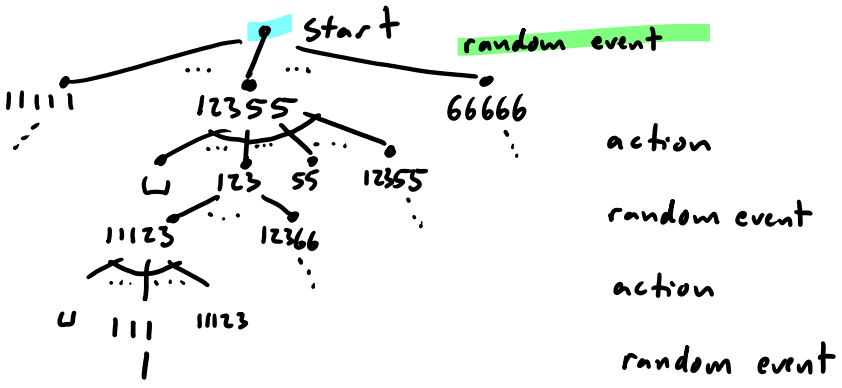
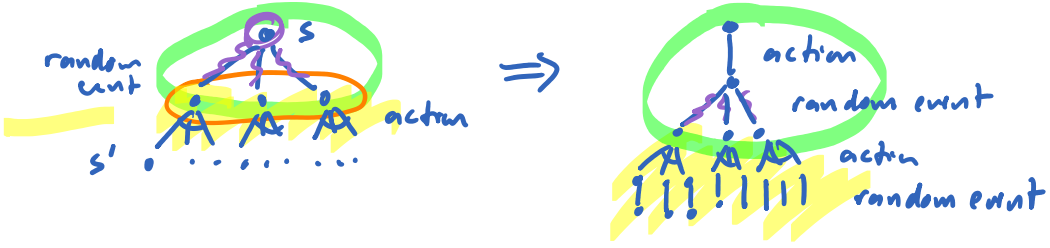
Alternative Views



Traditional Markov Decision Process:

$$v^*(s) = \max_a \sum_{s',r} p(s',r \mid s,a) \cdot [r + \gamma \cdot v^*(s')]$$

Typical Game: random event / action



for random event: $v^*(s) = \sum_{s',r} p(s',r \mid s) \cdot [r + \gamma \cdot v^*(s')]$

for player action: $v^*(s') = \max_a [r + \gamma \cdot v^*(s'')]$

$$V^*(s) = \sum_{s',r} p(s'/i'') \cdot v^*(s') \Rightarrow V^*(s) =$$

$$v^*(s') = \max_{i''} (r + v^*(s''))$$

alternative view

for P_2 : $V_2(s)$ = expected wins for P_2

$$V_2(s, r, t) = \begin{cases} 1.0 & \text{if no } s' \subseteq s \text{ has } sum(s') = r \text{ and } sum(s) < t \\ 0.5 & \text{if no } s' \subseteq s \text{ has } sum(s') = r \text{ and } sum(s) = t \\ 0.0 & \text{if no } s' \subseteq s \text{ has } sum(s') = r \text{ and } sum(s) > t \\ \max_{\substack{s' \subseteq s \\ sum(s') = r}} V_2(s - s', t) & \end{cases}$$

-- move outputs $\operatorname{argmax}_{\substack{s' \subseteq s \\ sum(s') = r}} V_2(s - s', t)$

$$V_2(s, t) = \sum_{r=\min(s)}^{\max(s)} p(\text{roll } r) \cdot V_2(s, r, t)$$

$V_2(\{1,3,5,8,9\}, 9, 1)$

$V_2(\{1,3,5,8,9\}, 9, 2)$ actions always $\{\{1,8\}, \{9\}, \{1,3,5\}\}$

$\vdots$

$V_2(\{1,3,5,8,9\}, 9, 43)$

for P_1 : $V_1(s)$ = expected wins for P_1

$$V_1(s, r) = \begin{cases} 1 - V_2(\{1, \ldots, 9\}, sum(s)) & \text{if no } s' \subseteq s \text{ has } sum(s') = r \\ 1.0 & \text{if } sum(s) = r \\ \max_{\substack{s' \subseteq s \\ sum(s') = r}} V_1(s - s') & \end{cases}$$

