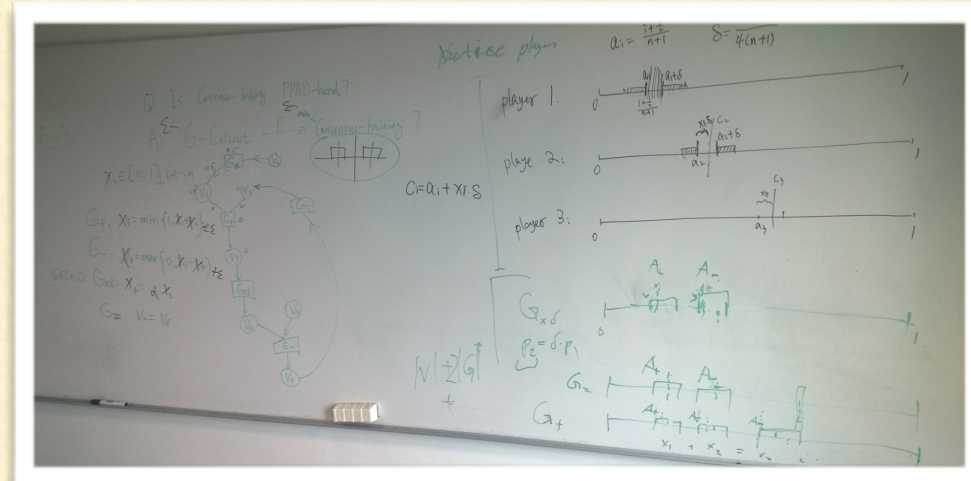
UNIVERSITY OF LIVERPOOL



# THE COMPLEXITY OF NECKLACE SPLITTING, CONSENSUS-HALVING AND DISCRETE HAM SANDWICH

From the papers:

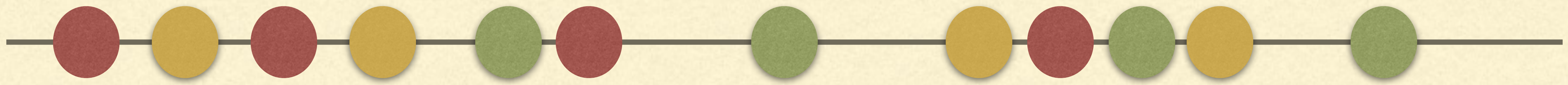
Consensus-Halving is PPA-Complete (STOC 2018).

The Complexity of Splitting Necklaces and Bisecting Ham Sandwiches (STOC 2019).

joint works with with P. W. Goldberg.

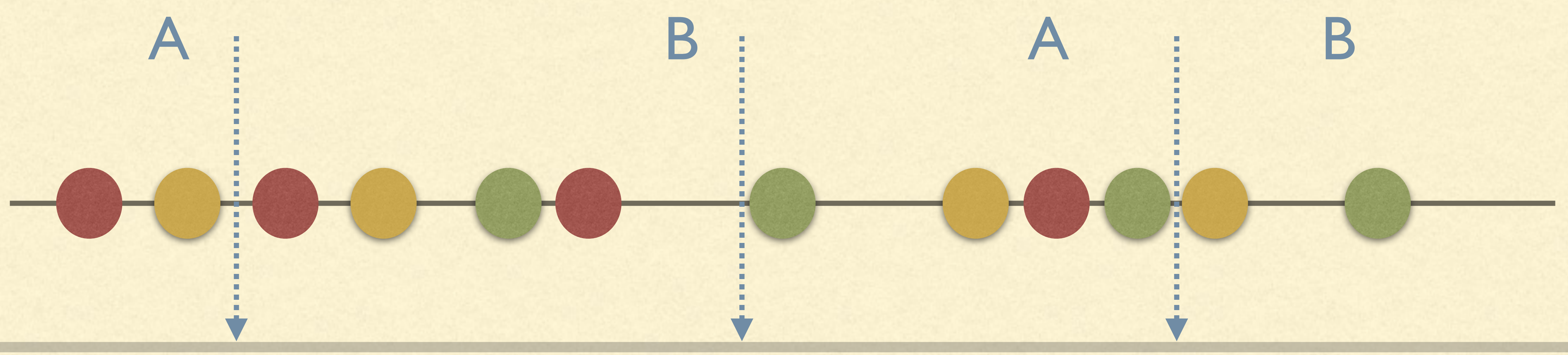
# NECKLACE SPLITTING (WITH TWO THIEVES)

- An open necklace with an even number of beads of each of n colours.
- Cut the necklace into parts using n cuts.
- Assign a label (A or B) to each part (the name of the thief that gets it).
- Goal: A partition such that A and B have the same number of beads of each colour.



# NECKLACE SPLITTING (WITH TWO THIEVES)

- An open necklace with an even number of beads of each of n colours.
- Cut the necklace into parts using n cuts.
- Assign a label (A or B) to each part (the name of the thief that gets it).
- Goal: A partition such that A and B have the same number of beads of each colour.



# THE HISTORY OF NECKLACE SPLITTING

# THE HISTORY OF NECKLACE SPLITTING

- Alon. Splitting Necklaces (Advances in Mathematics 1987).

# THE HISTORY OF NECKLACE SPLITTING

- Alon. Splitting Necklaces (Advances in Mathematics 1987).
- Alon and West. The Borsuk-Ulam Theorem and the Bisection of Necklaces (American Mathematical Society 1986).

# THE HISTORY OF NECKLACE SPLITTING

- Alon. Splitting Necklaces (Advances in Mathematics 1987).
- Alon and West. The Borsuk-Ulam Theorem and the Bisection of Necklaces (American Mathematical Society 1986).
- Bhatt and Leiserson. How to Assemble Tree Machines (STOC 1982).

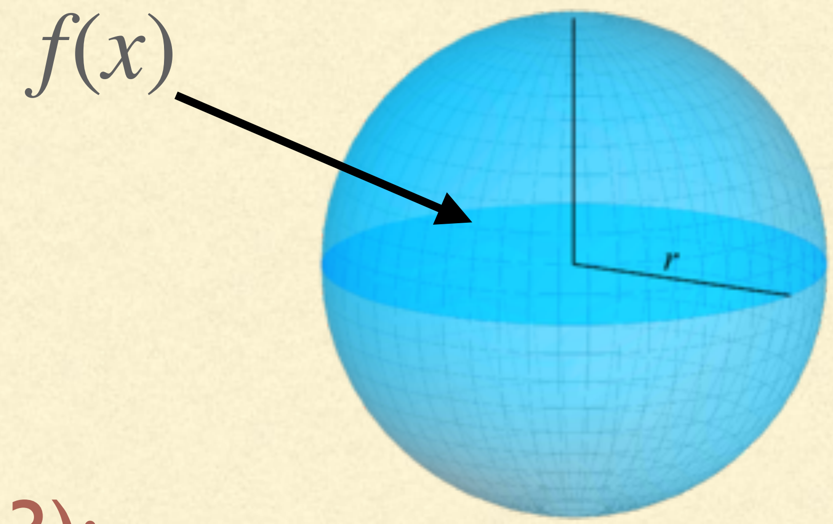
# THE HISTORY OF NECKLACE SPLITTING

- Alon. Splitting Necklaces (Advances in Mathematics 1987).
- Alon and West. The Borsuk-Ulam Theorem and the Bisection of Necklaces (American Mathematical Society 1986).
- Bhatt and Leiserson. How to Assemble Tree Machines (STOC 1982).
- Hobby and Rice. A Moment Problem in L1 Approximation (American Mathematical Society 1965).

# THE HISTORY OF NECKLACE SPLITTING

- Alon. Splitting Necklaces (Advances in Mathematics 1987).
- Alon and West. The Borsuk-Ulam Theorem and the Bisection of Necklaces (American Mathematical Society 1986).
- Bhatt and Leiserson. How to Assemble Tree Machines (STOC 1982).
- Hobby and Rice. A Moment Problem in L1 Approximation (American Mathematical Society 1965).
- Neyman. Un Théorème d' Existence (C.R. Academie de Science 1942).

# A TOTAL PROBLEM




- Total problem: A solution always exists.
- Proof by the Borsuk-Ulam Theorem (1933):
  - Let $f: S^n \to \mathbb{R}^n$ be a continuous function.
    Then, there exists x $\in S^n$ such that $f(x) = f(-x)$.

# FINDING A SOLUTION

# FINDING A SOLUTION

- Is there an efficient algorithm for finding a solution?

# FINDING A SOLUTION

- Is there an efficient algorithm for finding a solution?

**Alon. Non-constructive Proofs in Combinatorics (International Congress of Mathematicians, 1990).**

Consider, for example, the obvious algorithmic problem suggested by Theorem 1.1, namely, given a necklace satisfying the assumptions of the theorem, find a partition of it satisfying the conclusions of the theorem. This problem is in $FNP$, since it is a search problem, and given a proposed solution for it we can check in polynomial time that it is indeed a solution.

Notice that this problem always has a solution, by Theorem 1.1, and hence it seems plausible that finding one should not be a very difficult task. The situation is similar with all the other algorithmic problems corresponding to the various results mentioned here. Still, the problem of solving efficiently the corresponding search problems remains an intriguing open question.

# FINDING A SOLUTION

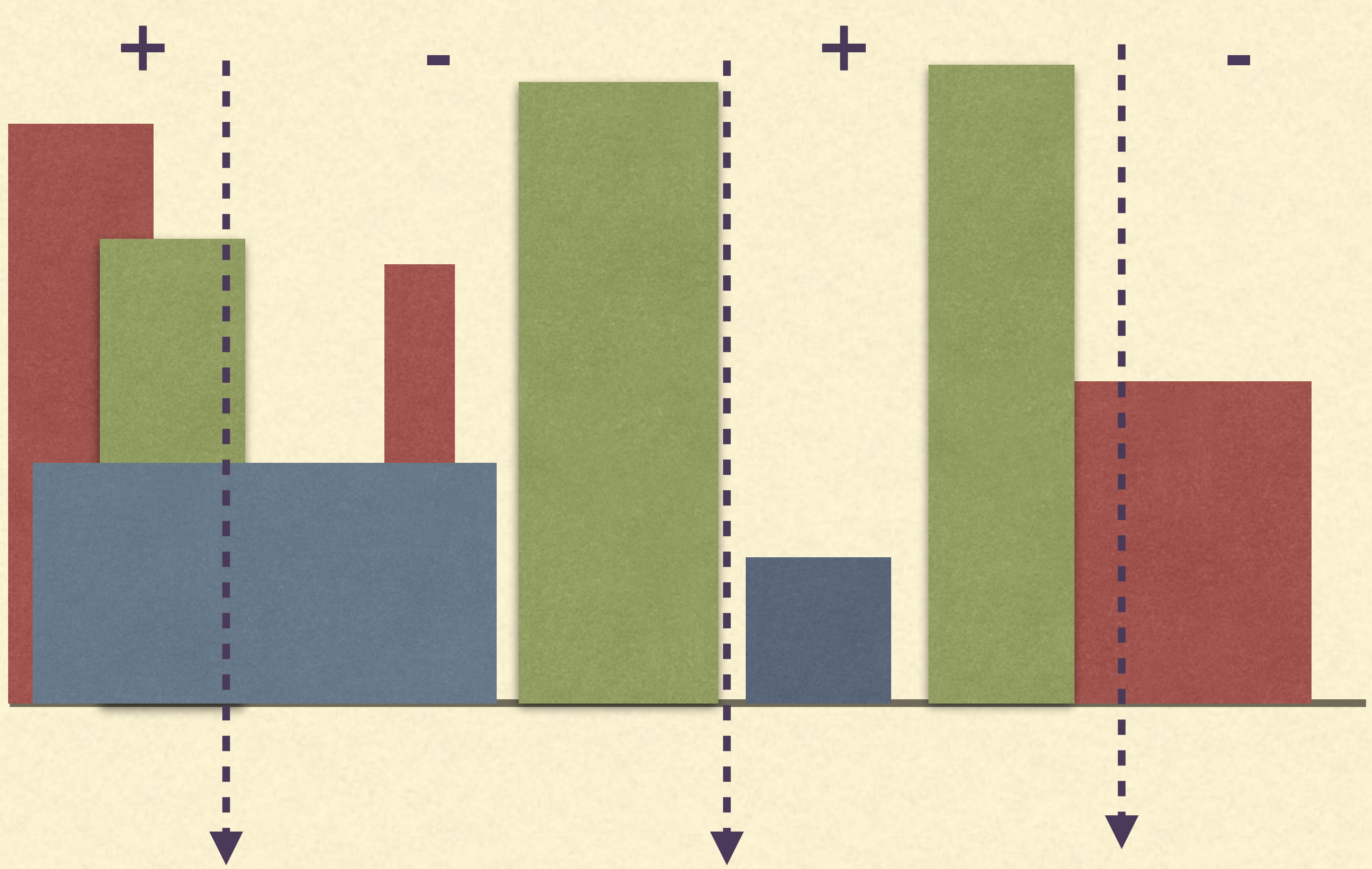
- Is there an efficient algorithm for finding a solution?
- Despite Alon's cautious optimism, no such algorithms exist!

Alon. Non-constructive Proofs in Combinatorics (International Congress of Mathematicians, 1990).

> Consider, for example, the obvious algorithmic problem suggested by Theorem 1.1, namely, given a necklace satisfying the assumptions of the theorem, find a partition of it satisfying the conclusions of the theorem. This problem is in $FNP$, since it is a search problem, and given a proposed solution for it we can check in polynomial time that it is indeed a solution.
>
> Notice that this problem always has a solution, by Theorem 1.1, and hence it seems plausible that finding one should not be a very difficult task. The situation is similar with all the other algorithmic problems corresponding to the various results mentioned here. Still, the problem of solving efficiently the corresponding search problems remains an intriguing open question.

# CONSENSUS-HALVING

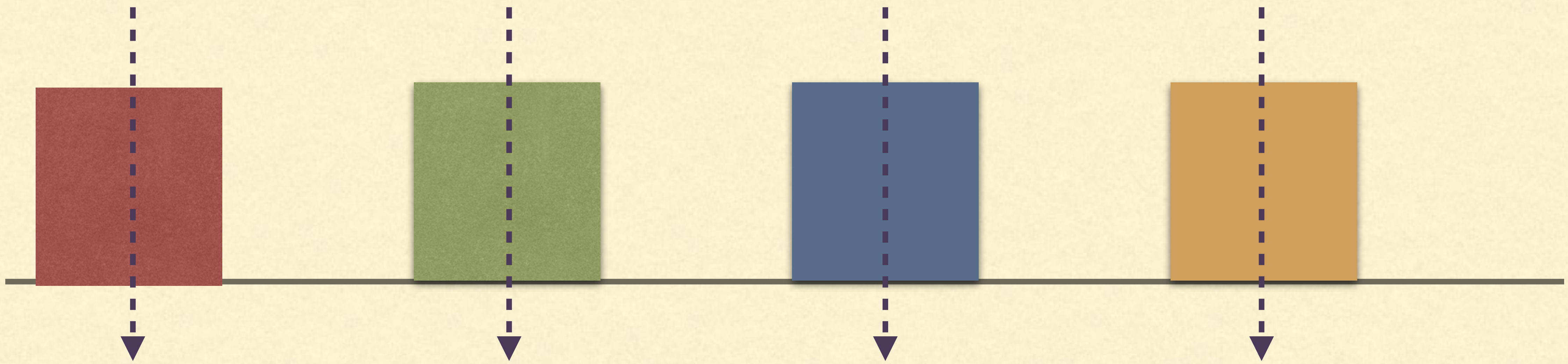
**F. Simmons and F. Su. Consensus-halving via theorems of Borsuk-Ulam and Tucker. Mathematical Social Sciences, (2003).**




- A set of n agents with valuation functions over an interval (a resource).
  - These functions are explicitly representable (in time poly(n)) and bounded.
  - Example: Piecewise constant functions.
- Halving: Cut the interval into pieces and label each piece by either (+) or (-).
- Consensus-halving: For each agent i, it holds that $v_i(+) = v_i(-)$

# CONSENSUS-HALVING

- A solution that uses n cuts is guaranteed to exist. Simmons and Su (2003).
- There are instances for which n-1 cuts are not enough. Simmons and Su (2003).



