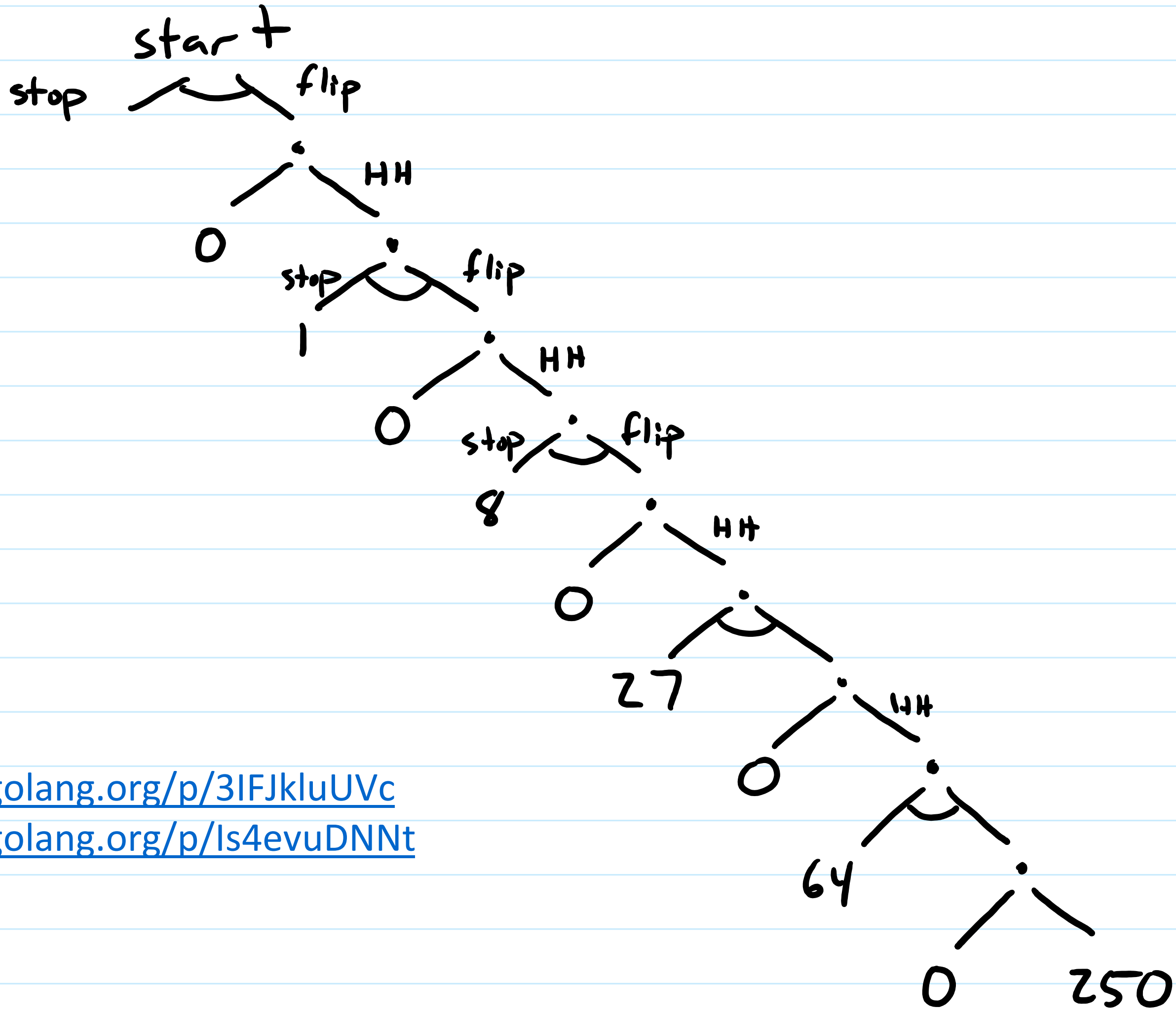


https://play.golang.org/p/3IFJkluUVc
https://play.golang.org/p/Is4evuDNNt

Analysis of 1-player Finite Probabilistic Games

E[pos] = expected winnings having reached position pos

For final positions pos, E[pos]

