

Computability in Anonymous Networks: Revocable vs. Irrevocable Outputs

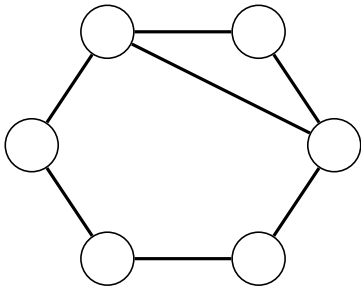


*Yuval Emek*¹ ***Jochen Seidel***² *Roger Wattenhofer*²

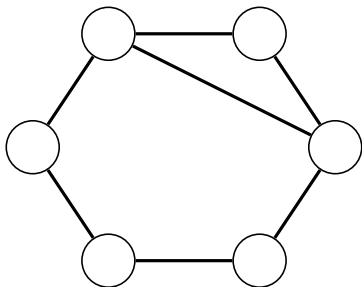
¹ *Technion – Faculty of Industrial Engineering and Management – ie.technion.ac.il*

² *ETH Zurich – Distributed Computing Group – www.disco.ethz.ch*

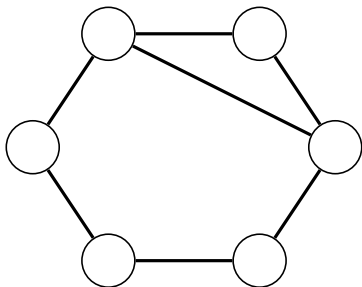
Anonymous Networks



Computability in Anonymous Networks



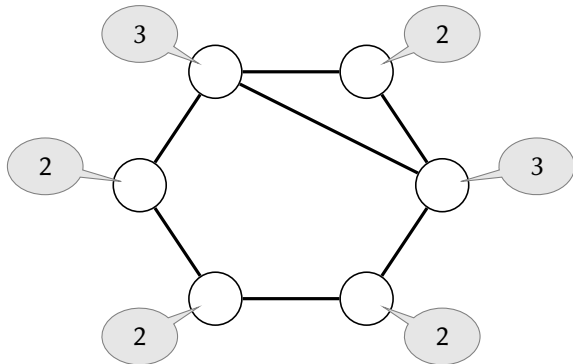
Computability in Anonymous Networks



Computable

Not Computable

Computability in Anonymous Networks

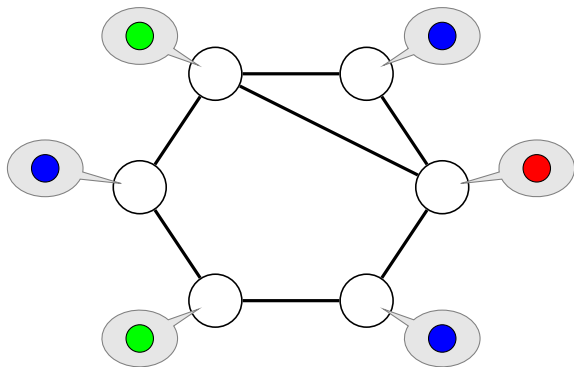


Computable

► Degree

Not Computable

Computability in Anonymous Networks

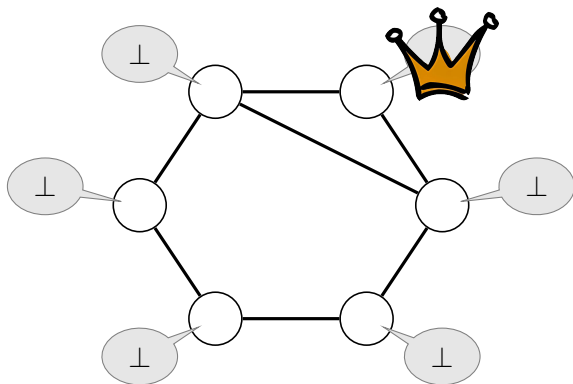


Computable

- ▶ Degree
- ▶ Coloring
- ▶ ...

Not Computable

Computability in Anonymous Networks



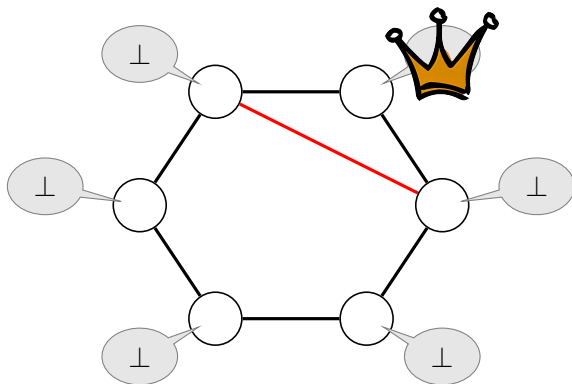
Computable

- ▶ Degree
- ▶ Coloring
- ▶ ...

Not Computable

- ▶ Leader Election

Computability in Anonymous Networks



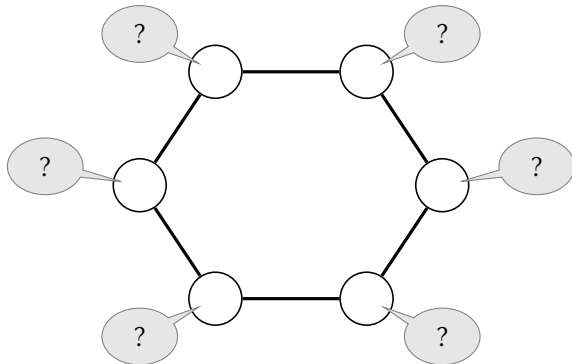
Computable

- Degree
- Coloring
- ...

