Combinatorial Game Representation and Analysis of Snort

Gio Carlo Cielo Borje

UC Irvine

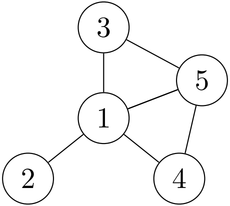
April 29, 2013

Snort Description

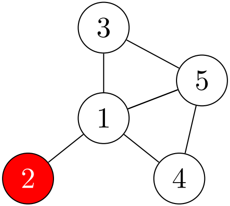
Two players	Red and Blue
Board	map (usually planar)
Moves	• Red colors an available region red • Blue colors an available region blue
Constraints	no two regions can have the opposing color
Gameover	player with no moves loses

Snort Description

Two players	Red and Blue
Board	map (usually planar)
Moves	• Red colors an available region red • Blue colors an available region blue
Constraints	no two regions can have the opposing color
Gameover	player with no moves loses

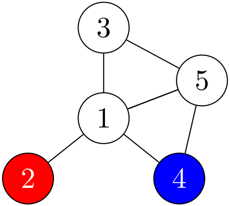


Snort Demo: 1



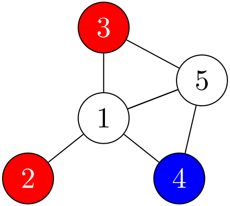
Red selects vertex 2

Snort Demo: 2



Blue selects vertex 4

Snort Demo: 3



Red selected vertex 3

Red player wins

Snort Classification

1. Determinate
2. Zero-sum
3. Asymmetric
4. Perfect information
5. Sequential
6. Normal-play
7. Unfair

Unfairness

Theorem

Snort is an unfair game.

Proof.

Any game that is a zero-sum, partisan, progressively-bounded game with no ties, has a winning strategy for a player that depends on the currently given state. □

Unfairness

Theorem

Snort is an unfair game.

Proof.

Any game that is a zero-sum, partisan, progressively-bounded game with no ties, has a winning strategy for a player that depends on the currently given state. □

Unfairness

Theorem

Snort is an unfair game.

Proof.

Any game that is a zero-sum, partisan, progressively-bounded game with no ties, has a winning strategy for a player that depends on the currently given state. □

Snort: Progressively Bounded

Lemma

Snort is progressively bounded.

Proof.

Given that there are n vertices and there are four vertex states, there is at most $o(4^n)$ possible game configurations. Hence, the state space is finite.
Further, each move locks a particular vertex to a configuration which reduces the state space size. Consequently, there will be at most $O(n)$ moves before a gameover state is reachd. $\square$

Unfairness

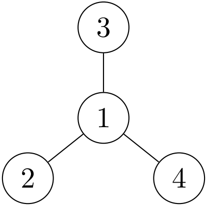
Conclusion:

Theorem

Snort is an unfair game.

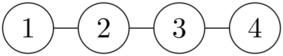
Trivial Graph Families: Star Graphs

All star graphs are **N**-positions.



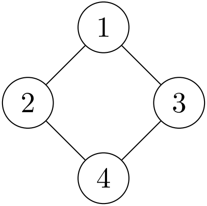
Nontrivial Graph Families: Path Graphs

All path graphs are **N**-positions.

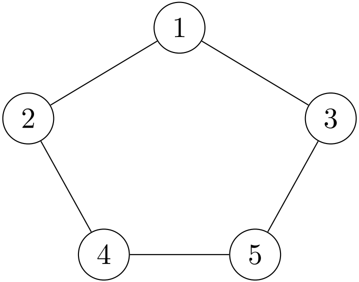


Nontrivial Graph Families: Even Cycle Graphs

All even cycle graphs are **P**-positions.



All odd graphs are **N**-positions.



Game Variants

Consider games of three players or more:

- The state space increases exponentially
- Collusion is an important factor
- Star graphs are still trivial

Consider expanding the space between player pieces:

- Fewer valid board configurations
- Upperbound on state space is still the same
- Star graphs are still trivial

Game Variants

Consider games of three players or more:

- The state space increases exponentially
- Collusion is an important factor
- Star graphs are still trivial

Consider expanding the space between player pieces:

- Fewer valid board configurations
- Upperbound on state space is still the same
- Star graphs are still trivial

Game Variants

Consider games of three players or more:

- The state space increases exponentially
- Collusion is an important factor
- Star graphs are still trivial

Consider expanding the space between player pieces:

- Fewer valid board configurations
- Upperbound on state space is still the same
- Star graphs are still trivial

Game Variants

Consider games of three players or more:

- The state space increases exponentially
- Collusion is an important factor
- Star graphs are still trivial

Consider expanding the space between player pieces:

- Fewer valid board configurations
- Upperbound on state space is still the same
- Star graphs are still trivial

Game Variants

Consider games of three players or more:

- The state space increases exponentially
- Collusion is an important factor
- Star graphs are still trivial

Consider expanding the space between player pieces:

- Fewer valid board configurations
- Upperbound on state space is still the same
- Star graphs are still trivial

Thank you. Snort simulator and solver are available on GitHub:
https://github.com/Hydrotoast/SnortSolver