LMPI 2022/2023 Homework

Matteo D'Agostino 1929765
Mirko Magistrado Dacara 1932131

A Logica e Metodi Probabilistici per L'Informatica Homework Assignment

July 30, 2023

Problem 1

Solution. 1a Indichiamo per semplicità di notazione i vari oggetti che si susseguono nella sequenza con l'indice di comparsa: al tempo t l'oggetto della sequenza sarà cioè rappresentato dal numero naturale $t \in \mathbb{N}$, dunque $N_t \equiv \{n \in \mathbb{N}^* \mid n \leq t\}$. L'insieme $S_t = \{1, 4, 6\}$ rappresenta ad esempio una possibile configurazione di memoria al tempo $t = 8$ per $k = 3$ Procediamo nella dimostrazione della formula presentata per induzione:

Proof. a) **Base induttiva**

$\forall 1 \leq t \leq k$ risulta che l'insieme S_t è univocamente determinato perchè la memoria non è stata ancora riempita. In particolare , per $t = k$, risulta che $S_t \equiv \{1, 2, ..., k\}$ e che $\mathbb{P}(S_t = X) = \binom{t}{k}^{-1} = \binom{k}{k}^{-1} = 1$ in accordo col fatto che X sarà l'unica configurazione di memoria possibile. Dunque per $t = k$ la formula è soddisfatta

b) **Passo induttivo**

Assumendo quanto esplicitato valido al tempo generico t bisogna analizzare cosa accade al tempo $t+1$. Siano X e Y rispettivamente la configurazione al tempo t e quella al tempo $t + 1$. Considero due eventi disgiunti: $X = Y$ e $X \neq Y$. Nel primo caso si nota come per avere la medesima configurazione ai tempi $t+1$ e t è necessario (per la proprietà per cui gli elementi scartati o rimossi non possono essere più considerati) che al tempo $t+1$ esca croce. Dunque, $\mathbb{P}(S_{t+1} = X) = \mathbb{P}(S_t = X) \cdot \mathbb{P}(EscaCroce_{t+1}) = \binom{t}{k}^{-1} \cdot \frac{t+1-k}{t+1} = \binom{t+1}{k}^{-1} (cvd)$. Nel caso in cui $X \neq Y$ vorrà dire che X e Y differiscono di un solo elemento: l'oggetto $t + 1$ entrerà in memoria nella configurazione Y. E' facile mostrare che per ogni possibile $S_{t+1} = Y$ esistono $t-(k-1) = t-k+1$ configurazioni $S_t = X$, tante quanti i modi di variare l'unico elemento che li distingue e che sarà rimpiazzato in Y con $t + 1$: a sostegno di ciò si prenda in considerazione un' istanza del problema per cui $k = 3,\ t = 6,\ S_7 = \{1, 2, 7\}$; allora $S_6 \in \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 2, 6\}\} = U$ e $\mid U \mid = t - k + 1 = 6 - 3 + 1 = 4$. Considerando che per ipotesi induttiva tutte le configurazioni sono equiprobabili e nello specifico hanno probabilità pari a $\binom{t}{k}^{-1}$ e che le probabilità che esca testa e che l'elemento sostituito sia proprio quello che lascerà il posto a $t + 1$ sono rispettivamente $\frac{k}{t+1}$ e $\frac{1}{k}$ si ha dunque la seguente relazione, analoga alla precedente

$$\mathbb{P}(S_{t+1} = Y) = (t - k + 1) \cdot \mathbb{P}(S_t = X) \cdot \frac{k}{t+1} \cdot \frac{1}{k} = (t - k + 1) \cdot \binom{t}{k}^{-1} \cdot \frac{1}{t+1} = \binom{t+1}{k}^{-1}$$

□

Solution. 1b Affinchè le variabili aleatorie S_t e S_l siano indipendenti deve risultare che $\mathbb{P}(S_t = X, S_l = Y) = \mathbb{P}(S_t = X) \cdot \mathbb{P}(S_l = Y)\ \forall\, X, Y\, possibili,\ \forall\, t, l \mid l \neq t$. Si consideri a tal riguardo il caso pocanzi descritto in cui $X = Y$ e $l = t + 1$ per concludere che

$$\mathbb{P}(S_t = X, S_{t+1} = X) = \mathbb{P}(EscaCroce_{t+1}) = \frac{t-k+1}{t+1} \neq \binom{t}{k}^{-1} \cdot \binom{t+1}{k}^{-1} = \mathbb{P}(S_t = X) \cdot \mathbb{P}(S_{t+1} = X)$$

Intuitivamente, la configurazione al tempo t influenza pesantemente quanto accadrà con determinata probabilità al tempo $t + 1$. Le due V.A. sono dunque dipendenti.