For non-final choice positions

E[pos] =

For non-final random event positions

E[pos] =

```
for every terminal position pos
      E[pos] ← payoff(pos)

for every nonterminal position pos
      if pos is a choice position
            max ← -∞
            argmax ← NIL
            for every choice c
                  e ← E[next(pos, c)]
                  if e > max
                        e ← max
                        argmax ← c
            E[pos] ← e

      else
            e ← 0.0
            for every outcome σ
                  e ← e + P(σ) · E[next(pos, σ)]
            E[pos] ← e
```

$E[pos] \leftarrow e$

Coins: Start with n coins.
On each turn, flip as many of your remaining coins as you wish.
If $\#T \geq \#H$, lose all the T
Else earn $\#H$ points
Win at X points
Lose if no coins left and $< X$ points

Yahtzee

anchor:

component:

number of anchors:

modification: $E[pos] =$

For nonterminal choice positions

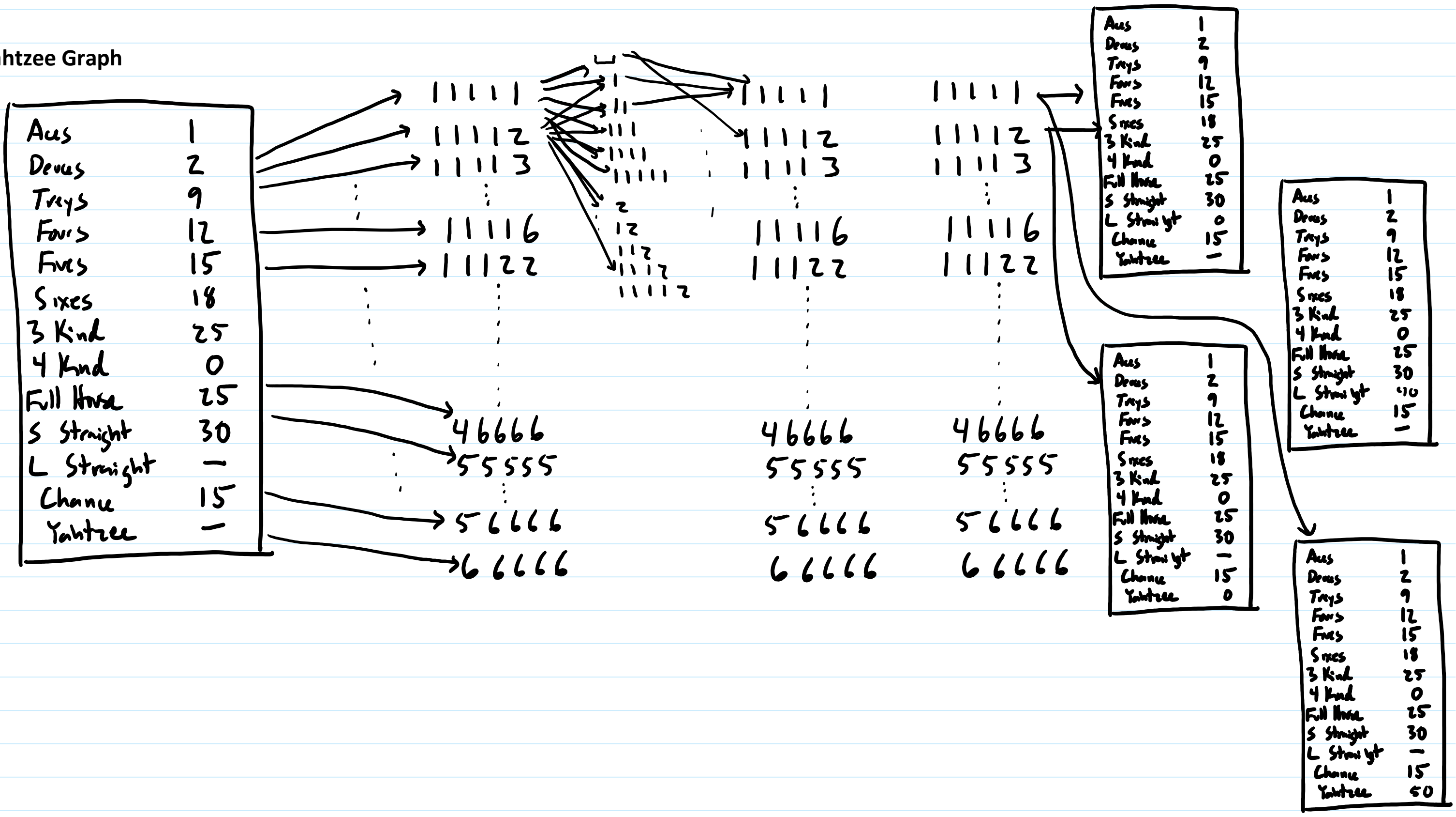
$$E[pos] = \max_{\text{choice } c} E[next(pos, c)]$$