Not Computable

- Leader Election

Computability in Anonymous Networks



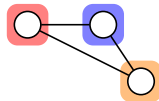
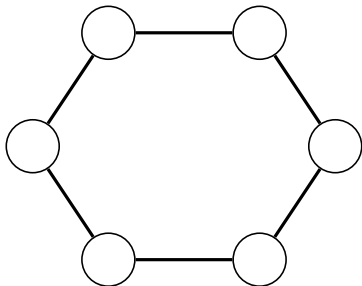
Computable

- ▶ Degree
- ▶ Coloring
- ▶ ...

Not Computable

- ▶ Leader Election

Computability in Anonymous Networks



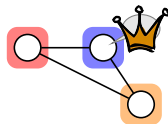
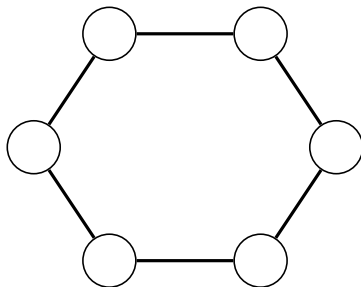
Computable

- ▶ Degree
- ▶ Coloring
- ▶ ...

Not Computable

- ▶ Leader Election

Computability in Anonymous Networks



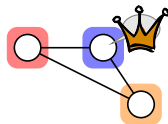
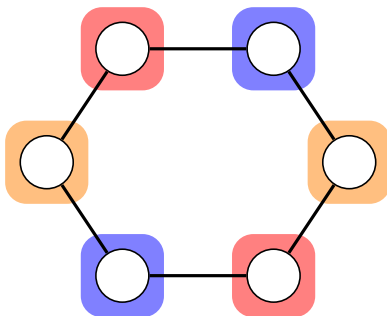
Computable

- ▶ Degree
- ▶ Coloring
- ▶ ...

Not Computable

- ▶ Leader Election

Computability in Anonymous Networks



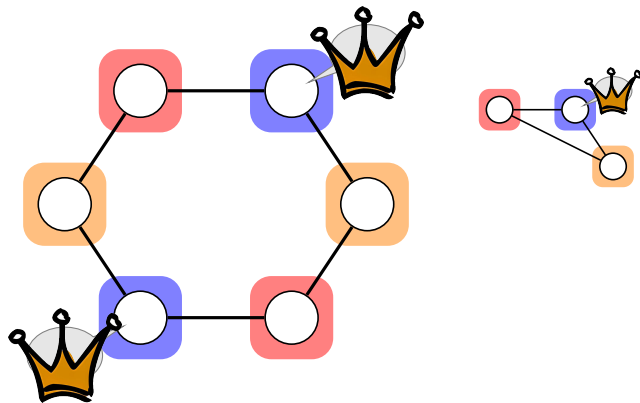
Computable

- ▶ Degree
- ▶ Coloring
- ▶ ...

Not Computable

- ▶ Leader Election

Computability in Anonymous Networks



Computable

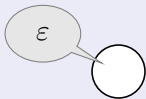
- ▶ Degree
- ▶ Coloring
- ▶ ...

Not Computable

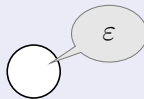
- ▶ Leader Election
- ▶ Network Size
- ▶ ...

Revocable vs. Irrevocable Outputs

Irrevocable

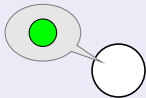


Revocable

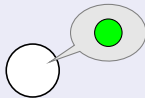


Revocable vs. Irrevocable Outputs

Irrevocable

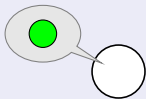


Revocable

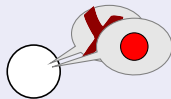


Revocable vs. Irrevocable Outputs

Irrevocable

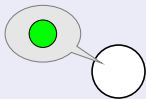


Revocable

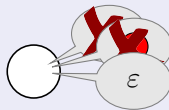


Revocable vs. Irrevocable Outputs

Irrevocable

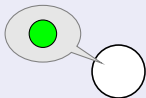


Revocable



Revocable vs. Irrevocable Outputs

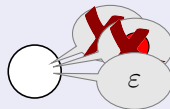
Irrevocable



Write-Once

- ▶ **WO**-algorithms
- ▶ Problem Class **WO**

Revocable



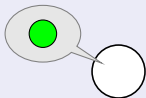
Re-Write

- ▶ **RW**-algorithm
- ▶ Problem Class **RW**

$$\mathbf{WO} \subseteq \mathbf{RW}$$

Revocable vs. Irrevocable Outputs

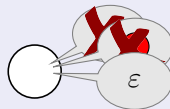
Irrevocable



Write-Once

- ▶ **WO**-algorithms
- ▶ Problem Class **WO**

Revocable

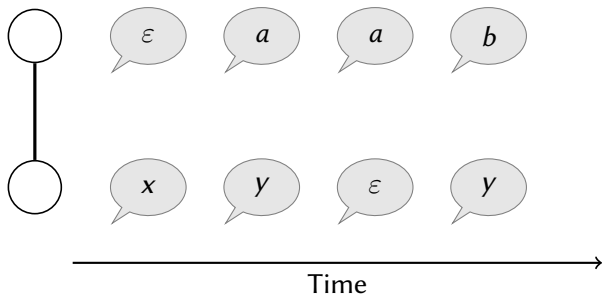


Re-Write

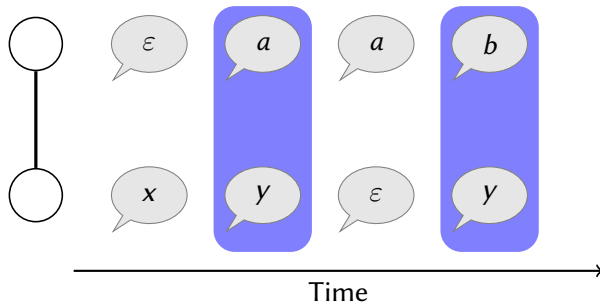
- ▶ **RW**-algorithm
- ▶ Problem Class **RW**

