CSCI 4560/6560 Computational Geometry

https://www.cs.rpi.edu/~cutler/classes/computationalgeometry/F23/

Lecture 25:
Sprouts &
Brussel Sprouts

Outline for Today

- Final Project Presentation Schedule
- Last Time: Robot Motion Planning, Minkowski Sums, etc.
- Paper & Pencil Game of Sprouts
- Computer Analysis of Sprouts
- Brussel Sprouts
- Hybrid Sprouts & Brussel Sprouts
- Next Time: Final Project Presentations!

Final Project Presentation Schedule

8 min for individual + 1 min for questions , 16 min for team of 2 + 2 min for questions

2:00

2:18

2:27

2:36

2:54

3:03

3:12

3:30

3:39

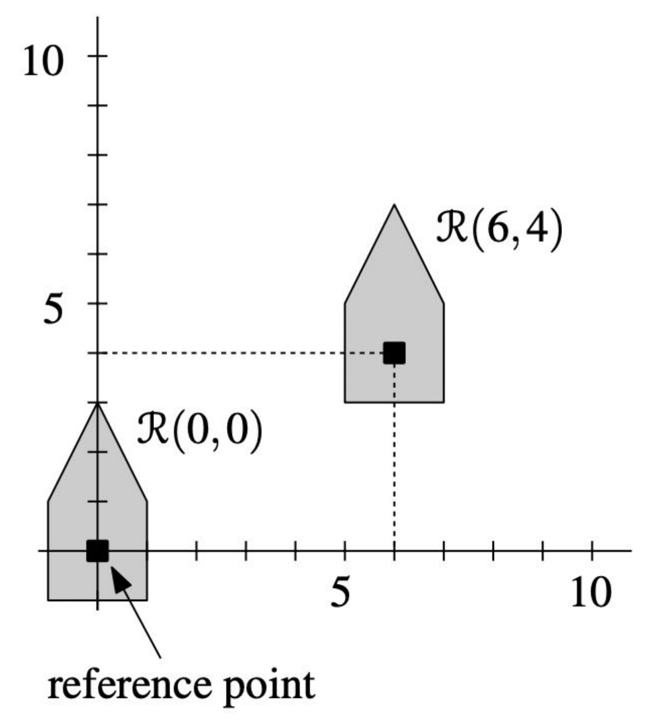
3:48 done!

Outline for Today

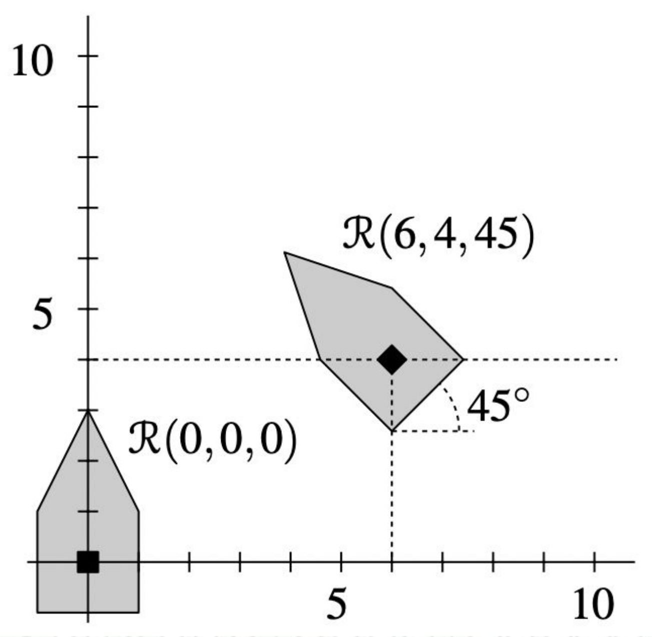
- Final Project Presentation Schedule
- Last Time: Robot Motion Planning, Minkowski Sums, etc.
- Paper & Pencil Game of Sprouts
- Computer Analysis of Sprouts
- Brussel Sprouts
- Hybrid Sprouts & Brussel Sprouts
- Next Time: Final Project Presentations!

Robot Degree of Freedom (DOF)

2D w/ Translation only → 2 DOF



2D w/ Translation & Rotation → 3 DOF

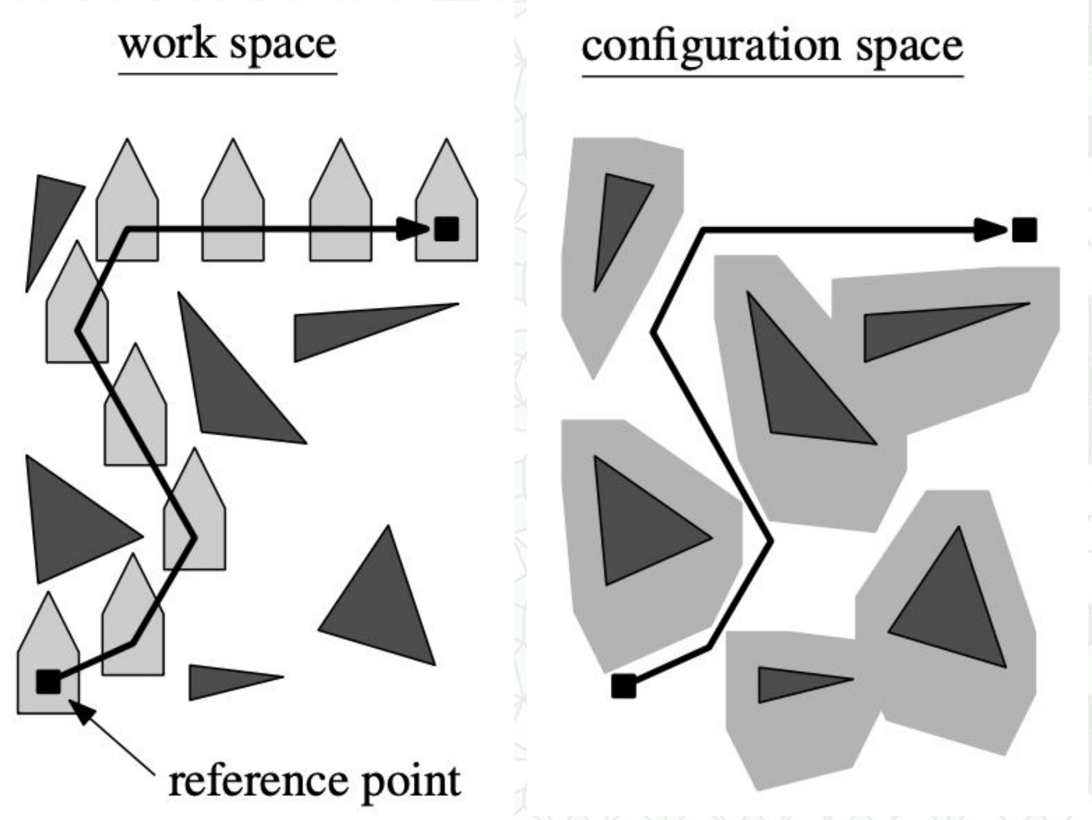


Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 13

Configuration Space

2D w/ translation only → 2 DOF

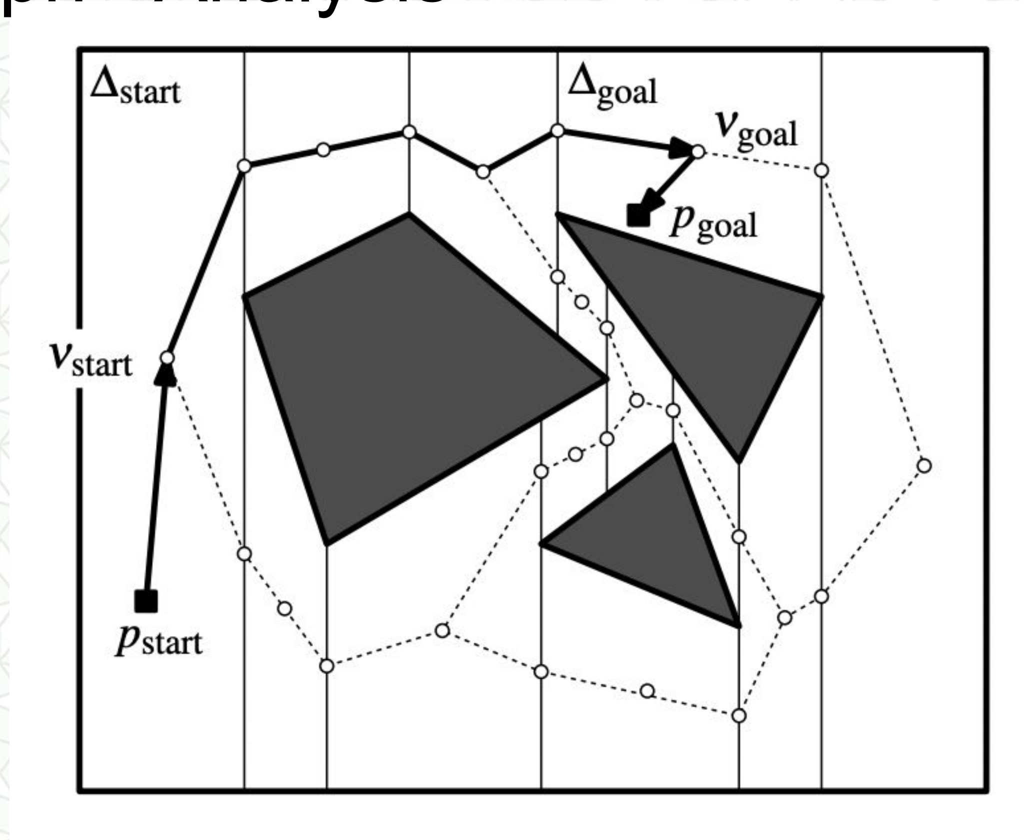
- The dimensions of configuration space match the DOF of the robot
- Usually configuration space is higher dimensional than the environment/workspace
- It is often useful to construct, visualize, and even solve the problem in "configuration space"



Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 13

Motion Planning Graph - Analysis

- Size of Trapezoid Map
  → O(n)
- Build Trapezoid Map
  → O(n log n)
- Locate start/end trapezoid
  → O(log n)
- Breadth first search
  → O(n)