# APPROXIMATE CONSENSUS-HALVING

- For each agent i, it holds that $|v_i(+)-v_i(-)| \leq \varepsilon$

# FINDING A SOLUTION

# FINDING A SOLUTION

- Is there an efficient algorithm for finding a solution?

# FINDING A SOLUTION

- Is there an efficient algorithm for finding a solution?
- Simmons and Su's proof is constructive, but not polynomial-time.

# FINDING A SOLUTION

- Is there an efficient algorithm for finding a solution?
- Simmons and Su's proof is constructive, but not polynomial-time.
- Actually:

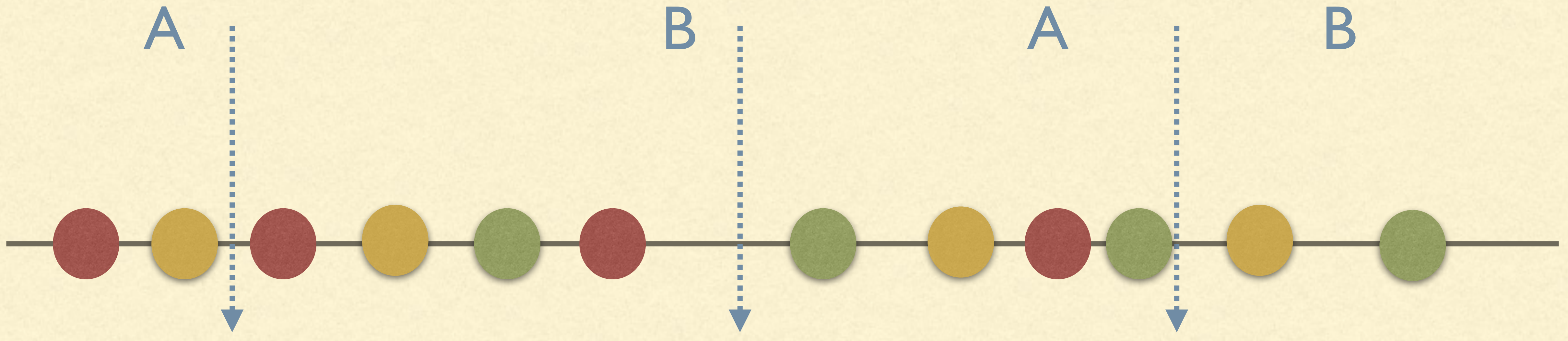
# FINDING A SOLUTION

- Is there an efficient algorithm for finding a solution?
- Simmons and Su's proof is constructive, but not polynomial-time.
- Actually:
  - Consensus-Halving is a continuous analogue of Necklace-Splitting with two thieves.

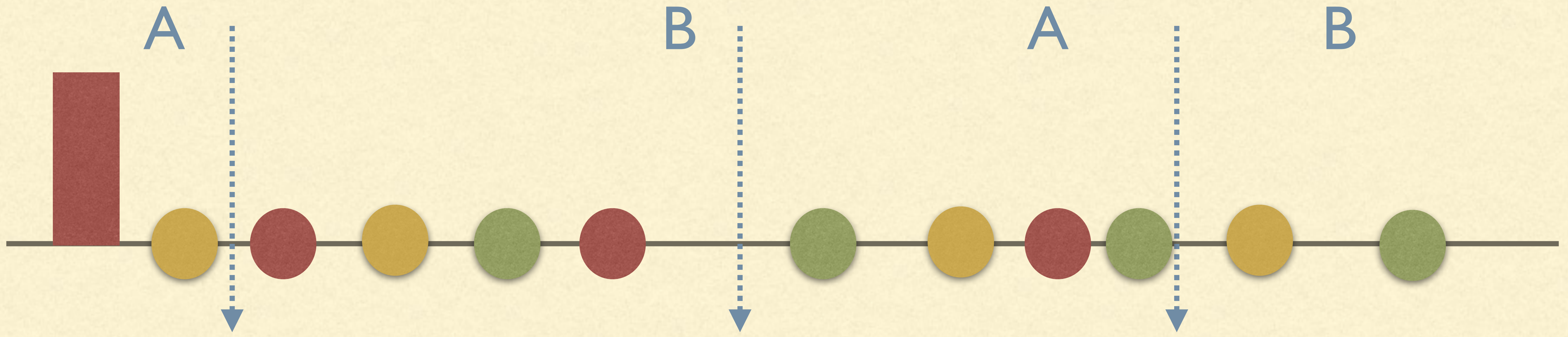
# FINDING A SOLUTION

- Is there an efficient algorithm for finding a solution?
- Simmons and Su's proof is constructive, but not polynomial-time.
- Actually:
  - Consensus-Halving is a continuous analogue of Necklace-Splitting with two thieves.
  - Alon's proof (1987) of existence for NS goes via CH.

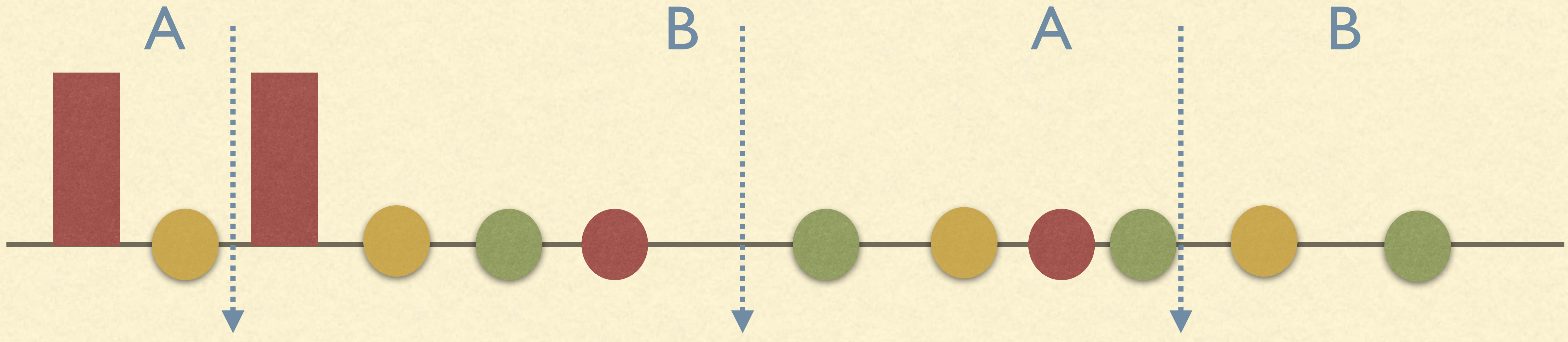
# FROM CONSENSUS-HALVING TO NECKLACE SPLITTING



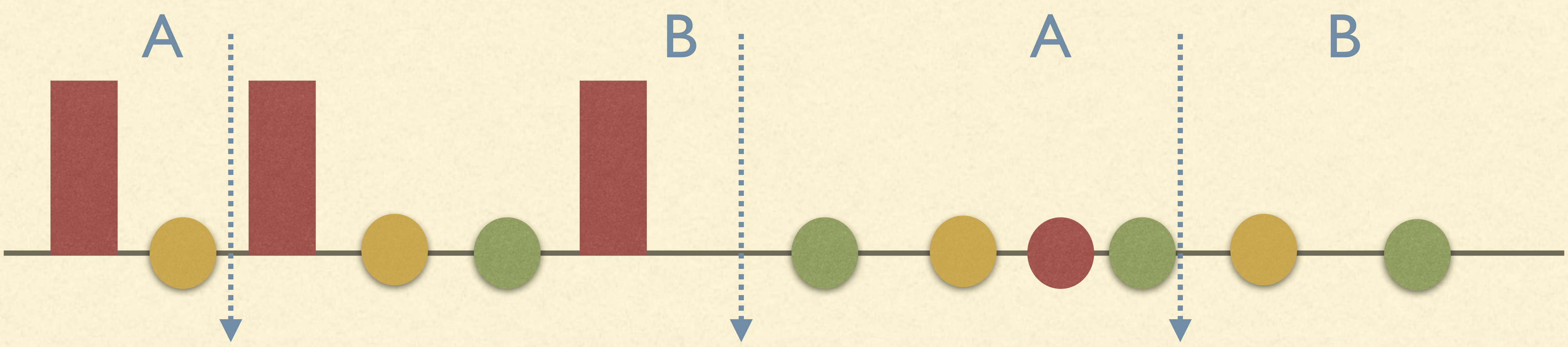
# FROM CONSENSUS-HALVING TO NECKLACE SPLITTING



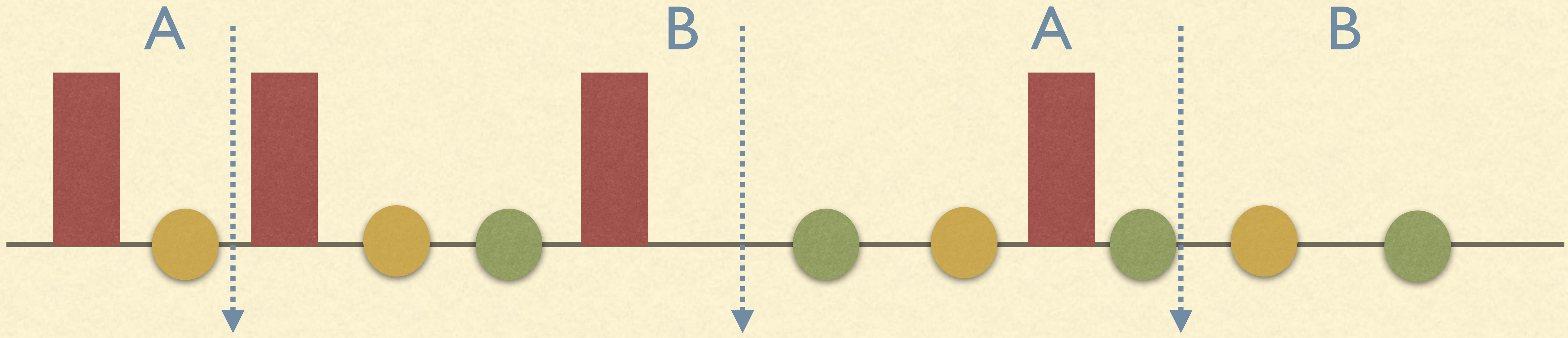
# FROM CONSENSUS-HALVING TO NECKLACE SPLITTING



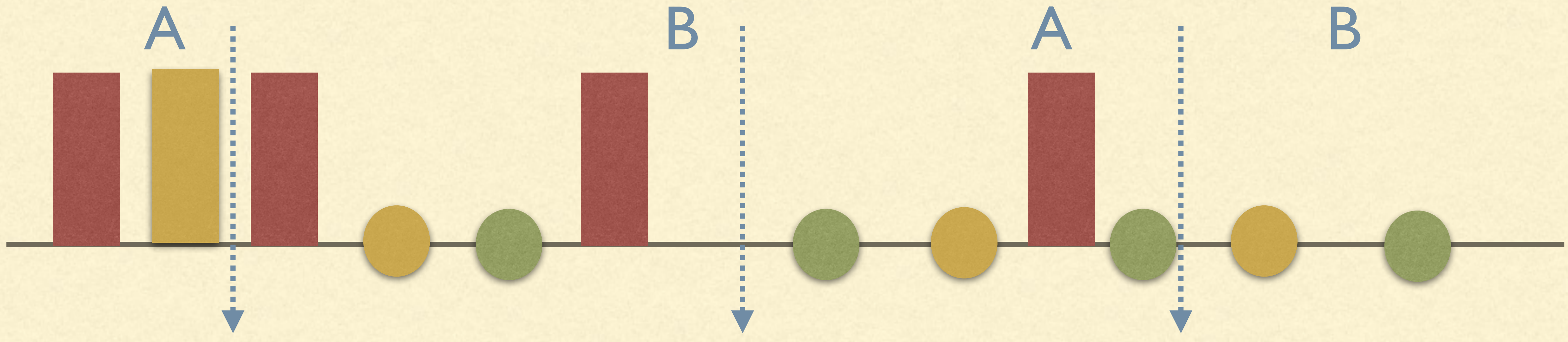
# FROM CONSENSUS-HALVING TO NECKLACE SPLITTING



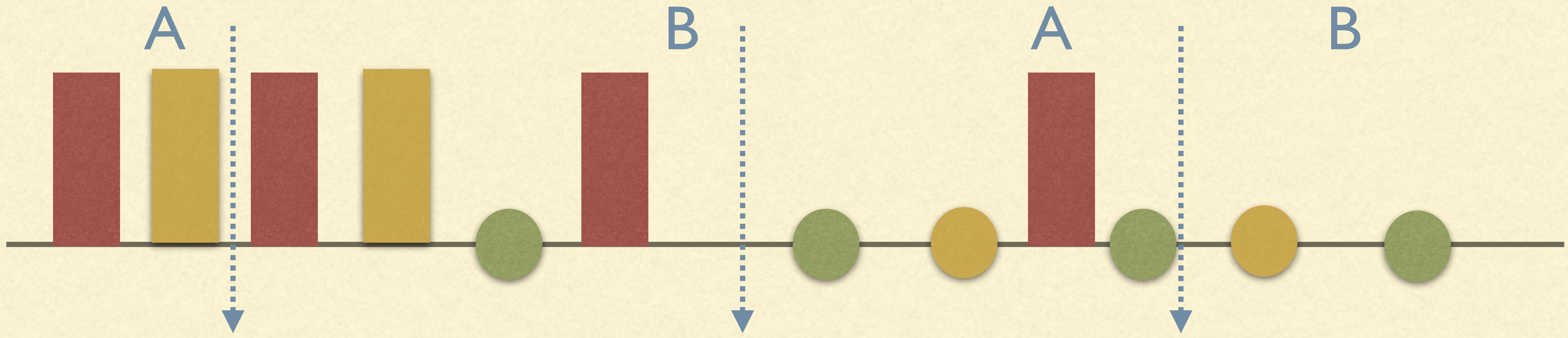
# FROM CONSENSUS-HALVING TO NECKLACE SPLITTING