For nonterminal random event positions

$$E[pos] = \sum_{\text{outcome } \sigma} P(\sigma) \cdot E[next(pos, \sigma)]$$

anchors =

Yahtzee Graph



Two-player Zero-sum, probabilistic finite games

For P1 choice position

$$E[pos] = \max \quad E[next(pos, c)]$$

For P2 choice position

$$E[pos] = \max \quad E[next(pos, c)]$$

For nonterminal random event positions

$$E[pos] = \sum_{\substack{\text{outcome} \\ \sigma}} P(\sigma) \cdot E[next(pos, \sigma)]$$

2-player Yahtzee anchors:

2-player Yahtzee variant:

1) get score distribution of optimal solitaire player

2) compute strategy that maximizes the probability of beating the optimal solitaire player

Pig

2-players, turn-based

On each turn
- roll
- if 1, then turn over
- else add number to turn total
- decide: repeat
  - stop (and add turn total to score)

1^{st} to 100 points wins

modify game:

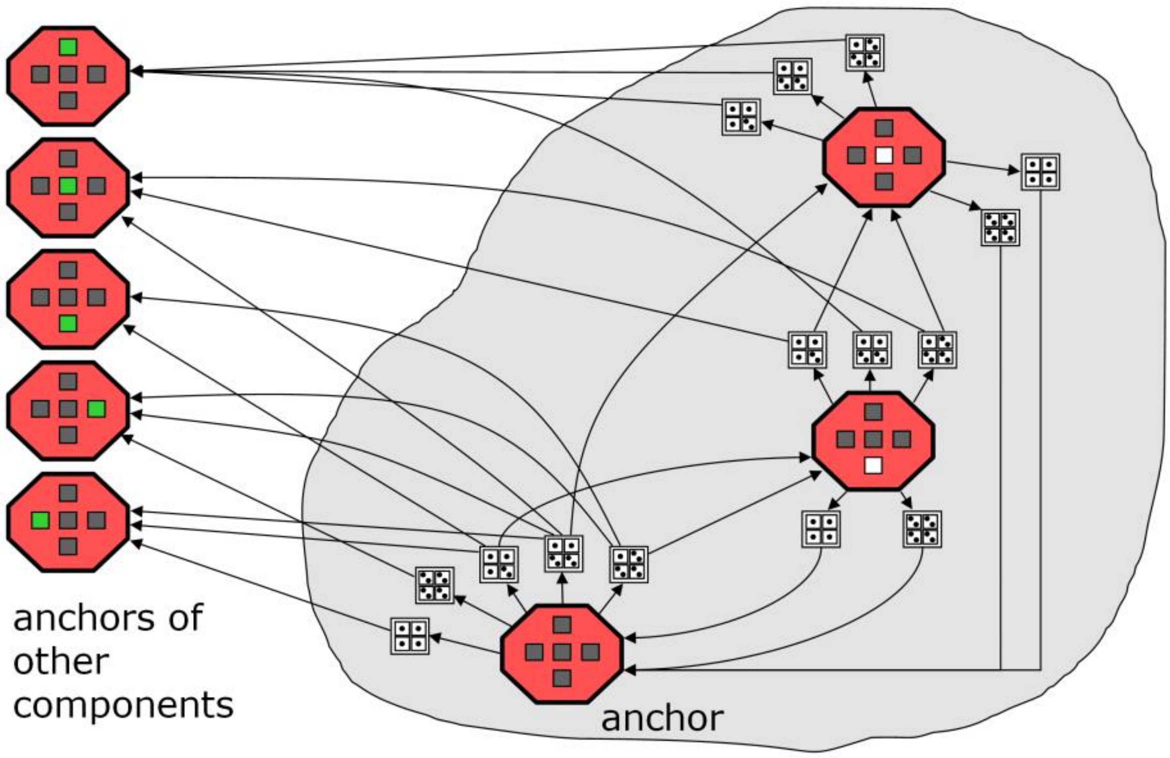
$$\underline{47}-36 \xrightarrow[\text{roll 6}]{} 47+6-36 \xrightarrow[\text{roll again}]{} \cdot \xrightarrow[\text{roll 1}]{} 47-\underline{36} \quad \text{roll 1}$$

