Complexity of Games on Graphs

Václav Blažej

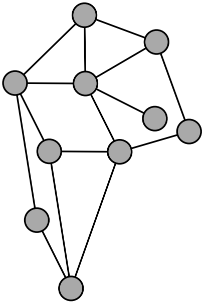
Department of Theoretical Computer Science
Faculty of Information Technology
Czech Technical University in Prague

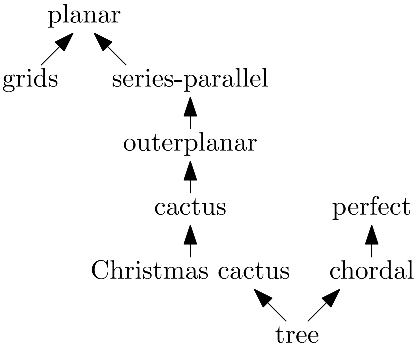
supervisor: doc. RNDr. Tomáš Valla, Ph.D.

29. September 2022

Complexity of Games on **Graphs**

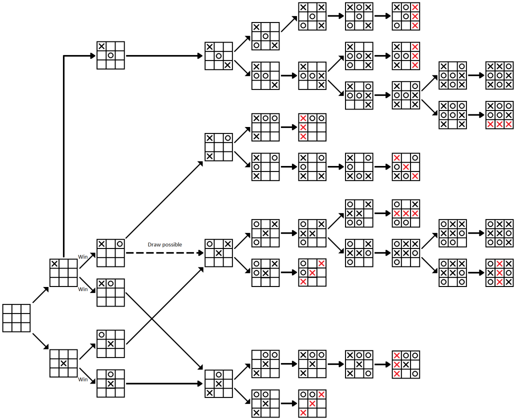
a graph consists of
vertices and edges





Complexity of **Games** on Graphs

- 2 players
- complete information
- no randomness
- play optimally

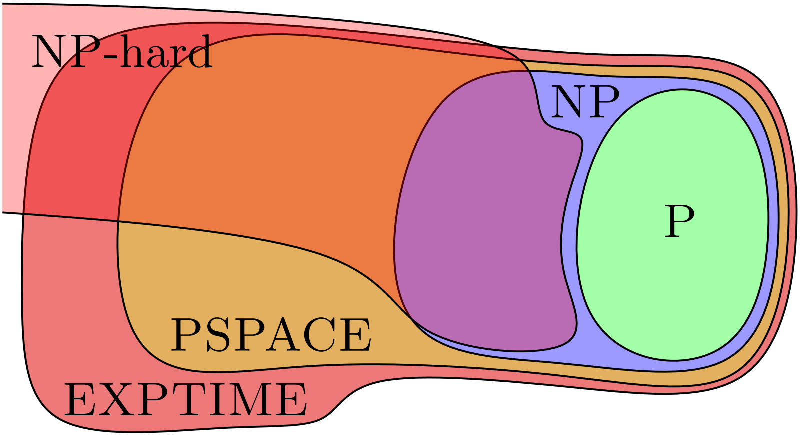


source: Wikimedia commons

Computational Complexity of Games on Graphs

Tractable: algorithm running in polynomial time (class P)

Intractable: under common assumptions, there is no algorithm that runs in polynomial time (class NP-hard)



Contents of the thesis

1. m-Eternal domination

   with Jan Matyáš Křišťan, and Tomáš Valla; in *Reachability Problems - 13th International Conference, RP 2019*

2. Hat Chromatic Number

   with Pavel Dvořák, and Michal Opler; in *Graph-Theoretic Concepts in Computer Science - 47th International Workshop, WG 2021*

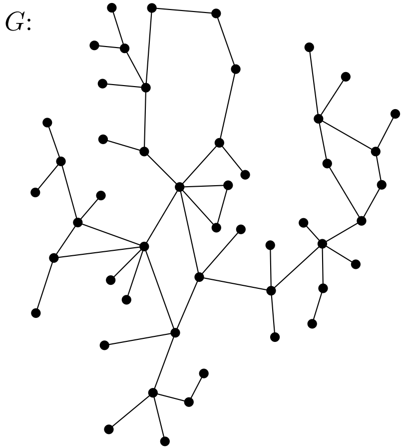
3. Online Ramsey Number

   with Pavel Dvořák, and Tomáš Valla; in *Computer Science - Theory and Applications - 14th International Computer Science Symposium in Russia, CSR 2019*

4. Group Identification

   with Dušan Knop, and Šimon Schierreich; in *Computer Science - Theory and Applications - 36th Conference on Artificial Intelligence, AAAI 2022, (student abstract, to appear)*.

Game setting



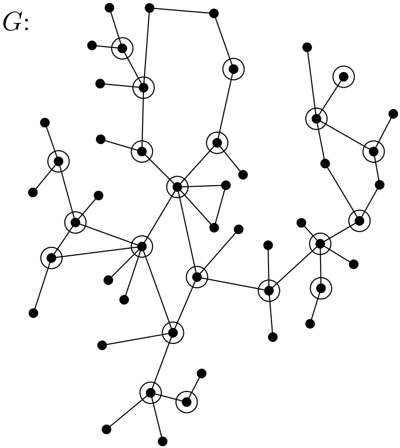
Given a graph G, you place k guards on its vertices.

Now, we perform 1 turn:

1. I will attack a vertex.
2. You may move each guard along at most one edge.
3. You defended my attack if a guard stands on the attacked vertex.

You win if you successfully defend.
I win if you did not defend.

Game setting



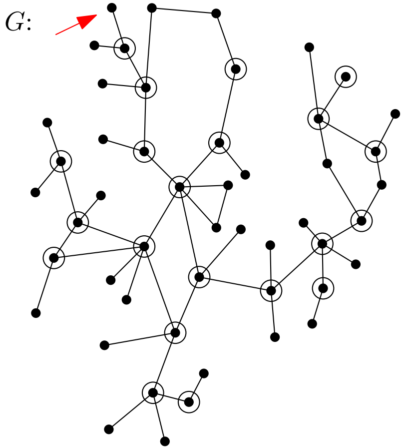
Given a graph G, you place k guards on its vertices.

Now, we perform 1 turn:

1. I will attack a vertex.
2. You may move each guard along at most one edge.
3. You defended my attack if a guard stands on the attacked vertex.

You win if you successfully defend.
I win if you did not defend.

Game setting



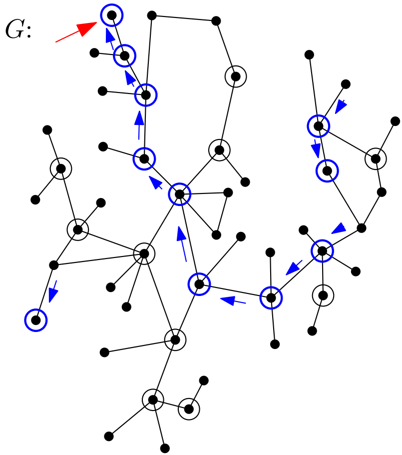
Given a graph G, you place k guards on its vertices.

Now, we perform 1 turn:

1. I will attack a vertex.
2. You may move each guard along at most one edge.
3. You defended my attack if a guard stands on the attacked vertex.

You win if you successfully defend.
I win if you did not defend.

Game setting



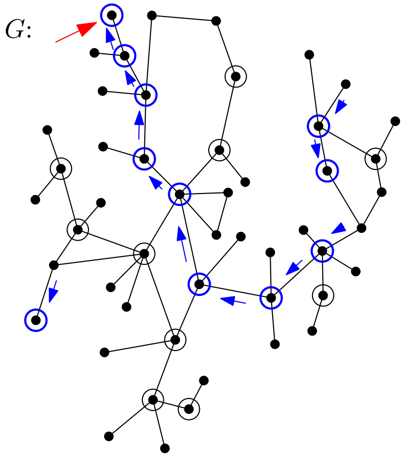
Given a graph G, you place k guards on its vertices.

Now, we perform 1 turn:

1. I will attack a vertex.
2. You may move each guard along at most one edge.
3. You defended my attack if a guard stands on the attacked vertex.

You win if you successfully defend.
I win if you did not defend.

Game setting



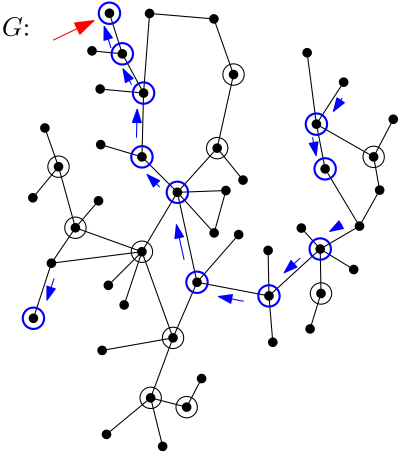
Given a graph G, you place k guards on its vertices.

Now, we perform 1 turn:

1. I will attack a vertex.
2. You may move each guard along at most one edge.
3. You defended my attack if a guard stands on the attacked vertex.

You win if you successfully defend.
I win if you did not defend.

Game setting



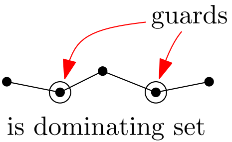
Given a graph G, you place k guards on its vertices.

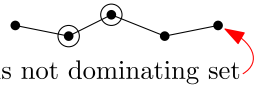
Now, we perform 1 turn:

1. I will attack a vertex.
2. You may move each guard along at most one edge.
3. You defended my attack if a guard stands on the attacked vertex.

You win if you successfully defend.
I win if you did not defend.

Dominating set





Game with 1 turn ≈ Dom. number

I pick a vertex and you have to defend it (step on it) by moving each guard along at most one edge.

