First Cycle Games

Benjamin Aminof (IST Austria) and Sasha Rubin (TU Wien)

Strategic Reasoning 2014

Games in computer science

Examples

geography, parity games, mean-payoff games, energy games, . . .

Types of Games

- Players: 1, 2, many.
- Information: perfect or imperfect
- Objectives: qualitative or quantitative.
- Duration: finite or infinite.
- Arena: finite or infinite.

Typical Question

Is the game **memoryless determined**? i.e. one of the players has a memoryless winning strategy?

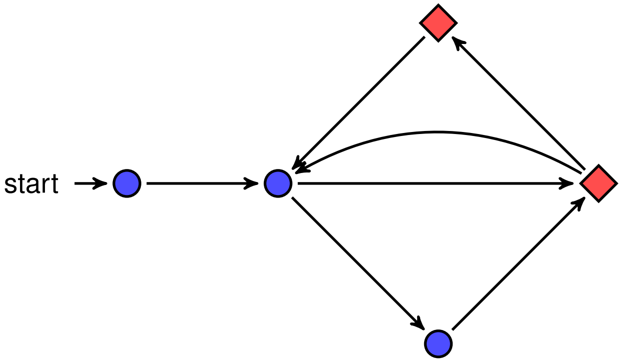
This talk

First Cycle Game FCG(P)

Two players move a token along the edges of a graph.

Player ● wins if the first cycle satisfies P; otherwise Player ◆ wins.

$P =$ even length



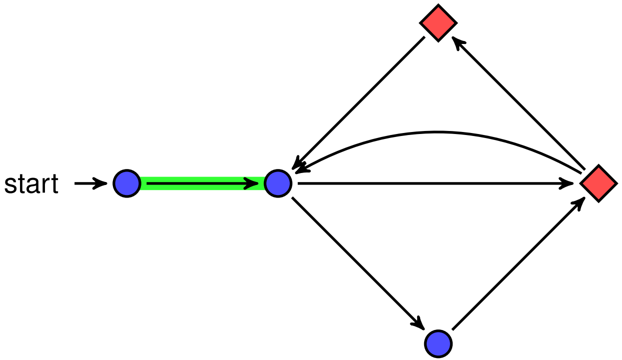
This talk

First Cycle Game FCG(P)

Two players move a token along the edges of a graph.

Player ● wins if the first cycle satisfies P; otherwise Player ◆ wins.

P = even length



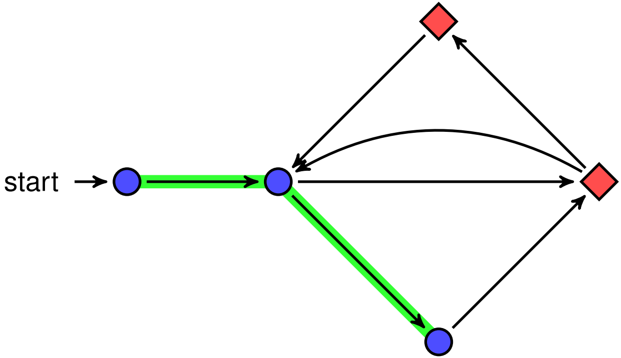
This talk

First Cycle Game FCG(P)

Two players move a token along the edges of a graph.

Player ● wins if the first cycle satisfies P; otherwise Player ◆ wins.

P = even length



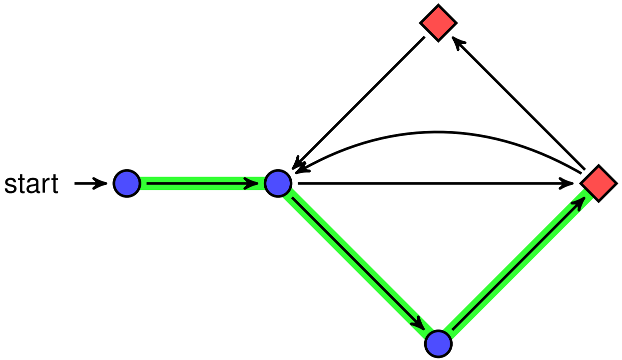
This talk

First Cycle Game FCG(P)

Two players move a token along the edges of a graph.

Player ● wins if the first cycle satisfies P; otherwise Player ◆ wins.