<u>cycle</u>

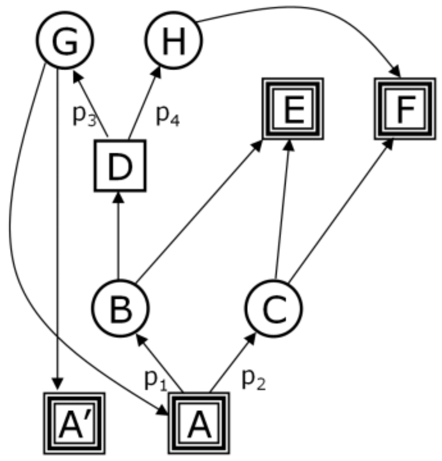
$E[x, y, n]$ = expected # wins for P1 given score is x to y w/ n turns left

=

Can't Stop Graph

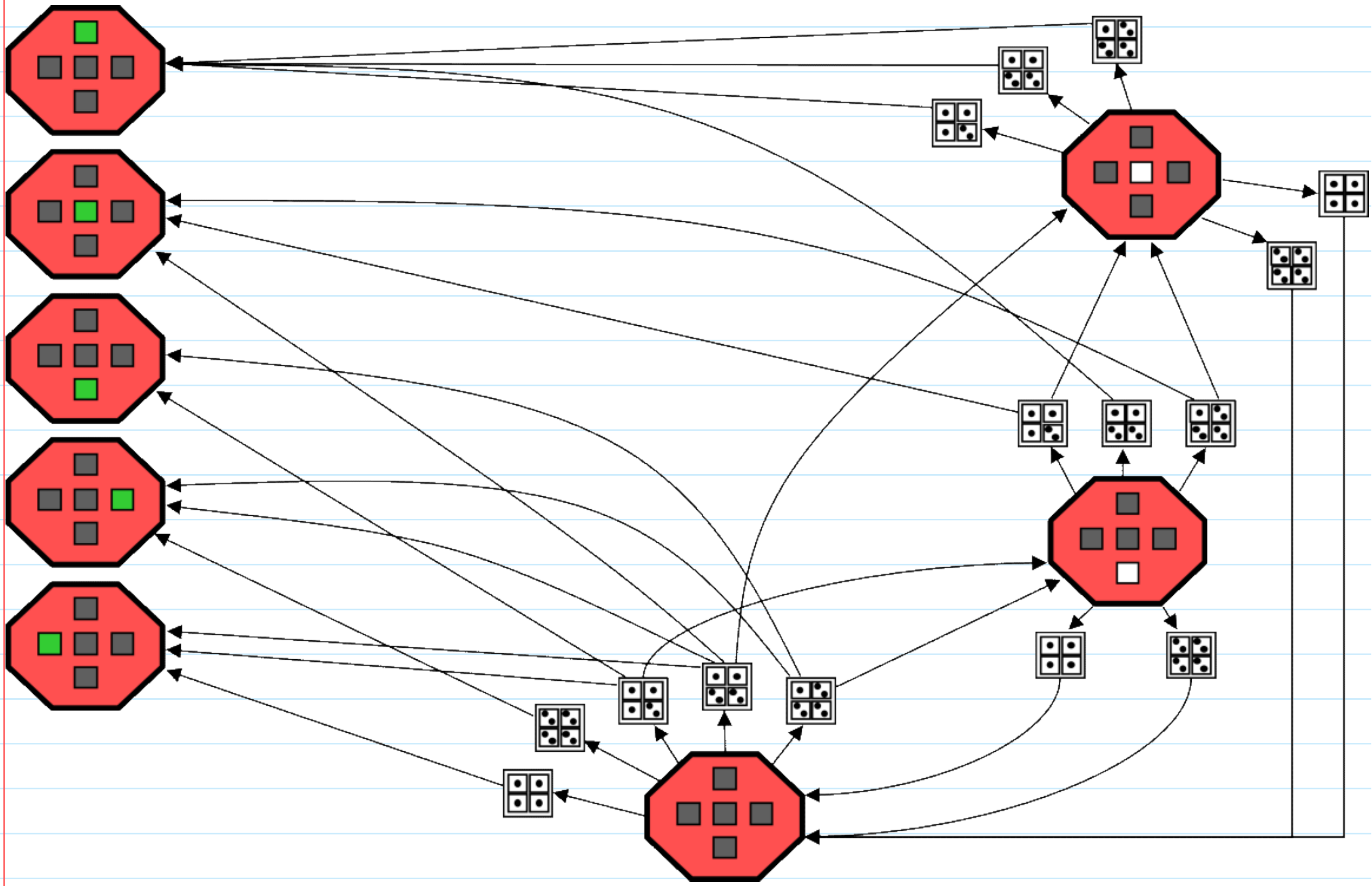


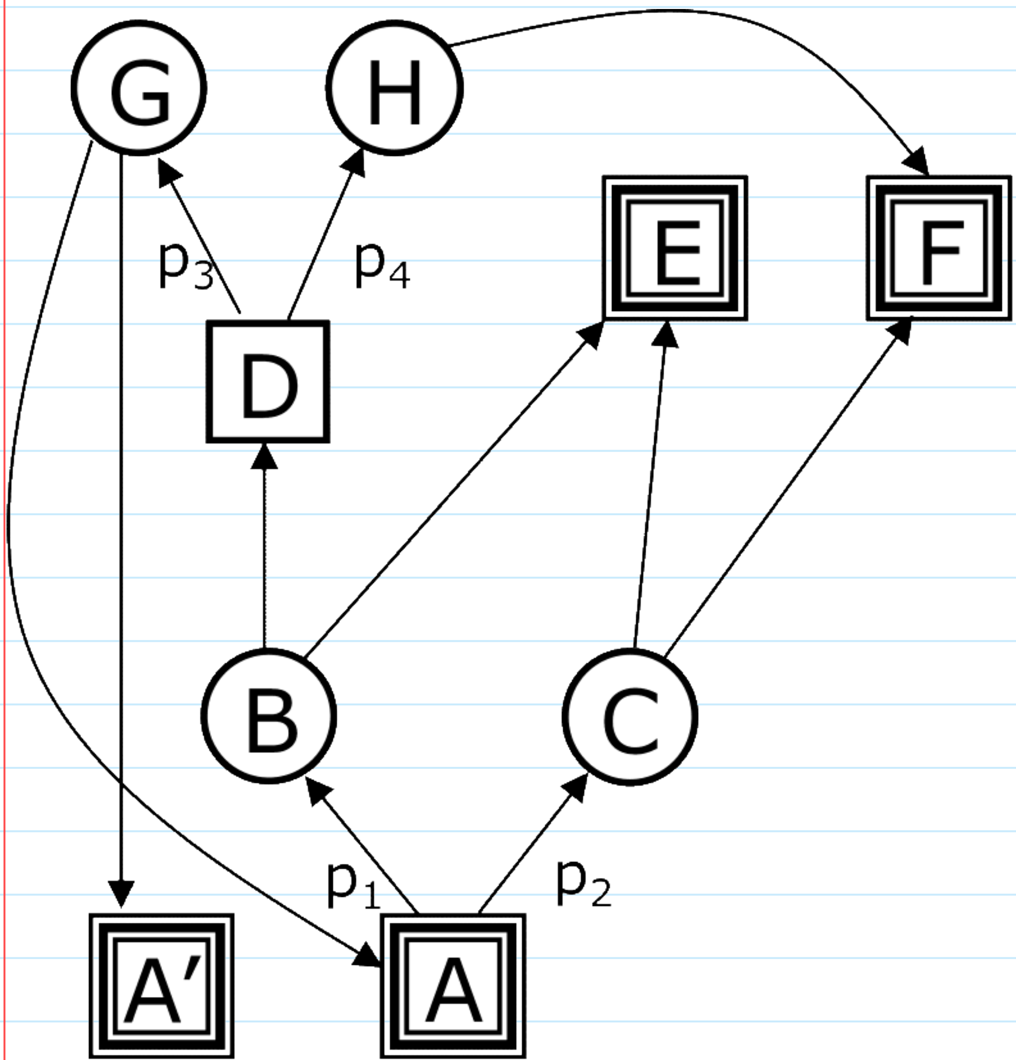
Numerical Analysis



