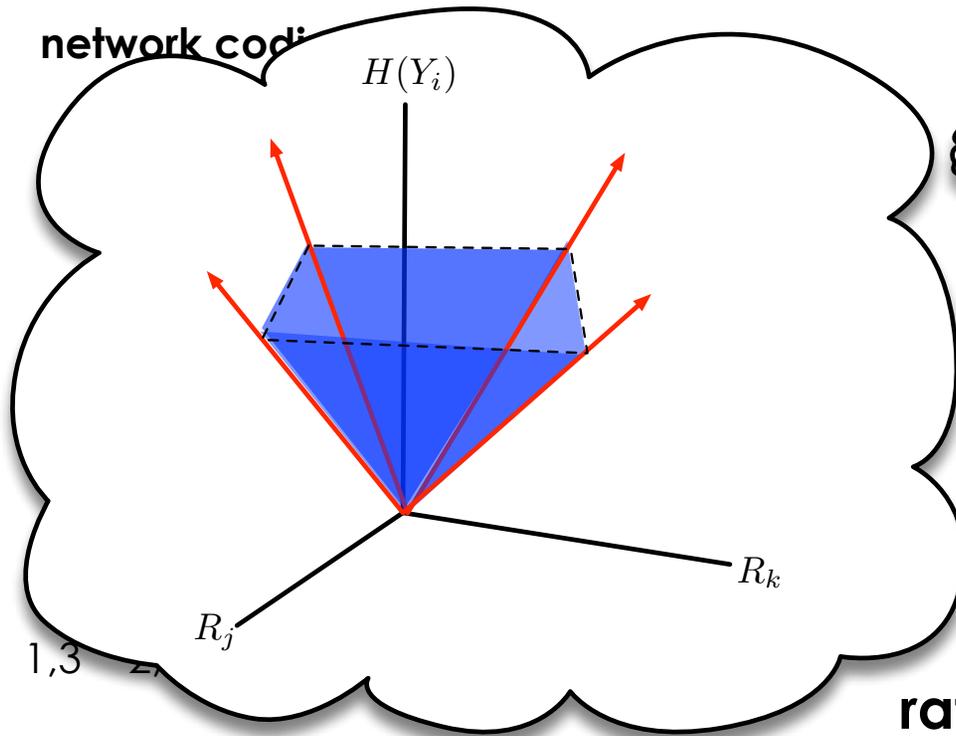


What is a network coding capacity region?



What is a network coding capacity region?

network coding



inequality description

$$R_4 \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$

$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

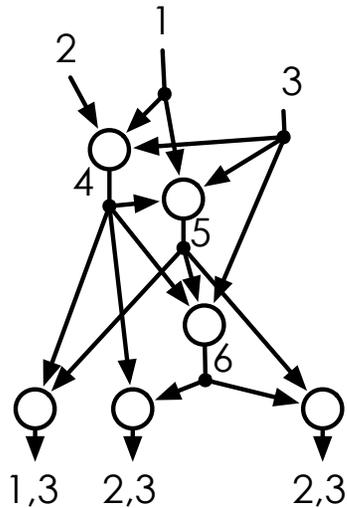
extreme ray description

$H(Y_1)$	1	0	0	0	0	1	0	0	0	1	0	0	0
$H(Y_2)$	0	1	0	0	1	2	1	0	0	0	0	0	0
$H(Y_3)$	0	0	0	1	1	0	0	2	1	0	0	0	1
R_4	0	1	1	1	1	2	1	1	1	1	0	0	0
R_5	1	1	0	1	1	1	0	1	0	0	0	1	1
R_6	0	0	0	0	1	1	1	1	1	0	1	0	1

rate region

What is a network coding capacity region?

network coding problem



inequality description

$$R_4 \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$

$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$



extreme ray description

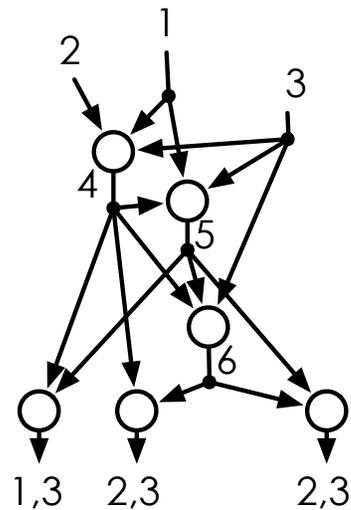


$H(Y_1)$	1	0	0	0	0	1	0	0	0	1	0	0	0
$H(Y_2)$	0	1	0	0	1	2	1	0	0	0	0	0	0
$H(Y_3)$	0	0	0	1	1	0	0	2	1	0	0	0	1
R_4	0	1	1	1	1	2	1	1	1	1	0	0	0
R_5	1	1	0	1	1	1	0	1	0	0	0	1	1
R_6	0	0	0	0	1	1	1	1	1	0	1	0	1

rate region

What is a network coding capacity region?

network coding problem



inequality description

$$R_4 \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$

$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

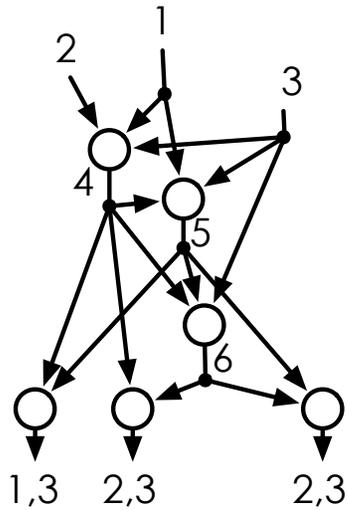
extreme ray description

$H(Y_1)$	1	0	0	0	0	1	0	0	0	1	0	0	0
$H(Y_2)$	0	1	0	0	1	2	1	0	0	0	0	0	0
$H(Y_3)$	0	0	0	1	1	0	0	2	1	0	0	0	1
R_4	0	1	1	1	1	2	1	1	1	1	0	0	0
R_5	1	1	0	1	1	1	0	1	0	0	0	1	1
R_6	0	0	0	0	1	1	1	1	1	0	1	0	1

rate region

What is a network coding capacity region?

network coding problem



inequality description

$$R_4 \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

converse proofs $R_5 + R_6 \geq H(Y_2) + H(Y_3)$

$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$

$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

Proof:

$$R_4 \geq H(U_4)$$

•
•
•

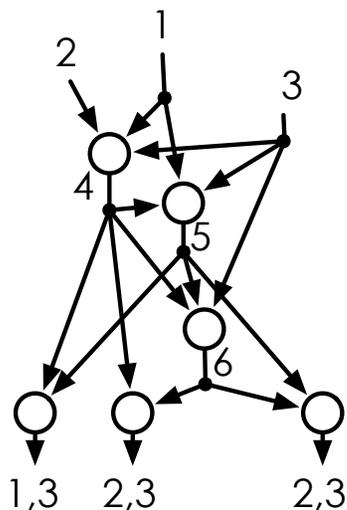
extreme ray description

$H(Y_1)$	1	0	0	0	0	1	0	0	0	1	0	0	0
$H(Y_2)$	0	1	0	0	1	2	1	0	0	0	0	0	0
$H(Y_3)$	0	0	0	1	1	0	0	2	1	0	0	0	1
R_4	0	1	1	1	1	2	1	1	1	1	0	0	0
R_5	1	1	0	1	1	1	0	1	0	0	0	1	1
R_6	0	0	0	0	1	1	1	1	1	0	1	0	1

rate region

What is a network coding capacity region?

network coding problem



inequality description

$$R_4 \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

converse $R_5 + R_6 \geq H(Y_2) + H(Y_3)$

proofs $R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$

$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

Proof:

$$R_4 \geq H(U_4)$$

•
•
•

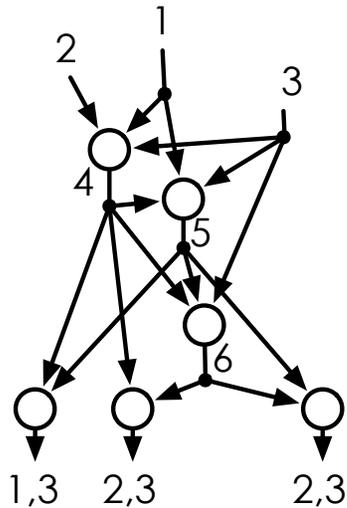
extreme ray description

$H(Y_1)$	1	0	0	0	0	1	0	0	0	1	0	0	0
$H(Y_2)$	0	1	0	0	1	2	1	0	0	0	0	0	0
$H(Y_3)$	0	0	0	1	1	0	0	2	1	0	0	0	1
R_4	0	1	1	1	1	2	1	1	1	1	0	0	0
R_5	1	1	0	1	1	1	0	1	0	0	0	1	1
R_6	0	0	0	0	1	1	1	1	1	0	1	0	1

rate region

What is a network coding capacity region?

network coding problem



inequality description

$$R_4 \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

converse proofs $R_5 + R_6 \geq H(Y_2) + H(Y_3)$

$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$

Proof:

$$R_4 \geq H(U_4)$$

•
•
•

$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

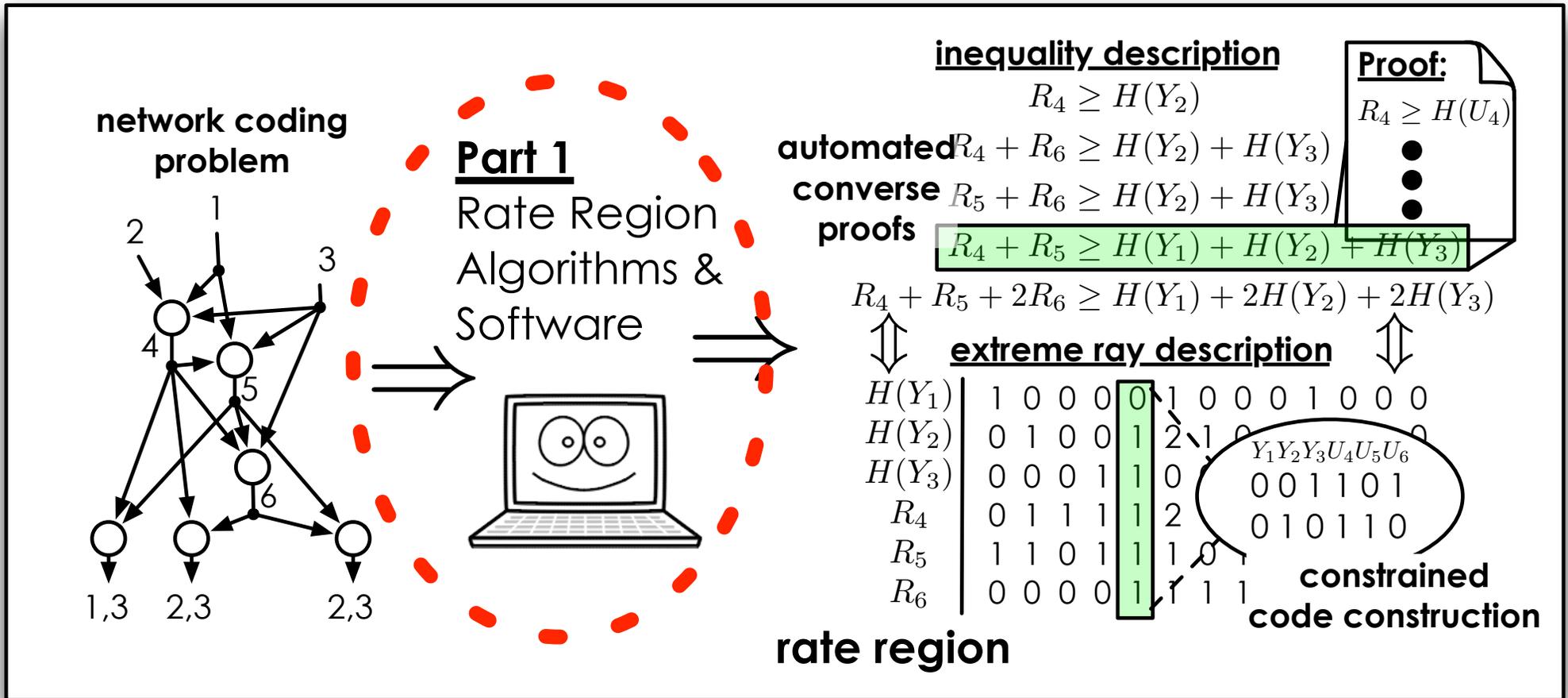
extreme ray description

$H(Y_1)$	1	0	0	0	0	1	0	0	0	1	0	0	0
$H(Y_2)$	0	1	0	0	1	2	1	0	0	0	0	0	0
$H(Y_3)$	0	0	0	1	1	0	0	0	0	0	0	0	0
R_4	0	1	1	1	1	2	0	0	0	0	0	0	0
R_5	1	1	0	1	1	1	0	0	0	0	0	0	0
R_6	0	0	0	0	1	1	1	1	0	0	0	0	0