Problem 2

Solution. Indichiamo con B_i il contributo di bias che l'$i-esimo$ messaggio di disturbo apporta al messaggio effettivamente inviato b, cioè $B_i = p_i \cdot b_i$. Definisco inoltre $B = \sum_{i=1}^{n} B_i$. Risulta che $\mathbb{E}(b_i) = \frac{1}{2} \cdot (-1) + \frac{1}{2} \cdot (+1) = 0$, $\mathbb{E}(B_i) = \mathbb{E}(p_i \cdot b_i) = p_i \cdot \mathbb{E}(b_i) = 0$ e dunque che $\mathbb{E}(B) = \mathbb{E}(\sum_{i=1}^{n} B_i) = \sum_{i=1}^{n} \mathbb{E}(B_i) = 0$ per la proprietà di linearità del valore atteso. Per quanto concerne la varianza si ha che $Var(b_i) = \mathbb{E}(b_i^2) - (\mathbb{E}(b_i))^2 = \frac{1}{2} + \frac{1}{2} + 0 = 1$, come intuibile dal fatto per cui entrambi i valori che b_i può assumere si discostano esattamente 1 dal valore atteso 0.

Utilizzando la linearità della varianza nel caso di V.A. non correlate (caso in cui ci troviamo poichè le b_i sono *i.i.d*) si ha $Var(B) = Var(\sum_{i=1}^{n} B_i) = \sum_{i=1}^{n} Var(B_i) = \sum_{i=1}^{n} Var(p_i \cdot b_i) = \sum_{i=1}^{n} p_i^2 \cdot Var(b_i) = \sum_{i=1}^{n} p_i^2$. Senza perdita di generalità supponiamo di essere nello specifico caso in cui $b = -1$ (il problema gode di una certa simmetria e dunque vale lo stesso anche per $b = 1$). Il ricevente sbaglierà a decodificare il messaggio se $B \geq 1$. Per la disuguaglianza di Chebyshev si ha:

$$\forall\ v.a.\ X,\ \forall\ a > 0,\ \mathbb{P}(\ |\ X - \mathbb{E}(X)\ |\ \geq a) \leq \frac{Var(X)}{a^2}$$

Pertanto $\mathbb{P}(B \geq 1) = \frac{1}{2}\mathbb{P}(\ |\ B\ |\ \geq 1) \leq \frac{1}{2} \cdot \frac{Var(B)}{1} = \frac{\sum_{i=1}^{n} p_i^2}{2}$. La prima uguaglianza è resa possibile dalla simmetria della variabile aleatoria B.

Per una stima più fine è possibile utilizzare la disuguaglianza di Hoeffding: le B_i sono variabili aleatorie indipendenti, sono limitate in valore tra $-p_i$ ed p_i. Dato che per definizione $B = \sum_{i=1}^{n} B_i$ risulta che $\mathbb{P}(B \geq 1) \leq exp(-\frac{2}{\sum_{i=1}^{n}(p_i - (-p_i))^2}) = e^{\frac{-1}{2\sum_{i=1}^{n} p_i}}$, limite superiore alla probabilità di errore nella decodifica del messaggio dipendente dal numero di elementi di disturbo.

Problem 3

Solution. 3a Nel caso in cui $k = n = 1$ l'algoritmo termina restituendo correttamente l'unico elemento della lista selezionato come pivot (viene soddisfatta la condizione di riga 6). Nel caso generale si ha invece che i si riduce progressivamente di almeno una unità tra una chiamata di funzione e l'altra e al tempo stesso k viene minorato da 0 e forzatamente maggiorato da i poichè se maggiore o uguale a quest'ultimo costretto a diminuire con la chiamata ricorsiva successiva. Si ha pertanto certezza di convergenza verso la base induttiva dell'algoritmo, rappresentato dalle righe 6-7.