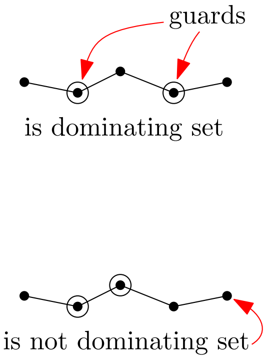
Dominating set

Set of guards $D \subseteq V(G)$ such that each vertex of the graph G either is in D or in its neighborhood.

Domination number

Smallest possible size of the set D.

Dominating set



Game with 1 turn ≈ Dom. number

I pick a vertex and you have to defend it (step on it) by moving each guard along at most one edge.

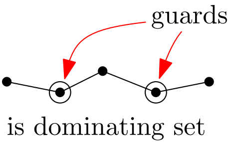
Dominating set

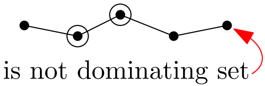
Set of guards $D \subseteq V(G)$ such that each vertex of the graph G either is in D or in its neighborhood.

Domination number

Smallest possible size of the set D.

Dominating set





Game with 1 turn ≈ Dom. number

I pick a vertex and you have to defend it (step on it) by moving each guard along at most one edge.

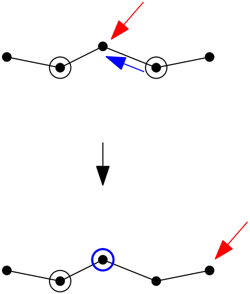
Dominating set

Set of guards $D \subseteq V(G)$ such that each vertex of the graph G either is in D or in its neighborhood.

Domination number

Smallest possible size of the set D.

Dominating set



Game with 1 turn ≈ Dom. number

I pick a vertex and you have to defend it (step on it) by moving each guard along at most one edge.

Dominating set

Set of guards $D \subseteq V(G)$ such that each vertex of the graph G either is in D or in its neighborhood.

Domination number

Smallest possible size of the set D.

Eternal Domination Number

Game of Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge.

Game of Eternal Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge, for an **infinite** number of turns.

What is the minimum number of guards you need to *eternally* defend the graph against an arbitrary sequence of attacks.

Eternal Domination Number

Game of Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge.

Game of Eternal Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge, for an **infinite** number of turns.

What is the minimum number of guards you need to *eternally* defend the graph against an arbitrary sequence of attacks.

Eternal Domination Number

Game of Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge.

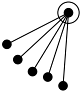
Game of Eternal Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge, for an **infinite** number of turns.

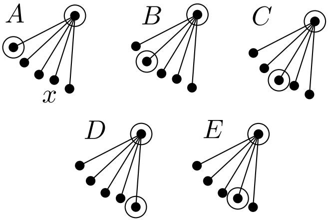
What is the minimum number of guards you need to *eternally* defend the graph against an arbitrary sequence of attacks.

Eternal Domination Number

dominating set



eternal domiation configurations



Game of Domination

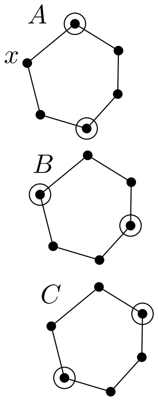
I pick a vertex and you have to defend it by moving each guard along at most one edge.

Game of Eternal Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge, for an **infinite** number of turns.

What is the minimum number of guards you need to *eternally* defend the graph against an arbitrary sequence of attacks.

Eternal Domination Number



Game of Domination

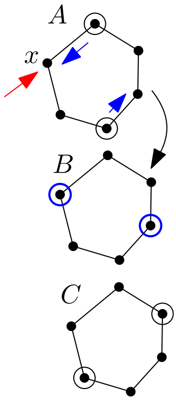
I pick a vertex and you have to defend it by moving each guard along at most one edge.

Game of Eternal Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge, for an **infinite** number of turns.

What is the minimum number of guards you need to *eternally* defend the graph against an arbitrary sequence of attacks.

Eternal Domination Number



Game of Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge.

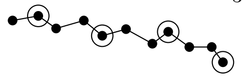
Game of Eternal Domination

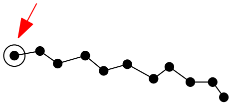
I pick a vertex and you have to defend it by moving each guard along at most one edge, for an **infinite** number of turns.

What is the minimum number of guards you need to *eternally* defend the graph against an arbitrary sequence of attacks.

Eternal Domination Number

$$|\text{ dominating set }| = \lceil \tfrac{n}{3} \rceil$$





Game of Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge.

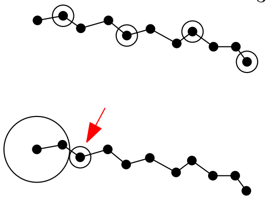
Game of Eternal Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge, for an **infinite** number of turns.

What is the minimum number of guards you need to *eternally* defend the graph against an arbitrary sequence of attacks.

Eternal Domination Number

$|\text{ dominating set }| = \lceil \frac{n}{3} \rceil$



Game of Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge.

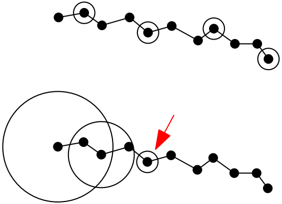
Game of Eternal Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge, for an **infinite** number of turns.

What is the minimum number of guards you need to *eternally* defend the graph against an arbitrary sequence of attacks.

Eternal Domination Number

$$|\text{ dominating set }| = \lceil \tfrac{n}{3} \rceil$$



Game of Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge.

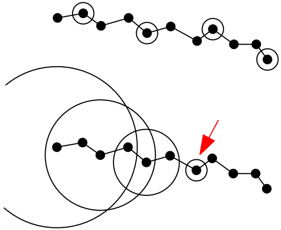
Game of Eternal Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge, for an **infinite** number of turns.

What is the minimum number of guards you need to *eternally* defend the graph against an arbitrary sequence of attacks.

Eternal Domination Number

$$|\text{ dominating set }| = \lceil \tfrac{n}{3} \rceil$$



Game of Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge.

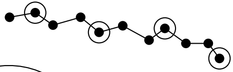
Game of Eternal Domination

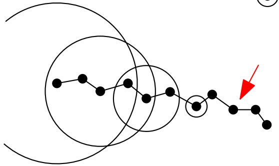
I pick a vertex and you have to defend it by moving each guard along at most one edge, for an **infinite** number of turns.

What is the minimum number of guards you need to *eternally* defend the graph against an arbitrary sequence of attacks.

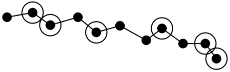
Eternal Domination Number

$|\text{ dominating set }| = \lceil \frac{n}{3} \rceil$





$|\text{ eternal dom. set }| = \lceil \frac{n}{2} \rceil$



Game of Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge.

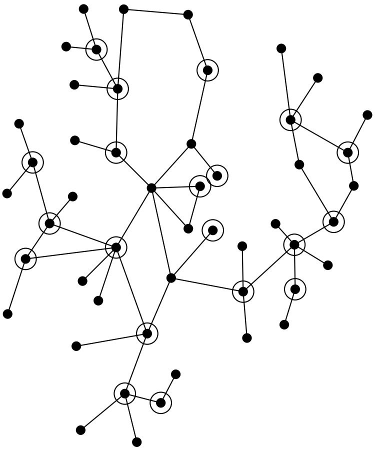
Game of Eternal Domination

I pick a vertex and you have to defend it by moving each guard along at most one edge, for an **infinite** number of turns.

What is the minimum number of guards you need to *eternally* defend the graph against an arbitrary sequence of attacks.

Relation to Domination number

domination $\leq$ eternal domination $\leq 2 \times$ domination



domination $\leq$ eternal domination

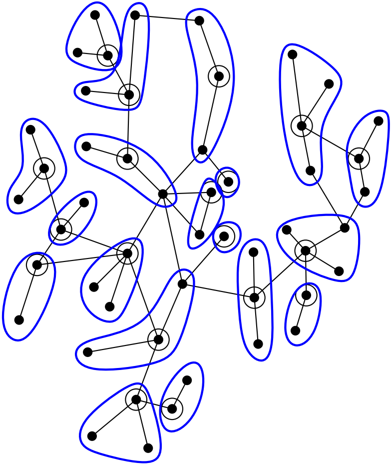
every configuration must form a dominating set

eternal domination $\leq 2 \times$ dom.

we can defend neighborhood of each vertex of the dominating set by a star-defending strategy

Relation to Domination number

domination $\leq$ eternal domination $\leq 2 \times$ domination



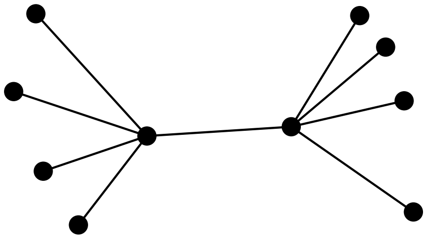
domination $\leq$ eternal domination

every configuration must form a dominating set

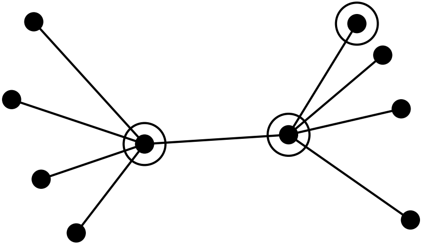
eternal domination $\leq 2 \times$ dom.

we can defend neighborhood of each vertex of the dominating set by a star-defending strategy