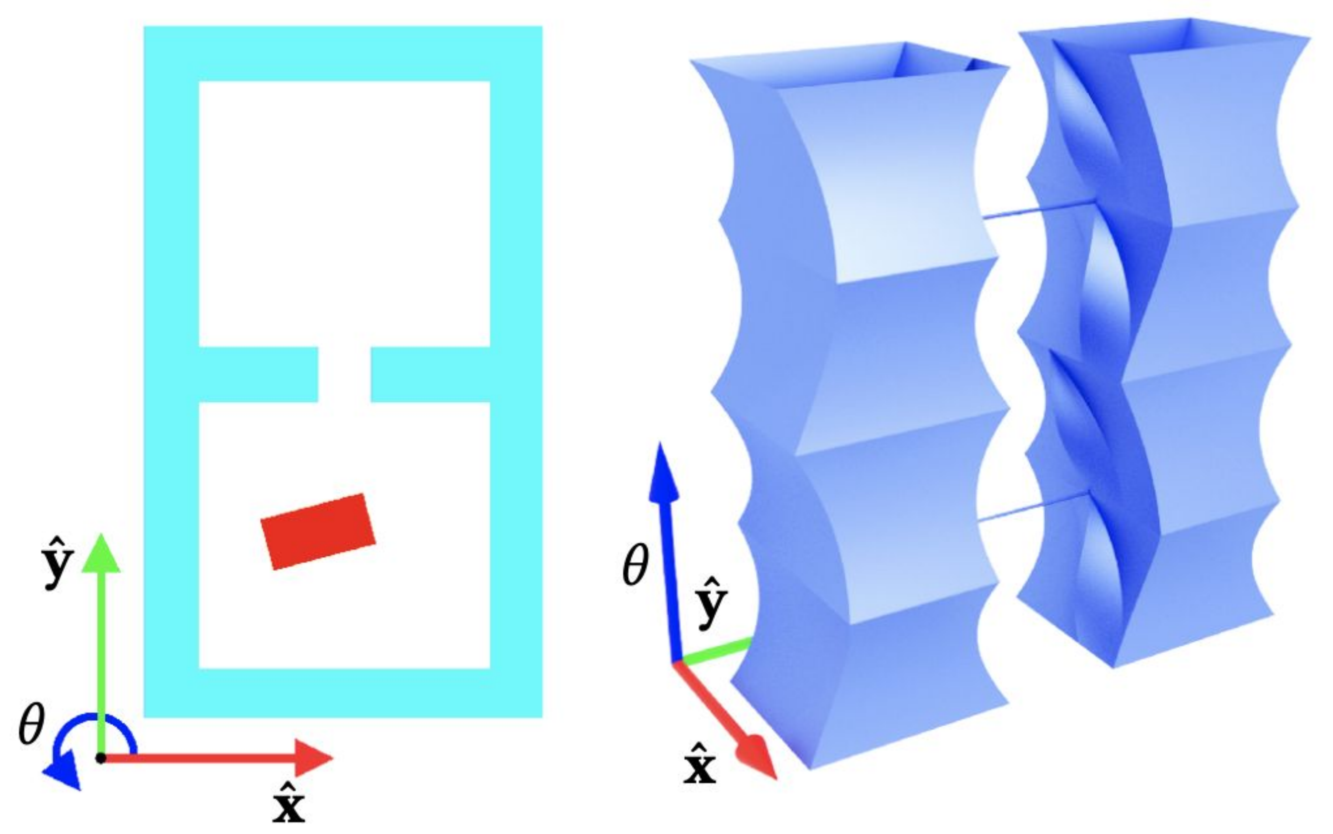


Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 13

Searching Configuration Space

"C-Space Tunnel Discovery for Puzzle Path Planning",
Zhang, Belfer, Kry, & Voucha,
SIGGRAPH 2020.

- Dimensionality becomes infeasible to construct & exhaustively search
- Randomized search is necessary

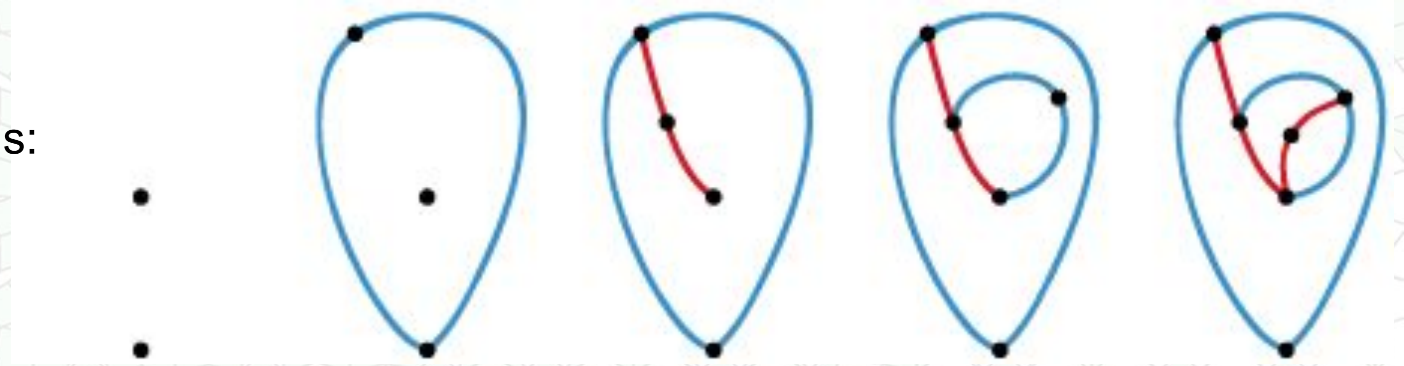


Outline for Today

- Final Project Presentation Schedule
- Last Time: Robot Motion Planning, Minkowski Sums, etc.
- Paper & Pencil Game of Sprouts
- Computer Analysis of Sprouts
- Brussel Sprouts
- Hybrid Sprouts & Brussel Sprouts
- Next Time: Final Project Presentations!

Sprouts Game Rules

Paper and Pencil Games
http://www.papg.com/show?1TMQ



- Draw *n* spots
- Players take turns:
  - Draw a line joining two spots, or a single spot to itself.
  - The line must not cross another line or pass through another spot.
  - Draw a spot on the new line.
  - No more than three lines can emerge from any spot.
- Normal Winning Condition: Winner is last person to make a move
- *Misère Winning Condition: Winner is first person who cannot make a move*

Sprouts Analysis

- Starting with *n* dots
- What's the maximum number of turns?
  What's the maximum number of lines drawn?
  What's the maximum number of new points added?

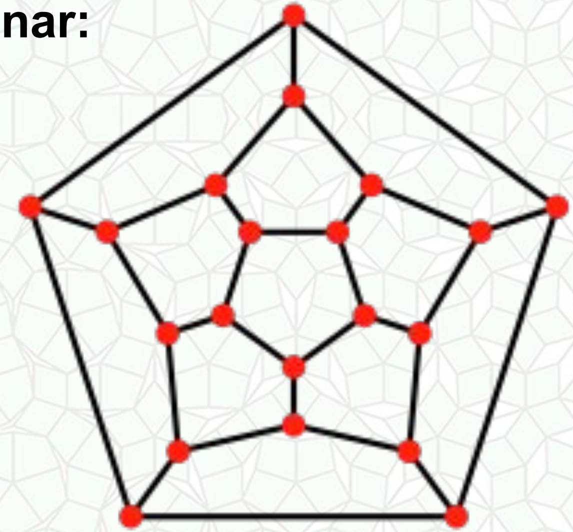
Sprouts Analysis

- Starting with *n* dots
- What’s the minimum number of turns?

Definition: Planar Graph

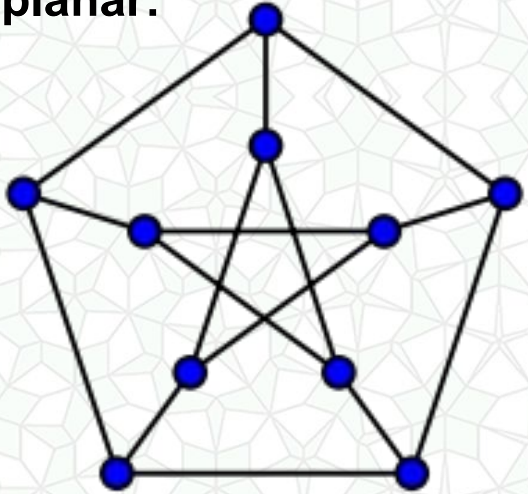
- Can be drawn in 2D without any edges crossing

planar:



https://en.wikipedia.org/wiki/Planar_graph#/media/File:Dodecahedron_schlegel.svg

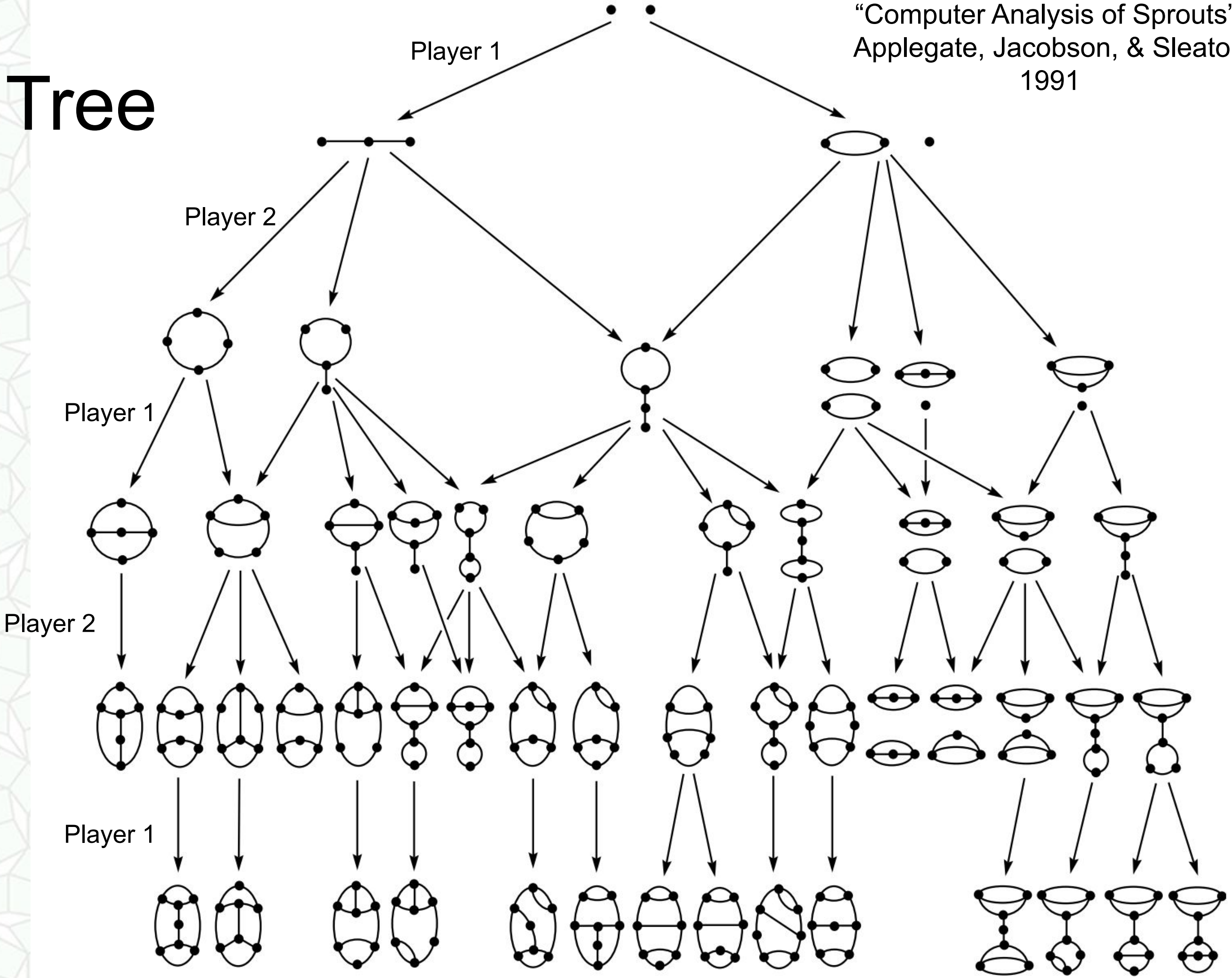
NOT planar:



https://en.wikipedia.org/wiki/Petersen_graph#/media/File:Petersen1_tiny.svg

Sprout Game Tree

Even for just
2 starting spots,
the full tree
of moves is
surprisingly large!