Solution. 3b Uno dei casi peggiori (si potrebbe dire *sfortunati*) si ha quando $k = 1$ e il pivot scelto ad ogni iterazione risulta essere sempre il maggiore degli elementi della lista S passata in input la quale si riduce continuamente di un solo elemento: $N_{max} = (n-1) + (n-2) + ... + 2 + 1 = \sum_{i=1}^{n-1} i = \frac{(n-1) \cdot n}{2} \in \mathcal{O}(n^2)$.

Solution. 3c Definiamo *buoni* una chiamata alla funzione proposta e relativo split sulla lista degli elementi in input se il pivot scelto divide la lista stessa in due sottoliste, entrambe di lunghezza pari al più a $\frac{2}{3}$ della prima: questo accade con probabilità $\mathbb{P}(ChiamataBuona) = \frac{1}{3}$. A seguito di ogni split buono la dimensione della lista su cui iterare si riduce almeno di un fattore $\frac{1}{3}$ e dunque dopo lo j-esimo split avrò al più $n \cdot (\frac{2}{3})^j$ elementi. Dalla relazione $n \cdot (\frac{2}{3})^k \geq 1$ si ha che il numero massimo di split buoni lungo un percorso dalla radice ad una delle foglie dell'albero di ricorsione è $\log_{\frac{3}{2}}(n)$. Indicando con B_i il numero di iterazioni che intercorrono dall' i-esimo split buono all'$i+1$-esimo si ha che B_i ha distribuzione geometrica $B_i \sim G(\frac{1}{3})\ \forall\ i$. Dunque il valore atteso è $\mathbb{E}(B_i) = 3$; ad ogni chiamata devo iterare sugli elementi presenti nella lista, dunque $\mathbb{E}(N) \leq \sum_{i=0}^{\log_{\frac{3}{2}} n} n \cdot (\frac{2}{3})^i \cdot \mathbb{E}(B_i) = 3n \cdot \sum_{i=0}^{\log_{\frac{3}{2}} n} (\frac{2}{3})^i = 3n \cdot \frac{1-(\frac{2}{3})^{\log_{\frac{3}{2}} n+1}}{1-\frac{2}{3}} = 9n - 6 \in \mathcal{O}(n)$.

Solution. 3d Analogamente a quanto visto nel precedente punto è possibile stimare per eccesso il numero di confronti N con una variabile aleatoria X definita come segue:

$$X = \sum_{i=0}^{\log_{\frac{3}{2}} n} (\frac{2}{3})^i \cdot n \cdot B_i$$

dove B_i ricordiamo essere il numero di split cattivi che intercorrono tra l'i-esimo buono e l'i+1 esimo. Qualora riuscissimo a limitare dall'alto (e con *"alta probabilità"*) le varie B_i (per definizione geometriche che vivono in tutto $\mathbb{N}$) con un valore $C \in \mathbb{N}$ allora avremmo (sempre con alta probabilità) che $N \leq X \leq Cn \sum_{i=0}^{\log_{\frac{3}{2}} n} (\frac{2}{3})^i \leq 3Cn \in \mathcal{O}(n)$.

Vogliamo pertanto calcolare la probabilità che tutte le B_i assumano valore inferiore a tale C, attraverso l'evento complementare.

$$\mathbb{P}(B_i > C) = \sum_{k=C+1}^{\infty} \mathbb{P}(B_i = k) \leq \sum_{k=C+1}^{\infty} (\frac{2}{3})^i = \sum_{k=0}^{\infty} (\frac{2}{3})^i - \sum_{k=0}^{C} (\frac{2}{3})^i = 3 - \frac{1-(\frac{2}{3})^{C+1}}{1-\frac{2}{3}} = 3(\frac{2}{3})^{C+1}$$

La probabilità che almeno una B_i assuma un valore superiore a C è rappresentata da (grazie a *Union Bound*)

$$\mathbb{P}(\bigcup_{i=0}^{\log_{\frac{3}{2}} n} (B_i > C)) \leq \sum_{i=0}^{\log_{\frac{3}{2}} n} \mathbb{P}(B_i > C) \leq 3(\frac{2}{3})^{C+1} \cdot \log_{\frac{3}{2}} n$$

Scegliendo $C = k + \log_{\frac{3}{2}}(\log_{\frac{3}{2}} n)$ la probabilità che una fase B_i duri più di C iterazioni è maggiorata da $2 \cdot (\frac{2}{3})^k$. Infine, scegliendo $k = \log_{\frac{3}{2}} n$ si ha che

$$\mathbb{P}(\bigcap_{i=0}^{\log_{\frac{3}{2}} n} B_i \leq C) \geq 1 - \frac{2}{n} \to 1 \ (per\ n \to \infty)$$

