KEEPING YOUR DISTANCE IS HARD

Joint work with Kyle Burke, Melissa Huggan, and Svenja Huntemann

Silvia Heubach
California State University Los Angeles

CGTC47, March9 , 2016

1

The Basics

- A two-player game is called a **combinatorial game** if there is no randomness involved and all possible moves are known to each player.
- A combinatorial game is called **impartial** if both players have the same moves, and **partizan** otherwise.
- **Examples:**

- We consider the case where the **last player to move wins** (normal play).

2

Distance Games

- **GRAPHDISTANCE(*D*,*S*)** is played on a graph G on which two players, **BLue** (Left) and **Red** (Right), alternately place pieces on empty vertices of G according to the restrictions of the sets ***D*** and ***S***.
- All vertices are empty at the beginning of the game.
- A **BLue** piece and a **Red** piece are not allowed to have distance ***d*** if ***d*** ∈ ***D*** (*D* is for "different")
- Two **BLue** pieces or two **Red** pieces are not allowed to have distance ***s* if *s*** ∈ ***S*** (*S* is for "same")
- Pieces may not be removed once they are placed, nor may they be moved.

3

Known Distance Games

- **COL**: adjacent vertices cannot have the **same** color

  COL= **GRAPHDISTANCE(Ø ,{1})**

- **SNORT**: adjacent vertices cannot have **different** colors.

  SNORT = **GRAPHDISTANCE({1}, Ø)**

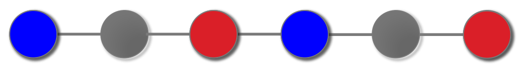
- **NODEKAYLES**: adjacent vertices cannot **both be colored**.

  NODEKAYLES = **GRAPHDISTANCE({1},{1})**

4

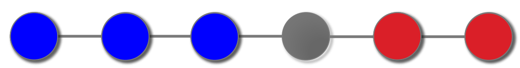
Let's Play a Game (or two)

COL – adjacent vertices cannot have **SAME** color



Game is over – **Red wins**!

SNORT – adjacent vertices cannot have **DIFFERENT** color



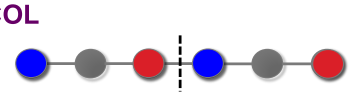
Game is over - **Blue wins!**

5

How Can We Analyze a Game?

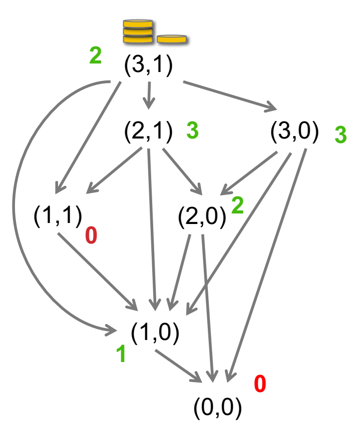
- Strategy stealing, mirroring

COL



- Create a game graph and then recursively label each position, starting from the terminal positions, as to who wins

Example: **NIM**



6

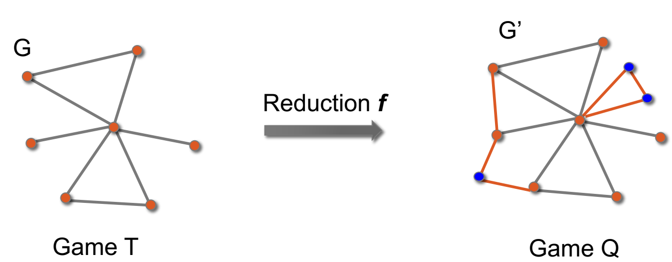
Complexity of Distance Games

- How hard is is to decide who wins from a given position in **GRAPHDISTANCE(*D*,*S*)** for general sets ***D*** and ***S***?
- We know that **COL**, **SNORT**, **NODEKAYLES**, and **BIGRAPH NODEKAYLES** played on graphs are **PSPACE-hard**
- If we know a game **T** is **PSPACE-hard** and want to show that another game **Q** is also **PSPACE-hard**, we need to find a function *f*, called a **reduction from T to Q**, such that
  - ***f*** maps the positions of **T** to the positions of **Q**
  - ***f*** can be computed in polynomial time
  - ***f*** preserves winnability

7

Specifics of the Reduction

- The reduction transforms the graph G on which game T is played to a graph G' on which Q is played via **insertion of a subgraph** called **gadget**



Known to be
PSPACE-hard

To be shown to be
PSCPAE-hard

8

Main Result

THEOREM
The games **GRAPHDISTANCE(*D*,*S*)** are **PSPACE-hard** when either ***S*** or ***D*** equals **{1,2,…,r}** and the other is a **subset** (or equal) to {1,2,…,r}.

