Hiring Problem and Goat Problem

more about probability

1 Combinatorics

Def: set of permutations

$$P_n = \{\pi \mid \pi : [1:n] \to [1:n] \quad \text{bijective}\}$$

number of permutations: $\#P_n$.

1 Combinatorics

Def: set of permutations

$$P_n = \{\pi \mid \pi : [1:n] \to [1:n] \quad \text{bijective}\}$$

number of permutations: $\#P_n$.

Recursion

$$\begin{aligned} \#P_1 &= 1 \\ \#P_n &= n \cdot \#P_{n-1} \end{aligned}$$

$$\#P_n = \prod_{i=1}^{n} i = n!$$

Def: n choose i

$$\binom{n}{i} = \#\{A \mid A \subseteq [1:n]\ ,\ \#A = i\}$$

number of subsets of $[1:n]$ with i elements

$$\binom{n}{i} = \frac{n \cdot (n-1) \ldots \cdot (n-i+1)}{i!} = \frac{n!}{(n-i)! \cdot i!}$$

can pick each subset in i! orders

2 Hiring Algorithm 1

Input: $(a(1),\ldots,a(n))$, where a is permutation.

$a(i)$: quality of applicant i

Hiring strategy: always hire best applicant seen so far

```
max = 0; i=1
while i <=n
{if a(i)> max {hire i ;max = a(i)};
i=i+1
}
```

2 Hiring Algorithm 1

Input: $(a(1),\dots,a(n))$, where a is permutation.

$a(i)$: quality of applicant i

Hiring strategy: always hire best applicant seen so far

```
max = 0; i=1
while i <=n
{if a(i)> max {hire i ;max = a(i)};
i=i+1
}
```

3 Hiring Problem 1

Input a is random, all permutations equally likely.

probability space

$$\begin{aligned} W &= (S,p) \\ S &= P_n \\ p(a) &= 1/n! \quad \text{for all } a \end{aligned}$$

$X(a)$ = number of applicants hired for permutation a

$$1 \leq X(a) \leq n$$

Problem:

$$E(X(a)) = ?$$

4 Indicator Variables

$W = (S, p)$ probability space

$A \subseteq S$ event

Indicator variable of A

$$X_A(a) = \begin{cases} 1 & a \in A \\ 0 & a \notin A \end{cases}$$

4 Indicator Variables

$W = (S, p)$ probability space

$A \subseteq S$ event

Indicator variable of A

$$X_A(a) = \begin{cases} 1 & a \in A \\ 0 & a \notin A \end{cases}$$

Lemma 1.

$$E(X_A) = p(A)$$

4 Indicator Variables

$W = (S, p)$ probability space

$A \subseteq S$ event

Indicator variable of A

$$X_A(a) = \begin{cases} 1 & a \in A \\ 0 & a \notin A \end{cases}$$

Lemma 1.

$$E(X_A) = p(A)$$

Proof.

$$\begin{aligned} E(X_A) &= \sum_{a \in S} X_A(a) \cdot p(a) \\ &= \sum_{a \in A} X_A(a) \cdot p(a) + \sum_{a \notin A} X_A(a) \cdot p(a) \\ &= \sum_{a \in A} 1 \cdot p(a) + \sum_{a \notin A} 0 \cdot p(a) \\ &= p(A) \end{aligned}$$

‚exciting…‘

4 Indicator Variables

$W = (S, p)$ probability space

$A \subseteq S$ event

Indicator variable of A

$$X_A(a) = \begin{cases} 1 & a \in A \\ 0 & a \notin A \end{cases}$$

Lemma 1.

$$E(X_A) = p(A)$$

Proof.

$$\begin{aligned} E(X_A) &= \sum_{a \in S} X_A(a) \cdot p(a) \\ &= \sum_{a \in A} X_A(a) \cdot p(a) + \sum_{a \notin A} X_A(a) \cdot p(a) \\ &= \sum_{a \in A} 1 \cdot p(a) + \sum_{a \notin A} 0 \cdot p(a) \\ &= p(A) \end{aligned}$$

‚exciting…‘

Event: candidate i is hired

$$H(i) = \{a \mid a(i) = \max\{a(1), \ldots a(i)\}\}$$

Indicator variable

$$X_i(a) = \begin{cases} 1 & a(i) = \max\{a(1), \ldots a(i)\} \\ 0 & \text{otherwise} \end{cases}$$

4 Indicator Variables

$W = (S, p)$ probability space

$A \subseteq S$ event

Indicator variable of A

$$X_A(a) = \begin{cases} 1 & a \in A \\ 0 & a \notin A \end{cases}$$

Lemma 1.