$$\mathbf{WO} \subsetneq \mathbf{RW}$$

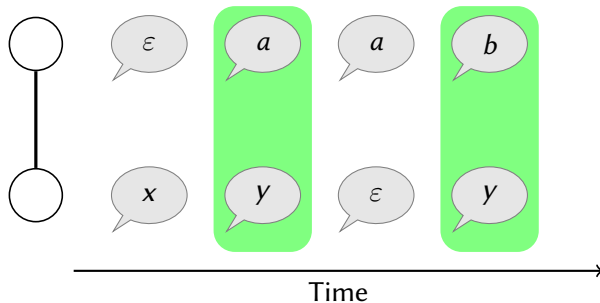
RW: How to Specify the Network's Output?



Ready Configurations



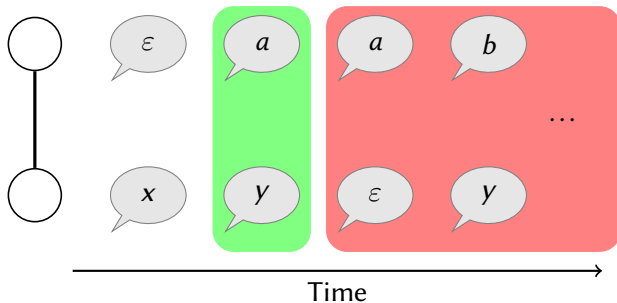
Correctness Notions



Default

(Every ready configuration correct)

Correctness Notions



Default

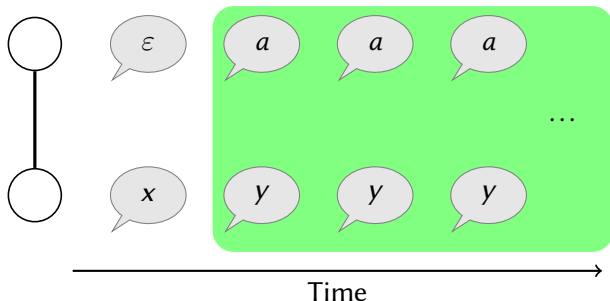
(Every ready configuration correct)



Loose

(First ready configuration correct)

Correctness Notions



Sustainable

(Output irrevocable after first ready configuration)



Default

(Every ready configuration correct)



Loose

(First ready configuration correct)

Correctness Notions

Theorem (Sustainable vs. Loose Correctness)

*The following three correctness notions are equivalent for **RW**:*

- | | | |
|--|-----------------------|---|
|  | 1. <i>Sustainable</i> | <i>(Output irrevocable after first ready configuration)</i> |
|  | 2. <i>Default</i> | <i>(Every ready configuration correct)</i> |
|  | 3. <i>Loose</i> | <i>(First ready configuration correct)</i> |

Correctness Notions

1. sustainable-**RW** $\subseteq$ **RW** $\subseteq$ loose-**RW**

Theorem (Sustainable vs. Loose Correctness)

*The following three correctness notions are equivalent for **RW**:*

- | | | |
|--|----------------|--|
|  | 1. Sustainable | (Output irrevocable after first ready configuration) |
|  | 2. Default | (Every ready configuration correct) |
|  | 3. Loose | (First ready configuration correct) |

Correctness Notions

1. sustainable-**RW** $\subseteq$ **RW** $\subseteq$ loose-**RW** ✓

Theorem (Sustainable vs. Loose Correctness)

*The following three correctness notions are equivalent for **RW**:*



1. *Sustainable*

(Output irrevocable after first ready configuration)



2. *Default*

(Every ready configuration correct)



3. *Loose*

(First ready configuration correct)

Correctness Notions

1. sustainable-**RW** $\subseteq$ **RW** $\subseteq$ loose-**RW** ✓
2. loose-**RW** $\subseteq$ sustainable-**RW** ?

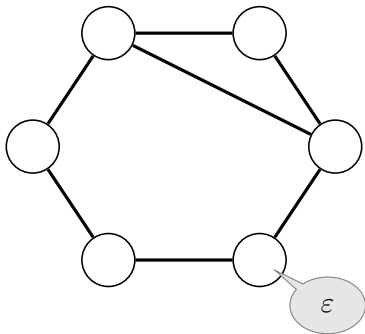
Theorem (Sustainable vs. Loose Correctness)

*The following three correctness notions are equivalent for **RW**:*

- | | | |
|--|----------------|--|
|  | 1. Sustainable | (Output irrevocable after first ready configuration) |
|  | 2. Default | (Every ready configuration correct) |
|  | 3. Loose | (First ready configuration correct) |

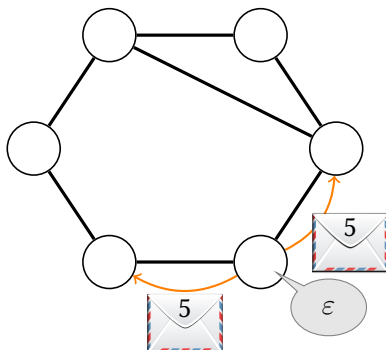
Proof Idea

1. sustainable-RW $\subseteq$ RW $\subseteq$ loose-RW $\checkmark$
2. loose-RW $\subseteq$ sustainable-RW ?