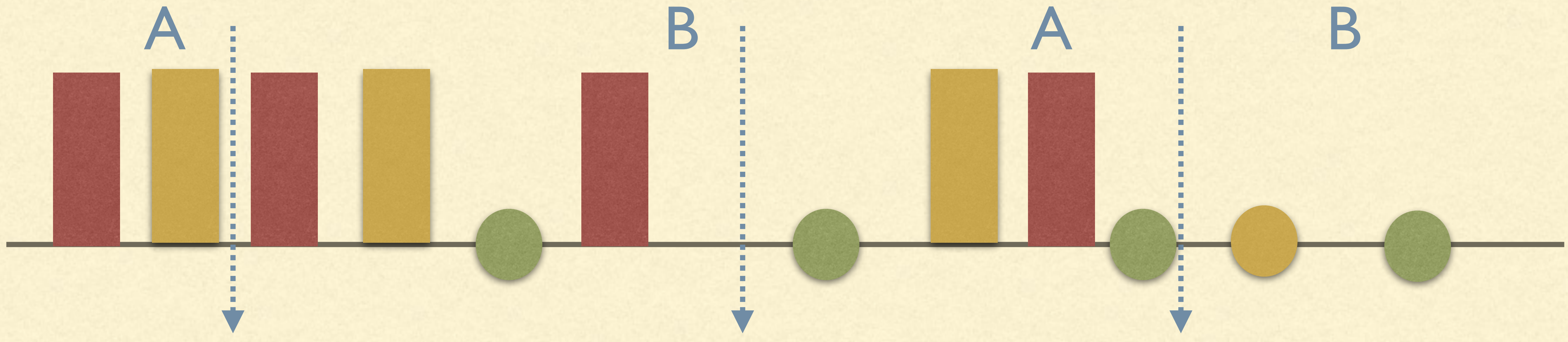
# FROM CONSENSUS-HALVING TO NECKLACE SPLITTING



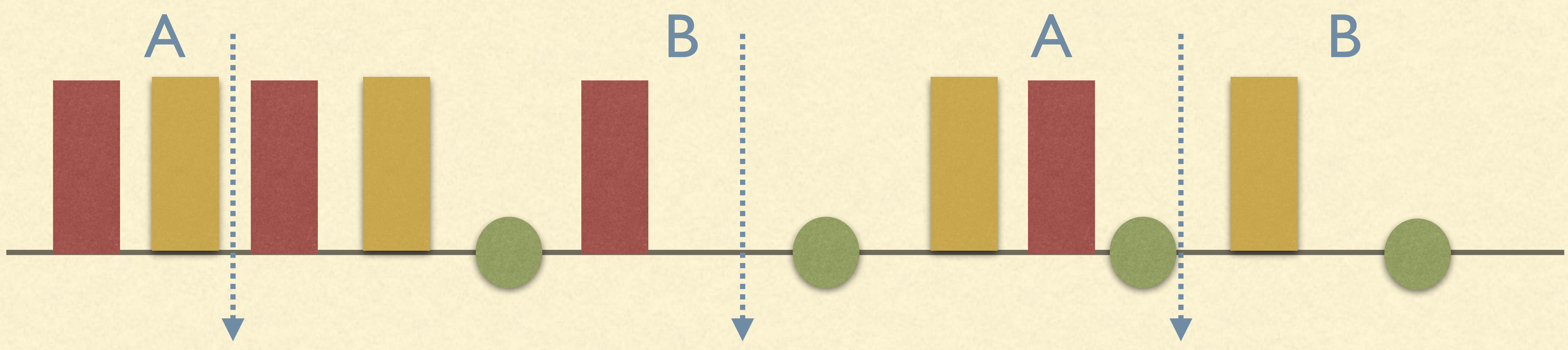
# FROM CONSENSUS-HALVING TO NECKLACE SPLITTING



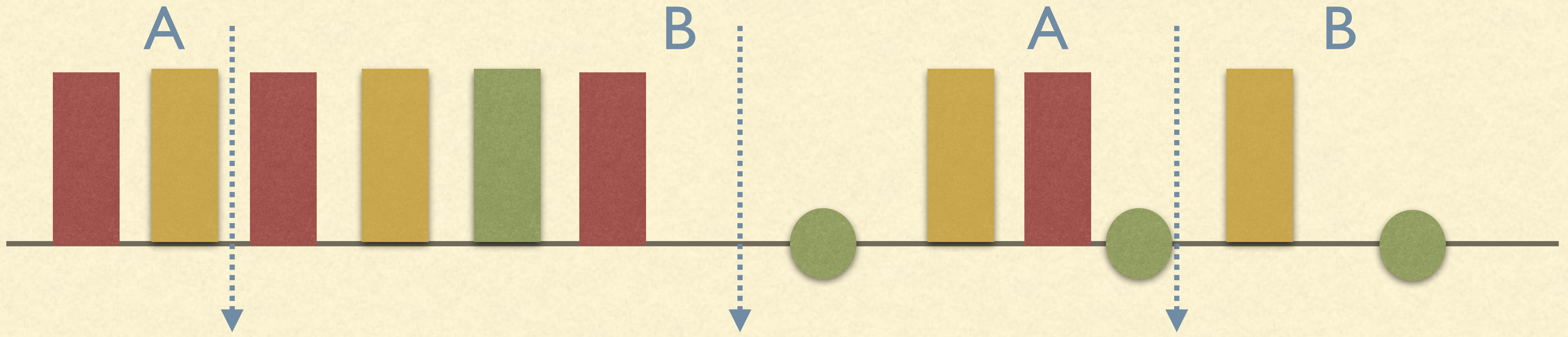
# FROM CONSENSUS-HALVING TO NECKLACE SPLITTING



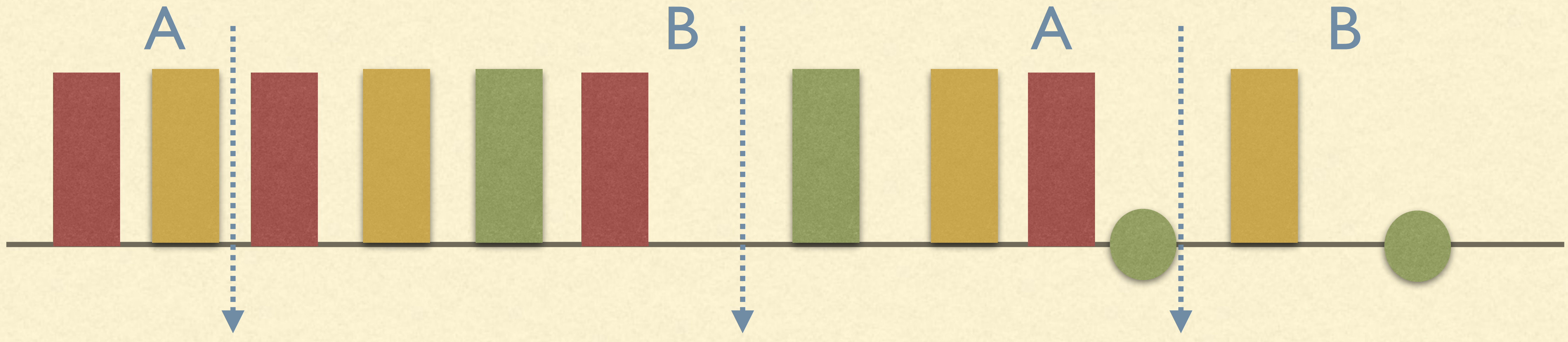
# FROM CONSENSUS-HALVING TO NECKLACE SPLITTING



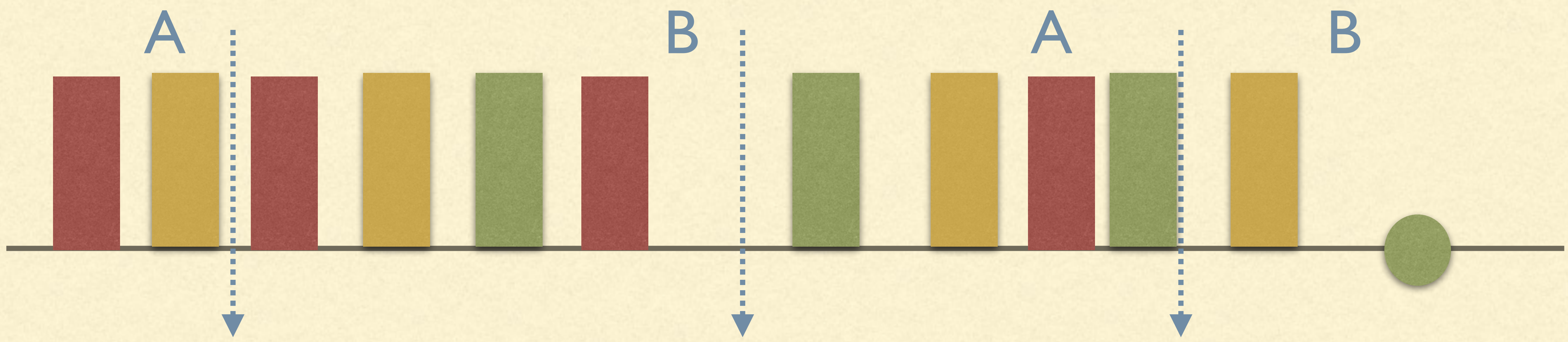
# FROM CONSENSUS-HALVING TO NECKLACE SPLITTING



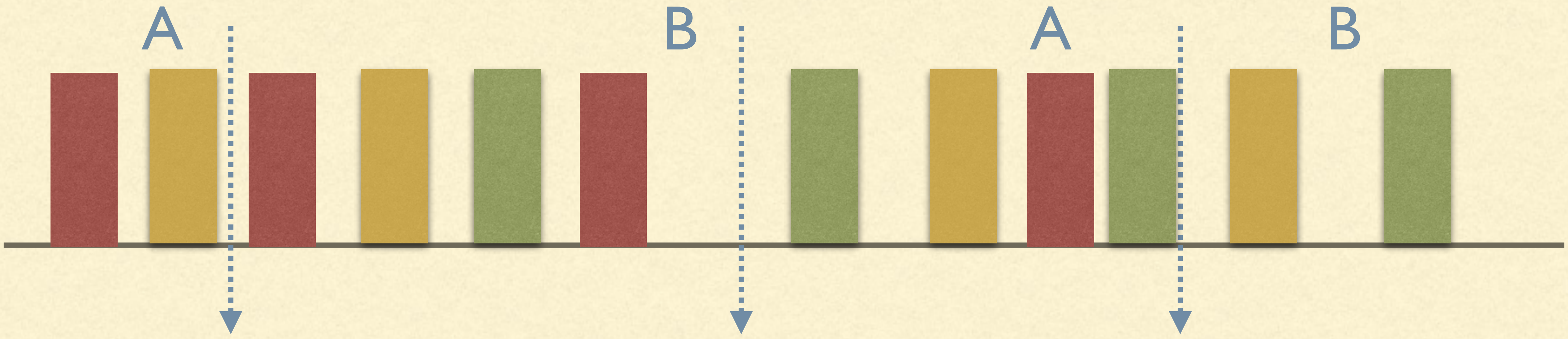
# FROM CONSENSUS-HALVING TO NECKLACE SPLITTING



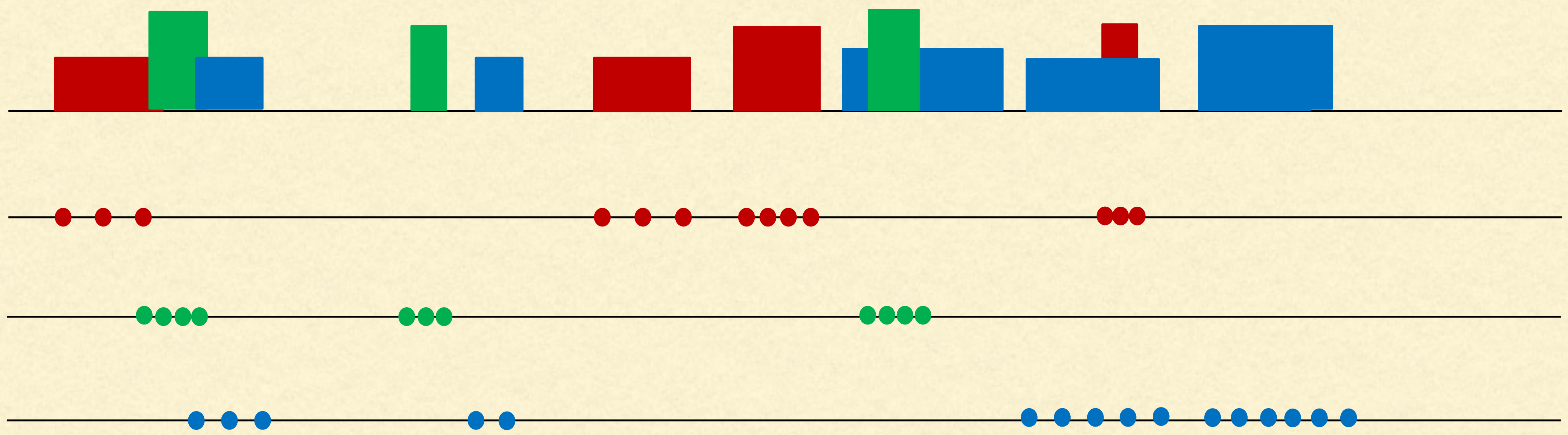
# FROM CONSENSUS-HALVING TO NECKLACE SPLITTING



# FROM CONSENSUS-HALVING TO NECKLACE SPLITTING



# FROM NECKLACE SPLITTING TO CONSENSUS-HALVING

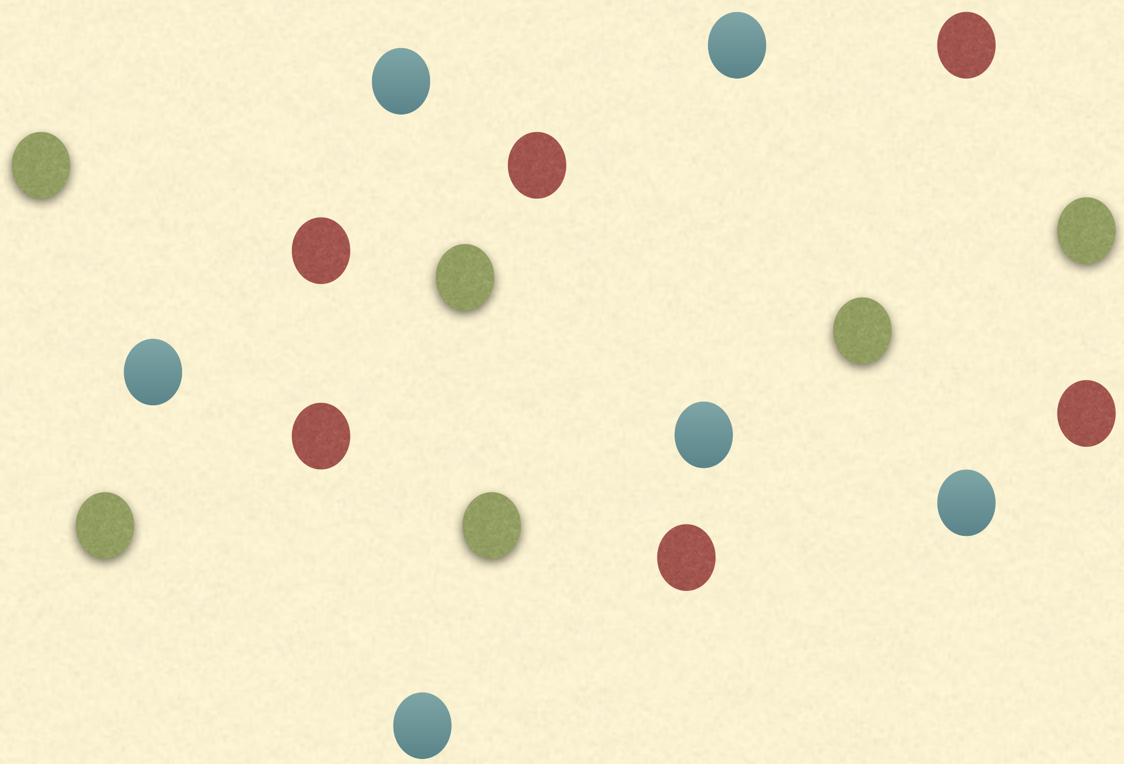


Idea: Simulate value blocks by beads
Denser blocks => more beads.