P = even length



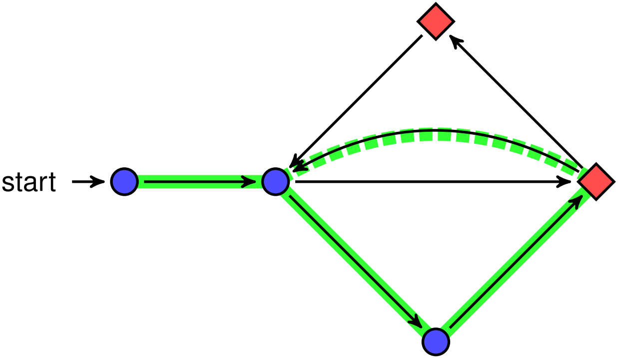
This talk

First Cycle Game FCG(P)

Two players move a token along the edges of a graph.

Player ● wins if the first cycle satisfies P; otherwise Player ◆ wins.

P = even length



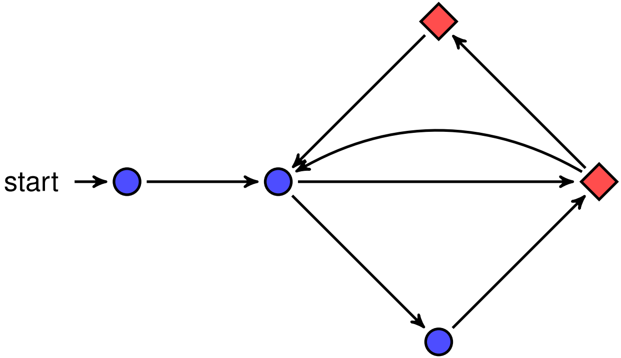
This talk

First Cycle Game FCG(P)

Two players move a token along the edges of a graph.

Player ● wins if the first cycle satisfies P; otherwise Player ◆ wins.

P = even length



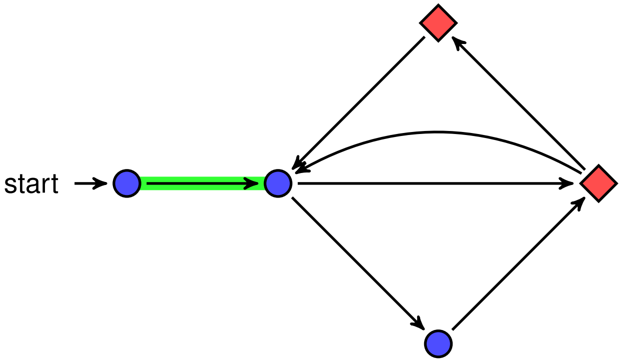
This talk

First Cycle Game FCG(P)

Two players move a token along the edges of a graph.

Player ● wins if the first cycle satisfies P; otherwise Player ◆ wins.

P = even length



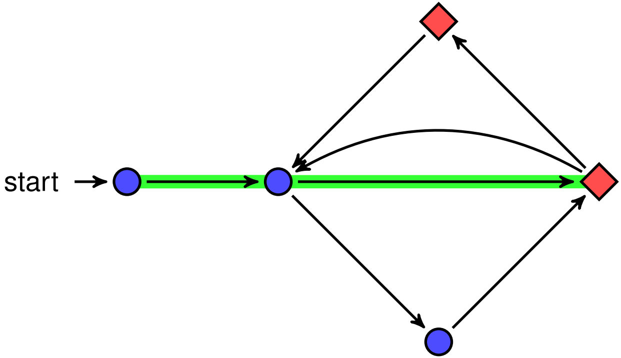
This talk

First Cycle Game FCG(P)

Two players move a token along the edges of a graph.

Player ● wins if the first cycle satisfies P; otherwise Player ◆ wins.

P = even length



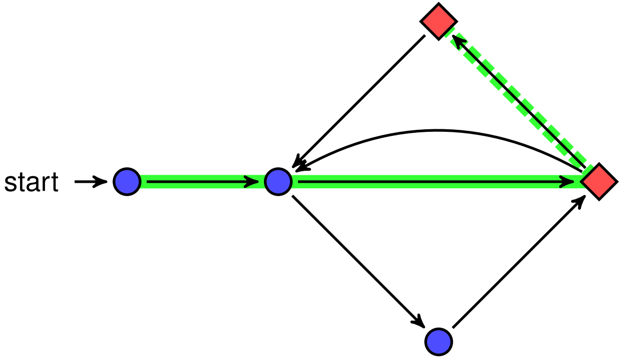
This talk

First Cycle Game FCG(P)

Two players move a token along the edges of a graph.

Player ● wins if the first cycle satisfies P; otherwise Player ◆ wins.