Attenzione! Si noti che il valore C trovato non è costante ma è in relazione logaritmica con n, pertanto si è dimostrato *con alta probabilità* che l'algoritmo di Quickselect presentato ha complessità $\mathcal{O}(nlogn)$. Intuitivamente, per dimostrare invece la linearità si potrebbe procedere come fatto per *Quicksort* e concludere che a differenza di quest'ultimo non vengono effettuati n confronti ad ogni livello dell'albero associato all'esecuzione dell'algoritmo in quanto si procede solo in una delle porzioni di array nelle quali quello originale viene splittato dal pivot. Non siamo tuttavia riusciti a dimostrarlo rigorosamente.

Problem 4

Solution. 4a

```
clustering_sum = 0
for v in V:
   C_v = 0
   already_visited = set()
   len_neighbors = len(N(v))
   for neighbor_1 in N(v):
      for neighbor_2 in N(v):
         if neighbor_1 != neighbor_2 and (neighbor_1,neighbor_2) not in already_visited and {neighbor_1, neighbor_2} in E:
            C_v += 1
            already_visited.add((neighbor_1, neighbor_2)) # to avoid counting twice the same edge
            already_visited.add((neighbor_2,neighbor_1)) # same as above
   if len_neighbors < 2:
      C_v = 0
   else:
      C_v = 2 * C_v / (len_neighbors * (len_neighbors - 1))
   clustering_sum += C_v
C_G = clustering_sum / len(V)
```

L'algoritmo fornisce necessariamente la corretta misura del coefficiente di clustering globale del grafo in quanto basato sulla scansione esaustiva dello stesso e sulla definizione di C_G. Analizziamone la complessità computazionale: per ogni nodo $v \in V$ (la cui scansione avviene in $\mathcal{O}(n)$) prendiamo a coppie i nodi che compongono il suo vicinato $N(v)$ e questo lo facciamo in $\mathcal{O}(n^2)$ per poi verificare se tale coppia sia o meno presente nell'insieme degli archi E (tale verifica possiamo ipotizzare avvenga nel caso peggiore in $\mathcal{O}(m)$. Dunque la complessità temporale dell'algoritmo proposto è $\mathcal{O}(n^3 \cdot m)$. Il fattore m può essere annullato mediante l'utilizzo di una *matrice di adiacenza* o di un *hashset* per memorizzare i vari archi, caso in cui il costo diverebbe con alta probabilità $\mathcal{O}(n^3)$.

Solution. 4b Bisogna dimostrare che X_i restituito dall'algoritmo randomizzato sia uno stimatore unbiased per il coefficiente di clustering del grafo C_G, cioè che $\mathbb{E}(X_i) = C_G$. Sappiamo che X_i è variabile indicatrice dell'evento *"avendo pescato nel campione i-esimo un qualsiasi nodo n, i due nodi v,w estratti dal vicinato di n sono collegati da un arco"* dunque il suo valore atteso coincide con la probabilità si verifichi l'evento stesso, $\mathbb{E}(X_i) = \mathbb{P}(X_i = 1)$. Denotiamo con Y_i la variabile indicatrice dell'evento *"avendo pescato esattamente il nodo i-esimo del grafo, i due nodi v,w estratti dal vicinato di i sono collegati da un arco"*; si noti che tale evento è differente da quello poc'anzi definito e che $\mathbb{P}(Y_i = 1) = C_i$, coefficiente di clustering locale relativo al nodo i . Si consideri inoltre l'evento $A_{i,k}$ *"l'i-esimo nodo estratto (i = 1,...,l) corrisponde al k-esimo nodo del grafo, k = 1,...,n"* Per il teorema della probabilità totale applicato a eventi mutuamente esclusivi si ha che $\mathbb{P}(X_i = 1) = \sum_{k=1}^{n} \mathbb{P}(Y_k = 1) \cdot \mathbb{P}(A_{i,k}) = \frac{1}{n} \cdot \sum_{k=1}^{n} \mathbb{P}(Y_k = 1) = \frac{1}{n} \cdot \sum_{k=1}^{n} C_k$. Tale termine per definizione coincide con il coefficiente di clustering globale del grafo C_G. Riassumendo,

$$\mathbb{E}(X_i) = \frac{1}{n} \cdot \sum_{k=1}^{n} C_k = C_G$$

cvd.

