A fun inductive problem:

Gen prob: comparing graphs embedded in $\mathbb{R}^2$

+ here comes a new idea: length-preserving (Fréchet) dist.

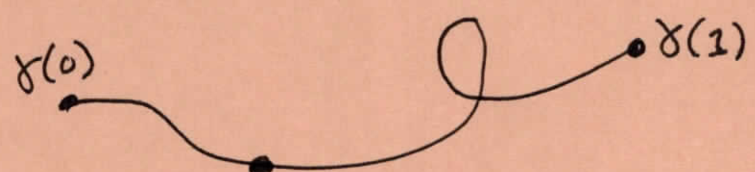
Given two metric spaces (X, d_X), (Y, d_Y), we say that a continuous fn

$$f : (X, d_X) \longrightarrow (Y, d_Y) \quad \text{is}$$

"length-preserving" iff $\forall x, y \in X$

$$d_X(x, y) = d_Y(f(x), f(y))$$

e.g., $\gamma : I \longrightarrow \mathbb{R}^2$ be a curve parametrized by arclength.



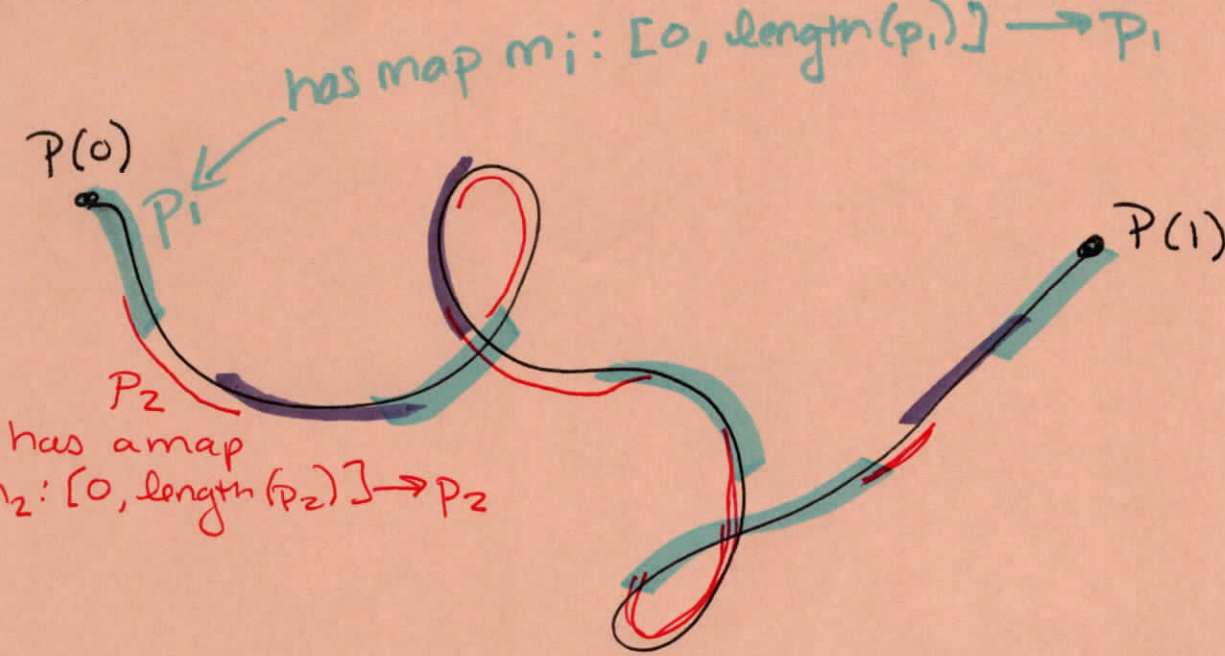
$\gamma(t) \leftarrow \gamma$ has traveled distance t.

Claim: Let $P : I \longrightarrow (X, d_X)$ be a path in (X, d_X) such that $\exists$ a finite cover of P $\{p_i \subset P\}$ where $\exists$ maps $m_i : [0, a_i] \longrightarrow p_i$ that are length-preserving bijections.

Then, $\exists$ a value a + a reparametrization $\alpha : I \to [0, a]$ such that $P \circ \alpha$ is length-preserving.

Picture:

$P: I \to X$ is some curve in X



Challenge: use induction to prove this

Backtracking

→ using recursion to try out every possibility, then send how you got there back up!