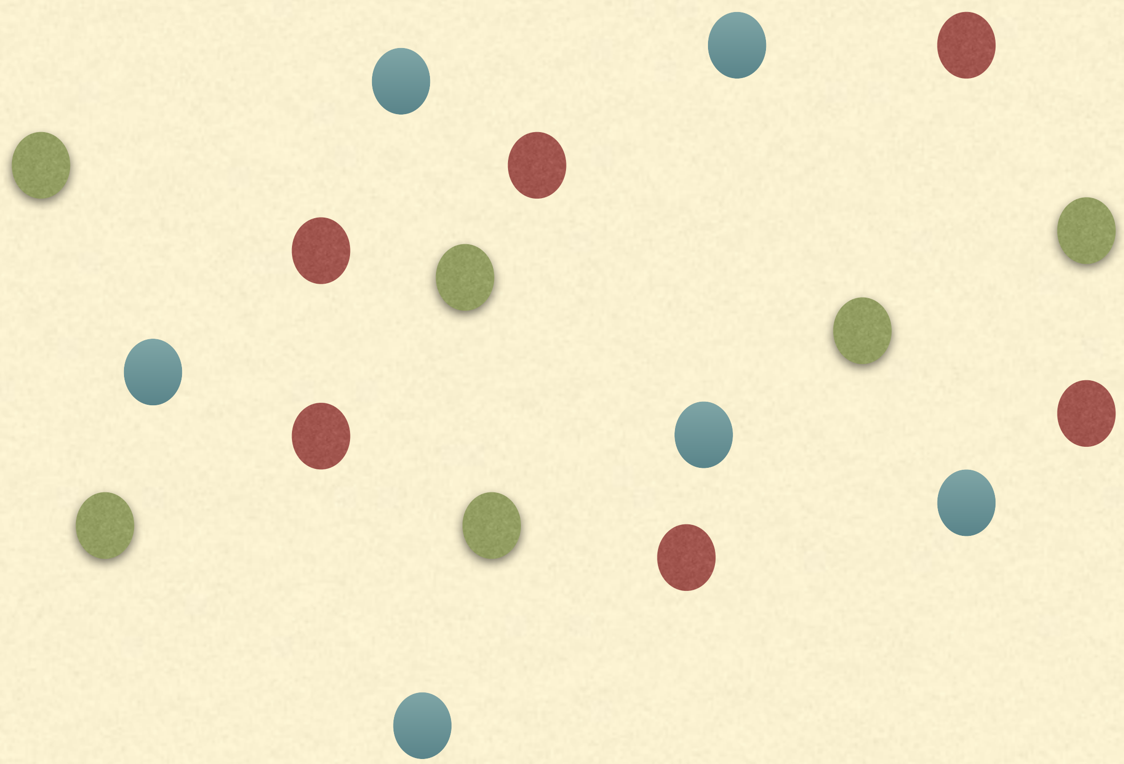
# IN TERMS OF COMPLEXITY…

- To prove computational hardness for NS, it suffices to prove computational hardness for ε-CH.

# DISCRETE HAM SANDWICH

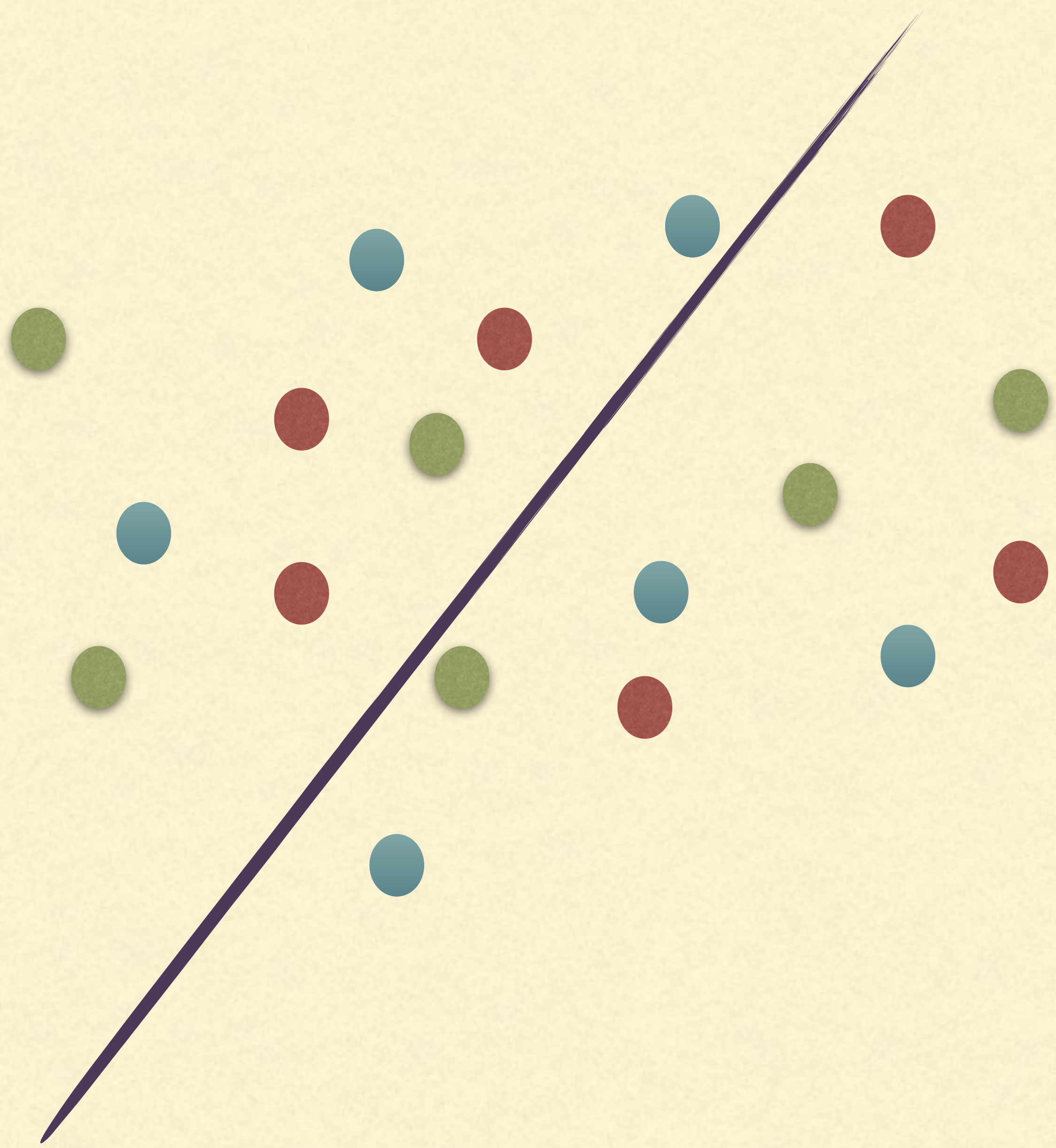


# DISCRETE HAM SANDWICH



- d sets of n points in d-dimensional Euclidean space.
- Find a hyperplane that splits all point sets in half.

# DISCRETE HAM SANDWICH



- d sets of n points in d-dimensional Euclidean space.
- Find a hyperplane that splits all point sets in half.

# HAM SANDWICHES THROUGHOUT THE YEARS

- Steinhaus. A Note on the Ham Sandwich Theorem (Mathesis Polska 1938).
- Stone and Turkey. Generalized "Sandwich" Theorems (Duke Mathematical Journal 1942).
- Edelsbrunner and Waupotitsch. Computing a Ham-Sandwich Cut in Two Dimensions (Symbolic Computation 1986).
- Lo, Matoušek and Steiger. Ham-Sandwich Cuts in R^d (STOC 1992).
- Lo, Matoušek and Steiger. Algorithms for Ham-Sandwich Cuts (Discrete and Computational Geometry 1994).

# FINDING A SOLUTION

- Total problem: A solution always exists.
  - Again, by the Borsuk-Ulam Theorem.

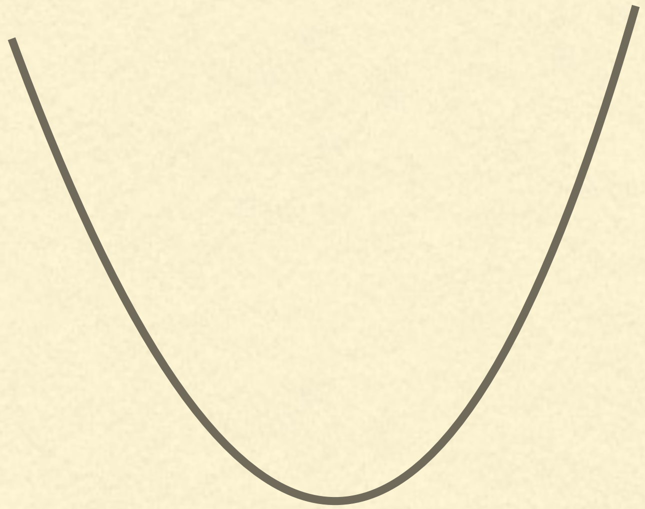
# FROM DISCRETE HAM SANDWICH TO NECKLACE SPLITTING

# FROM DISCRETE HAM SANDWICH TO NECKLACE SPLITTING

Consider the moment curve $(\alpha, \alpha^2, \ldots, \alpha^d)$, for $\alpha \in [0,1]$.

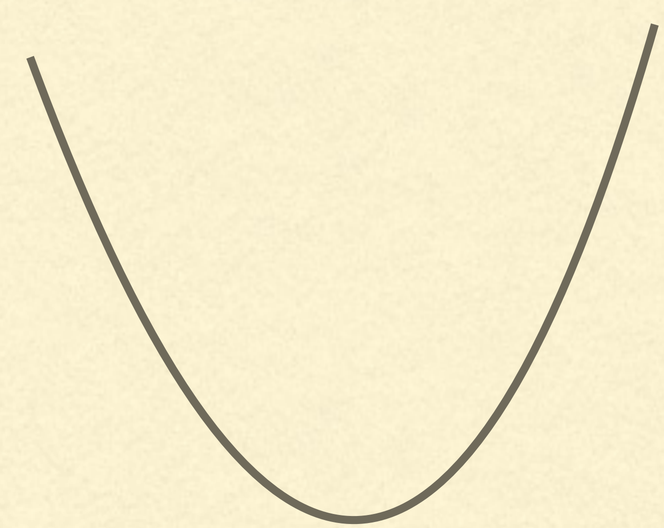
# FROM DISCRETE HAM SANDWICH TO NECKLACE SPLITTING

Consider the moment curve $(\alpha, \alpha^2, \ldots, \alpha^d)$, for $\alpha \in [0,1]$.



# FROM DISCRETE HAM SANDWICH TO NECKLACE SPLITTING

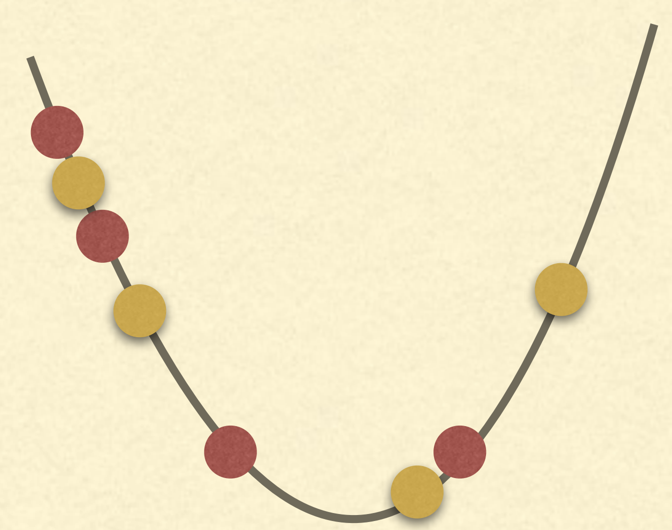
Consider the moment curve $(\alpha, \alpha^2, \ldots, \alpha^d)$, for $\alpha \in [0,1]$.

# FROM DISCRETE HAM SANDWICH TO NECKLACE SPLITTING

Consider the moment curve $(\alpha, \alpha^2, \ldots, \alpha^d)$, for $\alpha \in [0,1]$.

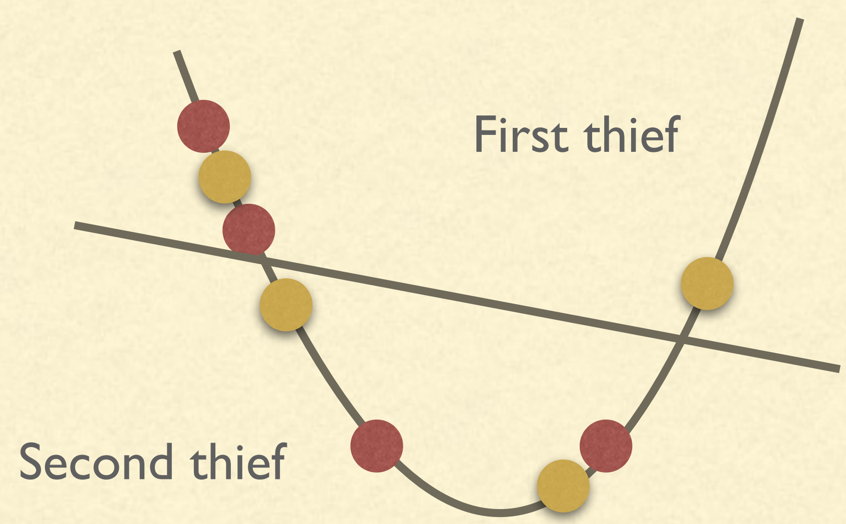

Insert a red point at $(\alpha, \alpha^2, \ldots, \alpha^d)$.



# FROM DISCRETE HAM SANDWICH TO NECKLACE SPLITTING

Consider the moment curve $(\alpha, \alpha^2, \ldots, \alpha^d)$, for $\alpha \in [0,1]$.


Insert a red point at $(\alpha, \alpha^2, \ldots, \alpha^d)$.




The two thieves take alternating pieces.

Any hyperplane intersects the moment curve in at most $d$ points.

# IN TERMS OF COMPLEXITY…

- To prove computational hardness for NS, it suffices to prove computational hardness for ε-CH.

# IN TERMS OF COMPLEXITY…

- To prove computational hardness for NS, it suffices to prove computational hardness for $\varepsilon$-CH.
- To prove computational hardness for DHS, it suffices to prove computational hardness for NS.

# IN TERMS OF COMPLEXITY…

- To prove computational hardness for NS, it suffices to prove computational hardness for ε-CH.
- To prove computational hardness for DHS, it suffices to prove computational hardness for NS.
- It suffices to prove computational hardness for ε-CH.