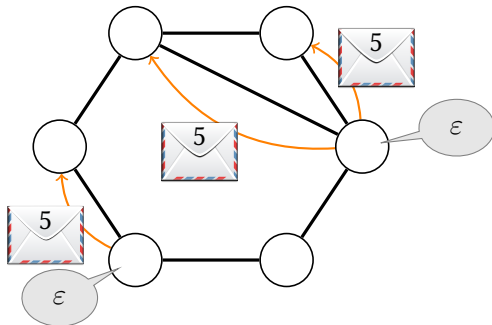
Proof Idea

1. sustainable-RW $\subseteq$ RW $\subseteq$ loose-RW $\checkmark$
2. loose-RW $\subseteq$ sustainable-RW ?



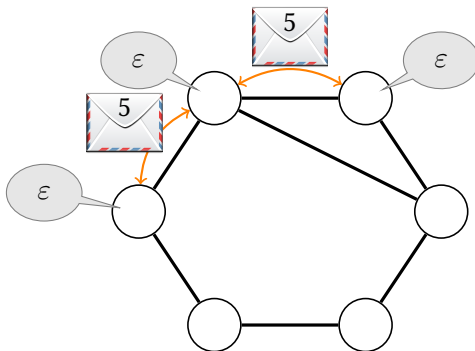
Proof Idea

1. sustainable-RW $\subseteq$ RW $\subseteq$ loose-RW $\checkmark$
2. loose-RW $\subseteq$ sustainable-RW ?



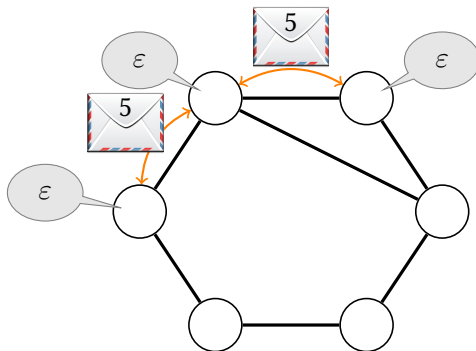
Proof Idea

1. sustainable-RW $\subseteq$ RW $\subseteq$ loose-RW $\checkmark$
2. loose-RW $\subseteq$ sustainable-RW ?



Proof Idea

1. sustainable-RW $\subseteq$ RW $\subseteq$ loose-RW $\checkmark$
2. loose-RW $\subseteq$ sustainable-RW ?



- ▶ Round 5 is *inhibited*.
- ▶ Pick output from first non-inhibited round
 $\implies$ sustainable correctness.

Proof Idea

1. sustainable-**RW** $\subseteq$ **RW** $\subseteq$ loose-**RW** ✓
2. loose-**RW** $\subseteq$ sustainable-**RW** ✓

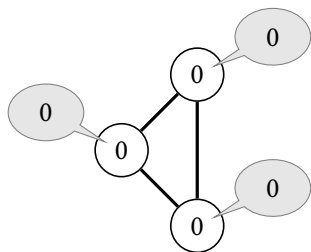
Theorem (Sustainable vs. Loose Correctness)

*The following three correctness notions are equivalent for **RW**:*

- | | | |
|--|----------------|--|
|  | 1. Sustainable | (Output irrevocable after first ready configuration) |
|  | 2. Default | (Every ready configuration correct) |
|  | 3. Loose | (First ready configuration correct) |

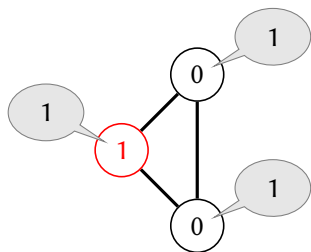
Logical OR is Computable

With Revocable Outputs



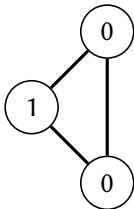
Logical OR is Computable

With Revocable Outputs



Logical OR is Computable

With Revocable Outputs



RW-algorithm for OR

Round 1: Output 0 if input = 0.

Round 2: Output 1.

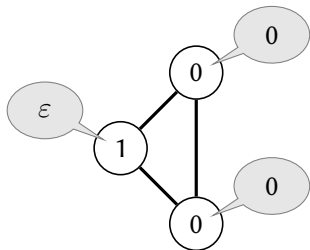


Loose

(First ready configuration correct)

Logical OR is Computable

With Revocable Outputs



RW-algorithm for OR

Round 1: Output 0 if input = 0.

Round 2: Output 1.

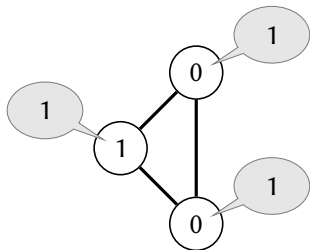


Loose

(First ready configuration correct)

Logical OR is Computable

With Revocable Outputs



RW-algorithm for OR

Round 1: Output 0 if input = 0.

Round 2: Output 1.