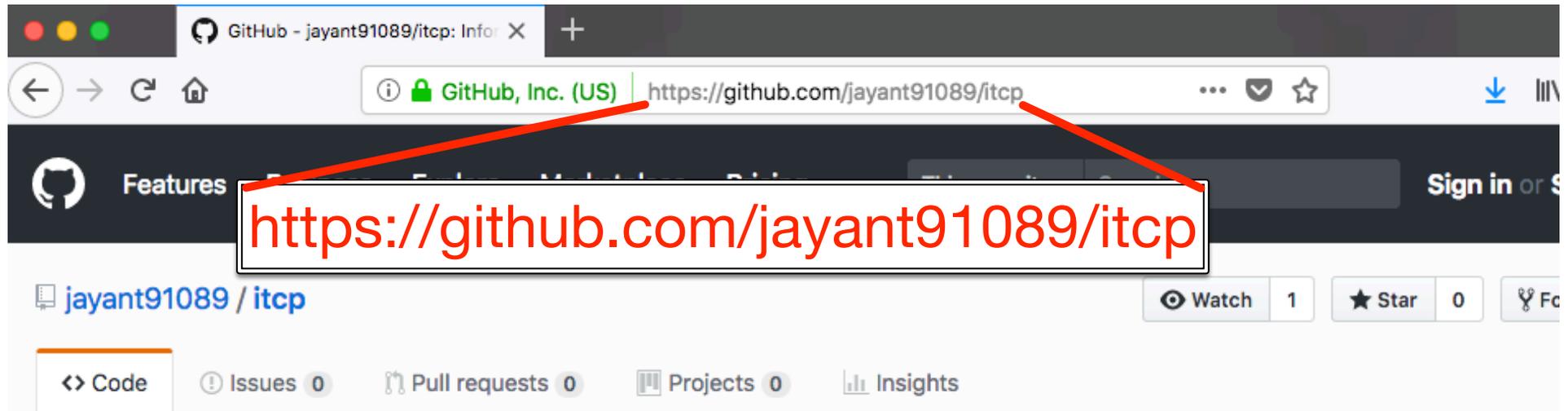
constrained code construction

rate region

Computationally Enabled Research Agenda: #1 = Prove Regions

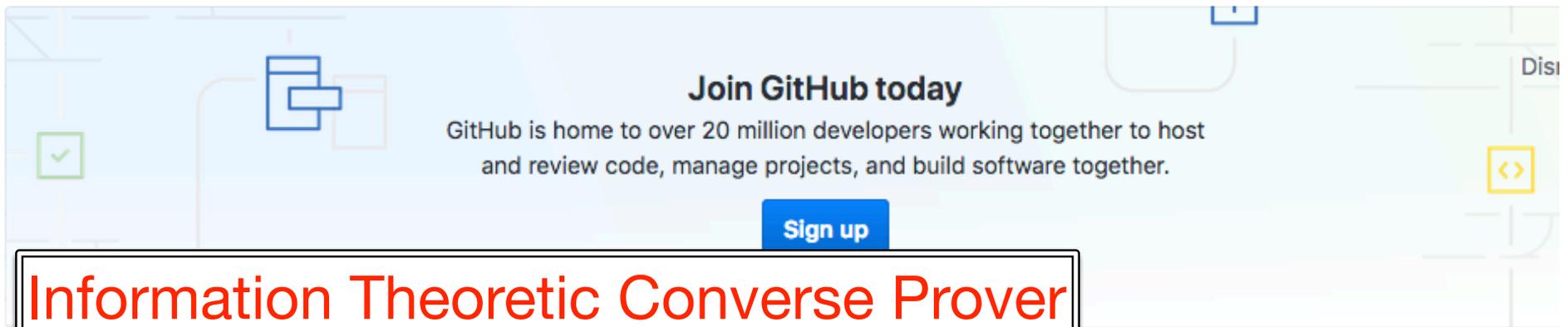


The Information Theoretic Converse Prover – ITCP (github)



A screenshot of a web browser showing the GitHub repository page for 'jayant91089/itcp'. The browser's address bar displays the URL 'https://github.com/jayant91089/itcp'. A red box highlights this URL, with red arrows pointing from the box to the address bar. Below the address bar, the repository name 'jayant91089 / itcp' is visible, along with navigation options like 'Code', 'Issues', 'Pull requests', 'Projects', and 'Insights'. The page also shows 'Watch 1', 'Star 0', and 'Fork' buttons.

<https://github.com/jayant91089/itcp>



A screenshot of a GitHub promotional banner. The banner features the text 'Join GitHub today' and 'GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.' Below this text is a blue 'Sign up' button. A red box highlights the text 'Information Theoretic Converse Prover' in red, with red arrows pointing from the box to the text.

Information Theoretic Converse Prover

Information Theoretic Converse Prover. A software for constructing explicit polyhedral converses in multi-source network coding. Also supports computation of weighted sum-rate bounds in network coding, worst case information ratio lower bounds in secret sharing, and graph guessing number upper bounds.

The Information Theoretic Converse Prover – ITCP is a GAP package!

https://www.gap-system.org

GAP

[Downloads](#) [Installation](#) [Overview](#) [Data Libraries](#) [Packages](#) [Documentation](#) [Contacts](#) [FAQ](#)
[GAP 3](#)

[Find us on GitHub](#)

[Sitemap](#)

Navigation Tree

[Start](#)

[Downloads](#)

[Installation](#)

[Overview](#)

[Data Libraries](#)

[Packages](#)

[Documentation](#)

[Contacts](#)

[FAQ](#)

[GAP 3](#)

>Welcome to

**GAP - Groups, Algorithms, Programming -
a System for Computational Discrete Algebra**

The current version is [GAP 4.8.10](#) released on 15 January 2018.

What is GAP?

GAP is a system for computational discrete algebra, with particular emphasis on [Computational Group Theory](#). GAP provides a [programming language](#), a library of thousands of functions implementing algebraic algorithms written in the GAP language as well as large [data libraries](#) of algebraic objects. See also the [overview](#) and the description of the [mathematical capabilities](#). GAP is used in research and teaching for studying groups and their representations, rings, vector spaces, algebras, combinatorial structures, and more. The system, including source, is distributed [freely](#). You can study and easily modify or extend it for your special use.

Tweets by @gap_system

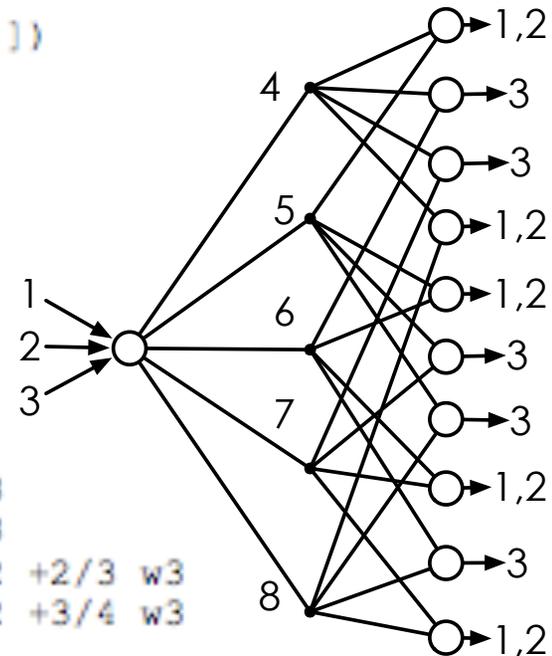
gap-system.org

The Information Theoretic Converse Prover – ITCP (github)

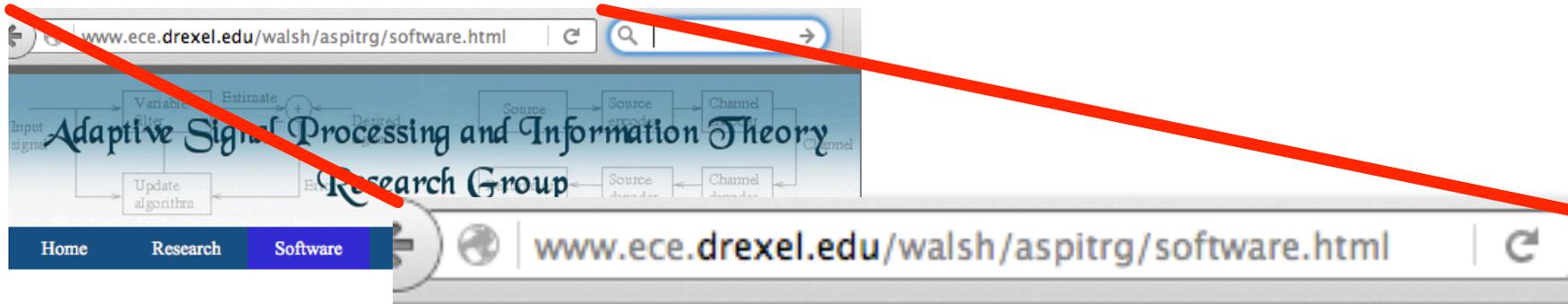
```

gap> # Define a size 8 IDSC instance
> idsc:=([ [ [ 1, 2, 3 ], [ 1, 2, 3, 4, 5, 6, 7, 8 ] ], \
>         [ [ 4, 5 ], [ 1, 2, 4, 5 ] ], [ [ 5, 6 ], [ 1, 2, 5, 6 ] ], \
>         [ [ 6, 7 ], [ 1, 2, 6, 7 ] ], [ [ 7, 8 ], [ 1, 2, 7, 8 ] ], \
>         [ [ 4, 8 ], [ 1, 2, 4, 8 ] ], [ [ 4, 6 ], [ 3, 4, 6 ] ], \
>         [ [ 5, 8 ], [ 3, 5, 8 ] ], [ [ 4, 7 ], [ 3, 4, 7 ] ], \
>         [ [ 5, 7 ], [ 3, 5, 7 ] ], [ [ 6, 8 ], [ 3, 6, 8 ] ] ], 3, 8 );
gap> G:=NetSymGroup(idsc);
Group([ (5,8) (6,7), (4,5) (6,8), (4,6) (7,8), (1,2) ])
gap> Size(G);
20
gap> rlist1:=NCRateRegionOB2(idsc,true,[]);;
gap> Display(rlist1[2]);
0 >= -w2
0 >= -w3
+R4 >= 0
+R4 +R6 >= +w3
+R4 +R5 >= +w1 +w2
+R4 +1/2 R5 +1/2 R8 >= +w1 +w2 +1/2 w3
+1/2 R4 +1/2 R5 +1/2 R6 +1/2 R7 >= +w1 +w2 +1/2 w3
+2/3 R4 +2/3 R5 +1/3 R6 +1/3 R8 >= +w1 +w2 +2/3 w3
+2/3 R4 +1/3 R5 +1/3 R6 +1/3 R7 +1/3 R8 >= +w1 +w2 +2/3 w3
+1/2 R4 +1/2 R5 +1/2 R6 +1/4 R7 +1/4 R8 >= +w1 +w2 +3/4 w3
+R4 +1/2 R5 +1/2 R6 +1/2 R7 >= +w1 +w2 +w3
+R4 +1/2 R5 +1/2 R6 +1/2 R8 >= +w1 +w2 +w3
+R4 +1/3 R5 +1/3 R6 +1/3 R7 +1/3 R8 >= +w1 +w2 +w3
+2/3 R4 +2/3 R5 +1/3 R6 +2/3 R7 +1/3 R8 >= +w1 +w2 +4/3 w3
+R4 +1/2 R5 +1/2 R6 +R7 >= +w1 +w2 +3/2 w3
+R4 +1/2 R5 +1/2 R6 +1/2 R7 +1/2 R8 >= +w1 +w2 +3/2 w3
+2 R4 +R6 +R7 >= +w1 +w2 +2 w3
+R4 +R5 +R6 +R7 >= +w1 +w2 +2 w3

```



The Information Theoretic Achievability Prover (ITAP)



Software Developed by our Research Group

Some of the software we developed for our research projects can be found below.

- **Information Theoretic Achievability Prover (itap)**

itap can perform following tasks:

1. Testing achievability of a rate vector for a network coding instance using vector linear codes over a specified finite field
2. Testing achievability of an information ratio in a secret sharing instance using multi-linear secret sharing

- **Information Theoretic Achievability Prover (itap)**

itap can perform following tasks:

1. Testing achievability of a rate vector for a network coding instance using vector linear codes over a specified finite field
2. Testing achievability of an information ratio in a secret sharing instance using multi-linear secret sharing schemes over a specified finite field
3. Testing representability of an integer polymatroid over a specified finite field.

This software is written in [GAP](#) and is available in form a GAP package (GAP v4.5+). The git repository containing itap can be found [here](#), while the user manual can be found [here](#). This software was developed by [Jayant Apte](#) and [John MacLaren Walsh](#).

transformation enables this same method to calculate projections of unbounded polyhedra. You can find our C implementation of this method [here](#) and a brief set of use instructions [here](#). The library makes use of rational arithmetic based [QSOptex](#) linear program solver and the [Fast Library for Number Theory](#).

This software was developed by [Jayant Apte](#) primarily to serve our needs to calculate non-Shannon inequalities

The Information Theoretic Achievability Prover (ITAP) – Rate Vector Verification

```

johnny@aspitrg4:~/install/gap> LoadPackage("itap");;
johnny@aspitrg4:~/install/gap> net:=[[ [1,2,3],[1,2,3,4]],...],3,6];;
johnny@aspitrg4:~/install/gap> myAns:=proverate(net,[0,1,1,1,1,1],GF(2),[]);;
johnny@aspitrg4:~/install/gap> myAns[1];
true
johnny@aspitrg4:~/install/gap> DisplayCode(myAns[2]);

```

network coding problem

```

graph TD
    1((1)) --> 2((2))
    1((1)) --> 3((3))
    2((2)) --> 4((4))
    2((2)) --> 5((5))
    3((3)) --> 4((4))
    3((3)) --> 5((5))
    4((4)) --> 6((6))
    5((5)) --> 6((6))
    6((6)) --> S1((1,3))
    6((6)) --> S2((2,3))
    6((6)) --> S3((2,3))

```

inequality description

$$R_4 \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$

$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

extreme ray description

$H(Y_1)$	1	0	0	0	0	1	0	0	0	1	0	0	0	0
$H(Y_2)$	0	1	0	0	0	2	1	0	0	0	0	0	0	0
$H(Y_3)$	0	0	0	1	0	0	0	1	1	0	1	0	0	0
R_4	0	1	1	1	1	2	0	0	0	0	0	0	0	0
R_5	1	1	0	1	1	1	1	0	0	0	0	0	0	0
R_6	0	0	0	0	0	1	1	1	1	1	1	1	1	1

rate region

constrained code construction

$Y_1 Y_2 Y_3 U_4 U_5 U_6$

```

0 1 0 0 0 1 0 0 0 0 0 0 0 0 0
1 1 0 0 0 2 1 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 1 1 0 1 0 1 0 1
0 1 0 1 1 0 0 1 1 1 0 1 1 1 0

```

Verification user interface shown. Listing interface available as well. Although enumeration oriented, when used as a verification algorithm (w/ specified rate vector) it can still be faster than the Groebner basis (w/ Singular) based path-gain verification of Subramanian & Thangaraj! (also included)