# THE STATE OF THE WORLD

Necklace Splitting
always exists.

ε-Consensus-Halving
always exists.

Discrete Ham Sandwich
always exists.

# COMPLEXITY CLASSES

# COMPLEXITY CLASSES

TFNP

Meggido and Papadimitriou (Theoretical Computer Science 1991).

“Total” Search Problems, for which a solution is guaranteed to exist and can be verified in polynomial time.

# COMPLEXITY CLASSES

TFNP

Meggido and Papadimitriou (Theoretical Computer Science 1991).

"Total" Search Problems, for which a solution is guaranteed to exist and can be verified in polynomial time.

PPA

Papadimitriou (Journal of Computer and System Sciences, 1994).

Problems reducible to the problem LEAF.

# COMPLEXITY CLASSES

| | |
|---|---|
| TFNP | **Meggido and Papadimitriou (Theoretical Computer Science 1991).**<br>"Total" Search Problems, for which a solution is guaranteed to exist and can be verified in polynomial time. |
| PPA | **Papadimitriou (Journal of Computer and System Sciences, 1994).**<br>Problems reducible to the problem **LEAF.** |
| PPAD | **Papadimitriou (Journal of Computer and System Sciences, 1994).**<br>Problems reducible to the problem **END-OF-LINE.** |

# COMPLEXITY CLASSES

TFNP

↑

PPA

↑

PPAD

**Meggido and Papadimitriou (Theoretical Computer Science 1991).**

"Total" Search Problems, for which a solution is guaranteed to exist and can be verified in polynomial time.

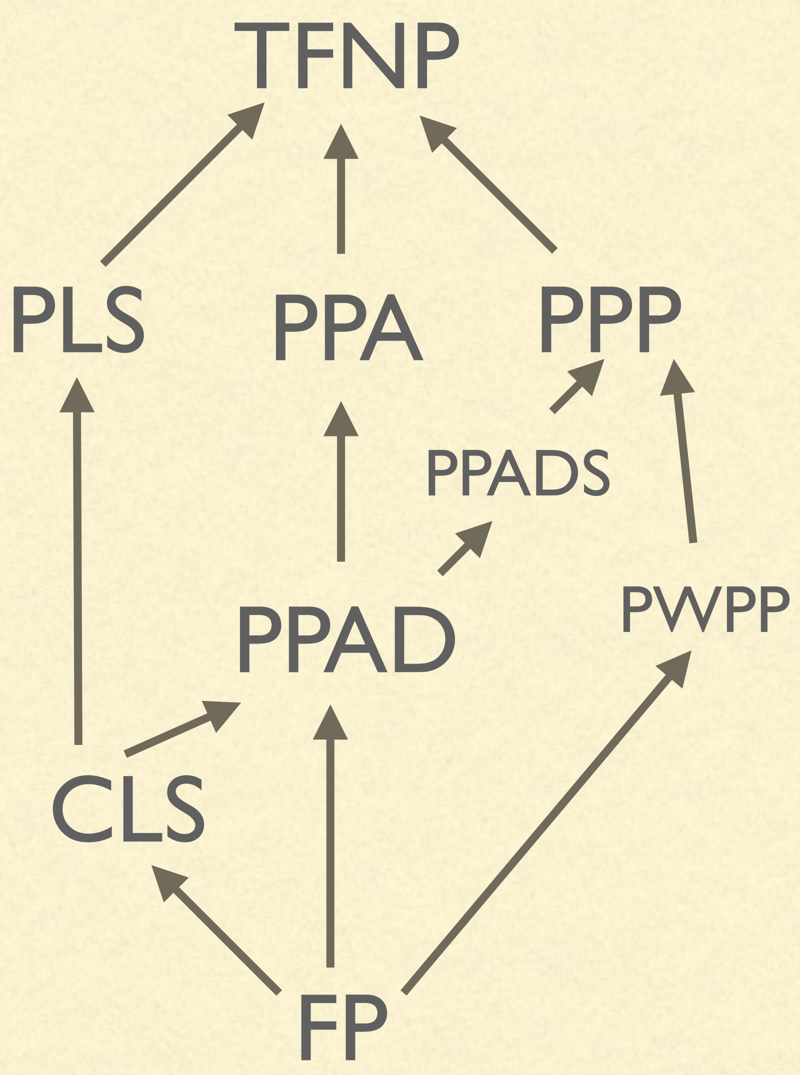
**Papadimitriou (Journal of Computer and System Sciences, 1994).**

Problems reducible to the problem **LEAF.**

**Papadimitriou (Journal of Computer and System Sciences, 1994).**

Problems reducible to the problem **END-OF-LINE.**

# COMPLEXITY CLASSES




**Meggido and Papadimitriou (Theoretical Computer Science 1991).**

"Total" Search Problems, for which a solution is guaranteed to exist and can be verified in polynomial time.

**Papadimitriou (Journal of Computer and System Sciences, 1994).**

Problems reducible to the problem **LEAF.**

**Papadimitriou (Journal of Computer and System Sciences, 1994).**

Problems reducible to the problem **END-OF-LINE.**

# SUCCESS OF PPAD

## How Constantinos Daskalakis solved Nash's riddle

August 2, 2018, 1:21 pm

Nevanlinna prize-winner Constantinos Daskalakis explained how he treated the problem of Nash equilibrium computing after 60 years.
The Greek mathematician confirmed that the so-called Nash equilibrium, a classic case of game theory, has the same complexity as NP-complete, that is, it is insoluble by any other algorithm. So, the behavior of competitors in a time compatible situation cannot always be predicted.

Daskalakis, Goldberg and Papadimitriou.
The Complexity of Computing a Nash equilibrium.
(SIAM Journal of Computing, 2009).

Chen, Deng and Tang
Settling the Complexity of Computing 2-Player Nash Equilibria.
(Journal of the ACM, 2009).

- 2011 SIAM Outstanding Paper Prize
- 2008 Kalai Prize
- 2008 ACM Doctoral Dissertation Award

# PPAD

- END-OF-LINE:
  - Input: A (exponentially large, with $2^n$ vertices, implicitly given) directed graph, where each vertex has in-degree and out-degree at most 1 and a vertex with in-degree 0.
  - Output: A vertex with in-degree or out-degree 0.

END-OF-LINE

**Input:** Two boolean circuits $S$ (for successor) and $P$ (for predecessor) with $n$ inputs and $n$ outputs such that $P(0^n) = 0^n \neq S(0^n)$.

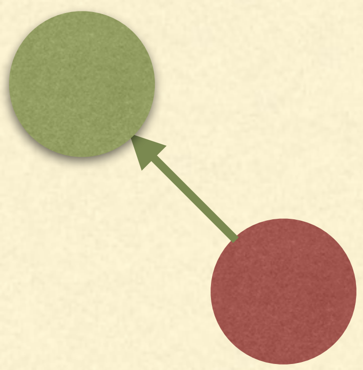
**Output:** A vertex $x$ such that $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0^n$.