$$E(X_A) = p(A)$$

Proof.

$$\begin{aligned} E(X_A) &= \sum_{a \in S} X_A(a) \cdot p(a) \\ &= \sum_{a \in A} X_A(a) \cdot p(a) + \sum_{a \notin A} X_A(a) \cdot p(a) \\ &= \sum_{a \in A} 1 \cdot p(a) + \sum_{a \notin A} 0 \cdot p(a) \\ &= p(A) \end{aligned}$$

‚exciting…‘

Event: candidate i is hired

$$H(i) = \{a \mid a(i) = \max\{a(1), \ldots a(i)\}\}$$

Indicator variable

$$X_i(a) = \begin{cases} 1 & a(i) = \max\{a(1), \ldots a(i)\} \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} X(a) &= \sum_{i=1}^{n} X_i(a) && (1) \\ E(X) &= \sum_{i=1}^{n} E(X_i) \quad \text{(linearity)} && (2) \\ &= \sum_{i=1}^{n} p(H_i) \quad \text{(lemma 1)} && (3) \\ &= \sum_{i=1}^{n} \frac{\#H_i}{n!} \quad \text{(definition of } p\text{)} && (4) \end{aligned}$$

Event: candidate i is hired

$$H(i) = \{a \mid a(i) = \max\{a(1),\dots a(i)\}\}$$

Indicator variable

$$X_i(a) = \begin{cases} 1 & a(i) = \max\{a(1),\dots a(i)\} \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned}
X(a) &= \sum_{i=1}^{n} X_i(a) && (1)\\
E(X) &= \sum_{i=1}^{n} E(X_i) \quad \text{(linearity)} && (2)\\
&= \sum_{i=1}^{n} p(H_i) \quad \text{(lemma 1)} && (3)\\
&= \sum_{i=1}^{n} \frac{\#H_i}{n!} \quad \text{(definition of } p\text{)} && (4)
\end{aligned}$$

Let

$$A \subseteq [1:n]\ ,\ \#A = i$$

$$\begin{aligned}
H(A,i) &= \{a \mid \{a(1),\dots,a(i)\} = A\ ,\ a(i) = \max\{a(1),\dots a(i)\}\}\\
\#H(A,i) &= (i-1)! \cdot (n-i)! \quad \text{(no choice where to put } a(i))
\end{aligned}$$

ways to choose first /last elements

Event: candidate i is hired

$$H(i) = \{a \mid a(i) = \max\{a(1), \ldots a(i)\}\}$$

Indicator variable

$$X_i(a) = \begin{cases} 1 & a(i) = \max\{a(1), \ldots a(i)\} \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned}
X(a) &= \sum_{i=1}^{n} X_i(a) && (1)\\
E(X) &= \sum_{i=1}^{n} E(X_i) \quad \text{(linearity)} && (2)\\
&= \sum_{i=1}^{n} p(H_i) \quad \text{(lemma 1)} && (3)\\
&= \sum_{i=1}^{n} \frac{\#H_i}{n!} \quad \text{(definition of } p\text{)} && (4)
\end{aligned}$$

Let

$$A \subseteq [1:n]\ ,\ \#A = i$$

$$\begin{aligned}
H(A,i) &= \{a \mid \{a(1), \ldots, a(i)\} = A\ ,\ a(i) = \max\{a(1), \ldots a(i)\}\}\\
\#H(A,i) &= (i-1)! \cdot (n-i)! \quad \text{(no choice where to put } a(i)\text{)}
\end{aligned}$$

ways to choose first /last elements

$$\begin{aligned}
\#H(i) &= \sum_{A} \#H(A,i)\\
&= \binom{n}{i} \#H(A,i)\\
&= \frac{n! \cdot (i-1)! \cdot (n-i)!}{i! \cdot (n-i)!}\\
&= \frac{n!}{i}
\end{aligned}$$

Event: candidate i is hired

$$H(i) = \{a \mid a(i) = \max\{a(1), \dots a(i)\}\}$$

Indicator variable

$$X_i(a) = \begin{cases} 1 & a(i) = \max\{a(1), \dots a(i)\} \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} X(a) &= \sum_{i=1}^{n} X_i(a) && (1) \\ E(X) &= \sum_{i=1}^{n} E(X_i) \quad \text{(linearity)} && (2) \\ &= \sum_{i=1}^{n} p(H_i) \quad \text{(lemma 1)} && (3) \\ &= \sum_{i=1}^{n} \frac{\#H_i}{n!} \quad \text{(definition of } p\text{)} && (4) \end{aligned}$$