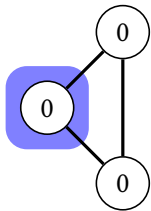


Loose

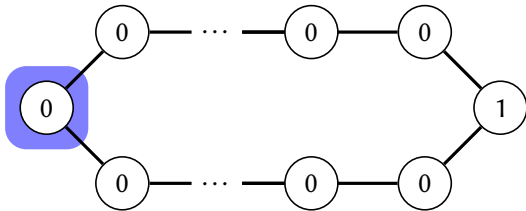
(First ready configuration correct)

Logical OR is Not Computable

With Irrevocable Outputs

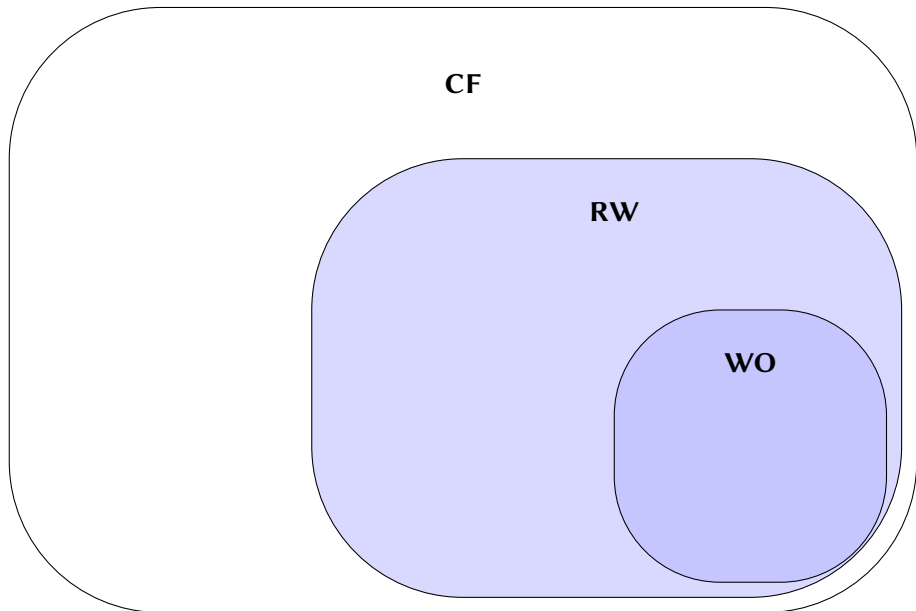


vs.



Problem Zoo

I've got 21 problems ...



Problem Zoo

I've got 21 problems ...

CF

MIN-COLORING
(≥ 3)-HOP-COLORING
(≥ 3)-HOP-MIS
FACTOR-MULTIPLICITY
SIZE-APX
LEADER-ELECTION
MIN-CUT
UNIQUENESS
IDS
DIAMETER
DIAMETER-APX

RW

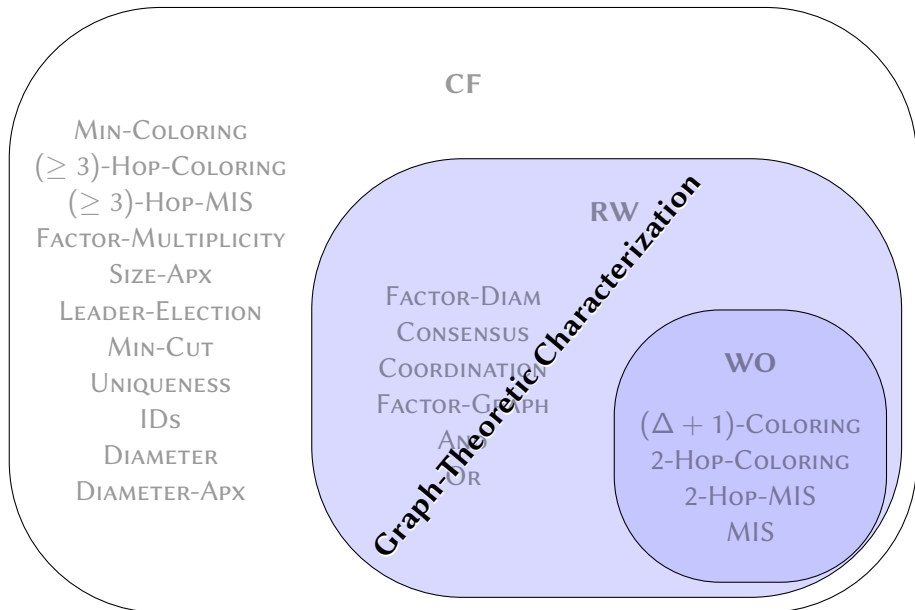
FACTOR-DIAM
CONSENSUS
COORDINATION
FACTOR-GRAPH
AND
OR

WO

($\Delta + 1$)-COLORING
2-HOP-COLORING
2-HOP-MIS
MIS

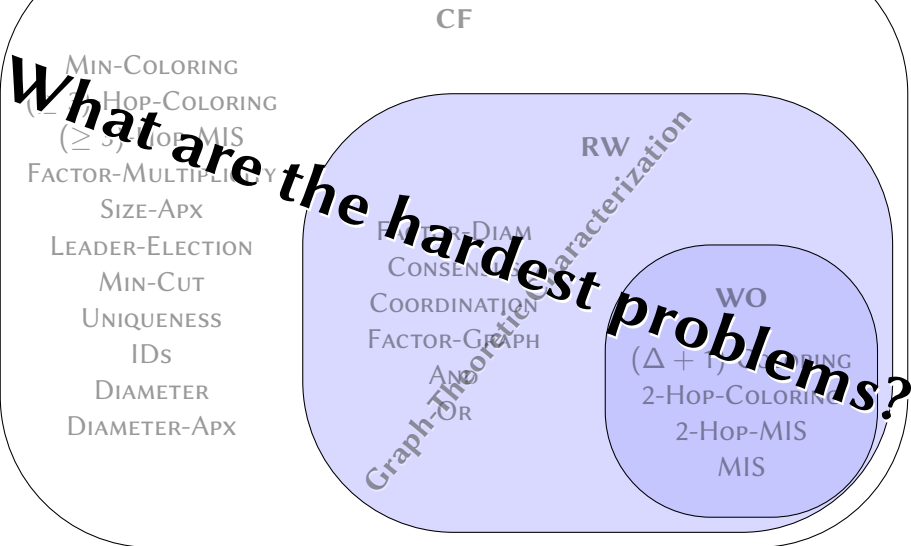
Problem Zoo

I've got 21 problems ...