The Information Theoretic Achievability Prover (ITAP) – Rate Vector Verification

```

johnny@asptrg4: ~/install/gap4r7 — ssh — 85x51
Try '?help' for help. See also '?copyright' and '?authors'
gap> LoadPackage("itap");;

Loading singular 12.04.28
by Marco Costantini (http://www.singular.uni-kl.de)
Homepage: http://www.gap-sing.com/

Loading itap 1.0
by Jayant Apte (https://www.jayantapte.com)
John Walsh (http://www.johnwalsh.com)
For help, type: ?itap

gap> net:=HyperedgeNet1();;
gap> rlist:=proveregion(net,2,GF(2),[4]);; #k=2,4=max. code dim.
gap> lrs_path:="/home/johnny/install/lrslib-061/";;
gap> rrcompute(rlist[1],net[2],net[3],lrs_path);;

-----
gap> net:=HyperedgeNet1();;
gap> rlist:=proveregion(net,2,GF(2),[4]);; #k=2,4=max. code dim.
gap> lrs_path:="/home/johnny/install/lrslib-061/";;
gap> rrcompute(rlist[1],net[2],net[3],lrs_path);;

-----
*redund:lrslib v.6.1 2015.11.20(lrs)
*Copyright (C) 1995,2015, David
*Input taken from file /tmp/tmgznl
*Output sent to file /tmp/tmgznl

*0.056u 0.000s 324Kb 0 flts 0 sw

*lrs:lrslib v.6.1 2015.11.20(lrs)
*Copyright (C) 1995,2015, David
*Input taken from file /tmp/tmgznl

H-representation
begin
***** 7 rational
 0 0 0 0 1 0 0
 0 1 0 0 0 -1 0
 0 0 0 0 0 1 0
 0 0 0 0 0 0 1
 0 0 0 1 0 0 0
 0 1 1 0 -1 -1 -1
 0 0 1 1 0 -1 -1
 0 0 1 0 0 0 0
 0 1 1 2 -1 -2 -2
 0 1 0 1 0 -1 -1
end

*Totals: facets 10 1 1 22
*Dictionary Cache: max size= 6 m
*lrs:lrslib v.6.1 2015.11.20(32bit,lrs)
*0.000u 0.000s 656Kb 0 flts 0 swaps 0 blks-in 0 blks-out

```

```

gap> LoadPackage("itap");;
gap> net:=HyperedgeNet1();;
gap> rlist:=proveregion(net,2,GF(2),[4]);; #k=2,4=max. code dim.
gap> lrs_path:="/home/johnny/install/lrslib-061/";;
gap> rrcompute(rlist[1],net[2],net[3],lrs_path);;

```

H-representation

begin

***** 7 rational

0	0	0	0	1	0	0
0	1	0	0	0	-1	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1
0	0	0	1	0	0	0
0	1	1	0	-1	-1	-1
0	0	1	1	0	-1	-1
0	0	1	0	0	0	0
0	1	1	2	-1	-2	-2
0	1	0	1	0	-1	-1

end

network coding problem

inequality description

$$R_4 \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$

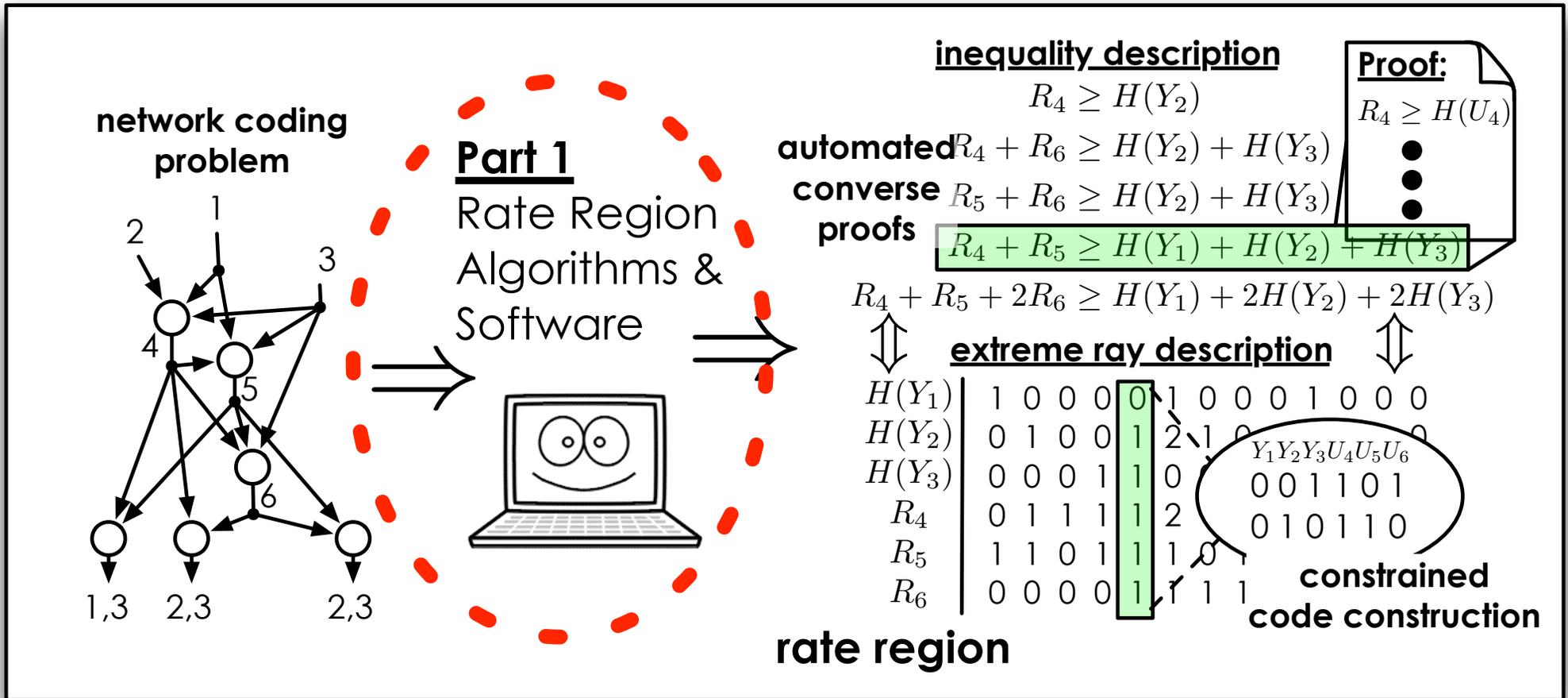
$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

↕ extreme ray description ↕

$H(Y_1)$	1	0	0	0	1	0	0	0	1	0	0	0
$H(Y_2)$	0	1	0	0	1	2	1	0	0	0	0	0
$H(Y_3)$	0	0	0	1	1	0	0	2	1	0	0	0
R_4	0	1	1	1	1	2	1	1	1	1	0	0
R_5	1	1	0	1	1	1	0	1	0	0	0	1
R_6	0	0	0	0	1	1	1	1	1	0	1	0

rate region

Computationally Enabled Research Agenda: #1 = Prove Regions



Computationally Enabled Research Agenda:

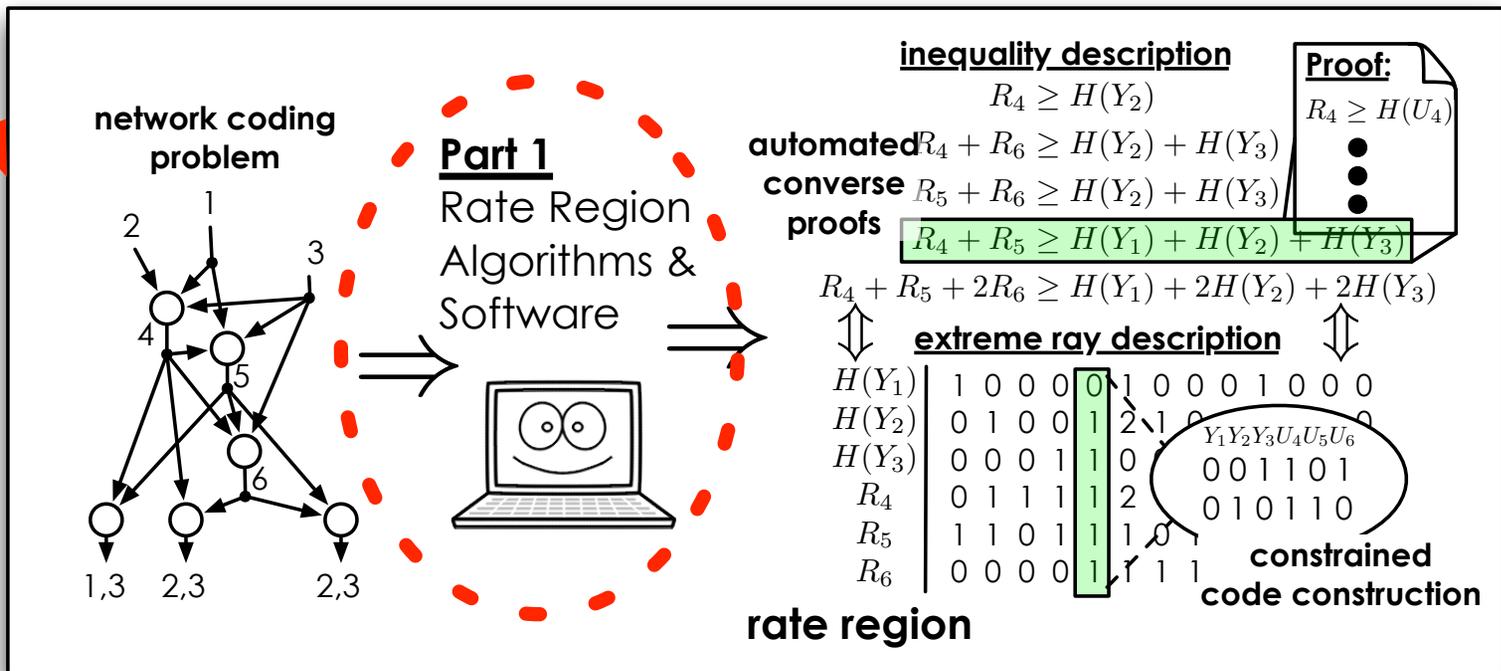
1. Train a computer to calculate network coding capacity regions and their proofs.
2. Build a database of all network coding capacity regions up to a certain size.
3. Organize this database to learn from it, and then to use it to create solutions to networks too large for the computer calculate directly.

Computationally Enabled Research Agenda:

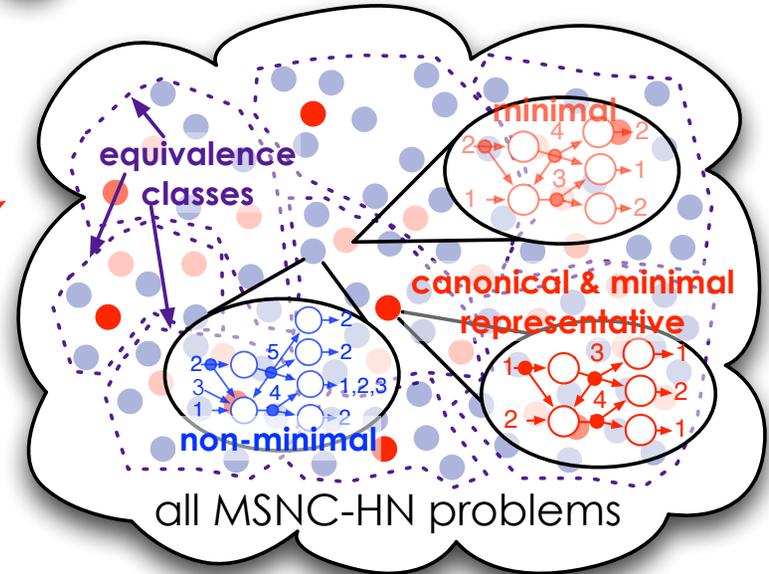
1. Train a computer to calculate network coding capacity regions and their proofs.

2. Build a database of all network coding capacity regions.

3. Organize and network calculations.



1 Formalize Problem Minimality and Equivalence



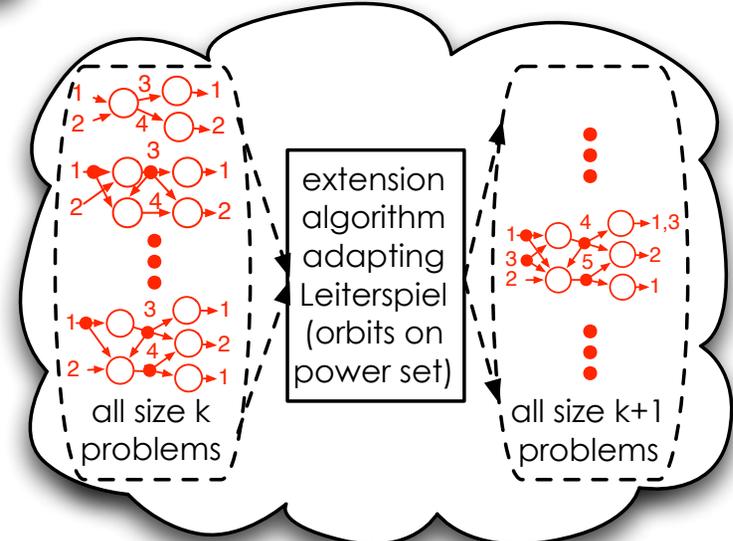
Computationally Enabled Research Agenda:

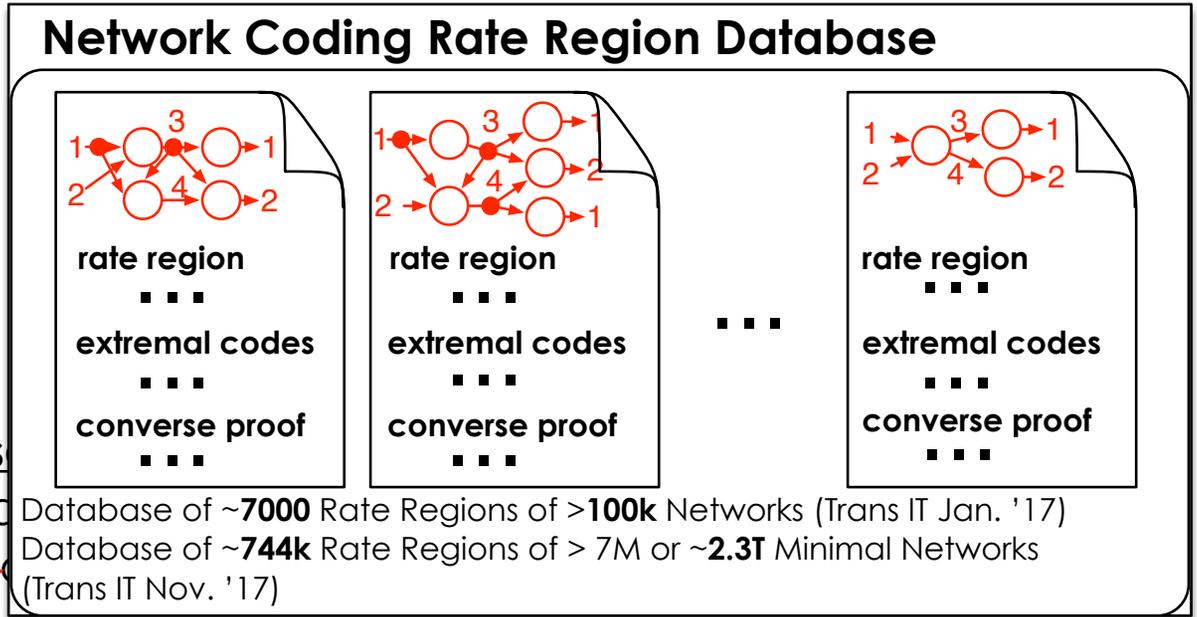
1. Train a computer to calculate network coding capacity regions and their proofs.