# END-OF-LINE

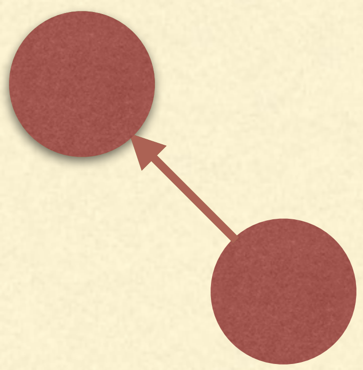
# END-OF-LINE

# END-OF-LINE

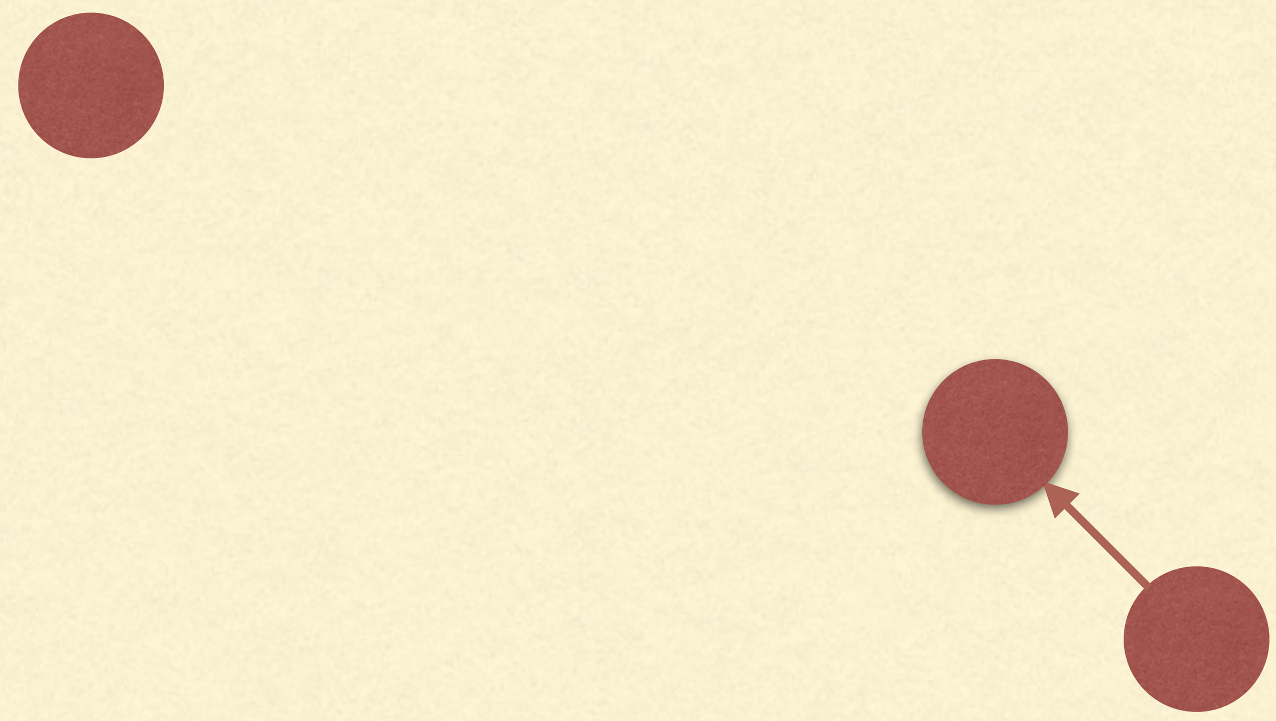
# END-OF-LINE



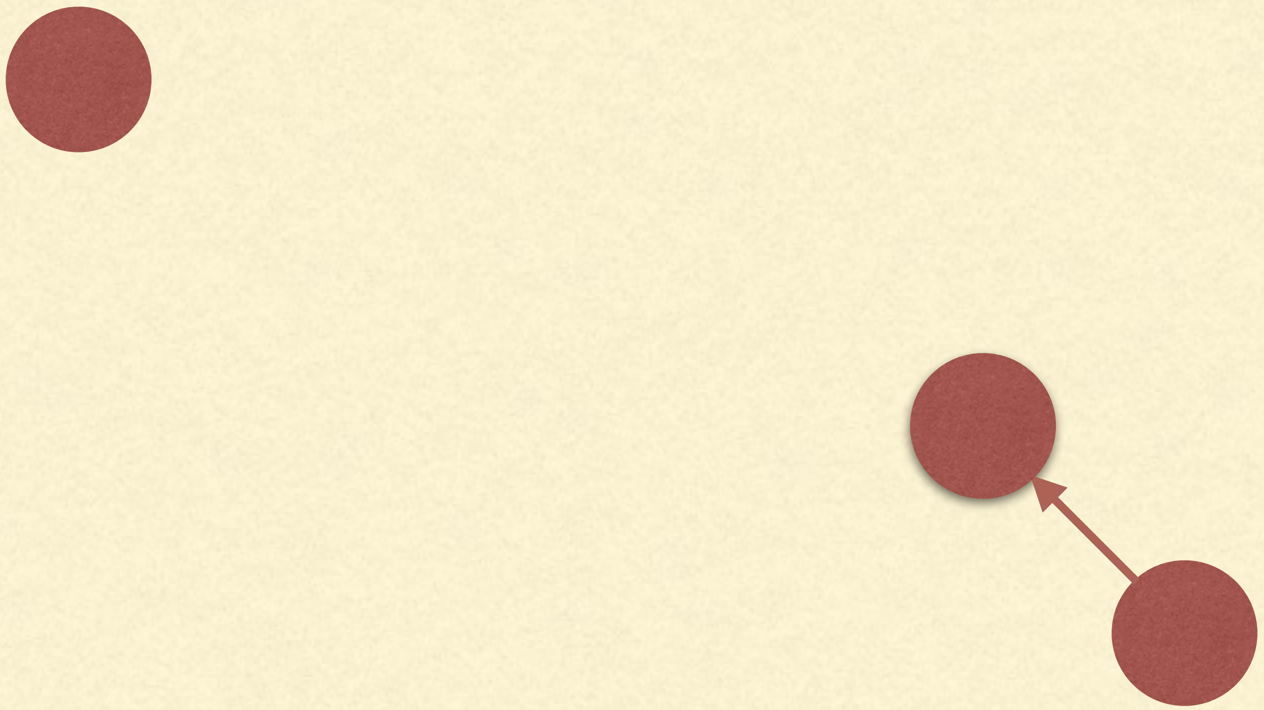
# END-OF-LINE



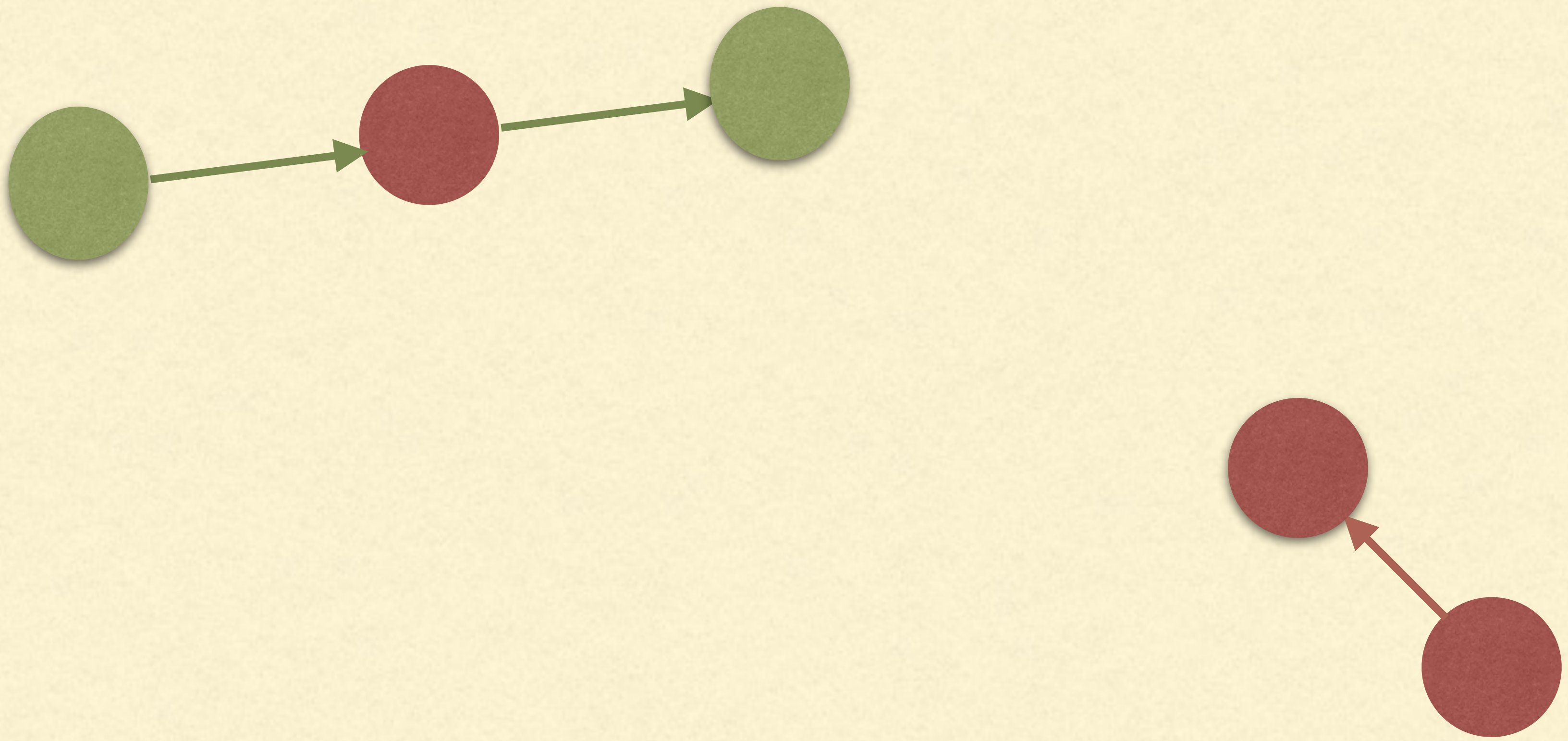
# END-OF-LINE



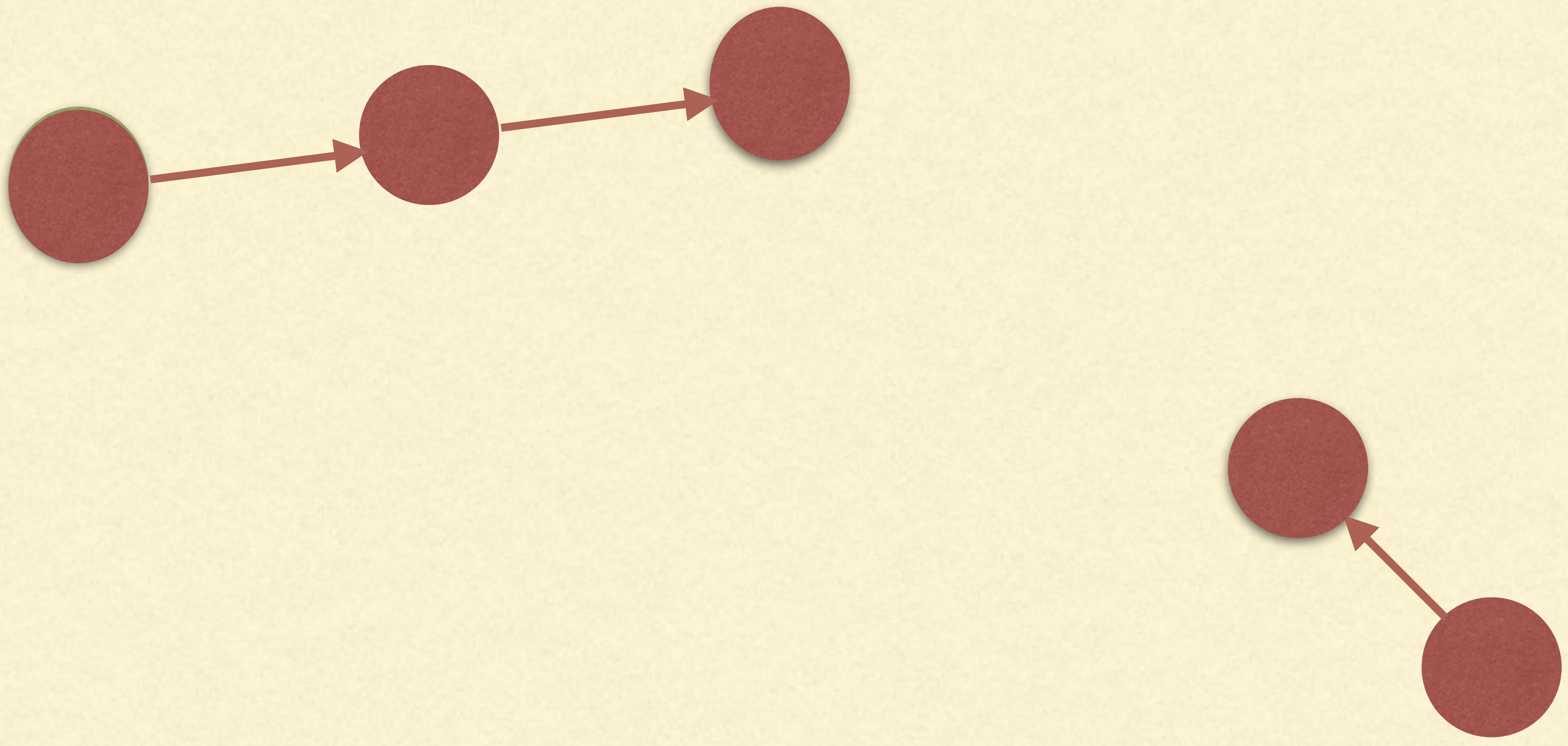
# END-OF-LINE



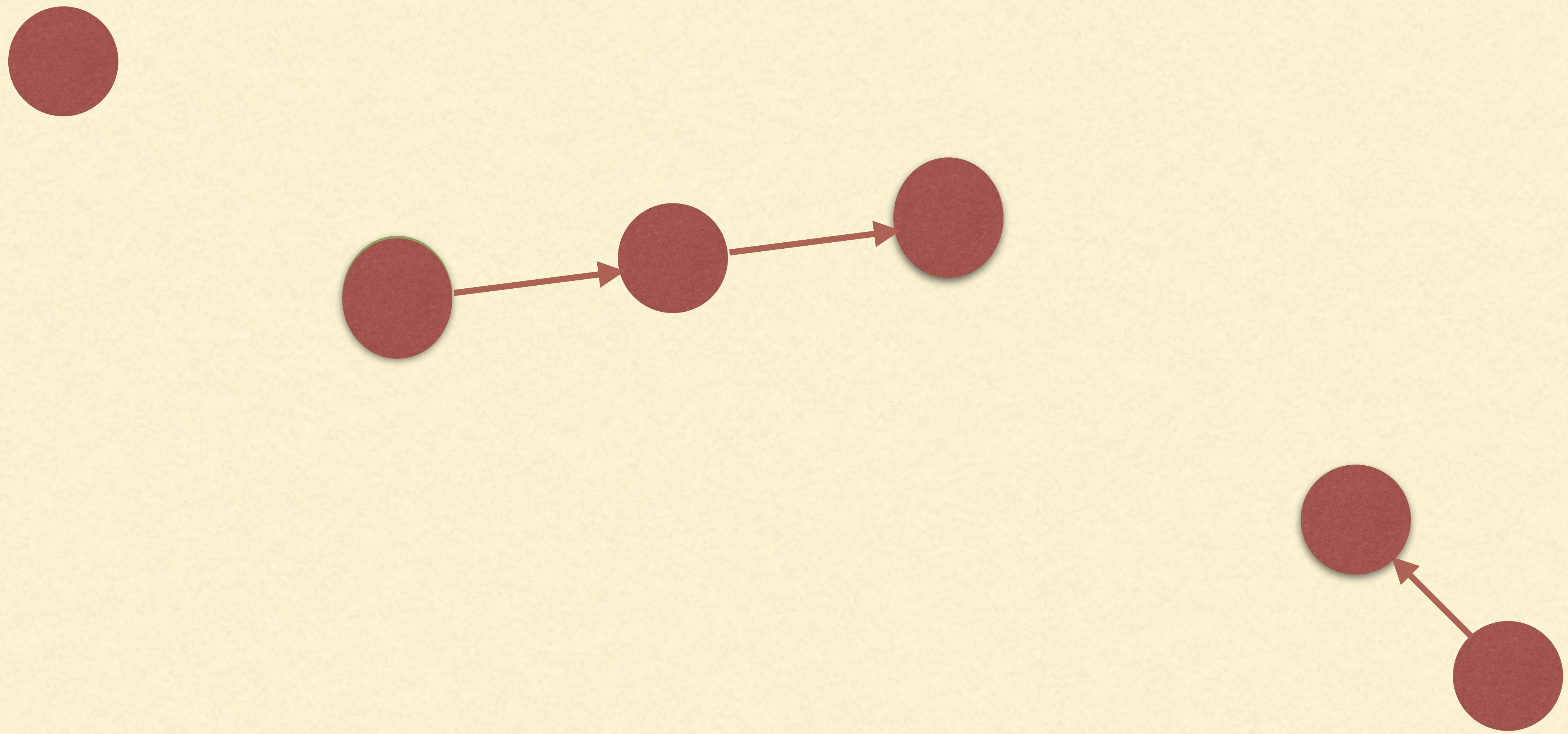
# END-OF-LINE



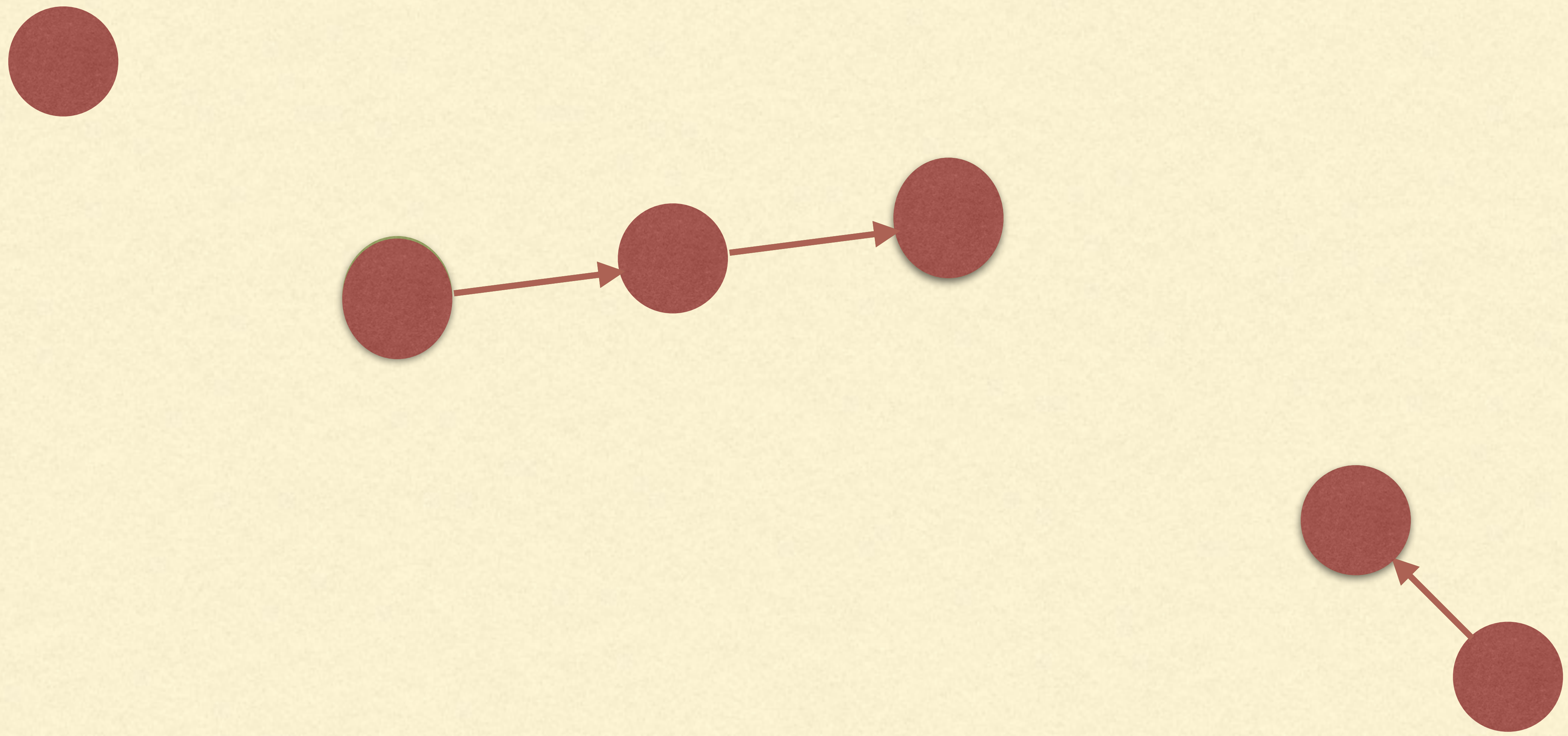
# END-OF-LINE



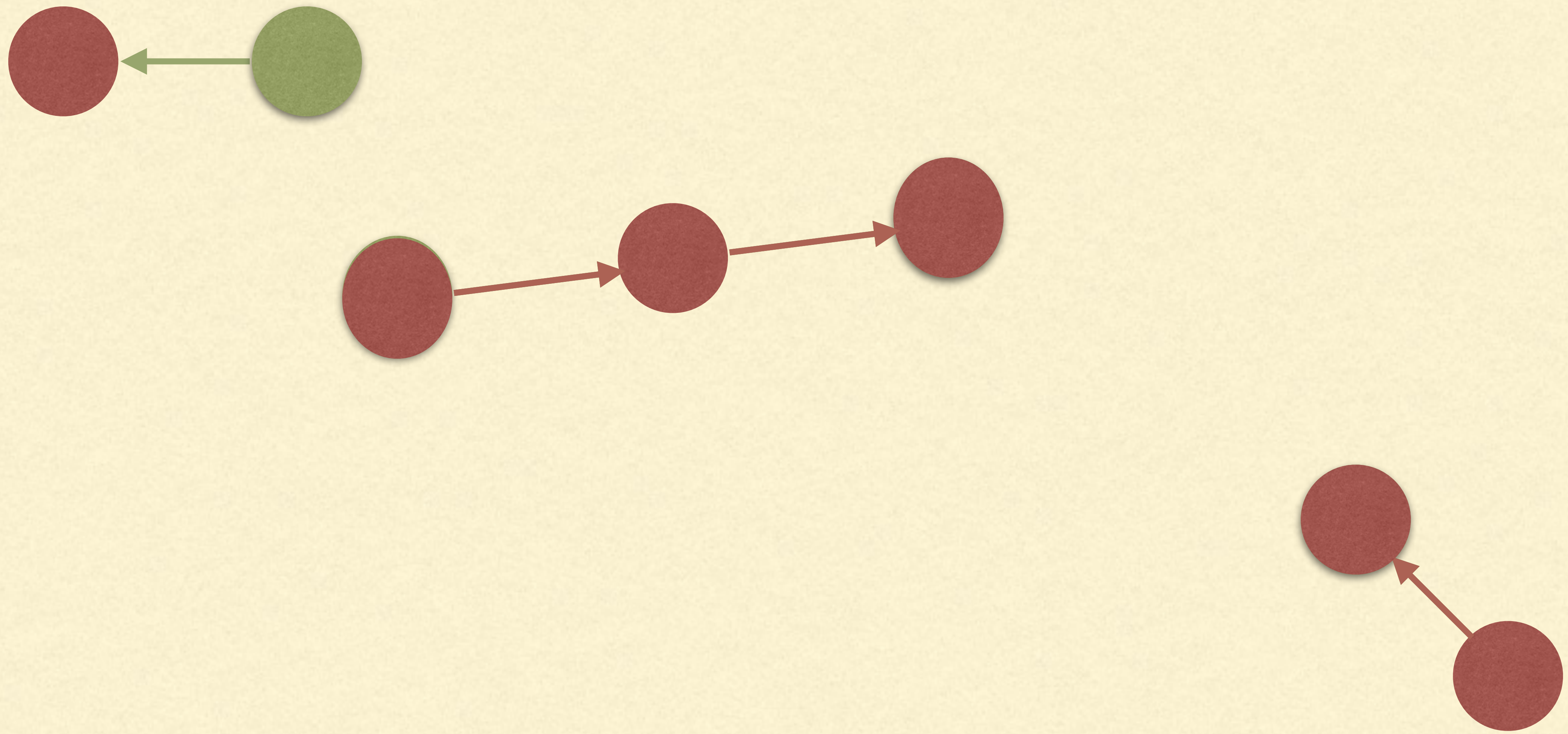
# END-OF-LINE



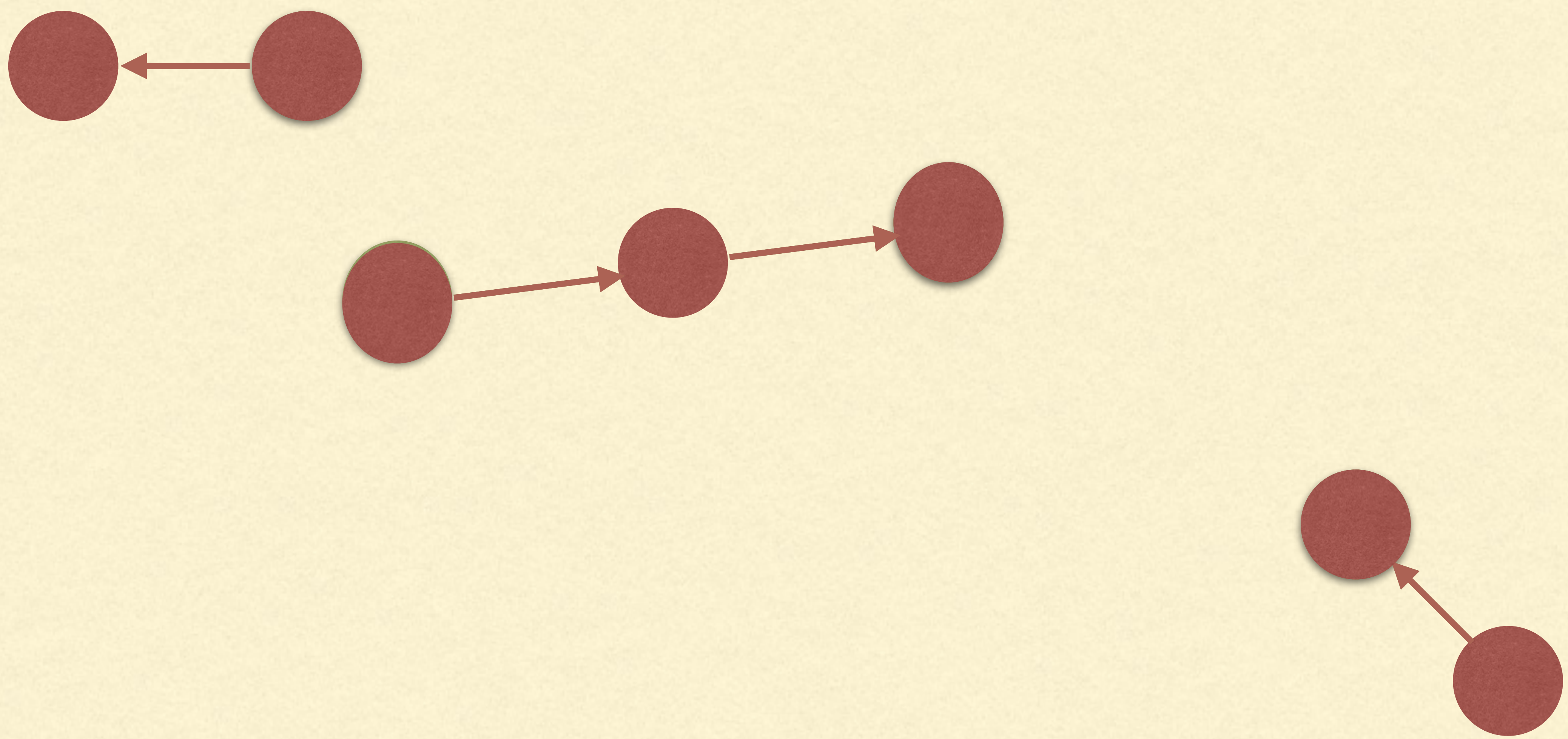
# END-OF-LINE



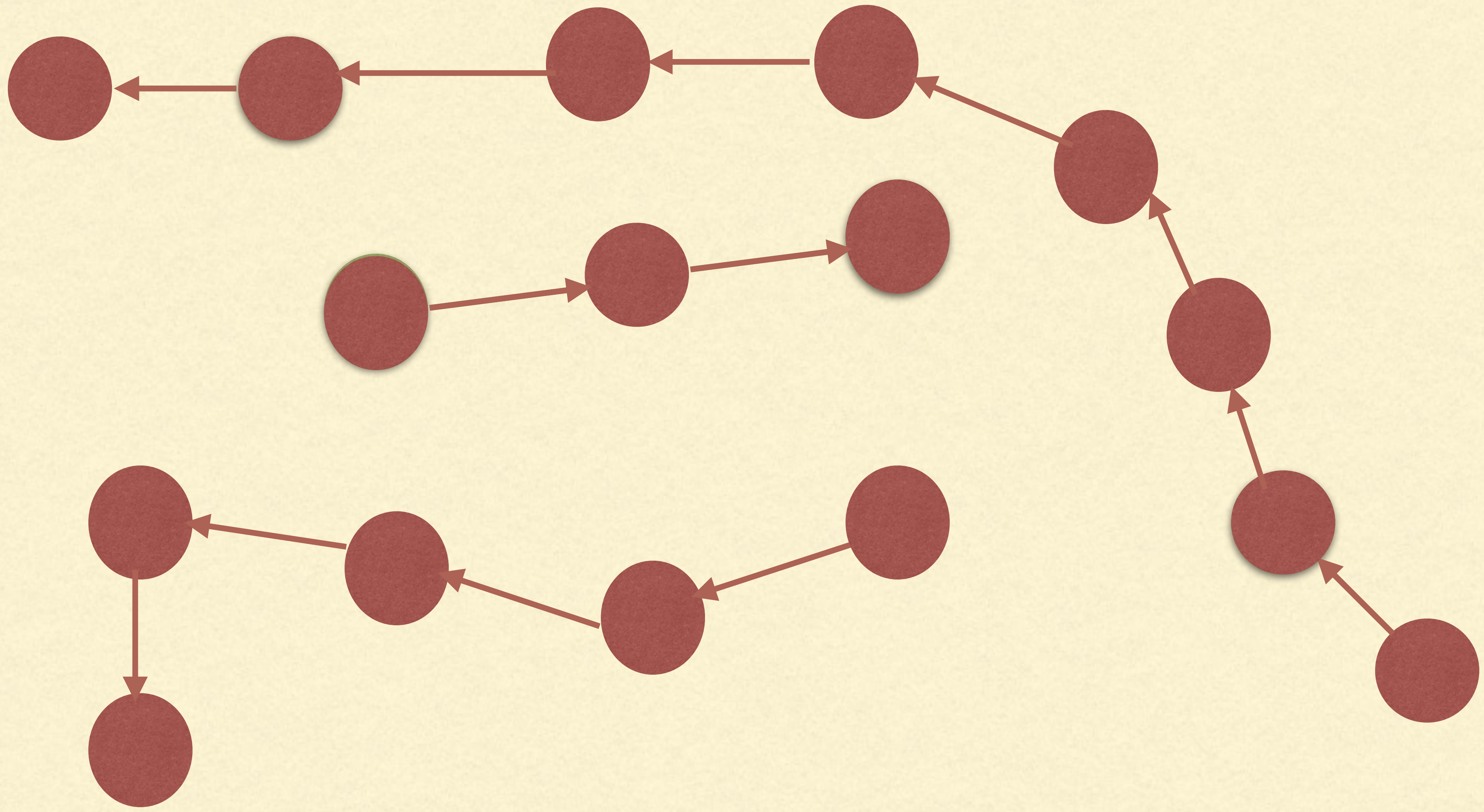
# END-OF-LINE



# END-OF-LINE



# END-OF-LINE



# PPA

- LEAF:
  - Input: An undirected (exponentially large, implicitly given) *undirected* graph where each vertex has degree at most 2 and a vertex of degree 1.
  - Output: Another vertex of degree 1.

LEAF

**Input:** A boolean circuit C with $n$ inputs and at most $2n$ outputs, outputting the set $\mathcal{N}(y)$ of (at most two) neighbours of a vertex $y$, such that $|\mathcal{N}(0^n)| = 1$.

**Output:** A vertex $x$ such that $x \neq 0^n$ and $|\mathcal{N}(x)| = 1$.

# PPAD AND PPA

# PPAD AND PPA

- PPAD

# PPAD AND PPA

- PPAD
- Stands for "Polynomial Parity Argument on a Directed graph".

# PPAD AND PPA

- PPAD
- Stands for "Polynomial Parity Argument on a Directed graph".
  - A problem is in PPAD if it is polynomial-time reducible to END-OF-LINE.

# PPAD AND PPA

- PPAD
- Stands for "Polynomial Parity Argument on a Directed graph".
    - A problem is in PPAD if it is polynomial-time reducible to END-OF-LINE.
    - A problem is PPAD-hard if END-OF-LINE is polynomial-time reducible to it.

# PPAD AND PPA

- PPAD
- Stands for "Polynomial Parity Argument on a Directed graph".
  - A problem is in PPAD if it is polynomial-time reducible to END-OF-LINE.
  - A problem is PPAD-hard if END-OF-LINE is polynomial-time reducible to it.
- PPA

# PPAD AND PPA

- PPAD
- Stands for "Polynomial Parity Argument on a Directed graph".
    - A problem is in PPAD if it is polynomial-time reducible to END-OF-LINE.
    - A problem is PPAD-hard if END-OF-LINE is polynomial-time reducible to it.
- PPA
- Stands for "Polynomial Parity Argument".

# PPAD AND PPA

- PPAD
- Stands for “Polynomial Parity Argument on a Directed graph”.
  - A problem is in PPAD if it is polynomial-time reducible to END-OF-LINE.
  - A problem is PPAD-hard if END-OF-LINE is polynomial-time reducible to it.
- PPA
- Stands for “Polynomial Parity Argument”.
  - Containment and hardness defined with respect to polynomial-time reductions to/from LEAF.

# THE COMPLEXITY OF THE THREE PROBLEMS.

# THE COMPLEXITY OF THE THREE PROBLEMS.

- They are all in PPA.
  [Papadimitriou 1994, F.R., Frederiksen, Goldberg and Zhang 2019, F.R. and Goldberg 2019].

# THE COMPLEXITY OF THE THREE PROBLEMS.

- They are all in PPA.
  [Papadimitriou 1994, F.R., Frederiksen, Goldberg and Zhang 2019, F.R. and Goldberg 2019].
  - Simmons and Su's proof already almost an "in PPA" result.

# THE COMPLEXITY OF THE THREE PROBLEMS.

- They are all in PPA.
  [Papadimitriou 1994, F.R., Frederiksen, Goldberg and Zhang 2019, F.R. and Goldberg 2019].
  - Simmons and Su's proof already almost an "in PPA" result.
- What about hardness?