Let

$$A \subseteq [1:n] \text{ , } \#A = i$$

$$\begin{aligned} H(A,i) &= \{a \mid \{a(1), \dots, a(i)\} = A \text{ , } a(i) = \max\{a(1), \dots a(i)\}\} \\ \#H(A,i) &= \underset{\uparrow}{(i-1)!} \cdot \underset{\uparrow}{(n-i)!} \quad \text{(no choice where to put } a(i)\text{)} \end{aligned}$$

ways to choose first /last elements

$$\begin{aligned} \#H(i) &= \sum_{A} \#H(A,i) \\ &= \binom{n}{i} \#H(A,i) \\ &= \frac{n! \cdot (i-1)! \cdot (n-i)!}{i! \cdot (n-i)!} \\ &= \frac{n!}{i} \end{aligned}$$

$$\boxed{\begin{aligned} E(X) &= \sum_{i=1}^{n} \frac{1}{i} \\ &< \int_{1}^{n+1} \frac{1}{x} dx \\ &= ln(n+1) \end{aligned}}$$

5 Hiring Algorithm 2

Input: $(a(1),\ldots,a(n))$, where a is permutation, possibly produced by adversary

$a(i)$: quality of applicant i

Hiring strategy: always hire best applicant seen so far, *but randomize order*

5 Hiring Algorithm 2

Input: $(a(1),\ldots,a(n))$, where a is permutation, possibly produced by adversary

$a(i)$: quality of applicant i

Hiring strategy: always hire best applicant seen so far, *but randomize order*

Notation:

$$b=(b(1),\ldots,b(i))\in[1:n]^i \quad , \quad u\neq v\rightarrow\ b(u)\neq b(v)$$

vector of length i, elements pairwise distinct

5 Hiring Algorithm 2

Input: $(a(1),\ldots,a(n))$, where a is permutation, possibly produced by adversary

$a(i)$: quality of applicant i

Hiring strategy: always hire best applicant seen so far, *but randomize order*

Notation:

$$b=(b(1),\ldots,b(i))\in[1:n]^i \quad , \quad u\neq v\rightarrow\ b(u)\neq b(v)$$

vector of length i, elements pairwise distinct

$$b-b(j)=(b(1),\ldots b(j-1),b(j+1),\ldots b(i))\in[1:n]^{i-1}$$

5 Hiring Algorithm 2

Input: $(a(1),\ldots,a(n))$, where a is permutation, possibly produced by adversary

$a(i)$: quality of applicant i

Hiring strategy: always hire best applicant seen so far, *but randomize order*

Notation:

$$b=(b(1),\ldots,b(i))\in[1:n]^i \quad , \quad u\neq v\rightarrow\ b(u)\neq b(v)$$

vector of length i, elements pairwise distinct

$$b-b(j)=(b(1),\ldots b(j-1),b(j+1),\ldots b(i))\in[1:n]^{i-1}$$

Random experiment with remaining candidates b:

- choose $j\in[1:i]$; random, equal probabilities. Output of function *random*(i).
- interview $b(j)$ and remove from list

```
b=a, max = 0; i=n
while i != 0
{j_i = random(i);
if b(j)> max {hire j ;max = b(j)};
b = b - b(j); i = i-1
}
```

5 Hiring Algorithm 2

Input: $(a(1),\ldots,a(n))$, where a is permutation, possibly produced by adversary

$a(i)$: quality of applicant i

Hiring strategy: always hire best applicant seen so far, *but randomize order*

Notation:

$$b=(b(1),\ldots,b(i))\in[1:n]^i \quad , \quad u\neq v\rightarrow\ b(u)\neq b(v)$$

vector of length i, elements pairwise distinct

$$b-b(j)=(b(1),\ldots b(j-1),b(j+1),\ldots b(i))\in[1:n]^{i-1}$$

Random experiment with remaining candidates b:

- choose $j\in[1:i]$; random, equal probabilities. Output of function *random*(i).
- interview $b(j)$ and remove from list

```
b=a, max = 0; i=n
while i != 0
{j_i = random(i);
if b(j)> max {hire j ;max = b(j)};
b = b - b(j); i = i-1
}
```

Intuition:

- $(b(j_n),\ldots,b(j_1))\in P_n$
- all such permutations are equally likely *(requires proof)*.
- then analysis of algorithm 1 with random inputs works.