Solution. 4c Ricordiamo che $X = \frac{1}{l} \cdot \sum_{i=1}^{l} X_i$ e che, per quanto dimostrato dal precedente punto e per linearità del valore atteso, $\mathbb{E}(X) = \mathbb{E}(X_i) = C_G$. L'ultima relazione scritta nella traccia del problema può essere ricondotta alla seguente per mezzo di semplici manipolazioni di calcolo:

$$\mid X - C_G \mid \leq \epsilon C_G$$

. Dobbiamo dunque confrontare con $1 - \delta$ i seguenti bound di deviazione dall'alto e dal basso di *Chernoff* $\mathbb{P}(\mid X - C_G \mid \leq \epsilon C_G) = 1 - \mathbb{P}(\mid X - C_G \mid \geq \epsilon C_G) = 1 - \mathbb{P}(X - C_G \leq -\epsilon C_G) - \mathbb{P}(X - C_G \geq \epsilon C_G) = 1 - \mathbb{P}(X \leq (1 - \epsilon) \cdot C_G) - \mathbb{P}(X \geq (1 + \epsilon) \cdot C_G) \geq 1 - e^{\frac{-C_G l \epsilon^2}{2}} - e^{\frac{-C_G l \epsilon^2}{3}} \geq 1 - 2e^{\frac{-C_G l \epsilon^2}{3}}$, ipotizzando che $\epsilon < 1$. Quest'ultimo termine deve risultare almeno pari a $1 - \delta$, dunque, ricapitolando:

$$\mathbb{P}(\mid X - C_G \mid \leq \epsilon C_G) \geq 1 - 2e^{\frac{-C_G l \epsilon^2}{3}} \geq 1 - \delta \iff 2e^{\frac{-C_G l \epsilon^2}{3}} \leq \delta \iff \frac{-C_G l \epsilon^2}{3} \leq ln(\frac{\delta}{2}) \iff \frac{C_G l \epsilon^2}{3} \geq ln(\frac{2}{\delta}) \iff$$

$$l \geq \frac{3}{C_G \epsilon^2} \cdot ln(\frac{2}{\delta})$$

cvd.

Problem 5

Solution. 5a Indichiamo rispettivamente con X_t, $\vec{p}_t(i)$ lo stato al tempo t e la probabilità al tempo t di stare nello stato i. $X_t \in S = \{0, 1, 2, ..., a, ..., a + b\}$ $\forall$ $t \in \mathbf{N}$ (gli stati sono cioè le posizioni sulla fune) , $X_0 = a$ e la matrice di transizione è

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}$$

In particolare si presenta come una matrice nulla con $\frac{1}{2}$ sulla sopra(e sotto)-diagonale principale. La prima e l'ultima riga si distinguono invece perchè gli stati 0 e $a + b$ sono stati pozzo (*absorbing states*) in quanto rappresentano rispettivamente la terra e il fiume ed in corrispondenza di essi ha termine la camminata aleatoria.

Solution. **5b** La catena è aperiodica in quanto non tutti gli stati sono periodici (gli stati 0 e $a+b$ hanno periodo pari a 1 mentre i restanti hanno periodo 2). Non è irriducibile in quanto gli *estremi* sono stati pozzo e non consentono cioè di raggiungere altri stati, diversi da loro stessi. Dunque, la catena di Markov non è ergodica: a sostegno di ciò è possibile individuare in essa stati che non sono *ergodici* poichè periodici (tutti quelli intermedi, diversi cioè da 0 e $a+b$).