2. Build a database of all network coding capacity regions up to a certain size.

3. Organize this database to learn from it, and then to use it to create solutions to networks too large for the computer calculate directly.

2 Develop Algorithm to List only Canonical & Minimal Problems





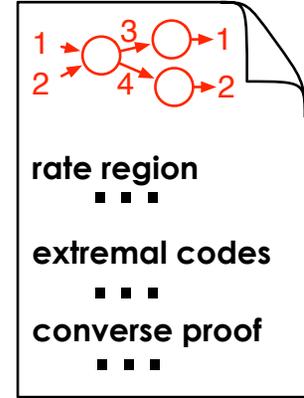
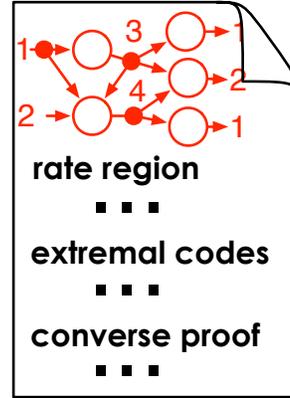
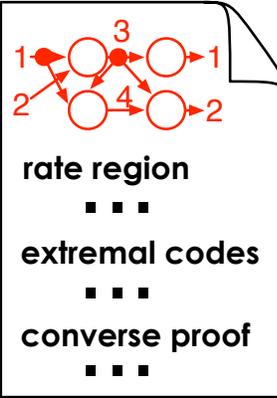
Computationally Enabled Res

1. Train a computer to calculate capacity regions and

2. Build a database of all network coding capacity regions up to a certain size.

3. Organize this database to learn from it, and then to use it to create solutions to networks too large for the computer calculate directly.

Network Coding Rate Region Database



Computationally Enabled Res

1. Train a computer to calculate capacity regions and

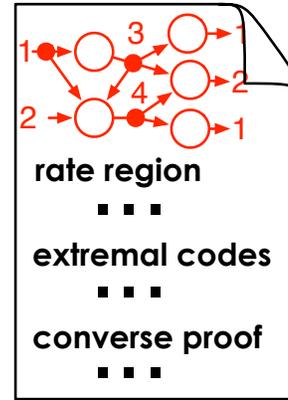
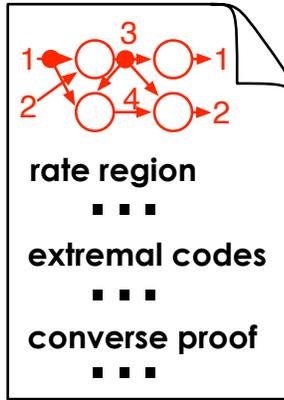
2. Build a database of all network regions up to a certain size.

3. Organize this database to learn from it, and then to use it to create solutions to networks too large for the computer calculate directly.

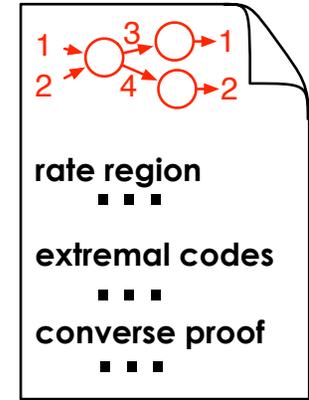
Database of ~7000 Rate Regions of >100k Networks (Trans IT Jan. '17)
Database of ~744k Rate Regions of > 7M or ~2.3T Minimal Networks (Trans IT Nov. '17)

What now?

Network Coding Rate Region Database



■■■



Computationally Enabled Res

1. Train a computer to calculate capacity regions and

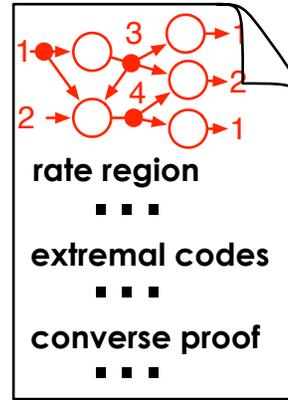
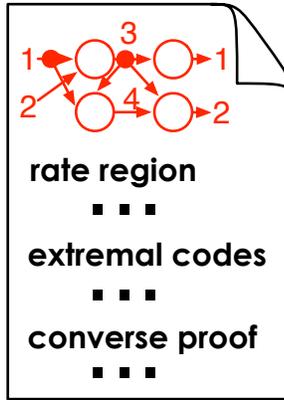
2. Build a database of all network regions up to a certain size.

3. Organize this database to learn from it, and then to use it to create solutions to networks too large for the computer calculate directly.

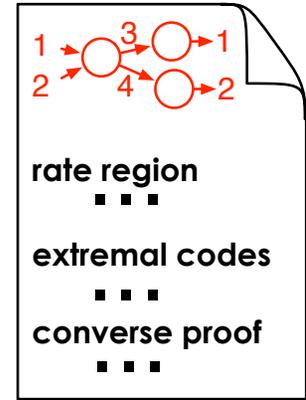
Database of ~7000 Rate Regions of >100k Networks (Trans IT Jan. '17)
Database of ~744k Rate Regions of > 7M or ~2.3T Minimal Networks (Trans IT Nov. '17)

What now? Submit 744,000 transactions papers?

Network Coding Rate Region Database



■ ■ ■



Computationally Enabled Research

1. Train a computer to calculate capacity regions and

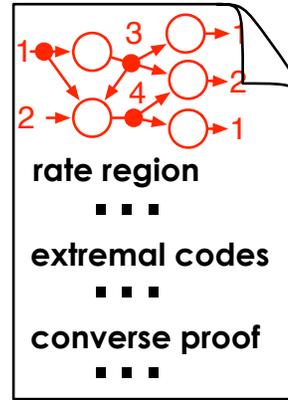
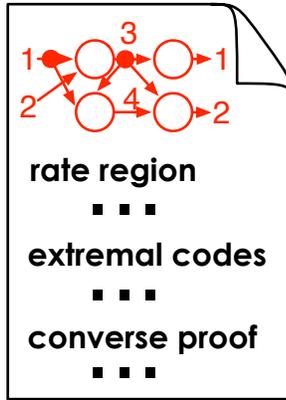
2. Build a database of all network regions up to a certain size.

3. Organize this database to learn from it, and then to use it to create solutions to networks too large for the computer calculate directly.

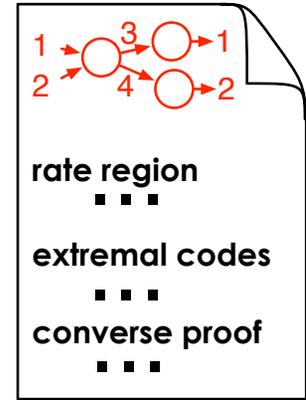
Database of ~7000 Rate Regions of >100k Networks (Trans IT Jan. '17)
 Database of ~744k Rate Regions of > 7M or ~2.3T Minimal Networks (Trans IT Nov. '17)

What now? Analyze, learn, and explain something!
 Can't read and remember 744,000 network proofs.

Network Coding Rate Region Database



...



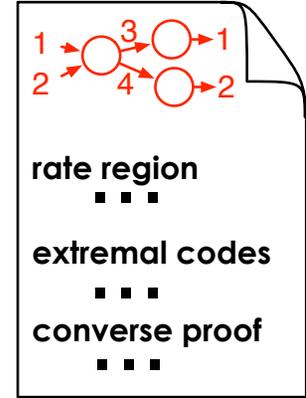
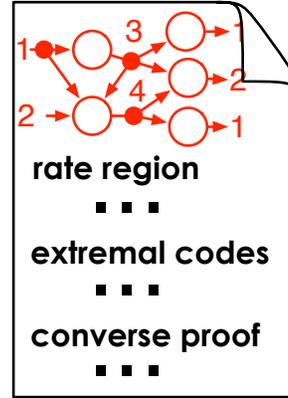
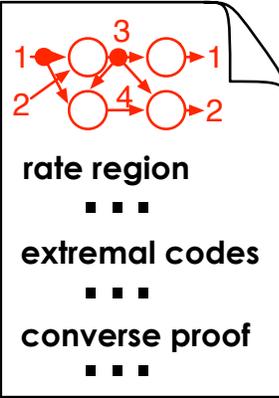
Computationally Enabled Res

1. Train a computer to calculate capacity regions and
2. Build a database of all network regions up to a certain size
3. Organize this database to use it to create networks too large for the calculate directly.

Database of ~7000 Rate Regions of >100k Networks (Trans IT Jan. '17)
 Database of ~744k Rate Regions of > 7M or ~2.3T Minimal Networks
 (Trans IT Nov. '17)

What now? Analyze, learn, and explain something!
 Can't read and remember 744,000 network proofs.
 Software handles only small nets (max 8-10 edges).
 Community taste (caching, storage) is for large graphs and low dim. projections of rate regions

Network Coding Rate Region Database



Computationally Enabled Res

1. Train a computer to calculate capacity regions and

2. Build a database of all network regions up to a certain size.

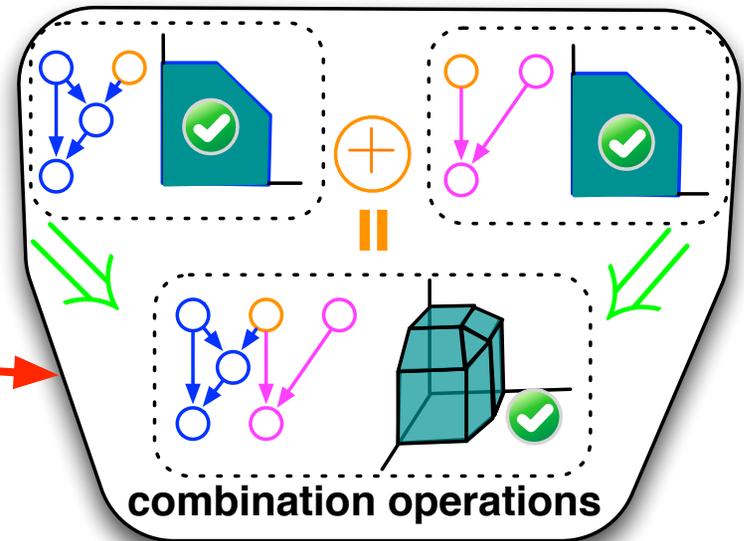
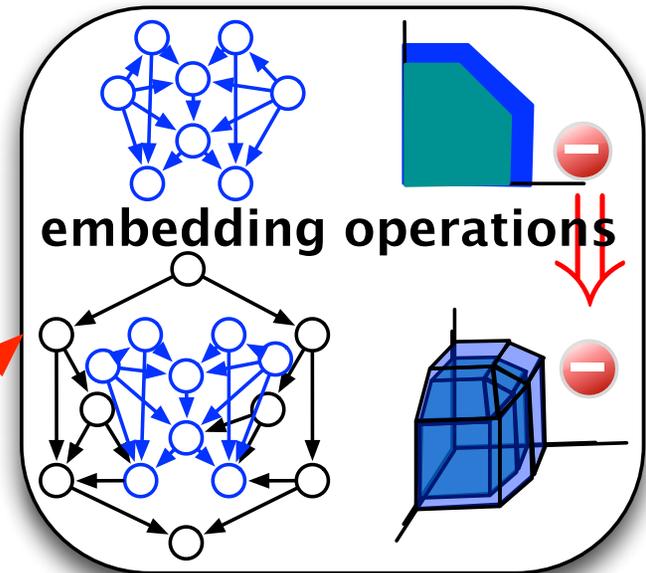
3. Organize this database to learn from it, and then to use it to create solutions to networks too large for the computer calculate directly.

Database of ~7000 Rate Regions of >100k Networks (Trans IT Jan. '17)
Database of ~744k Rate Regions of > 7M or ~2.3T Minimal Networks (Trans IT Nov. '17)

Investigate Structure through Network Hierarchy

Computationally Enabled Research Agenda:

1. Train a computer to calculate network coding capacity regions and their proofs.
2. Build a database of all network coding capacity regions up to a certain size
- 3. Organize this database to learn from it, and then to use it to create solutions to networks too large for the computer calculate directly.**



Background: Inspiration for Hierarchy – Well quasi-ordering of Graphs

Definition 1 (Graph Minor): A graph $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ is a *minor* of another graph $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ if \mathcal{G}_1 can be obtained through a sequence of node deletions, edge deletions, and contractions.

Background: Inspiration for Hierarchy – Well quasi-ordering of Graphs

Definition 1 (Graph Minor): A graph $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ is a *minor* of another graph $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ if \mathcal{G}_1 can be obtained through a sequence of node deletions, edge deletions, and contractions.

Theorem 1 (Kuratowski/ Wagner): A graph $(\mathcal{V}, \mathcal{E})$ is planar if and only if it has no $K_{3,3}$ or K_5 minor.

Background: Inspiration for Hierarchy – Well quasi-ordering of Graphs

Definition 1 (Graph Minor): A graph $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ is a *minor* of another graph $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ if \mathcal{G}_1 can be obtained through a sequence of node deletions, edge deletions, and contractions.

Theorem 1 (Kuratowski/ Wagner): A graph $(\mathcal{V}, \mathcal{E})$ is planar if and only if it has no $K_{3,3}$ or K_5 minor.

Observe: the set of planar graphs is closed under the operation of taking minors.

Background: Inspiration for Hierarchy – Well quasi-ordering of Graphs

Definition 1 (Graph Minor): A graph $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ is a *minor* of another graph $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ if \mathcal{G}_1 can be obtained through a sequence of node deletions, edge deletions, and contractions.

Theorem 1 (Kuratowski/ Wagner): A graph $(\mathcal{V}, \mathcal{E})$ is planar if and only if it has no $K_{3,3}$ or K_5 minor.

Observe: the set of planar graphs is closed under the operation of taking minors.

+ series of 20 papers over 20 years =

Background: Inspiration for Hierarchy – Well quasi-ordering of Graphs

Definition 1 (Graph Minor): A graph $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ is a *minor* of another graph $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ if \mathcal{G}_1 can be obtained through a sequence of node deletions, edge deletions, and contractions.

Theorem 1 (Kuratowski/ Wagner): A graph $(\mathcal{V}, \mathcal{E})$ is planar if and only if it has no $K_{3,3}$ or K_5 minor.