P = even length



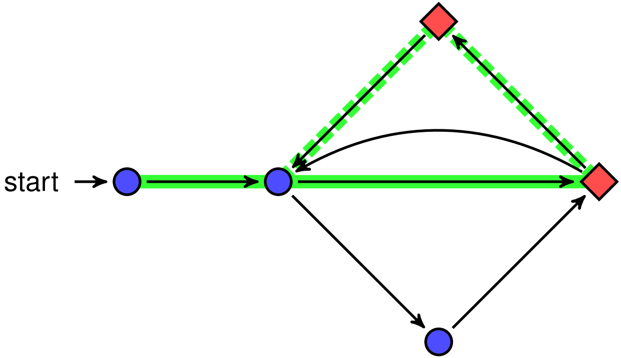
This talk

First Cycle Game FCG(P)

Two players move a token along the edges of a graph.

Player ● wins if the first cycle satisfies P; otherwise Player ◆ wins.

P = even length



1. FCGs are natural

Examples.

- The largest priority on the first cycle is even.
- The average weight of the first cycle is positive.
- The first cycle contains every vertex exactly once.
- ◆ closes the cycle.

⋮

Game = Arena + Objective

- A game is played on an **arena** (V, E, v_0, λ) where $V = V_{\bullet} \sqcup V_{\blacklozenge}$, $\lambda : V \to \mathbb{U}$ a **labeling**.
- A **play** is an infinite path starting in $v_0 \in V$.
- An **objective for** $\bullet$ is a set $W_{\bullet}$ of plays (usually depending on the labeling).
- A play is **won by** $\bullet$ if it is in $W_{\bullet}$; otherwise it is won by $\blacklozenge$.

Strategies

- A **strategy** for ⬟ is a function from finite plays ending in $V_{⬟}$ to V.
- A strategy for ⬟ is **winning** if every play consistent with it is won by ⬟.
- A strategy is **memoryless** if it only depends on the last vertex of the play.
- A game **memoryless determined** if one of the players has a memoryless winning strategy.

First Cycle Games

A **sequence property** is a set $P \subseteq \mathbb{U}^*$.

FCG(P)

A **First Cycle Game over** P is a game for which a play is in $W_\bullet$ iff the labeling of the **first cycle** of the play is in P.

Examples

- P = even length.
- P = average weight positive ($\mathbb{U} = \mathbb{Q}$).
- P = largest priority even ($\mathbb{U} = \mathbb{Z}$).
- P = permutations of V ($\mathbb{U} = V$).
- P = ends in ◆ ($\mathbb{U} = \{\bullet, ◆\}$).

2. FCGs have high memory requirements, in general.

A strategy uses **finite-memory** if it can be implemented by a Mealy machine that operates on vertices.

Theorem

For FCGs

- $O(|V|)!$ memory is sufficient for a winning strategy.
- $\Omega(|V|)!$ memory may be required for a winning strategy.

Game requiring $N!$ memory, realised as FCG size $O(N)$

- ◆ picks a permutation of $\{1, \cdots, N\}$, and then $i, j \leq N$.
- ● picks $x \in \{i, j\}$.
- To win, ● must ensure x appears before $\{i, j\} \setminus \{x\}$ in the permutation.

Take Away Messages

1. FCGs are natural.
2. FCGs have high memory requirements, in general.
3. Memoryless determined FCGs are typically easy to identify.

Easy-to-check properties

Definition

A sequence property $P \subseteq \mathbb{U}^*$ is

- **shift-closed** if $ab \in P \implies ba \in P$.
- **cat-closed** if $a, b \in P \implies ab \in P$.

Sequence property P	shift-closed	cat-closed
Largest priority is even	✔	✔
Average weight is positive	✔	✔
Length is even	✔	✔
Length is odd	✔	✘
Permutation of V	✔	✘
Ends in ◆	✘	✔

3. FCGs that are memoryless determined are typically easy to identify

Write $\neg P$ for $\mathbb{U}^* \setminus P$.

Theorem (Memoryless FCGs)

If

1. P is shift-closed, and
2. both P and $\neg P$ are cat-closed,

then every FCG(P) is memoryless determined.

Examples

Sequence property P	shift-closed	cat-closed	$\neg P$ cat-closed
Largest priority is even	✔	✔	✔
Average weight is positive	✔	✔	✔
Length is even	✔	✔	✘

This repairs an error in...

ELSEVIER

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®
Theoretical Computer Science 310 (2004) 365–378

Theoretical
Computer Science

www.elsevier.com/locate/tcs

Memoryless determinacy of parity and mean payoff games: a simple proof☆

Henrik Björklund, Sven Sandberg, Sergei Vorobyov*

Information Technology Department, Uppsala University, Box 337, SE-751 05 Uppsala, Sweden

Received 25 March 2003; accepted 29 July 2003
Communicated by A.S. Fraenkel

Abstract

We give a simple, direct, and constructive proof of memoryless determinacy for parity and mean payoff games. First, we prove by induction that the finite duration versions of these games, played until some vertex is repeated, are determined and both players have memoryless winning strategies. In contrast to the proof of Ehrenfeucht and Mycielski, *Internat. J. Game Theory*, 8 (1979) 109–113, our proof does not refer to the infinite-duration versions. Second, we show that memoryless determinacy straightforwardly generalizes to infinite duration versions of parity and mean payoff games.