Random experiment with remaining candidates b:

- choose $j \in [1:i]$; random, equal probabilities. Output of function $random(i)$.
- interview $b(j)$ and remove from list

```
b=a, max = 0; i=n
while i != 0
{j_i = random(i);
if b(j)> max {hire j ;max = b(j)};
b = b - b(j); i = i-1
}
```

Intuition:

- $(b(j_n), \ldots, b(j_1)) \in P_n$
- all such permutations are equally likely *(requires proof)*.
- then analysis of algorithm 1 with random inputs works.

Probability space once vetor $b \in [1:n]^i$ is reached

$$\begin{aligned} W_b &= (S_b, p_b) \\ S_b &= \bigcup_{j=1}^{i} \{b(j)\} \times S_{b-b(j)} \\ p_b(b(j), c) &= (1/i) \cdot p_{b-b(j)}(c) \end{aligned}$$

Random experiment with remaining candidates b:

- choose $j \in [1:i]$; random, equal probabilities. Output of function $random(i)$.
- interview $b(j)$ and remove from list

```
b=a, max = 0; i=n
while i != 0
{j_i = random(i);
if b(j)> max {hire j ;max = b(j)};
b = b - b(j); i = i-1
}
```

Intuition:

- $(b(j_n),\dots,b(j_1)) \in P_n$
- all such permutations are equally likely *(requires proof)*.
- then analysis of algorithm 1 with random inputs works.

Probability space once vetor $b \in [1:n]^i$ is reached

$$\begin{aligned} W_b &= (S_b, p_b) \\ S_b &= \bigcup_{j=1}^{i} \{b(j)\} \times S_{b-b(j)} \\ p_b(b(j),c) &= (1/i) \cdot p_{b-b(j)}(c) \end{aligned}$$

End of recursion:

$$\begin{aligned} b &= (b(1)) \\ S_b &= \{b(1)\} \\ p_b(b_1) &= 1 \end{aligned}$$

Random experiment with remaining candidates b:

- choose $j \in [1:i]$; random, equal probabilities. Output of function $random(i)$.
- interview $b(j)$ and remove from list

```
b=a, max = 0; i=n
while i != 0
{j_i = random(i);
if b(j)> max {hire j ;max = b(j)};
b = b - b(j); i = i-1
}
```

Intuition:

- $(b(j_n),\dots,b(j_1)) \in P_n$
- all such permutations are equally likely *(requires proof)*.
- then analysis of algorithm 1 with random inputs works.

nested pairs

Probability space once vetor $b \in [1:n]^i$ is reached

$$
\begin{aligned}
W_b &= (S_b, p_b)\\
S_b &= \bigcup_{j=1}^{i} \{b(j)\} \times S_{b-b(j)}\\
p_b(b(j),c) &= (1/i) \cdot p_{b-b(j)}(c)
\end{aligned}
$$

End of recursion:

$$
\begin{aligned}
b &= (b(1))\\
S_b &= \{b(1)\}\\
p_b(b_1) &= 1
\end{aligned}
$$

Outcome $b' \in S_b$:

$$b' = (b_{\pi(1)}, (b_{\pi(2)}, (\dots, (b_{\pi(i)})\dots))) \quad , \quad \pi \in P_i$$

Random experiment with remaining candidates b:

- choose $j \in [1:i]$; random, equal probabilities. Output of function $random(i)$.
- interview $b(j)$ and remove from list

```
b=a, max = 0; i=n
while i != 0
{j_i = random(i);
if b(j)> max {hire j ;max = b(j)};
b = b - b(j); i = i-1
}
```

Intuition:

- $(b(j_n),\dots,b(j_1)) \in P_n$
- all such permutations are equally likely *(requires proof)*.
- then analysis of algorithm 1 with random inputs works.

Probability space once vetor $b \in [1:n]^i$ is reached

$$
\begin{aligned}
W_b &= (S_b, p_b) \\
S_b &= \bigcup_{j=1}^{i} \{b(j)\} \times S_{b-b(j)} \\
p_b(b(j),c) &= (1/i) \cdot p_{b-b(j)}(c)
\end{aligned}
$$

nested pairs

End of recursion:

$$
\begin{aligned}
b &= (b(1)) \\
S_b &= \{b(1)\} \\
p_b(b_1) &= 1
\end{aligned}
$$

Outcome $b' \in S_b$:

$$b' = (b_{\pi(1)}, (b_{\pi(2)}, (\dots, (b_{\pi(i)})\dots))) \quad , \quad \pi \in P_i$$

all outcomes equally likely:

Lemma 2. *For all $i \in [1:n]$, for all $b \in [1:n]^i$ with pairwise distinct lements and for all $b' \in S_b$*

$$p_b(b') = \frac{1}{i!}$$

Proof. induction on i □

6 Why I am paranoid about probability: the goat problem

Game show:

- three doors
- gold behind 1 door, goats behind 2 other doors
- you guess door with gold with $p = 1/3$
- show master opens a door without gold (if you guessed right he chooses door with $q = 1/2$)
- now you are free to keep your guess or change it to the other closed door

6 Why I am paranoid about probability: the goat problem

Game show:

- three doors
- gold behind 1 door, goats behind 2 other doors
- you guess door with gold with $p = 1/3$
- show master opens a door without gold (if you guessed right he chooses door with $q = 1/2$)
- now you are free to keep your guess or change it to the other closed door

Intuition:

- changing your guess cannot improve your expectation to win
- except it does

6 Why I am paranoid about probability: the goat problem

Game show:

- three doors
- gold behind 1 door, goats behind 2 other doors
- you guess door with gold with $p = 1/3$
- show master opens a door without gold (if you guessed right he chooses door with $q = 1/2$)
- now you are free to keep your guess or change it to the other closed door

Intuition:

- changing your guess cannot improve your expectation to win
- except it does

notation: A finite set. Function $random(A)$

- returns each $a \in A$ with equal probability
- $W = (A, p)$ with $p(a) = 1/\#A$ for all a

6 Why I am paranoid about probability: the goat problem

Game show:

- three doors
- gold behind 1 door, goats behind 2 other doors
- you guess door with gold with $p = 1/3$
- show master opens a door without gold (if you guessed right he chooses door with $q = 1/2$)
- now you are free to keep your guess or change it to the other closed door

Intuition:

- changing your guess cannot improve your expectation to win
- except it does

notation: A finite set. Function $random(A)$

- returns each $a \in A$ with equal probability
- $W = (A, p)$ with $p(a) = 1/\#A$ for all a

$D = \{w, a, b\}$ doors; gold behind w (like 'win')
Construct probability space; two experiments

1. your guess $g = random(D)$
2. door opened by show master

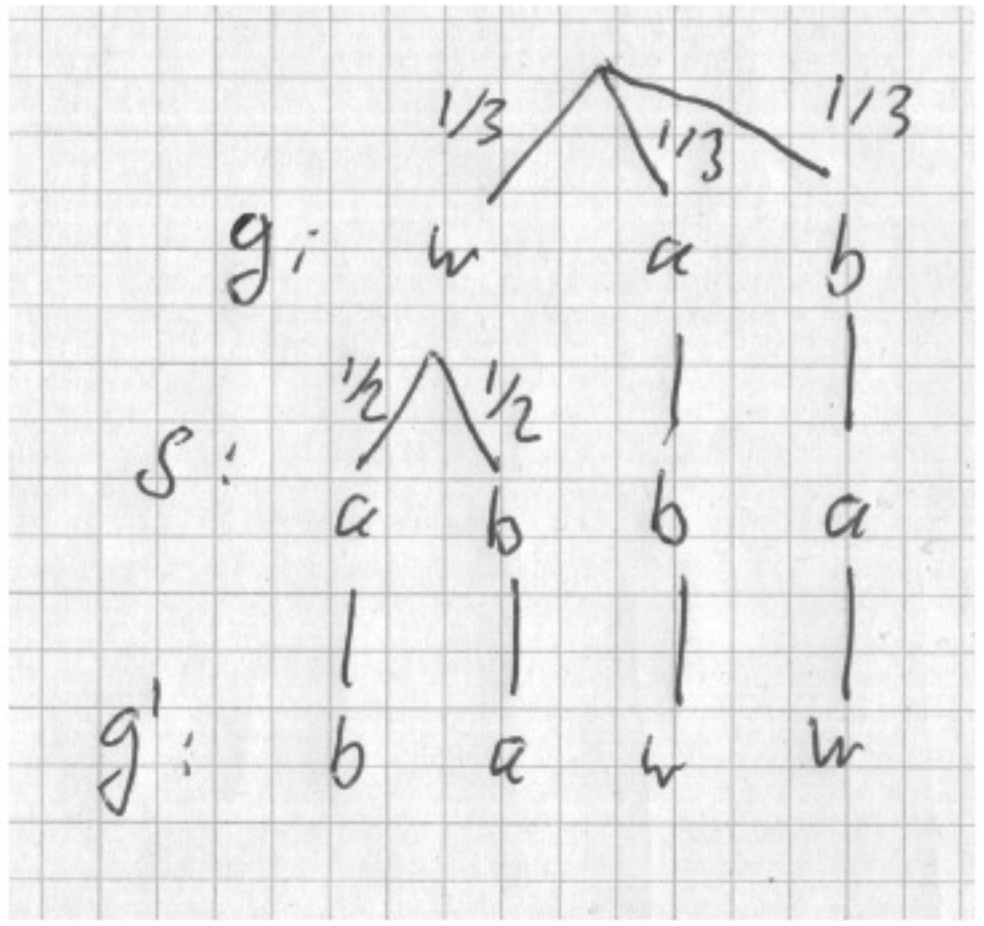
$$s = \begin{cases} random(\{a,b\}) & g = w \\ a & g = b \\ b & g = a \end{cases}$$

