Combinatorial Aspects of the Card Game War

Atharva Pathak
Mentor: Tanya Khovanova

MIT PRIMES Conference

16 October 2021
Preliminaries

- Two players, Alice and Bob
Preliminaries

- Two players, Alice and Bob
- A finite deck of cards labelled 1 through $n$
Preliminaries

- Two players, Alice and Bob
- A finite deck of cards labelled 1 through $n$
- The deck is shuffled and divided among Alice and Bob, which they keep in their stacks.

A round consists of the players drawing a card from the top of their stacks. The player with the higher card wins the round and gets both cards. The cards are returned to the bottom of a player's stack according to a putback rule - random putback, where the two cards are put back in either order randomly - WL-putback, where the winning card is put back before the losing card. Game ends when a player has no cards left.
Preliminaries

- Two players, Alice and Bob
- A finite deck of cards labelled 1 through $n$
- The deck is shuffled and divided among Alice and Bob, which they keep in their stacks.
- A round consists of the players drawing a card from the top of their stacks. The player with the higher card wins the round and gets both cards.
Preliminaries

- Two players, Alice and Bob
- A finite deck of cards labelled 1 through $n$
- The deck is shuffled and divided among Alice and Bob, which they keep in their stacks.
- A round consists of the players drawing a card from the top of their stacks. The player with the higher card wins the round and gets both cards.
- The cards are returned to the bottom of a player’s stack according to a putback rule.
Preliminaries

- Two players, Alice and Bob
- A finite deck of cards labelled 1 through $n$
- The deck is shuffled and divided among Alice and Bob, which they keep in their stacks.
- A round consists of the players drawing a card from the top of their stacks. The player with the higher card wins the round and gets both cards.
- The cards are returned to the bottom of a player’s stack according to a putback rule
  - random putback, where the two cards are put back in either order randomly
Preliminaries

- Two players, Alice and Bob
- A finite deck of cards labelled 1 through $n$
- The deck is shuffled and divided among Alice and Bob, which they keep in their stacks.
- A round consists of the players drawing a card from the top of their stacks. The player with the higher card wins the round and gets both cards.
- The cards are returned to the bottom of a player’s stack according to a putback rule
  - random putback, where the two cards are put back in either order randomly
  - WL-putback, where the winning card is put back before the losing card

Game of War
16 October 2021 2 / 20
Preliminaries

- Two players, Alice and Bob
- A finite deck of cards labelled 1 through $n$
- The deck is shuffled and divided among Alice and Bob, which they keep in their stacks.
- A round consists of the players drawing a card from the top of their stacks. The player with the higher card wins the round and gets both cards.
- The cards are returned to the bottom of a player’s stack according to a putback rule
  - random putback, where the two cards are put back in either order randomly
  - WL-putback, where the winning card is put back before the losing card
- Game ends when a player has no cards left
Preliminaries

- We represent a state of the game as

\[ a_1 a_2 \ldots a_i | a_{i+1} a_{i+2} \ldots a_n, \]

where \( a_1 \ldots a_i \) represent Alice’s stack from top to bottom and \( a_{i+1} \ldots a_n \) represents Bob’s stack from top to bottom.
Preliminaries

- We represent a *state* of the game as

  \[ a_1a_2 \ldots a_i|a_{i+1}a_{i+2} \ldots a_n, \]

  where \( a_1 \ldots a_i \) represent Alice’s stack from top to bottom and \( a_{i+1} \ldots a_n \) represents Bob’s stack from top to bottom.

- E.g. 2|134 with WL-putback:

  \[
  \begin{align*}
  2 & \mid 134 \\
  \Rightarrow 21 & \mid 34 \\
  \Rightarrow 1 & \mid 432 \\
  \Rightarrow & \mid 3241.
  \end{align*}
  \]
Passthroughs

**Definition (Passthrough)**

Given a state where a player has $m$ cards, their stack undergoes a **passthrough** (PT) after $m$ rounds. These $m$ rounds occur during the passthrough.

Example:

Again consider $2|134$.

$2|134 \Rightarrow 21|34 \Rightarrow 1|432 \Rightarrow |3241$

Bob has won on his first passthrough

Alice lost on her second passthrough

We only consider games where Bob wins within his first passthrough.
Passthroughs

**Definition (Passthrough)**

Given a state where a player has $m$ cards, their stack undergoes a passthrough (PT) after $m$ rounds. These $m$ rounds occur during the passthrough.

Example:

- Again consider $2|134$. 