© 2003 Elsevier B.V. All rights reserved.

and uses ideas from, and generalises, ...

Int. Journal of Game Theory, Vol. 8, Issue 2, page 109–113. ©*Physica-Verlag, Vienna.*

Positional Strategies for Mean Payoff Games

By *A. Ehrenfeucht* and *J. Mycielski*, Boulder[1])[2])

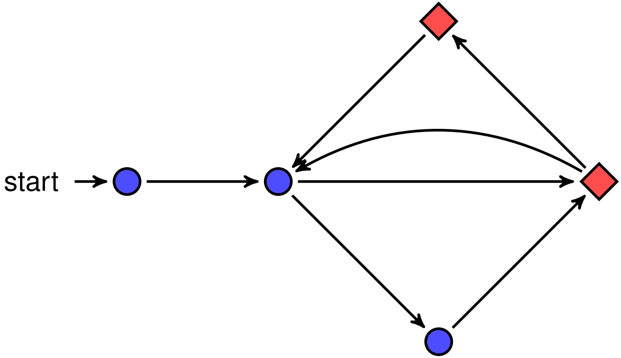
Abstract: We study some games of perfect information in which two players move alternately along the edges of a finite directed graph with weights attached to its edges. One of them wants to maximize and the other to minimize some means of the encountered weights.

Take Away Messages

1. FCGs are natural.
2. FCGs have high memory requirements, in general.
3. Memoryless determined FCGs are typically easy to identify.
4. FCGs are equivalent to certain infinite duration games.

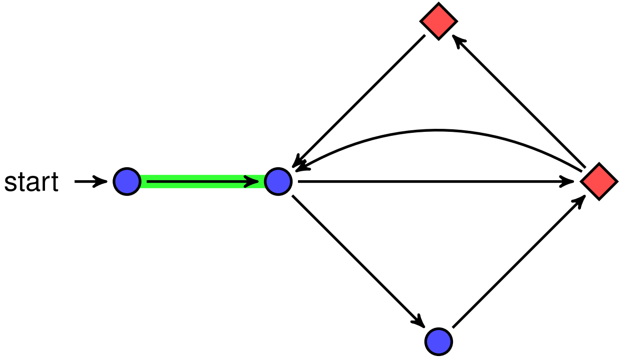
What is the connection?

Decomposition of a play into simple cycles



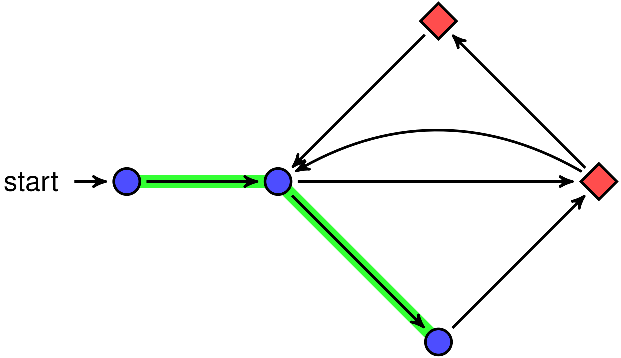
What is the connection?

Decomposition of a play into simple cycles



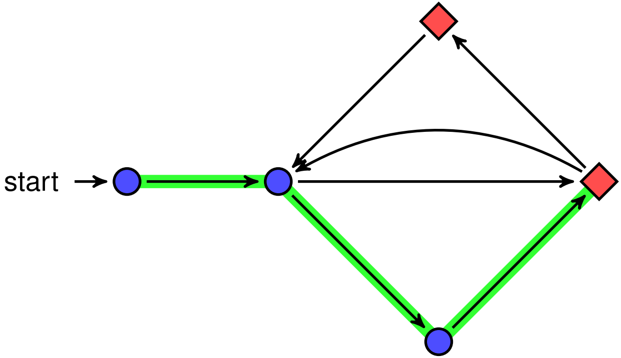
What is the connection?

Decomposition of a play into simple cycles



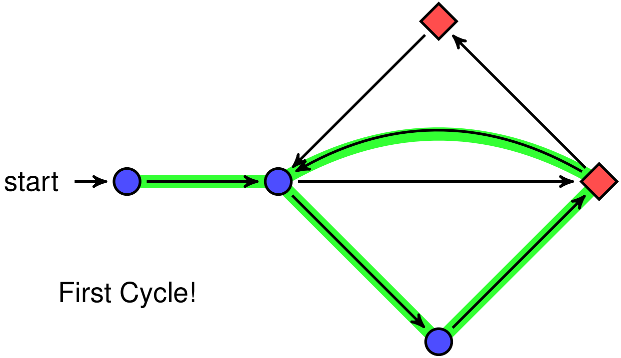
What is the connection?

