



UC Berkeley EECS
Lecturer SOE
Dan Garcia

CS10 The Beauty and Joy of Computing

Lecture #25 : Tree Recursion

2010-11-29

MS KINECT = BODY I/O

The newly released (and much-hyped) Microsoft Kinect system for the XBOX 360 used controller-free body motions to control games, music, and movies.



xbox.com/kinect/

Review: What's in a Strong Solution

- For every position

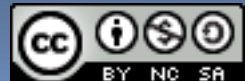
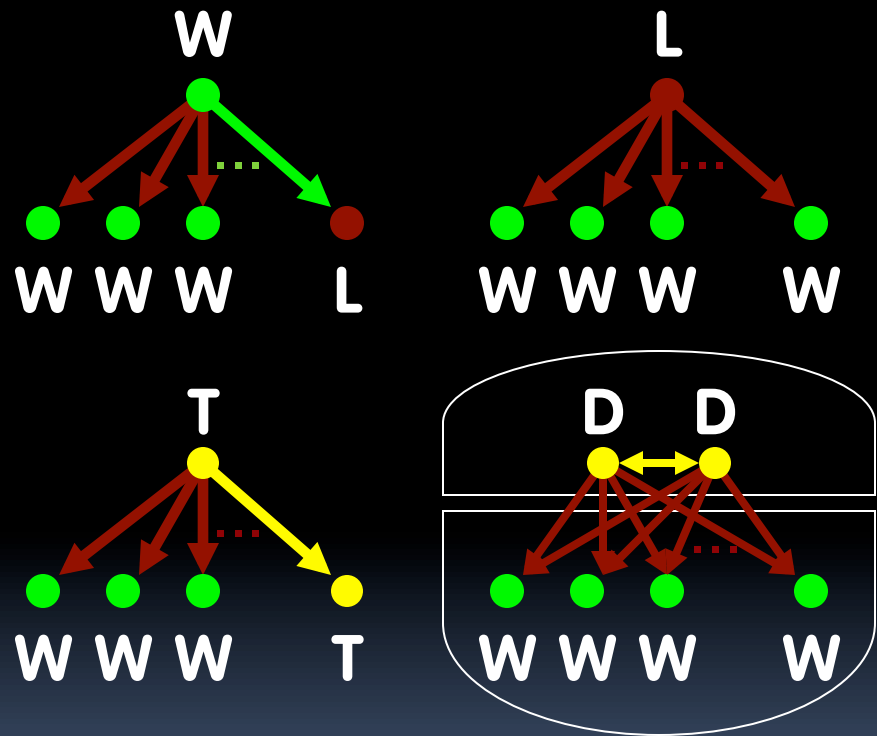
-
-

(for player whose turn it is)

- Winning (\exists losing child)
- Losing (All children winning)
- Tieing ($\neg \exists$ losing child, but \exists tieing child)
- Drawing (can't force a win or be forced to lose)

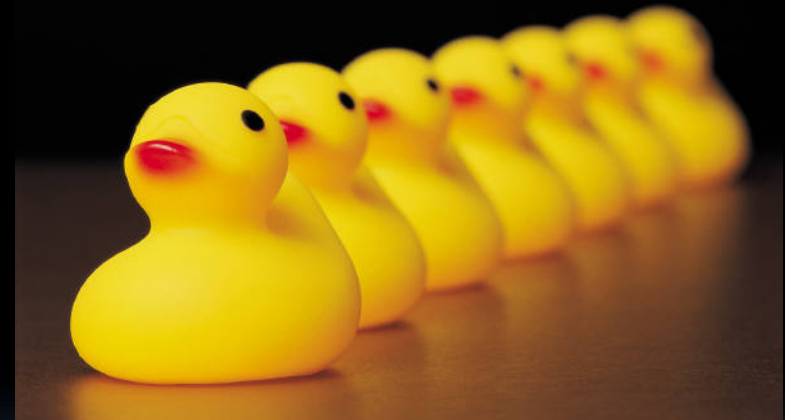
- Remoteness

- How long before game ends?



Review : Example: 1,2,...,10

- **Rules (on your turn):**
 - Running total = 0
- **Rules (on your turn):**
 - Add 1 or 2 to running total
- **Goal**
 - Be the FIRST to get to 10
- **Example**
 - Ana: "2 to make it 2"
 - Bob: "1 to make it 3"
 - Ana: "2 to make it 5"
 - Bob: "2 to make it 7" → photo
 - Ana: "1 to make it 8"
 - Bob: "2 to make it 10" I WIN!



7 ducks (out of 10)



Let's write code to determine value!

- 0 = Win
 - 1 = Lose
 - 2 = Win
 - 3 = Win
 - 4 = Lose
 - 5 = Win
 - 6 = Win
 - 7 = Lose
 - 8 = Win
 - 9 = Win
 - 10 = Lose
- P = Position
 - M = Move
 - We only need 3 blocks to define a game
 - Do Move M on Position P
 - → a new Position
 - Generate Moves from Position P
 - → list of Moves
 - Primitive Value of Position P
 - → {win, lose, tie, undecided}



Answer

```
Value P
if not Primitive Value P = CONSTANT Undecided
  report Primitive Value P
else
  script variables children child values <>
  set children to # map Do Move on Position P over
    Generate Moves from Position P <>
  set child values to # map Value over children <>
  if child values contains CONSTANT Lose
    report CONSTANT Win
  else
    if child values contains CONSTANT Tie
      report CONSTANT Tie
    else
      report CONSTANT Lose
```