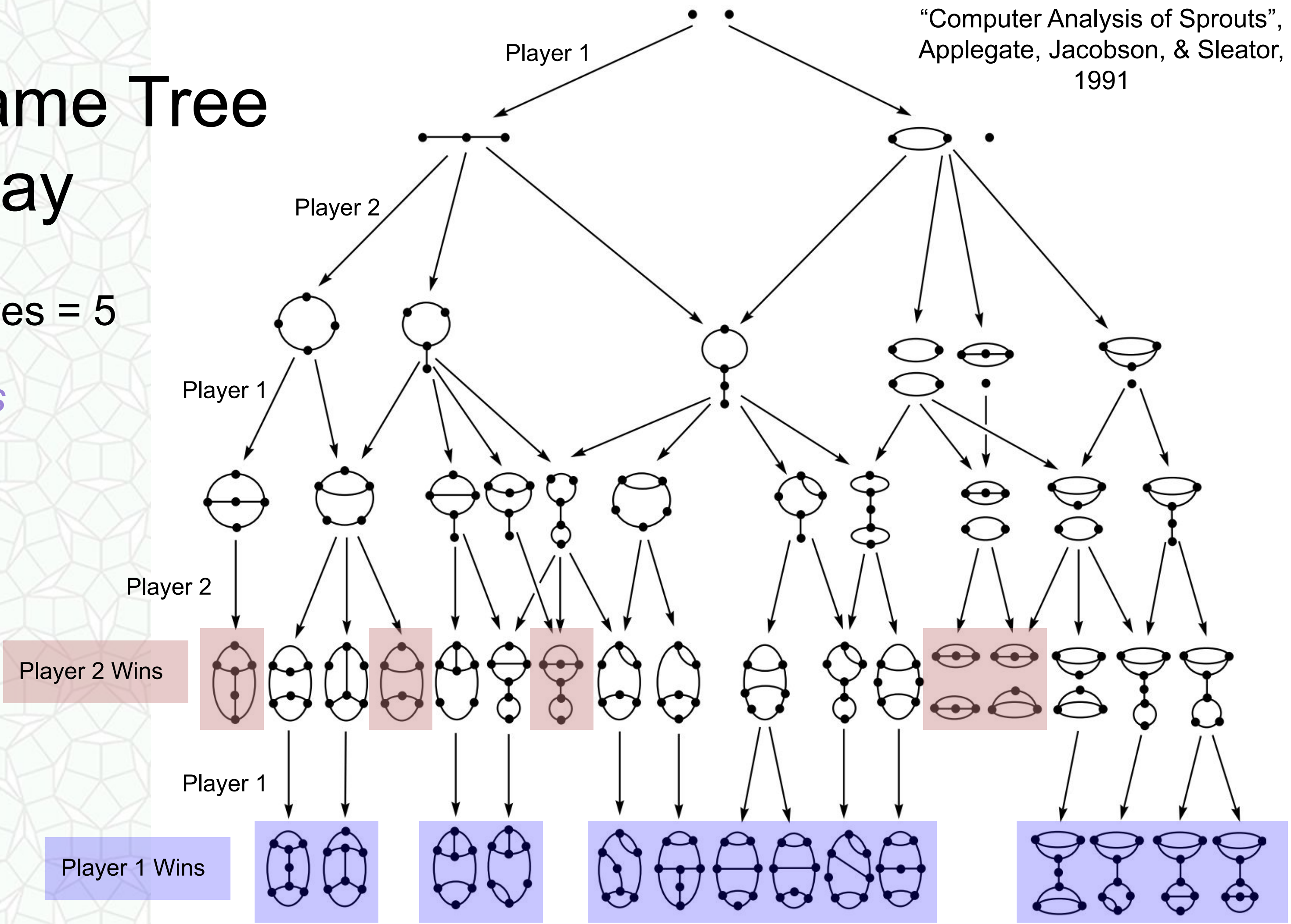


“Computer Analysis of Sprouts”,
Applegate, Jacobson, & Sleator,
1991

Sprout Game Tree
Normal Play

Maximum # moves = 5

If the game ends
in 5 moves,
Player 1 makes
the last move
and wins



“Computer Analysis of Sprouts”, Applegate, Jacobson, & Sleator, 1991

Player 1

Player 2

Player 1

Player 2

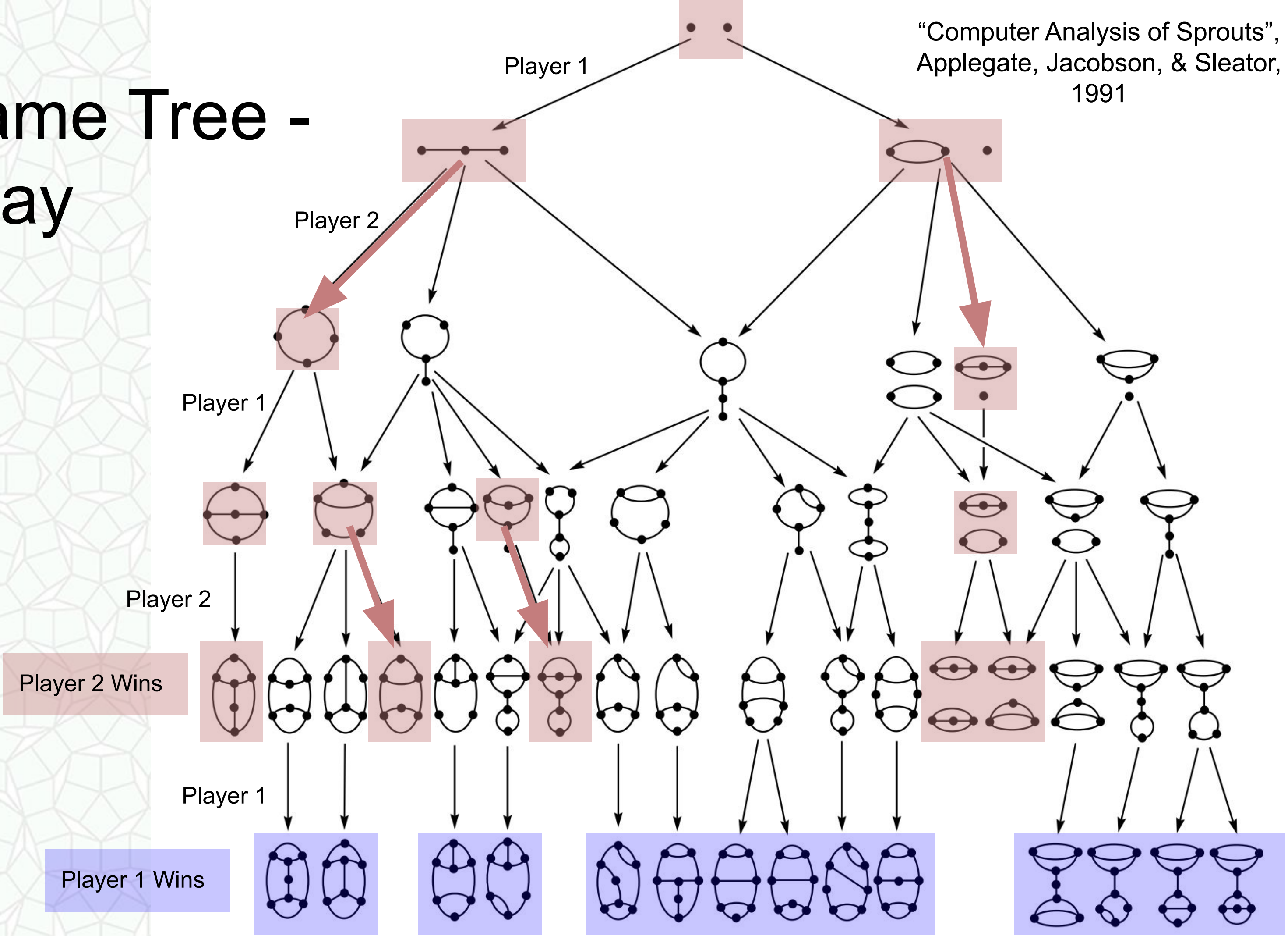
Player 2 Wins

Player 1

Player 1 Wins

Sprout Game Tree - Normal Play

If Player 2 can separate and isolate the 2 final dots, ending in 4 moves, they win!