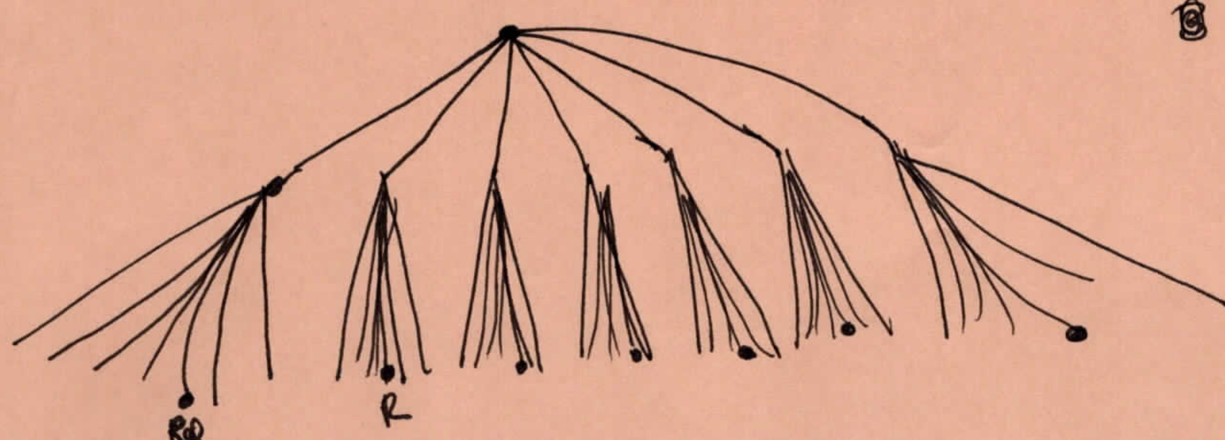
→ today, we'll be drawing lots of trees.