Problem Zoo

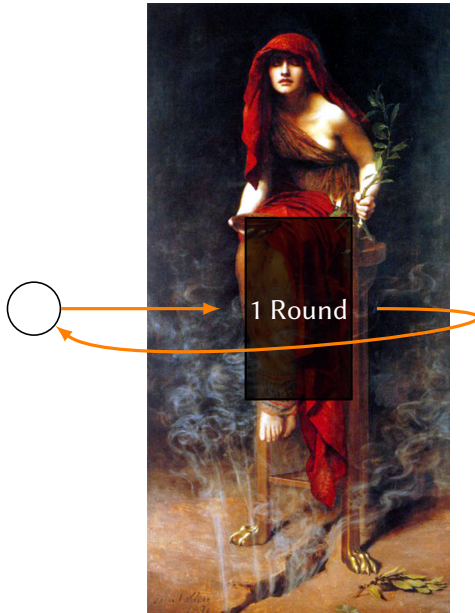
I've got 21 problems ...



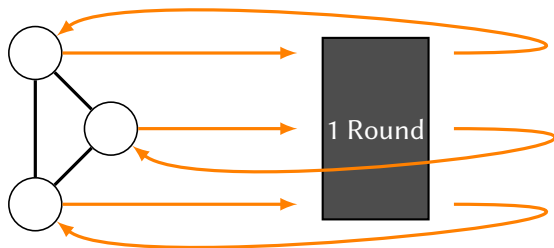
An Oracle



An Oracle



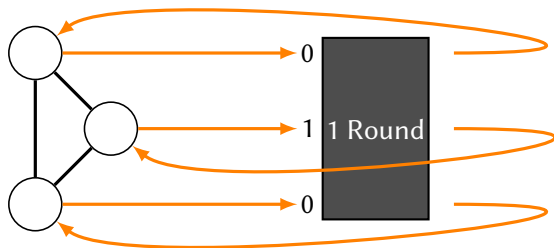
Accessing a Distributed Oracle



In each round:

- ▶ Set oracle input
(for next round)
- ▶ Get oracle answer
(from previous round)

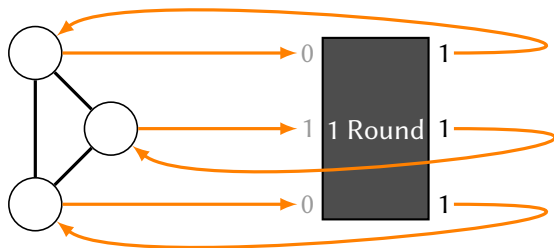
Accessing a Distributed Oracle



In each round:

- ▶ Set oracle input
(for next round)
- ▶ Get oracle answer
(from previous round)

Accessing a Distributed Oracle



In each round:

- ▶ Set oracle input
(for next round)
- ▶ Get oracle answer
(from previous round)

Distributed Oracles

An oracle does not help if you could answer the question yourself.

Theorem

Fix a class $\mathbf{C} \in \{\mathbf{RW}, \mathbf{WO}\}$, and let $Q \in \mathbf{C}$.

P solvable by $\mathbf{C}$ -algorithm accessing Q -oracle

$\implies P$ solvable by $\mathbf{C}$ -algorithm (without any oracle access).

Hard and Complete Problems

Hardness

Fix two classes $\mathbf{B} \supseteq \mathbf{C}$, and a problem Q .

$Q \in \mathbf{B-hard}_{\mathbf{C}}$, if

- ▶ for every $P \in \mathbf{B}$:
- ▶ there is a $\mathbf{C}$ -algorithm solving P with access to a Q -oracle

$$\text{“}\mathcal{NP}\text{-hard} = \mathcal{NP}\text{-hard}_{\mathcal{P}}\text{”}$$

Hard and Complete Problems

Hardness

Fix two classes $\mathbf{B} \supseteq \mathbf{C}$, and a problem Q .

$Q \in \mathbf{B-hard}_{\mathbf{C}}$, if

- ▶ for every $P \in \mathbf{B}$:
- ▶ there is a $\mathbf{C}$ -algorithm solving P with access to a Q -oracle

$$\text{“}\mathcal{NP}\text{-hard} = \mathcal{NP}\text{-hard}_P\text{”}$$

- ▶ $\mathbf{CF-hard}_{\mathbf{RW}}$
- ▶ $\mathbf{CF-hard}_{\mathbf{WO}}$
- ▶ $\mathbf{RW-hard}_{\mathbf{WO}}$

Problem Zoo

I've got 21 problems ...