Bob has won on his first passthrough

Alice lost on her second passthrough

We only consider games where Bob wins within his first passthrough.
Definition (Passthrough)

Given a state where a player has \( m \) cards, their stack undergoes a passthrough (PT) after \( m \) rounds. These \( m \) rounds occur during the passthrough.

Example:
- Again consider 2|134.
- \( 2|134 \Rightarrow 21|34 \Rightarrow 1|432 \Rightarrow |3241 \)
Passthroughs

Definition (Passthrough)

Given a state where a player has \( m \) cards, their stack undergoes a **passthrough** (PT) after \( m \) rounds. These \( m \) rounds occur during the passthrough.

Example:

- Again consider 2|134.
- \[ 2|134 \rightarrow 21|34 \rightarrow 1|432 \rightarrow |3241 \]
- Bob has won on his first passthrough
Passthroughs

Definition (Passthrough)

Given a state where a player has \( m \) cards, their stack undergoes a passthrough (PT) after \( m \) rounds. These \( m \) rounds occur during the passthrough.

Example:

- Again consider 2|134.
- 2|134 → 21|34 → 1|432 → |3241
- Bob has won on his first passthrough
- Alice lost on her second passthrough
Passthroughs

Definition (Passthrough)

Given a state where a player has $m$ cards, their stack undergoes a **passthrough** (PT) after $m$ rounds. These $m$ rounds occur during the passthrough.

Example:

- Again consider 2|134.
- 2|134 $\rightarrow$ 21|34 $\rightarrow$ 1|432 $\rightarrow$ |3241
- Bob has won on his first passthrough
- Alice lost on her second passthrough
- **We only consider games where Bob wins within his first passthrough.**
Visualizing Passthroughs

(Source: commons.wikimedia.org/wiki/File:Customer_divider_bar_1.jpg)
Definition (Block)

A block is a subset of cards which may have played in rounds against each other but haven’t played in rounds against any other cards.
Blocks

Definition (Block)
A block is a subset of cards which may have played in rounds against each other but haven’t played in rounds against any other cards.

Example:
- Take the state 2|134
Blocks

**Definition (Block)**

A block is a subset of cards which may have played in rounds against each other but haven’t played in rounds against any other cards.

Example:

- Take the state 2|134
- Initially each card is in its own block.
Blocks

Definition (Block)

A block is a subset of cards which may have played in rounds against each other but haven’t played in rounds against any other cards.

Example:

- Take the state 2|134
- Initially each card is in its own block.
- When Alice wins the first round, cards 1 and 2 form a block, while 3 and 4 are still in their own blocks.
Blocks

Definition (Block)
A block is a subset of cards which may have played in rounds against each other but haven’t played in rounds against any other cards.

Example:
- Take the state 2|134
- Initially each card is in its own block.
- When Alice wins the first round, cards 1 and 2 form a block, while 3 and 4 are still in their own blocks.
- When this game ends, there is only one block consisting of all the cards.
Blocks

Definition (Block)

A block is a subset of cards which may have played in rounds against each other but haven’t played in rounds against any other cards.

Example:

- Take the state 2|134
- Initially each card is in its own block.
- When Alice wins the first round, cards 1 and 2 form a block, while 3 and 4 are still in their own blocks.
- When this game ends, there is only one block consisting of all the cards.
- If instead the game started as 21|34, there would be two blocks at the end.
Blocks

**Definition (Block)**

A block is a subset of cards which may have played in rounds against each other but haven’t played in rounds against any other cards.

Example:

- Take the state 2|134
- Initially each card is in its own block.
- When Alice wins the first round, cards 1 and 2 form a block, while 3 and 4 are still in their own blocks.
- When this game ends, there is only one block consisting of all the cards.
- If instead the game started as 21|34, there would be two blocks at the end.
- Each block acts as its own mini-game of War.
Level-$k$ Single-Use States

**Definition (Level-$k$ Single-Use States)**

A level-$k$ single-use state is an initial state with WL-putback from which Bob wins during his first passthrough and Alice undergoes at most $k$ passthroughs.
Level-$k$ Single-Use States

Definition (Level-$k$ Single-Use States)

A level-$k$ single-use state is an initial state with WL-putback from which Bob wins during his first passthrough and Alice undergoes at most $k$ passthroughs.

$2|134$ is level-2 single-use.
Definition (Level-k Single-Use States)

A level-$k$ single-use state is an initial state with WL-putback from which Bob wins during his first passthrough and Alice undergoes at most $k$ passthroughs.