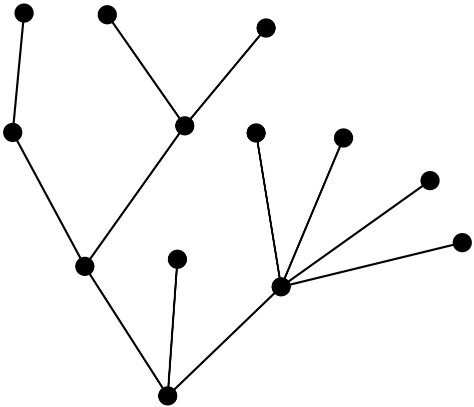
Defending trees



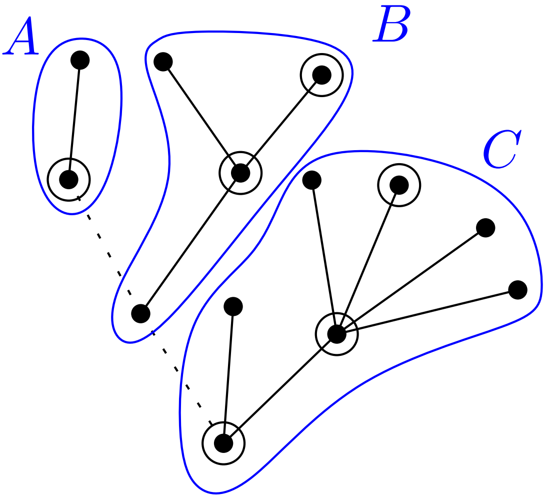
Defending trees



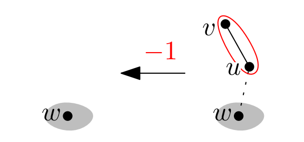
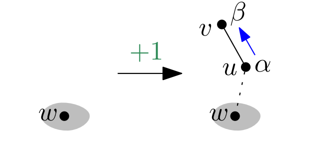
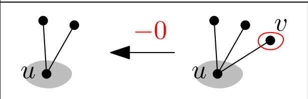
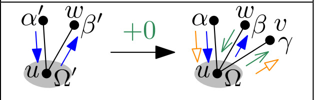
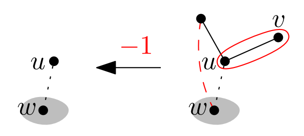
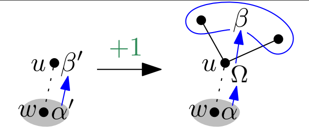
Defending trees



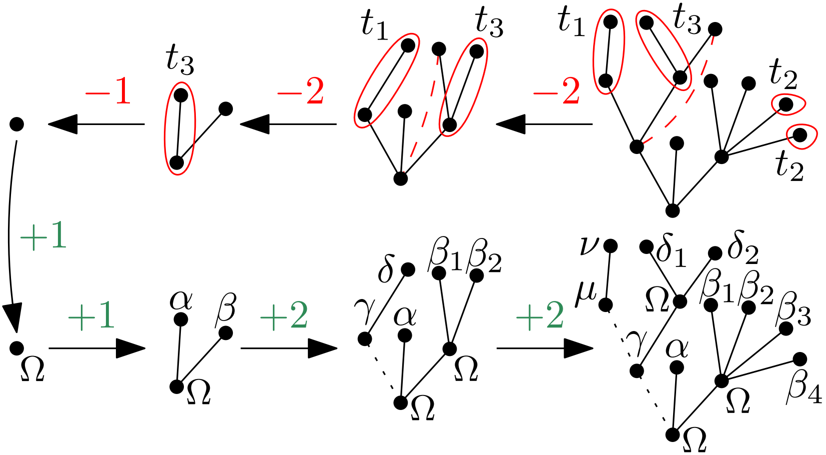
Defending trees



Defending trees

Reduction	Lower bound	Upper bound
t_1		
t_2		
t_3		

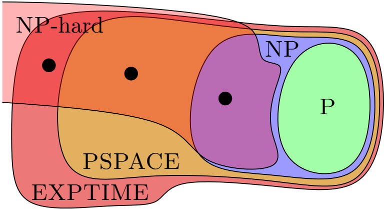
Defending trees



Decision variant of the problem

m-Eternal Domination - decision variant:

Input: Graph G, integer k
Output: Can k guards defend G against any sequence of attacks?



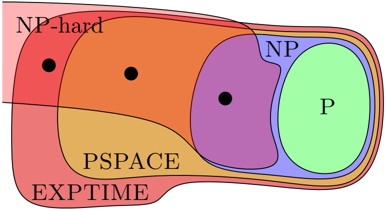
- Known to be NP-hard,
- lies in EXPTIME,
- unknown whether it lies in PSPACE.

We denote the minimum k which results in yes instance as $\gamma_{\mathrm{m}}^{\infty}(G)$.

Decision variant of the problem

m-Eternal Domination - decision variant:

Input: Graph G, integer k
Output: Can k guards defend G against any sequence of attacks?



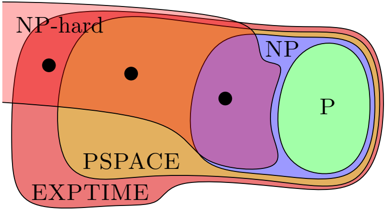
- Known to be NP-hard,
- lies in EXPTIME,
- unknown whether it lies in PSPACE.

We denote the minimum k which results in yes instance as $\gamma_{\mathrm{m}}^{\infty}(G)$.

Decision variant of the problem

m-Eternal Domination - decision variant:

Input: Graph G, integer k
Output: Can k guards defend G against any sequence of attacks?



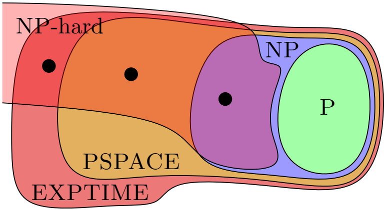
- Known to be NP-hard,
- lies in EXPTIME,
- unknown whether it lies in PSPACE.

We denote the minimum k which results in yes instance as $\gamma_{\mathrm{m}}^{\infty}(G)$.

Decision variant of the problem

m-Eternal Domination - decision variant:

Input: Graph G, integer k
Output: Can k guards defend G against any sequence of attacks?



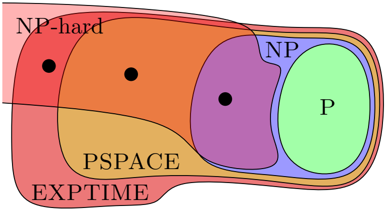
- Known to be NP-hard,
- lies in EXPTIME,
- unknown whether it lies in PSPACE.

We denote the minimum k which results in yes instance as $\gamma_{\mathrm{m}}^{\infty}(G)$.

Decision variant of the problem

m-Eternal Domination - decision variant:

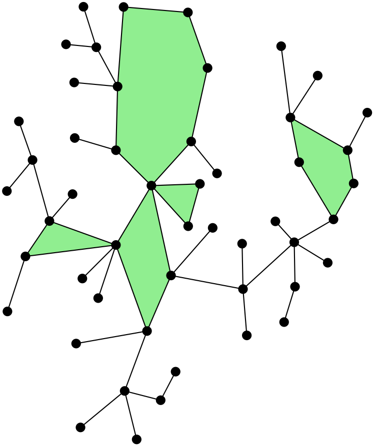
Input: Graph G, integer k
Output: Can k guards defend G against any sequence of attacks?



- Known to be NP-hard,
- lies in EXPTIME,
- unknown whether it lies in PSPACE.

We denote the minimum k which results in yes instance as $\gamma_{\mathrm{m}}^{\infty}(G)$.

Cactus graphs



Definition

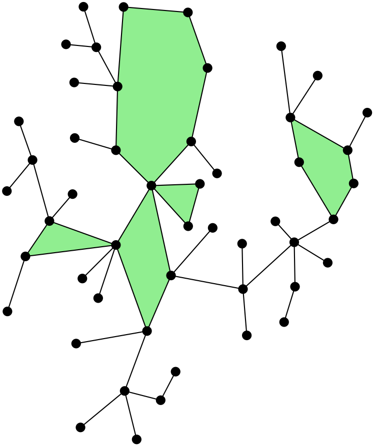
Graph is **cactus graph** when every edge belongs to at most one cycle.

Cactus graphs are characterized by one forbidden minor: $(K_4 \setminus e)$.

Theorem (B., Křišťan, Valla)

Let G be a cactus graph. Then there exists a polynomial algorithm which computes the minimum required number of guards to eternally defend G.

Cactus graphs



Definition

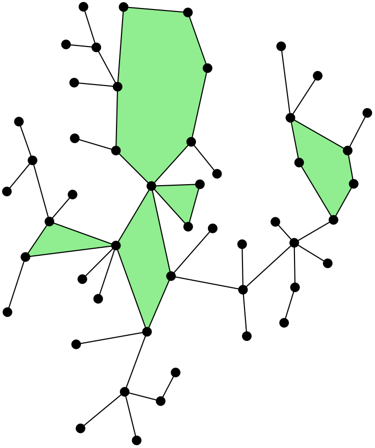
Graph is **cactus graph** when every edge belongs to at most one cycle.

Cactus graphs are characterized by one forbidden minor: $(K_4 \setminus e)$.

Theorem (B., Křišťan, Valla)

Let G be a cactus graph. Then there exists a polynomial algorithm which computes the minimum required number of guards to eternally defend G.

Cactus graphs



Definition

Graph is **cactus graph** when every edge belongs to at most one cycle.

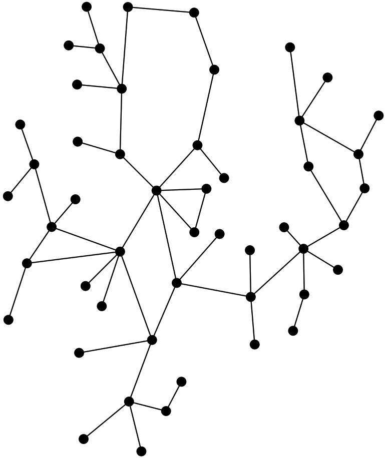
Cactus graphs are characterized by one forbidden minor: $(K_4 \setminus e)$.

Theorem (B., Křišťan, Valla)

Let G be a cactus graph. Then there exists a polynomial algorithm which computes the minimum required number of guards to eternally defend G.