Solution. **5c** Chiamiamo con p_s la probabilità che Frederick ha di cadere in acqua se parte dalla posizione s della fune (e dunque dallo stato s della catena di Markov sopra descritta). E' immediato che $p_0 = 0$ e che $p_{a+b} = 1$; condizionando rispetto al primo passo compiuto, risulta che $p_s = \frac{1}{2}p_{s-1} + \frac{1}{2}p_{s+1} \iff 2p_s = p_{s-1} + p_{s+1} \iff p_s - p_{s-1} = p_{s+1} - p_s$. Definendo $\Delta_i = p_{i+1} - p_i$ riscriviamo la precedente come $\Delta_{i-1} = \Delta_i$. Risulta pertanto che $\Delta_i = \Delta_{i-1} = \Delta_{i-2} = \cdots = \Delta_0 \ \forall\ i = 0, \cdots, a+b-1$. Dato che $\sum_{i=0}^{a+b-1} \Delta_i = (a+b) \cdot \Delta_0 = p_1 - p_0 + p2 - p1 + \cdots + p_{a+b} - p_{a+b-1} = p_{a+b} - p_0$ (*somma telescopica*) $= 1$ si evince che $\Delta_0 = p_1 - p_0 = \frac{1}{a+b}$. Sfruttando tale principio ogni probabilità p_n può essere scritta come $p_n = \sum_{i=0}^{n-1} \Delta_i = \Delta_0 \cdot \sum_{i=0}^{n-1} 1 = n \cdot \Delta_0 = \frac{n}{a+b}$. Dato che Frederick parte dalla posizione a risulta $p_a = \frac{a}{a+b}$.

Solution. **5d** Denotiamo con μ_i il valore atteso dei passi che Frederick deve compiere, a partire dallo stato (i.e. posizione) i, prima che la random walk termini (in uno qualsiasi degli estremi della corda). E' immediato che

$$\mu_0 = \mu_{a+b} = 0 \tag{1}$$

. Similmente a quanto visto prima risulta che $\mu_n = 1 + \frac{1}{2}\mu_{n-1} + \frac{1}{2}\mu_{n+1}\ \forall\ n = 1, \cdots, a+b-1$. Manipolando tale relazione si ottiene $\mu_n - \mu_{n-1} = \mu_{n+1} - \mu_n + 2$; definendo $\Delta_n = \mu_{n+1} - \mu_n$ si ottiene che $\Delta_n = \Delta_{n-1} - 2 = \Delta_{n-2} - 4 = \cdots = \Delta_{n-i} - 2i = \cdots = \Delta_0 - 2n$. Per (1) si ottiene che $\mu_{a+b} - \mu_0 = \sum_{i=0}^{a+b-1} \Delta_i = \sum_{i=0}^{a+b-1} (\Delta_0 - 2i) = (a+b) \cdot \Delta_0 - (a+b-1) \cdot (a+b) = 0 \iff \Delta_0 = a+b-1$ (escludiamo il caso non interessante in cui $a+b=0$). Troviamo infine $\mu_a = \sum_{i=0}^{a-1} \Delta_0 - 2i = a \cdot (a+b-1) - a^2 + a = ab$.

Solution. **5e** Con tale modifica la precedente relazione diventa

$$\mu_n = 1 + \frac{2}{3}\mu_{n-1} + \frac{1}{3}\mu_{n+1} \iff 2\mu_n - 2\mu_{n-1} = \mu_{n+1} - \mu_n + 3 \iff 2\Delta_{n-1} = \Delta_n + 3 \iff \Delta_n = 2\Delta_{n-1} - 3 = 2^i \Delta_{n-i} - 3\sum_{k=0}^{i-1} 2^k$$

Per $i = n$ si ha

$$\Delta_n = 2^n \Delta_0 - 3\sum_{k=0}^{n-1} 2^k = 2^n \Delta_0 - 3\frac{1-2^n}{1-2} = 2^n \Delta_0 + 3(1-2^n)$$

Per (1) si ottiene che

$$\mu_{a+b} - \mu_0 = \sum_{i=0}^{a+b-1} \Delta_i = \sum_{i=0}^{a+b-1} 2^i \Delta_0 + 3(1-2^i) = 3(a+b) + 3(1-2^{a+b}) - \Delta_0(1-2^{a+b}) = 0 \iff \Delta_0 = \frac{3(a+b+1-2^{a+b})}{1-2^{a+b}}$$

Quindi

$$\Delta_n = 2^n \Delta_0 + 3(1-2^n) = 2^n \frac{3(a+b+1-2^{a+b})}{1-2^{a+b}} + 3(1-2^n)$$

Troviamo il valore di interesse

$$\mu_a = \sum_{n=0}^{a-1} \Delta_n = \sum_{n=0}^{a-1} 2^n \frac{3(a+b+1-2^{a+b})}{1-2^{a+b}} + 3(1-2^n) = (2^a - 1) \cdot \frac{3(a+b+1-2^{a+b})}{1-2^{a+b}} + 3a + 3 \cdot (1-2^a) = 3(a - (a+b) \cdot \frac{1-2^a}{1-2^{a+b}})$$