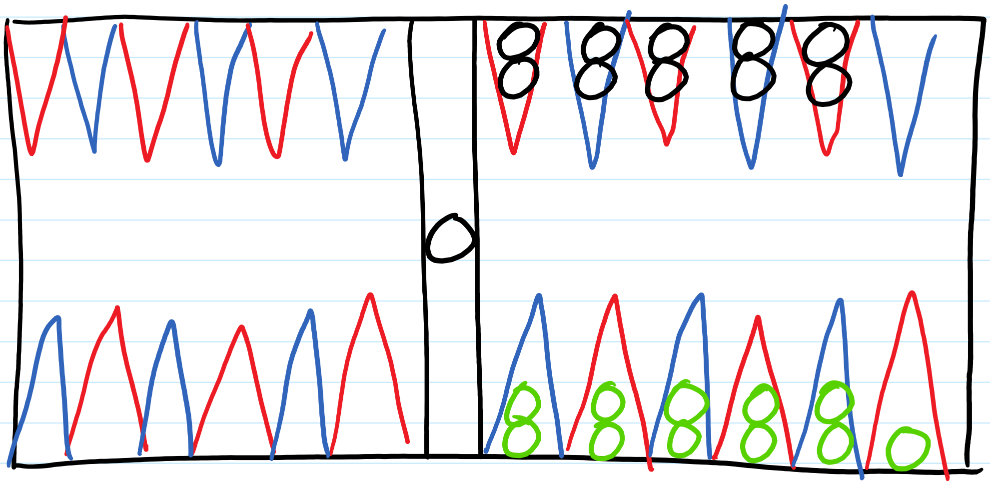
- Make a copy of anchor
- component is now a DAG
- Guess value of f(A')
- f(A) is a function of f(A')
- Want fixed point

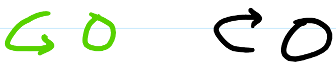
- Function is piecewise linear and continuous
- Fast convergence from Newton's method