We will illustrate the proof idea with an example of a generalization of **SNORT** = **GRAPHDISTANCE({1} ,Ø):**

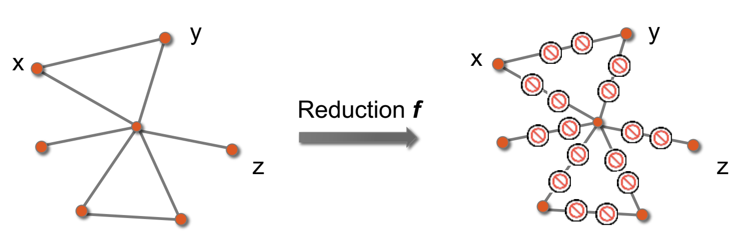
ENSNORT(r) := **GRAPHDISTANCE({1,2,…,r} ,Ø)**

9

Example for ENSNORT(3)

Play **SNORT**
***D* = {1}, *S* = Ø**

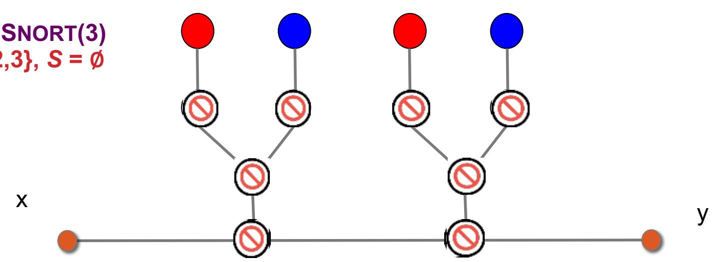
Play **ENSNORT(3)**
***D* = {1,2,3}, *S* = Ø**



10

Forbidden vertex gadget ENSNORT(3)

Play **ENSNORT(3)**
***D* = {1,2,3}, *S* = ∅**



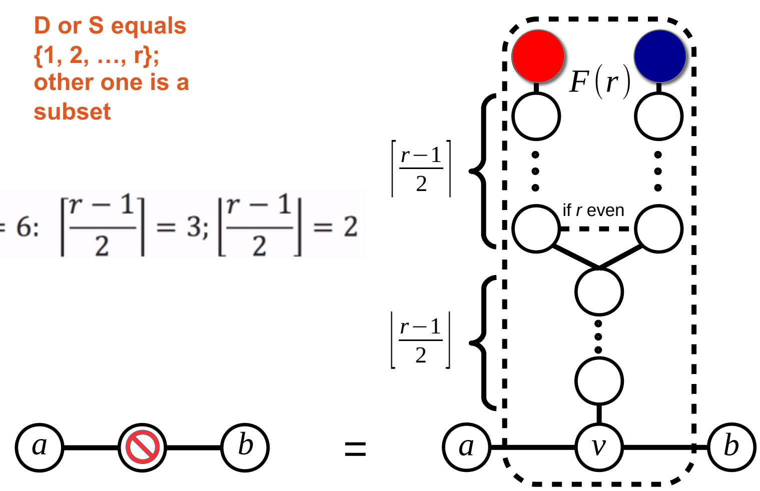
- Works also for ***S* ⊂*D*** and **max(*S*) ≤ r**

11

Forbidden Vertex Gadget F(r)

D or S equals {1, 2, …, r}; other one is a subset

$$r = 6:\ \left\lceil \frac{r-1}{2} \right\rceil = 3;\ \left\lfloor \frac{r-1}{2} \right\rfloor = 2$$



12

Proof of Main Result

THEOREM

The games **GRAPHDISTANCE(*D*; *S*)** are **PSPACE-hard** when either ***S*** or ***D*** equals **{1,2,…,r}** and the other is a **subset** (or equal) to {1,2,…,r}.