CF

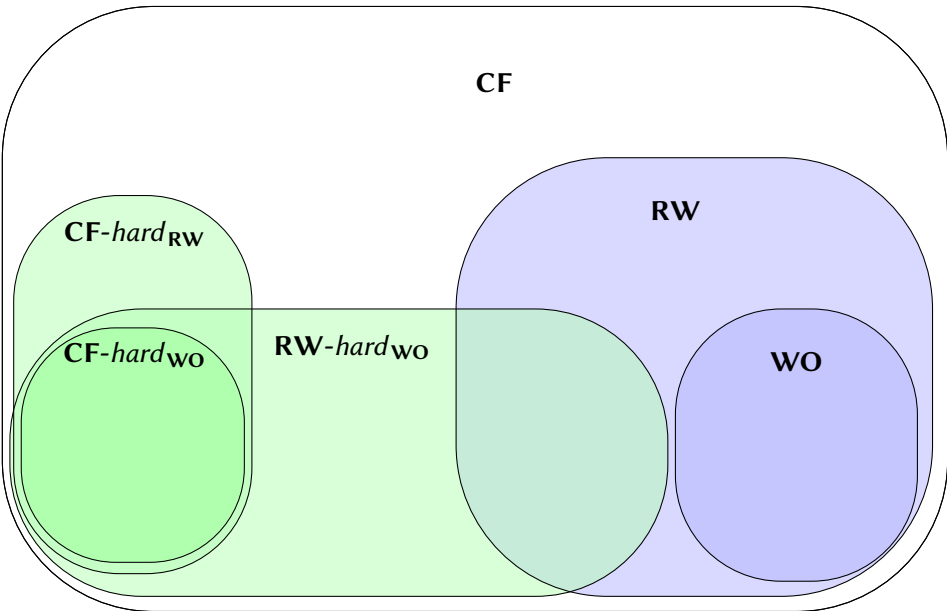
RW

CF-hard_{RW}

CF-hard_{WO}

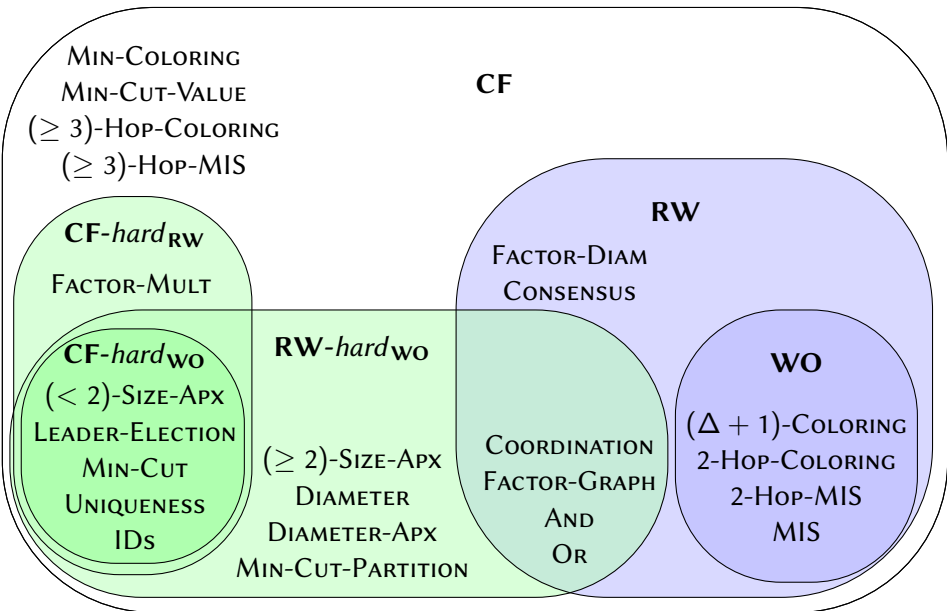
RW-hard_{WO}

WO



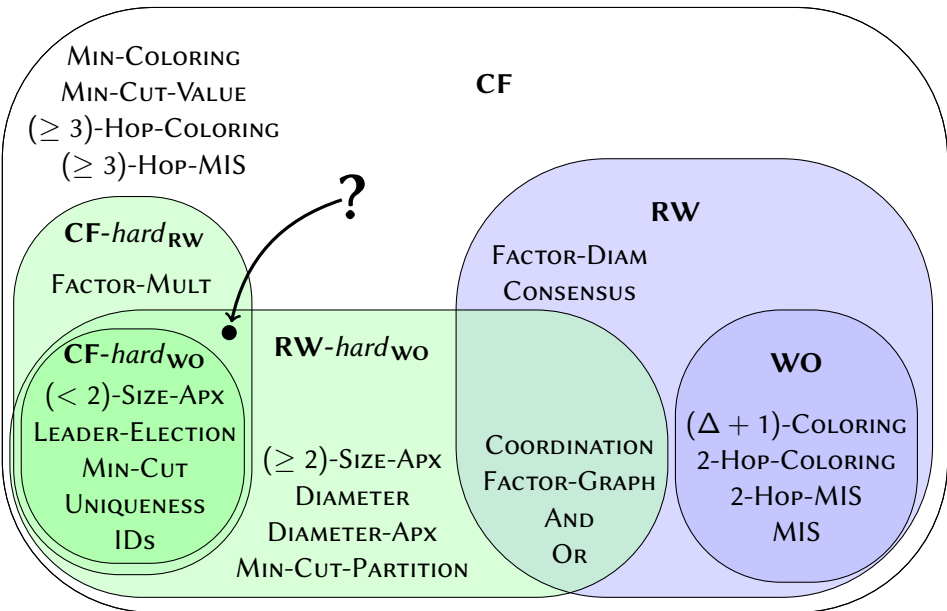
Problem Zoo

I've got 24 problems ...