-- expect

$$V_1(s) = \sum_{r=\min(s)}^{\max(s)} p(\text{roll } r) \cdot V_1(s, r)$$

Yahtzee

anchor: start of turns

component: start of one turn and the next

number of anchors:

aces...sixes	7^6
3K, 4K	28^2
C	27
FH, SS, LS, Y	3^4
Yahtzee bonus	13

22 trillion anchors

• 1800 states/anchor

≈ 4 quadrillion states

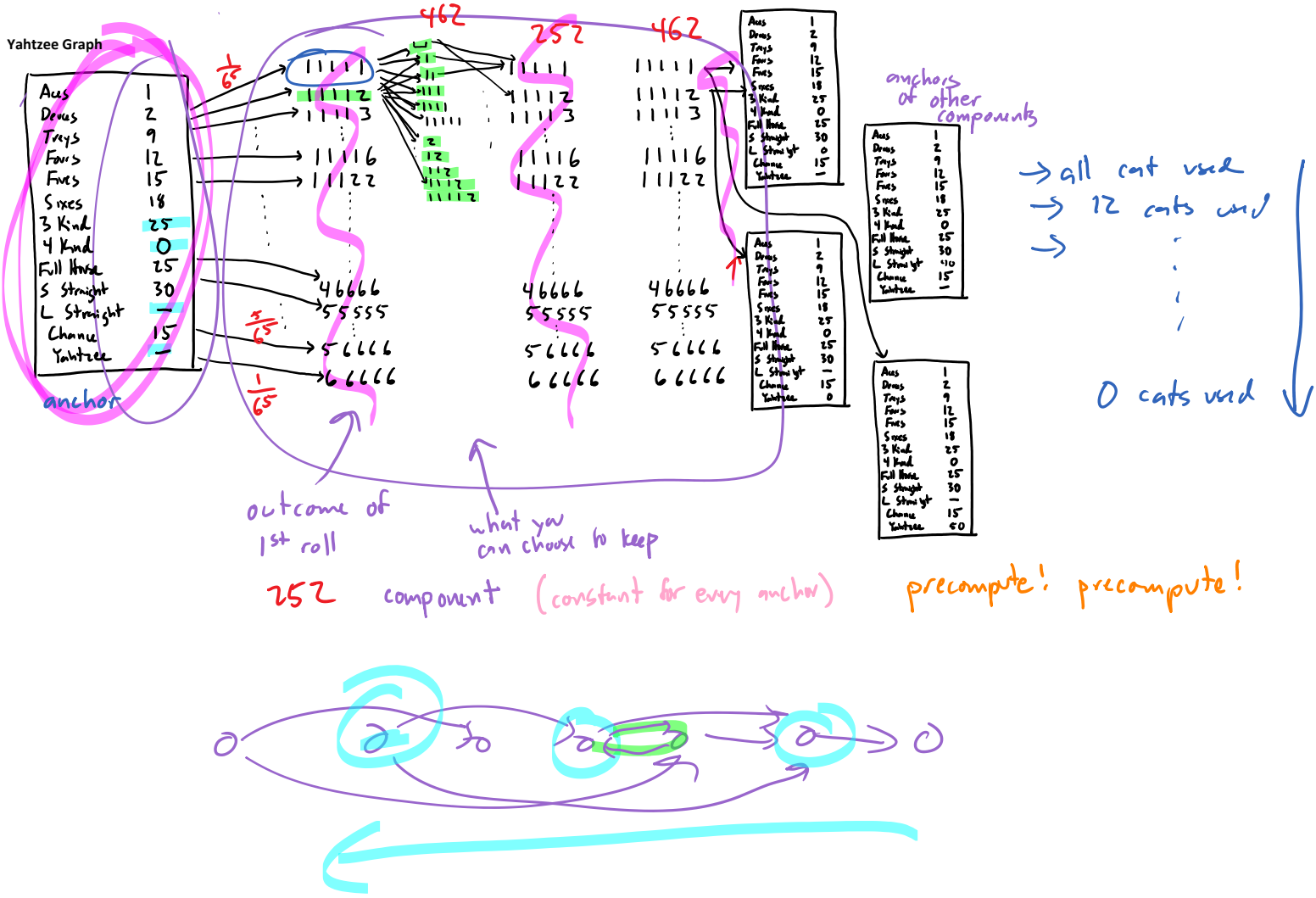
50 days @ 1 billion states/sec

aces...sixes	2^6
3K, 4K	2^2
C	2
FH, SS, LC	2^3
upper total	0...63 64
Y	used 50 used 0 unused 3

$\frac{3}{4}$ million

≈ 1.3 billion states

~ 1 min in Java code


Yahtzee Graph
462
252
462
anchors of other components
→ all cats used
→ 12 cats used
→
0 cats used
anchor
outcome of 1st roll
what you can choose to keep
252 component (constant for every anchor)
precompute! precompute!

Evaluating a Policy

$$v_\pi(s) = \sum_a \pi(a|s) \cdot \sum_{s'} \sum_r p(s', r | s, a) \cdot (r + \gamma v_\pi(s'))$$

$$= \sum_{s'} \sum_r p(s', r | s, \pi(s)) \cdot (r + \gamma v_\pi(s'))$$

for deterministic policy π

current estimate of $v_\pi(s)$

Initialize $V[s] \leftarrow 0$ for terminal and arbitrarily for other

$\Delta \leftarrow 0$

repeat

for each state s (in any order)

$v_{old} \leftarrow V[s]$

let $V[s] \leftarrow$

$\Delta \leftarrow \max(\Delta, |V[s] - v_{old}|)$

until Δ small enough