2|134 is level-2 single-use.
Note that in these single-use states, the order Bob puts his cards back doesn’t matter because they don’t show up in the game again. We can regard cards Bob wins as discarded.
Consider a state $a_1a_2\ldots a_m|a_{m+1}\ldots a_n$, where $n \geq m \cdot 2^k$. If this is a level-$k$ single-use state, we have the following:
Consider a state $a_1 a_2 \ldots a_m | a_{m+1} \ldots a_n$, where $n \geq m \cdot 2^k$. If this is a level-$k$ single-use state, we have the following:

**Proposition**

*No two of Alice’s initial cards ever come in a single block.*
Consider a state $a_1a_2...a_m|a_{m+1}...a_n$, where $n \geq m \cdot 2^k$. If this is a level-$k$ single-use state, we have the following:

**Proposition**

*No two of Alice’s initial cards ever come in a single block.*

We call a block $a_1|a_2...a_{2^k}$ a level-$k$ block if it is a level-$k$ single-use state when played as a game of War.
Blocks in Single-Use States

Consider a state \( a_1a_2\ldots a_m|a_{m+1}\ldots a_n \), where \( n \geq m\cdot2^k \). If this is a level-\( k \) single-use state, we have the following:

**Proposition**

*No two of Alice’s initial cards ever come in a single block.*

- We call a block \( a_1|a_2\ldots a_{2^k} \) a level-\( k \) block if it is a level-\( k \) single-use state when played as a game of War.
- From now on we only discuss blocks where Alice has a single card.
Probability that a State is Level-$k$ Single-Use

Theorem

The chance a random permutation of the $2^k$ cards in a state $a_1 | a_2 \ldots a_{2^k}$ is a level-$k$ block is $P_k$, where $P_1 = \frac{1}{2}$ and recursively

$$P_{k+1} = \frac{1}{2} + \frac{1}{2} P_k^2.$$
Probability that a State is Level-\(k\) Single-Use

**Theorem**

The chance a random permutation of the \(2^k\) cards in a state \(a_1|a_2\ldots a_{2^k}\) is a level-\(k\) block is \(P_k\), where \(P_1 = \frac{1}{2}\) and recursively

\[
P_{k+1} = \frac{1}{2} + \frac{1}{2}P_k^2.
\]

The first few terms in this sequence are \(P_2 = \frac{5}{8}\), \(P_3 = \frac{89}{128}\), \(P_4 = \frac{24305}{32768}\).
Win-Loss Sequences

Definition (Win-Loss Sequence)

A win-loss sequence is a string of $W$’s and $L$’s describing the progress of the game from the point of view of Alice; $W$’s are wins and $L$’s are losses for her.
Definition (Win-Loss Sequence)

A win-loss sequence is a string of $W$’s and $L$’s describing the progress of the game from the point of view of Alice; $W$’s are wins and $L$’s are losses for her.

We can mark off passthroughs of Alice’s stack with /’s in the win-loss sequence.
Win-Loss Sequences

**Definition (Win-Loss Sequence)**

A win-loss sequence is a string of $W$’s and $L$’s describing the progress of the game from the point of view of Alice; $W$’s are wins and $L$’s are losses for her.

We can mark off passthroughs of Alice’s stack with /’s in the win-loss sequence.

Example:

- Take 2|134 with WL-putback.
Win-Loss Sequences

**Definition (Win-Loss Sequence)**

A win-loss sequence is a string of $W$’s and $L$’s describing the progress of the game from the point of view of Alice; $W$’s are wins and $L$’s are losses for her.

We can mark off passthroughs of Alice’s stack with /’s in the win-loss sequence.

Example:

- Take 2|134 with WL-putback.
- $2|134 \Rightarrow 21|34 \Rightarrow 1|432 \Rightarrow |3241$
A win-loss sequence is a string of $W$’s and $L$’s describing the progress of the game from the point of view of Alice; $W$’s are wins and $L$’s are losses for her.

We can mark off passthroughs of Alice’s stack with '/'s in the win-loss sequence.

Example:

- Take $2|134$ with WL-putback.
- $2|134 \implies 21|34 \implies 1|432 \implies |3241$
- So win-loss sequence is $W/LL$. 
More on Win-Loss Sequences

Before each subsequent passthrough, the number of cards Alice has is twice the number of wins she had in the previous passthrough, because each win yields two cards back to Alice’s stack.