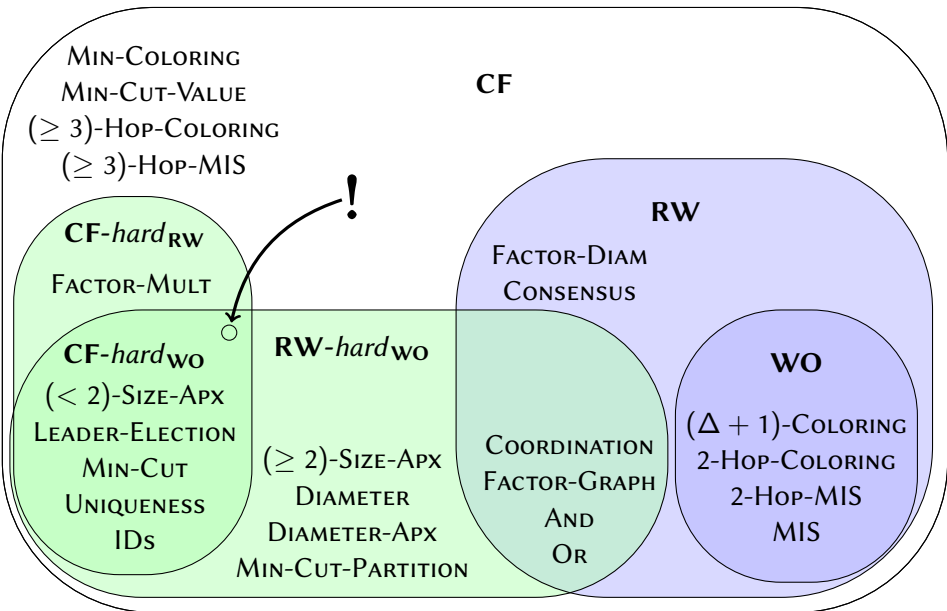
Problem Zoo

I've got 24 problems ...

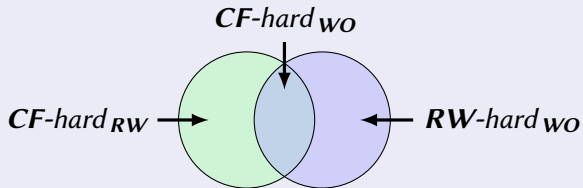


Problem Zoo

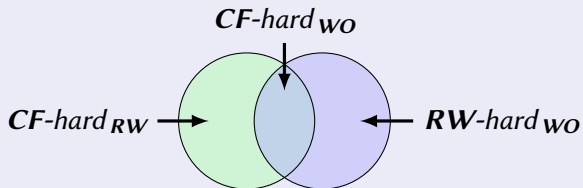
I've got 24 problems ...



Theorem

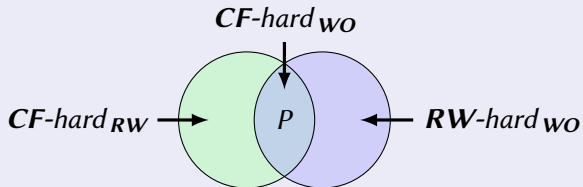


Theorem



Interesting Part: $CF-hard_{wo} \supseteq CF-hard_{RW} \cap RW-hard_{wo}$

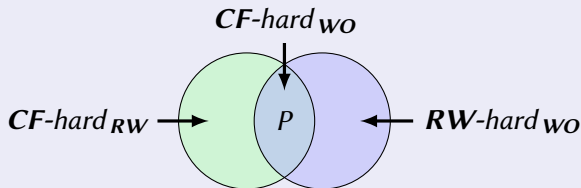
Theorem



Interesting Part: $CF-hard_{wo} \supseteq CF-hard_{RW} \cap RW-hard_{wo}$

$$P \in CF-hard_{RW} \cap RW-hard_{wo}$$

Theorem



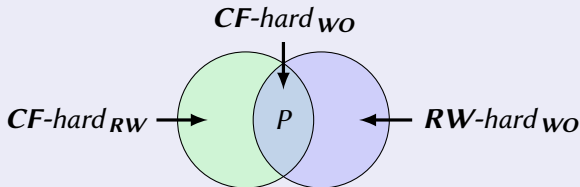
Interesting Part: $CF-hard_{wo} \supseteq CF-hard_{RW} \cap RW-hard_{wo}$

$$P \in CF-hard_{RW} \cap RW-hard_{wo}$$

With P -oracle,

- **RW**-algorithm for LEADER-ELECTION

Theorem



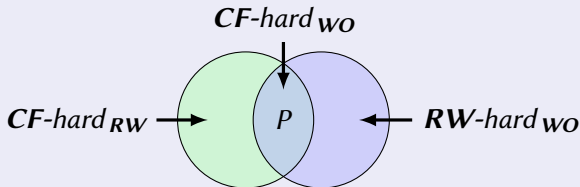
Interesting Part: $CF-hard_{WO} \supseteq CF-hard_{RW} \cap RW-hard_{WO}$

$$P \in CF-hard_{RW} \cap RW-hard_{WO}$$

With P -oracle,

- ▶ **RW**-algorithm for LEADER-ELECTION
- ▶ **WO**-algorithm for OR

Theorem



Interesting Part: $CF-hard_{WO} \supseteq CF-hard_{RW} \cap RW-hard_{WO}$

$$P \in CF-hard_{RW} \cap RW-hard_{WO}$$

With P -oracle,

- ▶ RW -algorithm for LEADER-ELECTION
 - ▶ WO -algorithm for OR
- } WO -algorithm

Summary

- ▶ 3 Correctness Notions for Revocable Outputs
 - ▶ Equivalent
- ▶ 3 Hardness Classes
 - ▶ 3rd is Intersection of the Other Two
- ▶ 21 – 24 Problems
 - ▶ Completely Characterized