Combined probability space (S, r)

$$S = \{(w,a), (w,b), (a,b), (b,a)\}$$

$$r(w,a) = r(w,b) = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}$$

$$r(a,b) = r(b,a) = \frac{1}{3} \cdot 1 = \frac{1}{3}$$



6 Why I am paranoid about probability: the goat problem

Game show:

- three doors
- gold behind 1 door, goats behind 2 other doors
- you guess door with gold with $p = 1/3$
- show master opens a door without gold (if you guessed right he chooses door with $q = 1/2$)
- now you are free to keep your guess or change it to the other closed door

Intuition:

- changing your guess cannot improve your expectation to win
- except it does

notation: A finite set. Function $random(A)$

- returns each $a \in A$ with equal probability
- $W = (A, p)$ with $p(a) = 1/\#A$ for all a

$D = \{w, a, b\}$ doors; gold behind w (like 'win')
Construct probability space; two experiments

1. your guess $g = random(D)$
2. door opened by show master

$$s = \begin{cases} random(\{a,b\}) & g = w \\ a & g = b \\ b & g = a \end{cases}$$

$D = \{w, a, b\}$ doors; gold behind w (like 'win')
Construct probability space; two experiments

1. your guess $g = random(D)$

2. door opened by show master

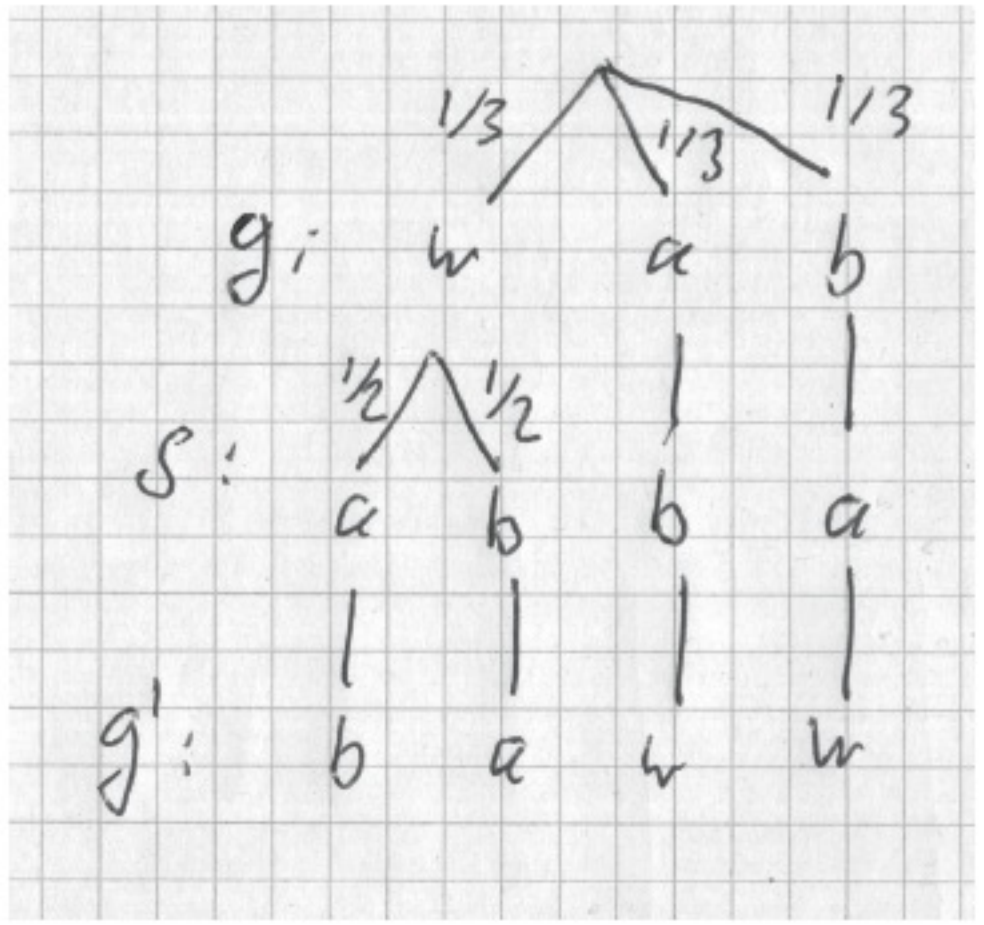
$$s = \begin{cases} random(\{a,b\}) & g = w \\ a & g = b \\ b & g = a \end{cases}$$