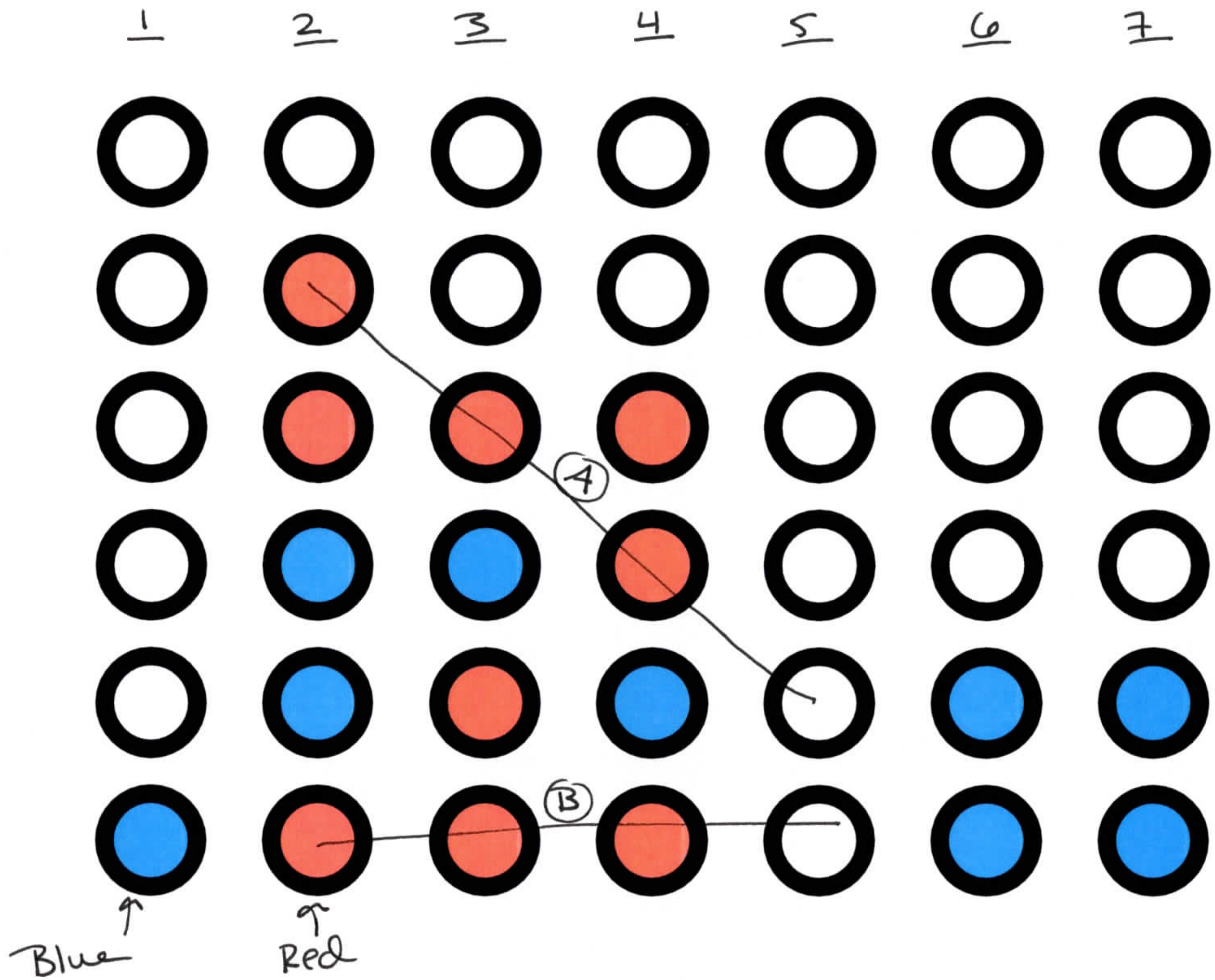
EX: connect 4

- If its the red players turn, they would likely place their chip in slot 5. + win!
- If its the blue players turn, they are a bit doomed if red wants to win.
  - → place it in slot 5 + the red will place in slot 5 to win. A
  - → place it in slot ~5 + then red will place it in slot 5 + win B

BLUE:

RED:




1
2
3
4
5
6
7
A
B
Blue
Red

5

Unless you've seen this game before[3], you probably don't have any idea how to play it well. Nevertheless, there is a relatively simple backtracking algorithm that can play this game—or any two-player game without randomness or hidden information that ends after a finite number of moves—*perfectly*. That is, if we drop you into the middle of a game, and it is *possible* to win against another perfect player, the algorithm will tell you how to win.

A ***state*** of the game consists of the locations of all the pieces and the identity of the current player. These states can be connected into a *game tree*, which has an edge from state x to state y if and only if the current player in state x can legally move to state y. The root of the game tree is the initial position of the game, and every path from the root to a leaf is a complete game.

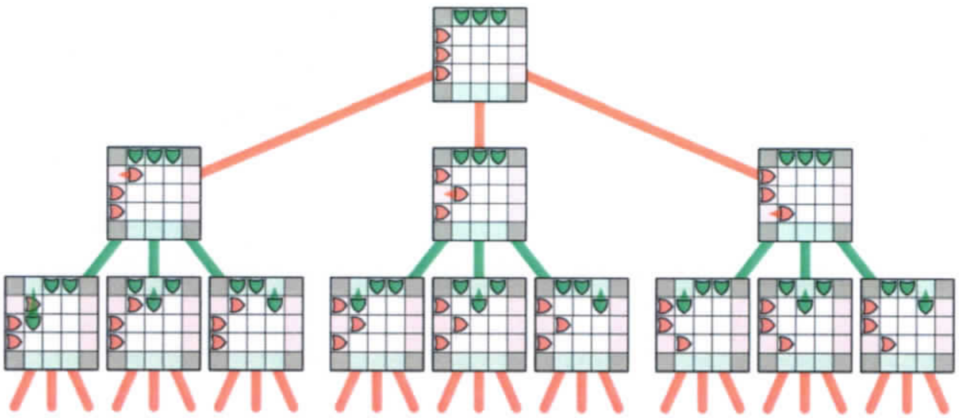


Figure 2.5. The first two levels of the fake-sugar-packet game tree.

To navigate through this game tree, we recursively define a game state to be ***good*** or ***bad*** as follows:

- A game state is *good* if either the current player has already won, or if the current player can move to a bad state for the opposing player.
- A game state is *bad* if either the current player has already lost, or if every available move leads to a good state for the opposing player.