Theorem

Win-loss sequences are in bijection with full binary trees.
Definition (Full Binary Tree)

A full binary tree (FBT) is a binary tree in which every node has either 2 children (left child and right child) or 0 children. A node with 0 children is called a leaf, and a node with 2 children is called a non-leaf.
Expressing Win-Loss Sequences as Full Binary Trees

Structure of full binary trees and win-loss sequences is the same:
- Each non-leaf yields two nodes in the next level,
Structure of full binary trees and win-loss sequences is the same:

- Each non-leaf yields two nodes in the next level,
- Each win yields two cards in the next passthrough.
Expressing Win-Loss Sequences as Full Binary Trees

Structure of full binary trees and win-loss sequences is the same:

- Each non-leaf yields two nodes in the next level,
- Each win yields two cards in the next passthrough.
- Each leaf yields zero nodes in the next level,
Structure of full binary trees and win-loss sequences is the same:

- Each non-leaf yields two nodes in the next level,
- Each win yields two cards in the next passthrough.
- Each leaf yields zero nodes in the next level,
- Each loss yields zero cards in the next passthrough.
Expressing Win-Loss Sequences as Full Binary Trees

Structure of full binary trees and win-loss sequences is the same:

- Each non-leaf yields two nodes in the next level,
- Each win yields two cards in the next passthrough.
- Each leaf yields zero nodes in the next level,
- Each loss yields zero cards in the next passthrough.
- **non-leaves = W’s, leaves = L’s, levels of FBT = passthroughs**
From Win-Loss Sequence to Full Binary Tree

- Initialize with a single unlabelled node.
- Write down the first passthrough in this unlabelled node.
- For each node labelled W, create two unlabelled children nodes underneath the node.
- Write down each subsequent passthrough in the unlabelled nodes of the next level left-to-right.

Finding FBT for W/WW/LWLL/LL

Atharva Pathak (MIT PRIMES)
From Win-Loss Sequence to Full Binary Tree

Finding FBT for W/WW/LWLL/LL
From Win-Loss Sequence to Full Binary Tree

- initialize with a single unlabelled node

Finding FBT for W/WW/LWLL/LL
From Win-Loss Sequence to Full Binary Tree

- initialize with a single unlabelled node

Finding FBT for W/WW/LWLL/LL
From Win-Loss Sequence to Full Binary Tree

- initialize with a single unlabelled node
- write down the first passthrough in this unlabelled node

Finding FBT for W/WW/LWLL/LL
From Win-Loss Sequence to Full Binary Tree

- initialize with a single unlabelled node
- write down the first passthrough in this unlabelled node

Finding FBT for W/WW/LWLL/LL

```
   W
     
    o   o
```

Atharva Pathak (MIT PRIMES)
From Win-Loss Sequence to Full Binary Tree

- initialize with a single unlabelled node
- write down the first passthrough in this unlabelled node
- for each node labelled $W$, create two unlabelled children nodes underneath the node

Finding FBT for $W/WW/LWLL/LL$

```
W
   O
  /     \
O       O
```

initialize with a single unlabelled node

write down the first passthrough in this unlabelled node

for each node labelled W, create two unlabelled children nodes underneath the node

write down each subsequent passthrough in the unlabelled nodes of the next level left-to-right

Finding FBT for W/WW/LWLL/LL
From Win-Loss Sequence to Full Binary Tree

- initialize with a single unlabelled node
- write down the first passthrough in this unlabelled node
- for each node labelled $W$, create two unlabelled children nodes underneath the node
- write down each subsequent passthrough in the unlabelled nodes of the next level left-to-right

Finding FBT for $W/WW/LWLL/LL$

```
  W
 / \
W   W
 / \ / \ \
•   •   •   •
```

```
  W
 / \
W   W
 / \ / \ \
•   •   •   •
```

Atharva Pathak (MIT PRIMES)
From Win-Loss Sequence to Full Binary Tree

- initialize with a single unlabelled node
- write down the first passthrough in this unlabelled node
- for each node labelled $W$, create two unlabelled children nodes underneath the node
- write down each subsequent passthrough in the unlabelled nodes of the next level left-to-right

Finding FBT for $W/WW/LWLL/LL$
From Win-Loss Sequence to Full Binary Tree

- initialize with a single unlabelled node
- write down the first passthrough in this unlabelled node
- for each node labelled $W$, create two unlabelled children nodes underneath the node
- write down each subsequent passthrough in the unlabelled nodes of the next level left-to-right

Finding FBT for W/WW/LWLL/LL

```
W  W
  W
    W
      W
        W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```