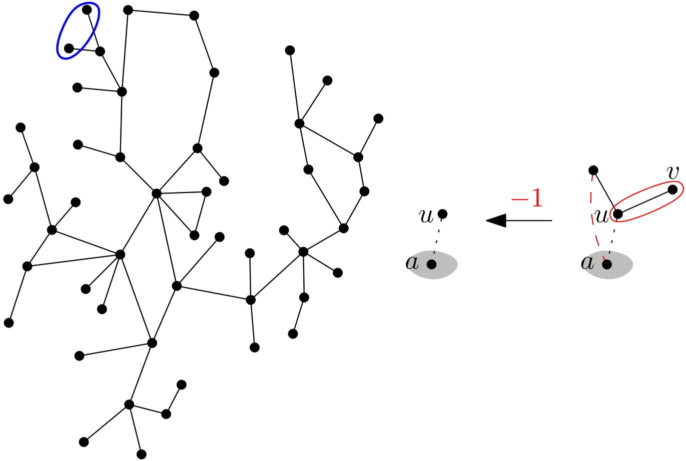
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



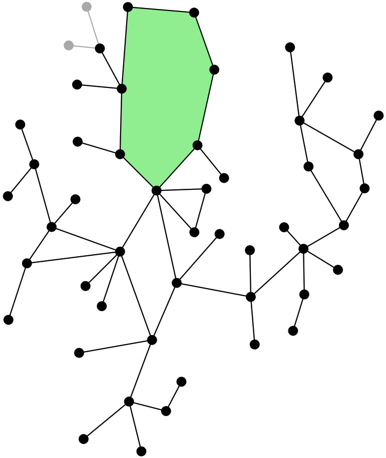
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



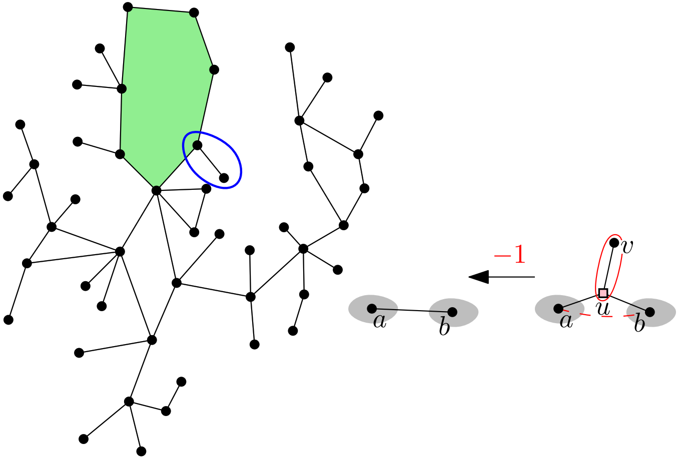
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



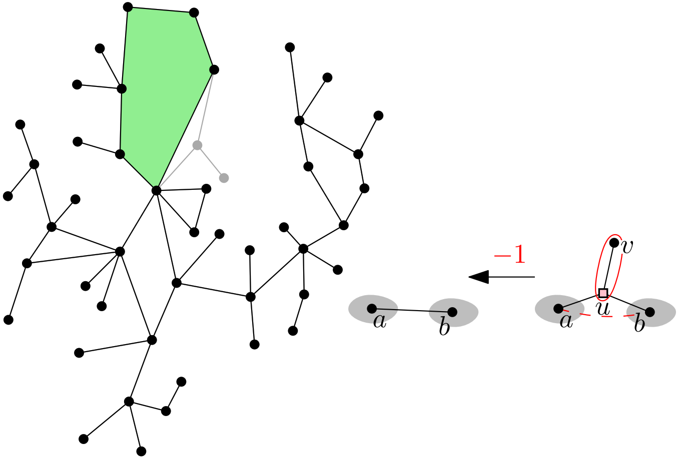
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



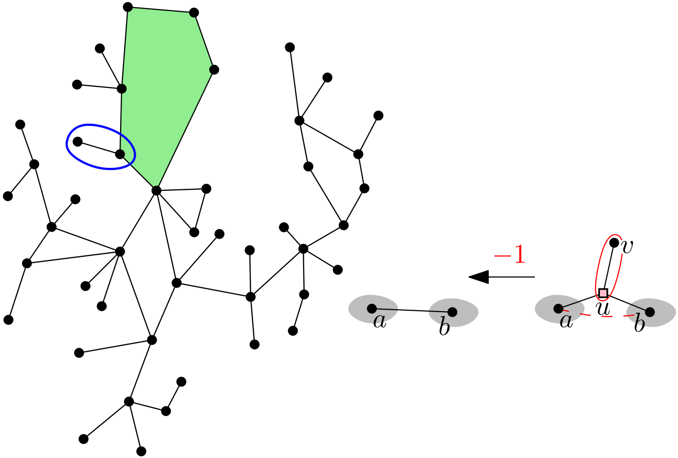
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



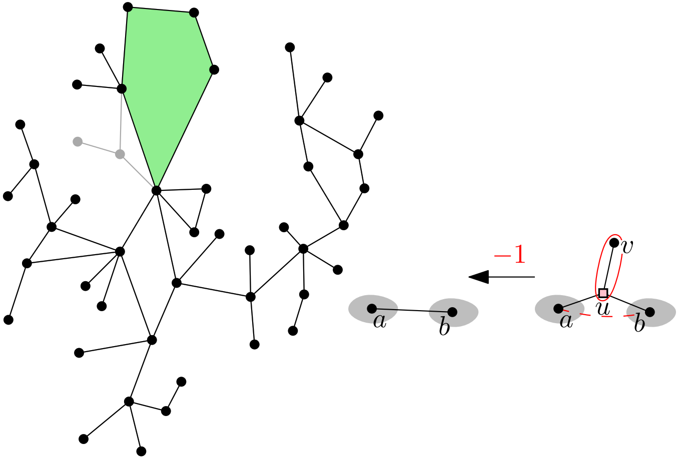
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



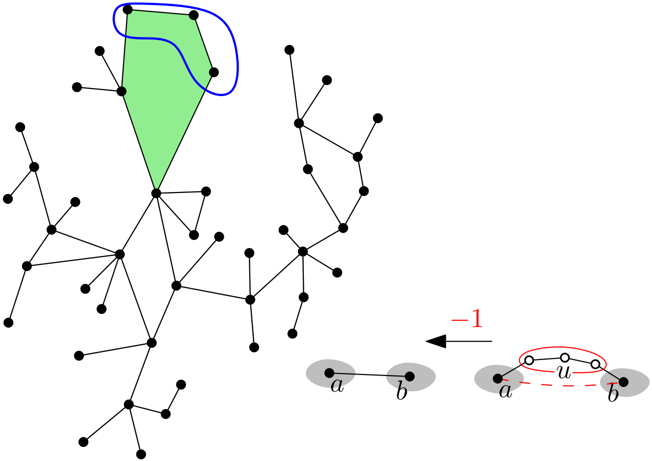
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



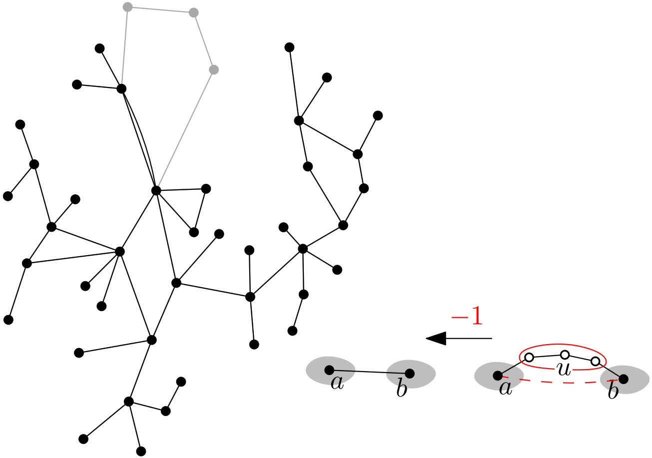
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



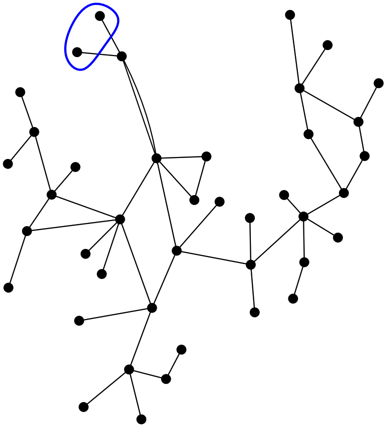
Reducing a cactus

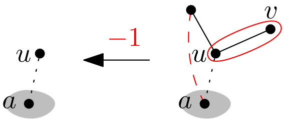
1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



Reducing a cactus

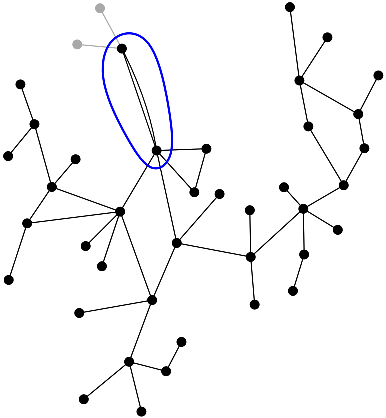
1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles

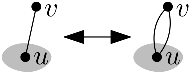




Reducing a cactus

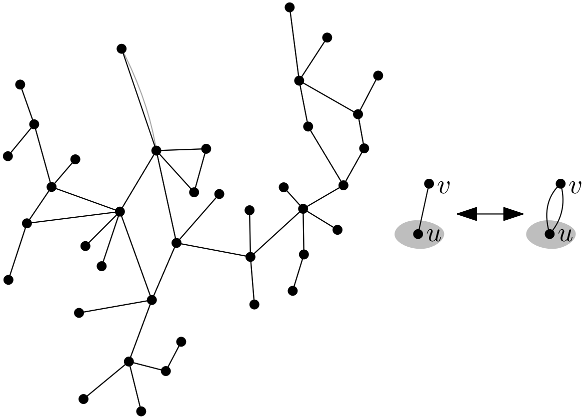
1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles





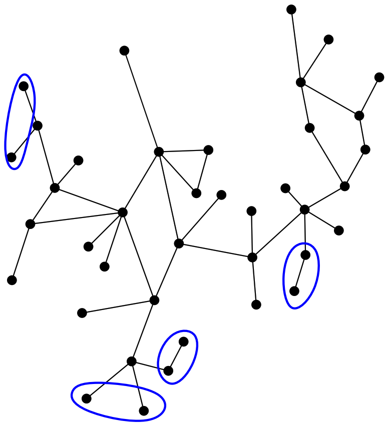
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



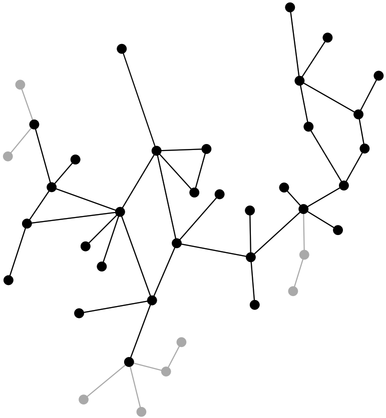
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



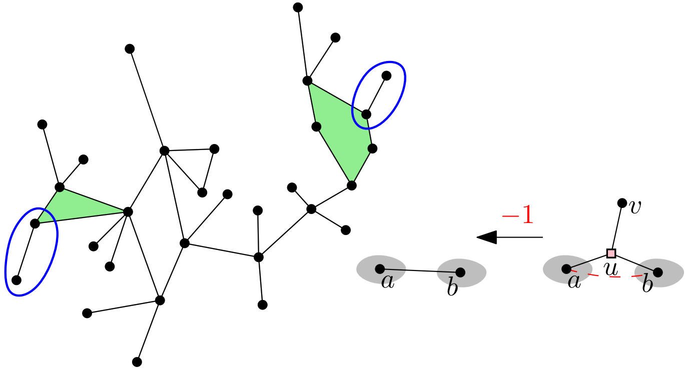
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



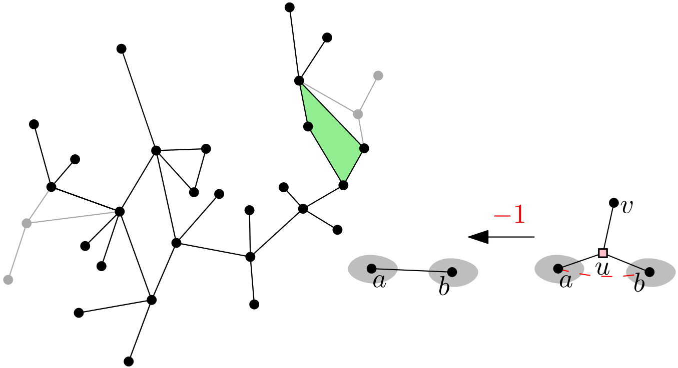
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



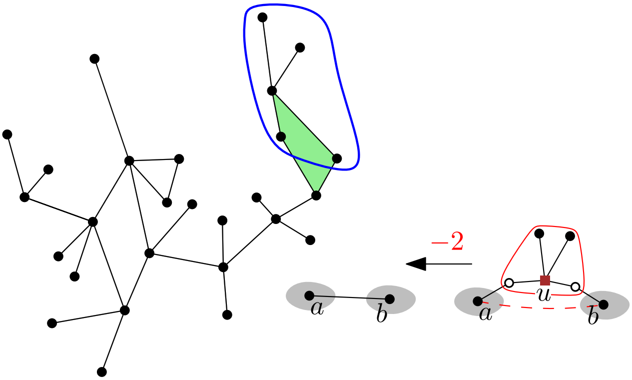
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



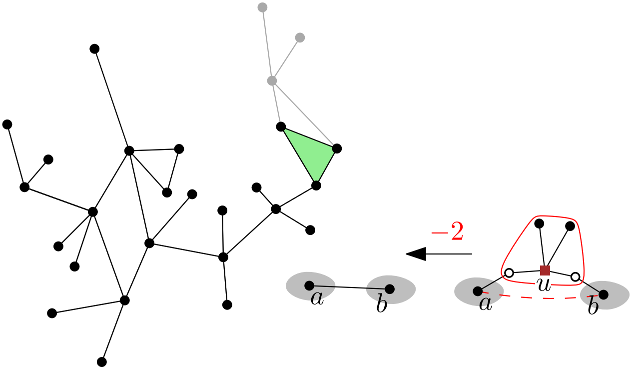
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



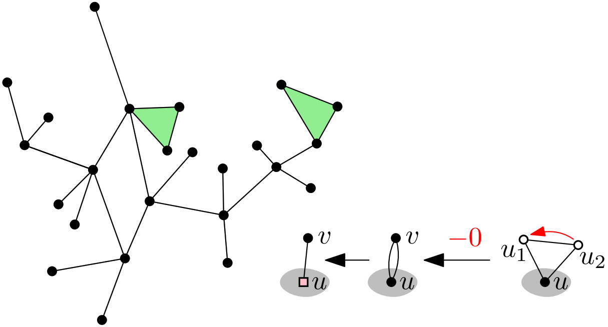
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



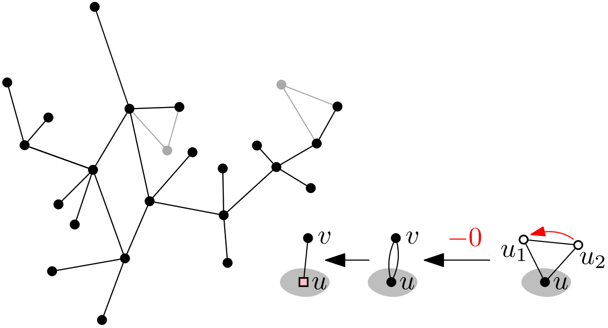
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



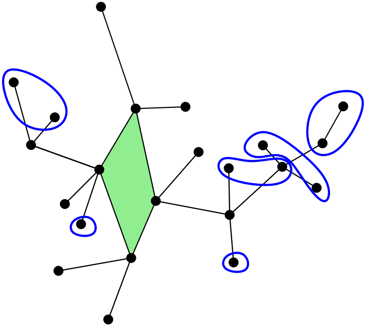
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



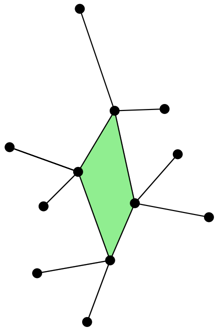
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



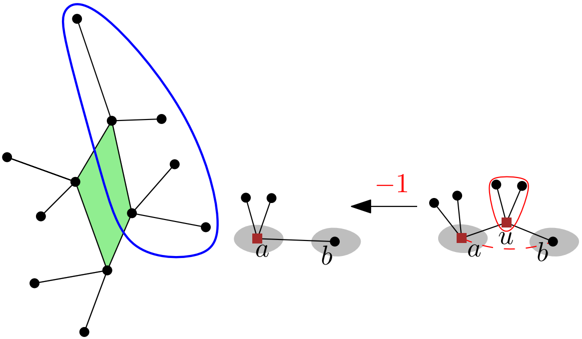
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



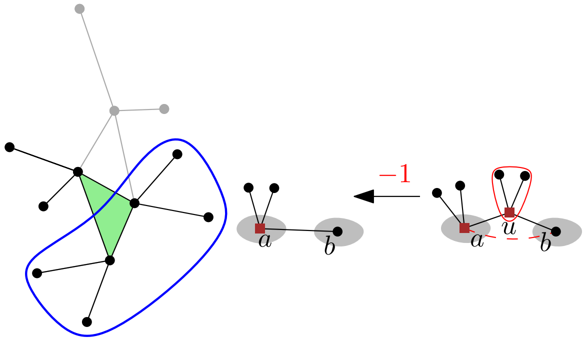
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



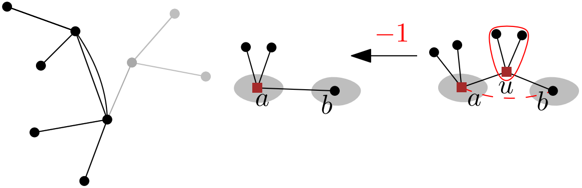
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



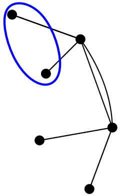
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



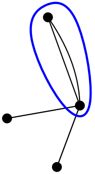
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



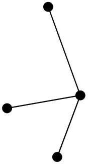
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



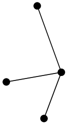
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles



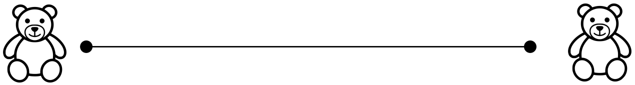
Reducing a cactus

1) reduce leaf trees, 2) shorten leaf cycles, 3) reduce cycles

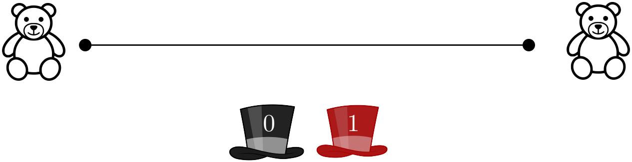


We have a polynomial algorithm that finds $\gamma_{\mathrm{m}}^{\infty}$ of any cactus graph.