Combined probability space (S, r)

$$S = \{(w,a),(w,b),(a,b),(b,a)\}$$

$$r(w,a) = r(w,b) = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}$$

$$r(a,b) = r(b,a) = \frac{1}{3} \cdot 1 = \frac{1}{3}$$



No change: winning event is

$$W = \{(w,a),(w,b)\} \quad , \quad r(W) = 1/3$$

$D = \{w, a, b\}$ doors; gold behind w (like 'win')
Construct probability space; two experiments

1. your guess $g = random(D)$

2. door opened by show master

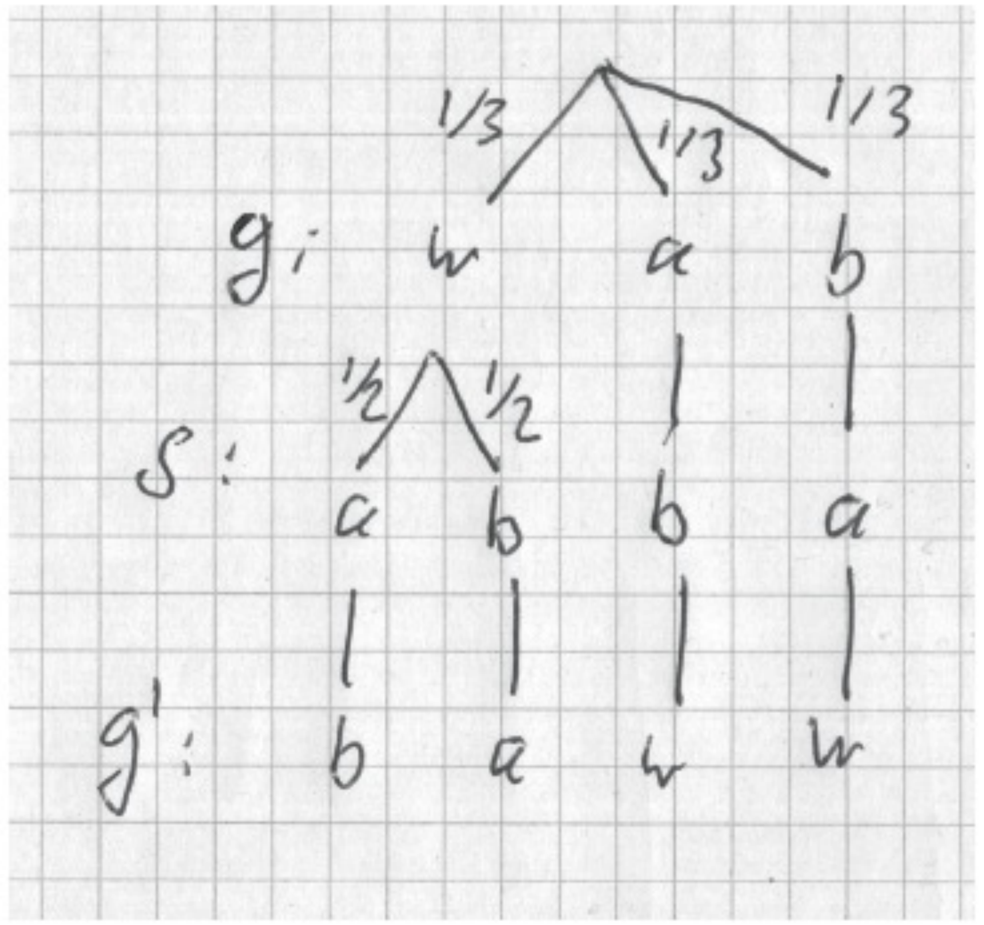
$$s = \begin{cases} random(\{a,b\}) & g = w \\ a & g = b \\ b & g = a \end{cases}$$