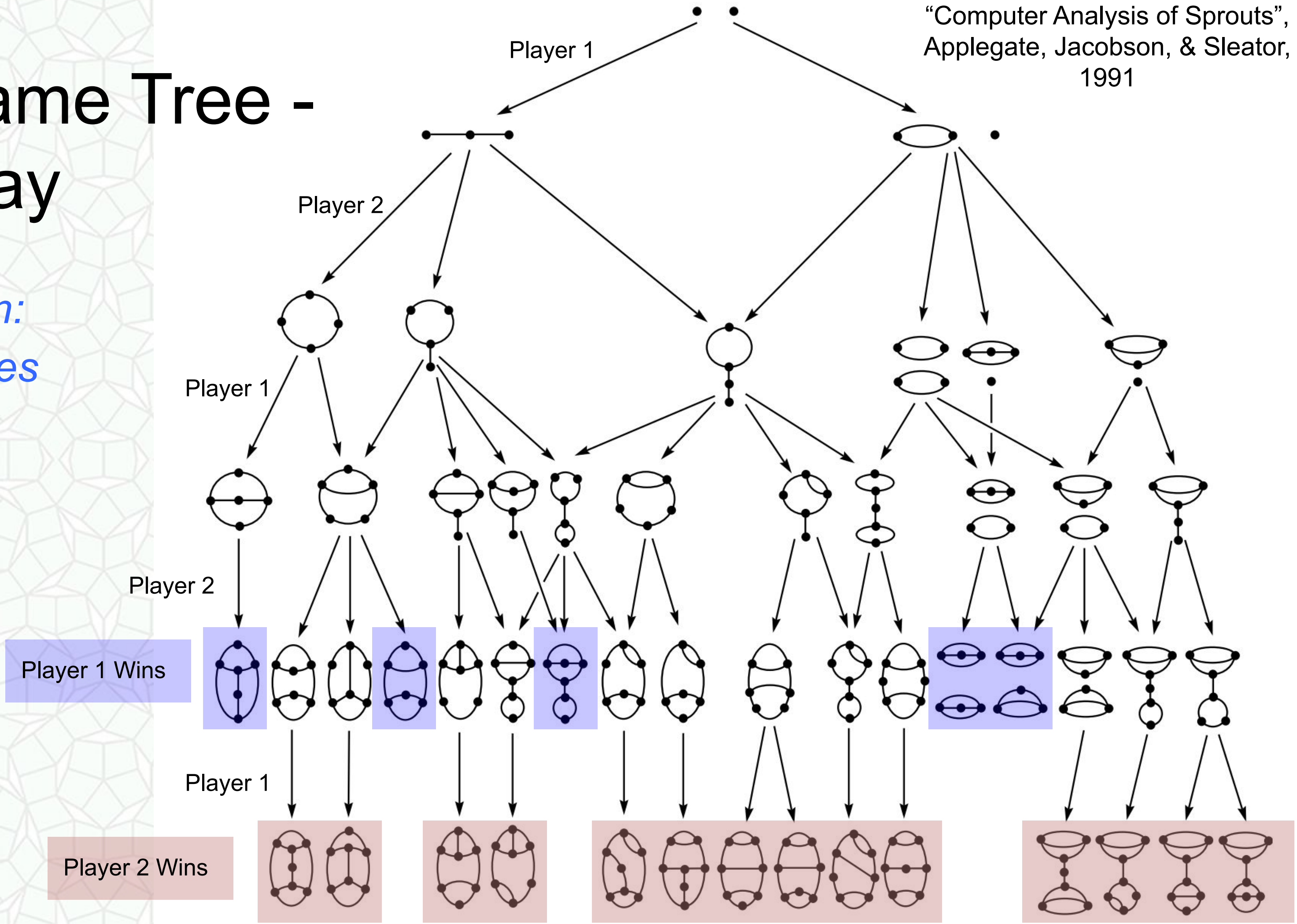


"Computer Analysis of Sprouts", Applegate, Jacobson, & Sleator, 1991

Sprout Game Tree - Misère Play

Flip the condition:
Player who makes last move loses

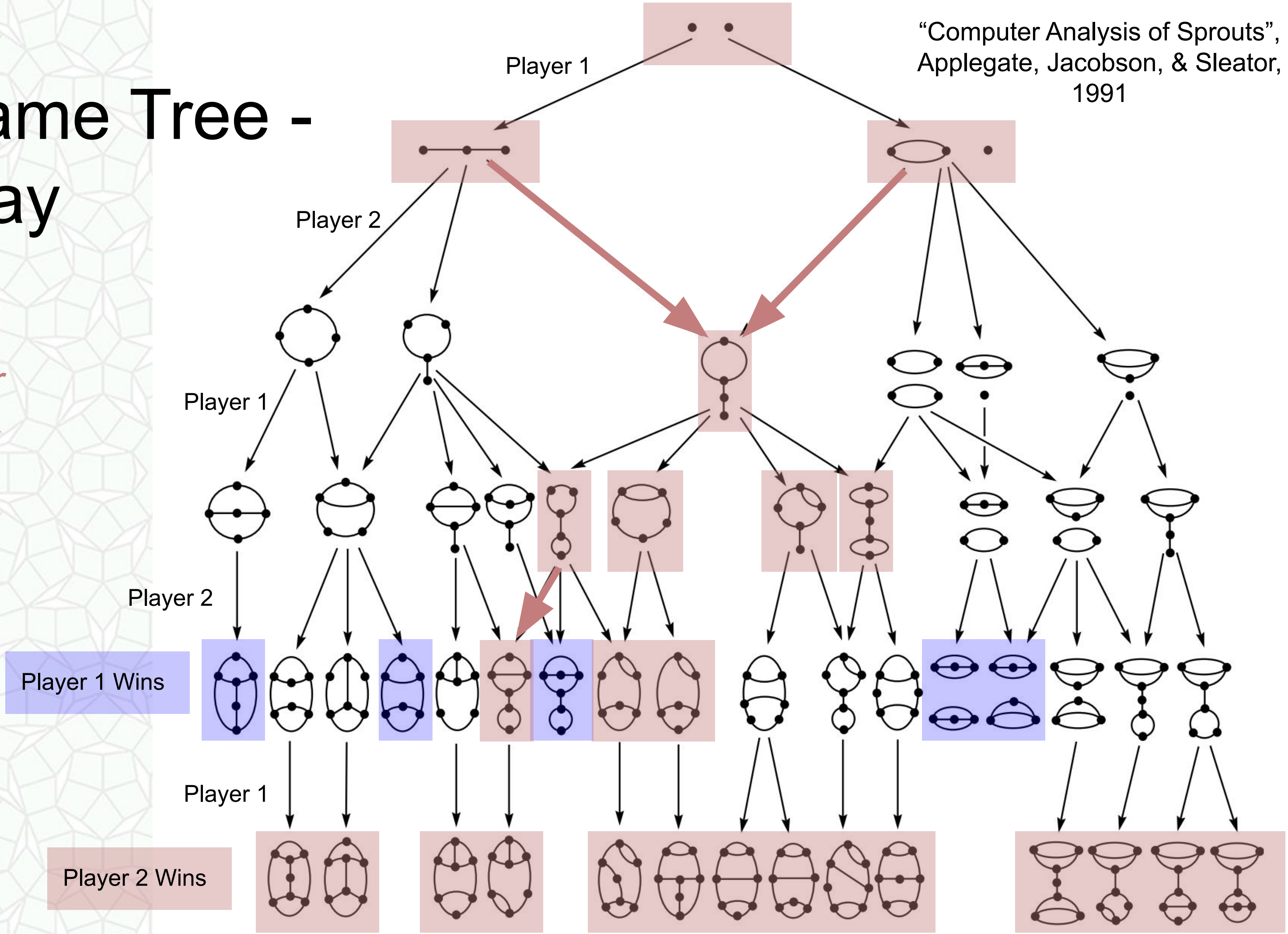
Does this mean Player 1 can guarantee a win?



"Computer Analysis of Sprouts", Applegate, Jacobson, & Sleator, 1991

Sprout Game Tree - Misère Play

Nope! Player 2 is still the winner with perfect play



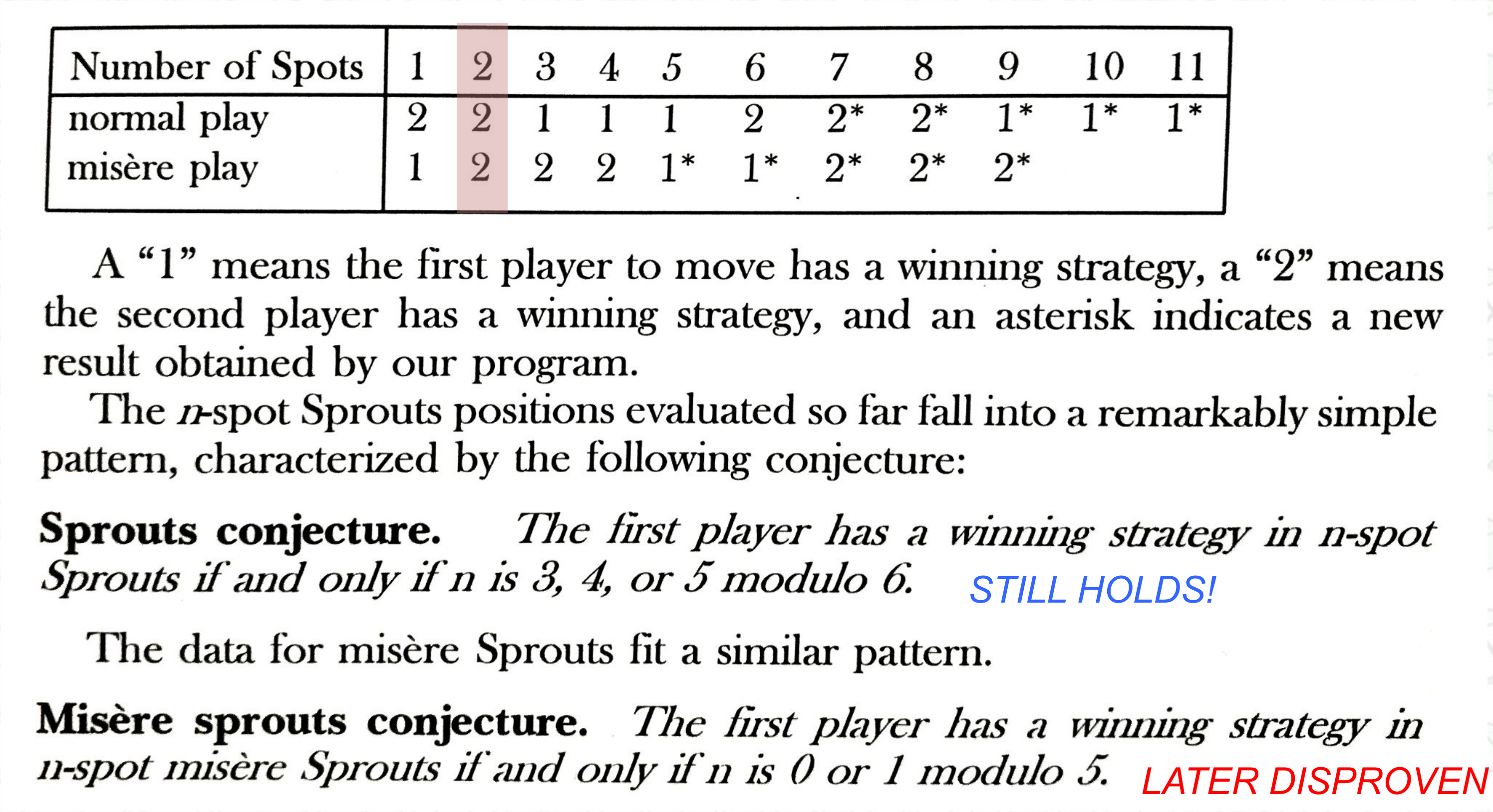
“Computer Analysis of Sprouts”, Applegate, Jacobson, & Sleator, 1991

Outline for Today

- Final Project Presentation Schedule
- Last Time: Robot Motion Planning, Minkowski Sums, etc.
- Paper & Pencil Game of Sprouts
- Computer Analysis of Sprouts
- Brussel Sprouts
- Hybrid Sprouts & Brussel Sprouts
- Next Time: Final Project Presentations!

Sprouts Computer Analysis

"Computer Analysis of Sprouts",
Applegate, Jacobson, & Sleator,
In *The Mathemagician and the Pied Puzzler:*
A Collection in Tribute to Martin Gardner (1999)



Number of Spots	1	2	3	4	5	6	7	8	9	10	11
normal play	2	2	1	1	1	2	2*	2*	1*	1*	1*
misère play	1	2	2	2	1*	1*	2*	2*	2*		

A "1" means the first player to move has a winning strategy, a "2" means the second player has a winning strategy, and an asterisk indicates a new result obtained by our program.

The *n*-spot Sprouts positions evaluated so far fall into a remarkably simple pattern, characterized by the following conjecture:

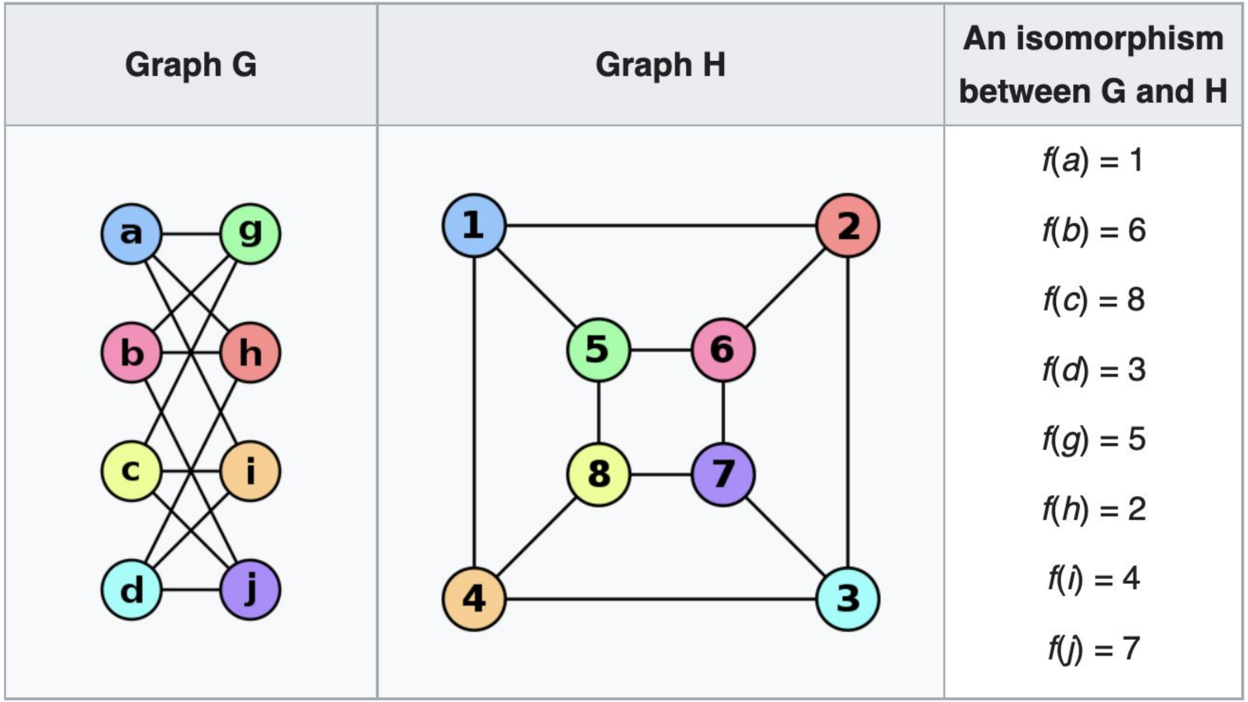
Sprouts conjecture. *The first player has a winning strategy in n-spot Sprouts if and only if n is 3, 4, or 5 modulo 6.* STILL HOLDS!