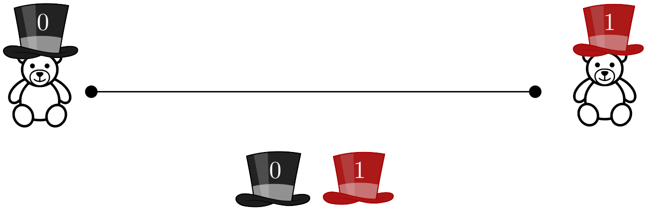
Hat Chromatic Number



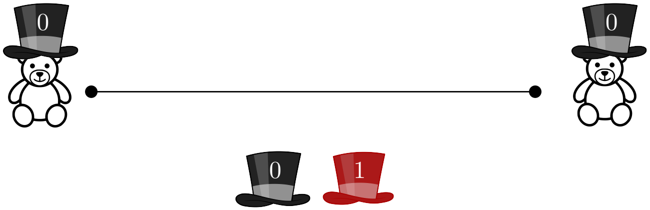
Hat Chromatic Number



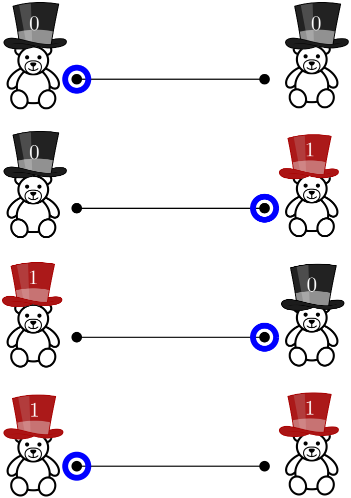
Hat Chromatic Number



Hat Chromatic Number



Hat Chromatic Number

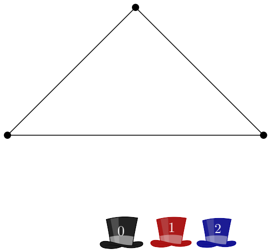


Players

Bears play against an evil *Demon*.

The game proceeds as:

1. Demon presents a graph G and number of colors k.
2. Bears can agree on a strategy. Then bears can no longer talk and are put on vertices of the graph.
3. Demon puts a colored hat on each bear's head.
4. All bears at once guess their hat colors based only on hat's colors of their neighbors.
5. Bears win if at least one bear guesses correctly.

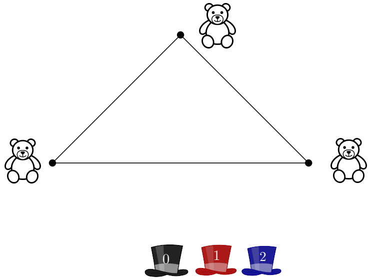


Players

Bears play against an evil *Demon*.

The game proceeds as:

1. Demon presents a graph G and number of colors k.
2. Bears can agree on a strategy. Then bears can no longer talk and are put on vertices of the graph.
3. Demon puts a colored hat on each bear's head.
4. All bears at once guess their hat colors based only on hat's colors of their neighbors.
5. Bears win if at least one bear guesses correctly.

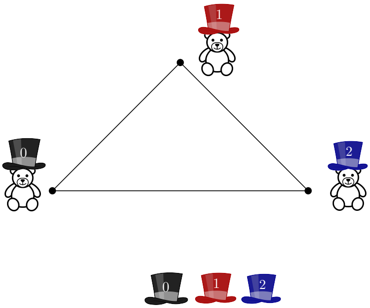


Players

Bears play against an evil *Demon*.

The game proceeds as:

1. Demon presents a graph G and number of colors k.
2. Bears can agree on a strategy. Then bears can no longer talk and are put on vertices of the graph.
3. Demon puts a colored hat on each bear's head.
4. All bears at once guess their hat colors based only on hat's colors of their neighbors.
5. Bears win if at least one bear guesses correctly.

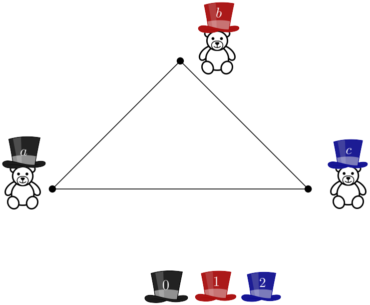


Players

Bears play against an evil *Demon*.

The game proceeds as:

1. Demon presents a graph G and number of colors k.
2. Bears can agree on a strategy. Then bears can no longer talk and are put on vertices of the graph.
3. Demon puts a colored hat on each bear's head.
4. All bears at once guess their hat colors based only on hat's colors of their neighbors.
5. Bears win if at least one bear guesses correctly.

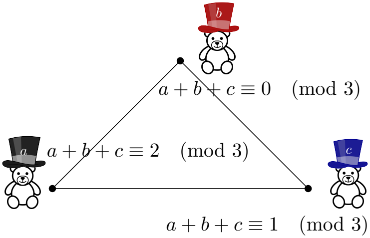


Players

Bears play against an evil *Demon*.

The game proceeds as:

1. Demon presents a graph G and number of colors k.
2. Bears can agree on a strategy. Then bears can no longer talk and are put on vertices of the graph.
3. Demon puts a colored hat on each bear's head.
4. All bears at once guess their hat colors based only on hat's colors of their neighbors.
5. Bears win if at least one bear guesses correctly.

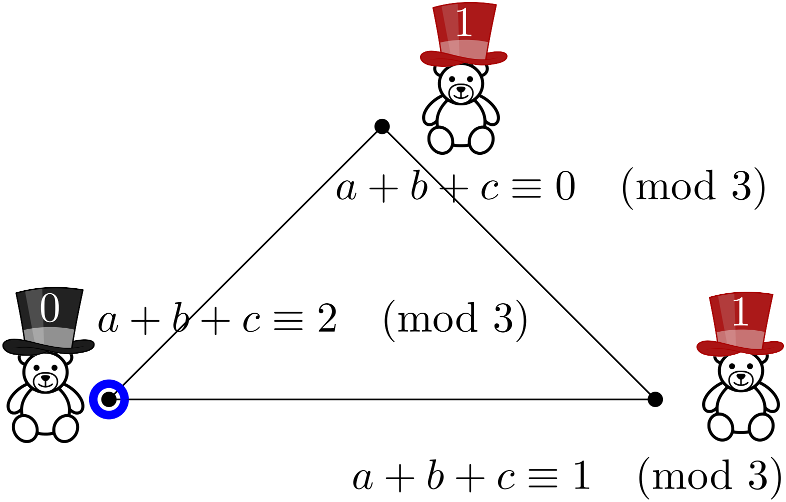


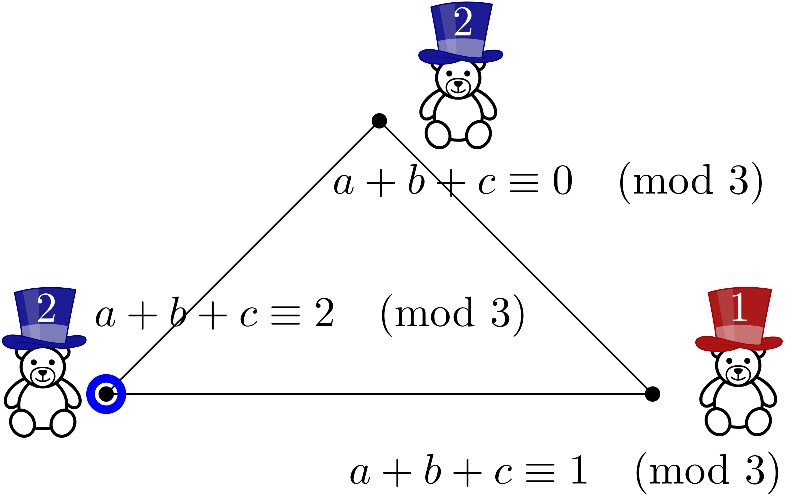
Players

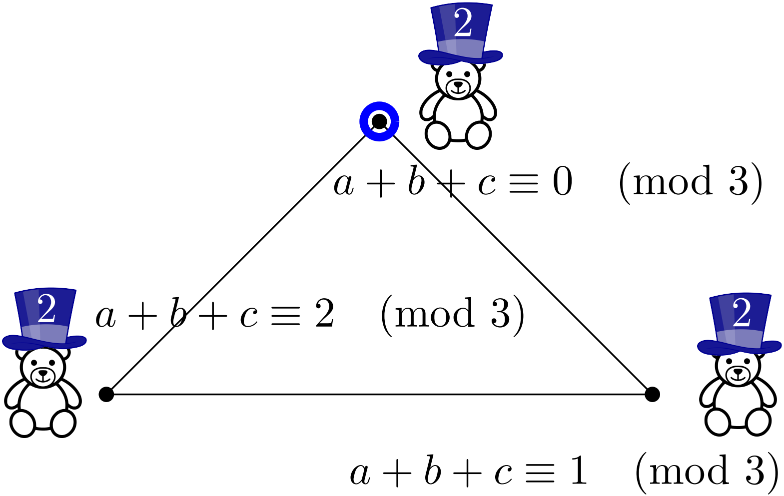
Bears play against an evil *Demon*.

The game proceeds as:

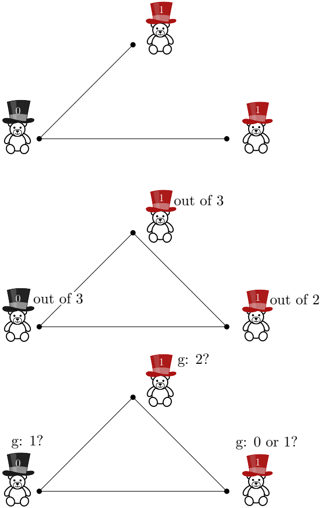
1. Demon presents a graph G and number of colors k.
2. Bears can agree on a strategy. Then bears can no longer talk and are put on vertices of the graph.
3. Demon puts a colored hat on each bear's head.
4. All bears at once guess their hat colors based only on hat's colors of their neighbors.
5. Bears win if at least one bear guesses correctly.