Observe: the set of planar graphs is closed under the operation of taking minors.

+ series of 20 papers over 20 years =

Theorem 2 (Robertson-Seymour Theorem): Any family of graphs that is closed under the operation of taking minors has at most a finite series of forbidden minors.

Background: Inspiration for Hierarchy – Well quasi-ordering of Graphs

Definition 1 (Graph Minor): A graph $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ is a *minor* of another graph $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ if \mathcal{G}_1 can be obtained through a sequence of node deletions, edge deletions, and contractions.

Theorem 1 (Kuratowski/ Wagner): A graph $(\mathcal{V}, \mathcal{E})$ is planar if and only if it has no $K_{3,3}$ or K_5 minor.

Observe: the set of planar graphs is closed under the operation of taking minors.

+ series of 20 papers over 20 years =

Theorem 2 (Robertson-Seymour Theorem): Any family of graphs that is closed under the operation of taking minors has at most a finite series of forbidden minors.

Equivalent to stating that there are no infinite anti-chains (any infinite sequence of graphs must have a pair with a minor relationship) and no infinite descending chains (WQO).

Background: Inspiration for Hierarchy – Forbidden Minors Continued

- network coding problems are part graph based (already enough)

Background: Inspiration for Hierarchy – Forbidden Minors Continued

- network coding problems are part graph based (already enough)
- Network coding rate regions are built not only from labelled graphs, but also from entropy functions, which are polymatroids, \supseteq matroids \supseteq independence in graphs

Background: Inspiration for Hierarchy – Forbidden Minors Continued

- network coding problems are part graph based (already enough)
- Network coding rate regions are built not only from labelled graphs, but also from entropy functions, which are polymatroids, \supseteq matroids \supseteq independence in graphs
- roughly, set sources = independent uniform RVs, call what is set on an edge e a RV U_e . Collect all RV.s into $\mathbf{X} = (Y_s, U_e | s \in \mathcal{S}, e \in \mathcal{E})$.

Background: Inspiration for Hierarchy – Forbidden Minors Continued

- network coding problems are part graph based (already enough)
- Network coding rate regions are built not only from labelled graphs, but also from entropy functions, which are polymatroids, \supseteq matroids \supseteq independence in graphs
- roughly, set sources = independent uniform RVs, call what is set on an edge e a RV U_e . Collect all RV.s into $\mathbf{X} = (Y_s, U_e | s \in \mathcal{S}, e \in \mathcal{E})$.

$$h(\mathcal{A}) = H(\mathbf{X}_{\mathcal{A}}), \quad \mathcal{A} \subseteq \mathcal{N} = \mathcal{S} \cup \mathcal{E} \quad (4)$$

is a *polymatroid*.

Background: Inspiration for Hierarchy – Forbidden Minors Continued

- network coding problems are part graph based (already enough)
- Network coding rate regions are built not only from labelled graphs, but also from entropy functions, which are polymatroids, \supseteq matroids \supseteq independence in graphs
- roughly, set sources = independent uniform RVs, call what is set on an edge e a RV U_e . Collect all RV.s into $\mathbf{X} = (Y_s, U_e | s \in \mathcal{S}, e \in \mathcal{E})$.

$$h(\mathcal{A}) = H(\mathbf{X}_{\mathcal{A}}), \quad \mathcal{A} \subseteq \mathcal{N} = \mathcal{S} \cup \mathcal{E} \quad (5)$$

is a *polymatroid*.

Definition 2 (polymatroid): A set function $\rho : 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$ is a polymatroid if $\forall \mathcal{A}, \mathcal{B}$, $\rho(\mathcal{A}) + \rho(\mathcal{B}) \geq \rho(\mathcal{A} \cup \mathcal{B}) + \rho(\mathcal{A} \cap \mathcal{B})$ – submodular, and for any $\mathcal{C} \subseteq \mathcal{D}$, $\rho(\mathcal{C}) \leq \rho(\mathcal{D})$ – non-decreasing.

Background: Inspiration for Hierarchy – Forbidden Minors Continued

- network coding problems are part graph based (already enough)
- Network coding rate regions are built not only from labelled graphs, but also from entropy functions, which are polymatroids, \supseteq matroids \supseteq independence in graphs
- roughly, set sources = independent uniform RVs, call what is set on an edge e a RV U_e . Collect all RV.s into $\mathbf{X} = (Y_s, U_e | s \in \mathcal{S}, e \in \mathcal{E})$.

$$h(\mathcal{A}) = H(\mathbf{X}_{\mathcal{A}}), \quad \mathcal{A} \subseteq \mathcal{N} = \mathcal{S} \cup \mathcal{E} \quad (6)$$

is a *polymatroid*.

Definition 2 (polymatroid): A set function $\rho : 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$ is a polymatroid if $\forall \mathcal{A}, \mathcal{B}$, $\rho(\mathcal{A}) + \rho(\mathcal{B}) \geq \rho(\mathcal{A} \cup \mathcal{B}) + \rho(\mathcal{A} \cap \mathcal{B})$ – submodular, and for any $\mathcal{C} \subseteq \mathcal{D}$, $\rho(\mathcal{C}) \leq \rho(\mathcal{D})$ – non-decreasing.

Definition 3 (matroid): A matroid is a polymatroid ρ taking values in $\mathbb{Z}_{\geq 0}$ for whom $\rho(\mathcal{A}) \leq |\mathcal{A}|$.

Background: Inspiration for Hierarchy – Rota's Conjecture

Definition 4 (Matroid Deletion): $\rho' : 2^{\mathcal{N}'} \rightarrow \mathbb{Z}_{\geq 0}$ is obtained by deleting $e \in \mathcal{N}$ from ρ if $\mathcal{N}' = \mathcal{N} \setminus \{e\}$ and $\rho'(\mathcal{A}) = \rho(\mathcal{A}), \forall \mathcal{A} \subseteq \mathcal{N}'$.

Definition 5 (Matroid contraction): $\rho' : 2^{\mathcal{N}'} \rightarrow \mathbb{Z}_{\geq 0}$ is obtained by contracting $e \in \mathcal{N}$ from ρ if $\mathcal{N}' = \mathcal{N} \setminus \{e\}$ and $\rho'(\mathcal{A}) = \rho(\mathcal{A} \cup \{e\}) - \rho(\{e\}), \forall \mathcal{A} \subseteq \mathcal{N}'$. (condition entropy on X_e).

Background: Inspiration for Hierarchy – Rota's Conjecture

Definition 4 (Matroid Deletion): $\rho' : 2^{\mathcal{N}'} \rightarrow \mathbb{Z}_{\geq 0}$ is obtained by deleting $e \in \mathcal{N}$ from ρ if $\mathcal{N}' = \mathcal{N} \setminus \{e\}$ and $\rho'(\mathcal{A}) = \rho(\mathcal{A}), \forall \mathcal{A} \subseteq \mathcal{N}'$.

Definition 5 (Matroid contraction): $\rho' : 2^{\mathcal{N}'} \rightarrow \mathbb{Z}_{\geq 0}$ is obtained by contracting $e \in \mathcal{N}$ from ρ if $\mathcal{N}' = \mathcal{N} \setminus \{e\}$ and $\rho'(\mathcal{A}) = \rho(\mathcal{A} \cup \{e\}) - \rho(\{e\}), \forall \mathcal{A} \subseteq \mathcal{N}'$. (condition entropy on X_e).

Definition 6 (Matroid Minor): ρ' is a minor of ρ if it can be obtained by a series of deletions and contractions.

Background: Inspiration for Hierarchy – Rota's Conjecture

Definition 4 (Matroid Deletion): $\rho' : 2^{\mathcal{N}'} \rightarrow \mathbb{Z}_{\geq 0}$ is obtained by deleting $e \in \mathcal{N}$ from ρ if $\mathcal{N}' = \mathcal{N} \setminus \{e\}$ and $\rho'(\mathcal{A}) = \rho(\mathcal{A}), \forall \mathcal{A} \subseteq \mathcal{N}'$.

Definition 5 (Matroid contraction): $\rho' : 2^{\mathcal{N}'} \rightarrow \mathbb{Z}_{\geq 0}$ is obtained by contracting $e \in \mathcal{N}$ from ρ if $\mathcal{N}' = \mathcal{N} \setminus \{e\}$ and $\rho'(\mathcal{A}) = \rho(\mathcal{A} \cup \{e\}) - \rho(\{e\}), \forall \mathcal{A} \subseteq \mathcal{N}'$. (condition entropy on X_e).

Definition 6 (Matroid Minor): ρ' is a minor of ρ if it can be obtained by a series of deletions and contractions.

Matroids do not exhibit WQO (\exists infinite antichains). HOWEVER

Theorem 3 (Tutte (1958)): A matroid is binary if and only if it has no $U_{2,4}$ minor
($\rho_{U_{2,4}}(\mathcal{A}) = \min\{|\mathcal{A}|, 2\}, |\mathcal{N}| = 4$).

Background: Inspiration for Hierarchy – Rota’s Conjecture

Definition 4 (Matroid Deletion): $\rho' : 2^{\mathcal{N}'} \rightarrow \mathbb{Z}_{\geq 0}$ is obtained by deleting $e \in \mathcal{N}$ from ρ if $\mathcal{N}' = \mathcal{N} \setminus \{e\}$ and $\rho'(\mathcal{A}) = \rho(\mathcal{A}), \forall \mathcal{A} \subseteq \mathcal{N}'$.

Definition 5 (Matroid contraction): $\rho' : 2^{\mathcal{N}'} \rightarrow \mathbb{Z}_{\geq 0}$ is obtained by contracting $e \in \mathcal{N}$ from ρ if $\mathcal{N}' = \mathcal{N} \setminus \{e\}$ and $\rho'(\mathcal{A}) = \rho(\mathcal{A} \cup \{e\}) - \rho(\{e\}), \forall \mathcal{A} \subseteq \mathcal{N}'$. (condition entropy on X_e).

Definition 6 (Matroid Minor): ρ' is a minor of ρ if it can be obtained by a series of deletions and contractions.

Matroids do not exhibit WQO (\exists infinite antichains). HOWEVER

Theorem 3 (Tutte (1958)): A matroid is binary if and only if it has no $U_{2,4}$ minor ($\rho_{U_{2,4}}(\mathcal{A}) = \min\{|\mathcal{A}|, 2\}, |\mathcal{N}| = 4$).

Similar lists for \mathbb{F}_3 & \mathbb{F}_4 : Seymour in 1979 & Geelen Gerards Kapoor in 2000, resp.

Background: Inspiration for Hierarchy – Rota’s Conjecture

Definition 4 (Matroid Deletion): $\rho' : 2^{\mathcal{N}'} \rightarrow \mathbb{Z}_{\geq 0}$ is obtained by deleting $e \in \mathcal{N}$ from ρ if $\mathcal{N}' = \mathcal{N} \setminus \{e\}$ and $\rho'(\mathcal{A}) = \rho(\mathcal{A}), \forall \mathcal{A} \subseteq \mathcal{N}'$.

Definition 5 (Matroid contraction): $\rho' : 2^{\mathcal{N}'} \rightarrow \mathbb{Z}_{\geq 0}$ is obtained by contracting $e \in \mathcal{N}$ from ρ if $\mathcal{N}' = \mathcal{N} \setminus \{e\}$ and $\rho'(\mathcal{A}) = \rho(\mathcal{A} \cup \{e\}) - \rho(\{e\}), \forall \mathcal{A} \subseteq \mathcal{N}'$. (condition entropy on X_e).

Definition 6 (Matroid Minor): ρ' is a minor of ρ if it can be obtained by a series of deletions and contractions.

Matroids do not exhibit WQO (\exists infinite antichains). HOWEVER

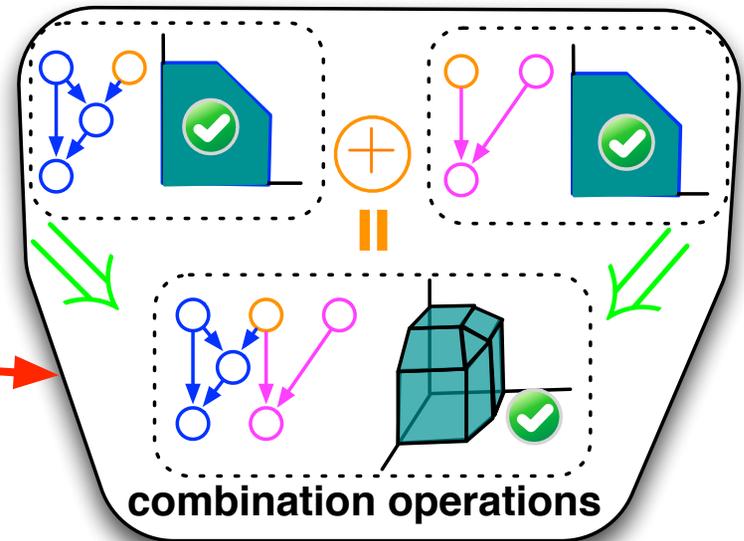
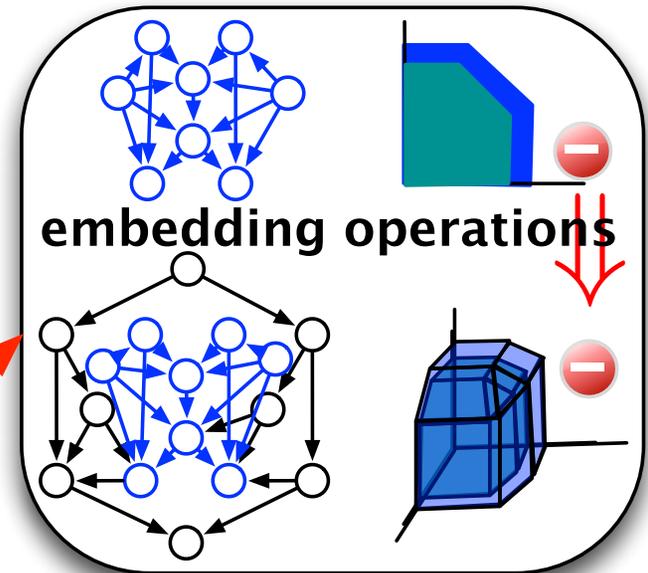
Theorem 3 (Tutte (1958)): A matroid is binary if and only if it has no $U_{2,4}$ minor ($\rho_{U_{2,4}}(\mathcal{A}) = \min\{|\mathcal{A}|, 2\}, |\mathcal{N}| = 4$).

Similar lists for \mathbb{F}_3 & \mathbb{F}_4 : Seymour in 1979 & Geelen Gerards Kapoor in 2000, resp.