The data for misère Sprouts fit a similar pattern.

Misère sprouts conjecture. *The first player has a winning strategy in n-spot misère Sprouts if and only if n is 0 or 1 modulo 5.* LATER DISPROVEN

Definition: Graph Isomorphism

Two graphs are isomorphic if there is a bijection between the labels of the graphs, and an edge exists between a pair of vertices in one graph if and only if an edge exists between the corresponding vertices in the other graph.

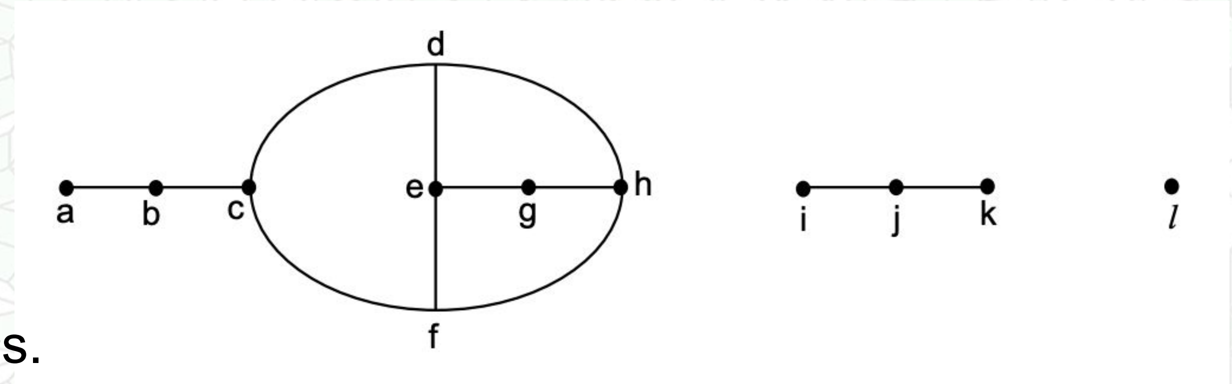


Graph G	Graph H	An isomorphism between G and H
		$f(a) = 1$ $f(b) = 6$ $f(c) = 8$ $f(d) = 3$ $f(g) = 5$ $f(h) = 2$ $f(i) = 4$ $f(j) = 7$

https://en.wikipedia.org/wiki/Graph_isomorphism

Sprouts Analysis Definitions

- Each spot is given a unique name.
- The curves of the sprout graph divide the plane into *regions*.



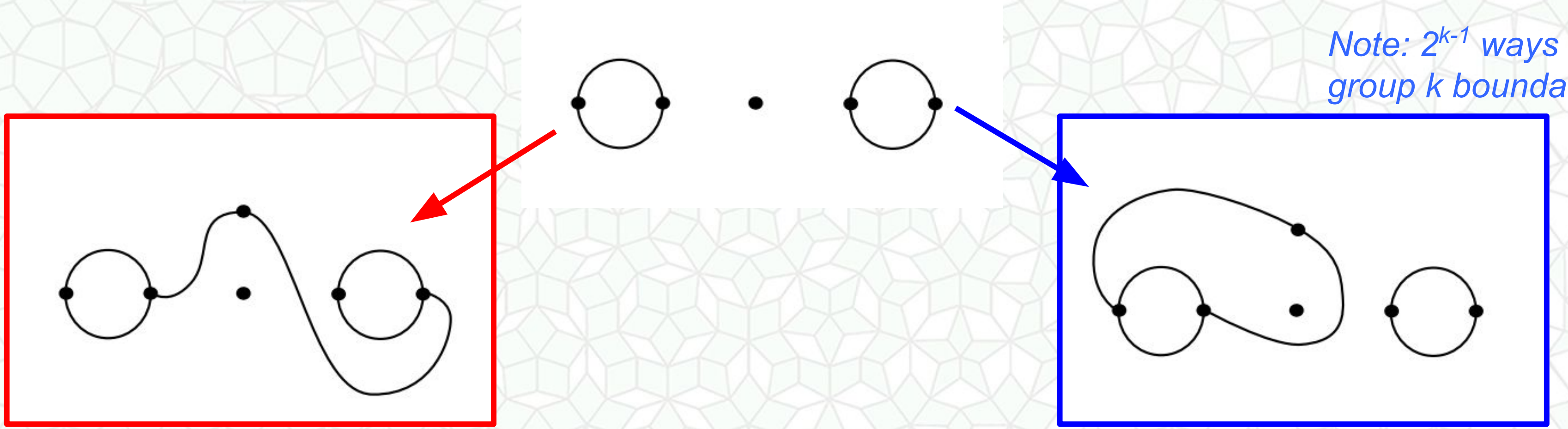
- The loop of spots and curves surrounding a region is called a *boundary*.
- The sprout graph can be encoded in *set representation*:
  { { (abcdhfcb) (ijkj) (l) } { (cfed) } { (degh) } { (efhg) } }

star of boundary　end of boundary　start of region　end of region

Sprouts Analysis Move Definitions

A *two-boundary move* joins spot x on boundary B_1 to spot y on boundary B_2 and adds spot z. Boundaries B_1 and B_2 are now joined. No new regions are formed.

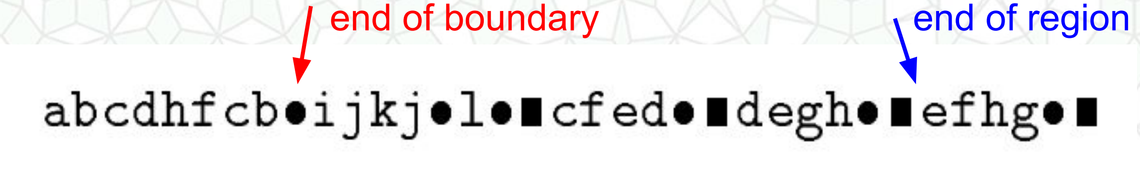
A *one-boundary move* connects spots on the same boundary of region R. Region R is split into two subregions R_1 and R_2. All boundaries of and within R must be designated for R_1, or R_2.



Note: 2^{k-1} ways to group k boundaries

Sprouts Analysis Representation Simplification

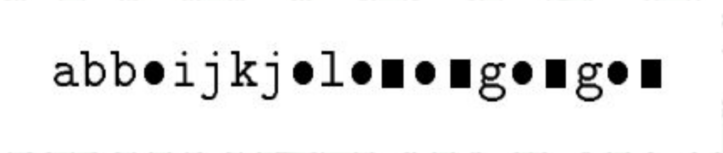
The String representation
can be simplified/compressed:

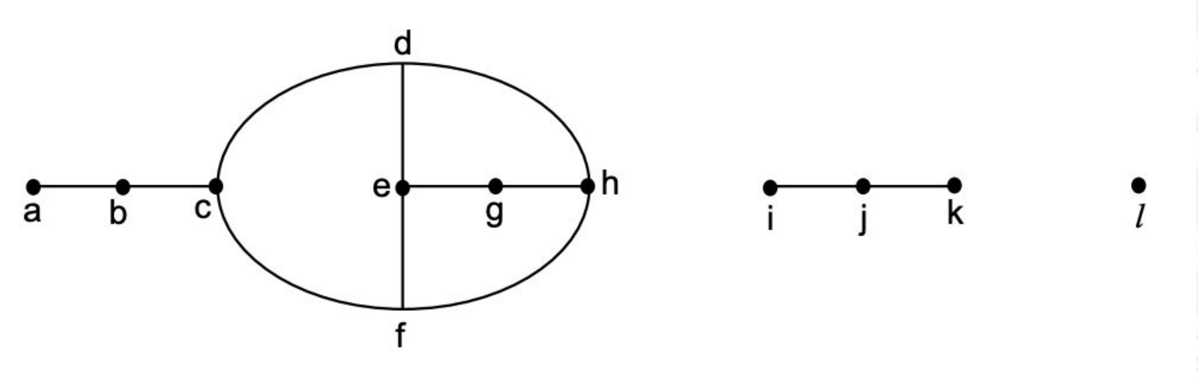

end of boundary

end of region

abcdhfcb•ijkj•l•■cfed•■degh•■efhg•■

- Spots of degree 3 can be thrown away
- Regions with fewer than 2 lives
  can be thrown away


abb•ijkj•l•■•■g•■g•■



Sprouts Analysis Representation Simplification

The String representation
can be simplified/compressed:

`abcdhfcb●ijkj●l●■cfed●■degh●■efhg●■`

- Spots of degree 3 can be thrown away
- Regions with fewer than 2 lives can be thrown away

`abb●ijkj●l●■●■g●■g●■`

- Boundaries with no live spots can be thrown away

`abb●ijkj●l●■`

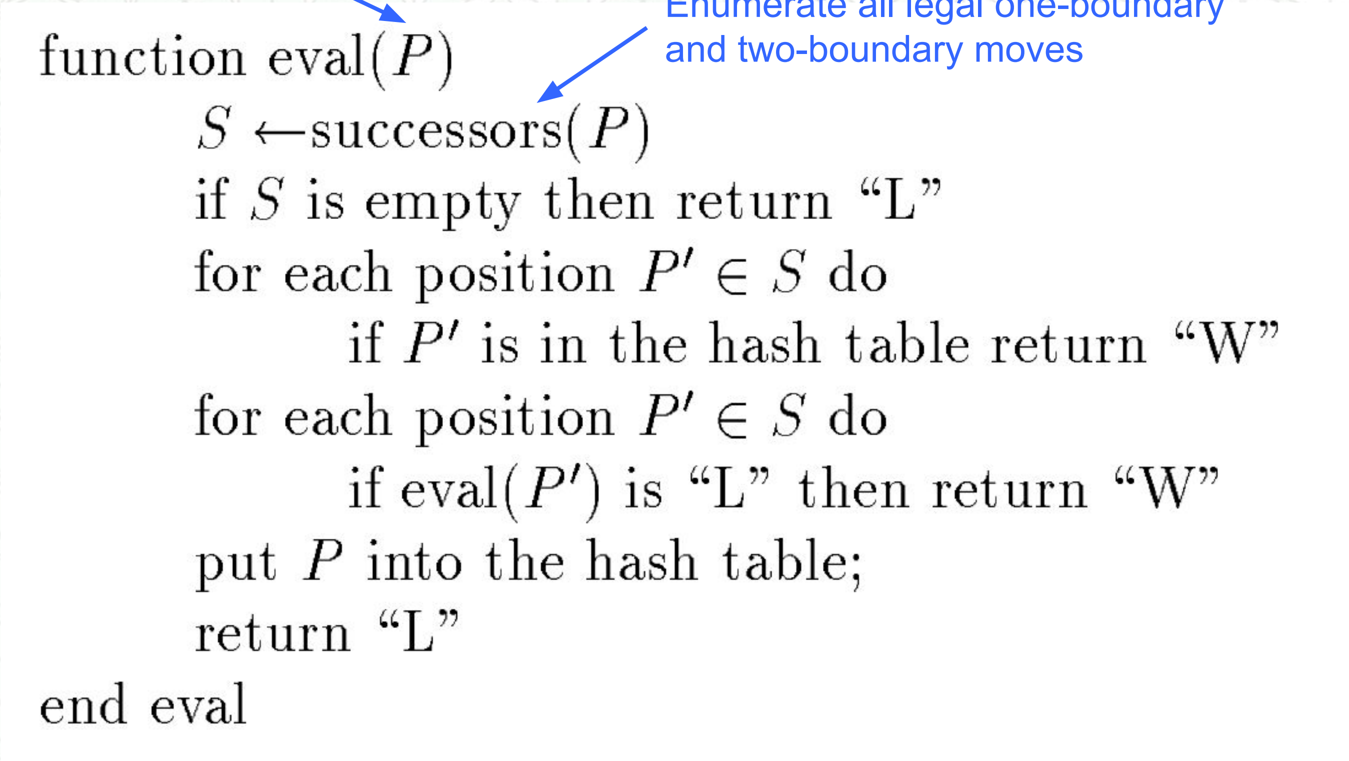
- Spots of degree zero or one do not appear on more than one boundary, so they don’t need unique names, label them 0 and 1.