Combined probability space (S, r)

$$S = \{(w,a),(w,b),(a,b),(b,a)\}$$

$$r(w,a) = r(w,b) = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}$$

$$r(a,b) = r(b,a) = \frac{1}{3} \cdot 1 = \frac{1}{3}$$



No change: winning event is

$$W = \{(w,a),(w,b)\} \quad , \quad r(W) = 1/3$$

Change guess to

$$g' = \begin{cases} b & (g,s) = (w,a) \\ a & (g,s) = (w,b \\ w & (g,s) = (a,b) \\ w & (g,s) = (b,a) \end{cases}$$

$D = \{w, a, b\}$ doors; gold behind w (like 'win')
Construct probability space; two experiments

1. your guess $g = random(D)$
2. door opened by show master

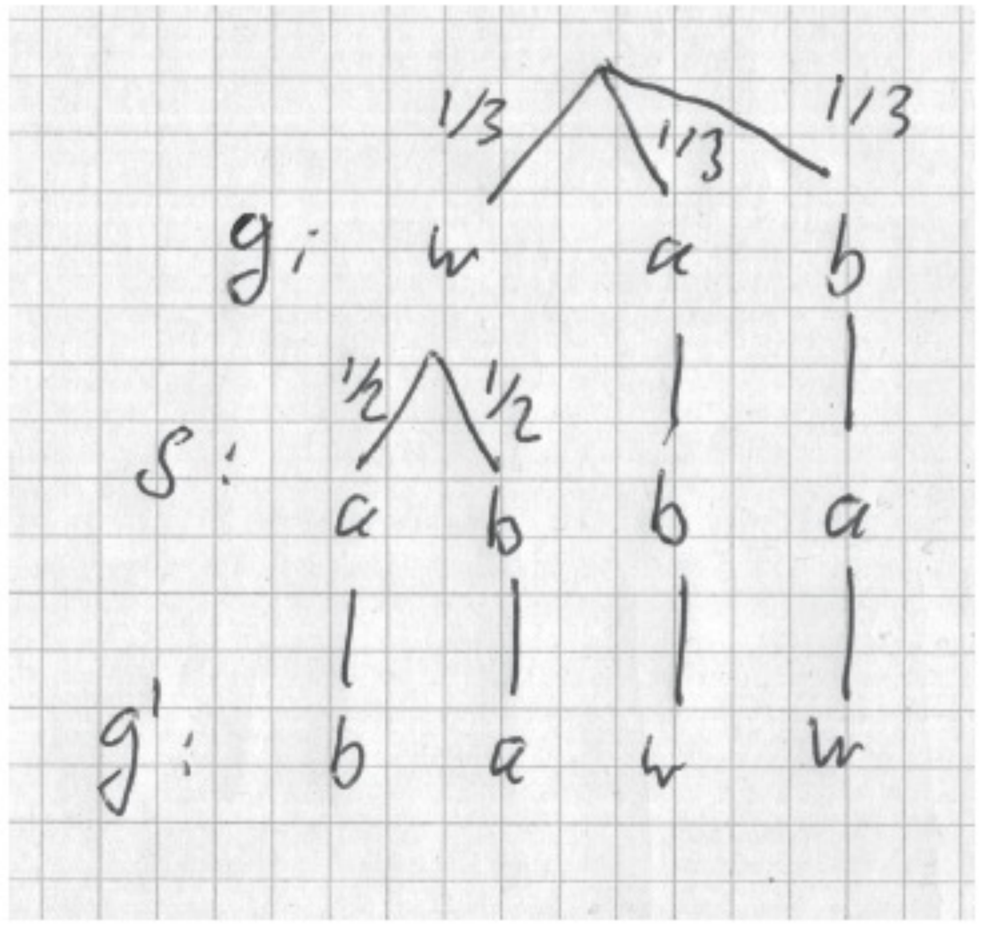
$$s = \begin{cases} random(\{a,b\}) & g = w \\ a & g = b \\ b & g = a \end{cases}$$

Combined probability space (S, r)

$$S = \{(w,a),(w,b),(a,b),(b,a)\}$$

$$r(w,a) = r(w,b) = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}$$

$$r(a,b) = r(b,a) = \frac{1}{3} \cdot 1 = \frac{1}{3}$$



No change: winning event is

$$\boxed{W = \{(w,a),(w,b)\} \quad , \quad r(W) = 1/3}$$

Change guess to

$$g' = \begin{cases} b & (g,s) = (w,a) \\ a & (g,s) = (w,b \\ w & (g,s) = (a,b) \\ w & (g,s) = (b,a) \end{cases}$$

You win if you first guessed wrong; you loose if you first guessed right. New winning event

$$\boxed{W' = \{(a,b),(b,a)\} \quad , \quad r(W') = 2/3}$$