Decomposition of a play into simple cycles



What is the connection?

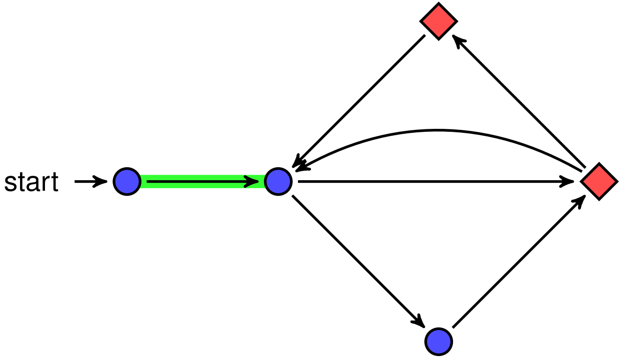
Decomposition of a play into simple cycles



First Cycle!

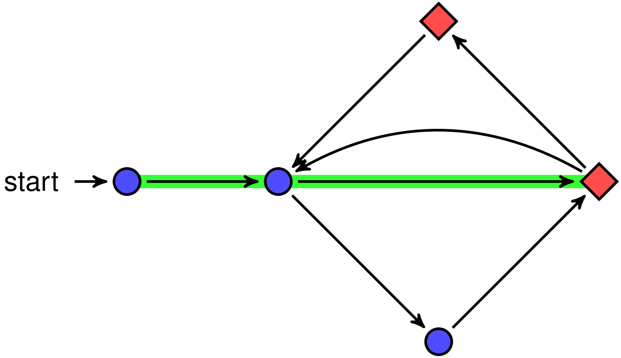
What is the connection?

Decomposition of a play into simple cycles



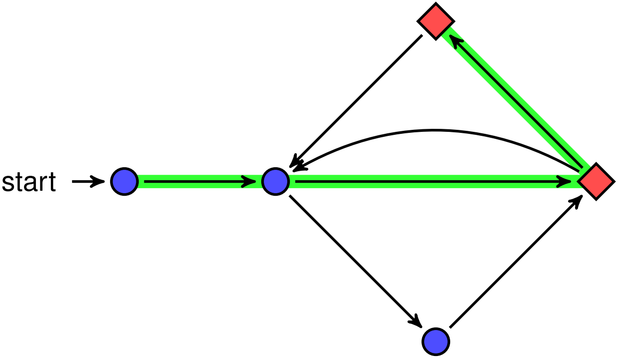
What is the connection?

Decomposition of a play into simple cycles



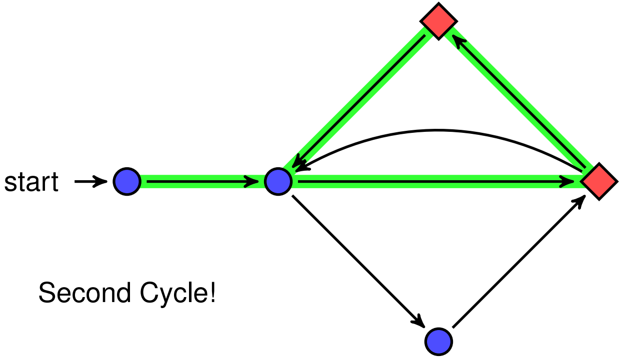
What is the connection?

Decomposition of a play into simple cycles



What is the connection?

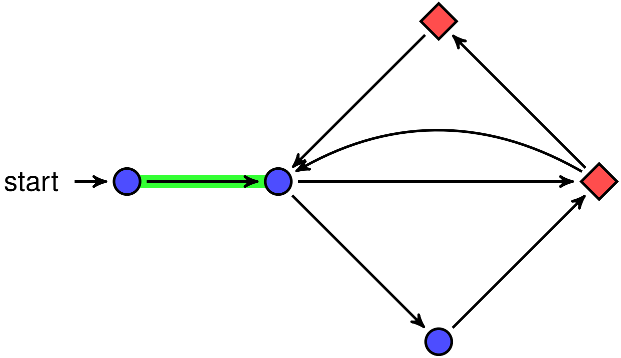
Decomposition of a play into simple cycles



Second Cycle!