Theorem 4 (Rota’s Conjecture (1970) Proved 2013 by Geelen, Gerards, Whittle): The set of matroids representable over \mathbb{F}_q (translation: set of $h(\mathcal{A})$ arising from scalar codes over \mathbb{F}_q) has a most a finite number of forbidden minors.

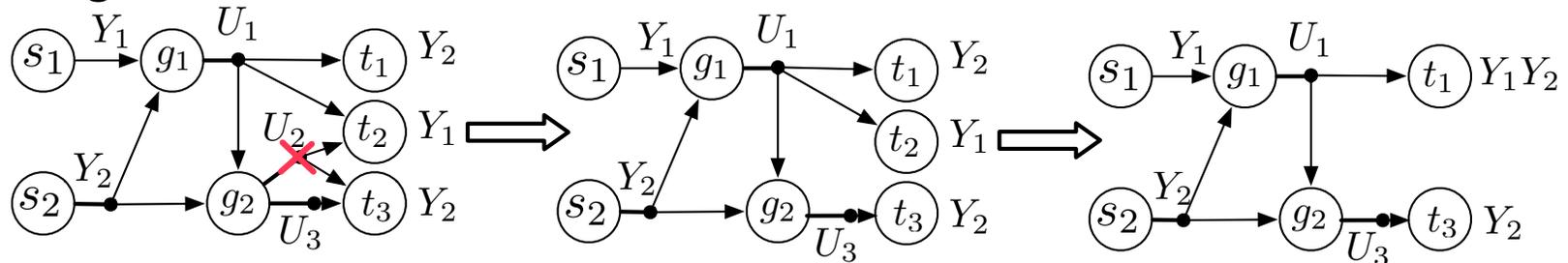
Computationally Enabled Research Agenda:

1. Train a computer to calculate network coding capacity regions and their proofs.
2. Build a database of all network coding capacity regions up to a certain size
- 3. Organize this database to learn from it, and then to use it to create solutions to networks too large for the computer calculate directly.**

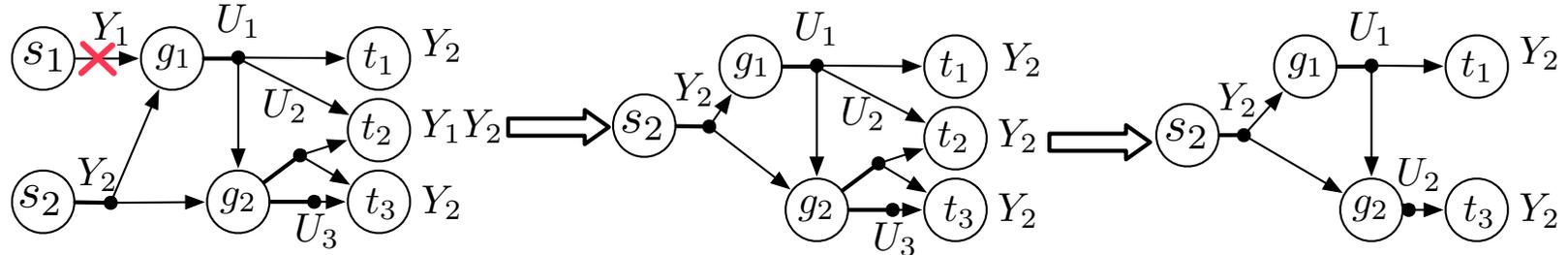


Computationally Enabled Research Agenda – Hierarchy: Embedding Operators

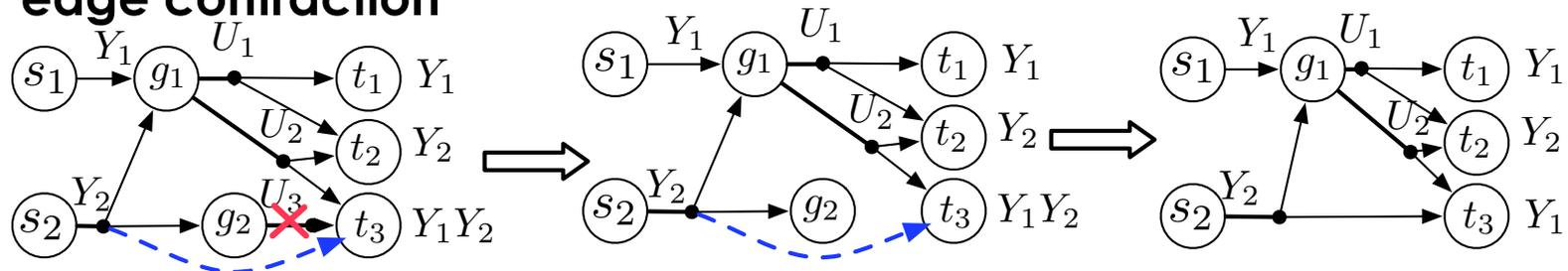
edge deletion



source deletion

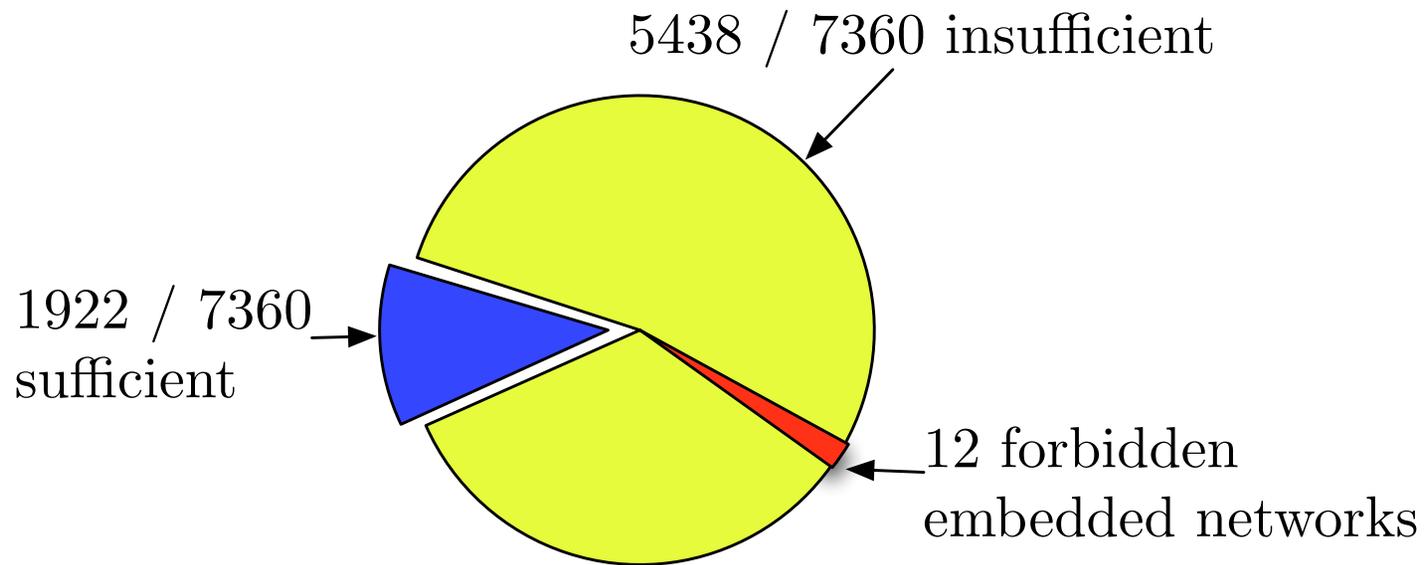


edge contraction



- Rate region (bound) of embedded network can be directly obtained from rate region (bound) of parent network.
- Insufficiency of class of codes of small \implies insufficiency of class of codes of big. (*forbidden network minor*)

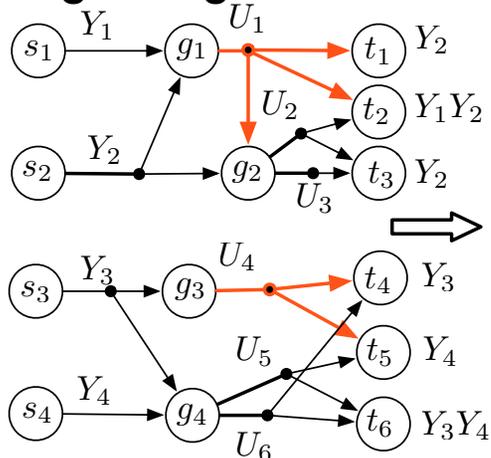
Computationally Enabled Research Agenda – Hierarchy: Embedding Operators



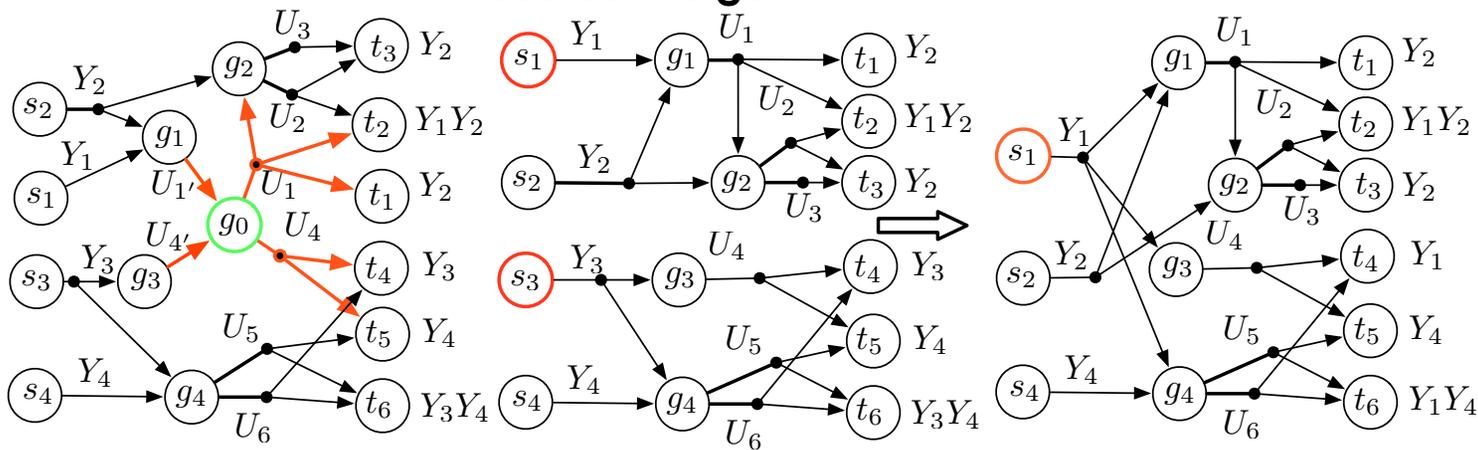
- First database: 5438 canonical MDCS for which scalar binary codes are insufficient can be boiled down to 12 forbidden minor networks.