Proof Outline: For **GRAPHDISTANCE(*D*; *S*)** with

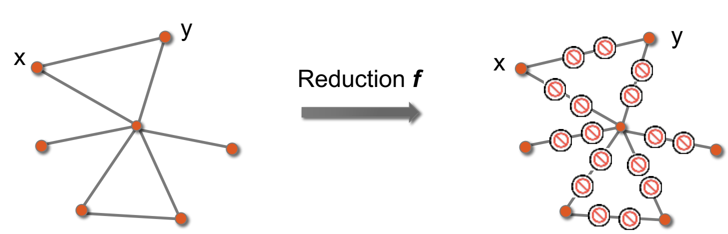
- ***D* = {1,2,…,r}, *S* ⊂*D*,** and **max(*S*) < r**, we reduce from **SNORT**
- ***S* = {1,2,…,r}, *D* ⊂*S***, and **max(*D*) < r**, we reduce from **COL**
- ***S*** or **D** is **{1,2,…,r}** and **max(*D*) = max(*S*)**, we reduce from **NODEKAYLES**

13

Why is case max(S) = max(D) different?

Play **SNORT**
***D* = {1}, *S* = ∅**

Play **GRAPHDISTANCE(D,S)**
***D* = {1,2,3}, *S* = {1,3}**



- We **can** color x and y in the same color in **SNORT**, but **cannot** in **GRAPHDISTANCE(D,S)**, so winnability is no longer the same.

14

Reduction for max(S) = max(D)

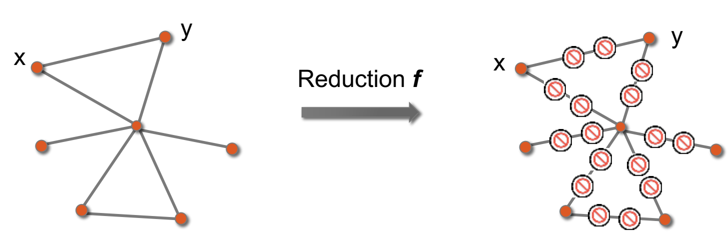
- When max(S) = max(D) = n, then the maximal reach for both same and different colors is the same
- **NODEKAYLES** = **GRAPHDISTANCE({1},{1})** fits the bill
- For the reduction, we replace every edge in G by n-1 gadgets of size n

15

Reduction for max(S) = max(D)

Play **NODEKAYLES**
***D* = {1}, *S* = {1}**

Play **GRAPHDISTANCE(D,S)**
***D* = {1,2,3}, *S* = {1,3}**



- We **cannot** color x and y in the same color in **NODEKAYLES**; likewise in **GRAPHDISTANCE(D,S)**, so winnability **is** the same.

16

Open Problem

Problem

Is **GRAPHDISTANCE(*D*; *S*)** **PSPACE-hard** for cases not covered by our results?

17

THANK YOU!

sheubac@calstatela.edu

Slides will be posted on my web site
http://web.calstatela.edu/faculty/sheubac/#presentations

18

References

1. E.R. Berlekamp, J.H. Conway, and R.K. Guy. *Winning Ways for Your Mathematical Plays* Vol. 1-4 (2014) AK Peters Ltd., Wellesley, MA, 2nd edition
2. C.L. Bouton (1901/02) “Nim, a game with a complete mathematical theory.” *Ann. of Math*. (2) 3(1-4):35–39.
3. J.I. Brown, D. Cox, A. Hoefel, N. McKay, R. Milley, R.J. Nowakowski, and A.A. Siegel. “Polynomial profiles of placement games.” To appear in: *Games of No Chance 5*
4. K. Burke and R.A. Hearn. “PSPACE-Complete Two-Color Placement Games.” Preprint 2015.
5. E.D. Demaine and R.A. Hearn (2009) “Playing Games with Algorithms: Algorithmic Combinatorial Game Theory.” In: M.H. Albert and R.J. Nowakowski (eds) *Games of No Chance 3*. Mathematical Sciences Research Institute Publications, Vol. 56, Cambridge University Press, pp 3–56.

19

References

6. S. Faridi, S. Huntemann, and R.J. Nowakowski. “Games and Complexes I: Transformation via Ideals.” To appear in: *Games of No Chance 5*, arXiv:1310.1281
7. A.S. Fraenkel and D. Lichtenstein “Computing a perfect strategy for n×n chess requires time exponential in N.” In: *Automata, languages and programming* (Akko, 1981). pp 278– 293.
8. S. Huntemann and R.J. Nowakowski (2014) “Doppelgänger Placement Games.” Recreational Mathematics Magazine 1:55–61.
9. C.H. Papadimitriou (1994) “On the complexity of the parity argument and other inefficient proofs of existence.” J. Comput. System Sci. 48(3):498–532.
10. T.J. Schaefer (1978) “On the complexity of some two-person perfect-information games.” J. Comput. System Sci., 16(2):185–225.

20