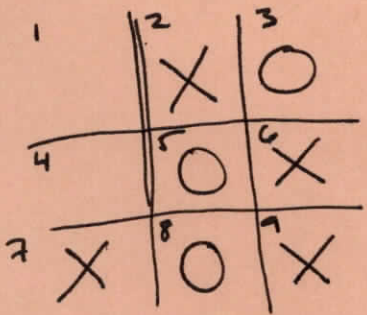
Equivalently, a non-leaf node in the game tree is good if it has at least one bad child, and a non-leaf node is bad if all its children are good. By induction, any player that finds the game in a good state on their turn can win the game, even if their opponent plays perfectly; on the other hand, starting from a bad state, a player can win only if their opponent makes a mistake. This recursive definition was proposed by Ernst Zermelo in 1913.[4]

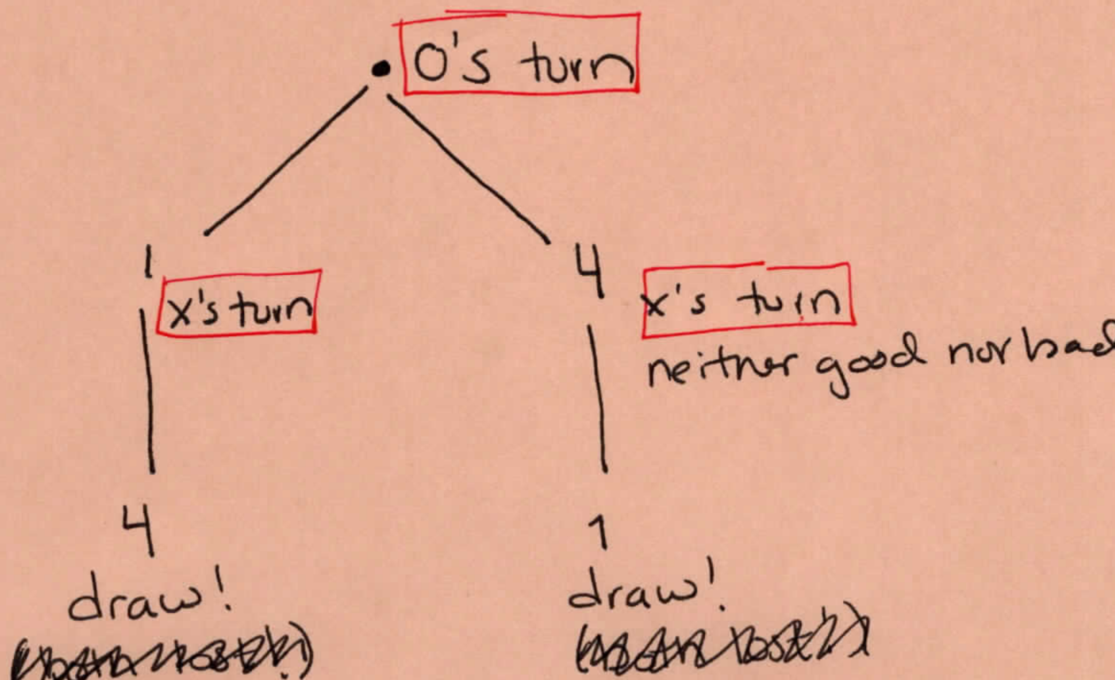
[3]If you have, please tell me where!

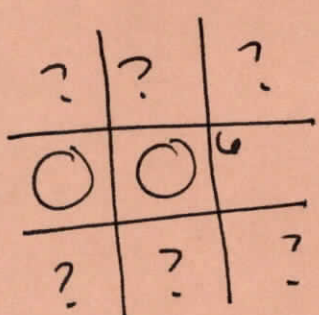
[4]In fact, Zermelo considered the more subtle class of games that have a finite number of states, but that allow infinite sequences of moves. (Zermelo defined infinite play to be a draw.)

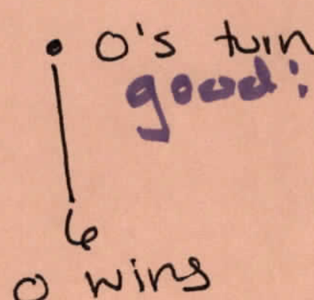
TIC-TAC-TOE

- X's go first









In groups, play 3 moves for X + 3 for O then draw the game tree (list all possibilities!) + mark if each node in the game tree is good or bad (or neither)

→ Leaf nodes can be considered O's turn.

cont:

1. If a draw means X wins, what the good/bad states of your game tree?
2. If a draw means O wins, what are the good / bad states?

Notes:

→ when encountering new def'n / algo, try it out on small examples + play w/ it to get a feel for it.

→ drawing trees - choose labels wisely
in general, what labeling conventions can you establish to show what you are doing clearly