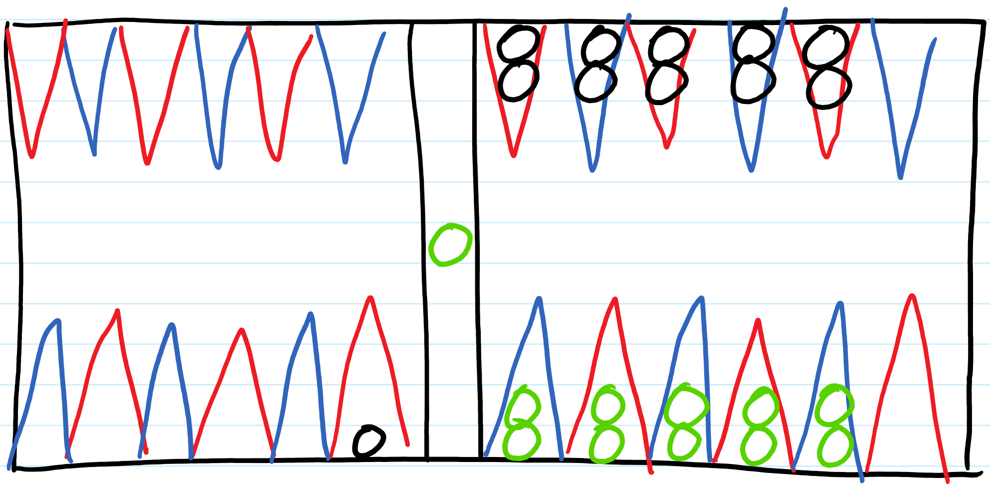

G
H
p_3
p_4
E
F
D
B
C
p_1
p_2
A′
A

Backgammon

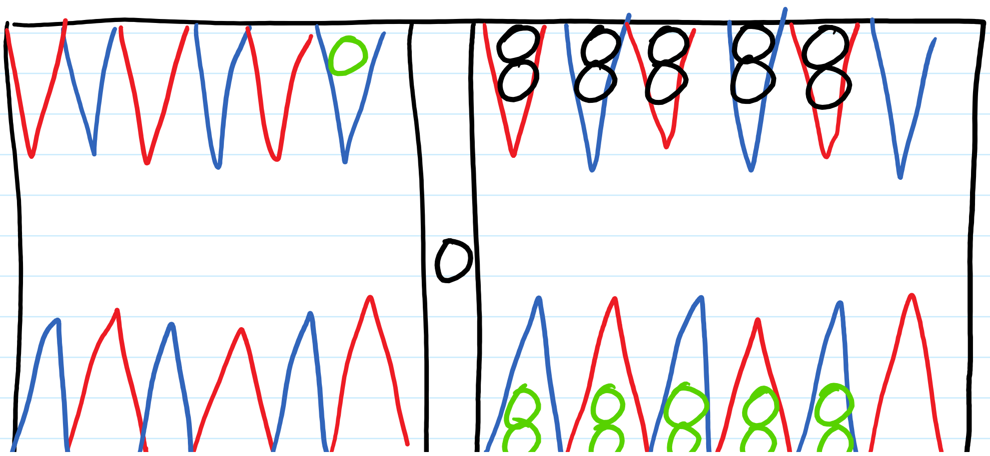


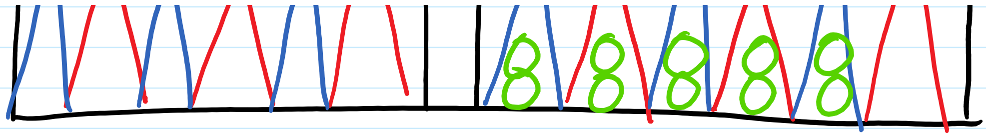


Black rolls 1-6



Green rolls 5-6 4-4 3-6 1-6
Black rolls 1-2 6-3 3-2





Black rolls 6-5 4-3 5-5 2-4 1-6

Green rolls 2-3 1-3 1-2 3-2

3+ Player Pig

$E_1[x,y,z]$ = expected wins for next player when score is $x - y - z$ (next, next next, prev)

$E_{1xyz}[t]$ = expected wins for next when score is x-y-z; subtotal t

$$= \begin{cases} 1.0 & \text{if } x+t \geq T \\ \max\left(E_3[y,z,x+t],\ \frac{1}{6}\left(E_3[y,z,x] + \sum_{r=2}^{6} E_{1xyz}(t+r)\right)\right) \end{cases}$$