Computationally Enabled Research Agenda – Hierarchy: Combination Operators

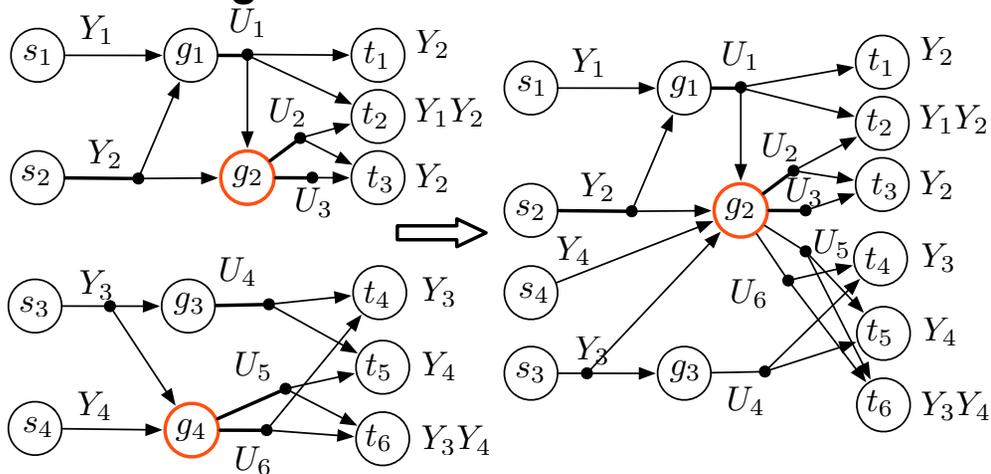
Edge Merge



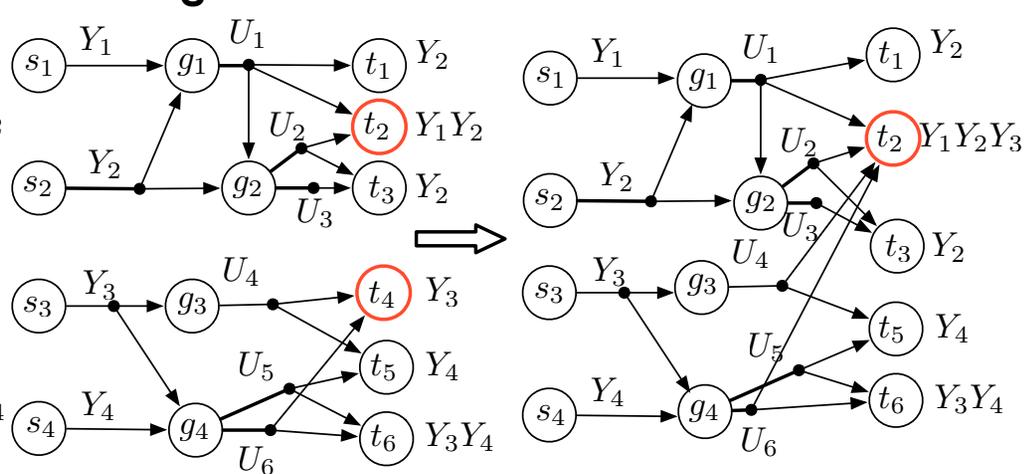
Source Merge



Node Merge

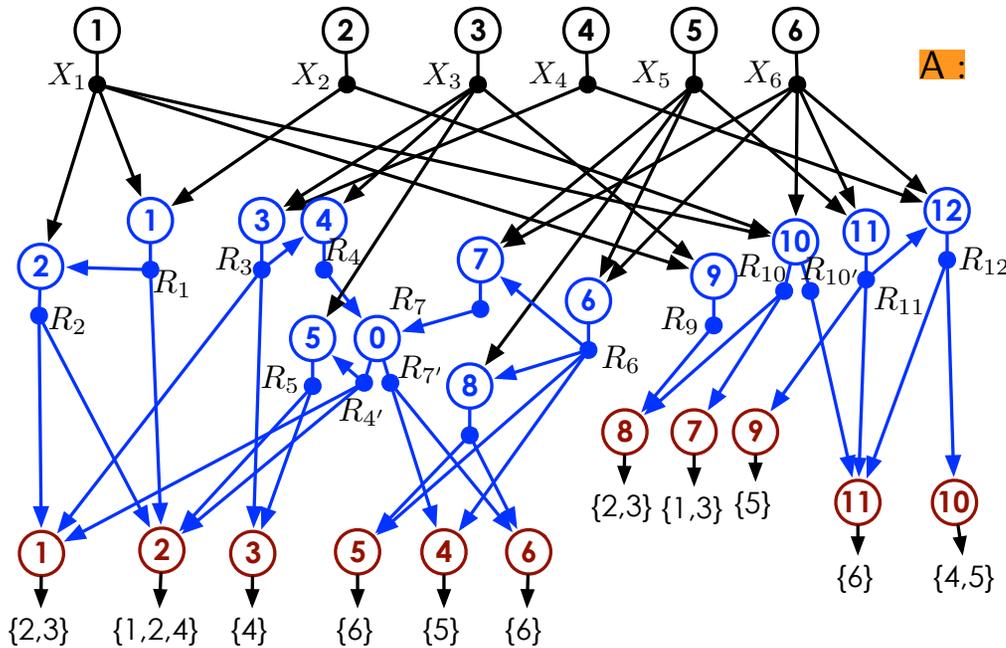


Sink Merge

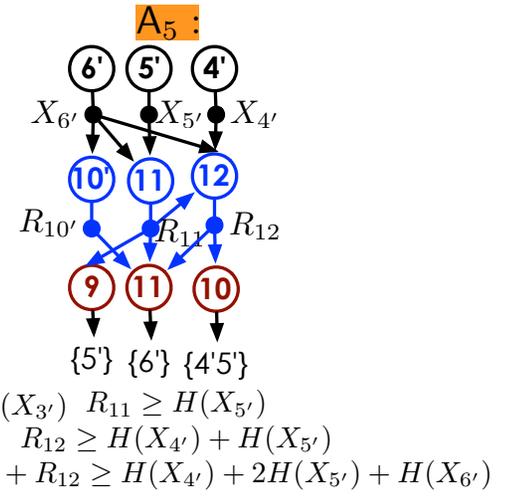
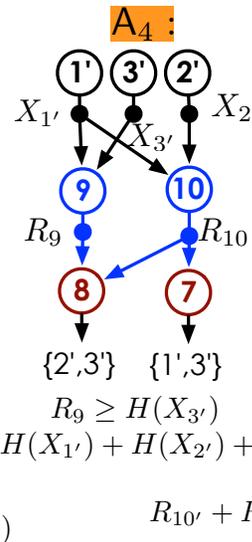
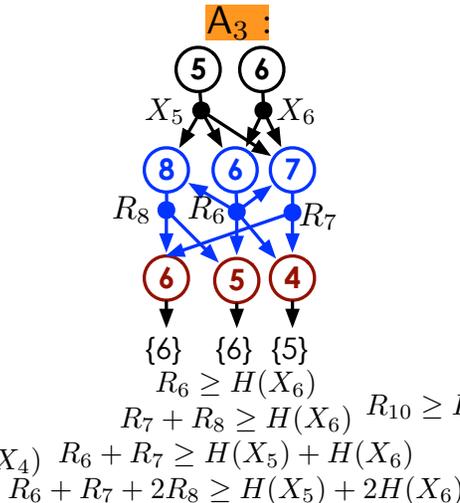
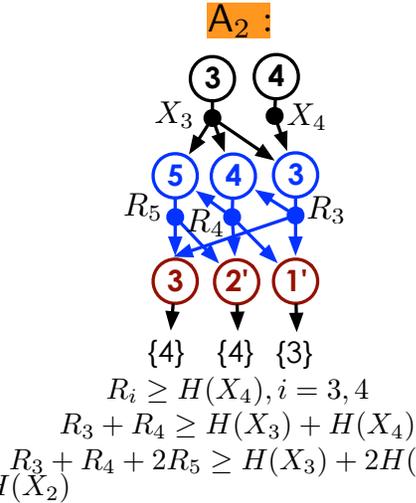
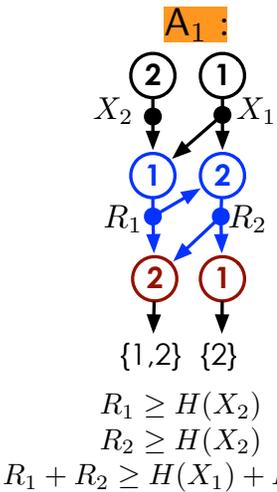


- rate region of big directly expressible from rate regions of smalls

Computationally Enabled Research Agenda – Hierarchy: Combination Operators

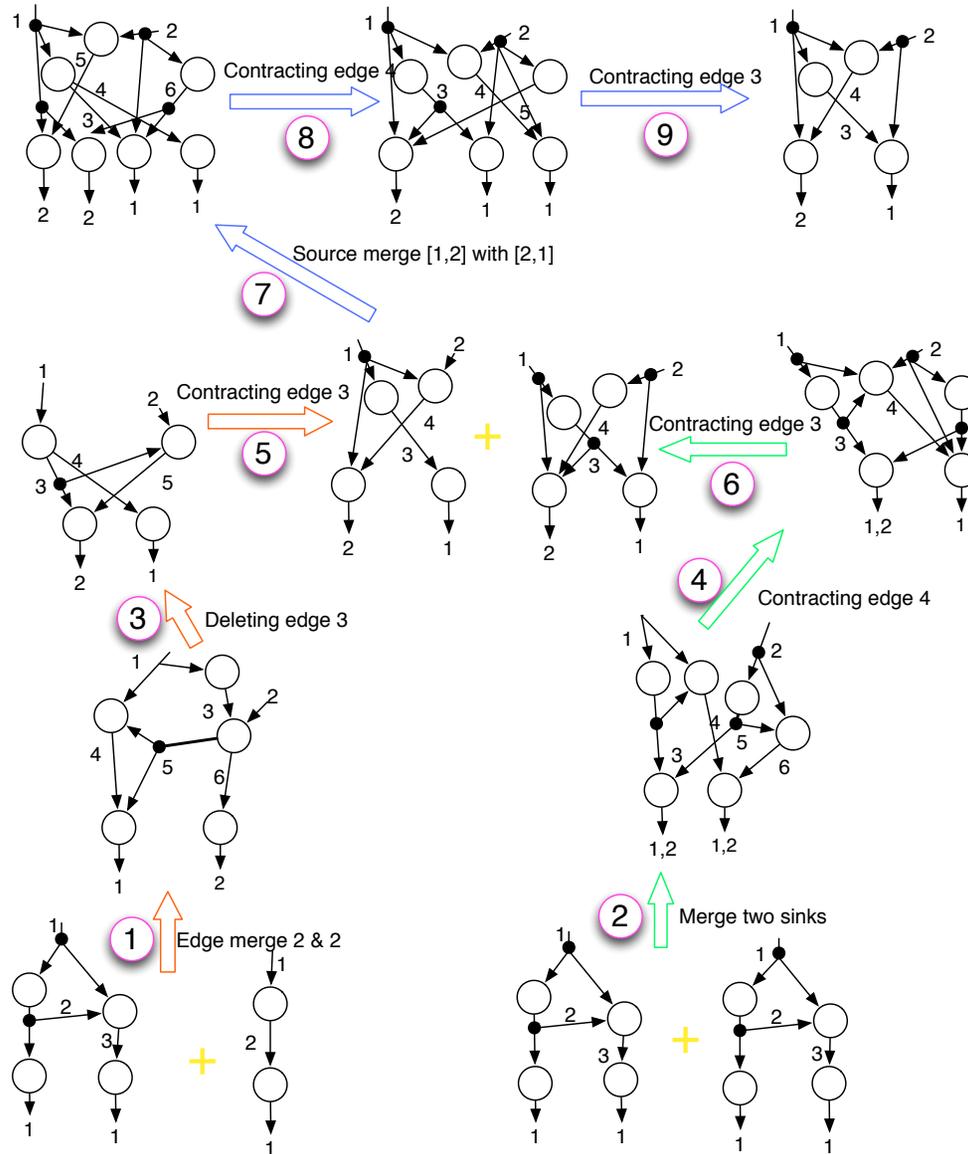


$$\begin{array}{l}
 R_i \geq H(X_2), i = 1, 2 \\
 R_1 + R_2 \geq H(X_1) + H(X_2) \\
 R_i \geq H(X_4), i = 3, 4, 4' \\
 R_3 + R_i \geq H(X_3) + H(X_4), i = 4, 4' \\
 R_3 + R_i + R_5 \geq H(X_3) + 2H(X_4), i = 4, 4' \\
 R_6 \geq H(X_6) \\
 R_i + R_8 \geq H(X_6), i = 7, 7' \\
 R_6 + R_i \geq H(X_5) + H(X_6), i = 7, 7' \\
 R_6 + R_i + 2R_8 \geq H(X_5) + 2H(X_6), i = 7, 7' \\
 R_9 \geq H(X_3) \\
 R_{10} \geq \sum_{i=1}^3 H(X_i) \\
 R_{11} \geq H(X_5) \\
 R_{12} \geq H(X_4) + H(X_5) \\
 R_{10'} + R_{11} + R_{12} \geq H(X_4) + 2H(X_5) + H(X_6)
 \end{array}$$

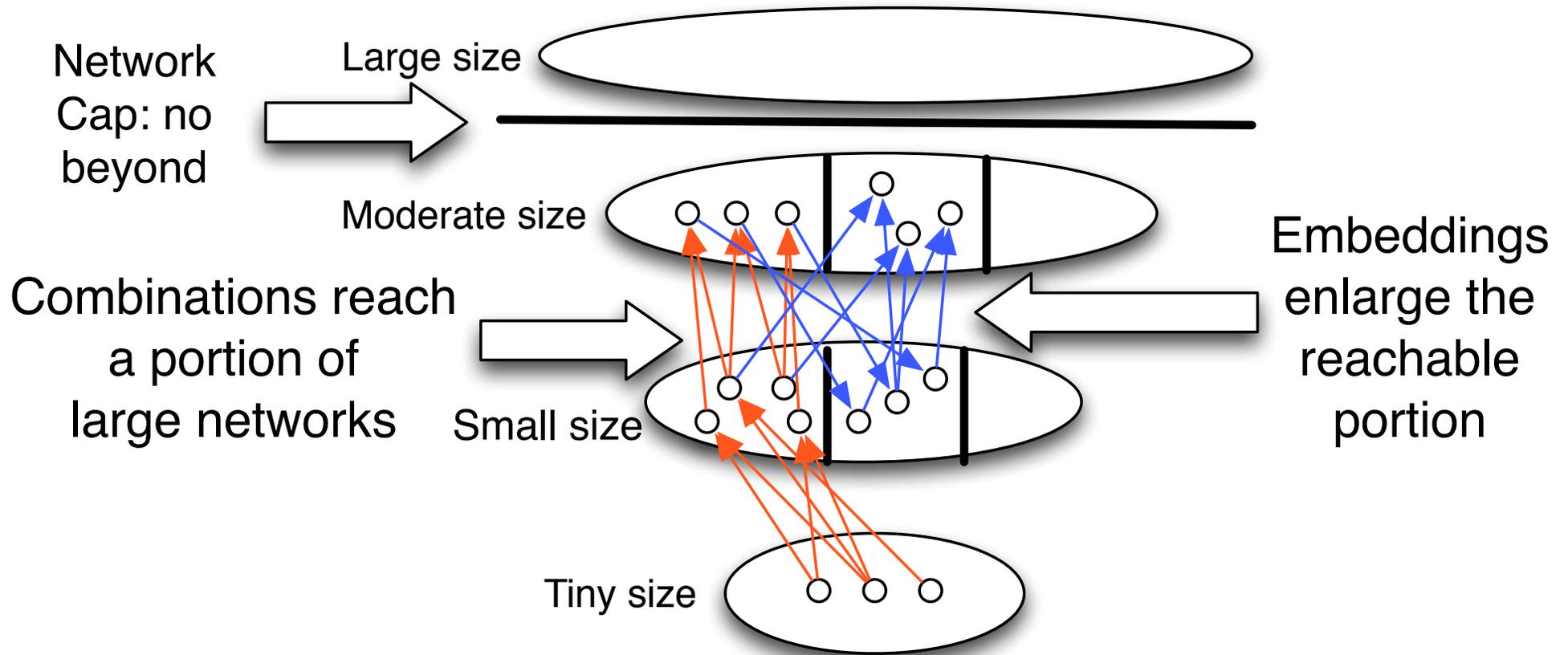


$$\begin{array}{l}
 R_9 \geq H(X_3') \\
 R_{10} \geq H(X_1') + H(X_2') + H(X_3') \\
 R_{10'} + R_{11} + R_{12} \geq H(X_4') + 2H(X_5') + H(X_6')
 \end{array}$$

Computationally Enabled Research Agenda – Hierarchy: Operator Concatenation



Computationally Enabled Research Agenda – Hierarchy: Operator Concatenation



Use operators *together* to get RR for big networks. Partial Network Closure.