# THE STATE OF THE WORLD

Necklace Splitting
always exists.

ε-Consensus-Halving
always exists.

Discrete Ham Sandwich
always exists.

# THE STATE OF THE WORLD

Necklace Splitting always exists. in PPA

ε-Consensus-Halving always exists. in PPA

Discrete Ham Sandwich always exists. in PPA

# A DEEPER LOOK INTO PPA-COMPLETE PROBLEMS

Let's see what we have to reduce from!

# COMPLETE PROBLEMS FOR PPA AND PPAD

# COMPLETE PROBLEMS FOR PPA AND PPAD

- SPERNER, BROUWER, KAKUTANI Papadimitriou (1994).
- NASH Daskalakis, Goldberg and Papadimitiou (2005, 2009), Chen and Deng (2007, 2009).
- EXCHANGE ECONOMY Papadimitriou (1994), Chen, Paparas and Yiannakakis (2013).
- ENVY-FREE CAKE CUTTING Deng, Qi and Saberi (2009, 2012).
- Many more …

# COMPLETE PROBLEMS FOR PPA AND PPAD

- SPERNER, BROUWER, KAKUTANI Papadimitriou (1994).
- NASH Daskalakis, Goldberg and Papadimitiou (2005, 2009), Chen and Deng (2007, 2009).
- EXCHANGE ECONOMY Papadimitriou (1994), Chen, Paparas and Yiannakakis (2013).
- ENVY-FREE CAKE CUTTING Deng, Qi and Saberi (2009, 2012).
- Many more …

- SPERNER for non-orientable spaces Grigni (2001), Friedl, Ivanyos, Santha and Verhoeven (2006).
- 2D-TUCKER, BORSUK-ULAM Aisenberg, Bonet and Buss (2020).
- OCTAHEDRAL TUCKER Deng, Feng and Kulkarni (2017).
- TUCKER, SPERNER on Möbius band and Klein bottle. Deng, Feng, Liu and Qi (2015).
- Not many more …

# COMPLETE PROBLEMS FOR PPA AND PPAD

**Consider a triangulated simplex and a polynomial-time machine (or a circuit) that assigns labels to the vertices of the triangulation…**

- SPERNER, BROUWER, KAKUTANI Papadimitriou (1994).
- NASH Daskalakis, Goldberg and Papadimitiou (2005, 2009), Chen and Deng (2007, 2009).
- EXCHANGE ECONOMY Papadimitriou (1994), Chen, Paparas and Yiannakakis (2013).
- ENVY-FREE CAKE CUTTING Deng, Qi and Saberi (2009, 2012).
- Many more …

- SPERNER for non-orientable spaces Grigni (2001), Friedl, Ivanyos, Santha and Verhoeven (2006).
- 2D-TUCKER, BORSUK-ULAM Aisenberg, Bonet and Buss (2020).
- OCTAHEDRAL TUCKER Deng, Feng and Kulkarni (2017).
- TUCKER, SPERNER on Möbius band and Klein bottle. Deng, Feng, Liu and Qi (2015).
- Not many more …

# COMPLETE PROBLEMS FOR PPA AND PPAD

- SPERNER, BROUWER, KAKUTANI Papadimitriou (1994).
- NASH Daskalakis, Goldberg and Papadimitiou (2005, 2009), Chen and Deng (2007, 2009).
- EXCHANGE ECONOMY Papadimitriou (1994), Chen, Paparas and Yiannakakis (2013).
- ENVY-FREE CAKE CUTTING Deng, Qi and Saberi (2009, 2012).
- Many more …

- SPERNER for non-orientable spaces Grigni (2001), Friedl, Ivanyos, Santha and Verhoeven (2006).
- 2D-TUCKER, BORSUK-ULAM Aisenberg, Bonet and Buss (2020).
- OCTAHEDRAL TUCKER Deng, Feng and Kulkarni (2017).
- TUCKER, SPERNER on Möbius band and Klein bottle. Deng, Feng, Liu and Qi (2015).
- Not many more …

# COMPLETE PROBLEMS FOR PPA AND PPAD

- SPERNER, BROUWER, KAKUTANI Papadimitriou (1994).
- NASH Daskalakis, Goldberg and Papadimitiou (2005, 2009), Chen and Deng (2007, 2009).
- EXCHANGE ECONOMY Papadimitriou (1994), Chen, Paparas and Yiannakakis (2013).
- ENVY-FREE CAKE CUTTING Deng, Qi and Saberi (2009, 2012).
- Many more …

- SPERNER for non-orientable spaces Grigni (2001), Friedl, Ivanyos, Santha and Verhoeven (2006).

