28 Sept. 2022

personal email: brittany.fasy@gmail.com

Reminders: HW due tonight
Proj, part 1 Friday
next HW next Friday
Exam 1, Wed, Oct 12

Backtracking

→ Given a problem where a sequence of decisions must be made.

→ We try every sequence (to evaluate which choice is the best)

---

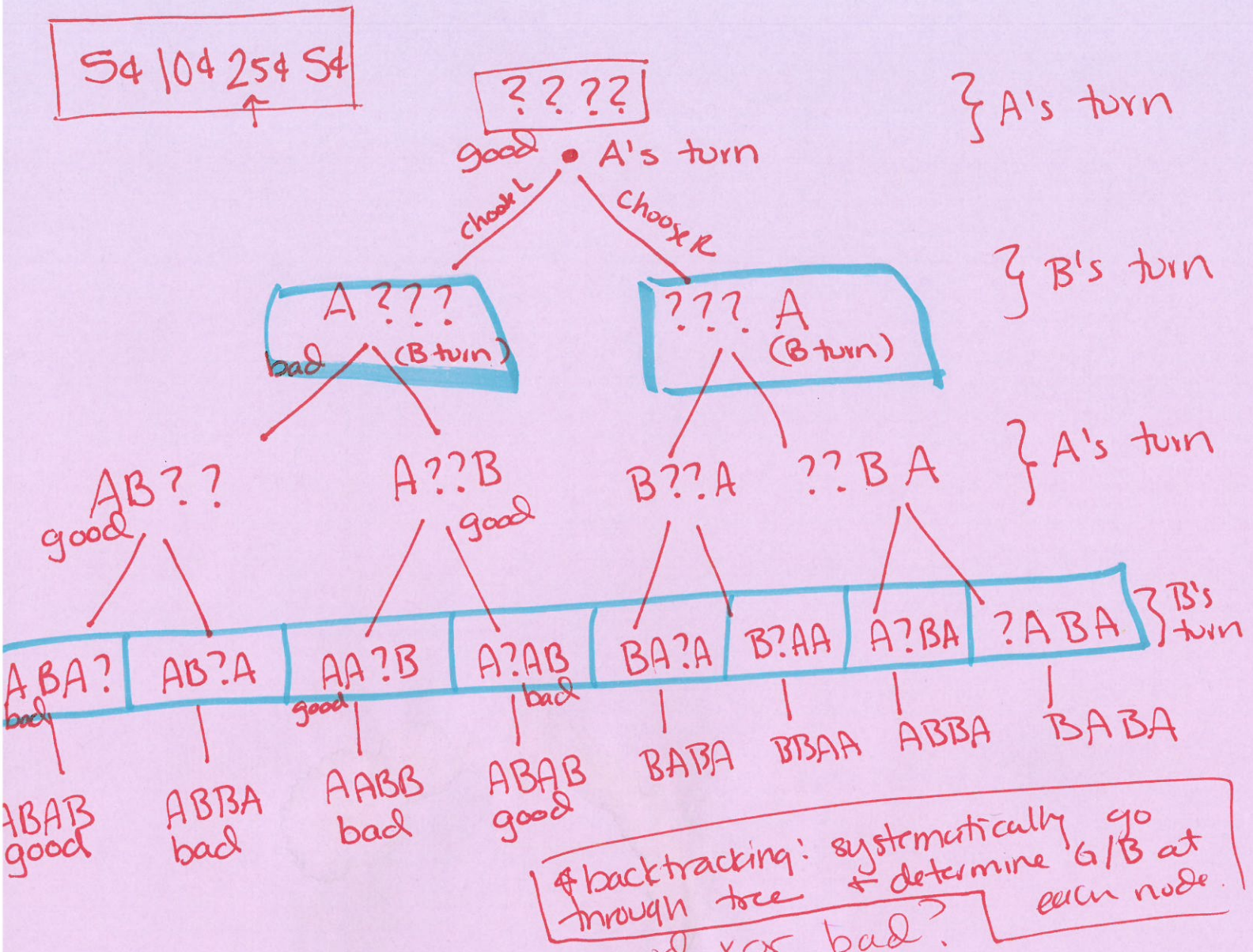
example: 2n coin game

→ Pick "best" move at each point in the game (alt. Choosing btwn A & B)

→ For any game, define:

(1) good state: cur player has won (ie, game over!)
-OR-
a move that forces the other player into a bad state.

(2) bad state: cur player has lost (game over!)
-OR-
~~~~ move leads to a good state for the opposing player.
~~~~



Q: Is every state either good xor bad?

A: Depends on the game!

- If ∃ "tie" leaf node, then that node is neither good nor bad (& potentially ancestors of it.)
- If all leaves result in a winner, then can prove by induction that each node is either good xor bad.

of for a 2 player game.

current state of the game
the current player
opponent is "¬player"

```
PLAYANYGAME(X, player):
  if player has already won in state X
      return GOOD
  if player has already lost in state X
      return BAD
  for all legal moves X ⇝ Y
      if PLAYANYGAME(Y, ¬player) = BAD
          return GOOD        ⟨⟨X ⇝ Y is a good move⟩⟩
  return BAD                 ⟨⟨There are no good moves⟩⟩
```

X = cur state, input
Y = a potential next state

X⇝Y
"the move"

output: GOOD/BAD, for palyer in state X

If we wind up here, then we know
we haven't returned Good in for loop
=> all moves result in "good" state for ¬player.

Subset sum:

Given: a set of values $X \subseteq \mathbb{R}$

$T \in \mathbb{R}$ "total"

Does there exist $X' \subseteq X$ such that $\sum_{x \in X'} x = T$?

Note: If I can enumerate all subsets, then I can figure this out: Check if each one sums to T.

```
Subset Sum (X, T)
    if T = 0
        return TRUE
    if T < 0 OR X = ∅
        return FALSE

    x ← any elt of X
    removing ← Subset Sum (X \ {x}, T)
    keeping ← Subset Sum (X \ {x}, T-x)
    return removing V keeping
                    ↑ "OR"
```

Runtime: $T(n) = 2T(n-1) + \Theta(1) = \Theta(2^n)$