`1bb●1j1j●0●■`

- If a degree two spot has no live spots between its two occurrences on one boundary, it only needs to be listed once.

`1b●1j1j●0●■`

Sprouts Analysis Pseudocode



Input sprout graph

Enumerate all legal one-boundary and two-boundary moves

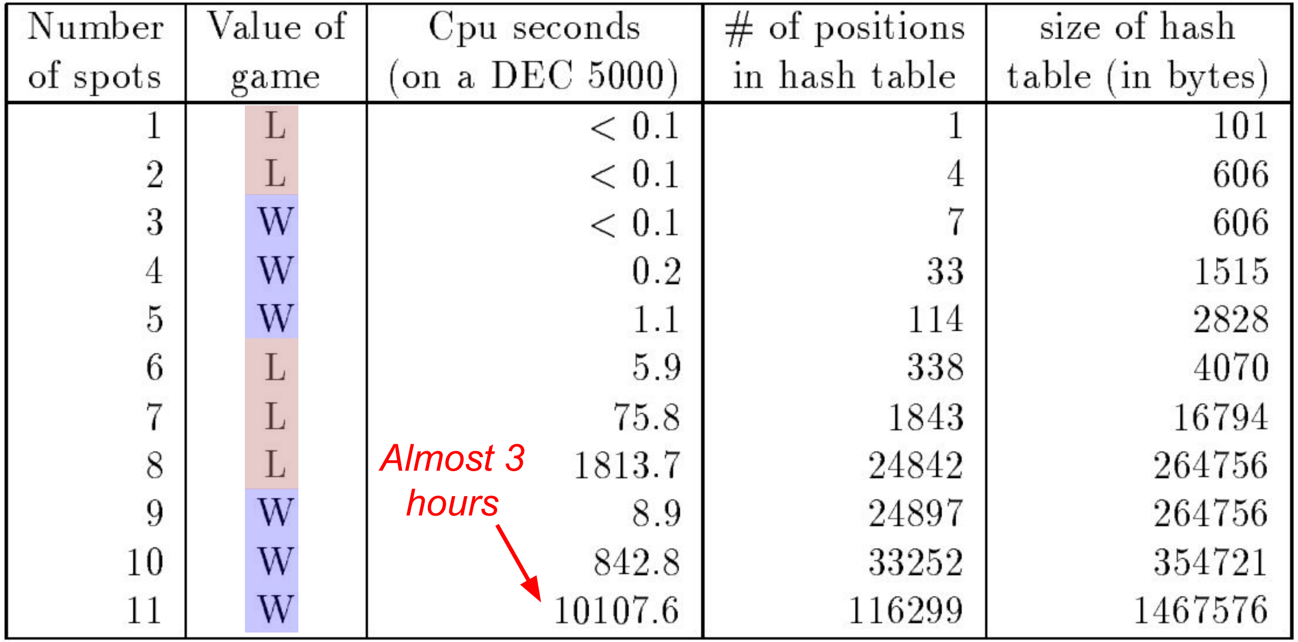
```
function eval($P$)
    $S$ ←successors($P$)
    if $S$ is empty then return “L”
    for each position $P' \in S$ do
        if $P'$ is in the hash table return “W”
    for each position $P' \in S$ do
        if eval($P'$) is “L” then return “W”
    put $P$ into the hash table;
    return “L”
end eval
```

Sprouts Analysis Implementation

- Lexicographically sort the boundaries & regions to remove (significantly reduce) duplicate/isomorphic graphs.
- Use a hash table to store compressed set representation of all sprout graphs that we have previously examined & labeled “W” or “L”.

- Implementation notes with 1990 hardware: *may be different now*
  - Memory was a more significant resource limitation than CPU/time.
  - There are 10X more “W” configurations than “L” configurations.
  - Therefore, only store “L” configurations in the hash table & recompute successors when needed.

Sprouts Perfect Play Winner - Normal Play



Number of spots	Value of game	Cpu seconds (on a DEC 5000)	# of positions in hash table	size of hash table (in bytes)
1	L	< 0.1	1	101
2	L	< 0.1	4	606
3	W	< 0.1	7	606
4	W	0.2	33	1515
5	W	1.1	114	2828
6	L	5.9	338	4070
7	L	75.8	1843	16794
8	L	1813.7	24842	264756
9	W	8.9	24897	264756
10	W	842.8	33252	354721
11	W	10107.6	116299	1467576

Almost 3 hours

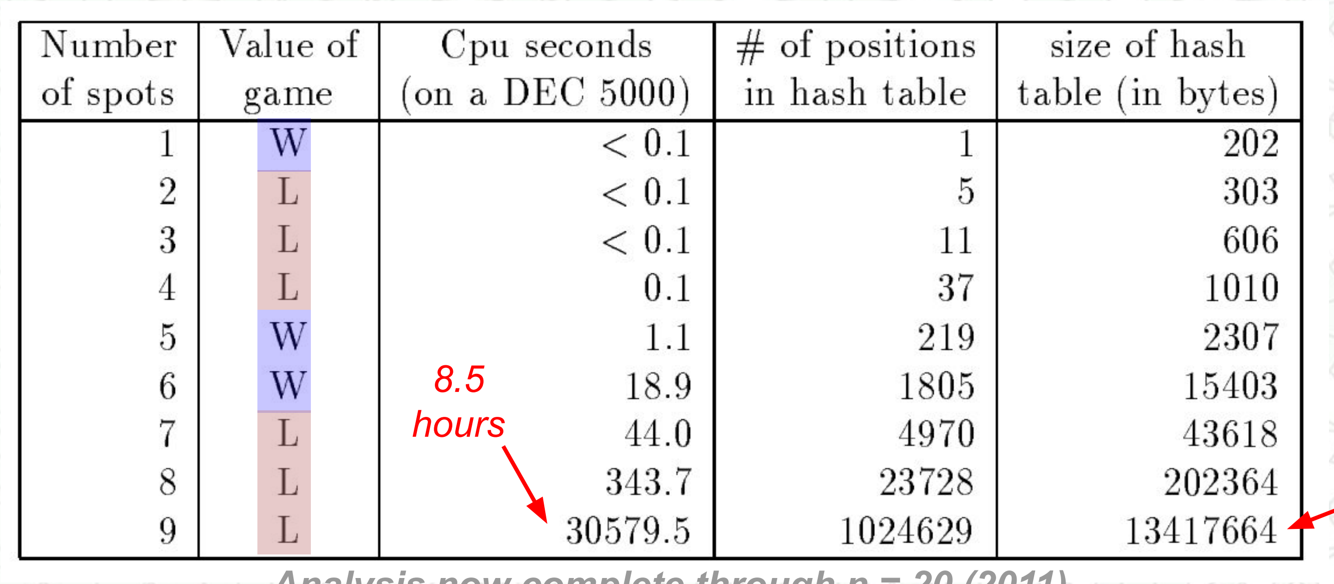
“Computer Analysis of Sprouts”, Applegate, Jacobson, & Sleator, 1991

1.5 MB,
max memory
for DEC was
8-480MB (after 1991?)

Analysis now complete through n = 44 (2011)

Conjecture: Player 1 wins if # spots % 6 == 3, 4, or 5

Sprouts Perfect Play Winner - Misère Play



Number of spots	Value of game	Cpu seconds (on a DEC 5000)	# of positions in hash table	size of hash table (in bytes)
1	W	< 0.1	1	202
2	L	< 0.1	5	303
3	L	< 0.1	11	606
4	L	0.1	37	1010
5	W	1.1	219	2307
6	W	18.9	1805	15403
7	L	44.0	4970	43618
8	L	343.7	23728	202364
9	L	30579.5	1024629	13417664

8.5 hours

“Computer Analysis of Sprouts”,
Applegate, Jacobson, & Sleator,
1991

13 MB,
max memory
for DEC was
8-480MB (after 1991?)

Analysis now complete through n = 20 (2011)

Original Conjecture was later disproven

Current Conjecture: Player 1 wins when # spots % 6 == 0, 4, or 5

EXCEPTIONS: Player 1 wins if # spots == 1 and Player 1 loses if # spots == 4

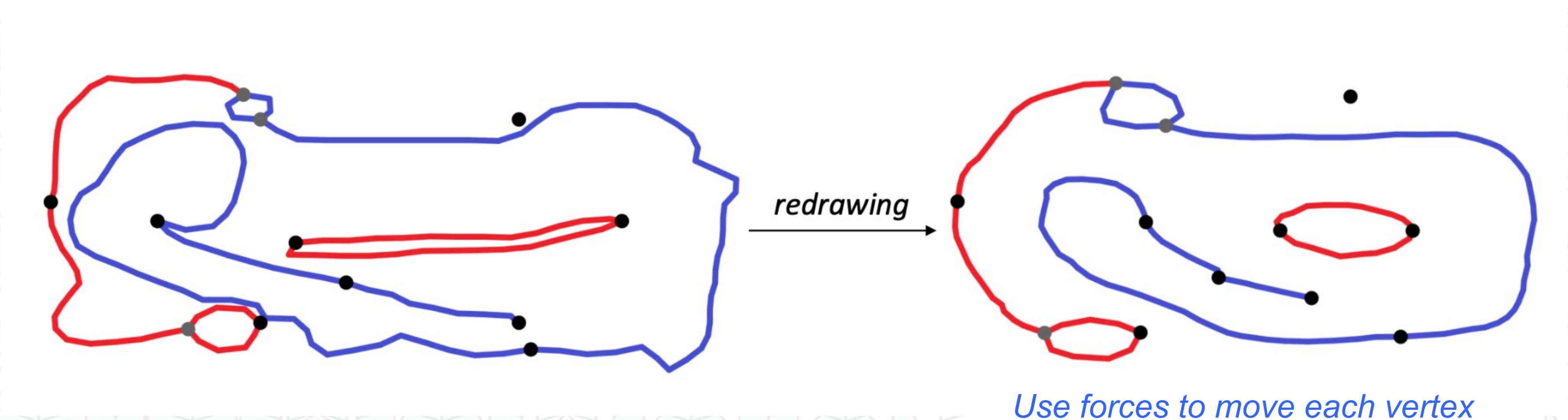
Discussion: Interactive Sprouts Implementation

- I couldn’t find a good interactive, sketch-based digital Sprouts game
  Many versions seem to require Flash, so I don’t know if they were any good
- What is hard about making a computer/phone implementation of Sprouts?