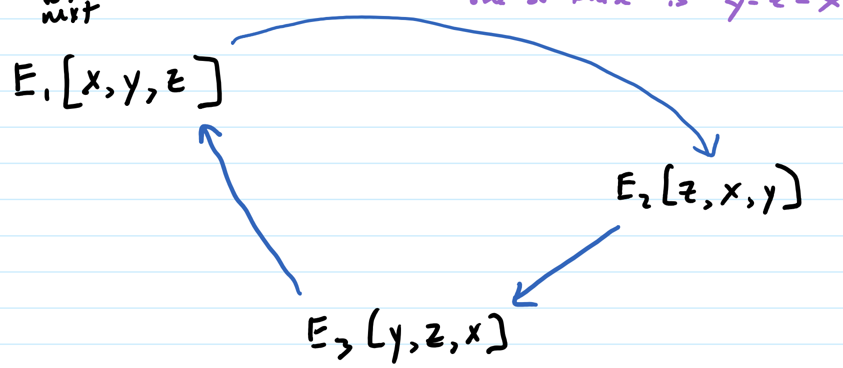
$E_2[x,y,z]$ = expected wins for next next player

$$= \sum_{\substack{\text{outcomes}\\ \text{for next}}} P(\text{outcome}) \cdot E_1[\text{outcome}]$$

$E_3[x,y,z]$ = expected wins for prev player

$$= \sum_{\substack{\text{outcomes}\\ \text{for}\\ \text{next}}} P(\text{outcome}) \cdot E_2[\text{outcome}]$$

one of these is $y - z - x$



note $E_1[x,y,z] + E_2[x,y,z] + E_3[x,y,z] = 1$

$E_1[x,y,z]$ $E_1[y,z,x]$ $E_1[z,x,y]$

$E_2[x,y,z]$ $E_2[y,z,x]$ $E_2[z,x,y]$