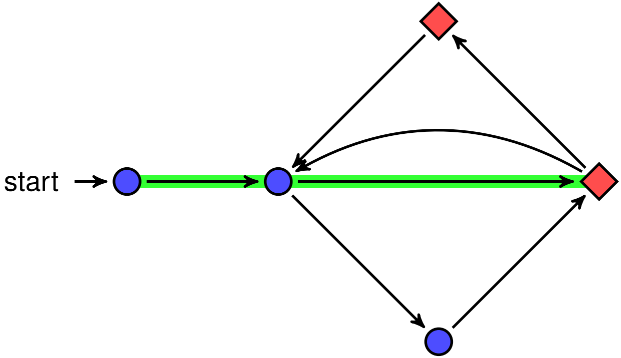
What is the connection?

Decomposition of a play into simple cycles



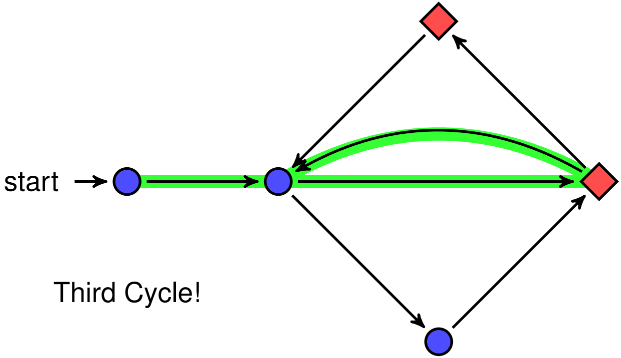
What is the connection?

Decomposition of a play into simple cycles



What is the connection?

Decomposition of a play into simple cycles



Third Cycle!

Greedy Games

Definition

A game G is P**-greedy** if for every play π:

1. every cycle of π satisfies $P \implies \pi$ is won by ●;
2. every cycle of π satisfies $\neg P \implies \pi$ is won by ◆.

Intuition

A player is guaranteed to win a greedy game if he ensures every cycle satisfies a property (or its complement).

Example

- Every Parity Game is P-greedy where $P =$ max priority is even.

4. FCGs are equivalent to certain infinite duration games

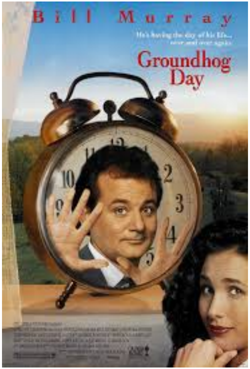
Theorem (Transfer)

If G is P-greedy then memoryless winning strategies transfer between G and FCG(P).

4. FCGs are equivalent to certain infinite duration games

Theorem (Transfer)

If G is P-greedy then memoryless winning strategies transfer between G and FCG(P).



A **time loop** or **temporal loop** is a common plot device in science fiction (especially in universes where time travel is commonplace) where a certain length of time (such as a few hours, or a few days) repeats over and over. When the time loop "resets", the memories of most characters are reset, and behave as though they're not aware of the loop.

Memoryless FCGs + Transfer

Corollary

The following games are memoryless determined (finite arenas):

1. Parity games
2. Mean payoff games
3. Energy games (initial credit problem)
4. Priority mean-payoff games (i.e. positive mean payoff of the subsequence selected by max priority occuring infinitely often)

⋮

Take Away Messages

1. FCGs are natural.
2. FCGs have high memory requirements, in general.
3. Memoryless determined FCGs are typically easy to identify.
4. FCGs are equivalent to certain infinite duration games.

Recipe for proving G is memoryless determined

1. 'Finitise' the winning condition of G to get a sequence property P.
2. Show that P is shift-closed, cat-closed, and $\neg P$ is cat-closed.
3. Show that G is P-greedy.

Extra Slides

All Cycles Game ACG(P)

All Cycles Game ACG(P)

Two players move a token along the edges of a graph.

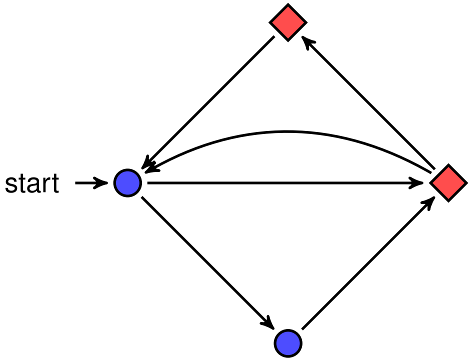
Player ● wins if **all** the cycles in the decomposition satisfy the property P.

Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Even Length

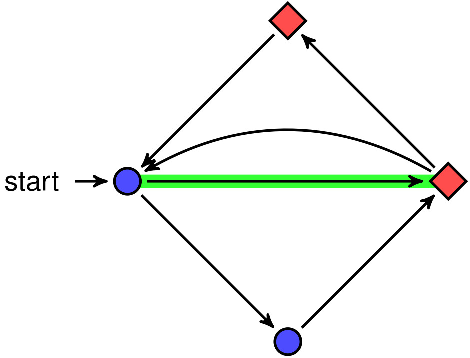


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Even Length

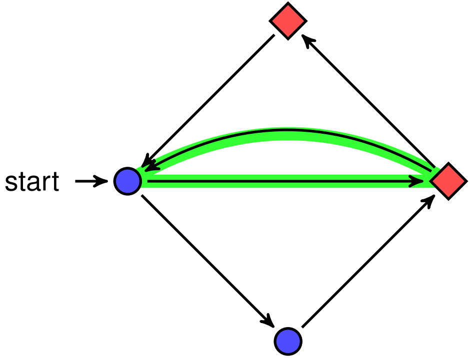


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Even Length

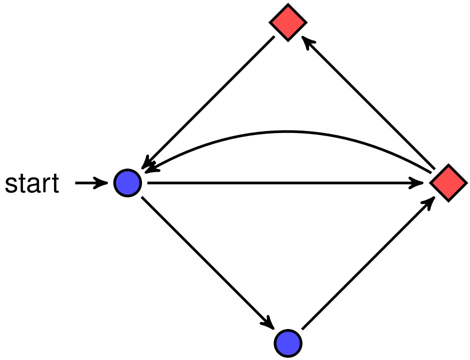


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Even Length

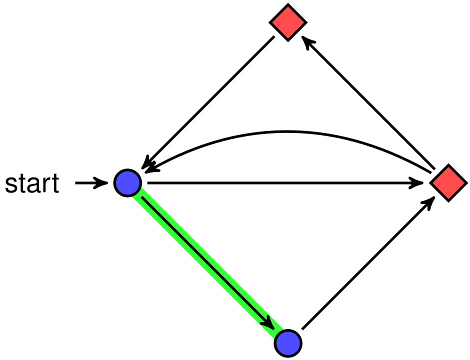


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Even Length

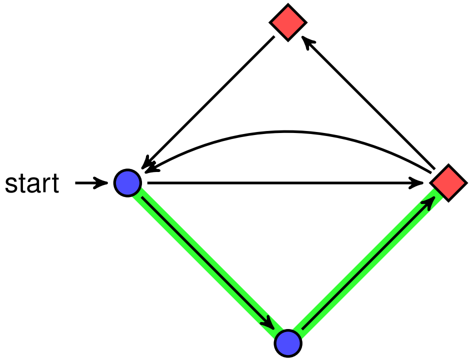


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Even Length

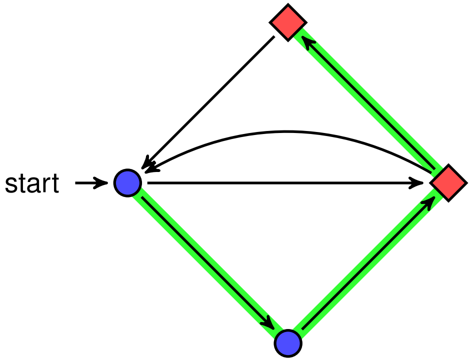


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Even Length

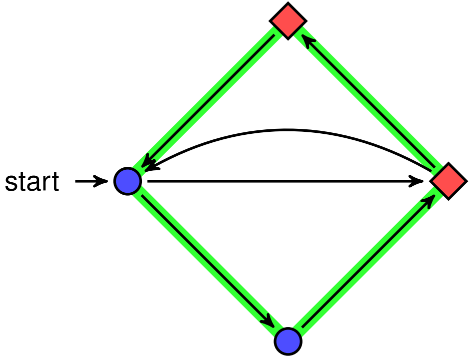


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Even Length

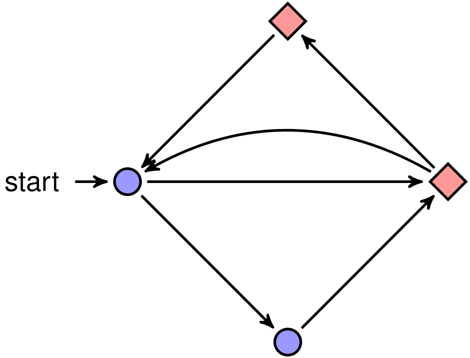


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Odd Length

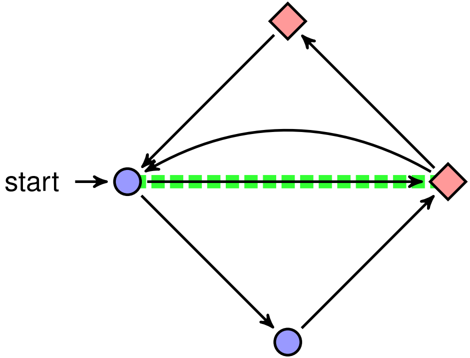


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Odd Length

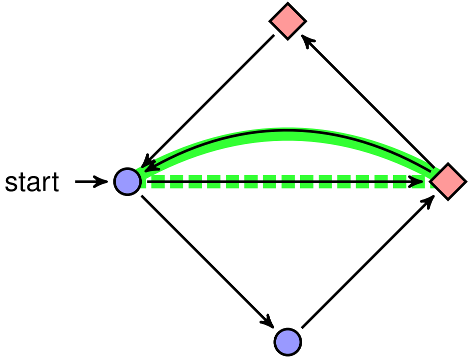


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Odd Length

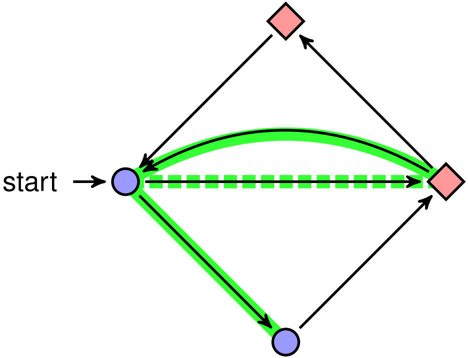


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Odd Length