1
$a+b+c \equiv 0 \pmod 3$
0
$a+b+c \equiv 2 \pmod 3$
1
$a+b+c \equiv 1 \pmod 3$


2
$a + b + c \equiv 0 \pmod 3$
2
$a + b + c \equiv 2 \pmod 3$
1
$a + b + c \equiv 1 \pmod 3$

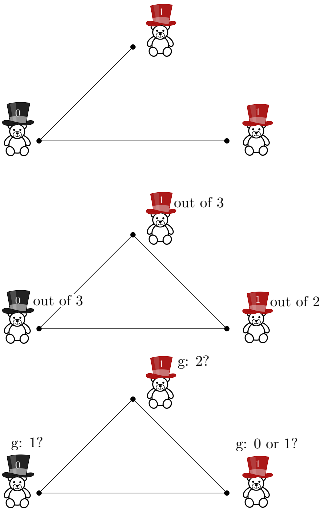

2
2
2
$a + b + c \equiv 0 \pmod 3$
$a + b + c \equiv 2 \pmod 3$
$a + b + c \equiv 1 \pmod 3$

Generalizations



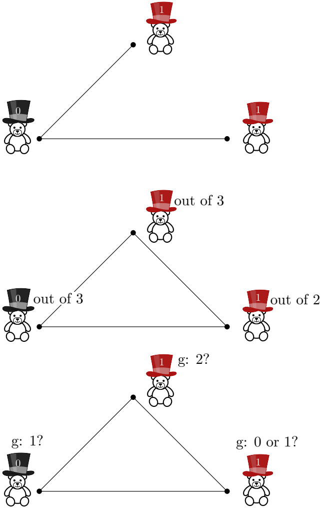
- restrict visibility
- possibly different number of hats for each bear
- allow multiple guesses

Generalizations



- restrict visibility
- possibly different number of hats for each bear
- allow multiple guesses

Generalizations



- restrict visibility
- possibly different number of hats for each bear
- allow multiple guesses

Fractional Hat Chromatic Number

A *hat chromatic number* $\mu(G)$ of a graph G is the maximum number of colors for which bears win.

Definition

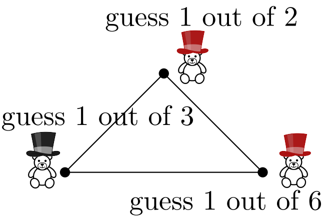
A *fractional hat chromatic number* $\hat{\mu}(G)$ is

$$\hat{\mu}(G) = \sup \{h/g \mid \text{bears win with } h \text{ colors and } g \text{ guesses}\}$$

- $\mu(K_n) = n$, i.e., bears win if $\sum_{v \in V} \frac{1}{h} \geq 1$
- bears win on K_n if $\sum_{v \in V} \frac{1}{h_v} \geq 1$

Theorem

Bears win a game $(K_n = (V, E), \mathbf{h}, \mathbf{g})$ if and only if $\sum_{v \in V} \frac{g_v}{h_v} \geq 1$.



$$\frac{1}{2} + \frac{1}{3} + \frac{1}{6} \geq 1$$

Fractional Hat Chromatic Number

A *hat chromatic number* $\mu(G)$ of a graph G is the maximum number of colors for which bears win.

Definition

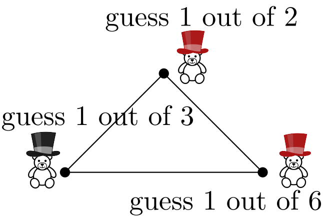
A *fractional hat chromatic number* $\hat{\mu}(G)$ is

$$\hat{\mu}(G) = \sup\{h/g \mid \text{bears win with } h \text{ colors and } g \text{ guesses}\}$$

- $\mu(K_n) = n$, i.e., bears win if $\sum_{v \in V} \frac{1}{h} \geq 1$
- bears win on K_n if $\sum_{v \in V} \frac{1}{h_v} \geq 1$

Theorem

Bears win a game $(K_n = (V, E), \mathbf{h}, \mathbf{g})$ if and only if $\sum_{v \in V} \frac{g_v}{h_v} \geq 1$.



$$\frac{1}{2} + \frac{1}{3} + \frac{1}{6} \geq 1$$

Fractional Hat Chromatic Number

A *hat chromatic number* $\mu(G)$ of a graph G is the maximum number of colors for which bears win.

Definition

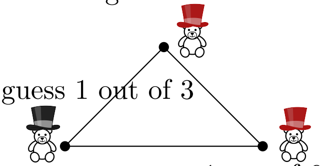
A *fractional hat chromatic number* $\hat{\mu}(G)$ is

$$\hat{\mu}(G) = \sup \{h/g \mid \text{bears win with } h \text{ colors and } g \text{ guesses}\}$$

- $\mu(K_n) = n$, i.e., bears win if $\sum_{v \in V} \frac{1}{h} \geq 1$
- bears win on K_n if $\sum_{v \in V} \frac{1}{h_v} \geq 1$

Theorem

Bears win a game $(K_n = (V, E), \mathbf{h}, \mathbf{g})$ if and only if $\sum_{v \in V} \frac{g_v}{h_v} \geq 1$.



$$\frac{1}{2} + \frac{1}{3} + \frac{1}{6} \geq 1$$

Fractional Hat Chromatic Number

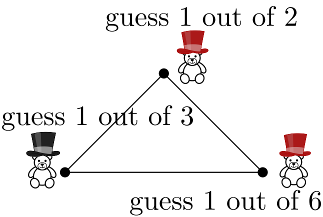
A *hat chromatic number* $\mu(G)$ of a graph G is the maximum number of colors for which bears win.

Definition

A *fractional hat chromatic number* $\hat{\mu}(G)$ is

$$\hat{\mu}(G) = \sup\left\{h/g \mid \text{bears win with } h \text{ colors and } g \text{ guesses}\right\}$$

- $\mu(K_n) = n$, i.e., bears win if $\sum_{v \in V} \frac{1}{h} \geq 1$
- bears win on K_n if $\sum_{v \in V} \frac{1}{h_v} \geq 1$



$$\frac{1}{2} + \frac{1}{3} + \frac{1}{6} \geq 1$$

Theorem

Bears win a game $(K_n = (V, E), \mathbf{h}, \mathbf{g})$ if and only if $\sum_{v \in V} \frac{g_v}{h_v} \geq 1$.

Fractional Hat Chromatic Number

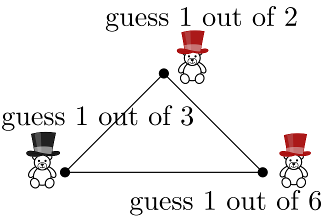
A *hat chromatic number* $\mu(G)$ of a graph G is the maximum number of colors for which bears win.

Definition

A *fractional hat chromatic number* $\hat{\mu}(G)$ is

$$\hat{\mu}(G) = \sup\{h/g \mid \text{bears win with } h \text{ colors and } g \text{ guesses}\}$$

- $\mu(K_n) = n$, i.e., bears win if $\sum_{v \in V} \frac{1}{h} \geq 1$
- bears win on K_n if $\sum_{v \in V} \frac{1}{h_v} \geq 1$



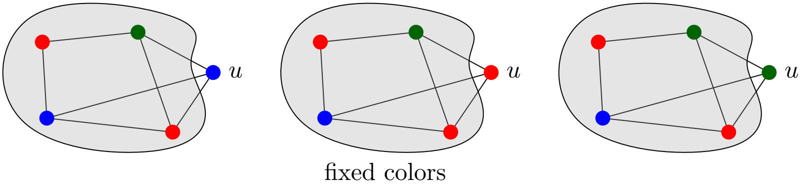
$$\frac{1}{2} + \frac{1}{3} + \frac{1}{6} \geq 1$$

Theorem

Bears win a game $(K_n = (V, E), \mathbf{h}, \mathbf{g})$ *if and only if* $\sum_{v \in V} \frac{g_v}{h_v} \geq 1$.

General case – a connection to Independent sets

How many different colorings can a vertex u guess correctly?

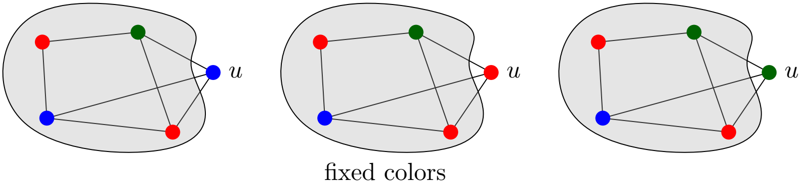


He guesses correctly in exactly $\frac{g_u}{h_u}$ fraction of all colorings.

Trying to count the number of such colorings naturally leads to the independence polynomial.

General case – a connection to Independent sets

How many different colorings can a vertex u guess correctly?

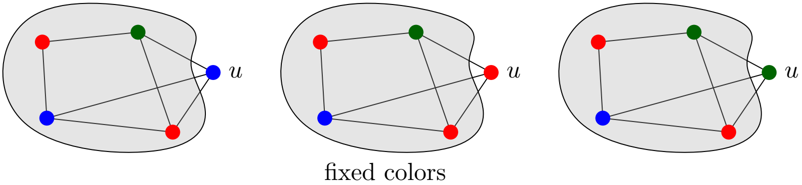


He guesses correctly in exactly $\frac{g_u}{h_u}$ fraction of all colorings.

Trying to count the number of such colorings naturally leads to the independence polynomial.

General case – a connection to Independent sets

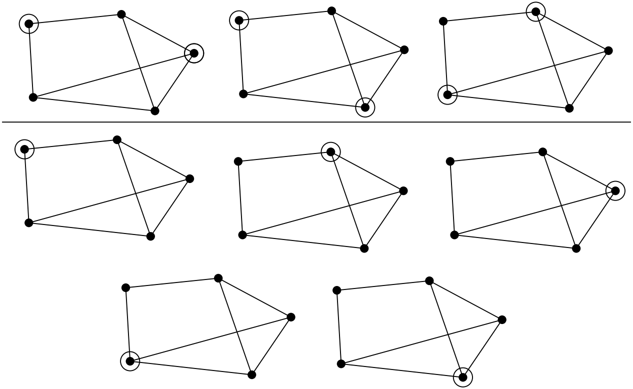
How many different colorings can a vertex u guess correctly?