```
W  W
  W
    W
      W
```
From Win-Loss Sequence to Full Binary Tree

- initialize with a single unlabelled node
- write down the first passthrough in this unlabelled node
- for each node labelled W, create two unlabelled children nodes underneath the node
- write down each subsequent passthrough in the unlabelled nodes of the next level left-to-right

Finding FBT for W/WW/LWLL/LL

---

Atharva Pathak (MIT PRIMES)
From Full Binary Tree Back To Win-Loss Sequence

write \( W \) in all non-leaves and \( L \) in all leaves read left-to-right, top-to-bottom.

\[
\begin{align*}
\text{F} & & \text{F} & & \text{F} \\
\text{W} & & \text{W} & & \text{L}
\end{align*}
\]

Reading left-to-right, top-to-bottom, we get \( W/W/W/LW/LL \).
From Full Binary Tree Back To Win-Loss Sequence

•

F

•

F

F

•

•

L

L

L

Reading left-to-right, top-to-bottom, we get W/W/WW/LWLL/LL.
write $W$ in all non-leaves and $L$ in all leaves
• write $W$ in all non-leaves and $L$ in all leaves
• read left-to-right, top-to-bottom
From Full Binary Tree Back To Win-Loss Sequence

- write $W$ in all non-leaves and $L$ in all leaves
- read left-to-right, top-to-bottom

\[
\begin{array}{c}
\text{\begin{tikzpicture}
        \node (root) at (0,0) {•};
        \node (left) at (-1,-1) {•};
        \node (right) at (1,-1) {•};
        \node (leftleft) at (-2,-2) {L};
        \node (lefright) at (-1,-2) {L};
        \node (rightleft) at (0,-2) {W};
        \node (rightright) at (1,-2) {L};
      
        \draw (root) -- (left);
        \draw (root) -- (right);
        \draw (left) -- (leftleft);
        \draw (left) -- (lefright);
        \draw (right) -- (rightleft);
        \draw (right) -- (rightright);
      \end{tikzpicture}}
\end{array}
\quad
\begin{array}{c}
\text{\begin{tikzpicture}
        \node (root) at (0,0) {W};
        \node (left) at (-1,-1) {W};
        \node (right) at (1,-1) {W};
        \node (leftleft) at (-2,-2) {L};
        \node (lefright) at (-1,-2) {L};
        \node (rightleft) at (0,-2) {L};
        \node (rightright) at (1,-2) {L};
      
        \draw (root) -- (left);
        \draw (root) -- (right);
        \draw (left) -- (leftleft);
        \draw (left) -- (lefright);
        \draw (right) -- (rightleft);
        \draw (right) -- (rightright);
      \end{tikzpicture}}
\end{array}
\]

Reading left-to-right, top-to-bottom, we get $W/LWLL$. 
write $W$ in all non-leaves and $L$ in all leaves
read left-to-right, top-to-bottom

Reading left-to-right, top-to-bottom, we get $W/WW/LWLL/LL$. 
Recall levels of a full binary tree are passthroughs of a win-loss sequence.

**Proposition**

The number of win-loss sequences $A_k$ that end within $k$ passthroughs for Alice satisfies $A_1 = 1$ and $A_{k+1} = A_k^2 + 1$.

**Proposition**

The number of win-loss sequences $B_k$ that end in exactly $2k + 1$ rounds is $C_k$, where $C_k$ is the $k$’th Catalan number $\frac{1}{k+1}\binom{2k}{k}$. 

Atharva Pathak (MIT PRIMES)
Poset for Necessarily Following a Win-Loss Sequence
What relations between the cards in $a_1|a_2a_3\ldots a_{10}$ are there to necessarily follow $W/WW/LWLL/LL$?
What relations between the cards in $a_1|a_2|a_3\ldots a_{10}$ are there to necessarily follow $W/WW/LWLL/LL$?

Tree Labelled With $W$’s and $L$’s
What relations between the cards in \(a_1|a_2a_3\ldots a_{10}\) are there to necessarily follow \(W/WW/LWLL/LL\)?

Tree Labelled With \(W\)’s and \(L\)’s  

Poset for Random Putback
Continued Research

- Counting states that end in a certain number of rounds with WL-putback
Acknowledgements

- My mentor Dr. Tanya Khovanova
- Dr. Slava Gerovitch, Prof. Pavel Etingof, and the entire PRIMES program


Michael Z. Spivey. “Cycles in war.”. eng. In: *Integers* 10.6 (2010),