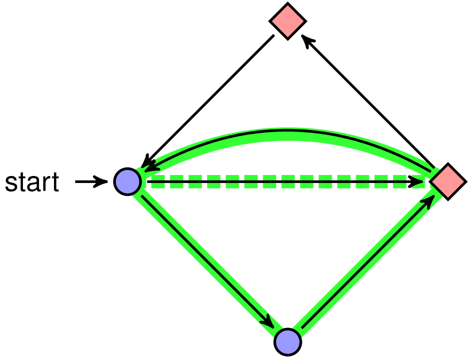


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Odd Length

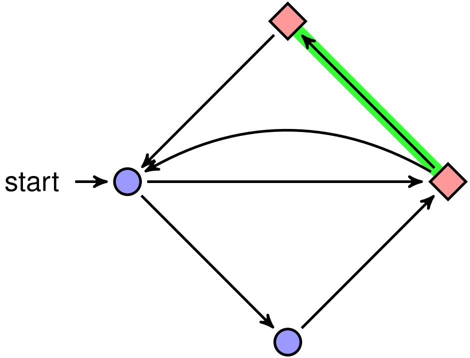


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Odd Length

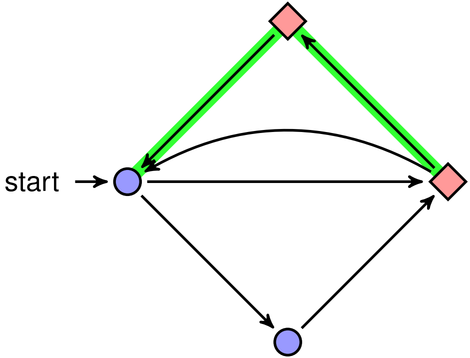


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Odd Length

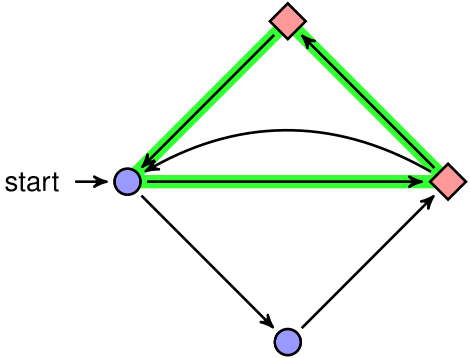


Suffix All Cycles Games

Two players move a token along the edges of a graph.

Player ● wins if he wins the ACG on some suffix of the play; otherwise ◆ wins.

Property = Odd Length



Unambigious properties

Definition

Sequence property P is **unambiguous** if there is no play in any arena that is won by ● in both $SCG(P)$ and SCG($\neg P$).

Examples

- $P =$ max priority even.
- $P =$ average weight is positive.

Unambigious properties

Definition

Sequence property P is **unambiguous** if there is no play in any arena that is won by $\bullet$ in both $SCG(P)$ and SCG($\neg P$).

Examples

- $P =$ max priority even.
- $P =$ average weight is positive.

Theorem (Generalisation of Ehrenfeucht+Mycielski)

If P is unambiguous then every FCG(P) is memoryless determined, and memoryless strategies transfer between FCG(P), ACG(P), SCG(P).