He guesses correctly in exactly $\frac{g_u}{h_u}$ fraction of all colorings.

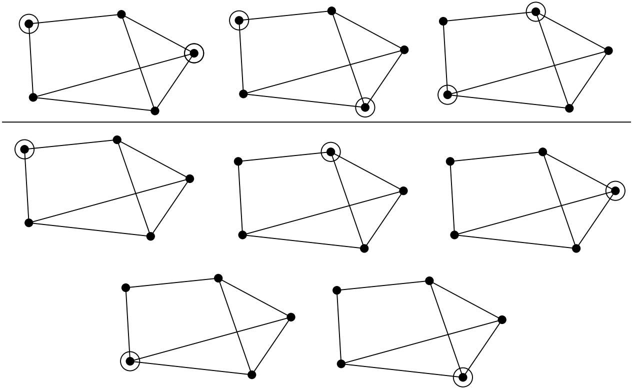
Trying to count the number of such colorings naturally leads to the independence polynomial.

General case – a connection to Independent sets



Independence polynomial: $3x^2 + 5x \qquad = 5 \cdot \frac{1}{3} - 3 \cdot \left(\frac{1}{3}\right)^2$

General case – a connection to Independent sets

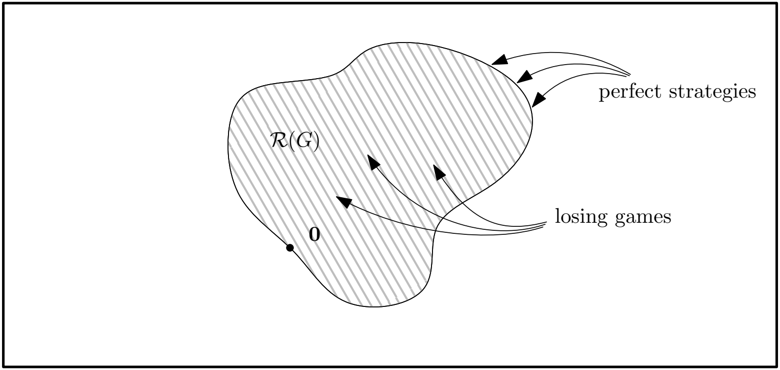


Independence polynomial: $3x^2 + 5x \qquad = 5 \cdot \frac{1}{3} - 3 \cdot \left(\frac{1}{3}\right)^2$

Perfect strategies

Definition

A strategy for a hat guessing game is *perfect* if it is winning and in every hat arrangement, no two bears that guess correctly are on adjacent vertices.



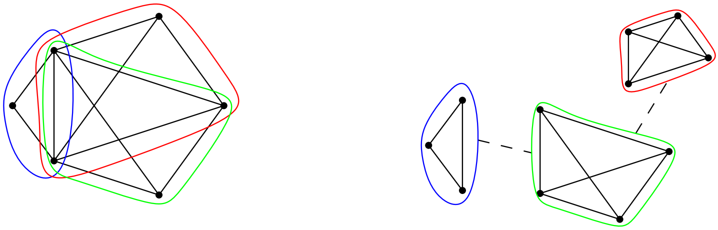
Chordal graphs and their decomposition

Definition

A *clique tree of a graph* G is a tree T whose vertex set is precisely the subsets of V that induce maximal cliques in G and for each $v \in V$ the vertices of T containing v induces a connected subtree.

Fact

G is chordal if and only if it possesses a clique tree.



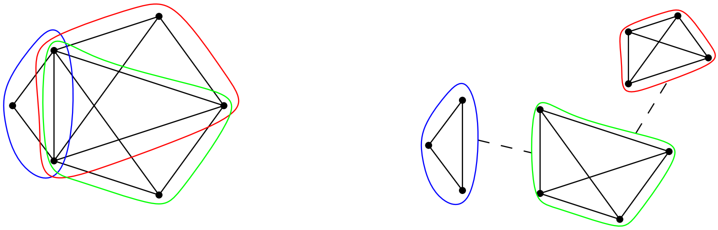
Chordal graphs and their decomposition

Definition

A *clique tree of a graph* G is a tree T whose vertex set is precisely the subsets of V that induce maximal cliques in G and for each $v \in V$ the vertices of T containing v induces a connected subtree.

Fact

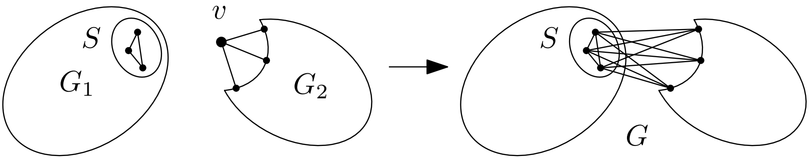
G is chordal if and only if it possesses a clique tree.



Clique join – an operation that builds chordal graphs

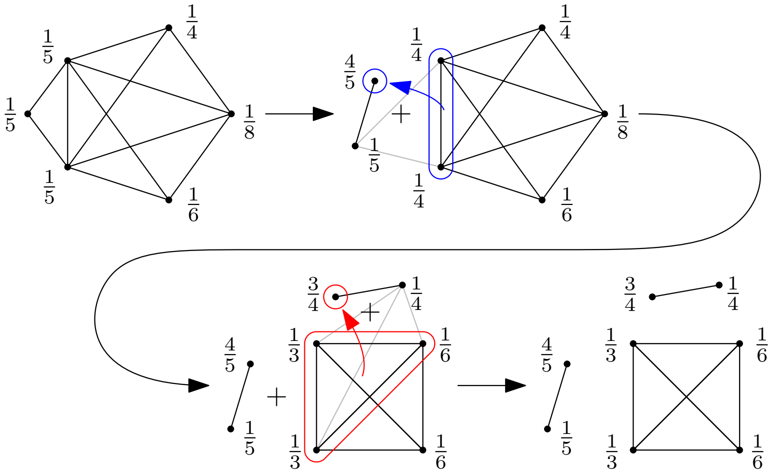
Definition (Clique join)

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be graphs, $S \subseteq V_1$ a clique in G_1 and $v \in V_2$. The *clique join of G_1 and G_2 with respect to S and v* is the graph G:



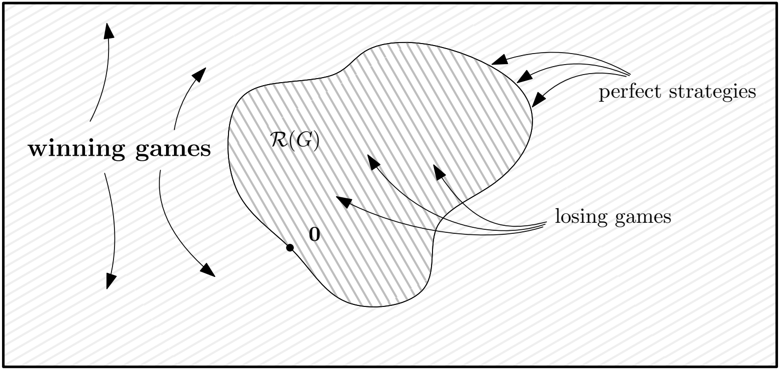
Theorem

There is an algorithm that computes an optimal strategy of bears of an arbitrary chordal graph in polynomial time.



Theorem

There is an algorithm that computes an optimal strategy of bears of an arbitrary chordal graph in polynomial time.



Contribution of the thesis

1. m-Eternal domination
   - provided a toolbox for obtaining bounds on solution size
   - polynomial algorithm for cactus graphs
2. Hat Chromatic Number
   - introduced a fractional generalization of the parameter
   - connected it to graph independence polynomial
   - designed a polynomial algorithm for chordal graphs
3. Online Ramsey Number
   - introduced a concept of Induced online Ramsey numbers
   - showed asymptotically tight constructions and showed an asymptotic gap from its non-game counterpart for trees
4. Group Identification
   - analyzed complexity of 2-player variant of the problem
   - provided a complete parameterized complexity picture

Contribution of the thesis

1. m-Eternal domination
   - provided a toolbox for obtaining bounds on solution size
   - polynomial algorithm for cactus graphs
2. Hat Chromatic Number
   - introduced a fractional generalization of the parameter
   - connected it to graph independence polynomial
   - designed a polynomial algorithm for chordal graphs
3. Online Ramsey Number
   - introduced a concept of Induced online Ramsey numbers
   - showed asymptotically tight constructions and showed an asymptotic gap from its non-game counterpart for trees
4. Group Identification
   - analyzed complexity of 2-player variant of the problem
   - provided a complete parameterized complexity picture

Thank you for your attention!

Question 1

The result about the asymptotic gap between size-Ramsey numbers and online Ramsey numbers is similar to the result of Conlon, which is about complete graphs and was proved using pseudo-random graphs.
Did you consider using these techniques for other graphs besides the complete graphs? Do they apply for trees as well? Can you compare your techniques and the ones used by Conlon?

Comment

It is not clear how one would apply this method to non-complete graphs. We aimed for a constructive result.

Question 2

You found an algorithm for determining the m-eternal domination number of cactus graphs. Can you say something about the growth rate of these numbers with respect to the number of vertices?

Comment

Stars are guarded by 2 guards, paths with $\frac{n}{2}$, and cycles with $\frac{n}{3}$ guards and n guards always suffice.

Our reductions on cactus graphs give the following ratios (guards/vertices): $\frac{0}{1}$, $\frac{1}{3}$, $\frac{2}{5}$, $\frac{3}{7}$, and $\frac{1}{2}$

$\implies$ Cactus graphs can always be defended with $\frac{n}{2} + 1$ guards.

Question 1

What are the difficulties in generalizing the results for eternal domination to graphs of treewidth at most 2?

Comment

- Graph with treewidth 2 have a nice decomposition using cuts of size 2, but they may contain many interlocking cycles.
- Complex cycles structure increase intricacy of the problem significantly.
- Partial results may be obtained by assumptions on the structure of the solution.