“Implementation of Sprouts: a graph drawing game”
Cizek & Balko, Graph Drawing 2021

Challenge 1:

Free form input drawings degenerate & make positions confusing over time



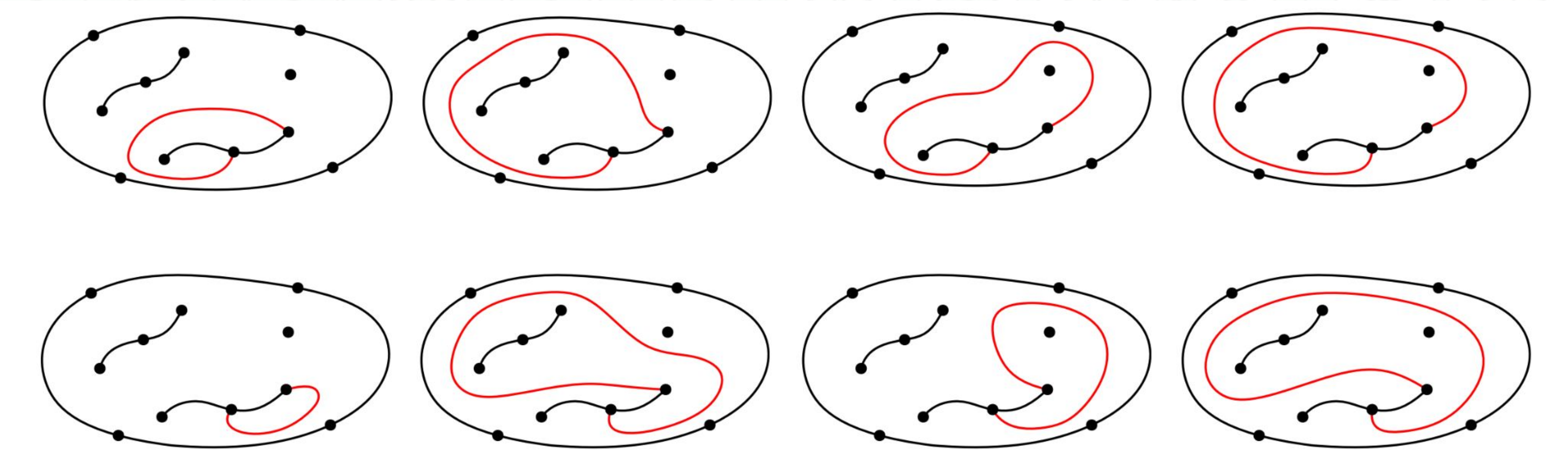
Use forces to move each vertex such that no crossing is introduced

https://algo.cs.uni-tuebingen.de/gd2021/data/uploads/Day%203/Session%203/Sprouts.pdf

“Implementation of Sprouts: a graph drawing game”
Cizek & Balko, Graph Drawing 2021

Challenge 2: Enormous Game Tree

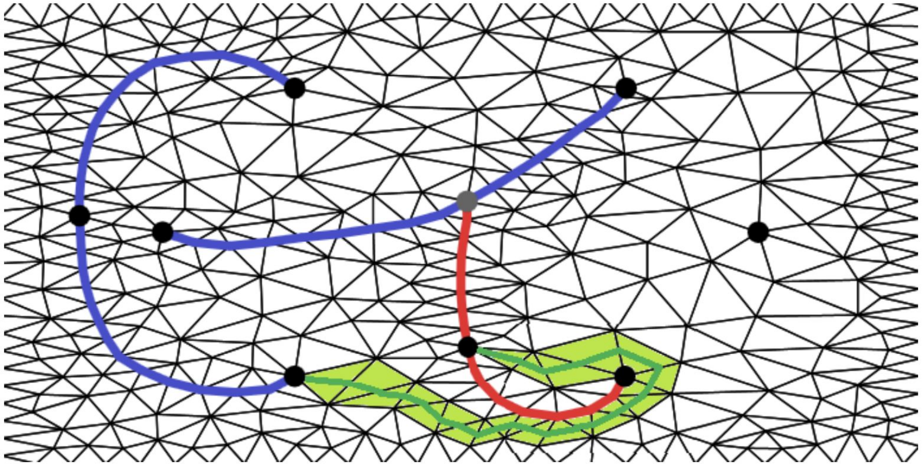
Challenge 3: Drawing a Computer Move



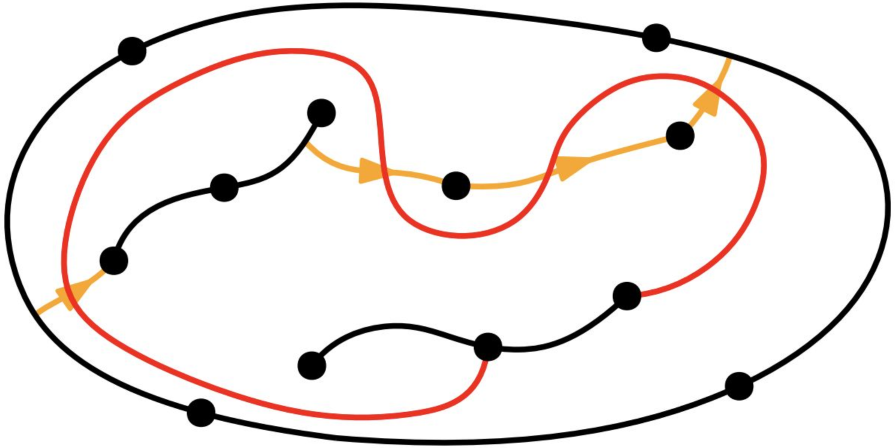
https://algo.cs.uni-tuebingen.de/gd2021/data/uploads/Day%203/Session%203/Sprouts.pdf

“Implementation of Sprouts: a graph drawing game”
Cizek & Balko, Graph Drawing 2021

triangulate surrounding region



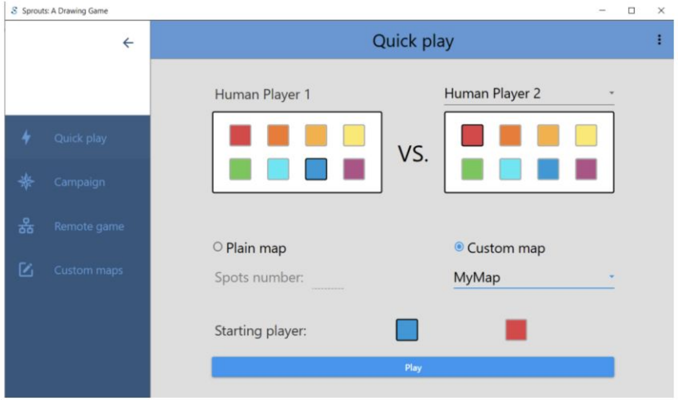
unique intertwining of the spindle



https://algo.cs.uni-tuebingen.de/gd2021/data/uploads/Day%203/Session%203/Sprouts.pdf

“Implementation of Sprouts: a graph drawing game”
Cizek & Balko, Graph Drawing 2021

- Support games up to 20 spots
- Perfect AI on up to 11 spots



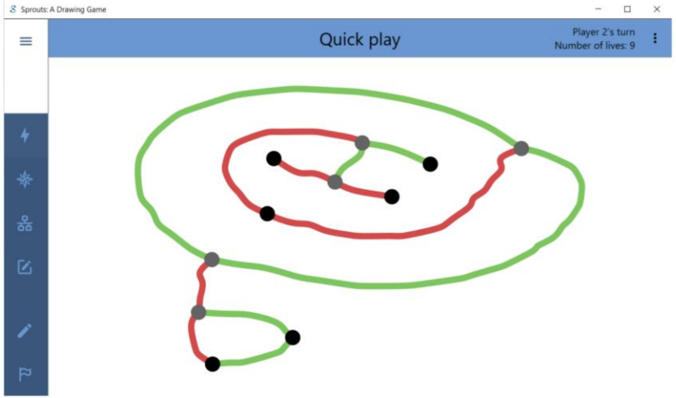
Human Player 1 VS. Human Player 2

○ Plain map ● Custom map

Spots number:

MyMap

Starting player:



Player 2's turn
Number of lives: 9

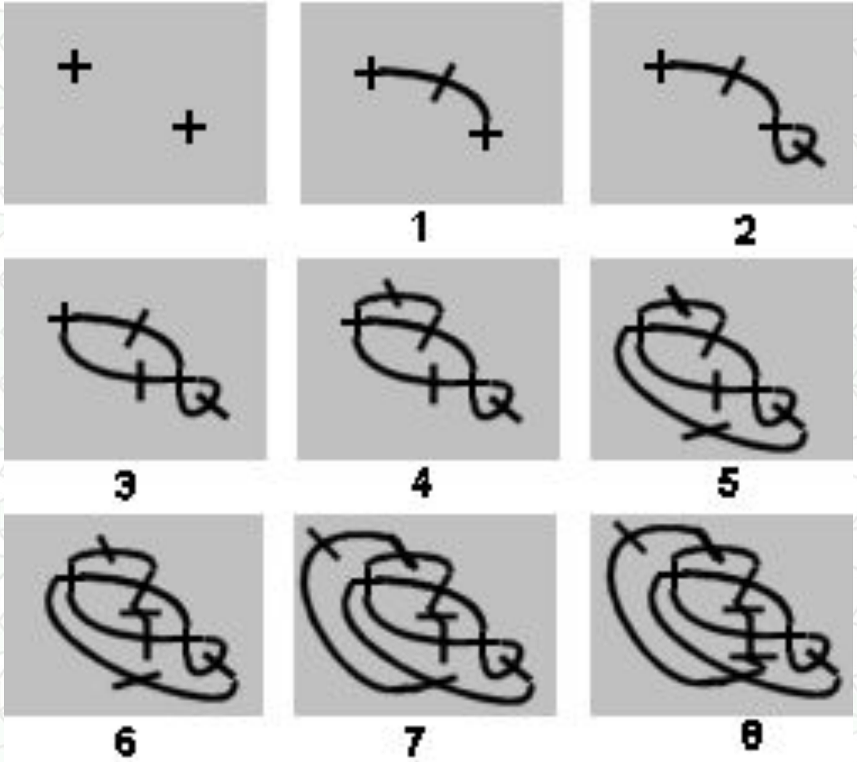
https://kam.mff.cuni.cz/~cizek/Sprouts/win_x64/

Outline for Today

- Final Project Presentation Schedule
- Last Time: Robot Motion Planning, Minkowski Sums, etc.
- Paper & Pencil Game of Sprouts
- Computer Analysis of Sprouts
- Brussel Sprouts
- Hybrid Sprouts & Brussel Sprouts
- Next Time: Final Project Presentations!