Computationally Enabled Research Agenda – Hierarchy: Operator Concatenation

Start with the single (1, 1), single (2, 1), and the four (1, 2) networks; These 6 tiny networks can generate new 11635 networks w/ small cap!

size\cap	combination operators only			embedding and combinations		
	(3,3)	(3,4)	(4,4)	(3,3)	(3,4)	(4,4)
(1,3)	4	4	4	4	4	4
(1,4)	0	10	10	0	10	10
(2,2)	3	3	3	8	15	16
(2,3)	13	16	16	30	131	155
(2,4)	0	97	101	0	516	648
(3,2)	2	3	2	4	10	11
(3,3)	24	24	24	42	353	833
(3,4)	0	135	135	0	2361	5481
(4,2)	0	0	3	0	0	3
(4,3)	0	0	17	0	0	44
(4,4)	0	0	253	0	0	4430
all	46	292	568	88	3400	11635

Computationally Enabled Research Agenda – Hierarchy: Operator Concatenation

With the increase of cap size, number of new networks increases!

size\cap	combination operators only			embedding and combinations		
	(3,3)	(3,4)	(4,4)	(3,3)	(3,4)	(4,4)
(1,3)	4	4	4	4	4	4
(1,4)	0	10	10	0	10	10
(2,2)	3	3	3	8	15	16
(2,3)	13	16	16	30	131	155
(2,4)	0	97	101	0	516	648
(3,2)	2	3	2	4	10	11
(3,3)	24	24	24	42	353	833
(3,4)	0	135	135	0	2361	5481
(4,2)	0	0	3	0	0	3
(4,3)	0	0	17	0	0	44
(4,4)	0	0	253	0	0	4430
all	46	292	568	88	3400	11635

Computationally Enabled Research Agenda – Hierarchy: Operator Concatenation

Embedding operations are important in the process!

size\cap	combination operators only			embedding and combinations		
	(3,3)	(3,4)	(4,4)	(3,3)	(3,4)	(4,4)
(1,3)	4	4	4	4	4	4
(1,4)	0	10	10	0	10	10
(2,2)	3	3	3	8	15	16
(2,3)	13	16	16	30	131	155
(2,4)	0	97	101	0	516	648
(3,2)	2	3	2	4	10	11
(3,3)	24	24	24	42	353	833
(3,4)	0	135	135	0	2361	5481
(4,2)	0	0	3	0	0	3
(4,3)	0	0	17	0	0	44
(4,4)	0	0	253	0	0	4430
all	46	292	568	88	3400	11635

Come see me for more detailed slides about:

Computationally Enabled Research Agenda:

1. Train a computer to calculate network coding capacity regions and their proofs.
2. Build a database of all network coding capacity regions up to a certain size.
3. Organize this database to learn from it, and then to use it to create solutions to networks too large for the computer calculate directly.

Come see me for more detailed slides about:

***View these as listing discrete objects that
are inequivalent under symmetry.
Study that notion of symmetry.***

Computationally Enabled Research Agenda:

1. Train a computer to calculate network coding capacity regions and their proofs.
2. Build a database of all network coding capacity regions up to a certain size.
3. Organize this database to learn from it, and then to use it to create solutions to networks too large for the computer calculate directly.

Come see me for more detailed slides about:

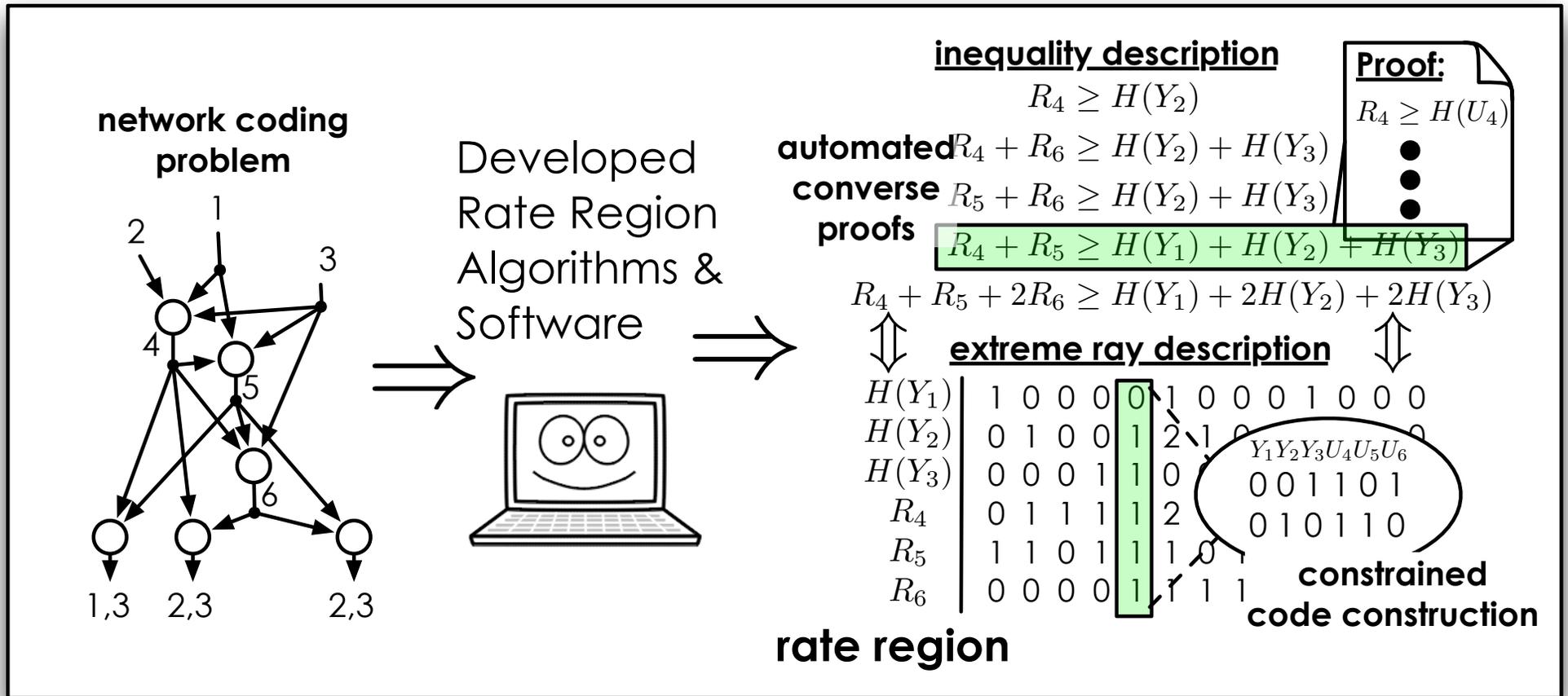
View these as listing discrete objects that are inequivalent under symmetry. Study that notion of symmetry.

Computationally Enabled Research Agenda:

1. Train a computer to calculate network coding capacity regions and their proofs.
2. Build a database of all network coding capacity regions up to a certain size.
3. Organize this database to learn from it, and then to use it to create solutions to networks too large for the computer calculate directly.

Propose a new rate region combination operator for connecting multiple sinks to multiple sources based on common information

What is a network coding capacity region?



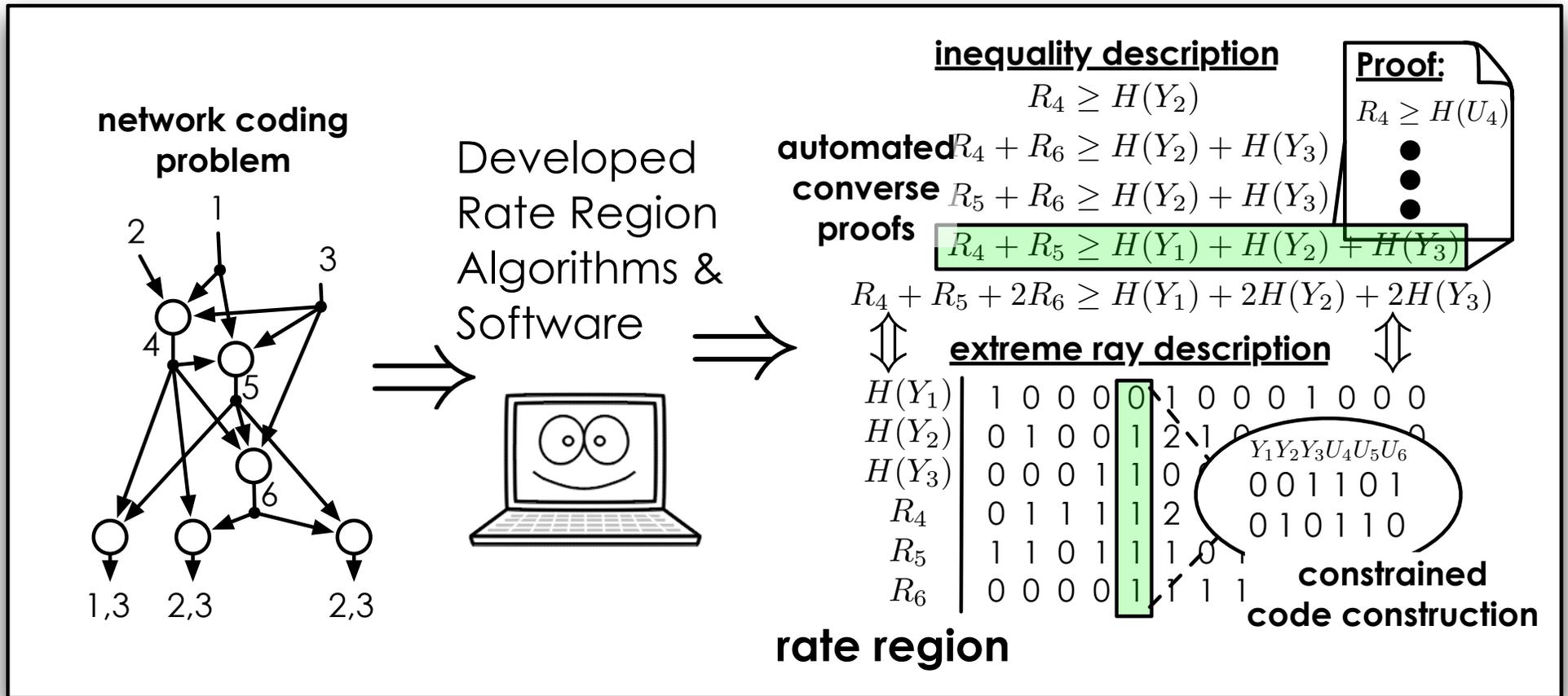
Substitutes outer/inner bounds to Γ_N^* into Yan, Yeung, Zhang '12

$$\mathcal{R}_* = \text{proj}_{R_{\mathcal{E}}, [H(Y_s) | s \in \mathcal{S}]} (\overline{\text{con}(\Gamma_N^* \cap \mathcal{L}_{123}) \cap \mathcal{L}_{45}}) \quad (7)$$

where $\mathcal{L}_{123} := \left\{ \mathbf{h} \mid h_{Y_s} = \sum_{s \in \mathcal{S}} h_{Y_s}, h_{X_{\text{Out}(i)} | X_{\text{In}(i)}} = 0 \right\}$ and

$$\mathcal{L}_{45} := \left\{ (\mathbf{h}^T, \mathbf{R}^T)^T \in \mathbb{R}_+^{2^N - 1 + |\mathcal{E}|} : R_e \geq h_{U_e}, e \in \mathcal{E}, h_{Y_{\beta(t)} | U_{\text{In}(t)}} = 0 \right\}$$

What is a network coding capacity region?



Substitutes outer/inner bounds to Γ_N^* into Yan, Yeung, Zhang '12

$$\mathcal{R}_* = \text{proj}_{R_{\mathcal{E}}, [H(Y_s) | s \in \mathcal{S}]} (\overline{\text{con}(\Gamma_N^* \cap \mathcal{L}_{123}) \cap \mathcal{L}_{45}}) \quad (8)$$

where $\mathcal{L}_{123} := \left\{ \mathbf{h} \mid h_{Y_s} = \sum_{s \in \mathcal{S}} h_{Y_s}, h_{X_{\text{Out}(i)} | X_{\text{In}(i)}} = 0 \right\}$ and

$$\mathcal{L}_{45} := \left\{ (\mathbf{h}^T, \mathbf{R}^T)^T \in \mathbb{R}_+^{2^N - 1 + |\mathcal{E}|} : R_e \geq h_{U_e}, e \in \mathcal{E}, h_{Y_{\beta(t)} | U_{\text{In}(t)}} = 0 \right\}$$

Come see me for more detailed slides about:

*View these as listing discrete objects that
are inequivalent under symmetry.
Study that notion of symmetry.*

Computationally Enabled Research Agenda:

1. Train a computer to calculate network coding capacity regions and their proofs.
2. Build a database of all network coding capacity regions up to a certain size.
3. Organize this database to learn from it, and then to use it to create solutions to networks too large for the computer calculate directly.

Notions of Symmetry – Formalize via Groups

1. Symmetries of $\Gamma_N \cap \mathcal{L}_A$ where $\mathcal{L}_A = \mathcal{L}_{123} \cap \mathcal{L}_{45}$
 - (a) Symmetries of polyhedral cones
 - (b) Symmetries of Γ_N
 - (c) Symmetries of $\Gamma_N \cap \mathcal{L}_A$
 - (d) Application – reduces the complexity of proving the converse
2. Symmetries between different network coding problem instances
3. Symmetries among network codes
 - (a) Symmetries among linear codes
 - (b) Application to proving matched inner bound
 - (c) Symmetries among nonlinear codes

Our Related Journal Submissions & Software

- [1] Congduan Li, Steven Weber, and John Walsh, “On Multi-source Networks: Enumeration, Rate Region Computation, and Hierarchy,” *IEEE Trans. Inform. Theory*, vol. 63, no. 11, pp. 7283–7303, Nov. 2017. [Online]. Available: <https://doi.org/10.1109/TIT.2017.2745620>
- [2] Congduan Li, Steven Weber, and John MacLaren Walsh, “On Multilevel Diversity Coding Systems,” *IEEE Trans. Inform. Theory*, vol. 63, no. 1, pp. 230–251, Jan. 2017. [Online]. Available: <https://doi.org/10.1109/TIT.2016.2628791>
- [3] Jayant Apte and John MacLaren Walsh, “Explicit Polyhedral Bounds on Network Coding Rate Regions via Entropy Function Region: Algorithms, Symmetry, and Computation,” *IEEE Trans. Inform. Theory*, 2016, Submitted July 22, 2016, revised July 6, 2017. [Online]. Available: <http://arxiv.org/abs/1607.06833>
- [4] —, “Constrained Linear Representability of Polymatroids and Algorithms for Computing Achievability Proofs in Network Coding,” *IEEE Trans. Inform. Theory*, Submitted May 15, 2016. [Online]. Available: <http://arxiv.org/abs/1605.04598>
- [5] Yunshu Liu and John MacLaren Walsh, “Mapping the Region of Entropic Vectors with Support Enumeration & Information Geometry,” *IEEE Trans. Inform. Theory*, Submitted December 08, 2015. [Online]. Available: <http://arxiv.org/abs/1512.03324>
- [6] Congduan Li and John MacLaren Walsh, “Network Coding Rate Region Calculation.” [Online]. Available: <http://www.ece.drexel.edu/walsh/aspitrg/software.html>
- [7] —, “Network Enumeration and Hierarchy.” [Online]. Available: <http://www.ece.drexel.edu/walsh/aspitrg/software.html>
- [8] Jayant Apte and John MacLaren Walsh, “Information Theoretic Achievability Prover.” [Online]. Available: <https://github.com/jayant91089/itap>
- [9] —, “Information Theoretic Converse Prover.” [Online]. Available: <https://github.com/jayant91089/itcp>