Consider a triangulated hypergrid and **a polynomial-time machine (or a circuit)** that assigns labels to the vertices of the triangulation…

- 2D-TUCKER, BORSUK-ULAM Aisenberg, Bonet and Buss (2020).
- OCTAHEDRAL TUCKER Deng, Feng and Kulkarni (2017).
- TUCKER, SPERNER on Möbius band and Klein bottle. Deng, Feng, Liu and Qi (2015).
- Not many more …

# COMPLETE PROBLEMS FOR PPA AND PPAD

- SPERNER, BROUWER, KAKUTANI Papadimitriou (1994).
- NASH Daskalakis, Goldberg and Papadimitiou (2005, 2009), Chen and Deng (2007, 2009).
- EXCHANGE ECONOMY Papadimitriou (1994), Chen, Paparas and Yiannakakis (2013).
- ENVY-FREE CAKE CUTTING Deng, Qi and Saberi (2009, 2012).
- Many more …

- SPERNER for non-orientable spaces Grigni (2001), Friedl, Ivanyos, Santha and Verhoeven (2006).
- 2D-TUCKER, BORSUK-ULAM Aisenberg, Bonet and Buss (2020).
- OCTAHEDRAL TUCKER Deng, Feng and Kulkarni (2017).
- TUCKER, SPERNER on Möbius band and Klein bottle. Deng, Feng, Liu and Qi (2015).
- Not many more …

# “NATURAL” PPA-COMPLETE PROBLEMS?

# "NATURAL" PPA-COMPLETE PROBLEMS?

**Papadimitriou (1994)**

(4) Can we show more problems complete for PPA and PPAD—especially ones that do not have a Turing machine embedded in the input? For example, is SMITH PPA-complete? We strongly suspect that it is. Also, is the "equal sums problem" complete for PPP?

# "NATURAL" PPA-COMPLETE PROBLEMS?

**Papadimitriou (1994)**

(4) Can we show more problems complete for PPA and PPAD—especially ones that do not have a Turing machine embedded in the input? For example, is SMITH PPA-complete? We strongly suspect that it is. Also, is the "equal sums problem" complete for PPP?

**Grigni (2001)**

We would still like a natural complete problem for either PPA or PPAD that does not have an explicit Turing machine in the input. The most promising candidate still seems to be SMITH: given a Hamiltonian cycle in a graph with all nodes of odd degree, find another.

# “NATURAL” PPA-COMPLETE PROBLEMS?

**Papadimitriou (1994)**

(4) Can we show more problems complete for PPA and PPAD—especially ones that do not have a Turing machine embedded in the input? For example, is SMITH PPA-complete? We strongly suspect that it is. Also, is the “equal sums problem” complete for PPP?

**Grigni (2001)**

We would still like a natural complete problem for either PPA or PPAD that does not have an explicit Turing machine in the input. The most promising candidate still seems to be SMITH: given a Hamiltonian cycle in a graph with all nodes of odd degree, find another.

**Aisenberg, Bonet and Buss (2020)**

As already mentioned, it is open whether problems such as integer factoring, or Smith’s theorem on cubic graphs give PPA-complete TFNP search problems. Papadimitriou [19] and Grigni [14] mention the Smith problem as a candidate for a PPA-complete problem that does not have a Turing machine explicitly encoded in its input.

# NATURAL PROBLEMS

- Problems that do not have a circuit explicit in their definition. (Papadimitriou (1994), Grigni (2001), Aisenberg, Bonet and Buss (2020)).
- Problems that were identified independently from the work on TFNP. (Goldberg (2019), Algorithms UK).

# THE STATE OF THE WORLD

Necklace Splitting always exists. — in PPA

ε-Consensus-Halving always exists. — in PPA

Discrete Ham Sandwich always exists. — in PPA

# THE STATE OF THE WORLD

Necklace Splitting always exists. — in PPA

ε-Consensus-Halving always exists. — in PPA

Discrete Ham Sandwich always exists. — in PPA

PPA: a lonely class
No natural complete problems

# NATURAL PROBLEMS

- Problems that do not have a circuit explicit in their definition. (Papadimitriou (1994), Grigni (2001), Aisenberg, Bonet and Buss (2020)).
- Problems that were identified independently from the work on TFNP. (Goldberg (2019), Algorithms UK).

# NATURAL PROBLEMS

- Problems that do not have a circuit explicit in their definition. (Papadimitriou (1994), Grigni (2001), Aisenberg, Bonet and Buss (2020)).
- Problems that were identified independently from the work on TFNP. (Goldberg (2019), Algorithms UK).

Necklace Splitting and Consensus-Halving are natural!
Two birds with one stone?

# A NATURAL PPA-COMPLETE PROBLEM

# A NATURAL PPA-COMPLETE PROBLEM

- F.R. and Goldberg. Consensus-Halving is PPA-complete. (STOC 2018).

# A NATURAL PPA-COMPLETE PROBLEM

- F.R. and Goldberg. Consensus-Halving is PPA-complete. (STOC 2018).
  - When ε is inversely exponential.

| $(n, k, \boxed{\epsilon})$-CHalving |
| --- |
| **Input:** The value density functions $v_i : O \to R_+, i = 1, \cdots, n$, for $n$ agents. |
| **Output:** A partition $(O_+, O_-)$ with $k$ cuts such that $\lvert u_i(O_+) - u_i(O_-)\rvert \leq \epsilon$ for all agents $i \in N$. |

inversely exponential — inversely polynomial — constant

problem becomes easier

hardness becomes more difficult to prove

# THE STATE OF THE WORLD

Necklace Splitting always exists. — in PPA

ε-Consensus-Halving always exists. — in PPA

Discrete Ham Sandwich always exists. — in PPA

PPA: a lonely class
No natural complete problems

# THE STATE OF THE WORLD

Necklace Splitting always exists. — in PPA

ε-Consensus-Halving always exists. — in PPA

Discrete Ham Sandwich always exists. — in PPA

CH is PPA-complete

[F.R. and Goldberg, 2018]

PPA: a lonely class
No natural complete problems

# THE STATE OF THE WORLD

CH is PPA-complete

[F.R. and Goldberg, 2018]

Necklace Splitting always exists. — in PPA

ε-Consensus-Halving always exists. — in PPA

PPA: a lonely class
No natural complete problems

Discrete Ham Sandwich always exists. — in PPA

PPA now has a natural complete problem.

# NECKLACES AND SANDWICHES

- F.R. and Goldberg. The Complexity of Splitting Necklaces and Bisecting Ham Sandwiches (STOC 2019).

# NECKLACES AND SANDWICHES

- F.R. and Goldberg. The Complexity of Splitting Necklaces and Bisecting Ham Sandwiches (STOC 2019).
    - We prove that ε-CONSENSUS HALVING is PPA-Complete for inverse-polynomial ε.

# NECKLACES AND SANDWICHES

- F.R. and Goldberg. The Complexity of Splitting Necklaces and Bisecting Ham Sandwiches (STOC 2019).
  - We prove that ε-CONSENSUS HALVING is PPA-Complete for inverse-polynomial ε.
  - This implies that NECKLACE SPLITTING is PPA-Complete.

# NECKLACES AND SANDWICHES

- F.R. and Goldberg. The Complexity of Splitting Necklaces and Bisecting Ham Sandwiches (STOC 2019).
    - We prove that ε-CONSENSUS HALVING is PPA-Complete for inverse-polynomial ε.
    - This implies that NECKLACE SPLITTING is PPA-Complete.
    - This also implies that DISCRETE HAM SANDWICH is PPA-Complete.

# THE STATE OF THE WORLD

Necklace Splitting always exists. — in PPA

ε-Consensus-Halving always exists. — in PPA

Discrete Ham Sandwich always exists. — in PPA

CH is PPA-complete

[F.R. and Goldberg, 2018]

PPA now has a natural complete problem.

PPA: a lonely class
No natural complete problems

# THE STATE OF THE WORLD

NS is PPA-complete

[F.R. and Goldberg 2019]

CH is PPA-complete

[F.R. and Goldberg, 2018]

Necklace Splitting always exists.

in PPA

PPA: a lonely class
No natural complete problems

ε-Consensus-Halving always exists.

in PPA

Discrete Ham Sandwich always exists.

in PPA

DHS is PPA-complete

[F.R. and Goldberg 2019]

PPA now has a natural complete problem.

PPA now has natural complete problems.

# COMPLETE PROBLEMS FOR PPA AND PPAD

- SPERNER, BROUWER, KAKUTANI Papadimitriou (1994).
- NASH Daskalakis, Goldberg and Papadimitriou (2005, 2009), Chen and Deng (2007, 2009).
- EXCHANGE ECONOMY Papadimitriou (1994), Chen, Paparas and Yiannakakis (2013).
- ENVY-FREE CAKE CUTTING Deng, Qi and Saberi (2009, 2012).
- Many more …
- SPERNER for non-orientable spaces Grigni (2001), Friedl, Ivanyos, Santha and Verhoeven (2006).
- 2D-TUCKER, BORSUK-ULAM Aisenberg, Bonet and Buss (2015).
- OCTAHEDRAL TUCKER Deng, Feng and Kulkarni (2017).
- TUCKER, SPERNER on Möbius band and Klein bottle. Deng, Feng, Liu and Qi (2015).

# COMPLETE PROBLEMS FOR PPA AND PPAD

- SPERNER, BROUWER, KAKUTANI Papadimitriou (1994).
- NASH Daskalakis, Goldberg and Papadimitriou (2005, 2009), Chen and Deng (2007, 2009).
- EXCHANGE ECONOMY Papadimitriou (1994), Chen, Paparas and Yiannakakis (2013).
- ENVY-FREE CAKE CUTTING Deng, Qi and Saberi (2009, 2012).
- Many more …
- SPERNER for non-orientable spaces Grigni (2001), Friedl, Ivanyos, Santha and Verhoeven (2006).
- 2D-TUCKER, BORSUK-ULAM Aisenberg, Bonet and Buss (2015).
- OCTAHEDRAL TUCKER Deng, Feng and Kulkarni (2017).
- TUCKER, SPERNER on Möbius band and Klein bottle. Deng, Feng, Liu and Qi (2015).
- CONSENSUS-HALVING, NECKLACE SPLITTING, DISCRETE HAM SANDWICH. F.R., Goldberg (2018, 2019)

# COMPLETE PROBLEMS FOR PPA AND PPAD

- SPERNER, BROUWER, KAKUTANI Papadimitriou (1994).
- NASH Daskalakis, Goldberg and Papadimitriou (2005, 2009), Chen and Deng (2007, 2009).
- EXCHANGE ECONOMY Papadimitriou (1994), Chen, Paparas and Yiannakakis (2013).
- ENVY-FREE CAKE CUTTING Deng, Qi and Saberi (2009, 2012).
- Many more …

- SPERNER for non-orientable spaces Grigni (2001), Friedl, Ivanyos, Santha and Verhoeven (2006).
- 2D-TUCKER, BORSUK-ULAM Aisenberg, Bonet and Buss (2015).
- OCTAHEDRAL TUCKER Deng, Feng and Kulkarni (2017).
- TUCKER, SPERNER on Möbius band and Klein bottle. Deng, Feng, Liu and Qi (2015).
- CONSENSUS-HALVING, NECKLACE SPLITTING, DISCRETE HAM SANDWICH. F.R., Goldberg (2018, 2019)
- More?

# NECKLACE SPLITTING WITH MANY THIEVES

# NECKLACE SPLITTING WITH MANY THIEVES

- For k=2, the problem is PPA-complete.

# NECKLACE SPLITTING WITH MANY THIEVES

- For k=2, the problem is PPA-complete.
- What about general k?

# NECKLACE SPLITTING WITH MANY THIEVES

- For k=2, the problem is PPA-complete.
- What about general k?
  - F.R., Hollender, Sotiraki and Zampetakis.
    A Topological Characterization of Modulo-p Arguments and Implications for Necklace Splitting (SODA 2020).

# NECKLACE SPLITTING WITH MANY THIEVES

- For k=2, the problem is PPA-complete.
- What about general k?
    - **F.R., Hollender, Sotiraki and Zampetakis.**
      **A Topological Characterization of Modulo-p Arguments and Implications for Necklace Splitting (SODA 2020).**
        - Necklace Splitting with p thieves is in PPA-p, for p a prime power.

# NECKLACE SPLITTING WITH MANY THIEVES

- For k=2, the problem is PPA-complete.
- What about general k?
  - F.R., Hollender, Sotiraki and Zampetakis.
    A Topological Characterization of Modulo-p Arguments and Implications for Necklace Splitting (SODA 2020).
    - Necklace Splitting with p thieves is in PPA-p, for p a prime power.
  - What about hardness?

# NECKLACE SPLITTING WITH MANY THIEVES

- For k=2, the problem is PPA-complete.
- What about general k?
  - F.R., Hollender, Sotiraki and Zampetakis.
    A Topological Characterization of Modulo-p Arguments and Implications for Necklace Splitting (SODA 2020).
    - Necklace Splitting with p thieves is in PPA-p, for p a prime power.
  - What about hardness?
  - F.R., Hollender, Sotiraki and Zampetakis.
    Consensus Halving: Does It Ever Get Easier?. (EC 2020).

# NECKLACE SPLITTING WITH MANY THIEVES

- For k=2, the problem is PPA-complete.
- What about general k?
  - F.R., Hollender, Sotiraki and Zampetakis.
    A Topological Characterization of Modulo-p Arguments and Implications for Necklace Splitting (SODA 2020).
    - Necklace Splitting with p thieves is in PPA-p, for p a prime power.
  - What about hardness?
  - F.R., Hollender, Sotiraki and Zampetakis.
    Consensus Halving: Does It Ever Get Easier?. (EC 2020).
    - Some evidence of hardness, but still far from it.

# NECKLACE SPLITTING WITH MANY THIEVES

- For k=2, the problem is PPA-complete.
- What about general k?
    - F.R., Hollender, Sotiraki and Zampetakis.
      A Topological Characterization of Modulo-p Arguments and Implications for Necklace Splitting (SODA 2020).
        - Necklace Splitting with p thieves is in PPA-p, for p a prime power.
    - What about hardness?
    - F.R., Hollender, Sotiraki and Zampetakis.
      Consensus Halving: Does It Ever Get Easier?. (EC 2020).
        - Some evidence of hardness, but still far from it.
- Biggest open problem: Is Necklace Splitting with p thieves PPA-p complete?