Brussel Sprouts Variant Game Rules

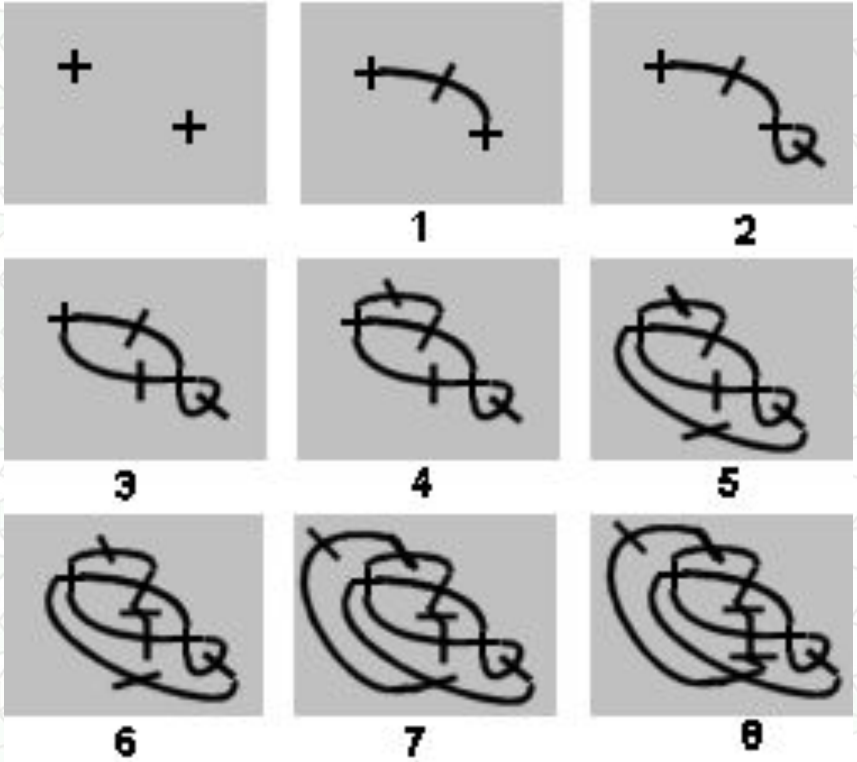
- Draw *n* crosses
- Players take turns:
  - Draw a line joining two cross "legs".
  - Draw a hash mark across the new line (a new "cross" position).
  - No more than four lines can emerge from any cross.



https://en.wikipedia.org/wiki/Sprouts_(game)#/media/File:Brussel_Sprouts_Game.png

Brussel Sprouts Analysis

- We start with *n* crosses and 4**n* live 'leg's
- Each move adds a line, uses up two legs, and adds two legs
- *Does it ever end?*



Brussel Sprouts Analysis

- Yes, in fact it does end!
  - Every face contains at least one live leg
  - We are done when no face has more than one live leg
- It must follow Euler’s Characteristic for Planar Graphs:
  - Let m = # of moves
  - # of edges = e = 2m
  - # of vertices = v = n + m (we start with n, and we add one each move)
  - # of faces = 4n, there is exactly 1 free end inside of each face at the end
  - $2 = f - e + v$
  - $2 = 4n -2m + n + m$
  - # of moves = m = 5n - 2

Outline for Today

- Final Project Presentation Schedule
- Last Time: Robot Motion Planning, Minkowski Sums, etc.
- Paper & Pencil Game of Sprouts
- Computer Analysis of Sprouts
- Brussel Sprouts
- Hybrid Sprouts & Brussel Sprouts
- Next Time: Final Project Presentations!

Hybrid Sprouts & Brussel Sprouts Game Rules

- Start with a mix of dots and/or crosses (which *player's choice?*)
- Draw a line connecting a dot or cross to itself or another dot or cross (*player choice*) and add a dot or a cross to the line (*player choice*)

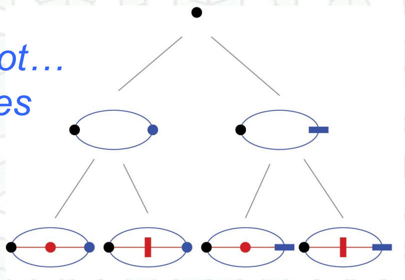
Hybrid Sprouts & Brussel Sprouts Analysis

For n = 1

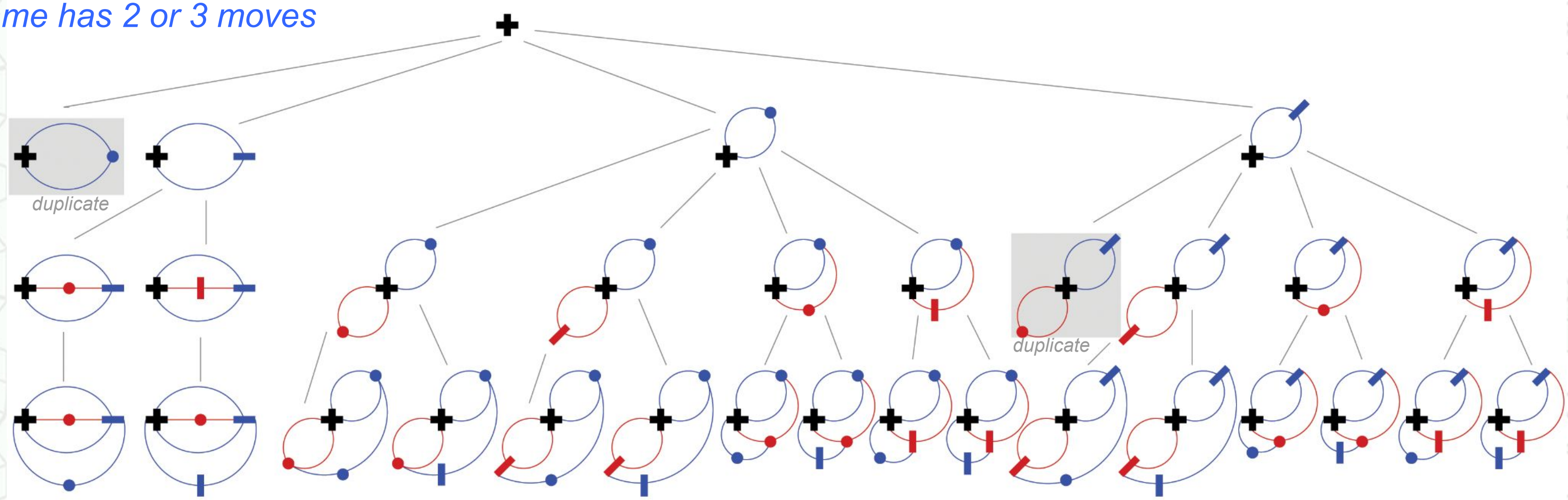
Hybrid Sprouts & Brussel Sprouts Analysis

For n = 1

If we start with one dot…
the game has 2 moves



If we start with one cross…
the game has 2 or 3 moves



Hybrid Sprouts & Brussel Sprouts Analysis

For n = 2?

Hybrid Sprouts & Brussel Sprouts Analysis

For n = 2?

Actually… for n > 1: Analysis is not completed! (according to Wikipedia)

Outline for Today

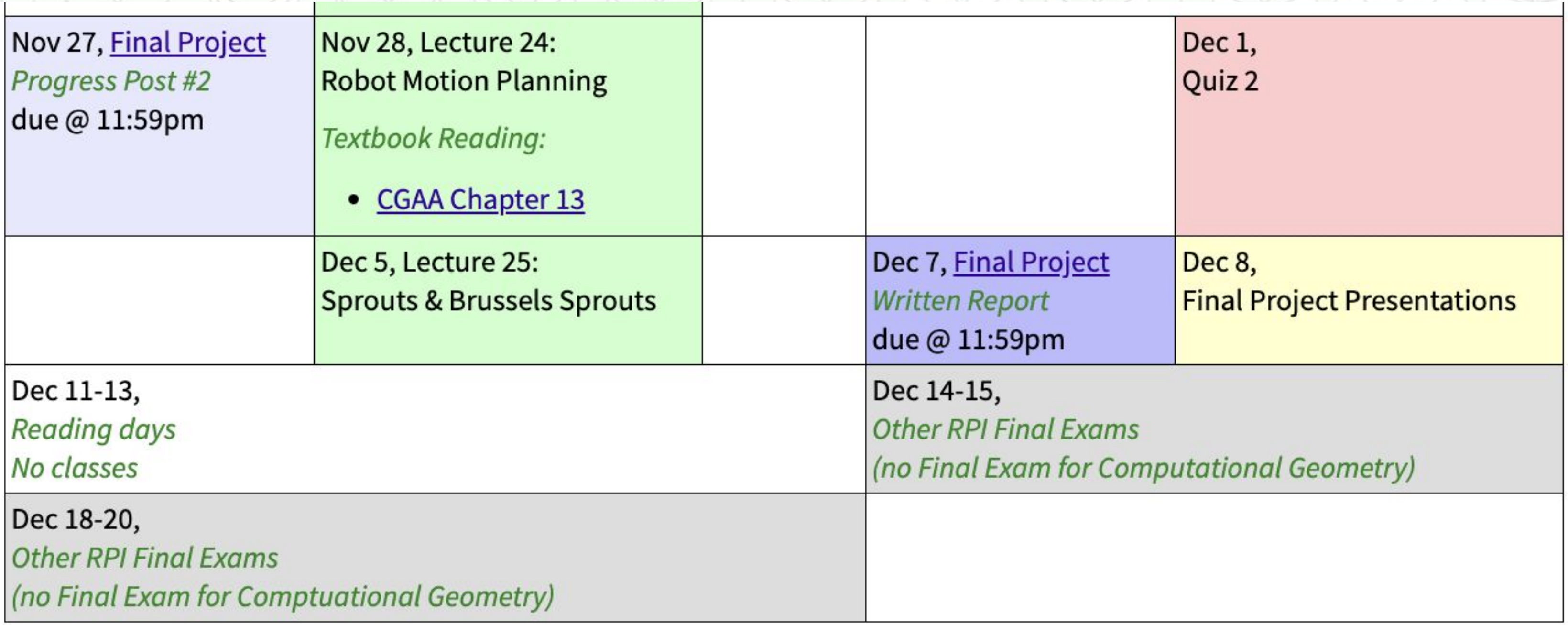
- Final Project Presentation Schedule
- Last Time: Robot Motion Planning, Minkowski Sums, etc.
- Paper & Pencil Game of Sprouts
- Computer Analysis of Sprouts
- Brussel Sprouts
- Hybrid Sprouts & Brussel Sprouts
- Next Time: Final Project Presentations!

Final Presentation

- Summarize prior work as necessary
  Can assume peers know algorithms/structures/papers from lecture
- Be technical:
  What were the challenges?
  How did you solve them?
- Live demo / video / lots of images (depends on project)
  Use plenty of examples (both of success & failure)
- Teams of 2: Both should present & make it clear who did what
- Use your time wisely! Practice! & Time yourself!
  I will stop you mid-sentence if you run over

Well-written Research Paper / Report

- Motivation / Context / Related Work
- Accomplishments / Contributions of this work
- Clear description of algorithm
    - Sufficiently-detailed to allow work to be reproduced
    - Work is theoretically sound (hacks/arbitrary constants discouraged, but must be documented)
- Results
    - Well chosen examples
    - Clear tables/illustrations/visualizations with descriptive captions!
- Conclusions & Potential Future Work
    - Limitations of the method are clearly stated



Nov 27, Final Project *Progress Post #2* due @ 11:59pm	Nov 28, Lecture 24: Robot Motion Planning *Textbook Reading:* • CGAA Chapter 13			Dec 1, Quiz 2
	Dec 5, Lecture 25: Sprouts & Brussels Sprouts		Dec 7, Final Project *Written Report* due @ 11:59pm	Dec 8, Final Project Presentations
Dec 11-13, *Reading days* *No classes*			Dec 14-15, *Other RPI Final Exams* *(no Final Exam for Computational Geometry)*	
Dec 18-20, *Other RPI Final Exams* *(no Final Exam for Comptuational Geometry)*				